УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА ЧАЧАК

Др Јасмина Ђ. Новаковић

РЕДУКЦИЈА ДИМЕНЗИОНАЛНОСТИ ПОДАТАКА У КЛАСИФИКАЦИОНИМ ПРОБЛЕМИМА ВЕШТАЧКЕ ИНТЕЛИГЕНЦИЈЕ

Докторска дисертација

Крагујевац, 2013. година

I. Аутор	
Име и презиме:	**Јасмина Новаковић**
Датум и место рођења:	**07.07.1965, Лозница**
Садашње запослење:	**Београдска пословна школа - Висока школа струковних студија**
II. Докторска дисертација	
Наслов: **РЕДУКЦИЈА ДИМЕНЗИОНАЛНОСТИ ПОДАТАКА У КЛАСИФИКАЦИОНИМ ПРОБЛЕМИМА ВЕШТАЧКЕ ИНТЕЛИГЕНЦИЈЕ**	
Број страница: **220**	
Број слика: **139**	
Број библиографских података: **86**	
Установа и место где је рад израђен:	**Факултет техничких наука Чачак**
Научна област (УДК):	**Вештачка интелигенција**
Ментор:	**Проф. др Алемпије Вељовић**
III. Оцена и обрана	
Датум пријаве теме:	**22.02.2012. године**
Број одлуке и датум прихватања докторске дисертације:	
Број: **852/2**	Датум: **15.05.2013. године**

Комисија за оцену подобности теме и кандидата:

1. Др Алемпије Вељовић, редовни професор Факултет техничких наука Чачак Универзитета у Крагујевцу на наставним предметима Менаџмент информациони системи и Менаџмент развоја, научна област Техничко технолошке науке, ужа научна област Менаџмент информациони системи.

2. Др Живадин Мицић, редовни професор Факултет техничких наука Чачак Универзитета у Крагујевцу на наставним предметима Информационе технологије и системи, научна област Техничко технолошке науке, ужа научна област Информационе технологије.

3. Др Драган Милановић, редовни професор Машинског факултета Универзитета у Београду на наставним предметима Пословно производни инфомациони системи, Увод у индустријско инжењерство, Организација производње, научна област Техничко технолошке науке, ужа научна област Индустријско инжењерство.

4. Др Божидар Раденковић, редовни професор Факултета организационих наука Универзитета у Београду на наставним предметима Симулација и симулациони језици, Симулација у пословном одлучивању, Управљање и динамика организационих система, Методе развоја информационих система,

Конкурентно програмирање, Електронско пословање, Интернет технологије, Мобилно рачунарство, ужа научна област Информационе технологије.

5. Др Данијела Милошевић, ванредни професор Факултет техничких наука Чачак, Универзитет у Крагујевцу на наставним предметима Основе рачунарства и информатике, Базе података, научна област Техничко технолошке науке, ужа научна област Информационе технологије.

Комисија за оцену докторске дисертације:

1. Др Алемпије Вељовић, редовни професор Факултет техничких наука Чачак Универзитета у Крагујевцу на наставним предметима Менаџмент информациони системи и Менаџмент развоја, научна област Техничко технолошке науке, ужа научна област Менаџмент информациони системи.

2. Др Живадин Мицић, редовни професор Факултет техничких наука Чачак Универзитета у Крагујевцу на наставним предметима Информационе технологије и системи, научна област Техничко технолошке науке, ужа научна област Информационе технологије.

3. Др Драган Милановић, редовни професор Машинског факултета Универзитета у Београду на наставним предметима Пословно производни инфомациони системи, Увод у индустријско инжењерство, Организација производње, научна област Техничко технолошке науке, ужа научна област Индустријско инжењерство.

4.Др Божидар Раденковић, редовни професор Факултета организационих наука Универзитета у Београду на наставним предметима Симулација и симулациони језици, Симулација у пословном одлучивању, Управљање и динамика организационих система, Методе развоја информационих система, Конкурентно програмирање, Електронско пословање, Интернет технологије, Мобилно рачунарство, ужа научна област Информационе технологије.

5. Др Данијела Милошевић, ванредни професор Факултет техничких наука Чачак, Универзитет у Крагујевцу на наставним предметима Основе рачунарства и информатике, Базе података, научна област Техничко технолошке науке, ужа научна област Информационе технологије.

Комисија за одбрану докторске дисертације:

1. Др Алемпије Вељовић, редовни професор Факултет техничких наука Чачак Универзитета у Крагујевцу на наставним предметима Менаџмент информациони системи и Менаџмент развоја, научна област Техничко технолошке науке, ужа научна област Менаџмент информациони системи.

2. Др Живадин Мицић, редовни професор Факултет техничких наука Чачак Универзитета у Крагујевцу на наставним предметима Информационе технологије и системи, научна област Техничко технолошке науке, ужа научна област Информационе технологије.

3. Др Драган Милановић, редовни професор Машинског факултета Универзитета у Београду на наставним предметима Пословно производни инфомациони системи, Увод у индустријско инжењерство, Организација производње, научна област Техничко технолошке науке, ужа научна област Индустријско инжењерство.

4.Др Божидар Раденковић, редовни професор Факултета организационих наука Универзитета у Београду на наставним предметима Симулација и симулациони језици, Симулација у пословном одлучивању, Управљање и

динамика организационих система, Методе развоја информационих система, Конкурентно програмирање, Електронско пословање, Интернет технологије, Мобилно рачунарство, ужа научна област Информационе технологије. **5. Др Данијела Милошевић**, ванредни професор Факултет техничких наука Чачак, Универзитет у Крагујевцу на наставним предметима Основе рачунарства и информатике, Базе података, научна област Техничко технолошке науке, ужа научна област Информационе технологије.
Датум одбране дисертације:

ЗАХВАЛНИЦА

При изради овог рада имала сам несебичну помоћ свог ментора проф. др Алемпија Вељовића, редовног професора Факултета техничких наука Чачак Универзитета у Крагујевцу, којем се најсрдачније захваљујем за суштинске савете и свесрдну подршку.

Желим и да се захвалим члановима комисије др Живадину Мицићу, редовном професору Факултета техничких наука Чачак Универзитета у Крагујевцу, др Драгану Милановићу, редовном професору Машинског факултета Универзитета у Београду, др Божидару Раденковићу, редовном професору Факултета организационих наука Универзитета у Београду и др Данијели Милошевић, ванредном професору Факултета техничких наука Чачак Универзитета у Крагујевцу који су ми пружили корисне савете и сугестије при изради ове докторске дисертације.

На пруженој подршци такође се захваљујем и својој породици која ми је пружила безусловну подршку и охрабрење током писања докторске дисертације.

Чачак, 2013. године *др Јасмина Ђ. Новаковић*

САДРЖАЈ

РЕЗИМЕ

Средишњи проблем машинског учења је идентификовање репрезентативног сета података на основу кога ће се конструисати класификациони модел за сваки појединачни задатак. У овој докторској дисертацији истражујемо проблем редукције димензионалности података у класификационим проблемима вештачке интелигенције коришћењем различитих метода за селекцију и екстракцију атрибута. Методе селекције атрибута обухватају: методе филтрирања, методе претходног учења и уграђене методе. Основна хипотеза је да је могуће знатно побољшати перформансе система за индуктивно учење правила у проблемима класификације, применом различитих метода и техника редукције димензионалности података. Евалуација сваког атрибута у сету података врши се на основу предложеног генеричког модела за селекцију и вредновање сваког појединачног атрибута.

У раду, биће предложен велики број алгоритама који се користе у редукцији димензионалности података и биће извршена њихова евалуација на вештачким и природним скуповима података. За потребе класификације користи се велики број алгоритама: *к*-најближи суседи, Бајесови класификатори, стабла одлучивања, вештачке неуронске мреже и потпорни вектори.

Експериментални резултати показују да се овим методама могу брзо идентификовати неважни, редундантни атрибути, као и шум у подацима ако он постоји; као и они атрибути који су значајни за изучавану појаву. У раду се истражује утицај редукције димензионалности података на изградњу модела, што је нарочито значајно када имамо велики број атрибута и велики број инстанци, што је чест случај посебно у биоинформатици, анализи докумената, слика и гласа. У раду, биће разматран утицај метода за селекцију и екстракцију атрибута на рад сваког појединачног алгоритма за класификацију, без обзира да ли он већ има уграђене методе за предселекцију атрибута. Ако алгоритам већ има уграђену предселекцију атрибута, биће истражена добит од независне предселекције атрибута.

Кључне речи: екстракција атрибута, класификација, методе филтрирања, методе претходног учења, редукција димензионалности података, вештачка интелигенција, уграђене методе.

ABSTRACT

The central problem of machine learning is to identify a representative set of data to construct a classification model for each individual task. In this doctoral dissertation, we investigate the problem of dimensionality reduction of data in the classification problems of artificial intelligence using different methods for selecting and extracting attributes. Methods of selection attributes include: filter, wrappers and embedded methods. The basic hypothesis is that it is possible to significantly improve the performance of the system for inductive learning of rules for classification problems, using different methods and techniques for data dimensionality reduction. The evaluation of each attribute in the data set is based on the proposed generic model for the selection and evaluation of each attribute.

This paper will be proposed a number of algorithms that are used in reducing the dimensionality of the data and their evaluation will be performed on artificial and natural data sets. For the purposes of classification is used a large number of algorithms: k-nearest neighbors, Bayesian classifiers, decision trees, artificial neural networks and support vector machine.

The experimental results show that these methods can quickly identify irrelevant or redundant attributes, as well as noise in the data, if it exists; also those attributes that are important for the studied problem. This paper will examines the impact of dimensionality reduction of data to build the model, which is especially important when we have a large number of attributes and a large number of instances, which is often the case, especially in bioinformatics, analysis of documents, images and voice. In this paper, the impact of methods for selection and extraction of the attributes will be considered for each algorithm for classification, regardless of whether they already have a built-in method for preselection of attributes. If the algorithm already has a built-in method for preselection of attributes, will be investigated the influence of an independent selection of attributes.

Keywords: attribute extraction, classification, filter methods, wrapper methods, data dimensionality reduction, artificial intelligence, embedded methods.

ПРЕГЛЕД СЛИКА

ПРЕГЛЕД ТАБЕЛА

ПРВИ ДЕО

1. УВОДНА РАЗМАТРАЊА

Живимо у информационом друштву у коме је прикупљање података једноставно, а њихово складиштење није скупо. Аутори Piatetsky-Shapiro и Frawley наводе да се износ ускладиштених информација удвостручује сваких двадесет месеци [Piatetsky-Shapiro, Frawley, 1991]. Нажалост, иако се повећава количина ускладиштених информација, способност да се исте разумеју и користите није у складу са њиховим повећањем. Машинско учење обезбеђује алате којима се велике количине података могу аутоматски анализирати. Једна од основа машинског учења је селекција атрибута. Селекцијом атрибута и идентификовањем најзначајнијих атрибута за учење, учећи алгоритми се усресређују на оне аспекте података који су најкориснији за анализу и будућа предвиђања. Хипотеза која се доказује у овом раду је да је могуће знатно побољшати перформансе система за индуктивно учење правила у проблемима класификације, применом различитих метода и техника редукције димензионалности података. Различите методе за селекцију атрибута примењене су у великом броју алгоритама за класификацију. У већини случајева, процес селекције атрибута је једноставан и брзо се извршава. Он омогућава елиминацију ирелевантних и редундантних података, и у многим случајевима, побољшава перформансе учећих алгоритама.

1.1. Предмет и хипотезе докторске дисертације

Вештачка интелигенција је једна од области рачунарства која се последњих деценија најбрже развија, а развој је одувек био заснован на комплементарном повезивању теорије и експеримената. Будући развој ове области рачунарства захтева проширивање и учвршћивање теоријских знања, пре свега математичких, али и знања о специфичностима области примене, као и њихову адекватну формализацију.

Предмет истраживања у докторској дисертацији је редукција димензионалности података применом претходне селекције атрибута уз помоћ метода филтрирања, претходног учења, уграђених метода и екстракције атрибута у

класификационим проблемима вештачке интелигенције. Разматраће се ефективна димензионалност података добијена применом ових метода, да би добили добру репрезентацију података. Предметом истраживања биће обухваћен велики број алгоритама за класификацију, и то коришћењем тачности класификације као мере за њихову евалуацију.

Основна хипотеза докторске дисертације је да је могуће знатно побољшати перформансе система за индуктивно учење правила у проблемима класификације, применом различитих метода и техника редукције димензионалности података.

Докторска дисертација је заснована на следећим релевантним **посебним хипотезама**:

1. Примена претходне селекције атрибута уз помоћ метода филтрирања, претходног учења, уграђених метода и метода екстракције атрибута, код великог броја алгоритама за класификацију доводи до смањења негативних ефеката високе димензионалности података.
2. Прецизно примењене методе претходне селекције атрибута доприносе повећању квалитета генерализације, јер се смањује вероватноћа претераног подешавања модела према тренирајућим подацима.
3. Претходна селекција атрибута уз помоћ метода филтрирања, претходног учења, уграђених метода и метода екстракције атрибута доводи у неким случајевима до значајног смањења времена за изградњу модела.
4. Применом метода филтрирања, претходног учења, уграђених метода и метода екстракције атрибута у оквиру система за индуктивно учење, могуће је у неким случајевима значајно побољшати тачност постојећих метода учења.

1.2. Преглед стања у подручју истраживања

Од 1970-тих година смањење димензионалности података је плодно тло за истраживање и развој, и то у статистичком препознавању облика [Wyse, 1980; Ben-Bassat, 1982], машинском учењу и *data mining*-у. Оно данас представља активно поље истраживања у рачунарству.

Смањење димензионалности података је фундаментални проблем у многим областима, нарочито у предвиђању, класификацији докумената, биоинформатици, препознавању објеката или у моделирању сложених технолошких процеса. У таквим

апликацијама, скупови података са хиљадама атрибута нису неуобичајени. За неке проблеме сви атрибути могу бити важни, али за неке друге проблеме само мали подскуп атрибута је обично релевантан.

Да би се превазишли проблеми које са собом носи висока димензионалност података, димензионалност података би требало да буде смањена. Ово се може урадити тако што се изабере само подскуп релевантних атрибута, или креирањем нових атрибута који садрже максимум информација о датој класи. Прва методологија се зове селекција атрибута, док се друга зове екстракција атрибута, и обухвата линеарне (PCA, Independent Component Analysis (ICA) и сл.) и нелинеарне методе екстракције атрибута. Проналажење нових подсупова атрибута је обично слабо решив проблем, као и многи проблеми у вези са екстракцијом атрибута који су се показали као *NP-hard* [Blum и Rivest, 1992].

Неки алгоритами класификације су наследили способност да се фокусирају на релевантне карактеристике и игноришу оне ирелевантне. Стабла одлучивања су пример такве класе алгоритама [Breiman *et al*., 1984; Quinlan, 1993], али и вишеслојни перцептрон (енг. *Multilayer Perceptron* - MLP) са јаким регулисањем улазног слоја који може искључити небитне атрибуте на аутоматски начин [Duch *et al*., 2001]. Такође, и такве методе могу имати користи од независне селекције или екстракције атрибута.

С друге стране, неки алгоритми немају могућност одабира или екстракције атрибута. Алгоритам *к*-најближег суседа (енг. K-*nearest neighbour* - k-NN) је једна породица таквих метода које се у процесу тренирања података, снажно ослања на методе одабира или екстракције релевантних и нередунтантних атрибута.

1.3. Значај и циљ истраживања са становишта актуелности у одређеној научној области

Методе филтрирања, претходног учења, уграђене методе и екстракције атрибута се могу користити за смањење димензионалности података. Оне су се показале као изузетно користан инструмент за смањење димензионалности мултиваријационих података са многим областима примене у анализи слике, препознавању облика и изгледа, компресији података, предвиђању временских серија, и анализи биолошких података - да поменемо само неке. Снаге ових метода могу проистицати из њиховог

ефикасног рачунарског механизма, или из чињенице да се добро разумеју, или из њихове опште применљивости.

Предложена истраживања у докторској дисертацији ће првенствено допринети бољем познавању ефеката и могућности примене ових метода у проблемима класификације. **Значај истраживања** у докторској дисертацији се огледа у чињеници да се ради о оригиналном истраживању ових метода са аспекта могућности њихове примене у различитим областима надгледаног и ненадгледаног учења. Такође, резултати истраживања се у великој мери могу користити за практичну примену чиме би се у значајној мери редуковале грешке и максимизирали позитивни ефекти примене наведених метода и алгоритама за класификацију.

Основни **циљ истраживања** у докторској дисертацији је да се идентификују, имплементирају и експериментално провере методе и технике које су посебно погодне за класификационе проблеме надгледаног учења. У односу на дефинисани основни циљ истраживања могуће је дефинисати и неколико циљева нижег ранга који се у првом реду односе на подешавање параметара сваке од метода тако да она максимизира своје ефекте, као и мерење утицаја ових метода на рад појединачних алгоритама за класификацију. Значајан циљ истраживања се односи и на испитивање могућности евалуације ових метода на основу одређених квалитативних и квантитативних показатеља. Поред наведеног, циљ истраживања је и дефинисање услова примене ових метода у класификационим проблемима у техници, биомедицини, обради слике и звука на начин који омогућава унапређење перформанси.

1.4. Методи истраживања

У сагледавању кључних изазова и трендова у редукцији димензионалности података код надзираног и ненадзираног учења коришћен је метод анализе који подразумева детаљну анализу примењених метода у различитим областима истраживања. Метод синтезе се користити у сврху добијања општих ставова и извођења одређених закључака у вези са могућностима и ефектима примене метода филтрирања, претходног учења, уграђених метода и екстракције атрибута. У домену евалуације метода редукције димензионалности података и алгоритама за класификацију у значајној мери коришћени су и квантитативни методи који подразумевају употребу одређених показатеља. Метод компарације се користи за

утврђивање оправданости примене одређених метода за редукцију димензионалности података и алгоритама за класификацију који су предмет истраживања и то кроз сагледавање тачности класификације као мере за њихову евалуацију.

Поред наведеног, у раду је коришћено индуктивно закључивање за извођење генералних ставова и дедукција која закључивањем од општег ка посебном обезбеђује закључке о применљивости предложених стратегија у конкретним ситуацијама. Прихватање или одбацивање формулисаних хипотеза спроведено је на реалним и вештачким *data set*-овима, преузетим из репозиторијума за потребе машинског учења, а који су намењени истраживачима (UCI Machine Learning Repository [Frank и Asuncion, 2010]), како би се добијени подаци могли упоредити са подацима које су други истраживачи добили.

У циљу лакшег и бољег сагледавања веза и односа кључних варијабли и њихове прегледније упоредивости коришћене су одговарајуће табеларне и графичке презентације.

1.5. Очекивани резултати докторске дисертације

У теоријском смислу, основни резултати докторске дисертације ће се односити на потврду формулисаних хипотеза. У односу на основну хипотезу докторске дисертације, да је могуће знатно побољшати перформансе система за индуктивно учењење правила у проблемима класификације, применом различитих метода и техника редукције димензионалности података. У односу на прву хипотезу, то би значило потврду става да примена претходне селекције атрибута уз помоћ метода филтрирања, претходног учења и уграђених метода код великог броја алгоритама за класификацију доводи до смањења негативних ефеката високе димензионалности података. У односу на другу хипотезу, потребно је потврдити став да прецизно примењене методе селекције атрибута доприносе повећању квалитета генерализације, јер се смањује вероватноћа претераног подешавања модела према тренирајућим подацима. У односу на трећу хипотезу, резултат дисертације би подразумевао потврду става да претходна селекција атрибута уз помоћ метода филтрирања, претходног учења и уграђених метода доводи до значајног смањења времена за изградњу модела код различитих класификатора. У односу на четврту хипотезу, резултат дисертације би подразумевао потврду става да применом метода филтрирања, претходног учења и

уграђених метода у оквиру система за индуктивно учење је могуће значајно побољшати тачност постојећих метода учења на основу примера у проблемима класификације.

У практичном смислу, основни резултат дисертације ће се односити на примену метода филтрирања, претходног учења и уграђених метода код алгоритама за класификацију, и то у великом броју области: бионформатици, анализи слике и звука и сл. Повезивањем резулата емпиријског истраживања, које ће бити спроведено током израде докторске дисертације, са постојећим резултатима истраживања, добиће се практичне смернице за примену метода филтрирања, претходног учења, уграђених метода и екстракције атрибута у најразличитијим задацима машинског учења. Резултати добијени емпиријским истраживањем ће показати у којој мери примена метода филтрирања, претходног учења и уграђених метода има утицај на тачност класификације великог броја примењених алгоритама.

1.6. Оквирни садржај дисертације

Истраживања у првом делу докторске дисертације су усмерена на предмет и циљ истраживања, постављање основних хипотеза, на методе истраживања и значај самог истраживања.

Истраживања у другом делу докторске дисертације су усмерена на изазове надгледаног и ненадгледаног учења и сагледавања елемената дизајна система који учи. Посебна пажња посвећена је циљној функцији, избору простора хипотеза, избору алгоритма и мери квалитета учења.

У трећем делу дисертације, предмет истраживања су методе редукције димензионалности података. Разматра се корелација међусобно независних и зависних атрибута с концептом, и извршена је класификација атрибута у четири дисјунктне класе: ирелевантни атрибути, слабо релевантни редундантни атрибути, слабо релевантни нередундантни атрибути и јако релевантни атрибути. Присуство ирелевантних и редундантних атрибута негативно утиче на перформансе индуктивног учења, због чега оптималан скуп атрибута за учење чине слабо релевантни нередундантни атрибути и јако релевантни атрибути. Услед потребе анализе метода селекције атрибута, предмет посебног истраживања су методе филтрирања, методе претходног учења, уграђене методе и методе екстракције атрибута. Посебан предмет

разматрања су и карактеристике алгоритама за селекцију атрибута, као што су: *Information Gain* (IG), *Gain Ratio* (GR), *Symmetrical Uncertainty* (SU), *Relief-F* (RF), *One-R* (OR) и *Chi-Squared* (CS).

У четвртом делу дисертације разматрају се мере за евалуацију класификцијских модела као и методе за оцену стварне фреквенције грешака класификацијског модела, које се разликују по приступу проблему и својствима које показују. Такође, приликом тренинга постоји могућност да се модел превише прилагоди специфичностима података за тренинг и да због тога даје лоше резултате када се примени на другим подацима, због чега се овај проблем посебно разматра.

У следећем петом делу дисертације, разматра се проблем класификације, који представља разврставања непознате инстанце у једну од унапред понуђених категорија. У овом делу анализирају се класификациони алгоритми, који су коришћени у експерименталним истраживањима за доказ постављених хипотеза. То су следећи алгоритми надзираног учења за изградњу модела: IBk, *Naïve Bayes*, SVM, J48 стабло одлучивања и RBF мрежа.

У шестом делу дисертације дат је приказ изабраних проблема учења, које у експерименталном истраживању користимо за доказ постављених хипотеза.

Седми део рада даје приказ коришћене методологије извођења експеримента и подешавања параметара модела. Разматра се тачност и прецизност којима меримо успешност добијеног модела, као и статистички тестови које користимо у истраживањима, са посебним освртом на стандардну девијацију и *t*-тест.

У следећем осмом делу дисертације, након разматрања поставки експерименталног истаживања, приказани су резултати истраживања за различите методе филтрирања, и то за сваки класификациони алгоритам посебно.

Девети део дисертације даје разматрање поставки експерименталног истаживања, приказ резултата истраживања за различите методе претходног учења за сваки класификациони алгоритам посебно.

У десетом делу дисертације, разматране су поставке експерименталног истаживања и приказани су резултати истраживања за екстракцију атрибута уз помоћ PCA методе за сваки класификациони алгоритам посебно.

У последњем делу докторске дисертације, дат је резиме рада, потом закључци разматрања о утицају претходне селекције атрибута на класификацијске перформансе

алгоритама надзираног учења. На крају, приказани су правци могућих даљих истраживања у овој области.

ДРУГИ ДЕО

2. ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА И МАШИНСКО УЧЕЊЕ: КОНЦЕПТИ И ДЕФИНИЦИЈЕ

У другом делу дисертације, биће речи о вештачкој интелигенцији и машинском учењу као области вештачке интелигенције која се бави изградњом прилагодљивих рачунарских сиситема који су способни да побољшавају своје перформансе користећи информације из искуства.

2.1. Изазови машинског учења

Једна од области рачунарства која се последњих деценија најбрже развија је **вештачка интелигенција**. За неке области рачунарства се сматра да су заокружене и у њима се не очекују нови значајни продори, али од вештачке интелигенције се резултати тек очекују, упркос томе што су већ развијени многи „интелигентни“ системи који функционишу изузетно добро. На овај начин, вештачка интелигенција добија на атрактивности, а нова теоријска истраживања и експерименти представљају пут ка новим применама у најразличитијим областима. Развој ове области је одувек био заснован на комплементарном повезивању теорије и експеримената, тако да и будући развој захтева проширивање и учвршћивање теоријских знања, пре свега математичких, али и знања о специфичним областима примене, као и њихову адекватну формализацију.

Област вештачке интелигенције обухвата два приступа вештачком учењу [Hutchinson, 1993]. Први је мотивисан проучавањем менталних процеса и каже да је вештачко учење проучавање алгоритама садржаних у људском уму. Циљ је открити како се то алгоритми могу превести у формалне језике и рачунарске програме. Други приступ је мотивисан са практичног становишта рачунара и има мање грандиозне циљеве. Он подразумева развој програма који уче из претходних података, и као такав је грана обраде података. Машинско учење, као напред наведени други приступ вештачком учењу, нагло се развијао од свог настанка средином седамдесетих година.

Машинско учење је област вештачке интелигенције која се бави изградњом прилагодљивих рачунарских сиситема који су способни да побољшавају своје перформансе користећи информације из искуства. Машинско учење је дисциплина која се бави проучавањем генерализације и конструкцијом и анализом алгоритама који генерализују. Прва теоријска разматрања машинског учења појавила су се касних 60-их у радовима Голда, али универзалне теоријске основе су се почеле учвршћивати тек током 80-их година прошлог века. У овој области, најважнији теоријски приступи су Голдов модел граничног учења (енг. *learning in the limit*), *Valiant*-ov PAC (енг. *Probably Approximately Correct*) модел и вероватно најкомплетнија — статистичка теорија учења.

Машинско учење је занимљиво и због своје тежње да се приближи људском учењу по ефикасности, као и да га објасни, односно пружи теоријски модел за њега. Нека од најважнијих питања машинског учења су [Јаничић и Николић, 2010]:

- Шта се може научити и под којим условима?
- Како се повећава ефикасност учења у зависности од обима искуства?
- Који су алгоритми погодни за које врсте проблема?

Одговоре на напред наведена најважнија питања машинског учења треба тражити кроз теоријске моделе учења у оквиру којих се у овом погледу већ дошло до значајних резултата. Практични резултати су често претходили теоријским, а разлог би лако могао бити тај што је ова област дубоко мотивисана практичним применама. У машинском учењу постигнути су добри резултати у многим областима, као што је препознавање говора, препознавање руком писаног текста, вожња аутомобила и слично. Али ма колико примене машинског учења биле разноврсне, постоје задаци који се често понављају. Зато је могуће говорити о врстама задатака учења које се често појављују. Један од најчешћих задатака учења који се јавља у пракси је класификација. Класификација представља препознавање врсте објеката, нпр. да ли одређено ткиво представља малигно ткиво или не. Регресија је задатак машинског учења у коме објектима одговарају вредности из скупа реалних бројева, као што је нпр. предвиђање потражње робе у зависности од разних фактора који на њу утичу.

Основном карактеристиком интелигентног понашавања може се сматрати **дедуктивно закључивање** вођено законима логике. Дедуктивно закључивање један је од основних начина закључивања код људи. Друга битна карактеристика интелигентног понашања је прилагођавање понашања јединке околини у којој се она

налази, која се може уочити и код живих организама. Путем еволутивних процеса, прилагодљивост се постиже и код нижих организама, али је ова способност са тачке гледишта вештачке интелигенције посебно занимљива код животиња и људи код којих се манифестује у току живота јединке. У току живота јединке, прилагођавање се постиже учењем на основу примера из искуства и применом научног знања у сличним ситуацијама у будућности.

Такође, могуће је говорити о доношењу закључака о непознатим случајевима, на основу знања о неким другим познатим случајевима. **Генерализација** или индуктивно закључивање је процес у коме се знање које важи за неки скуп случајева преноси на неки његов надскуп. На овај начин, генерализација представља један од основних концепата машинског учења. Са концептом генерализације је директно повезан концепт апстракције. Да би генерализација била успешна, одређени аспекти ентитета о којима се резонује морају бити занемарени уколико нису од суштинског значаја за генерализацију. Зато је једна од кључних тема у теоријском разматрању машинског учења контрола генерализације и апстракције.

Генерализација је један од основних начина за формирање представа о окружењу, ситуацијама или узрочно последичним односима, односно за прављење модела података из искуства. Ако су у неком домену грешке у закључивању прихватљиве, онда алгоритми генерализације омогућавају закључивање и без темељног познавања и комплетног формалног описивања домена на који се примењују. Некада алгоритми индуктивног закључивања могу бити ефикаснији и од алгоритама дедуктивног закључивања.

Постоји неколико разлога зашто системе машинског учења треба користити. У изучавању многих појава, ови системи су корисни у случајевима: где алгоритамска решења нису на располагању, где постоји недостатак формалних модела, или је ограничена стручност у разумевању сложених функција. Они имају потенцијал за откривање нових односа међу појмовима и хипотезама испитујући записе успешно решених предмета и могу гомилати знање које тек треба да буде формализовано.

2.1.1. Надгледано и ненадгледано учење

У машинском учењу постоје две главне формулације проблема учења, и то:

- Надгледано учење представља приступ проблему учења који се односи на ситуације у којима се алгоритму заједно са подацима из којих учи дају и жељени излази.
- Ненадгледано учење представља приступ проблему учења који се односи на ситуације у којима се алгоритму који учи пружају само подаци без излаза, а од алгоритма који учи очекује се да сам уочи неке законитости у подацима који су му дати.

Као пример надгледаног учења, већ је поменута класификација ткива на малигна и она која то нису. Пример ненадгледаног учења је тзв. кластеровање, односно уочавање групе сличних објеката када не знамо унапред колико група постоји и које су њихове карактеристике. У овом случају, ткива се могу кластеровати по њиховој сличности.

2.1.2. Циљна функција и хипотезе

У машинском учењу, оно што је потребно научити се дефинише циљном функцијом. Циљна функција дефинише жељено понашање система који учи. Ако лекар жели да препозна малигна ткива код пацијента, циљна функција таквим ткивима придружује 1, а осталим -1.

При учењу су грешке могуће и чак сасвим извесне, па тако учење представља приближно одређивање ове циљне функције, односно може бити виђено као апроксимирање функција. Моделима података или хипотезама називамо функцију којом апроксимирамо циљну функцију. У нашем примеру препознавања ткива модел може бити нпр. функција $sgn(ax + by + c)$ која је придруживала 1 свим тачкама са једне стране праве, а -1 тачкама са друге.

Простором хипотеза називамо скуп свих допустивих хипотеза. Потенцијалне репрезентације хипотеза су разноврсне, и оне могу представљати линеарне функције, правила облика IF...THEN и сл. У нашем примеру препознавања ткива хипотезе су репрезентоване правама дефинисаним преко вредности коефицијената a, b и c.

2.1.3. Налажење хипотезе

Налажење хипотезе која најбоље апроксимира циљну функцију можемо видети као претрагу простора хипотеза која је вођења подацима, а коју реализује алгоритам

учења. За квалитет учења је од фундаменталног значаја избор простора хипотеза. Иако изгледа парадоксално, претерано богатство простора хипотеза по правилу доводи до лошијих резултата, о чему ће бити дискутовано у наставку текста.

2.1.4. Подаци за тренинг и тестирање

Инстанце или примерци се у рачунару представљају у облику који је погодан за примену алгоритама учења. Код алгоритама машинског учења, најпогоднији начин за представљање инстанци је помоћу неких њихових својстава, односно атрибута (енг. *feature*, *attribute*). Та својства или атрибути представљају карактеристике инстанци, тако да сваки од изабраних атрибута може имати вредност која припада неком унапред задатом скупу. Вредности атрибута су често нумеричке, али могу бити и категоричке, односно могу представљати имена неких категорија којима се не могу једнозначно доделити смислене нумеричке вредности или уређење. У машинском учењу, када су изабрани атрибути помоћу којих се инстанце описују, онда се свака инстанца може представити вектором вредности атрибута које јој одговарају.

У машинском учењу, подаци на основу којих се врши генерализација, називају се **подацима за тренинг**, а њихов скуп тренинг скуп. С обзиром да тестирање наученог знања на подацима на основу којих је учено, обично доводи до значајно бољих резултата од оних који се могу касније добити у применама, потребно је пре употребе проценити квалитет наученог знања. Ово се обично постиже тако што се разматра колико је научено знање у складу са неким унапред датим подацима за тестирање. Тест скуп чине подаци за тестирање. Тест скуп треба да буде такав да је дисјунктан са тренинг скупом.

2.2. Елементи дизајн система који учи

Елементи дизајна система који учи су [Јаничић и Николић, 2010]:

- формулација проблема учења: надгледано или ненадгледано учење,
- запис примера,
- избор циљне функције,
- избор простора хипотеза,
- избор алгоритма,

- избор мера квалитета учења.

У већ поменутом класификовању ткива на малигна и она која то нису, разматраћемо могуће елементе дизајна система који учи. Као пример, можемо узети 1000 ткива која су разврстана у две унапред фиксиране категорије (бенигни и малигни), тако да је задатак учења у овом случају формулисан као задатак надгледаног учења.

За запис примера, можемо узети обележје ткива које се састоји од 12 импеданси мереним на различитим фреквенцијама.

Избор циљне функције може бити извршен нпр. тако да циљна функција f придружује вредност 1 малигним ткивима, а -1 осталим.

Избор простора хипотеза може бити извршен тако да нпр. простор хипотеза одговара скупу свих правих у одговарајућем простору. Хипотезе су функције које придружују вредност 1 тачкама са једне стране праве, а -1 тачкама са друге стране праве. Хипотезе се бирају избором вредности коефицијената a, b и c.

Избор алгоритма може бити такав да алгоритам учења представља нпр. градијентни спуст за минимизацију одступања између вредности циљне функције и хипотезе на датим примерима.

За меру квалитета учења може бити узет нпр. удео тачно класификованих ткива.

ТРЕЋИ ДЕО

3. РЕДУКЦИЈА ДИМЕНЗИОНАЛНОСТИ ПОДАТАКА

У трећем делу дисертације, разматра се проблем редукције димензиналности података. Редукција димензионалности података је активно поље у компјутерским наукама. У појединим апликацијама скупови података са хиљадама атрибута нису реткост. Сви атрибути могу бити значајни за неке проблеме, али за неке циљане намере само мали подскуп атрибута је обично релевантан.

3.1. Појам редукције димензионалности података

Селекција атрибута се може дефинисати као процес који бира минимални подскуп M атрибута из изворног скупа N атрибута, тако да је простор атрибута оптимално смањен према одређеном критерију оцењивања. Како се димензионалност домена шири, број атрибута N се повећава. Проналажење најбољег подскупа атрибута је обично нерешив проблем [Kohavi и John, 1997] и многи проблеми везани одабир атрибута су се показали да су *NP*-тешки [Blum и Rivest, 1992].

Селекција атрибута је активно поље истраживања у рачунарској науци. То је плодно поље за истраживање и развој од 1970 година у статистичком распознавање узорака [Wyse *et al.*, 1980; Ben-Bassat, 1982; Siedlecki и Sklansky, 1988], машинском учењу и *data mining*-у [Blum и Langley, 1997; Dash и Liu, 1997; Dy и Brodley, 2000; Kim *et al.*, 2000; Das, 2001; Mitra *et al.*, 2002].

Селекција атрибута је основни проблем у многим различитим подручјима, посебно у предикцији, класификацији, биоинформатици, препознавању објеката или у моделирању сложених технолошких процеса [Quinlan, 1993; Doak, 1992; Talavera, 1999; Liu и Motoda, 1998]. Скупови података са хиљадама атрибута нису реткост у таквим апликацијама. Сви атрибути могу бити важни за неке проблеме, али за нека циљана истраживања, само мали подскуп атрибута је обично релевантан.

Селекција атрибута смањује димензионалност података, уклања сувишне, неважне податке, или шум у подацима. То доноси непосредне ефекте: убрзање

алгоритама *data mining*-а, побољшање квалитета података, перформансе *data mining*-а, као и повећање разумљивости добијених резултата.

Алгоритми за селекцију атрибута могу се поделити на филтере (енг. *filter*) [Almuallim и Dietterich, 1991; Kira и Rendell, 1992], методе претходног учења (енг. *wrappers*) [Kohavi и John, 1997] и уграђене (енг. *embedded*) приступе [Blum и Langley, 1997]. Филтер методе оцењују квалитет одабраних атрибута, независно од алгоритма за класификацију, док су методе претходног учења методе које захтевају примену класификатора (који би требао бити трениран на одређеном подскупу атрибута) за процену квалитета. Уграђене методе обављају одабир атрибута током учења оптималних параметара (за на пример, неуронске мреже тежине између улазног и скривеног слоја).

Неки алгоритми класификације су наследили способност да се усресреде на релевантне атрибуте и занемарују оне неважне. Стабла одлучивања су репрезентативан пример класе таквих алгоритама [Breiman, 1984; Quinlan, 1993], али такође и MLP неуронске мреже, са јаким регулисањем улазног слоја, могу искључити неважне атрибуте на аутоматски начин [Duch *et al.*, 2001]. Такве методе, такође могу имати користи од независног избора атрибута. С друге стране, неки алгоритми немају начине да изврше избор релевантних атрибута. K-NN алгоритам припада фамилији таквих метода које класификују нове примере проналажењем најближих примерака за тренинг, снажно се ослањајући на методе за селекцију атрибута.

Истраживачи су проучавали различите аспекте селекције атрибута. Претрага је кључна тема у проучавању селекције атрибута [Doak, 1992], као што су почетна тачка претраге, правац претраге, и стратегија претраге. Други важан аспект селекције атрибута је како мерити да ли је одговарајући подскуп добар [Doak, 1992]. Постоје филтер методе [Siedlecki и Sklansky, 1988; Fayyad и Irani, 1992; Liu и Setiono, 1996], методе претходног учења [John *et al.*, 1994; Caruana и Freitag, 1994; Dy и Brodley, 2000], а од недавно и хибридне методе [Das, 2001; Xing *et al.*, 2001]. Према информацијама о класама које су доступне у подацима, постоје надзирани [Xing *et al.*, 2001; Dash и Liu, 1997] и ненадзирани приступи [Dash *et al.*, 1997; Dash и Liu, 1999; Talavera, 1999; Dy и Brodley, 2000].

Главни циљ овог рада је проверити утицај различитих филтер метода, метода претходног учења, уграђених метода и екстракције атрибута на тачност класификације. У раду показујемо да нема једне најбоље методе за редукцију димензионалности

података, и да избор зависи од особина посматраног скупа података и примењених класификатора. У практичним проблемима, једини начин како би били сигурни да је највиша прецизност добијена је тестирање датог класификатора са више различитих подскупова атрибута, добијених различитим методама за селекцију атрибута.

3.2. Ефекти претходне селекције атрибута

Неки од алгоритама за учење у току процеса учења врше селекцију атрибута неком уграђеном методом. Зашто је онда претходна селекција атрибута ипак неопходна, и у овим случајевима? Позитивни ефекти претходне селекције атрибута су [Мишковић, 2008]:

- смањење ефекта високе димензионалности, чиме се поправљају перформансе када се располаже ограниченим бројем примера (за исту прецизност, код повећања броја димензија за k, неопходно је n^{d+k} инстанци-тачака, што је повећање за фактор n^k);
- повећање квалитета генерализације, односно тачности предвиђања на новим примерима, јер је мања вероватноћа претераног подешавања према тренирајућим подацима, посебно у присуству шума;
- повећање разумљивости наученог знања;
- значајно смањење времена рачунања.

Примена претходне селекције атрибута је неопходна нпр. у бионформатици, анализи слике или звука и сл. Селекција атрибута је посебно значајна техника редукције димензионалности, јер чува оригинално значење атрибута, које је разумљиво човеку. Технике трансформације простора атрибута (нпр. Анализа главних компоненти) и технике компресије на основу теорије информација мењају оригинални модел проблема уводећи нове атрибуте који немају разумљиву интерпретацију у контексту проблема који се разматра.

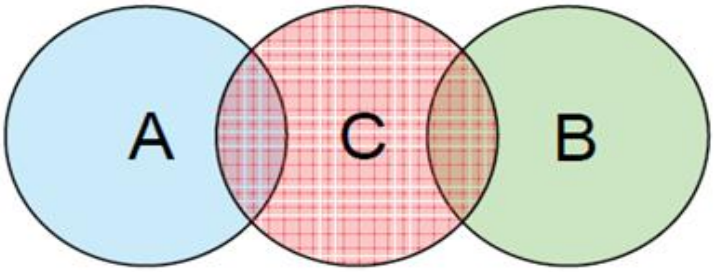
Велики значај у машинском учењу има интеракција атрибута, јер атрибути у реалним концептима и базама података углавном нису независни. Одређени број атрибута у моделу често није у корелацији са концептом и нема исти значај приликом класификације нових инстанци. Обично, услед претераног броја ирелевантних атрибута у моделу долази до претераног прилагођавања тренинг скупу и лоших перформанси учења.

Због тога што алгоритми учења не могу довољно успешно да разреше ове ситуације, посебно у случају великог броја атрибута, приступа се смањењу димензионалности података претходном селекцијом потенцијално релевантних атрибута.

3.3. Корелација међусобно независних и зависних атрибута с концептом

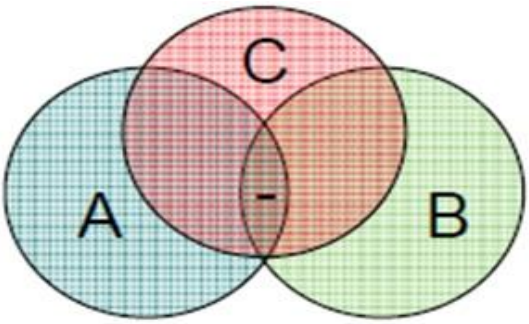
У машинском учењу, претходна селекција подскупа релевантних атрибута зависи од:

- њихове директне корелације са концептом, односно класификационим атрибутом,
- од њихових међусобних интеракција, преко којих атрибута може бити у јакој корелацији са концептом, иако сваки појединачно није у значајнијој директној корелацији с концептом.



$$\boldsymbol{I(AB;C) = I(A;C) + I(B;C)}$$

Слика 3.1: Корелација међусобно независних атрибута с концептом [Мишковић, 2008]



$$\boldsymbol{I(A;B|C) = 0}$$

Слика 3.2: Корелација међусобно зависних атрибута с концептом [Мишковић, 2008]

На слици 3.1, атрибути A и B су међусобно независни, иако оба атрибута дају информацију о класификацији C, они немају ништа заједничко. Овакву ситуацију претпостављају мере значајности атрибута које истовремено мере само значај појединачних атрибута, нпр. *Information Gain*. На слици 3.1, атрибути A и B су међусобно независни у односу на класификацију C и све што атрибути A и B имају заједничко је део информације о класификацији C. Наведену претпоставку о условној независности атрибута имају метод *Naïve Bayes* и *Bayes*-ове мреже. Ако атрибути нису условно независни у односу на ознаку класе, стабла одлучивања су неефикасна. У случају да је $\boldsymbol{I(A;C|B)=0}$, атрибут B је ирелевантан за предвиђање C, што је основа селекције атрибута филтрирањем. Појединачно уклањање атрибута који су у групној корелацији с концептом може значајно смањити перформансе наученог концепта, због чега је неопходно извршити анализу корелација и идентификовати значајне групне корелације. На слици 3.2, приказана је корелација међусобно зависних атрибута с концептом.

3.4. Класификација атрибута

Присуство ирелевантних и редундатних атрибута негативно утиче на перформансе индуктивног учења. Могућа је класификација атрибута у четири дисјунктне класе:

- ирелевантни атрибути,
- слабо релевантни редундантни атрибути,
- слабо релевантни нередундантни атрибути,
- јако релеватни атрибути.

Оптималан скуп атрибута за учење чине атрибути класе слабо релевантни нередундантни атрибути и јако релеватни атрибути.

3.5. Интеракција у селекцији атрибута

Због великог броја комбинација атрибута чије интеракције треба размотрити ($O(2^N)$, где је N број атрибута у моделу [Almuallim и Dietterich, 1991], долази до сложености анализе групних корелација, што је разлог због чега се обично прибегава

апроксимацији. Тако нпр. изврши се само делимична анализа корелације појединих атрибута с класом $O(N)$ или се анализирају само неке од могућих комбинација (интеракције дужине 2 или 3 атрибута).

Аутори Jakulin и Bratko [Jakulin и Bratko, 2004] предлажу откривање интеракција помоћу својства иредуцибилности, јер атрибут губи релевантност када се уклоне атрибути који су с њим у интеракцији. Исти аутори користе статистичку меру значајности за оцену и приказ значајних интеракција у форми графа интеракција.

Аутори [Jakulin и Bratko, 2003; Lavrac *et al.*, 2003] предлажу да се за откривање интеракција користи мера добитка интеракција (енг. *interaction gain*), помоћу кога се могу откривати интеракције атрибута са класом (енг. *2-way*) и два атрибута с класом (енг. *3-way*).

Препознавање присуства интеракција атрибута ради претходне селекције помоћу методе *Relief-F*, користићемо даље у експерименталном истраживању.

Алгоритам *Relief-F* врши оцену атрибута на основу тога како његове вредности из тренирајућег скупа разликују примере који су међусобно слични, односно блиски и врши апроксимацију разлике вероватноћа, што је дато следећим изразом:

$$\mathrm{F(A)=P(различита\ вредност\ A|најближи\ пример\ из\ различите\ класе)\ -\ P(различита\ вредност\ A|најближи\ пример\ из\ исте\ класе)} \quad (3.1)$$

Relief-F, за сваки пример *x*, тражи у тренирајућем скупу два најближа суседа, један из исте, а други из осталих класа (најближи „погодак“ x_{hit} и најближи „промашај“ x_{miss}) и рачуна суму међусобних растојања ових вредности посматраног атрибута A. На основу *n* примера, рачуна се сума растојања вредности за тај атрибут, њихова средња вредност представља оцену атрибута A:

$$F(A) = \frac{1}{n}\sum_1^n -difference(x, x_{hit}) + difference(x, x_{miss}) \quad (3.2)$$

Дистанца је 1 за различите вредности дискретног атрибута, за исте вредности је 0, док је за континуалне атрибуте растојање разлика самих вредности, нормализована на интервал $[0 \ldots 1]$.

Relief-F има два битна побољшања:

- у присуству шума у тренирајућем скупу ради поузданије оцене, користи се просечна удаљеност до *к* примера, уместо удаљености до најближег и најдаљег суседа;
- за случај изостављених вредности у примерима, проширена је дефиниција функција растојања и решен је проблем учења више класа.

Relief-F оцењује и рангира сваки атрибут глобалном функцијом оцене $[-1 \dots 1]$.

3.6. Методе претходне селекције

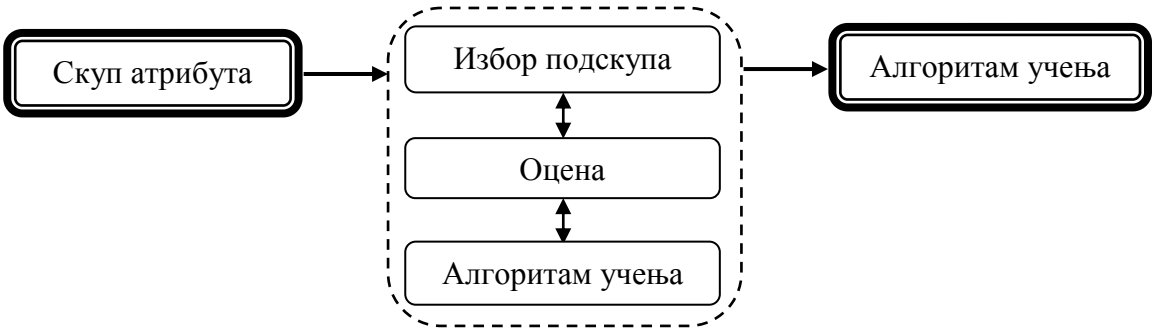
Разноврсне технике рангирања и селекције атрибута су предложене у литератури која обрађује проблематику машинског учења. Сврха ових техника је да одбаце ирелевантне (неважне) или редундантне (сувишне) атрибуте из датог скупа атрибута.

Методе претходне селекције погодног подскупа атрибута деле се на:

- методе филтрирања (енг. *filter methods*),
- методе претходног учења (енг. *wrapper methods*),
- уграђене методе (енг. *embeeded methods*).

Слика 3.3: Модел селекције атрибута филтрирањем, на основу [Kohavi и John, 1997]

Код методе филтрирањем, подскуп атрибута се бира независно од алгоритма учења, на основу неке оцене која рангира све атрибуте, нпр. то може бити коефицијент корелације вредности атрибута са вредностима класификационог атрибута (класе).



Слика 3.4: Метод селекције претходним учењем, на основу [Kohavi и John, 1997]

На слици 3.3. приказан је модел селекције атрибута филтрирањем, а на слици 3.4. метод селекције претходним учењем.

Код методе претходног учења подскуп атрибута се бира према естимацији тачности предвиђања коју дају изабрани класификатор након учења правила за сваки разматрани подскуп. Учење правила се врши након селекције најбоље оцењеног

подскупа. Исцрпно испитивање свих могућих подскупова, прихватљиво је за мали број атрибута, јер је сложеност таквог поступка из класе сложености *NP*-тежак [Kohavi и John, 1997].

У односу на селекцију претходним учењем, које селекцију атрибута посматра као спољашњи слој процеса индукције, уграђене методе селекције представљају део основног алгоритма индукције.

Типични представници ових метода су алгоритми за индуктивно учење стабала одлучивања и продукциона правила. Као што су нпр. ID3, C4.5, CART и RIPPER. Алгоритми за учење стабала, који креирају стабло од корена према листовима и алгоритми учења правила, који обично креирају коњуктивна правила додавањем једноставних логичких израза са само једним атрибутом, приликом креирања новог чвора или једноставног израза, користе функције за оцену и избор најпогоднијег атрибута за додавање у структуру.

Поступак се прекида када стабло или скуп правила обухватају све случајеве из обучавајућег скупа. Атрибути који су употребљени сматрају се релевантним, док се остали изостављају из даљег разматрања. Поред модела секвенцијалне селекције атрибута, постоје тежински модели, где се користе тежинске оцене.

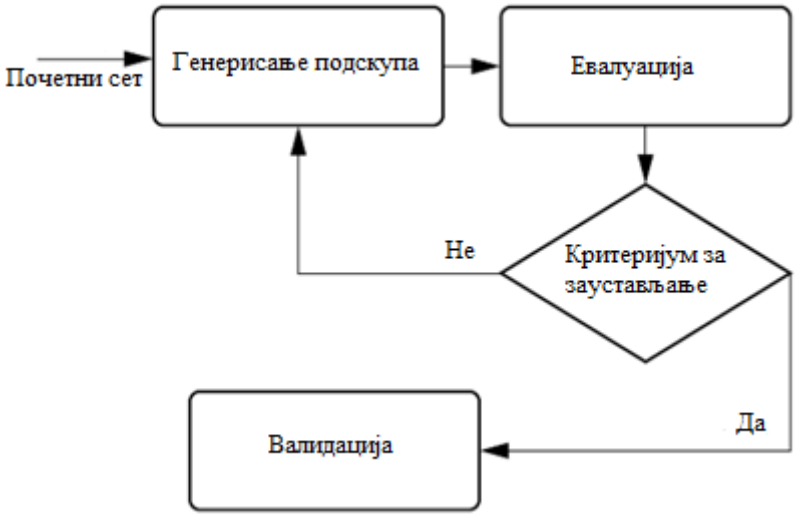
3.7. Генерална структура селекције атрибута

Општа архитектура већине алгоритама за селекцију атрибута састоји се од четири основна корака (види слику 3.5): генерисање подскупа, евалуација подскупа, критеријум за заустављање и резултат провере [Dash и Liu, 1997]. Алгоритми за селекцију атрибута генеришу подскуп, процењују подскуп, и раде то у петљи док се услов за заустављање не испуни. Коначно, пронађени подскуп се проверава уз помоћ алгоритма за класификацију на одређеним подацима.

Генерисање подскупа је процес тражења, што доводи до стварања подскупова атрибута који ће се проверавати. Укупан број кандидата за подскупове је 2^N, где је N број атрибута у изворном скупу података, што чини да је претраживање кроз простор свих могућих решења исцрпно, чак и са умереним бројем N. Не-детерминистички претрага попут еволуцијске претраге се често користи за изградњу подскупова [Yang и Honavar, 1998]. Такође, могуће је користити методе хеуристичког претраживања. Постоје две главне фамилије тих метода: додавање унапред [Koller и Sahami, 1996]

(почевши са празним подскупом, можемо додати атрибуте након локалне претраге) или елиминација унатраг (супротно).

Процена подскупа се ради јер сваки подскуп генерисан од стране поступка за генерисање треба бити оцењен коришћењем одређеног критеријума за оцењивање и да се упореди са претходним најбољим подскупом који је испоштовао овај критеријум. Ако се утврди да је бољи, онда он замењује претходни подскуп.



Слика 3.5: Генерална структура селекције атрибута, на основу [Dash и Liu, 1997]

Без одговарајућег **критеријума за заустављање**, поступак за избор атрибута се може исцрпно извршавати пре него што престане. Процес одабира атрибута може престати под једним од следећих разумних критеријума: (1) унапред дефинисати број могућих атрибута који ће бити селектовани, (2) унапред дефинисати број итерација које ће се извршавати, (3) у случају када приликом укидања или додавања атрибута не постигнемо добијање бољег подскупа, (4) оптимални подскуп према критеријуму процене је постигнут.

За одабрани најбољи подскуп атрибута треба извршити **валидацију** уз помоћ различитих тестова који се раде и на одабраном подскупу и на оригиналном скупу и упоредити резултате користећи вештачки генерисане скупове података и/или стварне скупове података.

3.8. Методе филтрирања

У наставку текста објаснићемо селекцију атрибута методом филтрирања и даћемо приказ следећих метода: Information Gain (IG), Gain Ratio (GR), Symmetrical Uncertainty (SU), Relief-F (RF), One-R (OR) и Chi-Squared (CS).

3.8.1. Селекција атрибута методом филтрирања

Методе филтрирања функционишу независно о одабраном алгоритму вештачког учења, за разлику од методе селекције претходним учењем. Код ових метода вредност атрибута се хеуристички процењује анализом општих карактеристика података из скупа за учење. Методе филтрирања користе више различитих техника одабира атрибута, јер постоји више начина хеуристичког вредновања атрибута. Ове методе се деле у две основне групе, зависно о томе вреднује ли коришћена хеуристика подскупове атрибута или појединачне атрибуте. На слици 3.6. графички је приказана селекција атрибута методама филтрирања.

Прва група метода, чија хеуристика вреднује појединачне атрибуте, одабир атрибута врши рангирањем атрибута према оцени вредности коју производи хеуристика, тако што у подскуп изабраних атрибута улазе они атрибути чија оцена вредности прелази неки унапред одабрани праг. Такође, постоји и могућност да се формира подскуп изабраних атрибута тако да се унапред одреди број (или релативни однос) атрибута које подскуп треба да садржи, па се одговарајући атрибути преузимају са врха рангиране листе.

Генерално, недостатак метода филтрирања који вреднују појединачне атрибуте је немогућност детекције редундантних атрибута, због чега корелисаност неког атрибута с другим резултира сличном оценом вредности за оба атрибута, па ће по правилу оба атрибута бити прихваћена или одбачена. Следећи недостатак ових метода је да је увршћавање атрибута у коначни подскуп препуштено спољним критеријима прага вредности или броја атрибута.

Друга група метода, чија хеуристика вреднује подскупове атрибута немају проблем увршћавања атрибута у коначни подскуп, јер резултат којег враћају није рангирана листа појединачних атрибута већ најбоље рангирани подскуп атрибута. С обзиром да се разматрају подскупови атрибута, могуће је проверавати и редундантност атрибута у подскупу. Код ове групе метода, потребно је конструисати хеуристичку

функцију вредновања на начин да пенализира постојање редундантних атрибута у посматраном подскупу, како би се редундантни атрибути елиминисали. Код ове методе, с обзиром да хеуристика вреднује подскупове атрибута, потребно је пронаћи подскуп који максимизира хеуристичку функцију вредновања. Исцрпно претраживање свих подскупова скупа атрибута је непрактично, што је случај и код метода селекције претходним учењем, због чега се користе и слични начини претраживања.

Слика 3.6: Селекција атрибута методама филтрирања [Ујевић, 2004]

Прихватљиве резултате дају позитивна селекција и негативна елиминација, а често се користи и метода најбољег првог. Методе филтрирања које вреднују подскупове атрибута су временски захтевније од филтера који вреднују појединачне атрибуте, јер постоји потреба претраживања подскупова атрибута. Међутим, ови захтеви су неупоредиво мањи у поређењу са методама претходног учења јер се у сваком кораку претраживања израчунава само вредност хеуристике вредновања, а није потребно више пута позивати алгоритам машинског учења.

Хеуристике које вреднују подскупове атрибута често имају упориште у статистичким поступцима анализе података. Тако на пример, једна од хеуристика заснива се на процени корелације међу различитим атрибутима, где се за сваки атрибут

посматраног подскупа оцењује корелација с атрибутом класе, као и међусобна корелација атрибута у подскупу. Са повећањем корелације атрибута и класе вредност хеуристичке функције расте, и опада ако се повећава међусобна корелисаност атрибута. Због тога ће неважни атрибути бити одбачени јер нису корелисани с класом, а редундантни због високе корелације са преосталим атрибутима подскупа.

Поред ослањања на статистичке поступке, хеуристика се може заснивати и на поступцима машинског учења. Неки алгоритми машинског учења имају уграђене механизме избора атрибута, па се механизми који су инхерентни једном поступку могу искористити и у другим поступцима код којих такви механизми не постоје. Такав пример је употреба стабала одлучивања за одабир атрибута, када се на потпуном скупу података за учење конструише стабло одлучивања, па се бирају само они атрибути који се заиста користе у конструисаном стаблу, а онда се у фази моделирања користи други алгоритам машинског учења. Описани поступак је оправдан ако алгоритам на редукованом скупу података за учење покаже боље класификацијске перформансе од алгоритма коришћеног за избор података.

Генерално, избор атрибута методама филтрирања траје знатно краће у поређењу са методама претходног учења, посебно кад су у питању скупови података са већим бројем атрибута [Hall, 1999], због чега су методе филтрирања често практичније решење за анализу података од других метода. Методе филтрирања се због независности о алгоритму машинског учења могу користити у комбинацији са било којом техником моделирања података, за разлику од метода претходног учења које се морају поново изводити при свакој промени циљне технике моделирања.

3.8.2. Приказ коришћених метода филтрирања у раду

У овом раду, користимо следеће методе филтрирања за рангирање атрибута које су статистички и ентропијски засноване, а показују добре перформансе у различитим доменима:

- *Information Gain* (IG),
- *Gain Ratio* (GR),
- *Symmetrical Uncertainty* (SU),
- *Relief-F* (RF),
- *One-R* (OR),

- *Chi-Squared* (CS).

Мера ентропије се обично користи у теорији информација [Abe и Kudo, 2005], која карактерише чистоћу произвољног узорка, односно меру хомогености скупа примера. Она је у основи следећих метода: IG, GR и SU. Мера ентропије се сматра мером непредвидљивости система. Ентропија *Y* може се представити као:

$$H(Y) = -\sum_{y \in Y} p(y) log_2 \left(p(y)\right) \tag{3.3}$$

где је $p(y)$ гранична функција густине вероватноће за случајну променљиву *Y*. Ако посматране вредности *Y* у тренирајућем скупу података *S* су подељене у складу са вредностима другог атрибута *X*, и ентропија од *Y* са обзиром на партицију узроковану *X*-ом је мања од ентропије *Y* пре поделе, онда постоји веза између атрибута *Y* и *X*. Ентропија од *Y* након посматрања *X* је тада:

$$H(Y/X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y/x) \log_2 \left(p(y/x)\right) \tag{3.4}$$

где је $p(y|x)$ условна вероватноћа од *Y* дата *X*-ом.

3.8.2.1. Information Gain

Статистичка вредност названа *информацисјки добитак* је добра квантитативна мера вредности атрибута за класификацију примера којом се мери како добро дати атрибут раздваја примере према њиховој класификацији. Поред ентропије као мере „нечистоће“ у скупу примера, можемо дефинисати и меру ефективности атрибута у класификацији примера. Информацијски добитак представља очекивану редукцију ентропије узроковану раздвајањем примера на основу тог атрибута.

С обзиром да је ентропија мерило нечистоће у *S* тренинг скупу, можемо дефинисати меру која одражава додатне информације о *Y* које смо добили од *X* која представља износ за који се смањује ентропија *Y* [Dash и Liu, 1999]. Ова мера је позната као IG. Дата је као:

$$IG = H(Y) - H(Y/X) = H(X) - H(X/Y) \tag{3.5}$$

IG је симетрична мера (види једначину (3.5)). Добијене информације о *Y* након посматрања *X* су једнаке информацијама добијеним о *X* након посматрања *Y*. Слабост IG критеријума је да је пристрасан у корист атрибута са више вредности чак и када нису више информативне.

3.8.2.2. Gain Ratio

Gain Ratio је не-симетрична мера која је уведена да надокнади пристраност IG [Hall и Smith, 1998]. GR је дато као:

$$GR = \frac{IG}{H(X)} \tag{3.6}$$

Као што је једначина (3.6) представља, када варијабла *Y* мора да се предвиди, можемо нормализирати IG дељењем ентропијом од *X*, и обратно. Због ове нормализације, GR вредности увек су у распону од [0, 1]. Вредност GR = 1 означава да је познавање *X* у потпуности предвиђа *Y*, и GR = 0 значи да не постоји однос између *Y* и *X*. Супротно од IG, GR фаворизује варијабле са мањим вредностима.

3.8.2.3. Symmetrical Uncertainty

Симетрична неизвесност (енг. *Symmetrical Uncertainity* – SU) представља метод селекције атрибута који из комплетног скупа атрибута елиминише ирелевантне атрибуте, на основу мере релевантности. SU се дефинише на основу ентропије атрибута *H* као:

$$SU(X,Y) = 2 \cdot \left[\frac{IG(X|Y)}{H(X)+H(Y)}\right] \tag{3.7}$$

где је $IG(X|Y) = H(X) - H(X|Y)$, $H(X|Y) = \sum_j P(y_j) \sum_i P(x_i|y_j) log_2 P(x_i|y_i)$.

Критеријум симетричне неизвесности компензује инхерентну (урођену) пристраности IG тако што дели збир ентропија од *X* и *Y* [Dash и Liu, 1999]. Може се приказати као:

$$SU = 2\frac{IG}{H(Y)+H(X)} \tag{3.8}$$

SU узима вредности, које су нормализоване у распону [0, 1], због корективног фактора два. Вредност SU = 1 значи да је познавање једног атрибута потпуно предвиђа, а SU = 0 означава да су *X* и *Y* некорелисани. Слично GR, SU је пристрасан према атрибутима са мање вредности.

3.8.2.4. Chi-Squared

Избор атрибута путем *chi square* (X^2) теста је још један, врло често коришћен метод [Liu и Setiono, 1995]. *Chi square* процена атрибута процењује вредност атрибута рачунањем вредности *chi square* с обзиром на класу. Почетна X0 хипотеза је

претпоставка да два атрибута нису повезана, и то је тестирано од стране *chi square* формуле:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad (3.9)$$

где је O_{ij} посматрана фреквенција и E_{ij} је очекивана (теоретска) фреквенција, потврђена нултом хипотезом. Што је већа вредност χ^2, то је већи доказ против хипотезе Х0.

3.8.2.5. One-R

One-R је једноставан алгоритам који је предлажио Holte [Holte, 1993]. Он гради једно правило за сваки атрибут у скупу података за учење, а затим одабира правило са најмањом грешком. Он третира све нумеричке вредности као континуиране и користи једноставан начин за дељење распона вредности у неколико дисјунктних интервала. Он обрађује недостајуће вредности као „недостаје“ и то као легитимну вредност. То је једна од најпримитивнијих шема, јер ствара једноставна правила на основу само једног атрибута. Иако представља минимални облик разврставања, он може бити користан за одређивање основних перформанси и као мерило успешности осталих алгоритама учења.

3.8.2.6. Relief-F

Relief–F за процену атрибута [Marko и Igor, 2003], процењује вредност атрибута понављајући узорковања инстанци и разматрајући вредност добијених атрибута од најближих инстанци исте или различите класе. Ова метода додељује оцену тежине за сваки атрибут на основу способности разликовања међу класама, а затим бира оне атрибуте чија тежина прелази кориснички дефинисани праг као одговарајућих атрибута.

Израчунавање тежина се заснива на вероватноћи најближих суседа из две различите класе које имају различите вредности за атрибуте и вероватноћи да два од суседа из исте класе има исту вредност атрибута. Ако је већа разлика између ове две вероватноће, атрибут је више значајан. Инхерентно, мера дефинисана за две класе проблема, може се проширити и на више класа, делећи проблем у низове од две класе проблема.

Relief–F [Marko и Igor, 2003] је метода филтрирања код које се вреднују појединачни атрибути, а оригинално је замишљена за класификацијске проблеме са само две класе. Ово је итеративна метода која у свакој итерацији коригује податак о важности појединог атрибута, где иницијално, сви атрибути имају једнаке тежинске вредности. Код ове методе у свакој итерацији поступка се на случајан начин бира један пример из скупа података за учење, а затим се у скупу података за учење проналазе његови најближи суседи из исте и супротне класе. Упоређивањем вредности сваког атрибута у изабраном примеру и пронађеним суседима ажурирају се тежинске вредности атрибута. Генерално, важни атрибути би требали имати блиске вредности за примере исте класе, а различите за примере супротних класа, због чега се различите вредности атрибута за примере супротних класа бодују позитивно, а за примере исте класе негативно. Код ове методе овај поступак се понавља унапред дефинисани број пута и на крају, тежинске вредности атрибута представљају оцену њихове вредности. Генерално, поузданост оцене вредности расте с бројем итерација, али се продужава и време извођења.

Код ове методе накнадно је описани поступак проширен на проблеме са више класа и додат је механизам третирања шума у подацима [Kononenko, 1994]. Проблеми са више класа третирају се посматрањем најближих суседа из свих преосталих класа, те се њихов утицај узима у зависности о априорној вероватноћи сваке од класа. Код ове методе умањује се утицај шума у подацима усредњавањем доприноса k најближих суседа из исте и различитих класа за сваки случајно одабрани пример.

Произвољност у избору примера је негативна страна ове методе, те ће свако покретање поступка вредновања атрибута произвести различите тежинске вредности због другачијег избора примера.

3.9. Методе претходног учења

Код метода претходног учења користе се одређени алгоритми за моделирање како би се оценили подскупови атрибута у односу на њихову класификацијску или предиктивну моћ. Код коришћења ових метода у пракси се појављују три питања:

- како претражити простор свих могућих подскупова атрибута,
- како проценити успешност алгоритма за моделирање с обзиром на претраживање скупа атрибута,

- који поступак моделирања користити као црну кутију за методе претходног учења.

Слика 3.7: Методе претходног учења и избор атрибута [Ујевић, 2004]

Код метода претходног учења вредност одређеног скупа атрибута изражава се помоћу степена исправности класификације коју постиже модел конструисан уз коришћење тих атрибута. То значи да су ове методе тесно везане за одабрани алгоритам машинског учења. За задати подскуп атрибута, исправност класификације се оцењује коришћењем техника узорковања, на пример унакрсном валидацијом (енг. *cross-validation*). Код ових метода за сваки посматрани подскуп атрибута изграђује се модел и оцењују се његове перформансе, тако да боље перформансе неког модела указују на бољи избор атрибута из којих је модел настао. Графички приказ методе претходног учења и избор атрибута дат је на слици 3.7.

Код метода претходног учења поступак избора атрибута је рачунски врло захтеван због учесталог извођења алгоритма машинског учења. Потребно је добити оцену перформанси одговарајућег модела за сваки посматрани подскуп атрибута, а методе оцене исправности модела углавном захтевају усредњавање резултата по већем броју изграђених модела. Код ових метода за сваки посматрани подскуп атрибута

изграђује се више модела, а укупан број подскупова експоненцијално расте с повећањем броја атрибута.

Исцрпно претраживање подскупова атрибута се може спровести само за мали број атрибута, будући да је тај проблем *NP*-тежак. Зато се користе разне технике претраживања, као што су: најбољи први (енг. *best-first*), гранај-па-ограничи (енг. *branch-and-bound*), симулирано каљење (енг. *simulated annealing*), генетски алгоритми и сл. [Kohavi и John, 1997]. У пракси се показује да похлепне технике претраживања дају добре резултате, што значи да се никад не проверавају већ донесене одлуке о томе да ли да се атрибут укључи (или искључи) из скупа.

Похлепне технике простор решења прелазе тако да у сваком кораку прегледају локално доступне алтернативе, па процес претраживања настављају од најбоље од њих (техника успона на врх). На слици 3.8. дат је приказ псеудо кода за похлепне технике.

1. $s :=$ почетно стање
2. понављај
   2.1. $state := s$
   2.2. пронађи све локално доступне алтернативе за $state$
   2.3. израчунај оцену $e(t)$ за сваку од пронађених алтернатива
   2.4. $s :=$ алтернатива са највећом оценом $e(s)$ док је $e(s) > e(state)$
3. врати $state$

Слика 3.8: Похлепна техника [Ујевић, 2004]

Похлепне технике се деле на избор атрибута унапред (енг. *forward selection*) и елиминација атрибута унатраг (енг. *backward elimination*) [Kohavi и John, 1997]. Могуће је и претраживање у оба смера (енг. *bidirectional search*).

Код избора атрибута унапред поступак почиње са празним скупом атрибута и у сваком кораку поступка додаје се по један нови атрибут. Елиминација атрибута унатраг је обрнути поступак који почиње са пуним скупом атрибута и у сваком кораку се одузима по један атрибут. Описани поступци су врло једноставни, али дају резултате упоредиве са сложенијим техникама похлепног претраживања као што су зракасто претраживање или метода најбољег првог.

Ако са n означимо укупан број атрибута, избор атрибута унапред и елиминација атрибута унатраг имају сложеност $O(n^2)$, и с обзиром да производе прихватљиве

резултате у разумном времену, управо ове две технике претраживања се најчешће користе у избору атрибута методама претходног учења.

Генерално, елиминација атрибута унатраг преферира веће подскупове атрибута и може резултирати нешто бољим класификацијским перформансама од одабира атрибута унапред. С обзиром да се вредност скупа атрибута мери проценом исправности класификације, онда се због само једне оптимистичне процене оба поступка могу преурањено завршити, и у том случају елиминација атрибута унатраг ће одабрати превише атрибута, а одабир атрибута унапред премало. Услед недостатка прогностичких атрибута може се ограничити способност закључивања, што ће одразити на нешто слабије класификацијске перформансе. Мањи број изабраних атрибута је пожељан у случајевима када је примарни циљ разумевање међузависности и правилности у подацима, јер су конструисани модели једноставнији и наглашавају најпредиктивније атрибуте.

Код метода претходног учења најважнији недостатак је спорост при извођењу условљена позивањем циљног алгоритма машинског учења више пута, због чега овим методама не одговарају обимни скупови података за учење са већим бројем атрибута.

Сматра се да методе претходног учења омогућују постизање нешто бољих перформанси класификације, због тесне повезаности с циљним алгоритмом машинског учења. Ово уједно може представљати и опасност јер претерано прилагођавање скупа за учење циљном алгоритму може нагласити његове недостатке.

3.10. Уграђене методе

Претходно објашњене методе селекције атрибута, методе филтрирања и претходног учења, разматрају селекцију атрибута као спољашњи слој процеса индукције. Уграђене методе врше селекцију атрибута у склопу основног алгоритма индуктивног учења, односно као део процеса генерализације.

Неки од алгоритама који врше селекцију атрибута на овакав начин су [Мишковић и Милосављевић, 2010]:

- алгоритми за индуктивно учење стабала одлучивања, на пример C4.5,
- алгоритми за индуктивно учење правила, нпр. C45Rules, RIPPER и Empiric.Rules,

- неки алгоритми учења неуронских мрежа, који могу да истовремено врше селекцију релевантних атрибута, нпр. *Optimal Brain Damage*,
- неки алгоритми учења методом потпорних (енг. *support*) вектора, нпр. *l1*-norm SVM и *lasso*.

Алгоритми за индуктивно учење стабала одлучивања и алгоритми за индуктивно учење правила минимизују функцију губитка додавањем у коначни опис само оне атрибуте чије испитивање довољно смањује грешку на обучавајућем скупу. Учење се прекида када научено правило обухвата највећи могући број случајева из обучавајућег скупа. Атрибути који су употребљени сматрају се релеватним, док остали атрибути се изостављају. На овакав начин, решење се добија брзо, и оно је разумљиво за корисника.

Алгоритми учења ансамбала у облику случајних шума (енг. *random forest*) се могу искористити и за оцену важности атрибута. Ови алгоритми користе технику зашумљавања, која се састоји у пермутовању вредности атрибута и учењу случајних стабала пре и после ове промене.

Алгоритам учења продукционих правила Empiric.Rules врши аутоматску селекцију релевантних атрибута уграђеним методом, избором неког од више информационих критеријума.

Код стандардне верзије алгоритама учења методом потпорних вектора, сви тежински коефицијенти су различити од нуле, тако да алгоритам користи равноправно све атрибуте. Ови алгоритми врше селекцију атрибута индиректно, тако што користе линеарну норму, тако да велики број тежина поприма вредност блиску нули, чиме се из модела имплицитно уклањају редундатни атрибути.

У многим случајевима обучавајући скупови су оскудни и постоје међусобне интеракције атрибута. За селектовање оптималног подскупа атрибута користе се различите естимације, које се заснивају на различитим статистичким претпоставкама, нпр. независност атрибута и довољан број обучавајућих примера, које нису увек задовољене. Због тога, уграђене методе селекције атрибута, нису увек довољне, па се у многим практичним ситуацијама користе методе претходне селекције атрибута како би се перформансе побољшале.

3.11. Екстракција атрибута

У неким апликацијама скупови података с хиљадама атрибута нису реткост, при чему некада сви атрибути могу бити значајни за неке проблеме, али за неке циљане намере само мали подскуп атрибута је обично релевантан. Проблем димензионалности се може превладати:

- тако да се одабере само подскуп релевантних атрибута, или
- стварањем нових атрибута који садрже највише информација о класи.

Прва методологија се зове селекција атрибута, а друга се зове екстракција атрибута, а то укључује линеарну (Анализа главних компонената (енг. *Principal Component Analysis* - PCA), Независну анализу компоненти (ICA) и сл.) и не-линеарну методу екстракције атрибута. У наставку текста биће речи о PCA.

Karl Pearson је 1901. године први описао могућности анализе главних компонената, али Hotelling је доста касније 1933. године разрадио практичне рачунске методе. Због комплексног рачуна, већа примена ове технике, уследила је са доступношћу рачунара [Manly, 1986].

PCA представља технику формирања нових, синтетских варијабли које су линеарне сложенице - комбинације изворних варијабли. Овом техником се редукује димензионалност, а користи се у сврху постизања прегледности и поједностављења великог броја података. Код ове технике максимални број нових варијабли који се може формирати једнак је броју изворних, а нове варијабле нису међусобно корелисане [Sharma, 1996]. Код ове технике најважнији аспект је сажимање и анализа линеарне повезаности већег броја мултиваријатно дистрибуираних, квантитативних, међусобно корелисаних варијабли у смислу њихове кондензације у мањи број компоненти, нових варијабли, међусобно некорелисаних, са минималним губитком информација.

	Варијабле				
Вектори	X_1	X_2	X_3	$\cdots$	X_p
1	x_{11}	x_{12}	x_{13}	$\cdots$	x_{1p}
2	x_{21}	x_{22}	x_{23}	$\cdots$	x_{2p}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
n	x_{n1}	x_{n2}	x_{n3}	$\cdots$	x_{np}

Слика 3.9: Улазни подаци за анализу главних компонената, на основу [Sharma, 1996]

За анализу главних компонената улазни подаци чине p варијабли (обележја, атрибути, параметри или слично) и n вектора (опажаја, индивидуа, испитаника, мерења, објеката или слично) и подаци се могу интерпретирати као n тачака у p-димензионалном векторском простору и имају облик матрице $p \; x \; n$. На слици 3.9. приказани су улазни подаци за анализу главних компонената.

Може се сматрати да су две варијабле које су високо корелисане истог или сличног садржаја, при чему се овом методом већи број таквих варијабли замењује мањим бројем варијабли. Зато се врши трансформација координатног система, где пројекције варијабли улазних података на координатне осе новог координатног система представљају нове, вештачке, варијабле – главне компоненте (енг. *principal component*) које се добијају креирањем p линеарних комбинација изворних варијабли. Циљ анализе је креирање p линеарних комбинација изворних варијабли које се називају главне компоненте [Sharma,1996]:

$$\begin{aligned}
\xi_1 &= \omega_{11}X_1 + \omega_{12}X_2 + \cdots + \omega_{1p}X_p \\
\xi_2 &= \omega_{21}X_1 + \omega_{22}X_2 + \cdots + \omega_{2p}X_p \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
\xi_p &= \omega_{p1}X_1 + \omega_{p2}X_2 + \cdots + \omega_{pp}X_p
\end{aligned} \tag{3.10}$$

где су ξ_1, ξ_2, ….ξ_p главне компоненте, а ω_{ij} су коефицијенти тј. константе које чине коефицијенте j-те варијабле за i-ту главну компоненту. Константе ω_{ij} се називају својствени или латентни вектори (енг. *eigenvectors*) и у геометријском смислу су у дводимензионалној структури уствари, синуси и косинуси углова нових оси, тј. главних компонената. Трансформисане вредности изворних варијабли (3.10) представљају збирове главних компонената (енг. *principal component scores*).

Константе ω_{ij} процењене су тако да је:

1. Укупна варијанса је сума варијанси свих изворних варијабли. Део те укупне варијансе објашњен једном главном компонентом назива се својствена вредност или латентни корен. Својствена вредност је највећа у првој главној компоненти и у свакој следећој њена је вредност све мања. Прва главна компонента, ξ_1, објашњава максимум варијансе из података, друга главна компонента, ξ_2, објашњава максимум варијансе која је остала необјашњена првом и тако даље.

2. $\omega_{i1}^2 + \omega_{i2}^2 + \cdots \; \omega_{ip}^2 = 1 \quad i = 1 \ldots p$ (3.11)

3. $\omega_{i1}\omega_{j1} + \omega_{i2}\omega_{j2} + \cdots + \; \omega_{ip}\omega_{jp} = 0$ за све $i \neq j$ (3.12)

Због неопходности фиксирања скале нових варијабли задат је услов да збир квадрата константи износи 1, из једначине (3.11), како не би било могуће повећати варијансу линеарне комбинације једноставном променом скале. Услов из једначине (3.12) осигурава међусобну некорелисаност нових варијабли, односно нове осе су међусобно ортогоналне.

С обзиром да је сума свих својствених вредности једнака укупној варијанси циљ је итерацијским поступком, издвојити већи део укупне варијансе у неколико првих главних компонената и тиме редуковати број изворних варијабли. Својствена вредност је заправо варијанса израчуната из сета збирова главне компоненте, што се може приказати следећим једначинама:

$$\begin{aligned}
\lambda x_1 &= \omega_{11}x_1 + \omega_{12}x_2 + \cdots + \omega_{1p}x_p \\
\lambda x_2 &= \omega_{21}x_1 + \omega_{22}x_2 + \cdots + \omega_{2p}x_p \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
\lambda x_p &= \omega_{p1}x_1 + \omega_{p2}x_2 + \cdots + \omega_{pp}x_p
\end{aligned}$$

или у облику матрице:

$$\lambda x = \ Wx \text{ или } (W - \lambda \mathrm{I})x = 0, \tag{3.13}$$

где је **I** јединична матрица *p x p* са вредности један у дијагонали, 0 је *p x 1* нул-вектор, а вредности скалара λ својствене су вредности матрице W. Ако се за *i*-ту својствену вредност λ_i, постави $x_1 = 1$, тада се резултирајући вектор са *x* вредности:

$x_i = \begin{bmatrix} 1 \\ x_{2i} \\ x_{3i} \\ \vdots \\ x_{ni} \end{bmatrix}$ зове *i*-ти својствени вектор матрице A.

Процес добијања својствених вектора и вредности је кључни математички проблем, а решава се помоћу растављања својствених вредности, који изражава било коју матрицу типа *n x p* (где је n ≥ p) као троструки продукт три матрице P, D и Q тако да

$$X = PDQ', \tag{3.14}$$

где је X матрица типа *n* x *p* ранга колоне *r*, P је *n* x *r* матрица, D је дијагонална матрица *r* x *r*, а Q′ је матрица *r* x *p*. Матрице P и Q су ортогоналне па је

$$P'P = \mathrm{I} \tag{3.15}$$

и

$$Q'Q = \mathrm{I}. \tag{3.16}$$

Колона p матрице Q′ садржи својствене векторе матрице X′X, а дијагонала матрице D садржи коренске вредности кореспондирајућих својствених вредности матрице X′X. Такође, својствене вредности матрица X′X и XX′ су исте. Улазна матрица може бити или матрица коваријанси или матрица корелација, зависно о проблему, типу варијабли и скали њиховог мерења. Матрица коваријанси C је симетрична, а коваријансе cov_{ii} су варијансе S_i^2:

$$C = \begin{bmatrix} cov_{11} & cov_{12} \cdots & cov_{1p} \\ \vdots & \ddots & \vdots \\ cov_{p1} & cov_{p2} \cdots & cov_{pp} \end{bmatrix} \tag{3.17}$$

Матрица корелација R (као и C) мора бити симетрична:

$$R = \begin{bmatrix} r_{11} & r_{12} \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} \cdots & r_{pp} \end{bmatrix} = \begin{bmatrix} 1 & r_{12} \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} \cdots & 1 \end{bmatrix} \tag{3.18}$$

Очекује се да ће већина нових варијабли чинити шум, и имати тако малу варијансу да се она може занемарити. Већину информација садржи првих неколико ξ варијабли - главних компоненти, чије су варијансе значајне величине. На тај начин, из великог броја изворних варијабли креирано је тек неколико главних компоненти које носе већину информација и чине главни облик.

Међутим, има ситуација када то није тако и то у случају када су изворне варијабле некорелисане, тада анализа не даје повољне резултате. Када су изворне варијабле високо позитивно или негативно корелисане могу се постићи најбољи резултати. У том случају се може очекивати да ће на пример 20-30 варијабли бити обухваћено са 2 или 3 главне компоненте.

У анализи главних компонената основни кораци су:

- Потребно је стандардизирати варијабле тако да им је просек 0, а варијанса 1 како би све биле на једнаком нивоу у анализи, јер је већина сетова података конструисана из варијабли различитих скала и јединица мерења.
- Израчунати матрице корелација између свих изворних стандардизованих варијабли.
- Потом, пронаћи својствене вредности главних компонената.
- И на крају, одбацити оне компоненте које носе пропорционално малог удела варијансе (обично првих неколико носе 80% - 90% укупне варијансе).

За интерпретацију главних компонената основу чине својствени вектори. Њихове вредности су у првој главној компоненти, најчешће, релативно равномерно распоређене по свим изворним варијаблама, док у другој главној компоненти долази до њихове веће диспропорције, што омогућава издвајање изворне варијабле (или тек неколико њих) са јачим учешћем и помаже у објашњавању и сажимању укупне варијабилности.

Добијени на овај начин, збирови главних компонената могу послужити још и:

- за даљњу интерпретацију резултата графичким представљањем, чиме се њихов релативни међусобни положај може и визуелно испитати,
- као улазне варијабле у другим мултиваријатним методама као нпр. кластер, регресијска и дискриминантна анализа. Предност коришћења збирова је у томе што нове варијабле нису међусобно корелисане чиме је решен проблем мултиколинеарности. Али, проблеме друге врсте тада може изазвати немогућност смислене интерпретације главних компонената.

ЧЕТВРТИ ДЕО

4. ЕВАЛУАЦИЈА КЛАСИФИКАЦИЈСКИХ МОДЕЛА

У четвртом делу дисертације разматрају се мере за евалуацију класификацијских модела као и методе за оцену стварне фреквенције грешака класификацијског модела, које се разликују по приступу проблему и својствима које показују. Такође, биће речи о томе да приликом тренинга постоји могућност да се модел превише прилагоди специфичностима података за тренинг и да због тога даје лошије резултате када се примени на другим подацима.

4.1. Мере за евалуацију класификацијских модела

За моделирање правилности у подацима постоји већи број метода. Такође, варирањем параметара методе, поједине методе на истом скупу примера за учење резултирају различитим моделима. С обзиром да исти проблем и исти скуп података за учење могу произвести већи број различитих модела, тиме се наглашава потреба за вредновањем квалитета модела у односу на посматрани проблем. То је разлог због чега је евалуација откривеног знања једна од битних компоненти процеса интелигентне анализе података. С обзиром да у овом раду разматрамо класификацијске проблеме, у даљем тексту ће бити речи о евалуацији класификацијских модела.

Задатак евалуације класификацијских модела је измерити у којем степену класификација сугерисана изграђеним моделом одговара стварној класификацији примера, и у зависности од начина посматрања перформанси модела, постоји више различитих мера за њихову евалуацију. У зависности од карактеристика посматраног проблема и начина његове примене, врши се избор најпогодније мере.

При евалуацији класификацијских модела основни појам је појам грешке. Уколико примена класификацијског модела на изабраном примеру доводи до резултата прогнозе класе која је различита од стварне класе примера онда је настала грешка приликом класификације. Ако је свака грешка подједнако значајна, тада је укупан број грешака на посматраном скупу примера добар индикатор рада класификацијског модела.

На овом приступу се заснива тачност као мера за евалуацију квалитета класификацијских модела. Ову меру можемо дефинисати као однос броја исправно класификованих примера према укупном броју класификованих примера.

$$\text{Тачност} = \frac{\text{број исправно класификованих примера}}{\text{укупан број примера}} \tag{4.1}$$

Основни недостаци тачности као мере за евалуацију су следећи: (1) занемарују се разлике између типова грешака; (2) зависна је о дистрибуцији класа у скупу података, а не о карактеристикама примера.

У већем броју случајева у практичном решавању проблема врло је важно разликовати одређене типове грешака. То је чест случај у медицини и нпр. откривању постојања обољења код пацијента. Ако систем треба да класификује ткива дојке на малигна и бенигна на основу мамографског снимка, онда ако систем погрешно означи оболело ткиво као здраво ткиво, грешка има већи значај, јер се неће уочити постојање болести и неће се применити одговарајућа терапија. У случају да систем препозна здраво ткиво као болесно, грешка има мањи значај, јер ће се операцијом и даљом дијагностиком утврдити да пацијент није оболео.

У случајевима када је потребно разликовати више типова грешака резултат класификације се приказује у облику дводимензионалне матрице грешака, где сваки ред матрице одговара једној класи и бележи број примера којима је то прогнозирана класа, а свака колона матрице такође је обележена по једном класом и бележи број примера којима је то стварна класа.

Ако посматрамо нпр. класификацијски проблем са 5 класа, у коме треба да класификујемо емотивна стања особа која се појављују на видео снимку у пет различитих емотивних категорија: срећан, тужан, бесан, нежан и уплашен, онда можемо матрицу грешке приказати као на слици 4.1.

		Стварна класа				
		срећан	тужан	бесан	нежан	уплашен
Прогнозирана класа	срећан	51	2	1	1	1
	тужан	3	23	1	1	0
	бесан	2	2	17	0	0
	нежан	0	1	2	9	1
	уплашен	1	0	1	1	18

Слика 4.1: Илустрација матрице грешака за класификацијски проблем препознавања емотивних стања

По дијагонали матрице налази се број тачно класификованих примера, док остали елементи матрице означавају број примера који су неисправно класификовани као нека од преосталих класа. Из матрице на слици 4.1. се види да су од укупног броја примера класе *срећан* погрешно класификована 6 примера, и то на следећи начин: три су сврстана у класу *тужан*, два у класу *бесан*, нула у класу *нежан*, и један у класу *уплашен*. Можемо закључити да се коришћењем матрице грешака омогућава квалитетнија анализа различитих типова грешака.

Иако највећи број мера за евалуацију класификацијских модела се односи на класификацијске проблеме са две класе, то не представља посебно ограничење за употребу тих мера, с обзиром да се проблеми са већим бројем класа могу приказати у облику низа проблема са две класе. При томе свака од тих мера посебно издваја једну од класа као циљну класу, при чему се скуп података дели на позитивне и негативне примере циљне класе, са тим да у негативне спадају примери свих осталих класа. То је разлог због чега у наставку текста разматрамо класификацијски проблем са две класе.

Матрице грешака за класификацијски проблем са две класе приказане су на слици 4.2. На основу слике може се закључити да су могућа четири различита резултата прогнозе. Стварно позитивни и стварно негативни исходи представљају исправну класификацију, док лажно позитивни и лажно негативни исходи представљају два могућа типа грешке. Лажно позитиван пример је негативан пример класе који је погрешно класификован као позитиван, а лажно негативан је у ствари позитиван пример класе који је погрешно класификован као негативан. У контексту нашег истраживања, улази у матрици грешака имају следеће значење [Kohavi и Provost, 1998]:

- *a* је број тачних предвиђања да је инстанца негативна,
- *b* је број нетачних предвиђања да је инстанца позитивна,
- *c* је број погрешних предвиђања да је инстанца негативна,
- *d* је број тачних предвиђања да је инстанца позитивна.

		Прогнозирано	
		Негативни	Позитивни
Стварно	Негативни	*a*	*b*
	Позитивни	*c*	*d*

Слика 4.2: Матрице грешака за класификацијски проблем са две класе

Неколико стандардних појмова су дефинисани за матрицу са две класе: тачност, одзив, лажна позитивна стопа, стварна негативна стопа, лажно негативна стопа и прецизност. Тачност је део предвиђања у укупном броју предвиђања који је тачан. Може се написати користећи следећу једначину:

$$\text{Тачност} = \frac{a+d}{a+b+c+d} \tag{4.2}$$

Одзив или стварно позитивна стопа је удео позитивних случајева који су правилно идентификовани и може се израчунати помоћу једначине:

$$\text{Одзив} = \frac{d}{c+d} \tag{4.3}$$

Лажна позитивна стопа је удео негативних случајева који су погрешно класификовани као позитивни, и израчунава се уз помоћ једначине:

$$\text{Лажна позитивна стопа} = \frac{b}{a+b} \tag{4.4}$$

Стварна негативна стопа је дефинисана као удео негативих случајева који су класификовани исправно, и израчунава се помоћу једначине:

$$\text{Стварна негативна стопа} = \frac{a}{a+b} \tag{4.5}$$

Лажна негативна стопа је удео позитивних случајева који су погрешно класификовани као негативни, и израчунавају се помоћу једначине:

$$\text{Лажна негативна стопа} = \frac{c}{c+d} \tag{4.6}$$

Коначно, прецизнизност је удео предиктивних позитивних случајева који су тачни, и израчунава се помоћу једначине:

$$\text{Прецизност} = \frac{d}{b+d} \tag{4.7}$$

Постоје случајеви када прецизност није адекватна мера. Тачност одређена помоћу једначине (4.2) не може бити адекватна мера перформансе када број негативних случајева је много већи од броја позитивних случајева [Kubat *et al.*, 1998]. Ако постоје две класе и једна је значајно мања од друге, могуће је добити високу прецизност тако што ће се све инстанце класификовати у већу групу. Претпоставимо да постоји 1000 случајева, од тога 995 негативних случајева и пет који су позитивни случајеви. Ако их систем класификује све негативно, тачност ће бити 99,5%, иако класификатор пропушта све позитивне случајеве. Или, нпр. у тестовима који установљавају да ли је пацијент оболео од неке болести, а ту болест има само 1% људи у популацији, тест који би увек пријављивао да пацијент нема болест би имао

прецизност од 99%, али је неупотребљив. У оваквим случајевима, прецизност као мера квалитета модела није одговарајућа, већ је битна мера осетљивост класификатора, односно његова могућност да примети инстанце које се траже, у овом случају болесне пацијенте.

У машинском учењу, већина класификатора претпоставља једнак значај класе у смислу броја инстанци и нивоа важности, односно све класе имају исти значај. Стандардне технике у машинском учењу нису успешне, када се предвиђају мањинске класе у неуравнотеженом скупу података или када се лажно негативни сматрају важнијим од лажно позитивних. У стварним ситуацијама, неједнаки трошкови погрешних класификација су чести, посебно у медицинској дијагностици, тако да асиметрични трошкови погрешне класификације морају бити узети у обзир као важан фактор.

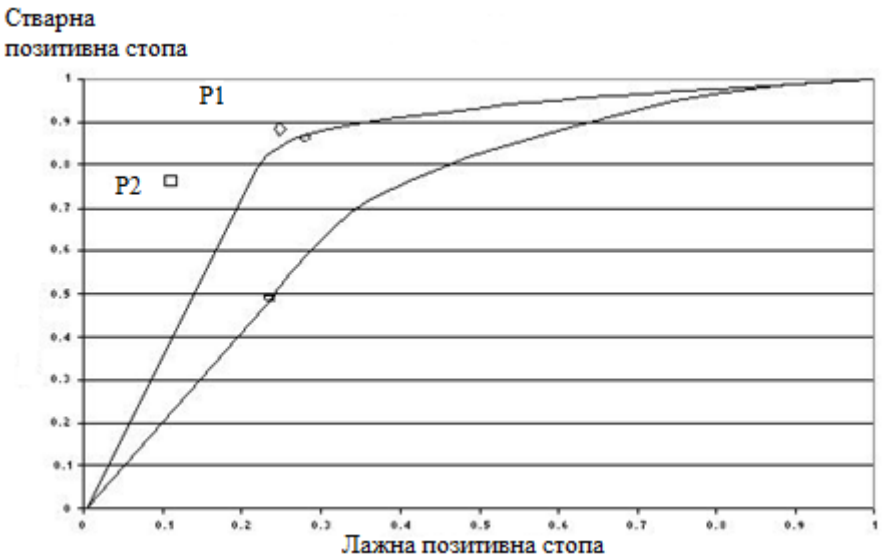
Класификатори осетљиви на трошкове (енг. *cost-sensitive*) прилагођавају моделе према трошковима погрешне класификације у фази учења, са циљевима како би се смањили трошкови погрешне класификације уместо максимизирања тачности класификације. Будући да многи практични проблеми класификације имају различите трошкове повезане са различитим врстама грешака, разни алгоритми за оцену осетљивости класификација се користе.

Комплементарност је једно од битних обележја евалуације класификацијских модела. Употребом парова мера може се приказати специфична тачност класификацијског модела са донекле супротстављених позиција. Тако се на пример варирањем параметара одабране технике моделирања може науштрб једне од мера повећати специфична тачност модела приказана другом мером. Ово је оптимизацијски проблем у којем се уз избор одговарајућих параметара на основу вредности једне мере жели максимизирати друга мера. У неким случајевима је квалитет класификацијског модела потребно изразити једним бројем, а не паром зависних мера, што се постиже употребом парова мера. Употребом парова мера вредност једне мере се фиксира и уз тај услов се посматра само друга мера. Тако на пример, може се посматрати мера прецизности уз фиксирану вредност одзива од 20% и на овај начин изведена мера назива се прецизност на 20%. Али, чешће се користи усредњавање једне од мера по више фиксираних вредности друге мере, на пример средња прецизност у три тачке (по правилу се ради о вредностима одзива од 20%, 50% и 80%).

Поред изведених мера, постоје и мере које се не заснивају на фиксирању једне компоненте пара мера, као нпр. *F-мера*, која се дефинише на следећи начин:

$$F - \text{мера} = \frac{2 \times \text{одзив} \times \text{прецизност}}{\text{одзив} + \text{прецизност}} \tag{4.8}$$

Други начин за испитивање перформанси класификатора је ROC граф. ROC графови су још један начин осим матрице грешака за испитивање перформанси класификатора [Swets, 1988]. ROC граф је дводимензионални приказ који на X оси представља лажно позитивну стопу и на Y оси представља стварну позитивну стопу. Тачка (0,1) је савршен класификатор: класификује све позитивне и све негативне случајеве исправно. То је (0,1), јер је лажно позитивна стопа 0 (нула), а стварна позитивна стопа је 1 (све). Тачка (0,0) представља класификатор који предвиђа све случајеве да буду негативни, док тачка (1,1) одговара класификатору који предвиђа да сваки случај буде позитиван. Тачка (1,0) је класификатор који је нетачан за све класификације. У многим случајевима, класификатор има параметар којим се може подешавати повећање стварне позитивне стопе по цену повећања лажно позитивне стопе или смањења лажно позитивне стопе по цени пада вредности стварно позитивне стопе. Свака поставка параметара даје пар вредности за лажно позитивну стопу и стварно позитивну стопу и низ таквих парова може се користити за представљање ROC криве. Непараметарски класификатор је представљен једном ROC тачком, којој одговара пар вредности за лажно позитивну стопу и стварно позитивну стопу.



Слика 4.3: Пример ROC графа, http://www2.cs.uregina.ca/

Слика 4.3. приказује пример једног ROC графа са две ROC криве, као и две ROC тачке означене са P1 и P2. Непараметарски алгоритми производе једну ROC тачку за одређени скуп података. Особине ROC графа су:

- ROC крива или тачка је независна од дистрибуције класе или трошкова грешака [Kohavi и Provost, 1998].
- ROC граф садржи све информације садржане и у матрици грешака [Swets, 1988].
- ROC крива пружа визуелни алат за испитивање способности класификатора за исправно препознавање позитивних случајева и број негативних случајева који су погрешно разврстани.

Простор испод једне ROC криве може се користити као мера тачности у многим апликацијама, и она се назива тачност мерења заснована на површини [Swets, 1988]. Provost и Fawcett су 1997. године [Provost и Fawcett, 1997] тврдили да коришћење тачности класификације за поређење класификатора није адекватна мера, осим ако су трошкови класификовања и расподеле класе потпуно непознати, а један класификатор мора бити изабран за сваку ситуацију. Они предлажу методу процене класификатора помоћу ROC графа и непрецизних трошкова и расподеле класе.

Други начин упоређивања ROC тачака је помоћу једначине која изједначава тачност са Еуклидском удаљености од савршеног класификатора, односно од тачке (0,1) на графу. На тај начин укључујемо тежинске факторе који нам омогућују да дефинишемо релативне неправилне трошкове класификације, ако су такви подаци доступни.

4.2. Методе за евалуацију класификацијских модела

Појам грешке, односно фреквенције грешака на неком скупу примера је у основи свих мера за евалуацију класификацијских модела. Стварна фреквенција грешака класификацијског модела је статистички дефинисана као фреквенција грешака на асимптотски великом броју примера који конвергирају стварној популацији примера.

Такође, без обзира на тип посматраних грешака, емпиријска фреквенција грешака може се дефинисати као однос броја погрешно класификованих примера наспрам укупног броја посматраних примера. На основу дефиниције проистиче да

емпиријска фреквенција грешака неког класификатора битно зависи о скупу посматраних примера. То значи да мерења на различитим скуповима примера резултирају различитим вредностима емпиријске фреквенција грешака.

Зато, када имамо неограничен броја примера, емпиријска фреквенција грешака тежи ка стварној како се број посматраних примера приближава бесконачности. Али, у реалним ситуацијама број примера је увек коначан и релативно мали. Зато је основни задатак метода за евалуацију класификацијских модела екстраполација емпиријске фреквенције грешака измерене на коначном броју примера на стварну, асимптотску вредност. За оцену стварне фреквенције грешака класификацијског модела постоји више метода, које се разликују по приступу проблему и својствима које показују. У наставку текста биће приказане неке од метода.

4.2.1. Метода евалуације на основу тестног скупа примера

На основу скупа примера за учење изграђује се класификацијски модел. Рекласификацијска фреквенција грешака је фреквенција грешака мерена на скупу примера за учење. Ако постоји неограничен број примера за учење који конвергирају стварној популацији примера, онда би рекласификацијска фреквенција грешака измерена на довољно великом скупу примера за учење била врло близу стварне фреквенције грешака. У реалним ситуацијама то никад није случај, због чега долази до претераног прилагођавања подацима за учење. Услед претераног прилагођавања подацима за учење резултирајући модел ће добро класификовати примере из скупа података за учење, али ће тачност класификације нових примера бити значајно мања. То значи да је рекласификацијска фреквенција грешака по правилу знатно мања од стварне.

Разлика стварне и рекласификацијске фреквенције грешака је добра мера степена претераног прилагођавања модела подацима за учење. Способност исправне класификације примера који нису били укључени у процес стварања модела одређује стварни квалитет класификатора. Уобичајено је да се у поступку генерисања модела не користе сви доступни примери познате класификације, већ се иницијални скуп примера дели на два дела: (1) скуп примера за учење који се користи за генерисање модела, (2) тестни скуп примера који служи за евалуацију резултирајућег модела.

Важни захтеви при оцени стварне фреквенције грешака класификацијског модела је да ова два скупа буду случајно одабрана и независна. При томе, случајан

избор подразумева да резултирајући скупови примера морају бити случајни узорци посматране популације примера. Независност забрањује постојање било какве корелације између ова два скупа, осим чињенице да потичу из исте популације примера.

Ако ови захтеви нису остварени, постоји вероватноћа да ће процена грешке бити нетачна, јер узорак није репрезентативан, односно лоше представља стварне карактеристике популације. Да би се осигурало да се евалуација модела одвија над подацима који нису коришћени при његовој изградњи, врши се подела иницијалног скупа примера на скуп за учење и тестни скуп. С обзиром да се при коришћењу модела ради са дотад невиђеним примерима, фреквенција грешака мерена на тестном скупу примера представља процену стварне грешке класификацијског модела.

Овај приступ не гарантује добру процену на свим дистрибуцијама примера, због чега је потребно размотрити и питање поузданости процене фреквенције грешака коришћењем тестног скупа. Више метода се користи за оцену стварне фреквенције грешака класификацијског модела, а оне се разликују по приступу проблему и особинама које показују.

Поузданост процене значајно зависи о броју примера у тестном скупу, при чему је процена поузданија што је тестни скуп бројнији. Кључан предуслов за поузданост процене фреквенције грешака је довољан број примера у тестном скупу, али је исто тако битно да у фази конструкције класификатора у скупу примера за учење буде довољан број примера. Са премалим бројем примера у скупу за учење дизајн класификацијског модела не може бити квалитетан, због чега је уобичајено да се из иницијалног скупа примера већи део одвоји у скуп за учење. Уобичајено се 2/3 укупног броја примера издвоји у скуп примера за учење, а 1/3 у тестни скуп примера.

Фаза конструкције и фаза евалуације класификацијског модела располагат ће са довољним бројем примера, само ако је иницијални скуп примера довољно бројан. Ако иницијални скуп примера није довољно бројан бољи резултати ће се постићи коришћењем метода евалуације које су примереније таквој ситуацији. Те друге методе се углавном заснивају на вишеструком понављању поступка процене грешке на тестном скупу, уз адекватну поделу иницијалног скупа примера.

4.2.2. Метода унакрсне валидације

У проблемима вештачке интелигенције, није редак случај да је доступан само сразмерно мали број унапред класификованих примера. Често се дешава, поготово у области медицинских истраживања да се прелиминарно испитивање спроводи на врло малом броју пацијената. Процес конструкције класификацијског модела, као и његова евалуација су тада посебно отежани, због чега је важно да се иницијални скуп класификованих примера што боље искористи и при конструкцији и при евалуацији модела. Метода евалуације коришћењем тестног скупа у овом случају може бити непрецизна, поготово ако се ослања на једну, могуће некарактеристичну партицију скупа примера за учење, односно тестирање модела.

Случајан избор је један од основних захтева при формирању скупова података за учење и тестирање. Међутим, могућност да изабрани скупови података не представљају репрезентативан узорак популације расте када се укупан број расположивих примера смањује. Због тога евалуација коришћењем тестног скупа може резултирати непрецизном проценом фреквенције грешака, и то због специфичности у скуповима података за учење односно тестирање који нису својство популације него дефект узрокован избором скупова.

Вишеструким понављањем процеса евалуације на тестном скупу користећи различите случајно изабране скупове за учење и тестирање, као и усредњавањем добијених процена фреквенције грешака, могу се избећи ове аномалије. На овом принципу се заснива метода унакрсне валидације, уз одговарајућу замену скупа података за учење и тестног скупа у свакој итерацији [Kohavi, 1995].

Код методе κ-струке унакрсне валидације најпре се иницијални скуп примера по начелу случајног избора подели у κ међусобно различитих партиција приближно исте величине. Поступак је итеративан са тим да се у једној итерацији κ–1 партиција користи као скуп за учење, а конструисани модел се тестира на преосталој партицији која представља тестни скуп. Поступак се понавља κ пута, тако да је свака од партиција по једном у улози тестног скупа примера. Оцену стварне фреквенције грешака класификацијског модела представља просечна фреквенција грешака свих κ итерација поступка.

Код унакрсне валидације, често се поступак случајног избора модификује на начин да осигура приближно једнаку заступљеност класа у свакој од партиција и овај поступак се назива стратификација. Поступком стратификације се побољшава

репрезентативност сваке од партиција. Стратификација не обезбеђује репрезентативност скупа за учење или тестирање, иако се овим начином постиже да је у свакој итерацији заступљеност класа у скупу за учење и тестирање приближно једнака заступљености у иницијалном скупу примера. Ипак, експериментални резултати показују да стратификација благо побољшава резултате евалуације, посебно на мањим скуповима примера.

Рачунарска сложеност евалуације класификацијског модела овом методом зависи о броју партиција, односно итерација унакрсне валидације, при чему свака итерација укључује засебно конструисање и тестирање модела. У пракси се најчешће користи стратификована 5-струка или 10-струка унакрсна валидација, јер се показала као довољно тачна, а није рачунарски презахтевна.

Ова метода за евалуацију класификацијских модела има предност да су сви доступни примери искоришћени за тестирање, а и конструкција модела се у свакој итерацији користи великом већином доступних примера. Метода унакрсне валидације има умерене рачунске захтеве у односу на неке друге итеративне методе евалуације, али ипак приметно веће од класичне евалуације коришћењем тестног скупа.

Ипак постоји код ове методе одређена варијабилност, иако усредњавање грешке по више тестних скупова у великој мери ублажава зависност о избору скупа примера за учење и тестирање. Узрок овоме је случајност при формирању партиција на почетку процеса евалуације, због чега треба очекивати да ће метода унакрсне валидације уз исту технику моделирања и скуп доступних података, али уз различито партиционирање тог скупа може дати нешто другачију процену фреквенције грешака. Понављањем целог поступка унакрсне валидације више пута, као и усредњавањем резултата може се постићи ублажавање овог ефекта.

4.2.3. Метода изостављања једног примера

Специјалан случај методе унакрсне валидације је метода евалуације класификацијских модела изостављањем једног примера. Означимо са n укупан број иницијално доступних примера. Ако у свакој од n итерација изоставимо један пример класификацијски модел се конструише на основу n–1 примера, а тестира на једном преосталом примеру. Коначна процена фреквенције грешака у овом случају је усредњавање грешака по свим итерацијама поступка.

Добре стране методе изостављања једног примера су: (1) максимална искористљивост иницијалног скупа примера и (2) поступак је детерминистичког карактера. Прва предност ове методе је максимална искористљивост иницијалног скупа примера што означава да се сваки појединии пример користи као тестни пример, а у свакој итерацији класификацијски модел се гради на максималном могућем скупу података за учење. Друга предност ове методе је детерминистички карактер поступка, што значи да се свака партиција састоји од само једног примера, чиме је избегнут процес случајног узорковања.

Ово за последицу има да евалуација овом методом, уз исту технику моделирања и исти скуп доступних примера, увек резултира истом проценом фреквенције грешака. Метода евалуације изостављањем једног примера постиже посебно добре резултате у процени стварне фреквенције грешака и ретко када систематски одступа од ње.

Поред ових предности, највећи недостатак методе изостављања једног примера је велика рачунарска сложеност поступка, будући да се класификацијски модел конструише и тестира *n* пута. Ова метода је добра за мање скупове примера, док за веће скупове ова метода може бити рачунарски сложена и поступак обраде података може бити захтеван.

4.2.4. Bootstrap метода

Као и метода изостављања једног примера, тако и *bootstrap* метода евалуације класификацијских модела је намењена углавном проблемима са малим бројем доступних примера [Efron и Tibshirani, 1993]. Због недостатака који постоје код методе изостављања једног примера, а то се пре свега односи на велику варијансу процене грешке на малом броју примера, која доминира у укупној непрецизности ове методе, примењује се *bootstrap* метода евалуације како би се смањио ефекат велике варијансе. При формирању скупа података за учење *bootstrap* метода се базира на узорковању са понављањем. Ова метода омогућава мултиплицирање примера из иницијалног скупа, што значи да се у скупу података за учење исти пример из иницијалног скупа примера може појавити више пута.

Ако са *n* означимо укупан број иницијално доступних примера, код *bootstrap* методе скуп примера за учење такође садржи *n* елемената, а настаје случајним избором са понављањем. Готово сигурно у скупу за учење доћи ће до понављања неких примера из иницијалног скупа, али такође постојаће примери из иницијалног скупа који уопште

нису заступљени у скупу за учење, због чега ће они формирати тестни скуп примера. Код *bootstrap* методе свака итерација укључује овакво формирање скупа примера за учење и тестирање, као и конструкцију и тестирање класификацијског модела. У поступку, број итерација није чврсто одређен, а у пракси се обично ради о неколико стотина понављања.

Код ове методе средња вредност грешке по свим итерацијама назива се *e0* проценом грешке. Код умерено великих скупова података *e0* процена даје песимистичну процену стварне грешке, а разлог за ово је да се класификацијски модел изграђује на основу релативно малог броја различитих примера, јер просечан број међусобно различитих примера у скупу за учење износи 0.632 укупног броја примера. Са становишта вероватноће можемо објаснити овако израчунат број међусобно различитих примера, јер приликом избора сваког појединог примера у скуп за учење, вероватноћа да одређени пример из иницијалног скупа неће бити одабран је $1-\frac{1}{n}$. Код ове методе у скуп за учење бира се *n* примера, а вероватноћа да одређени пример неће бити изабран у *n* покушаја дат је следећим изразом:

$$\left(1-\frac{1}{n}\right)^n \approx e^{-1} \approx 0.368 \qquad (4.9)$$

Апроксимација дата претходним изразом (4.9) вреди за довољно велико *n*. То значи да ће тестни скуп садржавати 0.368 укупног броја примера за довољно велике иницијалне скупове примера, док остатак од 0.632 укупног броја примера даје број различитих примера у скупу за учење.

Табела 4.1. Упоредне карактеристке метода за оцену грешке класификацијског модела [Ујевић, 2004]

	процена на основу тестног скупа	*к* унакрсна валидација	метода изостављања једног примера	*bootstrap* метода
број примера у скупу за учење	*j* (најчешће 2/3 *n*)	*(к-1)n/к*	*n-1*	*n, j* различитих ($j \approx 0.632\, n$)
број примера у тестном скупу	*n-j* (најчешће 1/3 *n*)	*n/к*	*1*	*n-j* ($\approx 0.368\, n$)
број итерација	1	*к*	*n*	неколико стотина

Bootstrap метода је рачунски прилично захтевна због већег броја итерација па се углавном примењује на проблеме са мањим бројем доступних примера. И поред тога што *bootstrap* метода нема проблем велике варијансе, ова метода није увек супериорна

методи изостављања једног примера на мањим скуповима примера. Мала вредност *e0* процене грешке је бољи индикатор доброг модела од процене методом изостављања једног примера. Табела 4.1. приказује упоредне карактеристике споменутих метода за оцену грешке класификацијског модела.

4.3. Претерано прилагођавање модела подацима за тренинг

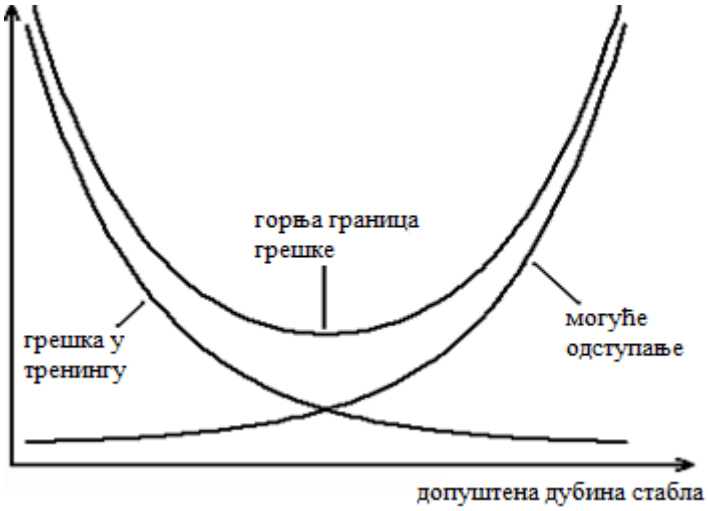
Приликом тренинга постоји могућност да се модел превише прилагоди специфичностима података за тренинг и да због тога даје лоше резултате када се примени на другим подацима. Тако, на пример подаци за тренинг могу имати одређене карактеристике као плод случајности, шума или представљати пристрасан узорак целог скупа података. Дешава се у пракси да је податке тешко сакупити, због чега се обично мора радити са подацима који су расположиви без обзира на њихове мањкавости.

И поред тога што је приликом тренирања модела потребно да се постигне висок ниво прецизности, потребно је такође и пазити да не дође до претераног прилагођавања подацима. До претераног прилагођавања подацима долази услед богатства простора хипотеза, односно скупа допустивих модела. Ако је скуп богатији онда је лакше наћи модел који добро одговара подацима. Тако нпр., уколико се при учењу допуштају само стабла дубине 1, која тестирају само један атрибут сваке инстанце, јасно је да таква стабла не могу лако постићи високу прецизност класификације.

Поред ових стабала и стабла која су врло дубока и прецизно описују сваку и најнебитнију специфичност података за тренинг, у пракси се показују непоузданим, пошто шири скупови података не морају увек имати све специфичности скупа података за тренинг. Адекватна хипотеза би требало да апстрахује, односно занемари, такве специфичности. Међутим, са повећањем дозвољене дубине стабла, повећава се моћ учења, односно вероватноћа да ће у скупу допустивих модела бити нађен онај који добро описује податке, због чега се смањује грешка класификације.

Ако стабло дубине, на пример 1 има високу прецизност, то значи да је у подацима нађена јака и врло једноставна законитост. Међутим, ако је стабло високе прецизности врло дубоко, то значи да је уочена законитост у подацима врло комплексне природе и стога може бити специфична само за податке у тренинг скупу,

због чега очекивано одступање грешке на ширем скупу података од грешке на тренинг скупу може бити велико.



Слика 4.4: Грешка класификације у зависности од богатства скупа допустивих модела [Јаничић и Николић, 2010]

На слици 4.4. приказане су три криве. Прва крива која представља понашање грешке класификације на тренинг скупу у зависности од дозвољене дубине стабла је опадајућа. Друга крива која представља понашање одступања грешке на ширем скупу података од грешке на тренинг подацима у зависности од дозвољене дубине стабла је растућа. И последња трећа крива представља горњу границу грешке класификације у зависности од дозвољене дубине стабла и она је збир претходне две. Можемо закључити да и премале и превелике вредности за дозвољену дубину стабла воде лошим резултатима, прве услед нефлексибилности дозвољених модела, а друге због претераног прилагођавања модела тренинг подацима.

Код нпр. стабала одлучивања, проблем претераног прилагођавања тренинг подацима, могуће је решити коришћењем два приступа: (1) заустављањем раста стабла у току његове изградње и (2) накнадним одсецањем. Користи се најчешће друга могућност, при чему се одсецање врши тако што се итеративно понавља у чворовима у којима се највише повећава прецизност класификације на скупу за тестирање све док даље одсецање не почне да смањује прецизност класификације. Да не би дошло до претераног прилагођавања тренинг подацима код стабала одлучивања, одсецање стабла у одређеном чвору представља замену целог подстабла чији је то корен тим

чвором, с тим што му се додељује ознака класе у коју се подаци у том подстаблу најчешће класификују.

Исто резоновање се може спровести и за друге методе машинског учења. Алгоритам за учење добро генерализује из приказаних примера када модел који најбоље апроксимира циљну функцију на расположивим инстанцама, такође најбоље апроксимира циљну функцију на свим могућим инстанцама. Успех добре генерализације лежи у адекватном управљању богатством простора хипотеза. Тако нпр., неки алгоритми учења попут методе потпорних вектора су конструисани тако да приликом избора модела аутоматски решавају и овај проблем.

ПЕТИ ДЕО

5. ПРОБЛЕМ КЛАСИФИКАЦИЈЕ

У петом делу дисертације, разматра се проблем класификације, који представља разврставања непознате инстанце у једну од унапред понуђених категорија. У овом делу биће речи о класификационим алгоритмима, који су касније коришћени у експерименталним истраживањима за доказ постављених хипотеза. То су следећи алгоритми надзираног учења за изградњу модела: IBk, *Naïve Bayes*, SVM, J48 стабло одлучивања и RBF мрежа.

5.1. Појам класификације

Класификација је један од најчешћих задатка машинског учења, и представља проблем разврставања непознате инстанце у једну од унапред понуђених категорија — класа. У нашој природи да ствари око себе, како бих их боље схватили или организовали, класификујемо и категоризујемо. Тако нпр. класификација се користи у: дијагностификовању болести, прогнози болести код пацијента, одабиру најбоље терапије за пацијента од неколико могућих, класификацији кредитиних захтева клијената, процени да ли ће и који корисници купити одређени производ, избору циљне групе клијената за маркетиншке кампање, анализи слике, анализи гласа за биометријска потребе или за потребе анализе здравственог стања особе, препознавању емотивног стања особа на основу слике и гласа, дијагностификовању здравственог стања биљака или животиња и слично. Примена класификације је велика и у решавању проблема у другим областима. Важно запажање код класификације је да је циљна функција у овом проблему дискретна. У општем случају, ознакама класа се не могу смислено доделити нумеричке вредности нити уређење. То значи да је атрибут класе, чију је вредност потребно одредити, категорички атрибут.

На примеру откривања да ли ће бити повратка болести рака дојке код жена објаснићемо проблем класификације. Предвиђање може да се ради на основу следећих података: година пацијента, наступања менопаузе, величине тумора, величине чворова, степена малигнитета, која дојка је захваћена тумором, положаја тумора, да ли је

вршено зрачење или не код пацијента и слично. На основу прикупљених података о већем броју пацијента, при чему скуп података садржи и податке када нема повратка болести рака дојке и када има повратка болести рака дојке, врши се класификација. Свака инстанца у скупу података се односи на стање једног пацијента које је описано са одговарајућим бројем атрибута.

Класификација неког објекта се заснива на проналажењу сличности са унапред одређеним објектима који су припадници различитих класа, при чему се сличност два објекта одређује анализом њихових карактеристика. При класификацији се сваки објекат сврстава у неку од класа са одређеном тачношћу. Задатак је да се на основу карактеристика објеката чија класификација је унапред позната, направи модел на основу кога ће се вршити класификација нових објеката. У проблему класификација, број класа је унапред познат и ограничен.

Процес класификације се састоји из две фазе, при чему се у првој фази гради модел на основу карактеристика објеката чија класификација је позната. За изградњу модела се користе подаци који се најчешће налазе у табелама. Свака инстанца узима само једну вредност атрибута класе, а атрибут класе може да има коначан број дискретних вредности које нису уређене.

Класификациони алгоритам учи на основу познатих класификација тј. на основу инстанци објеката чија класификације је позната. При томе, на основу вредности њихових атрибута и атрибута класе, гради се скуп правила на основу којих ће се касније вршити класификација. Методе класификације су најчешће засноване на стаблима одлучивања, Бајесовим класификаторима, неуронским мрежама, итд.

Након учења, модел се тестира тј. процењује се његова тачност, при чему под тачношћу подразумевамо проценат инстанци које су тачно класификоване. Вредност атрибута класе сваке тестне инстанце пореди се са вредношћу атрибута класе која је одређена на основу модела. Важно је напоменути да се за тестирање модела користе инстанце које нису коришћене у фази учења.

Постоји више начина за издвајање тестних инстанци, али се најчешће издвајају случајним избором, пре фазе учења, од инстанци чија је класификација позната. При томе, ако је тачност модела задовољавајућа онда се даље користи у класификацији објеката чија вредност атрибута класе није позната.

Постоје следећи критеријуми којима се пореде и оцењују методе класификације и то су:

- Тачност, што представља способност класификатора да тачно класификује инстанцу непознате вредности атрибута класе.
- Брзина, што представља број операција које се изврше при конструкцији и примени класификатора.
- Робустност, што представља прецизност класификатора када се примени на подацима са шумом или подацима којима недостају вредности неких атрибута.
- Скалабилност, што представља ефикасност методе ако се примењује на велике количине података.
- Интерпретабилност, што представља јасан приказ и разумевање резултата.

Методе рангирања рангирају сваки атрибут у скупу података. Резултати се потврђују коришћењем различитих алгоритама за класификацију. Широк распон алгоритама за класификацију је на располагању, сваки са својим предностима и слабостима. Не постоји такав алгоритам учења који најбоље ради са свим проблемима надзираног учења. Машинско учење укључује велики број алгоритама као што су:

- вештачке неуронске мреже,
- генетски алгоритми,
- пробабилистички модели,
- индукцијска правила,
- стабла одлучивања,
- статистичке или методе распознавање узорака,
- *к*-најближи суседи,
- *Naïve Bayes* класификатори и
- дискриминаторна анализа.

У овом раду коришћени су следећи алгоритми надзираног учења за изградњу модела, а то је IBk, *Naïve Bayes*, SVM, J48 стабло одлучивања и RBF мрежа. Предност IBk је да су они у могућности да уче брзо са врло малим скупом података. Предност *Naïve Bayes* класификатора је да захтева малу количину тренинг података за процену параметара потребних за класификовање. Предност SVM над другим методама је пружање бољих предвиђања невиђених тест података, пружање јединствених оптималних решења за проблем у тренирању и постојање мање параметара за оптимизацију у поређењу са другим методама. J48 стабло одлучивања има разне предности: једноставан за разумевање и интерпретацију, захтева малу припрему

података, робустан је, добро ради и са великим бројем података у кратком времену. RBF мреже нуде низ предности, укључујући и захтевање мање формалних статистичких тренинга, способност да се имплицитно детектују сложени нелинеарни односи између зависних и независних варијабли, способност детектовања свих могућих интеракција између предикторских варијабли и доступност више алгоритама за тренинг. Ово поглавље даје кратак преглед ових алгоритама.

5.2. Методе класификације засноване на инстанцама

У наставку текста биће речи о методама класификације које су засноване на истанцама. Биће дат приказ модела, разматраће се стабилност класификације помоћу алгоритма *к* најближих суседа (енг. *n-nearest neighbours*), предности и недостаци овог алгоритма, као и приказ псеудо кода.

5.2.1. Приказ модела класификације заснован на инстанцама

Шема *к* најближих суседа користила се још у педесетим годинама двадесетог века [Fix и Hodges, 1951], а као поступак класификације појављује се десетак година касније [Johns, 1961], а најинтензивније се користила на подручју распознавања узорака.

Класификација заснована на инстанцама спада у најједноставније технике интелигентне анализе података, јер не врши експлицитну генерализацију циљног појма на основу својстава која су изводива из скупа за учење, већ се своди на меморисање скупа за учење, односно појединачних инстанци које садржи. Основни облик алгоритма не укључује процесирање инстанци из скупа за учење у фази конструкције модела, већ само њихово меморисање.

Класификација нових инстанци се обавља према принципу најближег суседа, где се нова инстанца упоређује с меморисаним инстанцама из скупа за учење коришћењем дефинисане метрике. Метрика дефинише растојање инстанци на основу вредности њихових атрибута, а одговара интуитивном схватању сличности инстанци, тако да ако су инстанце сличније, растојање је мање. Нова инстанца се класификује на основу претраживања скупа за учење са циљем проналажења инстанце која му је у

смислу растојања најближа. Нова инстанца која се класификује добија класу те инстанце.

Значи, елемент технике класификације засноване на инстанцама који утиче на облик модела је метрика, при чему је у употреби више различитих метрика, а најчешће се користи *Еуклидска*. Ако са $x = (x_1, x_2, \dots, x_n)$ означимо вектор вредности атрибута произвољне инстанце, Еуклидска метрика је тада дефинисана изразом на следећи начин:

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{5.1}$$

Еуклидско растојање инстанци x и y се може претставити на следећи начин:

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5.2}$$

На сличан начин могу се дефинисати и друге метрике варирањем потенције координата вектора у дефиницији Еуклидске метрике. На пример, изостављање квадрирања уз употребу апсолутне вредности даје тзв. правоугаону метрику. Генерално, велике разлике у вредностима појединих координата додатно се наглашавају вишим потенцијама, науштрб координата код којих је разлика у вредности мала.

У напред датим изразима (5.1) и (5.2) се имплицитно претпоставља да су сви атрибути нумеричког типа, тј. да су вредности координата бројеви, а како би се дефиниција Еуклидског растојања могла применити на номиналне атрибуте, потребно је дефинисати операцију разлике над номиналним вредностима.

Ако су са a_i, $a_j \in Dom(A_i)$ означене две произвољне вредности номиналног атрибута A_i, онда је разлика вредности a_i и a_j дефинисана *0–1 функцијом разликовања* (5.3), на следећи начин:

$$a_i - a_j = \begin{cases} 0, \text{за } a_i = a_j \\ 1, \text{ иначе} \end{cases} \tag{5.3}$$

Због различитих скала мерења за различите нумеричке атрибуте постоји проблем везан за коришћење различитих метрика. Тако на пример, атрибут чији се распон вредности креће унутар десетих делова мерне јединице имат ће занемарив утицај на коначни резултат у односу на атрибут са распоном вредности од неколико десетина мерних јединица. Зато је код метода класификације засноване на инстанцама уобичајен поступак нормализације свих нумеричких атрибута на интервал [0,1], коришћењем функције (5.4):

$$f(x) = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (5.4)$$

где x_{min} и x_{max} означавају најмању, односно највећу вредност посматраног атрибута. Код метода класификације заснованим на инстанцама неодређене вредности атрибута се третирају слично номиналним атрибутима, тј. проширује се дефиниција операције разлике, тако да се разлика дефинише на начин да је неодређена вредност максимално удаљена од било које посматране вредности атрибута.

Стандардна дефиниција Еуклидског растојања из израза (5.2) се може користити и за инстанце с неодређеним вредностима атрибута, ако се користи овако дефинисана функција разлике. Слика 5.1. приказује псеудо код за основни класификатор *к* најближих суседа.

ПОЧЕТАК

Улаз: $D = \{(x_1, c_1), \dots, (x_N, c_N)\}$

$x = (x_1, \dots, x_n)$ нова инстанца за класификацију

За сваку инстанцу која је класификована (x_i, c_i) израчунај $d(x_i, x)$

Наредба $d(x_i, x)$ од најниже до највише, $(i = 1, \dots., N)$

Изабери *к* најближе инстанце за x:D_x^K

Додели за *x* најфреквентнију класу у D_x^K

КРАЈ

Слика 5.1: Псеудо код за основни класификатор *к* најближих суседа (http://biocomp.cnb.csic.es/~coss/Docencia/ADAM/Sample/Sample_Classification.pdf)

5.2.2. Претраживање простора решења

Код класификације засноване на инстанцама не спроводи се експлицитна генерализација својстава циљног појма, тј. не претражује се простор решења у потрази за што бољим моделом. Код класификације се појављује само један имплицитни класификацијски модел који је у потпуности одређен скупом за учење и функцијом растојања. Основни алгоритам се може надограђивати тако да се у одређеном опсегу модификује скуп инстанци за учење или функција растојања.

Низом операција над моделом се спроводи модификација основног класификацијског модела, па се може се говорити о поступку претраживања простора решења. Једна од варијанти класификације засноване на инстанцама настоји

редуковати број инстанци у скупу за учење, првенствено ради смањења опсега претраживања при класификацији нових инстанци, јер скуп инстанци за учење по правилу садржи велики број редундантних инстанци. Код проблема класификације најважније су инстанце које се налазе у близини граница међу класама, а инстанце из унутрашњости омеђеног подручја класа могу се изоставити без последица на тачност класификације.

Да би се добио генерализован модел, поступак формирања подскупа инстанци је итеративан, а састоји се од увршавања или елиминисања инстанци према унапред дефинисаном критеријуму. Потребно је у таквом моделу настојати да се задрже репрезентативне инстанце, које посредством функције растојања добро генерализирају подручје у којем се налазе, а из скупа избацити инстанце који битно не придоносе обликовању подручја одговарајуће класе. Формирање репрезентативног скупа инстанци, претраживању приступа по начелу одоздо на горе, тј. од појединачних инстанци ка репрезентативним инстанцама са проширеном сфером утицаја. Постоји више различитих критеријума прихватања односно елиминисања инстанци, али се углавном ради о неповратним стратегијама претраживања похлепног карактера. Према најједноставнијем критеријуму просуђује се инстанца према резултату класификације коришћењем скупа до тада издвојених репрезентативних инстанци. Ако се ради о нетачној класификацији инстанце, онда се она придодаје у скуп репрезентативних инстанци, јер је евидентно да мења границе класа. Ако се ради о тачној класификацији инстанце, онда се она проглашава сувишном, под претпоставком да је њена информативност већ садржана у скупу помоћу којег је класификована.

Напред дати критеријум за избор инстанци има више недостатака:

- у почетној фази процеса претраживања постоји незанемарива вероватноћа одбацивања инстанци које се могу показати важним за тачност класификације резултирајућег модела;
- поред овога, изабрани подскуп репрезентативних инстанци не зависи само о полазном скупу, већ и о редоследу евалуације инстанци;
- и можда најбитнији недостатак се односи на лоше понашање у условима шума у подацима, јер с обзиром да се у скуп увршавају и нетачно класификоване инстанце, овај критеријум има тенденцију акумулирања инстанци са шумом у резултирајућем скупу инстанци, што доводи до смањења његове репрезентативности.

Због свега напред реченог, у пракси се чешће употребљавају други, нешто сложенији критеријуми за избор репрезентативних инстанци.

5.2.3. Удаљеност инстанци

Осим избором инстанци за памћење, на облик класификацијског модела се може утицати и модификацијом функције растојања. Једнак утицај свих атрибута у инстанци на коначан резултат је једно од својстава Еуклидског растојања, али су у пракси ретки проблеми код којих сви атрибути имају једнаку класификацијску вредност, чиме се ствара могућност за побољшање технике класификације засноване на инстанцама. Модификација функције растојања на начин да валоризује класификацијски потенцијал различитих атрибута је једно од могућих решења. Начин да се ово постигне је стандардно проширење Еуклидског растојања које подразумева увођење тежинских вредности атрибута. Ако са w_i означимо тежинску вредност придружену атрибуту A_i, онда модификовано Еуклидско растојање инстанци x и y можемо представити на следећи начин:

$$d_w(x,y) = \sqrt{\sum_{i=1}^{n} w_i^2 (x_i - y_i)^2} \tag{5.5}$$

Већи утицај на прорачун растојања инстанци пружа му већа тежинска вредност придружена атрибуту. Варирање тежинских вредности атрибута је један од начина корекције класификацијског модела у техници класификације засноване на инстанцама. И поред тога што постоје независни поступци оцене класификацијске вредности атрибута, у контексту ове технике моделирања тежине атрибута се најчешће одређују интерно, приликом формирања релевантног подскупа инстанци за учење.

Свим атрибутима је иницијално придружена тежинска вредност 1, која се итеративно модификује при разматрању сваке од инстанци из скупа за учење. У подскупу релевантних инстанци се проналази инстанца y најближа посматраној инстанци x, као и при класификацији инстанци. Ако инстанце x и y припадају истој класи, смањује се тежинска вредност атрибута чије се вредности у инстанцама x и y највише разликују, јер се разлика у вредностима тих атрибута приписује слабијој корелацији са класом, као и што се у случају да инстанце x и y припадају различитим класама, тежинска вредност атрибута са највећом разликом вредности повећава. Повећање, односно смањење тежинске вредности пропорционално је разлици вредности атрибута у инстанцама x и y.

Генерално, постоје и радикално другачији приступи дефинисању функције растојања, при чему је један од њих приступ вероватноће, код којег се дефинишу операције трансформације инстанци за учење. Посматрањем низа операција помоћу којих се једна од инстанци може трансформисати у другу, као и израчунавање вероватноће да се таква трансформација догоди уз случајан избор операција и њихов редослед, може се утврдити растојање две инстанце [Cleary и Trigg, 1995]. Посматрањем свих низова операција које доводе до тражене трансформације инстанци, заједно са вероватноћом сваке од њих се побољшава робусност.

Ако се овако дефинише растојање, предност коју добијамо је могућност униформног третирања нумеричких и номиналних атрибута, дефинисањем одговарајућих операција трансформације за сваки од њих. Значајна предност у неким применама, је и да вероватноћа интерпретација растојања осим категоричке класификације може као резултат понудити и дистрибуцију вероватноће припадања свакој од класа.

5.2.4. Шум у подацима за учење

Базични облик технике класификације заснован на инстанцама је прилично подложан проблему шума у подацима за учење, а разлог томе је да се класификација нове инстанце ослања на само једну (најближу) инстанцу из скупа за учење.

Битно смањење утицаја шума може се спровести проширењем поступка класификације према принципу *к* најближих суседа, где се уместо издвајања само једне најближе инстанце из скупа за учење, издваја *к* најближих инстанци, за неки унапред одређени мали број *к*. У класификацији нове инстанце учествује *к* пронађених инстанци, по принципу већинског гласања, при чему се инстанци придружује најфреквентнија класа унутар издвојених *к* инстанци. Специјални случај овог уопштења за *к*=1 представља основни алгоритам класификације.

Генерално, вредност константе *к* зависи о количини шума у подацима за учење, при чему ако је више шума, повољније је изабрати веће вредности константе *к*. На овај начин се постиже побољшање тачности класификације у условима шума, због чега је варијанта *к* најближих суседа готово у потпуности истиснула основни облик алгоритма. За посматрани скуп инстанци, може се доказати да за $|C|\to\infty$ и $к\to\infty$ на начин да $k/|S| \to 0$, вероватноћа погрешне класификације тежи теоретском минимуму.

Постоји и други приступ третирању шума у подацима који се састоји од детекције инстанци са шумом и њиховог издвајања из скупа за учење, при чему је праћење класификацијских перформанси сваке инстанце из скупа за учење уобичајен начин детекције непожељних инстанци. Код овог приступа се унапред одреде два прага тачности класификације, у сврху одбацивања и прихватања инстанци у скуп инстанци за памћење. Ако тачност класификације истанце падне испод прага одбацивања онда се оне елиминишу из скупа за учење, а ако тачност класификације пређе праг прихватања те истанце се користе за класификацију. При томе се инстанце чија се тачност класификације налази између два прага не користе при класификацији, али се њихове класификацијске перформансе прате и коригирају сваки пут када су издвојене као најближе инстанце приликом класификације других инстанци.

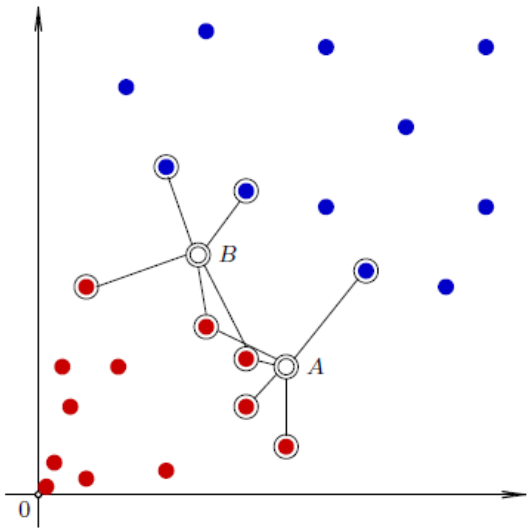
За одређивање прагова одбацивања и прихватања инстанци користи се претпоставка *Bernoulli*-јевог процеса, при чему тачна класификација одговара успешном исходу експеримента. Ови прагови се одређују упоређењем очекиване фреквенције успеха поједине инстанце и очекиване фреквенције успеха слепе класификације за одговарајућу класу, тј. оне класификације која увек предвиђа споменуту класу. За критеријум прихватања потребно је да доња граница интервала поузданости посматране инстанце буде виша од горње границе интервала поузданости слепе класификације, а за критеријум одбацивања потребно је да за посматрану истанцу горња граница интервала поузданости оцене фреквенције успеха буде нижа од доње границе интервала поузданости слепе класификације. Није неопходно користити исти ниво поузданости за одређивање прага одбацивања и прихватања инстанци, тако нпр., варијанта алгоритма класификације заснована на инстанцама позната под називом *IB3* (енг. *Instance-Based learner ver. 3*), користи критеријум одбацивања који је нешто блажи од прага прихватања, јер се не губи пуно одбацивањем инстанци умерено слабих класификацијских перформанси – велики су изгледи да ће у каснијој фази процеса бити надомештени инстанцама са бољим резултатима класификације.

5.2.5. Стабилност класификације помоћу алгоритма *к* најближих суседа

Слика 5.2. приказује непознате инстанце A и B. Методом *к* најближих суседа уз коришћење Еуклидског растојања, инстанца A, бива класификована у црвену класу за све вредности *к* од 1 до 5. Класификација инстанце A је постојана зато што се она налази близу црвених инстанци, а удаљено од плавих инстанци. За разлику од

инстанце A, класа инстанце B може да варира у зависности од броја *к*, и то тако да за *к* = 1 инстанца B се класификује у црвену класу, за *к* = 2 не може се одлучити, за *к* = 3 инстанца B се класификује у плаву класу, за *к* = 4 поново није могуће одлучити, а за *к* = 5, она се поново класификује у црвену класу [Јаничић и Николић, 2010]. Класификација инстанце B није постојана јер се она налази близу инстанци из обе класе. Из напред наведеног, можемо закључити да је метода *к* најближих суседа постојана у унутрашњости области коју заузимају инстанце једне класе, али је непостојана на ободу те области.

Непостојаност класификације осим што може да се демонстрира мењањем параметра *к*, она се такође може анализирати и за фиксирану вредност параметра *к*, и то тако да је за мање вредности параметра *к* непостојаност при варирању вредности атрибута инстанце већа него за веће вредности параметра *к*. Емпиријски се одређује вредност параметра *к*, евалуацијом успешности класификације за различите вредности параметра *к*, тако што се бира вредност *к* за коју је класификација била најуспешнија. Локалност је важно својство метода заснованих на инстанцама, јер се непозната инстанца класификује искључиво или углавном на основу познатих инстанци које се налазе у њеној близини. Због овог својства, методе класификације помоћу алгоритма *к* најближих суседа доприносе флексибилности модела које граде.



Слика 5.2: Стабилност класификације помоћу алгоритма *к* најближих суседа [Јаничић и Николић, 2010]

5.2.6. Предности и недостаци класификације засноване на инстанцама

Методе класификације засноване на инстанцама не граде експлицитан модел података у виду неке функције као што то ради већина метода машинског учења. Зато се класификација не врши на основу већ формулисаног модела, него на основу скупа инстанци за тренинг, тако што инстанце предвиђене за тренирање се чувају и бивају употребљене тек кад је потребно класификовати непознату инстанцу. На овај начин се већина израчунавања премешта из фазе учења у фазу примене.

Метода *к* најближих суседа се заснива на једноставном принципу да непознату инстанцу треба класификовати у класу чије су инстанце најсличније непознатој. Концепт сличности се најједноставније формализује преко функција растојања, при чему што је растојање између два објекта веће, то је сличност између њих мања и обрнуто. Могуће је бирати различите функције растојања, при чему је претпоставка да изабрана функција растојања релеванта за посматрани домен и да стварно осликава различитост између два објекта. Пошто се изабере функција растојања, метода *к* најближих суседа се састоји у налажењу *к* инстанци из тренинг скупа које су најближе непознатој инстанци и њеном класификовању у класу чији се елементи најчешће јављају међу пронађених *к* најближих суседа.

Основни облик класификације заснован на инстанцама има више недостатака:

- Поступак класификације нових инстанци може бити спор у случају великих скупова за учење, будући да класификација сваке инстанце захтева претраживање целог скупа за учење.
- Без коришћења више најближих инстанци показује приличну осетљивост на шум у подацима за учење.
- Такође, није прилагођен проблемима код којих атрибути имају различит класификацијски потенцијал, а класификацијске перформансе посебно нарушавају ирелевантни атрибути.

Овај тип класификације постаје поново популаран почетком 1990-их кроз радове D. Aha [Aha, 1992], у којима се надоградњама основног поступка умањују споменути недостаци, тако што се уводе тежинске вредности атрибута и поступак филтрирања инстанци са шумом, чиме се значајно побољшавају класификацијске способности ове технике, подижући их на ниво упоредив са осталим популарним техникама. Поред овога, одбацивање непотребних инстанци значајно редукује број

инстанци који се памте, чиме се значајно смањују потребни ресурси и време класификације.

Можемо закључити да је најважнија предност технике класификације засноване на инстанцама у односу на стабла одлучивања и класификацијска правила могућност изражавања произвољних по деловима линеарних граница међу класама, а основни недостатак је чињеница да класификацијски модел није изражен експлицитно, у облику који би био дескриптиван у терминима домена класификацијског проблема.

5.3. Методе *Bayes*-ове класификације засноване на вероватноћи

Пробабилистички приступ индукцији знања приказан у овом раду додељује вероватноћу класификације инстанци у поједине класе. *Bayes*-ова теорема је основ оваквог пробабилистичког приступа.

5.3.1. Основе *Bayes*-ове теореме

Bayes-ова теорема омогућује избор највероватније хипотезе из скупа хипотеза *H* на основу скупа за учење *D*, а уз утицај предодређених вероватноћи сваке од понуђених хипотеза у скупу *H*. На слици 5.3. приказан је Thomas Bayes, који је живео и радио у осамнаестом веку.

Слика 5.3: Thomas Bayes (1701 –1761)

Најпре, потребно је дефинисати вероватноће:

- $P(h)$ – почетна вероватноћа хипотезе h, која нам омогућава да прикажемо почетно знање о вероватноћама различитих хипотеза. Ако то знање не поседујемо можемо свим хипотезама придодати једнаку почетну вероватноћу.
- $P(D)$ – почетна вероватноћа појављивања инстанце D, која означава вероватноћу појављивања D без обзира на то која је хипотеза исправна.
- $P(D|h)$ – услова вероватноћа појављивања D уз услов исправности хипотезе.
- $P(h|D)$ – условна вероватноћа исправности хипотезе h након појављивања инстанце D, и она је занимљива са становишта индукције знања јер омогућава процену исправности хипотеза након посматрања појаве нових инстанци D.

Ова теорема нам омогућава да израчунамо $P(h|D)$ преко израза:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{5.6}$$

У многим проблемима потребно је наћи *maximum a posteriori* (*MAP*) хипотезу, односно највероватнију хипотезу h из H уз услов појављивања D. Примењујући *Bayes*-ов теорем на сваку хипотезу h из скупа H и затим бирајући највероватнију, лако израчунавамо MAP хипотезу:

$$h_{MAP} = \arg\max_{h\in H} P(h|D) = \arg\max_{h\in H} \frac{P(D|h)P(h)}{P(D)} = \arg\max_{h\in H} P(D|h)P(h) \tag{5.7}$$

У изразу 5.7. искључили смо $P(D)$ јер та вероватноћа представља константу независну од h. Уколико претпоставимо да су све хипотезе из скупа H једнако вероватне, тада можемо занемарити утицај параметра $P(h)$ и тада процењујемо h_{MAP} само на основу $P(D|h)$. Хипотеза која максимизира $P(D|h)$ називамо ML (енг. *maximum likelihood*) хипотеза:

$$h_{ML} = \arg\max_{h\in H} P(D|h) \tag{5.8}$$

У класификационим проблемима се доста користи *Bayes*-ова теорема. У овом раду коришћен је метод класификације под називом *Naïve Bayes* класификатор.

5.3.2. Naïve Bayes класификатор

Ако класификацију представимо као проналажење највероватније класификације v_{MAP} тада се она може израчунати на следећи начин:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2 \dots a_n) \quad (5.9)$$

У изразу 5.9. је дат највероватнији елемент коначног скупа V свих могућих класификација улазне инстанце. Ако сваку инстанцу прикажемо као скуп вредности атрибута, и ако је познат скуп тренинг инстанци који је дефинисан такође истим скупом атрибута, онда претходни израз можемо писати као:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2 \dots a_n) \quad (5.10)$$

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2, \dots, a_n | v_j) P(v_j)}{P(a_1, a_2, \dots, a_n)} \quad (5.11)$$

$$v_{MAP} = \arg\max_{v_j \in V} P(a_1, a_2, \dots, a_n | v_j) P(v_j) \quad (5.12)$$

На основу података за тренирање рачунамо вредност израза. Проблем са израчунавањем израза $P(a_1, a_2, \dots, a_n | v_j)$ произилази из међусобне зависности вредности атрибута тако да је број могућих израза једнак броју свих могућих различитих n-торки $\{a_1, a_2, \dots, a_n\}$ помножених са бројем свих могућих класификација. Сваки пример је потребно видети у улазном скупу много пута како би могли поуздано оценити тражене вероватноће.

Овај класификатор уводи поједностављење у виду претпостављене међусобне независности вредности атрибута у n-торкама $\{a_1, a_2, \dots, a_n\}$ тако да се израз може написати као:

$$P(a_1, a_2, \dots, a_n | v_j) = \prod_i P(a_i | v_j) \quad (5.13)$$

Можемо написати израз за класификацију *Naïve Bayes* класификатором на следећи начин:

$$v_{NBj} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (5.14)$$

У овом случају, број различитих вероватноћа које треба израчунати из података за тренинг износи број различитих вредности атрибута помножено са бројем различитих могућих класификација што представља много мањи број него број потребан како би се добила вероватноћа $P(a_1, a_2, \dots, a_n | v_j)$.

У реалним ситуацијама, услов независности који је претпостављен релативно је строг и може представљати проблем, али у практичној употреби *Naïve Bayes* класификатор је показао корисност због једноставности имплементације и задовољавајућих резултата. Ако су заиста сви посматрани атрибути независни, *Naïve Bayes* класификација v_{NBj} је једнака класификацији v_{MAP}.

Као пример коришћења *Naïve Bayes* класификатора можемо посматрати случај одласка у шетњу:

Ако Време. Сунчано и Температура. Хладно и Ветровито. Да и Влажност. Висока,
тада Отићи у шетњу. ДА (0.1) и Отићи у шетњу. НЕ (0.9).

Пробабилистички приступ у откривању знања додељује вероватноћу класификације инстанци у поједине класе, где је 0.1 вероватноћа за *Отићи у шетњу.ДА* 0.1, а 0.9 за *Отићи у шетњу.НЕ*. У овом случају се ради о бинарној класификацији, где је сума вероватноћа за *Отићи у шетњу.ДА* и за *Отићи у шетњу.НЕ* једнака 1. Вероватноћа се одређује френквенцијском интерпретацијом и посматрањем сваког атрибута независно, што представља претпоставку „наивности“.

5.3.3. Предности и недостаци класификације засноване на *Naïve Bayes* класификатору

Naïve Bayes класификатор представља веома брз класификатор, погодан за класификацију јер има мале захтеве што се тиче употребе меморије. Он је једноставна статистичка шема учења и врло се често користи у класификационим проблемима, а некада је успешнији и од многих сложенијих приступа. Робустан је за нерелевантне податке, јер ће се они међусобно поништавати, а такође, добро се показао и у доменима, где постоји велики број подједнако релевантних података. Овај класификатор је оптималан уколико је тачна предпоставка независности података. Можемо закључити да *Naïve Bayes* класификатор има следеће особине:

- мали захтеви за меморијом,
- брзи тренинг и брзо учење,
- једноставност,
- често ради изненађујуће добро.

5.3.4. Псеудо код за *Naïve Bayes* класификатор

На слици 5.4. приказујемо псеудо код за *Naïve Bayes* класификатор који је дао Yang (http://ourmine.googlecode.com/svn/trunk/share/pdf/YangWebb03.pdf).

```
function train (i) {
    Instances++
```

```
        if (++N[$Klass]==1) Klasses++
        for (i=1; i<=Attr; i++)
                if (i !=Klass)
                        if( $i !~/\? /)
                                symbol (i, $i, $Klass)
}
function symbol (col, value, klass) {
        Count [klass, col, value]++;
}
```

Слика 5.4: Псеудо код за *Naïve Bayes* класификатор (http://code.google.com/p/ourmine/wiki/LectureNaiveBayes#Pseudo-code)

Док тестирамо, проналазимо вероватноћу (енг. *likelihood*) сваке хипотетичке класе и враћамо ону која је највероватнија. Слика 5.5. приказује упрошћену верзију вероватноће сваке хипотетичке класе.

```
function likelihood (l, klass, i, inc, temp, prior, what, like) {
        like = -10000000000;
        for (klass y N) {
                prior = N[klass] / Instances;
                temp = prior
                for (i=1; i<=Attr; i++) {
                        if (i !=Klass)
                                if ( $i !~/\? / )
                                        temp *= Count [klass, i, $i] / N[klass]
                }
                l[klass] = temp
                if (temp >= like) {like = temp; what = klass}
        }
        return what
}
```

Слика 5.5: Упрошћена верзија вероватноће сваке хипотетичке класе (http://code.google.com/p/ourmine/wiki/LectureNaiveBayes#Pseudo-code)

Много реалистичнија верзија вероватноће сваке хипотетичке класе приказана је на слици 5.6.

```
function likelihood (l, klass, i, inc, temp, prior, what, like) {
        like = -10000000000;
        for (klass in N) {
                prior = (N[klass] + K) / (Instances + (K* Klasses));
                temp = log (prior)
                for (i=1; i<=Attr; i++) {
                        if (i !=Klass)
                                if ( $i !~/\? / )
```

```
                                        temp+=log((Count[klass,i,$i]+M* prior)/N[klass]+ M))
                }
                l[klass] = temp
                if (temp >= like) {like = temp; what = klass}
        }
        return what
}
```

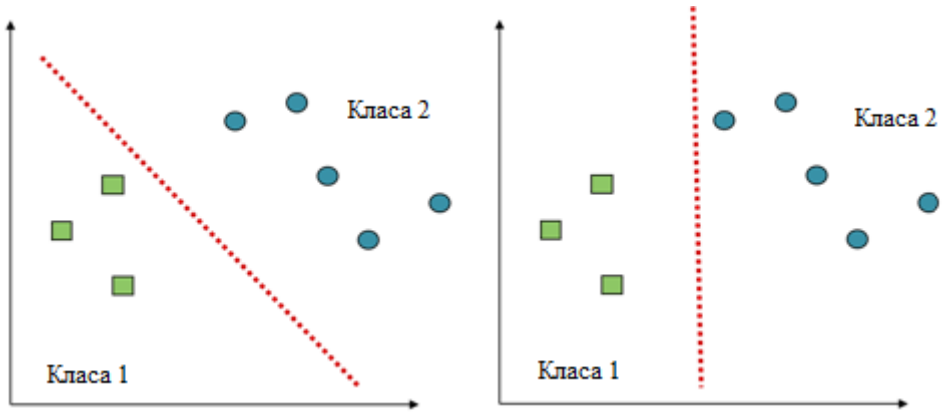
Слика 5.6: Много реалистичнија верзија вероватноће сваке хипотетичке класе (http://code.google.com/p/ourmine/wiki/LectureNaiveBayes#Pseudo-code)

5.4. Метода потпорних вектора

Метода потпорних вектора (енг. *Support Vector Machine* - SVM) је бинарни класификатор који конструкцијом хипер-равни у високо-димензионалном простору ствара модел који предвиђа којој од две класе припада нова инстанца. Ова метода је развијена од стране *Vapnik*-а и сарадника 1995. године и ужива велику популарност због веома добрих резултата који се добијају.

5.4.1. Основне поставке

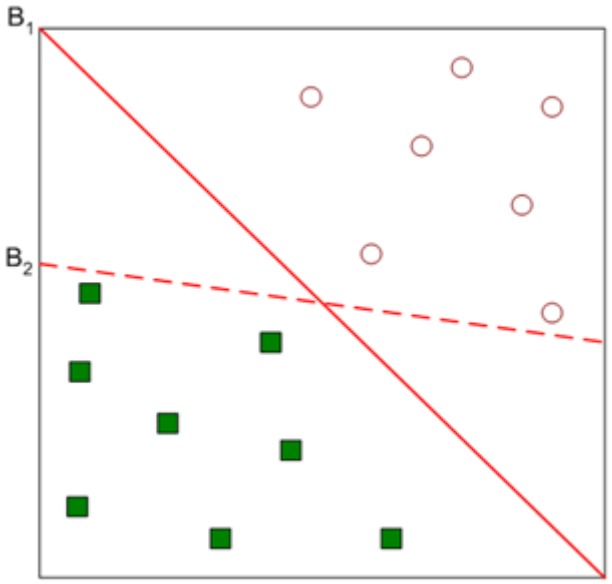
У машинском учењу, метода потпорних вектора је популарна због својих добрих перформанси. Као надзирана метода која анализира податке и препознаје обрасце, она је строго утемељена на статистичким теоријама учења и истовремено смањује тренинг и тест грешке. Основна идеја ове методе је да се у векторском простору у коме су подаци представљени, нађе раздвајајућа хипер-раван тако да су сви подаци из дате класе са исте стране равни, што је приказано на слици 5.7.



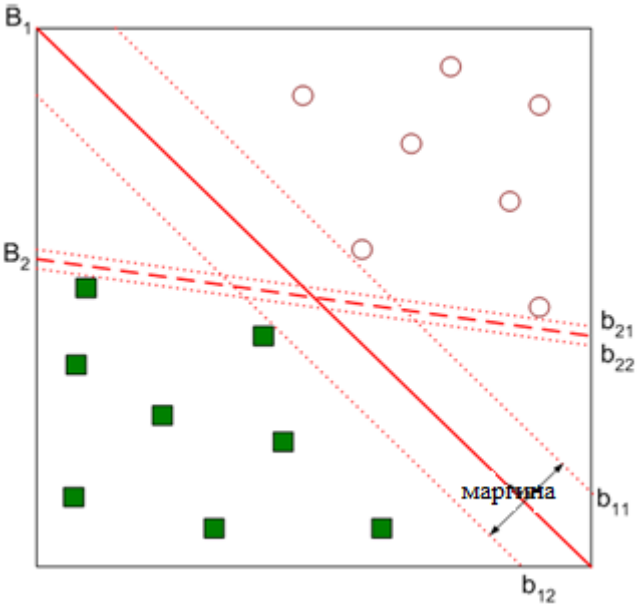
Слика 5.7: Задатак фазе тренинга: нађи оптималну раван која раздваја податке за тренинг, *poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt*

Код коришћења ове методе, поставља се питање које је решење боље и на који начин дефинисати „боље“ решење, што је приказано на слици 5.8.

Ако претпоставимо да су подаци линерано раздвојиви, у фази тренирања треба наћи оптималну раздвајајућу хипер-раван, односно раван са максималном „маргином“ (што претставља растојање од тренирајућих података). У том случају нађена хипер-раван (тј. њена једначина) је модел (слика 5.9). Потом, на основу модела израчунавамо растојање од хипер-равни и на основу тога одређујемо класу (изнад/испод равни).



Слика 5.8: Које решење је боље B_1 или B_2 и како дефинисати „боље“ решење?, *poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt*



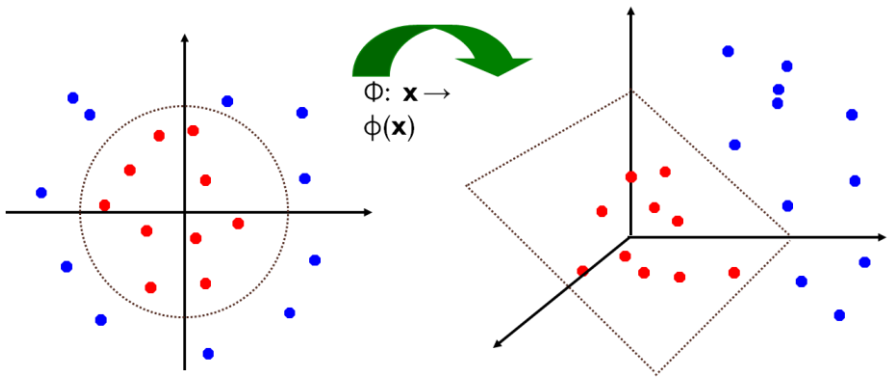
Слика 5.9: Нађи хипер-раван која максимизује величину маргине→ B_1 је боље од B_2 , *poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt*

На слици 5.10. приказан је линеарни класификатор SVM, код кога права $ax + by - c = 0$ представља границу одлучивања.

Слика 5.10: Линеарни класификатор SVM,
poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt

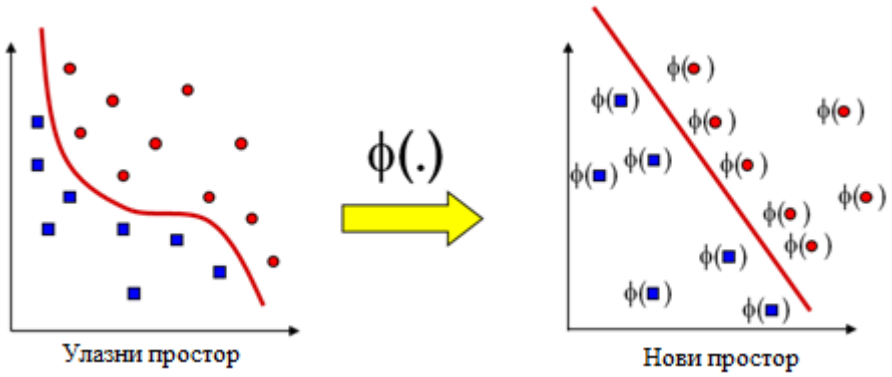
SVM одређује оптимално решење које максимизује раздаљину између хипер-равни и тачака које су близу потенцијалне линије раздвајања и представља интуитивно решење: ако нема тачака близу линије раздвајања, онда ће класификација бити релативно лака.

У случају линеарно не-раздвајајућих проблема, користимо нелинеарни SVM, при чему је основна идеја да се основни (улазни) векторски простор преслика у неки више-димензиони простор у коме је скуп података за тренинг линеарно раздвојив. На слици 5.11. приказано је пресликавање у више-димензиони простор у коме је скуп података за тренинг линеарно раздвојив.



Слика 5.11: Пресликавање у више-димензиони простор у коме је скуп података за тренинг линеарно раздвојив,
poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt

SVM конструише хипер-раван или скуп хипер-равни у високом димензионалном простору, који се може користити за класификацију, регресију, или друге проблеме. Многе хипер-равни могу служити за класификовање података, најбоља хипер-раван је она која представља највеће раздвајање, или маргину између две класе. Генерално говорећи, када је већа маргина онда је мања грешка генерализације класификатора. Изабрана је хипер-раван са максималном маргином, за који важи да је растојање од ње до најближе тачке података на свакој страни максимална. На слици 5.12. приказан је нелинеарни SVM.



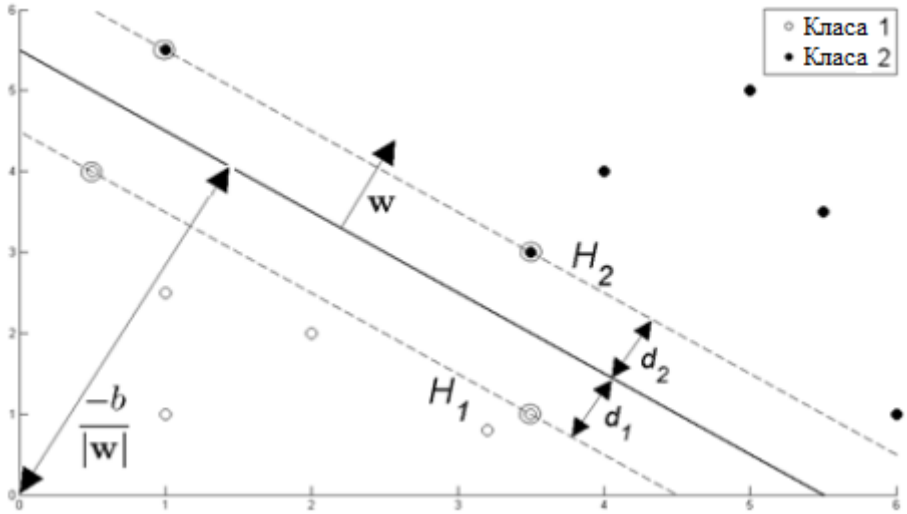
Слика 5.12: Нелинеарни SVM,
poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt

5.4.2. Линеарно одвојиве класе

Ако у скупу за учење имамо L вектора, односно тачака у D-димензионалном простору, где сваки узорак x_i има D атрибута, односно компоненти вектора и припада једној од две класе $y_i = -1\ ili\ 1$, тада облик једног улазног податка можемо приказати изразом:

$$\{x_i, y_i\} \text{ где је } i = 1 \dots L,\ y_i \in \{-1,1\}\ x \in R^D, \tag{5.15}$$

где се претпоставља да су подаци линеарно одвојиви, што значи да можемо нацртати правац у координатном систему са осама x_1 и x_2 за случај $D = 2$, односно хипер-раван за случај $D > 2$. Изразом $w \cdot x + b = 0$ можемо описати хипер-раван при чему је w нормала хипер-равни и $\frac{b}{\|w\|}$ вертикална удаљеност хипер-равни од исходишта координатног система. Узорци најближи раздвајајућој хипер-равни су потпорни вектори и зато се најтеже класификују. Циљ метода потпорних вектора јесте да изабере хипер-раван максимално удаљену од најближих узорака обе класе. На слици 5.13. је дат графички приказ две линеарно одвојиве класе.



Слика 5.13: Приказ две линеарно одвојиве класе [Fletcher, 2009]

На овакав начин се имплементација методе потпорних вектора своди на избор параметара w и b, таквих да улазне податке можемо описати следећим изразима:

$$x_i \cdot w + b \geq +1 \text{ за } y_i = +1 \tag{5.16}$$

$$x_i \cdot w + b \leq -1 \text{ за } y_i = -1 \tag{5.17}$$

Комбиновањем два претходна израза добијамо:

$$y_i(x_i \cdot w + b) - 1 \geq 0, \forall i \tag{5.18}$$

Равни H_1 и H_2 на којима леже потпорни вектори можемо приказати следећим изразима:

$$x_i \cdot w + b = +1 \text{ за } H_1 \tag{5.19}$$

$$x_i \cdot w + b = -1 \text{ за } H_2 \tag{5.20}$$

Ако дефинишемо вредности d_1 и d_2 као растојање од H_1 и H_2 до хипер-равни, еквидистантност хипер-равни од H_1 и H_2 подразумева $d_1 = d_2 = \frac{1}{\|w\|}$, при чему вредност d_1, односно d_2 називамо маргином. Да би изабрали хипер-раван максимално удаљену од потпорних вектора, потребно је максимизирати маргину, што је еквивалентно проналажењу:

$$min\|w\| \text{ такав да } y_i(x_i \cdot w + b) - 1 \text{ за } \geq 0, \forall i \tag{5.21}$$

5.4.3. Линеарно неодвојиве класе

Да би методе са потпорним векторима користили и за на линеарно неодвојиве класе, потребно је ублажити услове (5.16) и (5.17) увођењем ненегативне вредности ξ_i:

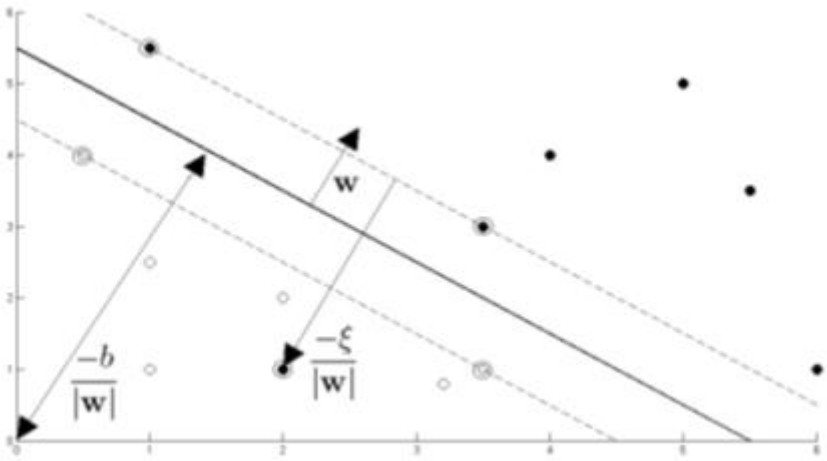
$$x_i \cdot w + b \geq +1 - \xi_i \text{ за } y_i = +1 \quad (5.22)$$

$$x_i \cdot w + b \leq -1 + \xi_i \text{ за } y_i = -1 \quad (5.23)$$

Комбиновањем претходна два израза добијамо следећи израз:

$$y_i(x_i \cdot w + b) - 1 \ + \xi_i \ \geq 0, \ \xi_i \geq 0, \forall i \quad (5.24)$$

Примењена метода се назива метода меке маргине (енг. *soft margin method* [Fletcher, 2009]), а изворно је настала са идејом дозвољавања погрешног означавања класа пре самог поступка учења. Слика 5.14. приказује хипер-раван кроз две линеарно неодвојиве класе, где је видљив и узорак са погрешне стране хипер-равни због којег простор није линеарно одвојив. Мера растојања тог узорка од припадајућег потпорног вектора је ξ.



Слика 5.14: Приказ две линеарно неодвојиве класе [Fletcher, 2009]

Избор раздвајајуће хипер-равни своди се на проналажење:

$$min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{L}\xi_i \text{ такав да } \ y_i(x_i \cdot w + b) - 1 \ + \xi_i \ \geq 0, \forall i \quad (5.25)$$

где вредност C представља фактор грешке, којим дозвољавамо одређене грешке при тренирању, без чега проналазак хипер-равни не би био могућ.

Проблем се може решити употребом методе Лагранжових коефицијената, што се може показати корисним код нелинеарних језгара, и тада се добија ефикасан итеративан алгоритам LSVM. Пример веома једноставне и ефикасне имплементације је SMO (енг. *Sequential Minimal Optimization*), који користи разбијање на најмањи подпроблем где се онда лако одређује вредност једног по једног преосталог коефицијента [Platt, 1999] уместо скупог нумеричког решавања проблема квадратног

програмирања. У наставку рада приказаћемо псеудо код за SMO који смо у раду користили.

Осим SMO, позната је и SVMLight имплементација коју користи програм за етикетирање SVMTool (http://www.lsi.upc.edu/~nlp/SVMTool/http://svmlight.joachims.org/), као и SVR (енг. Support Vector Regression) који користи модел такве функције који користи само део скупа примера а остале игнорише.

У експлоатацији уз дате претпоставке SMV је изузетно ефикасна метода, и није метода учења инстанцама у основном облику - међутим, постоје имплементације (SVMHeavy, [http://www.ee.unimelb.edu.au/staff/apsh/svm/]) које подржавају и „лењо“ учење. Инкрементално, односно лењо учење, насупрот радозналим методама (енг. *eager*, или *batch learning*) је пожељна особина учења ако постоји захтев за сталним мењањем базе знања, где свака таква промена не повлачи понављање целог поступка учења већ само ефикасно инкрементално додавање знања.

Иако обука код SVM може бити захтевнија за велики број примера и класа, она је у суштини линеарно комплексна $O(nm)$ (где је m димензија простора) за разлику од осталих сличних познатих метода машинског учења које махом експоненцијално зависе од m.

5.4.4. Кернел функција

Описани линеарни класификатор се назива класификатор оптималне границе (енг. *maximum margin classifier*). Метода потпорних вектора је уопштени класификатор оптималне границе за нелинеарну класификацију, што се постиже поступком познатим под називом Кернел трик (енг. *Kernel trick*) [Fletcher, 2009]. Основна идеја је да се у изразу (5.25) замени улазни вектор x_i са функцијом $\phi(x_i)$, која улазни вектор пресликава из n-димензионалног у m-димензионални простор, уз $m \gg n$, како би добили узорке који су линеарно одвојиви. Рачунање унутарњег продукта вектора $\phi(x_i)$ и w представља проблем јер је нова димензионалност пуно већа, понекад и бесконачна. Зато се користи Кернел функција (енг. *Kernel function*) $K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j)$, помоћу које је могуће извршити израчунавање на много једноставнији начин.

Да би методе потпорних вектора вршиле добро класификацију потребно је добро одабрати параметаре Кернел функције и раније поменути параметар C – фактор грешке.

Постоји теорија о томе како контруисати исправан кернел за дати проблем. Обично се користе уобичајени кернели или њихова комбинација. Математичка теорија (теорема *Mercer*-а) дефинише услове које дата функција треба да задовољи да би представљала скларни производ у неком векторском простору: симетричност, позитивна дефинитност, итд. Свака функција која задовољи те услове може да буде коришћена као кернел. Наведени примери кернел функција су довољни у већини примена (нарочито ако се узме у обзир затвореност).

Избор одговарајућег кернела за одређену примену је често тежак задатак. Нужан и довољан услов за кернел да би био ваљан је да мора задовољити *Mercer* теорему, али осим тога, не постоји математички структурирани приступ који каже који кернел треба користити. Наравно, очекује се да нелинеарни кернел који се користи у C-SVC буде бољи него линеарни кернел, ако је познато да су подаци не линеарно одвојиви. Избор кернела резултира у различитим врстама C-SVC са различитим нивоима успешности.

Користи се неколико стандардних облика кернел функција:

- линеарна - $K(x_i,\ x_j) = {x_i}^T \cdot x_j$,
- полиномна - $K(x_i,\ x_j) = ({x_i}^T \cdot x_j + a)^b$,
- Гаусова (енг. *Radial Basis Function* - RBF) - $K(x_i,\ x_j) = e^{-\gamma\|x_i - x_j\|^2}, \gamma > 0$,
- рационална квадратна - $K(x_i,\ x_j) = 1 - \frac{\|x_i - x_j\|^2}{\|x_i - x_j\|^2 + \tau}$,
- сигмоидална - $K(x_i,\ x_j) = \tanh(kx_i^T \cdot x_j + a)$.

Линеарни кернел је најједноставнији кернел. Показује добре перформансе за линеарно одвојиве податаке, али зачудо, ради јако добро чак и у случајевима нелинеарних података. Будући да је исти принцип и за линеарни и полиномни кернел, а трансформација доводи до различитог простора решења, перфоманса полиномног кернела се очекује да буде отприлике иста као линеарног кернела. Очекује се да RBF кернел буде пуно бољи од линеарног или полиномног кернела, али овај кернел је тешко дизајнирати, јер је тешко доћи до оптималног γ и одабира одговарајућег *C* који ради најбоље за задати проблем. RBF кернел је често први избор у поређењу са полиномним кернелом, јер број хиперпараметара утиче на сложеност одабира модела, а полиномни кернел има више хиперпараметара од RBF кернела. RBF кернел има најширу примену, која одговара пресликавању у бесконачно-димензионални простор. Све што је линеарно раздвојиво у почетном простору карактеристика, раздвојиво је и у

простору одређеним овом функцијом. Ширину „звона“ *Gaussove* криве одређује параметар γ. За класификацију, сигмоидни кернел није тако ефикасан као што су остала три кернела. Сигмоидни кернел не мора нужно бити позитивно дефинисан, а параметри γ и r морају бити исправно одабрани.

За потребе овог истраживања се користио RBF кернел.

5.4.5. Класификација у случају постојања више класа

Метода потпорних вектора је бинарни класификатор, што значи да разврстава неки непознати узорак у једну од две класе. Ако је потребно извршити класификацију узорака у више од две класе, овај проблем не можемо решити само једним класификатором. Проблем се решава на следећа два начина:

- Први начин је „један-против-свих“ (енг. *one-versuss-all*). Конструисањем n бинарних класификатора од којих сваки разврстава узорке или у једну од класа или у преосталих n-1 класа. Нови узорци се класификују коришћењем стратегије „победник-односи-све“ (енг. *winner-takes-all*), што значи да сваки класификатор, осим излаза, даје и меру сигурности у свој избор. Од свих класификатора чији избор није „all“ узима се избор онога који је најсигурнији у свој избор, а ако сви класификатори одаберу „all“, вероватно се ради о непостојећој класи или узорку којег није могуће класификовати. У случају да сви класификатори одаберу „all“, као избор се најчешће узима избор супротан оном класификатору који је одабрао „all“ са најмањом сигурношћу.
- Други начин је „један-против-једног“ (енг. *one-versus-one*), при чему се конструише $\frac{n(n-1)}{2}$ бинарних класификатора од којих сваки сврстава узорке у једну од две класе. Поступком гласања се врши класификација нових узорака, при чему се свака бинарна класификација сматра једним гласањем за једну од две класе, чиме се број гласова за класу која је одабрана при тој бинарној класификацији увећава за један. Када се изведе свих $\frac{n(n-1)}{2}$ поступака гласања, узорку се додељује класа са највише постигнутих гласова, а у случају да две класе имају једнак број гласова, најчешће се врши избор класе са мањим индексом.

5.4.6. Класификација помоћу библиотеке libSVM

Библиотека libSVM [Chang и Lin, 2001] (енг. *A Library for Support Vector Machines*) садржи подршку за класификацију узорака методом потпорних вектора, уз низ додатних алата који олакшавају припрему улазних података и избор исправних параметара. Библиотека libSVM је имплементирана у програмским језицима *C++*, *Java*, *Python* и *Matlab*. У овом раду је коришћена имплементација у *Java* програмском језику.

Најпре је потребно спровести поступак скалирања улазних података на распон [-1, 1], а након тога потребно је изабрати оптималне параметре C и γ за RBF функцију. Ова библиотека нуди алат за избор оптималних параметара поступком унакрсне валидације у скрипти *grid*. Поступак унакрсне валидације обавља се тако да се скуп улазних података за учење подели у n подскупова, и тада се сваки од n подскупова тестира коришћењем SVM-а наученог на преосталих (n-1) подскупова. Вредност параметара C и γ се експоненцијално повећава и сваки пут се изводи унакрсна валидација, како би се пронашли најбољи параметри. Након проналаска оптималних параметара, прелази се на детаљнију унакрсну валидацију око добијених параметара како би се додатно повећала тачност. Претходно скалирани скуп улазних вредности прима скрипта *grid* и црта граф успешности унакрсне валидације са различитим параметрима.

5.4.7. Псеудо код за SMO алгоритам

Псеудо код за све SMO алгоритме представљамо на слици 5.15.

```
target = desired output vector
point = training point matrix

procedure takeStep(i1,i2)
        if (i1 == i2) return 0
        alph1 = Lagrange multiplier for i1
        y1 = target[i1]
        E1 = SVM output on point[i1] - y1 (check in error cache)
        s = y1*y2
        Compute L, H
        if (L == H)
                return 0
        k11 = kernel(point[i1],point[i1])
```

```
        k12 = kernel(point[i1],point[i2])
        k22 = kernel(point[i2],point[i2])
        eta = 2*k12-k11-k22
        if (eta < 0)
        {
                a2 = alph2 - y2*(E1-E2)/eta
                if (a2 < L) a2 = L
                else if (a2 > H) a2 = H
        }
        else
        {
                Lobj = objective function at a2=L
                Hobj = objective function at a2=H
                if (Lobj > Hobj+eps)
                        a2 = L
                else if (Lobj < Hobj-eps)
                        a2 = H
                else
                        a2 = alph2
        }
        if (a2 < 1e-8)
                a2 = 0
        else if (a2 > C-1e-8)
                a2 = C
        if (|a2-alph2| < eps*(a2+alph2+eps))
                return 0
        a1 = alph1+s*(alph2-a2)
        Update threshold to reflect change in Lagrange multipliers
        Update weight vector to reflect change in a1 & a2, if linear SVM
        Update error cache using new Lagrange multipliers
        Store a1 in the alpha array
        Store a2 in the alpha array
        return 1
endprocedure

procedure examineExample(i2)
        y2 = target[i2]
        alph2 = Lagrange multiplier for i2
        E2 = SVM output on point[i2] - y2 (check in error cache)
        r2 = E2*y2
        if ((r2 < -tol && alph2 < C) || (r2 > tol && alph2 > 0))
        {
                if (number of non-zero & non-C alpha > 1)
                {
                        i1 = result of second choice heuristic
                        if takeStep(i1,i2)
                                return 1
                }
                loop over all non-zero and non-C alpha, starting at random point
                {
```

```
                    i1 = identity of current alpha
                    if takeStep(i1,i2)
                        return 1
                }
                loop over all possible i1, starting at a random point
                {
                    i1 = loop variable
                    if takeStep(i1,i2)
                        return 1
                }
        }
        return 0
endprocedure

main routine:
        initialize alpha array to all zero
        initialize threshold to zero
        numChanged = 0;
        examineAll = 1;
        while (numChanged > 0 | examineAll)
                {
                    numChanged = 0;
                    if (examineAll)
                        loop I over all training examples
                            numChanged += examineExample(I)
                    else
                        loop I over examples where alpha is not 0 & not C
                            numChanged += examineExample(I)
                    if (examineAll == 1)
                        examineAll = 0
                    else if (numChanged == 0)
                        examineAll = 1
        }
```

Слика 5.15: Псеудо код за све SMO алгоритме [Platt, 1999]

5.5. Стабла одлучивања

Примене методе стабла одлучивања укључује решавање проблема попут нивоа комплексности стабла, третман континуираних атрибута, третман атрибута с неодређеним вредностима, побољшања ефикасности алгоритма и сл. У даљем тексту детаљније ћемо се позабавити овим проблемима.

5.5.1. Представљање модела

Учење стабала одлучивања је метода апроксимације дискретних циљних функција у коме се научена функција представља у виду стабла, где сваком чвору стабла одговара тест неког атрибута инстанце, гране које излазе из чвора различитим вредностима тог атрибута, а листовима одговарају вредности циљне функције. Инстанце посматране појаве су описане вредностима својих атрибута. Поступак класификације се врши полазећи од корена, потом спуштајући се низ грану која одговара вредности тестираног атрибута инстанце коју класификујемо и када се дође до листа, класа се додељује инстанци.

Ако стабло одлучивања инстанци додељује неку класу, то значи да инстанца испуњава све услове који су дефинисани путањом од корена до одговарајућег листа кроз стабло и облика су *атрибут=вредност*. Путање кроз стабло представљају коњункције оваквих услова и за сваку класу могуће је уочити путање које се завршавају листовима који одговарају тој класи. Дисјункција свих таквих коњункција дефинише инстанце које припадају датој класи према датом стаблу.

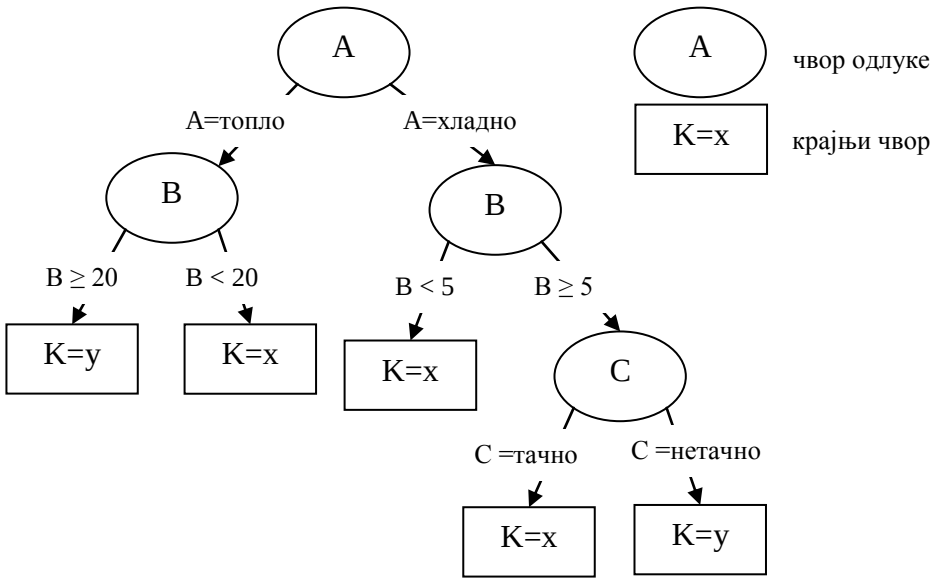
Код стабла одлучивања разликујемо два типа чворова повезаних гранама (Слика 5.16):

- крајњи чвор (енг. *leaf node*) - којим завршава одређена грана стабла и он дефинише класу којој припадају примери који задовољавају услове на тој грани стабла;
- чвор одлуке (енг. *decision node*) - овај чвор дефинише одређени услов у облику вредности одређеног атрибута, из којег излазе гране које задовољавају одређене вредности тог атрибута.

За класификацију примера, стабло одлучивања се може користити тако да се крене од првог чвора одлучивања у корену стабла и креће се по оним гранама стабла који пример са својим вредностима задовољава све до крајњег чвора који класификује пример у једну од постојећих класа проблема.

Стабло одлучивања може се посматрати у форми повезаног графа са структуром стабла, где су унутрашњи чворови у стаблу одлучивања означени са називима атрибута, а гране које излазе из унутрашњих чворова означене су могућим вредностима одговарајућег атрибута. Примери се разврставају у класе које представљају листове стабла одлучивања. У овом случају се класификација спроводи тако што се следи одређени пут од корена стабла до неког од листова. Унутрашњи

чворови стабла представљају тест на вредност одређеног атрибута, па се пут гради додавањем оне гране која одговара вредности атрибута у посматраном примеру, а пут се завршава у неком од листова. На овај начин се пример класификује у класу којом је лист означен.



Слика 5.16: Пример једноставног стабла одлучивања

Код овог модела, простор претраживања се састоји од свих стабала које је могуће конструисати користећи атрибуте и вредности из скупа података за учење. Модификација стабла се врши на начин да се један од листова замени подстаблом висине 1 и то операцијом трансформације којом се прелази простор претраживања.

Поступак претраживања је усмераван функцијом вредновања, која зависи о тачности класификације, али и величини резултирајућег стабла. Деловање функције вредновања засновано је на концептима из теорије информација, а огледа се у избору гранања при конструкцији стабла одлучивања.

Према принципу *Ockham*-ове оштрице, за објашњење неког феномена треба претпоставити што је могуће мање претпоставки, елиминишући тј. одсецајући као оштрицом оне претпоставке, које не доприносе предвиђањима хипотезе или теорије. Када више различитих теорија има једнаку могућност предвиђања, принцип препоручује да се уведе што је могуће мање претпоставки и да се постулира са што је могуће мање хипотетичких ентитета. Ако применимо овај принцип на стабла

одлучивања, уз сличну тачност класификовања, изгледније је да ће једноставнија (тј. мања) стабла одлучивања боље класификовати дотад невиђене примере.

5.5.2. Поступак претраживања

Основни алгоритам конструкције стабала одлучивања стар је неколико деценија, а развио га је J. Ross Quinlan [Quinlan, 1986]. Основна верзија алгоритма позната је под називом ID3 (енг. *Induction of Decision Trees*), док су касније верзије алгоритма уклањале нека од ограничења изворног алгоритма, и побољшавале класификацијске перформансе. Алгоритам конструкције стабала одлучивања C4.5 је данас најпознатији и вероватно највише коришћен алгоритам [Quinlan, 1993].

Код ових алгоритама претраживању се приступа по начелу *одозго на доле*, тј. од општег ка специфичном и користи се стратегија неповратног претраживања и то похлепна метода успона на врх. Претраживање је релативно брзо јер се прегледа само мањи део простора претраживања, али је поступак подложан замци локалних максимума.

Основни алгоритам је у својој основи рекурзиван, што значи да поступак у својој дефиницији користи сам себе, односно захтева да делови проблема које је раздвојио од других бивају независно подвргнути истом поступку.

Ако је са S означен скуп података за учење, са $C=\{C_i,\ 1 \leq i \leq |C|\}$ скуп одговарајућих класа, а са $A=\{A_i,\ 1 \leq i \leq |A|\}$ скуп одговарајућих атрибута, са $S' \subseteq S$ и $A' \subseteq A \subseteq A$ параметри који се прослеђују алгоритму, с тим да се иницијално у рекурзију улази са $S' = S$ и $A' = A$, онда су основни кораци алгоритма [Ујевић, 2004]:

1. Ако је S' празан, стабло одлучивања је лист означен глобално најфреквентнијом класом C_i унутар S.
2. Ако се S' састоји од примера само једне класе C_j, стабло одлучивања је лист означен класом C_j.
3. Ако је A' празан, стабло одлучивања је лист означен најфреквентнијом класом C_k унутар S'.
4. Иначе, изабери атрибут $A_i \in A'$. Ако је $\{a_j,\ 1 \leq j \leq n\}$ скуп свих вредности атрибута A_i. Подели S' на n подскупова S'_j тако да S'_j садржи све примере из S' код којих атрибут A_i има вредност a_j, односно $S'_j = \{s \in S',\ A_i(s) = a_j\}$. Створи унутрашњи чвор означен атрибутом A_i, као и гране означене његовим

вредностима a_j. За грану означену вредношћу a_j конструиши подстабло рекурзивним позивом поступка са параметрима S'_j и $A' - \{A_j\}$.

Алгоритам конструише стабло одлучивања од корена према листовима, при чему се у сваком кораку рекурзије генерише подстабло висине 1, уз то се користе само они примери који припадају том подстаблу. Подстабла се конструишу избором једног од атрибута, при чему су из разматрања искључени сви они атрибути који су пре искоришћени у истој грани стабла. Сваки атрибут се може појавити највише једном на било којем путу од корена до листа. У овом основном облику алгоритма, имплицитно се претпоставља да су сви атрибути номиналног типа. Поступак рекурзије се одвија све док за посматрани чвор стабла није задовољен један од два критеријума. Први критеријум је да је скуп података за учење који припада чвору празан или се састоји од примера само једне класе, због чега је даље гранање непотребно. Други критеријум подразумева да су сви атрибути већ искоришћени на путу од корена до посматраног чвора, због чега даље гранање није могуће.

Једном одабрани атрибут постаје основа за гранање стабла, без могућности да се тај избор накнадно преиспита. Начин избора атрибута је пресудан за квалитет коначног резултата, јер структура стабла одлучивања зависи искључиво о избору атрибута за гранање у сваком кораку рекурзије. То је разлог због кога критеријум избора атрибута за гранање представља централни део алгоритма, који усмерава претраживање у скупу потенцијалних решења. Да ли ће резултирајуће стабло бити гломазна структура претерано прилагођена скупу за учење, или компактни приказ општих правилности које постоје у подацима, зависи од начина избора атрибута.

5.5.3. Начин избора атрибута

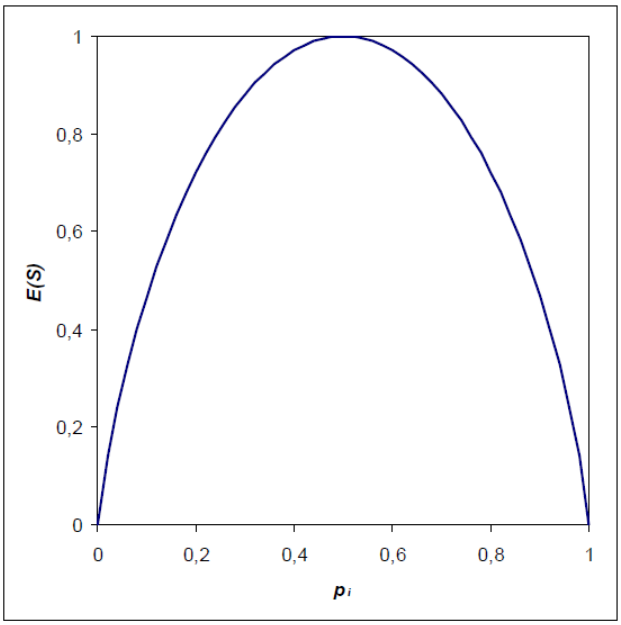
У алгоритму конструкције стабала одлучивања, функција вредновања апроксимације везана је уз избор атрибута који ће послужити као критеријум гранања у унутрашњим чворовима стабла. При томе, треба настојати што раније зауставити рекурзивни процес гранања, јер је циљ конструкција што мањег стабла одлучивања. Базични начин заустављања рекурзије су чворови којима припадају примери само једне од класа, због чега је пожељно као критеријум гранања изабрати оне атрибуте који производе што хомогеније подскупове примера за учење као резултат гранања.

Ентропија или информацијска вредност представља добру меру (не)хомогености неког скупа. За класификацијски проблем са две класе означимо са p_1

релативну фреквенцију класе C_1, а са p_2 релативну фреквенцију класе C_2 у скупу примера за учење S. Следећим изразом можемо претставити ентропију скупа за учење:

$$E(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \tag{5.26}$$

уз претпоставку да важи $0\log_2 0 = 0$. Логаритми у изразу (5.26) су негативни јер је $p_1, p_2 \leq 1$ ($p_1 + p_2 = 1$), па је ентропија увек већа или једнака 0. Минималну вредност 0 ентропија добија када сви примери из скупа за учење S припадају истој класи, тј. када је $p_1 = 1$ или $p_2 = 1$. Максималну вредност 1 ентропија достиже када скуп за учење S садржи једнак број примера обе класе, тј. када је $p_1 = p_2$.



Слика 5.17: Ентропија у зависности од релативне фреквенције класа код бинарне класификације [Ујевић, 2004]

На слици 5.17. приказана је функција ентропије у зависности од релативне фреквенције једне од класа у скупу података за учење за класификацијски проблем са две класе.

Ентропија се мери у битовима, јер једна од интерпретација ентропије из теорије информација каже да она специфицира минималну количину информације (изражену у битовима) потребне да се кодира класификација случајно изабраног примера из скупа S, уз чињеницу да пример припада посматраном чвору. Код класификацијских проблема са више класа потребно је уопштити дефиницију ентропије, али задржати пожељна својства везана за досезање минимума и максимума. Следећим изразом можемо дефинисати ентропију скупа примера S:

$$E(S) = \sum_{i=1}^{|C|} - p_i \log_2 p_i \tag{5.27}$$

при чему $|C|$ означава број класа присутних у скупу података за учење S, а p_i релативну фреквенција класе C_i унутар S, $1 \leq i \leq |C|$. У овом случају максимална вредност ентропије износи $log_2 |C|$, а постиже се при једнакој заступљености свих класа унутар S, док минимум ентропије и даље износи 0, за случај када сви примери из S припадају истој класи.

Информацијски добитак се дефинише на основу ентропије и он служи као мера ефективности атрибута у класификацији примера. Информацијски добитак атрибута A_i у односу на S дефинише се следећим изразом:

$$IGain(S, A_i) = E(S) - \sum_{a_j \in Dom(A_i)} \frac{|S_j|}{|S|} E(S_j) \tag{5.28}$$

где је са A_i означен произвољни атрибут који се појављује у скупу података за учење S, а $S_j \subseteq S$ означава скуп $S_j = \{s \in S, A_i(s) = a_j\}$.

Информацијски добитак $IGain(S, A_i)$ представљен изразом (5.28) представља очекивану редукцију ентропије (тј. добитак на хомогености) узроковану познавањем вредности атрибута A_i, где први члан израза представља ентропију оригиналног скупа S, а тежинска сума у другом члану исказује очекивану вредност ентропије подскупова насталих гранањем на основу атрибута A_i.

Као критеријум за избор атрибута у алгоритму стабала одлучивања користи се управо информацијски добитак, будући да се гранањем настоји што раније постићи хомогеност резултирајућих подскупова. У сваком чвору од доступних атрибута за гранање изабере се онај који производи највећи информацијски добитак. Хеуристика заснована на теорији информација не гарантује конструкцију најмањег стабла одлучивања, али добро испуњава своју улогу редукције опсега претраживања у скупу потенцијалних решења.

Својство информацијског добитка да фаворизује атрибуте са већим бројем вредности, при конструкцији стабала одлучивања може представљати проблем. Екстремни случај је да атрибут има различиту вредност за сваки пример из скупа за учење, при чему гранање на основу овог атрибута партиционира скуп за учење на једночлане подскупове. Ентропија таквог гранања је 0, па је информацијски добитак максималан, а као резултат добијамо стабло које се састоји само од корена и листова за сваки од примера из скупа за учење. У смислу предиктивних способности, добијено стабло је бескорисно.

Да би се смањио утицај овог проблема, користи се корекција критеријума за избор атрибута. Ова корекција у обзир узима број и кардиналност подскупова који настају као резултат гранања. Као критеријум избора атрибута, при конструкцији стабала одлучивања, по правилу се користи коригована мера добитка приказана следећим изразом:

$$Gain(S, A_i) = \frac{IGain(S, A_i)}{-\sum_{a_j \in Dom(A_i)} \frac{|S_j|}{|S|} \log_2 \frac{|S_j|}{|S|}} \tag{5.29}$$

У изразу (5.29) корективни фактор је именилац, који расте с повећањем броја резултирајућих подскупова односно смањењем њихове кардиналности, чиме се пенализирају атрибути с већим бројем вредности. Такође, постоји могућност да корекција информацијског добитка у неким случајевима претерано преферира атрибуте с мањим бројем вредности, на штету информацијски вреднијих атрибута. За елиминисање ове могућности користи се стандардна провера која тражи да атрибут који максимизира кориговани добитак мора имати информацијски добитак једнак или већи од просека свих посматраних атрибута.

5.5.4. Избегавање непотребног гранања стабла

Алгоритам у принципу може генерисати стабло, довољно комплексно да тачно класификује све примере из скупа података за учење, иако је то у одређеним случајевима разумна стратегија, у већини ситуација то рађа додатне проблеме, било због шума у подацима, или пак недовољно великог узорка података који би требао репрезентовати популацију примера за одређени класификацијски проблем. Једноставни алгоритам би генерисао стабло које се претерано добро прилагођава подацима за учење (енг. *over-fitting*).

Значајну потешкоћу у примени метода стабла одлучивања, али и других техника моделирања података представља *over-fitting*. Могућа решења за избегавање *over-fitting*-а су:

- решења која заустављају процес раста стабла пре него што се постигне савршена класификација примера из скупа података за учење,
- решења у којима се најпре генерише стабло које савршено класификује примере, а потом се одређене гране стабла „скраћују“ према претходно дефинисаном критеријуму.

Други се приступ у пракси показао поузданијим, иако се на први поглед први приступ чини директнијим, што је последица тога што је тешко унапред дефинисати жељену комплексност стабла одлучивања.

Одређивање оптималне комплексности, односно величине стабла за конкретни проблеме могуће је уз помоћ следећих приступа:

- коришћење посебног скупа примера, односно валидацијског скупа, који је различит од оног коришћеног за генерисање стабла, да би се оценила успешност „скраћивања“ стабла,
- коришћење посебног статистичког теста на чворовима који су кандидати за „скраћивање“, којима се показује да ли ће се избацивањем тог чвора постићи побољшање,
- коришћење експлицитне мере комплексности кодирања примера стаблом одлучивања, која зауставља раст стабла када је тај критеријум задовољен.

Први приступ се и најчешће користи, и он подразумева да се примери деле у два скупа: скуп за учење који се користи за генерисање стабла, и скуп за проверу, који се користи за проверу ефикасности методе скраћивања стабла.

Напред приказани алгоритам конструкције стабала одлучивања настоји конструисати стабло које савршено класификује примере из скупа података за учење, али у томе не успева само у гранама за које се рекурзија зауставља због недостатка атрибута за гранање. Тада, скуп примера S' који одговара посматраном листу садржи више од једне класе, због чега се лист означава најфреквентнијом класом међу њима. Пробабилистичка интерпретација је алтернатива претходном приступу, и по овом приступу лист се означава свим класама из S', уз придруживање припадајуће вероватноће свакој од њих, што одговара релативној фреквенцији класе унутар S'. Алгоритам врши разгранавање стабла настојећи акомодирати сваки пример из скупа података за учење док год постоје могући атрибути за гранање.

Ипак, савршена класификација примера из скупа података за учење не гарантује добре класификацијске перформансе на дотада невиђеним примерима, а узрок томе може бити чињеница да скуп података за учење није репрезентативан узорак целе популације примера или због постојања шума у подацима за учење, како у прогностичким атрибутима тако и у класи. Као последицу имамо претерано разгранато стабло одлучивања због гранања на атрибутима који само привидно производе

информацијски добитак, док је стварни узрок гранања шум у примерима за учење, и ова појава се назива претерана прилагођеност подацима за учење.

Ако посматрамо примере из скупа података за учење S случајно распоређене у класе C_1 и C_2, односно тако да не постоји корелација између прогностичких атрибута и класе примера, и нека је релативна фреквенција класе C_1 унутар S означена са p, а класе C_2 са $1–p$, тада се без смањења општости може претпоставити $p \geq 0.5$.

У случају најједноставнијег стабла одлучивања које се састоји само од корена означеног класом C_1, очекивана фреквенција грешака износи $1–p$, будући да такво стабло сваки пример класификује у класу C_1.

У случају стабла које се састоји од корена и два листа означена класама C_1 и C_2 и ако тест у корену стабла функционише тако да примеру додељује класу C_1 с вероватноћом q, а класу C_2 с вероватноћом $1–q$, очекивана фреквенција грешака таквог стабла може се претставити следећим изразом:

$$q(1-p) + (1-q)p = p + q - 2pq \tag{5.30}$$

И с обзиром да вреди:

$$1 - p \leq p + q - 2pq, \text{ за сваки } q, \text{ уз } p \geq 0.5 \tag{5.31}$$

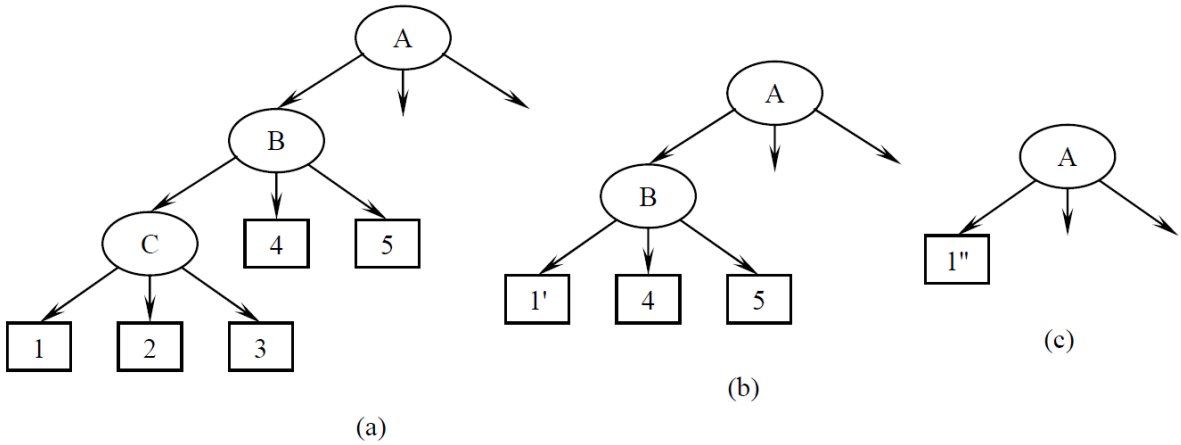
Због овога, класификацијске перформансе једноставног неразгранатог стабла са једним чвором надмашиће било које бинарно стабло са 3 чвора. Постоје два начина за избегавања непотребног гранања стабла. Први начин је *заустављање гранања* приликом конструкције стабла одлучивања пре постизања савршене класификације примера из скупа за учење, док је други начин *накнадно подрезивање* разгранатог стабла које се спроводи након процеса конструкције стабла одлучивања. Сваки од приступа има своје предности. Ефикаснији приступ је заустављање гранања стабла, јер се избегава конструкција непотребних подстабала која опет треба посебним поступком подрезивати, али постоји ризик од прераног заустављања раста стабла. У случају два атрибута које одвојено посматрамо, они се могу се чинити готово неважнима, али њихова комбинација може имати изразите предиктивне способности, због чега је у овом случају боље користити приступ подрезивања, које као подлогу узима потпуно разгранато стабло.

Приликом конструкције стабала одлучивања, критерији за заустављање гранања углавном се ослањају на статистичке технике оцене релевантности атрибута изабраног за гранање, и при томе се често се користи χ^2 тест. Тест χ^2 настоји да утврди статистичку независност вредности атрибута A_i и класе примера у скупу за учење S.

Ако са $p_k(S)$ означимо релативну фреквенцију класе C_k унутар S, а са $p'_k(S_j)$ очекивану релативну фреквенцију класе C_k унутар S_j, и при томе $S_j = \{s \in S, A_i(s) = a_j\}$, и ако су вредности атрибута A_i и класе независне, тада за све скупове S_j који настају као резултат гранања на атрибуту A_i вреди $p'_k(Sj) = p_k(S)$. У том случају израз (5.32) има χ^2 дистрибуцију са $|Dom(A_i)|-1$ степена слободе.

$$\sum_{a_j \in Dom(A_i)} \sum_{k=1}^{|C|} \frac{(p_k(S_j) - p'_k(S_j))^2}{p'_k(S_j)} \qquad (5.32)$$

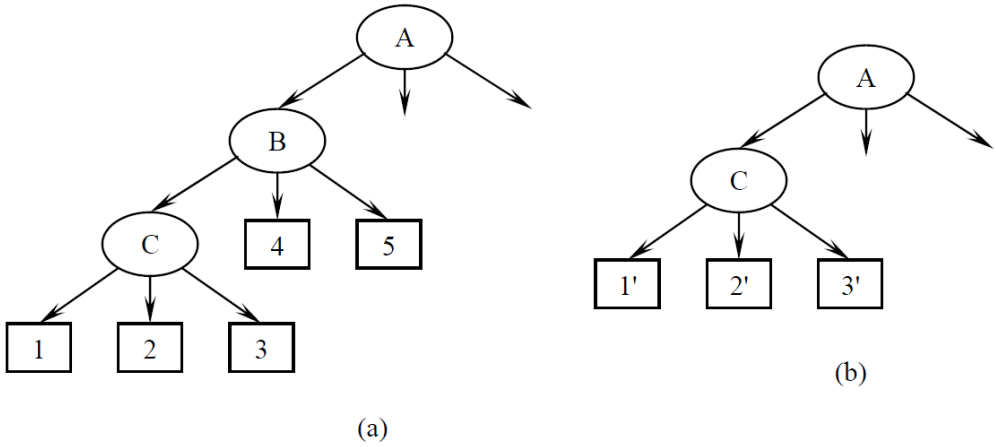
У алгоритму конструкције стабала одлучивања, заустављање гранања се може спровести на начин да се у обзир узимају само они атрибути чија независност у односу на класу може бити одбачена са врло високом поузданошћу. Најпознатије технике подрезивања стабала одлучивања су замена подстабала и издизање подстабала, а примена једне од њих не искључује другу.



Слика 5.18: Пример замене подстабала [Witten *et al.*, 2005]

Једноставнија техника је замена подстабала, која може цело подстабло редуковати на један лист, при чему се мењају вероватноће класа у листу. Код ове технике поступак се спроводи над унутрашњим чворовима стабла који као децу имају само листове (односно посматрају се само подстабла висине 1) и за свако такво подстабло разматра се оправданост замене само једним листом, уз припадајућу корекцију расподеле вероватноће међу класама. Овај поступак се спроводи од листова према корену стабла, тако да се узастопном применом листом могу заменити подстабла произвољне висине. На слици 5.18. дат је пример узастопне примене замене подстабала на чворовима *C* и *B*, тако да је цело подстабло са кореном *B*, на слици (а) у завршном решењу замењено једним листом *1"* на слици (c).

Сложенија операција подрезивања је издизање стабала, а спроводи се над унутрашњим чворовима који као децу имају барем један чвор који није лист, и изабере се једно дете таквога чвора, и њиме се замењује полазни чвор. Цело подстабло под одабраним дететом је издигнуто на један ниво више у односу на полазно стабло. Описаним поступком ефективно нестају остала деца полазног чвора, због чега је потребно примере који су припадали њима рекласификовати у новонастало подстабло.



Слика 5.19: Пример издизања подстабала, где је чвор C издигнут [Witten *et al.*, 2005]

Слика 5.19. приказује стабло одлучивања пре и после примене издизања подстабла на чвору *B*, при чему се чвор *B* замењује дететом *C*, а примери који су припадали осталој деци чвора *B* рекласификују се у ново подстабло са кореном *C*. Рекласификација утиче на вероватноће придружене класама у листовима новог подстабла. Издизање стабала је временски потенцијално захтевна операција, због потребе за рекласификацијом. У практичним имплементацијама, као кандидат за издизање разматра се само најпопуларније дете полазног чвора, односно оно којем припада највише примера за учење.

Код технике подрезивања стабла одлучивања кључан елемент је критеријум на основу којег се одлучује треба ли потенцијалну операцију подрезивања заиста и спровести. Циљ подрезивања је смањење фреквенције грешака на дотада невиђеним примерима, због чега је неопходно проценити фреквенцију грешака у сваком од чворова стабла. Поређењем процењене фреквенције грешака за оригинално подстабло и његове предложене алтернативе добијене подрезивањем, доноси се одлука да ли треба спровести операцију подрезивања.

За процену стварне фреквенције грешака у чворовима стабла одлучивања, користи се више техника, од којих најчешће коришћена користи стандардну

верификацијску технику на одвојеном скупу примера. При томе се изворни скуп примера за учење дели на два дела, скуп који ће служити за генерисање стабла одлучивања и валидацијски скуп који ће служити за проверу оправданости операције подрезивања стабла. Недостатак овог приступа заснива се на чињеници да се стабло одлучивања конструише из мањег броја примера, због издвајања валидацијског скупа. Поред ове, постоје и друге технике које се ослањају на скуп података за учење, а имају хеуристички приступ оцени стварне фреквенције грешака. Статистичка утемељеност поступка је под знаком питања, и поред тога што се користе неким статистичким израчунавањима, јер користи исти скуп података за учење на основу којег је изграђено стабло. У пракси ипак показују добре резултате при одређивању опсежности операција подрезивања, чиме се оправдава њихова примена.

5.5.5. Типови атрибута код алгоритма за конструкцију стабла одлучивања

Основни облик алгоритма за конструкцију стабла одлучивања, а то је ID3, ограничен је на номиналне атрибуте. Први захтев је да циљни атрибут мора имати ограничен број класа, а други захтев је да атрибути који се тестирају у чворовима одлучивања такође морају имати дискретне вредности. Други захтев се може релативно лако задовољити и у случају да је атрибут нумеричког типа, односно у случају реалних нумеричких варијабли, и то претходном дискретизацијом.

Међутим, касније верзије алгоритма на једноставан начин уводе ефикасну подршку раду са нумеричким атрибутима. Коришћењем номиналног атрибута, гранање стабла резултира граном за сваку од могућих вредности посматраног атрибута. Описани облик теста не може се применити на нумеричке атрибуте, због чега се код њих по правилу тестирање вредности ограничава на бинарни тест облика $A_i(s) < x$, где је x константа изабрана за тај тест. Код оваквог теста унутрашњи чвор стабла има две излазне гране, и то једну за позитивни, а другу за негативни исход теста. За описано гранање, припадајући информацијски добитак се рачуна на стандардни начин, уз напомену да нотација суме по различитим вредностима атрибута није прикладна, већ се сумира по (бинарној) партицији скупа за учење. Описани тестови на нумеричким атрибутима суделују у процесу избора атрибута за гранање, заједно са осталим атрибутима који су на располагању.

Једини проблем је избор границе интересантних интервала за формирање теста на нумеричком атрибуту, односно вредности константе x. Због тога што се у скупу

података за учење појављује само коначан скуп његових вредности, уобичајено је да се као кандидати за границу x посматрају аритметичке средине суседних вредности, због чега је потребно сортирати примере за учење према вредности посматраног нумеричког атрибута.

Ако нумерички атрибут A_i у скупу за учење S' може да поприми m различитих вредности, и нека је $(v_j)_{j=1}^{m}$ низ узлазно сортираних вредности атрибута A_i, односно $v_j \leq v_{j+1}$ за сваки $j \in \{1,\ldots, m-1\}$, онда израз (5.33) даје $m-1$ кандидата за вредност границе теста.

$$x_j = \frac{v_j + v_{j+1}}{2}, j \in \{1, \ldots, m-1\} \tag{5.33}$$

Вредност x_j за коју је информацијски добитак максималан бира се за границу теста. Вредности x_j које максимизирају информацијски добитак увек се налазе између примера који припадају различитим класама, због чега је довољно посматрати само оне вредности x_j за које скуп $\{s \in S', A_i(s)=v_j \vee A_i(s)=v_j+1\}$ садржи примере барем две класе.

Потребно је извршити следеће измене алгоритма конструкције стабала одлучивања, ако постоје нумерички атрибути. Прва измена се односи на сортирање примера према вредности сваког од нумеричких атрибута. Чини се да је сортирање потребно спроводити у сваком кораку рекурзије, односно за сваки унутрашњи чвор стабла у којем се разматрају нумерички атрибути, али с обзиром да редослед примера у родитељу индуцира редослед у деци, можемо закључити да је сортирање потребно извршити само једном, у корену стабла. Друга измена се односи на прослеђивање скупа атрибута у следећи корак рекурзије.

При гранању стабла на номиналном атрибуту A_i, свака грана одговара једној вредности тог атрибута, па је свако даље испитивање вредности атрибута A_i у било којој од насталих грана непотребно, због чега се у следећем кораку рекурзије прослеђује скуп атрибута $A' - \{A_i\}$. Међутим, ово не важи код нумеричких атрибута, јер бинарни тест не искоришћава у потпуности информације које атрибут носи, због чега поновно гранање коришћењем истог нумеричког атрибута, али уз другу границу теста, може резултирати повећањем информацијског добитка. Ово значи да вишеструко тестирање истог атрибута на путу од корена до листа има смисла у случају нумеричких атрибута, због чега се код гранања на нумеричком атрибуту у следећи корак рекурзије прослеђује цели скуп атрибута A'.

5.5.6. Недостајуће вредности атрибута

Ако у примерима недостају вредности атрибута, онда настају потешкоће при конструкцији стабла одлучивања и при класификацији примера изграђеним стаблом одлучивања. У неким практичним применама постоје атрибути код којих одређени проценат примера има недостајуће вредности, као на пример, у медицинској области где је чест случај да су одређени резултати лабораторијских тестова доступни само за део пацијената. Тада је уобичајено да се вредности тих атрибута одреде на основу осталих пацијената који поседују резултате тих тестова.

Ако размотримо ситуацију у којој треба израчунати $Gain(S, A)$ за чвор n у стаблу, за атрибут A да би одредили да ли тај атрибут представља кандидата за тест на чвору n. Ако уведемо претпоставке да је вредност $A(x)$ непозната, где $c(x)$ представља вредност класе примера x, онда су могући начини решавања овог проблема:

- први, да се уместо неодређене вредности за атрибут A користи најчешћа вредност за тај атрибут у примерима који се налазе на чвору n.
- други, да се надомешћује са најчешћом вредности тог атрибута код примера исте класе $c(x)$, на чвору n.

У случају да вредност атрибута у примерима није забележена, и да то носи неко значење, онда је оправдано увођење нове вредности типа „непознато“, којом се мењају све недостајуће вредности атрибута у примерима за учење и у примерима за класификацију. Тако уведена вредност се и при конструкцији и при класификацији третира на стандардан начин, па никакве измене алгоритама нису потребне.

5.5.7. Предности и недостаци стабала одлучивања

Ова техника моделирања правилности у подацима је интензивно коришћена и често изучавана, при чему су истраживане и различите варијације поступка конструкције стабала, од различитих критеријума за избор атрибута гранања, до других метода подрезивања стабла, или модификованог облика тестова у чворовима (нпр. коришћењем више од једног атрибута или вредности [Breiman *et al.*, 1984]).

Стабла одлучивања врло су моћна и популарна техника моделирања за класификацијске и предикцијске проблеме, а њена привлачност лежи пре свега у чињеници да нуди моделе података у „читљивом“, разумљивом облику - односно у облику правила. За неке је проблеме од кључне важности само тачност класификације

или предикције модела и у таквим случајевима читљивост модела није од пресудне важности. Но, у другим ситуацијама управо способност интерпретирања модела „људским“ језиком је од кључне важности.

Нарочито важно код примене на великим скуповима података је примена корекције поступка који смањује потребне рачунарске и временске захтеве. Пример једне такве корекције је да се при конструкцији стабла користи само мањи, случајно изабрани део скупа примера за учење, тзв. *прозор*. Овако конструисано стабло класификује преостали део скупа за учење, и издваја погрешно класификоване примере, који се придодају прозору и поступак се итеративно понавља на овако измењеном скупу примера, све док се не постигне задовољавајућа тачност класификације на преосталим примерима. Овај поступак се по правилу зауставља након само неколико итерација и он је приметно бржи од конструкције стабла на целом скупу за учење.

За коришћење технике стабла одлучивања основни предуслови су:

- опис у облику парова вредности-атрибута - инстанце морају бити описане коначним бројем атрибута;
- претходно дефинисан коначан број класа - којима инстанце припадају морају бити дефинисане унапред и треба их бити коначан број;
- класе морају бити дискретне - свака инстанца мора припадати само једној од постојећих класа, којих мора бити знатно мање него број инстанци;
- значајан број инстанци - обично је пожељно да у скупу инстанци за генерисање стабла одлучивања постоји барем неколико стотина инстанци.

Предности коришћења технике стабала одлучивања код класификацијских проблема су:

- способност за генерисање разумљивих модела,
- експлицитно издвајање атрибута битних за одређени класификацијски проблем,
- релативно мали захтеви за рачунарске ресурсе (време и меморија),
- способност коришћења свих типова атрибута (категоричких и нумеричких),
- стабла одлучивања јасно одражавају важност појединих атрибута за конкретни класификацијски проблем.

Техника стабла одлучивања је посебно применљива у случају када је неопходно представљање дисјункција услова, када подаци за тренинг садрже грешке и када у тренинг скупу постоје инстанце којима недостају вредности неких атрибута.

Недостаци коришћења технике стабла одлучивања код класификацијских проблема су:

- да су мање прикладне за проблеме код којих се тражи предикција континуираних вредности циљног атрибута,
- да су склона грешкама у више-класним проблемима са релативно малим бројем инстанци за учење модела,
- да у неким ситуацијама генерисање стабла одлучивања може бити рачунарски захтеван проблем. Тако на пример, сортирање кандидата за тестирање на чворовима стабла може бити захтевно, као и методе „скраћивања“ стабла, код којих је често потребно генерисати велик број стабала да би одабрали оно које је најбоље за класификацију примера одређеног проблема,
- да нису добро решење за класификацијске проблеме код којих су регије одређених класа „омеђане“ нелинеарним кривама у више-димензионалном атрибутном простору. Многе методе стабла одлучивања тестирају у својим чворовима вредности једног атрибута, и тиме формирају правоугаоне регије у више-димензионалном простору.

Ова техника није подједнако погодна за све проблеме учења, на пример у случају када је потребно инстанце представити помоћу вредности фиксног броја атрибута и онда када скуп вредности није дискретан и мали. Ако постоје континуалне вредности атрибута може се применити дискретизација тако што би се скуп поделио у подинтервале, а сваком подинтервалу се придружује ознака која замењује вредности атрибута из тог интервала у записима инстанци. Основни недостатак стабала одлучивања је склоност претераном прилагођавању подацима за учење. При избору технике моделирања за конкретни класификацијски проблем потребно је имати у виду набројане предности и недостатке.

5.5.8. Псеудо код за стабла одлучивања

Општи алгоритам за изградњу стабала одлучивања у псеудо коду [Quinlan, 1993; Kotsiantis, 2007] приказан је на слици 5.20.

1. Проверити основне случајеве
2. За сваки атрибут *a*
3. Нађи нормализовани информацијски добитак од поделе на *a*
4. Нека *a_best* буде атрибут са највећим нормализованим информацијским добитком
5. Направити чвор одлуке који дели на *a_best*
6. Конструисање чвора рекурзивним позивом поступка

Слика 5.20: Псеудо код за C4.5 алгоритам [Quinlan, 1993; Kotsiantis, 2007] (исти као WEKA J48 алгоритам)

5.6. RBF неуронске мреже

У наставку текста биће речи о методама класификације које су засноване на неуронским мрежама. Биће дате основе развоја неуронских мрежа, приказ модела неуронских мрежа и статичке неуронске RBF мреже, предности и недостаци овог алгоритма, као и приказ псеудо кода за RBF мрежу.

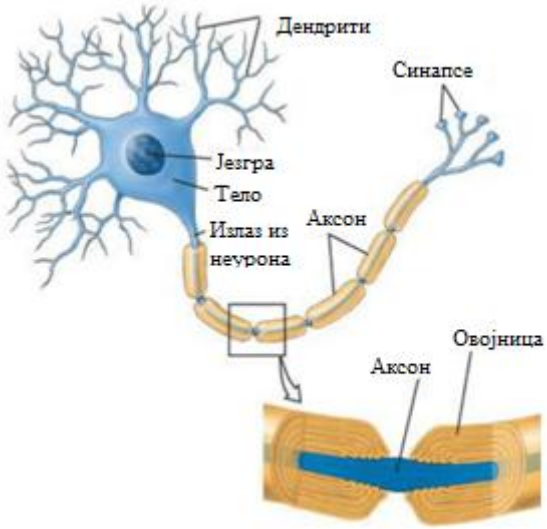
5.6.1. Основе развоја неуронских мрежа

Ове мреже настале су из тежње за развијањем математичких структура које би биле у могућности да опонашају рад људског мозга, као и да користе те структуре у решавању практичних проблема. Постоји више различитих врста неуронских мрежа, али их све можемо сврстати у статичке или динамичке неуронске мреже. У овом раду користи се модел статичке неуронске RBF мреже.

Како би разумели основну структуру вештачких неуронских мрежа потребно је размотрити основну структуру људског мозга. Људски мозак, чија је структура сложена, а мрежа неурона густа, састоји се од око 10^{11} неурона који су међусобно повезани у слојеве, који чине сложену мрежу. Биолошка ћелија која обрађује информације је неурон. Због сложене структуре неурона још увек није дошло до детаљнијих сазнања о функционисању људског мозга.

Биолошки неурон, који је приказан на слици 5.21, послужио је као модел за вештачки неурон. Биолошки неурон се може поједностављено приказати као станица

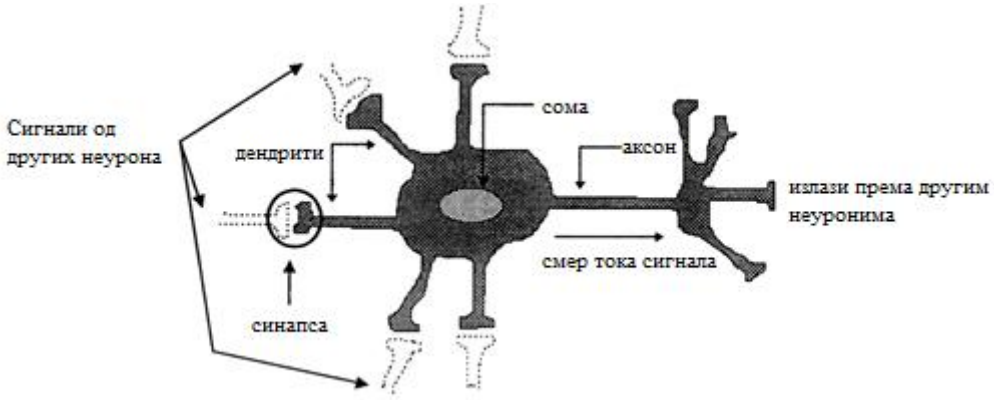
састављена од тела (сома), великог броја дендрита и аксона. Шематски приказ биолошког неурона дат је на слици 5.22.



Слика 5.21: Биолошки неурон, на основу: *The Biological Basis of Behavior*, http://cwx.prenhall.com/bookbind/pubbooks/morris5/chapter2/custom1/deluxecontent

Тело биолошког неурона има нуклеон који садржи информацију о наслеђеним обележијима и плазму која омогућава продуковање сигнала потребних неурону, док се аксон може приказати као танка цевчица чији је један крај повезан на тело неурона, а други се дели на низ грана. Крајеви ових грана завршавају се малим задебљањима која најчешће додирује дендрите, а ређе тело неурона. Синапса се назива мали размак између завршетка аксона претходног неурона и дендрита или тела следећег неурона. Кроз дендрите неурон прима импулсе од осталих неурона, а сигнале које производи тело предаје преко аксона. Код биолошког неурона функција аксона је да формира синаптичке везе с другим неуронима. Тело неурона генерише импулсе који путују кроз аксон до синапси и ти сигнали долазе до дендрита зависно о синаптичком преносу који је условљен већим бројем фактора, између осталих и ранијим синаптичким преносима. На одређен начин, синапсе представљају меморијске чланове биолошке неуронске мреже. Сигнали који дођу до тела другог неурона, могу бити побуђујући или смирујући. Ако је збир сигнала побуђујући, тело неурона ће генерисати импулсе према другима неуронима. Можемо закључити да се рад биолошког неурона одвија кроз две операције: синаптичка операција (придодавање важности улазним сигналима у неурон) и соматска операција („сабирање“ улазних сигнала и генерисање импулса зависно о

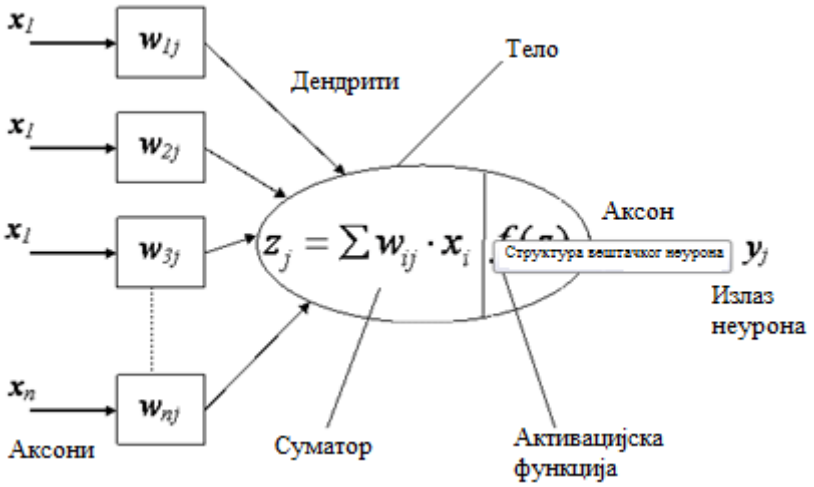
резултату). Дати приказ биолошког неурона уопштено приказује његов рад и поставља оквир у којем су се развили први модели вештачких неурона.



Слика 5.22: Шематски приказ биолошког неурона [Врањеш, 2003]

Неуронска мрежа (енг. *neural network*) је скуп вештачких неурона који су међусобно повезани и интерактивни кроз операције обраде сигнала. Ова мрежа је уређена по узору на рад људског мозга. За вештачки неурон се може рећи да идејом опонаша основне функције биолошког неурона, где се тело биолошког неурона замењује суматором, улогу дендрита преузимају улази у суматор, излаз суматора је аксон вештачког неурона, а улога прага осетљивости биолошких неурона пресликава се на тзв. активацијске функције.

На слици 5.23. представљена је структура вештачког неурона. Функцијске синаптичке везе биолошког неурона са његовом околином пресликавају се на тежинске факторе, преко којих се и остварује веза вештачког неурона са његовом околином.



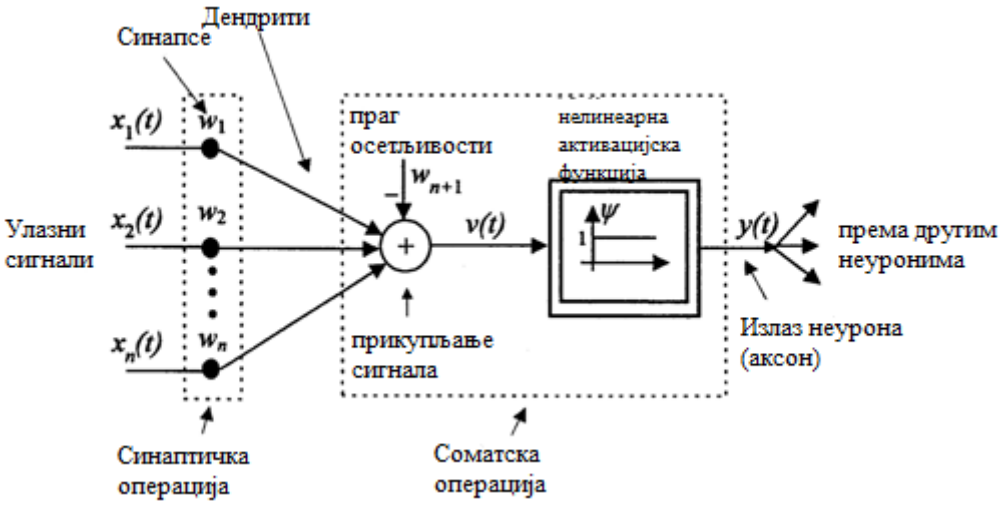
Слика 5.23: Вештачки неурон, на основу *Umjetne neuronske mreže*, http://www.tsrb.hr/meha/index.php?option=com_content&task=view&id=14&Itemid

5.6.2. Вештачки модели неурона

Вештачке моделе неурона могуће је разврстати у две основне групе: статичке и динамичке моделе неурона.

5.6.2.1 Статички модели неурона

Први модел неурона развили су McCulloch и Pitts и он је обрађивао сигнале помоћу претходно наведених операција, соматске и синаптичке. Овај једноставни модел неурона зове се перцептрон. Синаптичку операцију представља множење сваког улазног сигнала x_i са тежинским коефицијентом w_i. Сумирањем тако отежаних сигнала и упоређивањем збира са прагом осетљивости неурона w_{n+1} представља соматску операцију. У случају када је збир већи од прага осетљивости активацијска функција ψ генерише излазни сигнал y износа 1, а ако је збир мањи, генерише се излазни сигнал износа 0. Значајну теорему о учењу перцептрона која гласи: *перцептрон може научити све што може представити,* доказао је Росенблат 1962. године. Представљање је способност апроксимирања одређене функције, а учење поступак који подешавањем параметара мреже постиже то да она постане задовољавајућа апроксимација одређене функције. Шематски приказ перцептрона дат је на слици 5.24.



Слика 5.24: Шематски приказ перцептрона [Врањеш, 2003]

Математички опис перцептрона дат је следећим изразима:

$$v(t) = \sum_{i=1}^{n} w_i(t) \cdot x_i(t) - w_{n+1}, \tag{5.34}$$

$$y(t) = \psi(v), \tag{5.35}$$

где је:

- $\mathbf{x_u}(t) = [x_1(t), \dots, x_n(t)]$ - вектор улазних сигнала неурона, побудни вектор;
- $\mathbf{w_s}(t) = [w_1(t), \dots, w_n(t)]$ - вектор синаптичких тежинских коефицијената;
- w_{n+1} - праг осетљивости неурона;
- $v(t)$ - излаз операције конфлуенције – мера сличности улазних сигнала са синаптичким коефицијентима;
- $\psi(v)$ - активацијска функција;
- $y(t)$ - излаз неурона.

Када се вектор улаза прошири чланом x_{n+1}=1, тада израз (5.36) можемо написати на следећи начин:

$$v(t) = \sum_{i=1}^{n+1} w_i(t) \cdot x_i(t) = \mathbf{w}^T(t)\mathbf{x}(t), \tag{5.36}$$

где су:

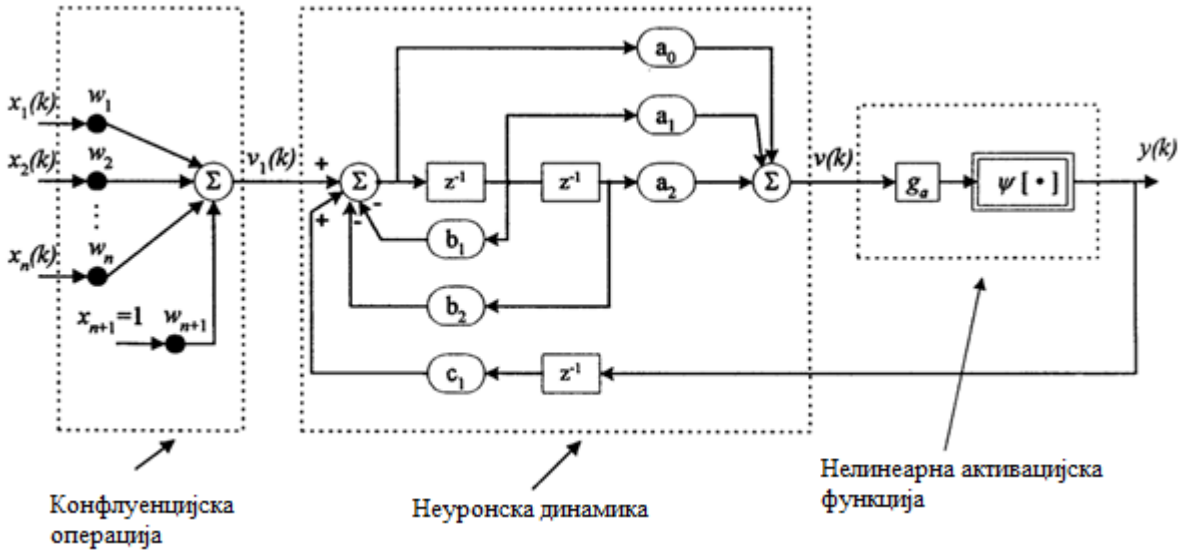
- $\mathbf{x}(t) = [x_1(t), \dots, x_n(t), x_{n+1}(t)]$-проширени вектор улазних сигнала неурона;
- $\mathbf{w}(t) = [w_1(t), \dots, w_n(t), w_{n+1}(t)]$ - вектор синаптичких тежинских коефицијената проширен прагом осетљивости неурона.

Са математичког гледишта, вештачки неурон има две операције: операцију конфлуенције (5.36) и активацијску функцију (5.35). Са биолошког гледишта, операција конфлуенције представља додељивање тежине улазним сигналима $x(t)$ зависно о акумулираном знању у синапсама $w(t)$, док са математичког гледишта, операција конфлуенције представља скаларни производ вектора $x(t)$ и $w(t)$. Меру сличности између проширеног улазног вектора $x(t)$ и вектора тежинских коефицијената $w(t)$ представља излаз операције конфлуенције. Као операцију конфлуенције, већина неуронских мрежа има у себи скаларни производ, али не и RBF неуронске мреже код којих се уместо скаларног производа користи Еуклидско растојање између вектора $x(t)$ и $w(t)$. Пресликавање излазне вредности операције конфлуенције $v(t)$ у излазни сигнал неурона $y(t)$ ограничен унутар [0,1] за униполарне и [-1,1] за биполарне сигнале, врши активацијска функција $\psi(v)$.

Због тога што статички неурон не садржи динамичке чланове, његов излаз зависи само о тренутним вредностима улазних сигнала и тежинским коефицијентима, што га чини структурно стабилним.

5.6.2.2. Динамички модел неурона

Код динамичког модела неурона излаз зависи осим о тренутним вредностима улаза и тежинским коефицијентима синаптичких веза, и о прошлим стањима синаптичких веза. Динамички модел неурона омогућава простирање сигнала и унапред, али и уназад преко унутрашњих повратних веза. На слици 5.25. приказан је уопштени динамички модел неурона, који се састоји од операције конфлуенције, дискретног динамичког члана другог реда, нелинеарне активацијске функције промењивог угла и повратног сигнала са излаза неурона.



Слика 5.25: Уопштени динамички модел неурона [Врањеш, 2003]

Уопштени модел динамичког неурона можемо приказати следећим изразима:

$$v_1(k) = \sum_{i=1}^{n+1} w_i(k) \cdot x_i(k) \tag{5.37}$$

$$v(k) = a_0 v_1(k) + a_1 v_1(k-1) + a_2 v_1(k-2) + c_1 a_0 y(k-1) + c_1 a_1 y(k-2) + c_1 a_2 y(k-3) + c_1 a_2 y(k-3) - \frac{b_1}{a_2} v(k-1) - \frac{b_2}{a_2} v(k-2) \tag{5.38}$$

$$y(k) = \psi\left[g_a \cdot v(k)\right] \tag{5.39}$$

где је *k* ознака дискретног времена. Ако у уопштеном моделу динамичког неурона неким параметрима придружимо непромењиве вредности, онда добијамо моделе неурона често коришћених неуронских мрежа:

- MLP мреже: $a_0=1,a_1=a_2=b_1=b_2=c_1=0,g_a=1;$
- повратне неуронске мреже: $a_0=1,a_1=a_2=b_1=b_2=0,g_a=1;$
- неуронске мреже са временским кашњењем: $b_1=b_2=c_1=0,g_a=1;$
- динамичке неуронске мреже: $c_1=0.$

5.6.3. RBF мреже

Ове неуронске мреже су двослојне статичке неуронске мреже, где нулти (улазни) слој прослеђује улазе у мрежу на улаз првог слоја сачињеног од неурона са активацијским функцијама са кружном основицом и представља њено тзв. рецептивно поље.

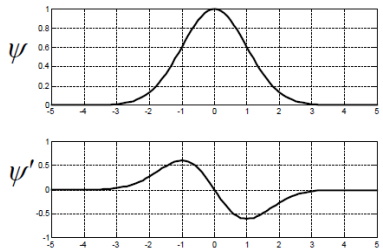
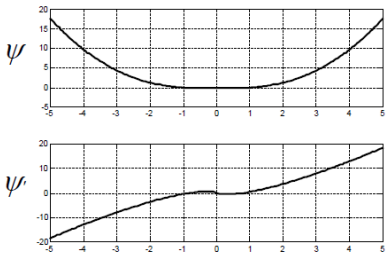
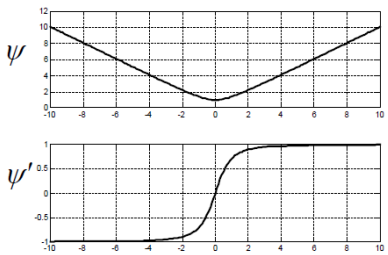
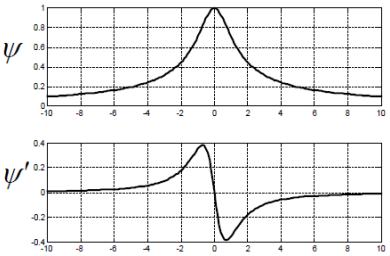
Други слој је слој мреже, који је уједно и њен излазни слој, и састоји се од перцептрона са линеарном активацијском функцијом јединичног активацијског појачања. Активацијске функције у првом слоју RBF мреже које су најчешће коришћене приказане су у табели 5.1.

RBF мрежа има способност апроксимације произвољне континуиране нелинеарне функције, и њена апроксимациона способност одређена је положајем средишта RBF неурона, варијацијом активацијских функција, као и износима тежинских коефицијената излазног слоја мреже. Алгоритмима учења се израчунавају одговарајуће вредности ових параметара RBF мреже. RBF неуронске мреже се посебно користе у случају апроксимација једноставних и временски мало променљивих нелинеарности када је могуће унапред на одговарајући начин распоредити средишта и одредити износе варијансе RBF неурона, а учење мреже се може свести само на подешавања тежинских коефицијената излазног слоја. Понашање RBF неуронских межа, у овом случају, постаје линеарно зависно о параметрима.

Својства RBF мреже значајно одређује распоред средишта RBF неурона. RBF функције се традиционално користе за интерполацију нелинеарних више варијабилних функција, при чему је број средишта једнак броју података, тако да се у сваки улазни податак поставља по једно средиште. Апроксимацију произвољне нелинеарне

континуиране функције могуће је постићи и са мањим бројем добро распоређених средишта.

Табела 5.1. Најчешће коришћене активацијске функције код RBF модела [Петровић, 2009]

Назив функције	Израз за функцију и њену деривацију	Графички приказ функције и деривације
***Gauss*-ова функција**	$\psi(v) = e^{\frac{v^2}{2\sigma^2}}$ $\psi'(v) = -\frac{v}{\sigma^2} e^{\frac{v^2}{2\sigma^2}}$	
***Thin-plate-splin* функција**	$\psi(v) = v^2 \ln(v)$ $\psi'(v) = 2v \ln(v) + v$	
Вишеквадратна функција	$\psi(v) = \sqrt{v^2 + \sigma^2}$ $\psi'(v) = \frac{v}{\sqrt{v^2 + \sigma^2}}$	
Инверзна вишеквадратна функција	$\psi(v) = \frac{1}{\sqrt{v^2 + \sigma^2}}$ $\psi'(v) = \frac{v}{(v^2 + \sigma^2)^{\frac{3}{2}}}$	
Напомена: $\boldsymbol{v \geq 0, \sigma > 0}$; У примерима $\sigma = 1$.		

У својим радовима Broomhead и Lowe [Broomhead и Lowe, 1988] су предложили да се средишта поставе у случајно одабране улазне податке. Постоји и могућност једноликог распореда средишта у простору улазних података. Варијансе активацијских функција мање утичу на понашање мреже и обично се изаберу као други корен производа растојања неурона од два најближа суседна неурона према Moody и Darken, 1989. године [Moody и Darken, 1989]. Ове мреже и са случајним равномерним

распоредом средишта RBF неурона могу апроксимирати произвољну континуирану нелинеарну функцију, али потребни број RBF неурона може бити јако велики. Проширењем поступка учења мреже и на подешавање положаја средишта можемо постићи смањење броја RBF неурона. Понашање RBF мреже, у овом случају, постаје нелинеарно зависно о параметрима, али и са упоредивим апроксимацијским својствима [Петровић, 2009].

Табела 5.1. приказује најчешће коришћене активацијске функције код RBF модела. У овом раду користе се активацијске функције са *Gauss*-ovom функцијом и овакве RBF неуронске мреже још се називају *Gauss*-ove RBF неуронске мреже.

5.6.4. Тренинг RBF мрежа

Оптимална архитектура RBF мреже се обично одређује експериментално, али неке практичне смернице постоје. Процедура решавања проблема помоћу неуронских мрежа се састоји од: прикупљања и припреме података, тренинга мреже, тестирања мреже, и одређивања оптималних параметара мреже и тренинга експерименталним путем (број неурона, број слојева неурона, параметри алгоритма за учење и подаци за тренинг).

Припрема података за RBF мреже обухвата: филтрирање, нормализацију и редукцију димензионалности. Успех решавања у потпуности зависи од података који се користе за тренинг мреже. Потребно је водити рачуна о теоријској оправданости – репрезентативности коришћених података за одређени проблем. Ово је врло специфично у зависности од проблема који се решава. Тренинг RBF мреже обухвата: одређивање оптималних параметара мреже и алгоритма за тренинг, одређивање броја скривених слојева и броја неурона у сваком слоју (више не значи боље, циљ је имати што мање), динамичко подешавање параметара, валидацију параметара (са пробним скупом), одређивање тренинг и тест скупа података и решавање проблема претрениравања и генерализације.

Тренинг излазних тежина (енг. *outputs weights*) је једноставан када излазни неурони користе линеарну активацију. У RBF мрежи постоје три врсте параметара које је потребно да буду одређени за прилагођавање мреже одређеном задатку: средишњи вектори C_i, излазне тежине ω_i и RBF параметри ширине β_i. У секвенцијалном тренингу тежине се ажурирају у сваком временском кораку. За неке задатке има смисла дефинисати функцију циља и одабрати вредности параметара које минимизирају њену

вредност. Најчешћа циљна функција је најмања квадрата функција, која експлицитно укључује зависности од тежина.

$$K(\omega) \stackrel{\text{def}}{=} \sum_{t=1}^{\infty} K_t(\omega) \tag{5.40}$$

где

$$K_t(\omega) \stackrel{\text{def}}{=} [y(t) - \varphi(x(t), \omega)]^2. \tag{5.41}$$

Минимизација најмање квадратне циљне функције уз помоћ оптималног избора тежина оптимизује тачност.

Постоје ситуације у којима више циљева, попут глаткоће и тачности, мора бити оптимизиран. У том случају је корисно оптимизовати регулисану циљну функцију као

$$H(\omega) \stackrel{\text{def}}{=} K(\omega) + \lambda S(\omega) \stackrel{\text{def}}{=} \sum_{t=1}^{\omega} H_t(\omega) \tag{5.42}$$

где $S(\omega) \stackrel{\text{def}}{=} \sum_{t=1}^{\omega} S_t(\omega)$ и $H_t(\omega) \stackrel{\text{def}}{=} K_t(\omega) + \lambda S_t(\omega)$ при чему оптимизација C максимизира глаткоћу и λ.

5.6.5. Својства класификације неуронским мрежама

На тежим класификацијским проблемима, класификација неуронским мрежама се показала врло добром управо код оних проблема код којих је тешко или немогуће користити класичне технике симболичког учења. Поред овога, неуронске мреже су добро прилагођене класификацији у условима шума у подацима.

Ова мрежа има способност апроксимације произвољне нелинеарне континуиране функције, при чему три параметра одређују њену апроксимацијску способност, а то су: положај средишта неурона, варијансе активацијских функција неурона и тежински коефицијенти излазног слоја мреже.

Коришћењем различитих алгоритама учења ови параметри се подешавају да би се добило одговарајуће понашање мреже. Ове мреже су посебно ефикасне у случајевима када је могуће унапред распоредити средишта неурона и одредити износе варијанси RBF неурона, чиме се учење мреже своди на подешавање тежинских коефицијената излазног слоја. У овом случају, понашање RBF неуронске мреже постаје линеарно зависно о параметрима, што је велика предност, али да би се добили квалитетни резултати на овај начин, потребан је јако велики број неурона. Како би ово избегли, поступак учења мреже проширује се и на подешавање средишта и варијанси неурона RBF мреже, како би се знатно смањио број RBF неурона. На овај начин, понашање RBF мреже постаје нелинеарно зависно о параметрима.

Недостатак неуронских мрежа је релативно спор и захтеван процес индукције модела у поређењу са класичнијим техникама, чак до неколико редова величине [Quinlan, 1994].

Још један важан недостатак је и чињеница да класификацијски модел репрезентован неуронском мрежом није експлицитно изражен, у облику структурног описа важних односа међу варијаблама. Модел је имплицитан и скрива односе варијабли у мрежној структури и великом броју тежинских вредности, није разумљив ни подложан верификацији или интерпретацији у оквиру домена изворног класификацијског проблема.

5.6.6. Псеудо код

Псеудо код за RBF тренинг приказан је на слици 5.26.

```
trainRBF (in, out, width, MaxError, data) {
        hidden = 0;
        net = initRBFNetwork (in, out, hidden);
        do {
                / / пронаћи вектор података који производи највећи грешку
                i = findMaxNetworkError (data, net); / / i = индекс вектора
                 / / додавање неурона RBF слоју на истом месту где је вектор података
                addRBFNeuron (net, width, data (i)); / / data (i) = средишња тачка
                 / / пронаћи укупну грешку мреже
                NetError = trainOutputWeights (net, data);
        } while (NetError> MaxError);
}
```

Слика 5.26: Псеудо код за RBF тренинг [Russell и Norvig, 2003]

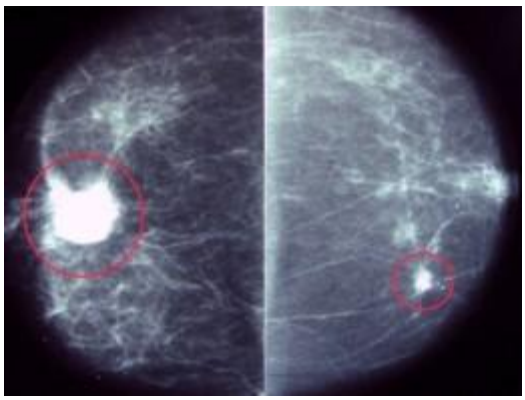
ШЕСТИ ДЕО

6. ОПИС ИЗАБРАНИХ ПРОБЛЕМА УЧЕЊА

У шестом делу дисертације биће дат приказ изабраних проблема учења, које ћемо у експерименталном истраживању користити за доказ постављених хипотеза.

За потребе експерименталног истраживања користили смо 15 реалних скупова података и 3 вештачка, преузета из UCI репозиторијума [Frank и Asuncion, 2010], који је намењен истраживачима који прочавају проблеме вештачке интелигенције.

Рак дојке (breast cancer – bc): задатак овог сета података је да предвиди да ли има или нема повратка болести рака дојке код пацијената. Предвиђање се ради на основу година (при чему су пацијенти разврстани по следећим категоријама годишта: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99), наступања менопаузе (пре 40 година, после 40 година, или није дошло до менопаузе), величине тумора (при чему је величина тумора разврстана у следеће категорије: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59), величине чворова (разврставање је урађено по следећим категоријама: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39), степена малигнитета (степен малигнитета је разврстан у три категорије: 1, 2, 3), захваћене дојке тумором (лева дојка, десна дојка), положаја тумора (лево доле, лево горе, десно доле, десно горе, централно) да ли је вршено зрачење или не код пацијента.



Слика 6.1: Мамографски снимци дојке [http://weinsteinimaging.com]

На слици 6.1. приказани су мамографски снимци дојке. У овом сету података постоји 201 инстанца једне класе (нема повратка болести рака дојке) и 85 инстанци

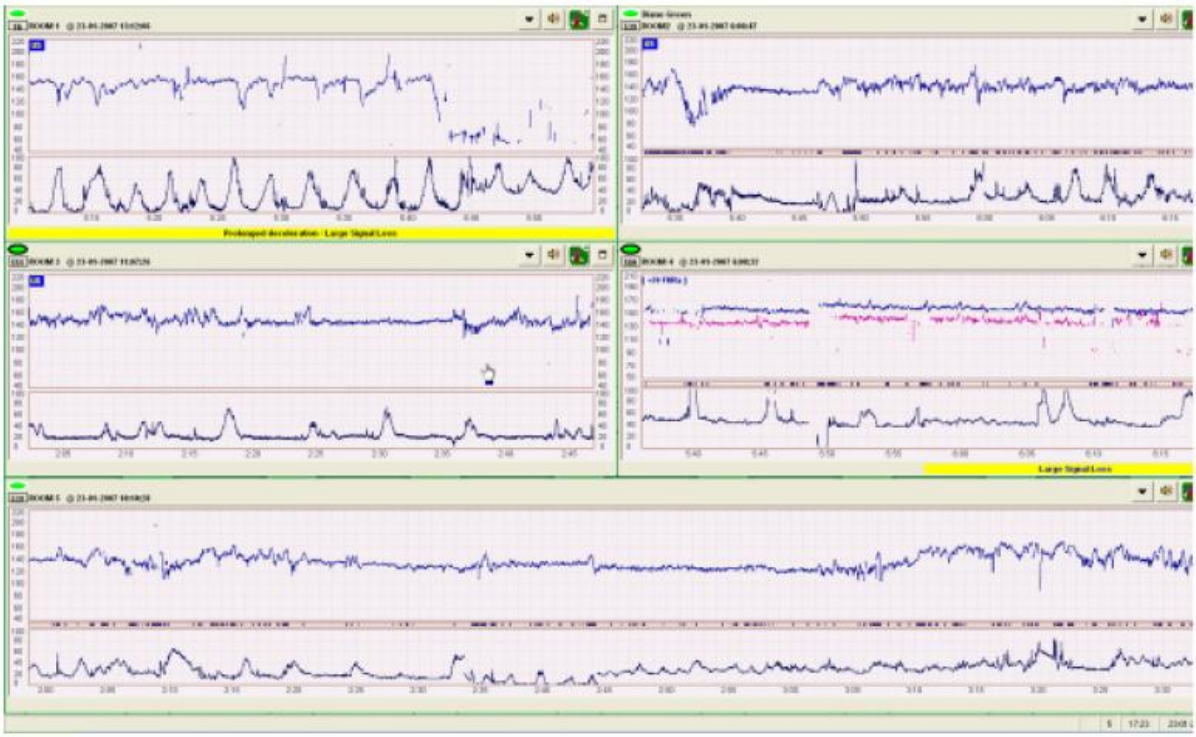
друге класе (има повратка болести рака дојке). Свака инстанца која се односи на стање једног пацијента је описана са 9 атрибута. У овом скупу података постоје вредности које недостају, односно непостоје вредности за све атрибуте свих инстанци.

Одобравање кредита (credit approval - ca): овај сет садржи податке који се односе на коришћење кредитне картице [Quinlan, 1987; Quinlan, 1993]. Код овог сета података, сви атрибути имена и вредности су промењене у бесмислене симболе како би се заштитила тајност података. Овај сет података је интересантан за истраживање јер постоји добра мешавина атрибута – категоричких и нумеричких вредности. Сет података за одобравање кредита садржи 690 инстанци, 15 атрибута и две класе чије су вредности одобрити кредит или га не одобрити (једна класа је заступљена са 44.5%, а друга класа је заступљена са 55.5%). У овом сету података у 37 случајева (5% свих случајева) недостаје једна или више вредности.

Кредитни подаци (Statlog german credit data - cg): овај скуп података омогућава класификовање потенцијалних корисника кредита на оне који имају мали или висок ризик за одобравање кредита. Ово класификовање се врши на основу статуса постојећег текућег рачуна и времена када је он отворен, кредитне историје (да ли је корисник до сада узимао кредит и да ли је био редован приликом његовог враћања), сврхе кредита (нови ауто, постојећи аутомобил, намештај/опрема, радио/ТВ, кућни апарати, поправке, образовање, одмор, преквалификација, пословни разлози и други), износа кредита, штедног рачуна/обвезница, садашњег запослења (и ако није запослен, колико дуго је без посла), вредности рате у односу на расположив доходак, особни статус (разведен, у браку, самац), пола, да ли је корисник дужник/јемац по неком другом кредиту, садашње пребивалиште (колико дуго је држављанин) и својство некретнине, постојање полисе осигурања живота, старост у годинама, броја постојећих кредита у тој банци, радног односа и врсте радног односа и врсте посла које обавља (неквалификован, квалификован и висококвалификован радник), броја људи који могу гарантовати за кредит и да ли је корисник кредита страни радник или не. Овај скуп података има 1000 инстанци и 20 атрибута (7 нумеричких и 13 категоричких).

Ултразвук (cardiography – ct): овај скуп података састоји се од атрибута мерења феталног откуцаја срца и атрибута контракције материце на ултразвуку које су класификовали доктори [Ayres de Campos *et al.*, 2000]. Слика 6.2. приказује фетални кардиограм. Фетални кардиограми су аутоматски обрађени и одговарајући дијагностички показатељи су мерени. Феталне кардиограме су разврстали три експерт

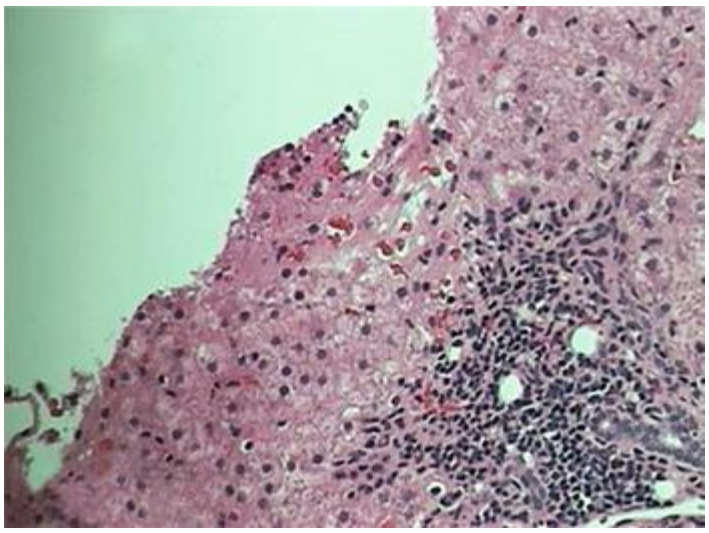
акушера и сваком од кардиограма додељена је одређена класа. Класификација је урађена у односу на морфолошке обрасце и на стање фетуса. Овај сет података садржи 2126 инстанци и 23 атрибута. Атрибути који су посматрани у овом сету података се односе на: брзину откуцаја у минути, број убрзања у секунди, број феталних покрета у секунди, број контракција материце у секунди, број лаких успорења у секунди, број тешких успорења у секунди, број продужених успорења у секунди, проценат времена са абнормалним краткорочним варијабилностима, средњу вредност краткотрајне варијабилности, проценат времена са абнормалним дугорочним варијабилностима, средњу вредност дуготрајне варијабилности, ширину хистограма, минимум на хистограму, максимум на хистограму, број пикова на хистограму, број нултих вредности на хистограму, мод хистограма, средњу вредност хистограма, варијансу хистограма и тренд хистограма. Сет података се може користити у експериментима који користе 10 класа (класе су разврстане бројчано од 1 до 10) или 3 класе (класе су разврстане као нормално, сумњиво и патолошко стање).



Слика 6.2: Фетални кардиограм [Ayres-de-Campos *et al.*, 2008]

Хепатитис (hepatitis – he): главни циљ овог скупа података је предвидети хоће ли са хепатитисом пацијенти умрети или не. Ово предвиђање се врши на основу стања пацијента и то: његовог узраста (развстано по класама годишта: 10, 20, 30, 40, 50, 60, 70, 80), пола, коришћења стероида (вредности могу бити да или не), коришћења

антивирусних лекова (вредности могу бити да или не), постојања умора (вредности могу бити да или не), малаксалости (вредности могу бити да или не), анорексије (вредности могу бити да или не), величине јетре (увећана јетра или не) и облика јетре, болести слезине, билирубина (вредности су разврстане у следеће категорије: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00), албумина (вредности су разврстане у следеће категорије: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0), АЛК фосфата (вредности су разврстане у следеће категорије: 33, 80, 120, 160, 200, 250), хистопатологије и сл. [Diaconis и Efron, 1983; Cestnik *et al*., 1987]. Слика 6.3. приказује ткиво јетре и патолошке промене на њему услед присуства хроничног хепатитиса Ц. У овом скупу података, постоје две класе за предвиђање: прва класа која предвиђа да ће пацијент преживети (123 инстанце) и друга класа која предвиђа да ће пацијент умрети (32 инстанце). Овај скуп података садржи 155 инстанци и 19 атрибута, са вредностима које недостају за поједине атрибуте.



Слика 6.3: Ткиво јетре и патолошке промене на њему услед присуства хроничног хепатитиса Ц [http://www.cpmc.org/advanced/liver/patients/topics/HepatitisC-profile.html]

Јетра (liver disorders - li): у скупу података под називом јетра, првих пет атрибута су тестови крви пацијената и то: запремина еритроцита, алкална фосфатаза, аланине аминотрансферазе, аспартат аминотрансфераза, гама-глутамил транспептидасе; док се друга два атрибута односе на број попијених алкохолних пића, и да ли је пацијент свакодневно пијан. Сматра се да ови атрибути указују на болести јетре, која би могла произаћи између осталог и из претераног конзумирања алкохола.

Слика 6.4. приказује оштећену јетру услед претераног конзумирања алкохола. Свака инстанца у скупу података односи се на податке једног мушког пацијента који се

подвргао тесту. У овом скупу података постоје 345 инстанце и 6 атрибута, без вредности које недостају за атрибуте.

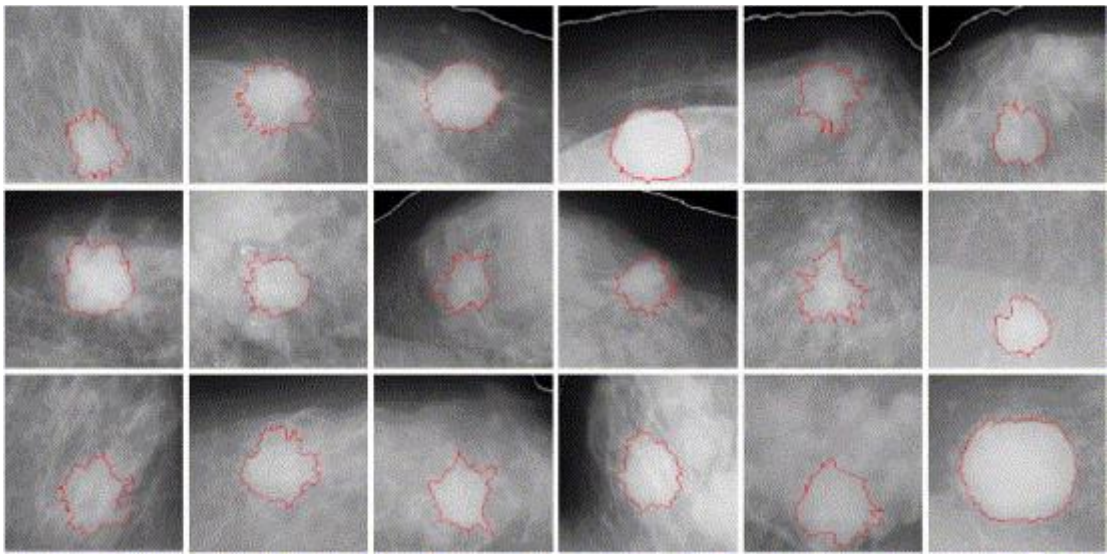
Слика 6.4: Алкохолом оштећена јетра
[http://www.treatment4addiction.com/addiction/alcohol/liver-damage/]

Слика 6.5: Рентгенски снимак рака плућа [http://medigalenic.blogspot.com/2009/12/lung-cancer-treatments.html]

Рак плућа (lung cancer –lc): сет података за рак плућа садржи податке који описују три врсте патолошког облика рака плућа. Рентгенски снимак рака плућа приказан је на слици 6.5. Ове податке су прво користили истраживачи Hong и Young за илустрацију добрих перформанси оптимално дискриминативних равни, чак и у лоше датим поставкама [Hong и Yang, 1991].

Аутори не дају никакву информацију о појединачним варијаблама, нити о томе где су подаци изворно коришћени. Постоје 32 инстанце и 56 атрибута, са вредностима које недостају за поједине атрибуте. Сви атрибути у овом скупу података су нумерички и имају целобројне вредности од 0 до 3.

Мамографска маса (mammographic mass - ma): задатак овог скупа података је да предвиди озбиљност (бенигни или малигни) мамографских лезија на основу BI-RADS атрибута и старости пацијента [Elter *et al.*, 2007]. Сматра се да је данас мамографија најефикаснија метода за скрининг рака дојке која је доступна. Међутим, ниска позитивна предиктивна вредност биопсије дојке на основу интерпретације мамограма доводи до приближно 70 посто непотребних биопсија са бенигним исходом. Да би се смањио висок број непотребних биопсија дојке, предложени су рачунарски програми који треба да помогну докторима у одлуци да ли је неопходно обављање биопсије дојке када постоје сумњиве лезије које се виде на мамографском снимку или је потребно само пратити пацијента.



Слика 6.6: Издвајање контуре на основу сенки мамографске масе [Nakagawa *et al.*, 2004]

Слика 6.6. приказује начин издвајања контура на основу сенки мамографске масе. Овај скуп података може се користити за процену тежине (бенигне или малигне) лезије на основу BI-RADS атрибута и годишта пацијента. На Институту за радиологију Универзитета Erlangen-Nürnberg између 2003 и 2006. године прикупљено је 516 бенигних и 445 малигних маса које су идентификоване путем дигиталних мамограма. Овај скуп података садржи следеће атрибуте: старост пацијента која је изражена у годинама (целобројна вредност); посматрани облик масе који може бити окарактерисан

као округао, овални, лобуларни, неправилни; потом маргина масе која може бити окарактерисана као омеђана, са микро променама, замагљена, лоше дефинисана, сумљива; потом густоћа масе која може бити висока, средња, ниска и са садржајем масти; и озбиљност стања пацијента које може бити бенигно или малигно. У скупу података, свака инстанца је повезана са BI-RADS проценом која се креће у у распону од 1 (дефинитивно бенигни) до 5 (врло сугестивна малигност) која је додељена на основу процене два радиолога. У овом скупу података недостају вредности за поједине атрибуте.

***MONK* проблеми**: ови проблеми припадају класи вештачких (синтетичких) домена, при чему сваки од три проблема користи исту репрезентацију података за упоређење алгоритама машинског учења. *Monk* проблеми су били основни проблеми који су изучавани на првој међународној конференцији посвећеној упоредној анализи различитих алгоритама за учење [Thrun *et al.*, 1991]. Једна значајна карактеристика овог поређења је да је изведена од стране више истраживача, од којих је сваки био заговорник технике коју је тестирао (при чему су истраживачи често били и креатори тих метода). У том смислу, резултати су мање пристрасни у поређењу са резултатима које добија једна особа која уобичајено заговара одређени начин учења, и резултати тачније одражавају проблем генерализације различитим техникама учења. Овај скуп података има 432 инстанце и има 7 атрибута (6 атрибута описује појаву, док је седми атрибут јединствени симбол за сваку инстанцу, и њега не узимамо у разматрање) и нема недостајуће вредности за атрибуте. За сваки проблем, скуп података је подељен на тренинг и тест сет података. Скуп садржи податке за примерке робота који је описан са шест номиналних обележја:

$$\text{Облик главе} \in \{\text{округла, квадратна, осмоугаона}\}$$
$$\text{Облик тела} \in \{\text{округао, квадратни, осмоугаони}\}$$
$$\text{Да ли је насмејан} \in \{\text{да, не}\}$$
$$\text{Шта држи} \in \{\text{мач, балон, заставу}\}$$
$$\text{Боја јакне} \in \{\text{црвена, жута, зелена, плава}\}$$
$$\text{Да ли има кравату} \in \{\text{да, не}\}$$

Постоје три *Monk* проблема и сваки проблем има 432 инстанце.

Monk1 (m1): проблем код овог сета података се може изразити на следећи начин:

$$(\text{облик главе} = \text{облику тела}) \text{ или } (\text{боја јакне} = \text{црвена})$$

Овај проблем је тежак због интеракција између прва два атрибута. Можемо приметити да је једино вредност боје јакне корисна.

Monk2 (m2): проблем код овог сета података се може изразити на следећи начин:

Тачно два атрибута имају прву вредност која је додељена сваком од атрибута.

То значи да тачно два исказа за робота треба да буду тачна у скупу свих датих исказа: {облик главе је округао, облик тела је округао, робот је насмејан, робот држи мач, боја јакне је црвена и робот има кравату}. Проблем је тежак због појава удвојених интеракција атрибута и чињенице да је само једна вредност сваког атрибута значајна. Можемо приметити да је свих шест атрибута релевантно за овај проблем.

Monk3 (m3): проблем код овог сета података се може изразити на следећи начин:

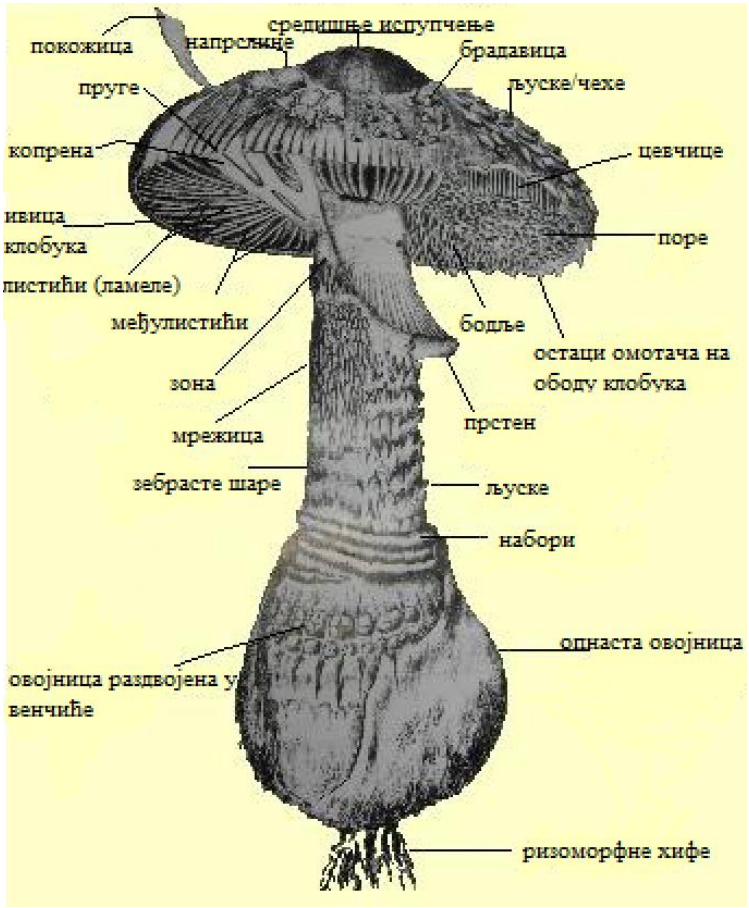
$$(\text{боја јакне} = \text{зелена и шта држи} = \text{мач}) \text{ или}$$
$$(\text{боја јакне} \neq \text{плаве и облик тела} \neq \text{осмоугаони})$$

Стандардни тренинг сет за овај проблем има 5% додатог шума. То је једини *Monk* проблем коме је додат шум. Могуће је постићи око 97% тачности класификације користећи само израз: боја јакне ≠ плаве и облик тела ≠ осмоугаони.

Гљиве (mushroom – mu): овај скуп података укључује описе хипотетичких узорака који одговарају 23 врсти гљива *Agaricus* и *Lepiota* фамилији [Schlimmer, 1987]. Свака врста је идентификована као дефинитивно јестива, дефинитивно отровна или непознатог јестивог састава и не препоручује се за јело. Не постоји једноставно правило за одређивање јестивости гљива на основу њихових карактеристика. На слици 6.7. приказана је грађа гљива, док слика 6.8. приказује узорке гљива. Овај скуп података има 8124 инстанци и 23 атрибута.

Да ли је гљива отровна или не, утврђује се на основу следећих карактеристика: облик клобука (звоно, конус, конвексан, раван, утонуо); површине клобука (влакнаста, са каналима, покривена љуспама, глатка); боје клобука (смеђа, боја коже, боја цимета, сива, зелена, ружичаста, љубичаста, црвена, бела, жута); трусишта, тј. дела који се налази на доњој страни клобука (на листићима, у унутрашњости цевчица, на унутрашњој површини, у унутрашњости плодног тела); мириса (бадема, аниса, смрада, устајали мирис, без мириса, опор мирис); стручка (место где је стручак причвршћен за

клобук, да ли је срастао са другим деловима, висине и дебљине, спољног облика, облика доњег дела стручка, површине стручка); боје спора (црна, смеђа, боја коже, чоколада, зелена, наранчаста, љубичаста, бела, жута); како су насељене гљиве (у изобиљу, груписано, бројно, разасуте, свега неколико, осамљено); и станишта (трава, лишће, ливаде, стазе, урбано, на отпаду, шуме). Све вредности у овом скупу података су категоричке вредности. Код неких атрибута постоје недостајуће вредности.



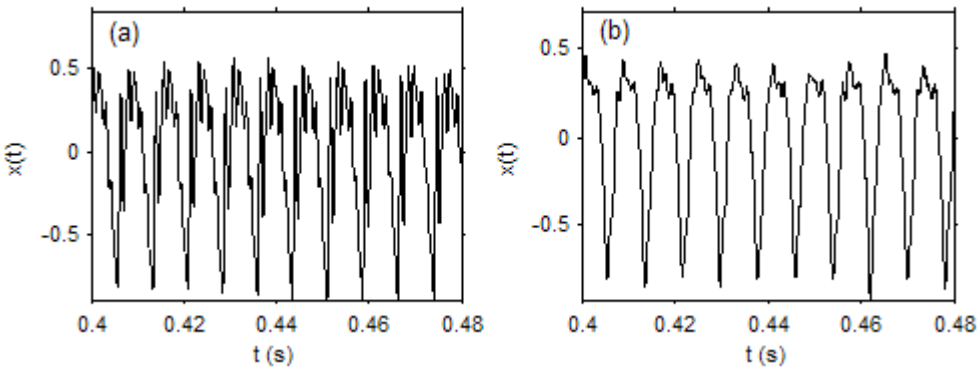
Слика 6.7: Грађа гљива [http://www.pcelica.co.rs/gljive/gradja/gradja-gljive.php]

Слика 6.8: Гљиве

Паркинсон (Parkinson – ра): овај скуп података је креирао Max Little са Универзитета у Оксфорду, у сарадњи са Националним центром за глас и говор, који је

смештен у Денверу, Колорадо. Оригинална студија објављена је у сврху екстракције атрибута из говорних сигнала код особа које имају говорни поремећај. Слика 6.9. приказује два примера говорног сигнала: (а) здраве особе, (б) особе оболеле од Паркинса [Little *et al.*, 2009]. Овај сет података се састоји од низа биомедицинских мерења гласа код 31 особе, од тога 23 оболеле од Паркинсонове болести [Little *et al.*, 2007]. У овом сету података постоји 195 инстанци и 23 атрибута. Атрибути овог сета података су: просечна основна говорна фреквенција, највећа основна говорна фреквенција, минимална основна говорна фреквенција, подрхтавање, апсолутно подрхтавање, неколико мера варијације у фундаменталној фреквенцији, неколико мера варијације у амплитуди, две мере односа буке на тонским компонентама у гласу, две нелинеарне динамичке мере, сигнал фрактала и три нелинеарне мере основне варијације фреквенције.

Свака колона у табели је особена карактеристика гласа особе, а сваки ред одговара једном од 195 снимака гласа одређене особе. Главни циљ овог скупа података је одвајање здравих људи од оних особа које су оболеле од Паркинса, на основу колоне „Статус“ која има могуће вредности 0 за здраве особе и 1 за особе са Паркинсоновом болешћу.



Слика 6.9: Два примера говорног сигнала: (а) здраве особе, (б) особе оболеле од Паркинса [Little *et al.*, 2009]. Хоризонтална оса представља време у секундама, на вертикалној оси је приказана амплитуда сигнала (без јединичне мере)

Дијабетес (Pima Indijans dijabetes – pi): ради дијагностификовања дијабетеса из већег скупа података издвојени су подаци за жене које су старије од 21 годину и припадају Пима Индијанцима [Smith *et al.*, 1988]. У овом сету података дијагностификовано је да ли пацијент показује знакове дијабетеса према критеријима Светске здравствене организације (тј. ако се открије током рутинске медицинске контроле или ако 2 сата након оптерећења глукоза је барем 200 мг/дл у сваком

испитивању). Становништво које је учествовало у овом истраживању живи у непосредној близини Phoenix-а, Аризона у САД-у (слика 6.10).

Слика 6.10: Аризона Пима Индијанци
[http://indiancountrytodaymedianetwork.com/article/mexico-vs.-arizona-pima-indians-3258]

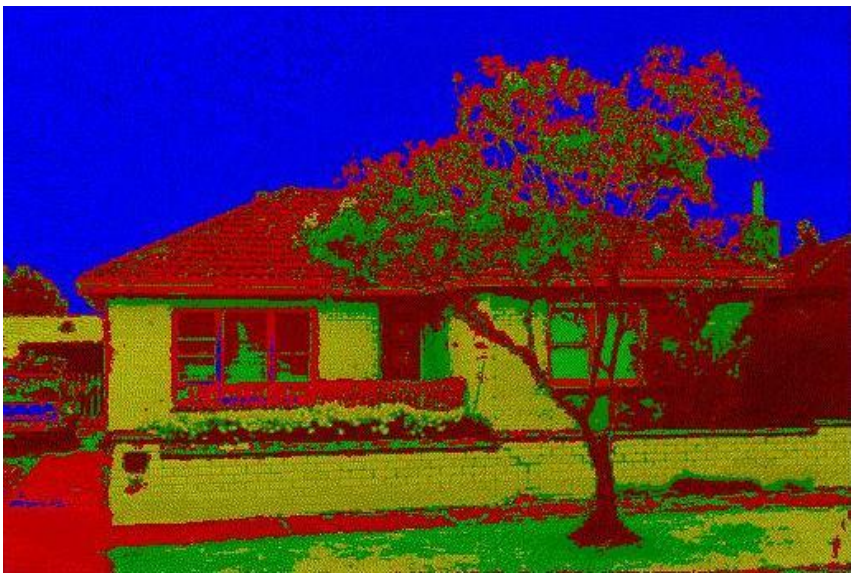
У овом скупу података постоји 768 инстанци и 8 атрибута који имају нумеричке вредности. Атрибути у овом скупу података су: број трудноћа, концентрација глукозе на таште и после 2 сата у оралном тесту оптерећења глукозом, дијастолни крвни притисак, дебљина кожног набора над трицепсом, да ли се након 2 сата вредност инсулина вратила на почетну вредност, индекс телесне масе, постојање дијабетеса у породици и старост. Скуп података садржи податке о 500 особа које нису дијабетичари и 268 које су дијабетичари. Код неких инстанци, за неке атрибуте постоје вредности које недостају.

Сегментација слике (image segmentation – se): случајеви су извучени случајним избором из базе података 7 слика спољног окружења [Piater *et al*., 1999]. Слике су ручно сегментиране како би се извршила класификација за сваки пиксел. Свака инстанца у скупу података је 3x3 регија. У овом скупу података има 210 података за тренинг и 2100 тест података. Скуп података садржи 19 нумеричких атрибута, без недостајућих вредности за атрибуте. Класа овог скупа података има могуће вредности: површина цигле, небо, лишће, цемент, прозор, пут и трава. Скуп података има 30 случајева по класи за обуку података и 300 случајева по класи за тестирање података.

Основни саставни елементи свих дигиталних слика су пиксели. Пиксели су мали суседни квадрати у матрици преко дужине и ширине дигиталне слике. Сви пиксели у било којој дигиталној слици су исте величине. Пиксели су једнобојни, при

чему је сваки пиксел једна боја која је уклопљена из неке комбинације три примарне боје црвена, зелена и плава. Сегментација слике је процес класификовања пиксела слике у различите класе према неким унапред дефинисаним критеријима. Стварање класификатора високих перформанси обично укључује значајну количину људског напора будући да се тренинг обавља преко *off-line* руком означених пиксела као тренинг примера. Корисност тренирајућег сета је тешко одредити *a priori*, па велике количине података обично морају бити на располагању.

Слика 6.11: Необрађена слика [http://vis-www.cs.umass.edu/old/projects/itl/example.html]



Слика 6.12: Обрађена слика након пиксел класификације [http://vis-www.cs.umass.edu/old/projects/itl/example.html]

Предложена метода пиксел класификације [Piater, 1999] омогућује кориснику селекцију пиксела на слици тако што ће се рећи програму која је врста површине тај

пиксел и то са кликом на дугме. Систем тада рекласификује слику и приказује њену класификацију кориснику. Ако је корисник задовољан са урађеним, посао је обављен, а ако није задовољан класификацијом, корисник може кликнути на подручје слике које је погрешно класификовано и систем ће рекласификовати слику према клику. Када је корисник задовољан са резултатом, онда се класификација може користити и на другим сликама. У наставку текста дат је пример класификације, тако што се обрада претходне слике ради тако да се пиксели разврстају у 4 категорије и то: црвени у кров, плави у небо, зелени у траву и жути у циглу. Необрађена слика приказана је на слици 6.11, док је обрађена слика приказана је на слици 6.12.

Соја (soybean – so): задатак је дијагностификовати болести у биљкама соје [Michalski и Chilausky, 1980]. У овом скупу података постоји 307 примерака описаних са 35 категоричких атрибута. Вредност атрибута је мерена посматрањем својстава лишћа и различитих биљних абнормалности. У сету података постоји 19 класа за болести соје. Неке од болести соје су: пламењача и бактериозна пегавост које су најчешће обољење листа, на стаблу су најштетнији рак стабла и бела трулеж, на корену угљенаста трулеж, док семе најчешће оболева од трулежи. Различите болести соје приказане су на слици 6.13. Овај скуп података за неке атрибуте има непознате вредности.

Слика 6.13: Различите болести соје
[http://www.agweb.com/article/Prevent_soybean_diseases_with_Headline_207165/]

Срце (Statlog heart - sh): задатак је предвидети одсутности или присутности болести срца на основу старости, пола, одговарајућег типа бола у грудима, крвног притиска у мировању, нивоа холестерола и шећера у крви, електрокардиографских резултата, највећег броја откуцаја срца, промене показатеља приликом напора и слично. Срчане болести су један од најчешћих разлога смртности у свету, а последица су нездравог начина живота, пре свега неумерености у јелу и пићу, слабој телесној активности и високом нивоу стреса (слика 6.14). Овај сет података садржи 13 атрибута

(који су извађени из већег скупа од 75 атрибута). Класа за овај сет података има две вредности: одсуство и присуство болести срца. Постоји 270 посматрања, без вредности које недостају за поједине атрибуте.

Слика 6.14: Срце [http://archive.ics.uci.edu/ml/datasets/Statlog+(Heart)]

Гласање конгресмена (congressional voting records – vo): у овом сету података страначку припадност америчког Представничког дома карактерише како су конгресмени гласали на 16 кључних питања као што су трошење на образовање и имиграција [Schlimmer, 1987]. У Америци Представнички дом је један од два дома Конгреса САД-а; други је Сенат. У Дому свака држава је пропорционално представљена према уделу у укупном становништву и има право на најмање једног представника; најмногољуднија држава тренутно има 53 представника. У Дому укупан број заступника је 435 према посебном закону из 1911. године, иако Конгрес може законом изменити тај број. Сваки представник овог Дома служи мандат од две године. Слика 6.15. приказује гласање конгресмена САД-а.

Слика 6.15: Гласање конгресмена
[http://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records]

Атрибути које овај скуп података садржи односи се на трошење средстава за: хендикепирану децу, пројекат поделе трошкова за воду, замрзавање лекарских такси, помоћ верским групама у школи, развој сателита и ракета, помоћ имиграцији, смањење пореза корпорацијама, образовање, судство, казнено-поправне домове, као и бесцаринski извоз. Класе у овом скупу података имају две вредности: демократе и републиканци. У овом скупу података постоји 435 инстанци (267 демократи, 168 републиканци), сви атрибути су бинарни и постоје недостајуће вредности.

Табела 6.1. Приказ сетова података. Подразумевана тачност класификација је тачност предвиђања већинске класе на целом скупу података. Сви скупови података су реални, осим *m1*, *m2* и *m3* који су вештачки. ,,CV“ означава 10-струку унакрсну валидацију

Скуп	**Атрибути**			**Број класа**	**Величина за тренирање**	**Величина за тестирање**	**Референтна тачност**
	укупно	категорички	нумерички				
bc	9	9	0	2	286	CV	70.30
ca	15	9	6	2	690	CV	55.50
cg	20	13	7	2	1000	CV	50.10
ct	23	0	23	3	2126	CV	95.00
he	19	13	6	2	155	CV	78.10
li	6	0	6	2	345	CV	58.10
lc	56	0	56	3	32	CV	26.80
ma	5	0	5	2	961	CV	84.00
m1	6	6	0	2	124	308	50.00
m2	6	6	0	2	169	263	67.13
m3	6	6	0	2	122	310	52.78
mu	22	22	0	2	8124	CV	51.80
pa	23	0	23	2	195	CV	76.00
pi	8	0	8	2	768	CV	65.10
se	19	0	19	7	2310	CV	14.30
so	35	35	0	19	683	CV	13.47
sh	13	3	10	2	270	CV	55.00
vo	16	16	0	2	435	CV	61.40

У табели 6.1. приказане су упоредне карактеристике посматраних сетова података. Постоји 18 скупова података, од тога 15 скупова података су реални скупови, што значи да су добијени прикупљањем података из реалних система који постоје. Остала три скупа података *m1*, *m2* и *m3* су вештачки скупови података, што значи да подаци нису добијени из реалног система, већ су податке креирали истраживачи за потребе истраживања. Да би добили референтне податке током истраживања у раду

смо користили и реалне и вештачке скупове података за доказивање постављених хипотеза.

При томе, посматран је укупан број атрибута у сваком сету података, као и број оних атрибута који припадају категорији категоричких или нумеричких атрибута. Код оних алгоритама и метода, чији улаз података не подржава категоричке или нумеричке атрибуте, вршена је одговарајућа припрема података, пре саме обраде података. Пет сетова података има више атрибута од 20, и то *lc* са 56, *so* са 35, *pa* и *ct* са 23 и *mu* са 22. Најмање атрибута имају сетови података *ma* са 5, *li*, *m1*, *m2* и *m3* са 6 атрибута. Можемо закључити да се у посматраним скуповима података налазе и скупови са изузетно великим бројем атрибута, као и они скупови који имају мали број атрибута, што је добро са становишта истраживања. Посматрани скупови података су балансирани јер постоје скупови који садрже само или категоричке или нумеричке атрибуте, као и скупови података који садрже и категоричке и нумеричке податке.

Што се тиче броја класа у посматраним скуповима података, само два скупа података имају већи број класа од 3, и то *se* који има 7 класа и *so* који има 19 класа. Разлог за ово је чињеница, што се у највећем броју случаја у проблемима класификације разврставање постојећих инстанци врши у две, евентуално три класе, а ређе у већи број класа.

У табели 6.1. видимо да број инстанци предвиђен за тренирање варира од малог броја прикупљених истанци што је случај са *lc* који има само 32 истанце до скупова који имају много већи број истанци као што је нпр. случај са *mu* који има 8124 истанци за тренинг. Што се тиче величине скупа за тестирање, иницијално код свих реалних скупова података, имали смо припремљен један скуп података, из кога смо методом 10-струке унакрсне валидације издвајали податке који ће служити за тестирање. Истраживачи који су креирали вештачке скупове података *m1*, *m2* и *m3* су одвојили податке у две групе и то оне који ће служити за тренирање и оне који ће служити за тестирање, при чему је мањи број података коришћен за тренинг (у просеку око 25%), а већи део служи за тестирање тачности класификације. У последњој колони табеле приказана је референтна тачност за реалне и вештачке скупове података.

СЕДМИ ДЕО

7. РЕЗУЛТАТИ УЧЕЊА И ЕСТИМАЦИЈА ПЕРФОРМАНСИ НАУЧЕНОГ ЗНАЊА

У седмом делу рада, биће речи о методологији извођења експеримента и подешавању параметара модела. Биће разматрана тачност и прецизност којима меримо успешност добијеног модела, као и статистички тестови које користимо у истраживањима, са посебним освртом на стандардну девијацију и *t*-тест.

7.1. Опис методологије извођења експеримента

Експеримент је рађен уз помоћ WEKA (Waikato Environment for Knowledge Analysis), алата за припрему и истраживање података развијен на Waikato Универзитету на Новом Зеланду. Овај алат поседује подршку за цео процес истраживања почевши од припреме података преко процене и коришћења различитих алгоритама.

WEKA је написана у Јави и дистрибуира се под GNU General Public Licence. Овај алат ради на скоро свим платформама и теситран је на Linux, Windows и Macintosh оперативним системима. Верзија која је коришћена је последња стабилна верзија WEKA-е 3.6 и може се преузети са *web* адресе http://www.cs.waikato.ac.nz/ml/weka/index.html. У овом алату имплементиране су најчешће методе које се јављају у истраживању података, а то су: класификација, регресија, кластеровање, асоцијација и избор атрибута. Већина имплементираних метода омогућава подешавање параметара према конкретном проблему и подацима. WEKA поседује чак 49 метода за припрему података. Ови подаци могу бити учитани у неколико различитих формата датотека. WEKA подржава разне формате датотека, али је препручљиво користити ARFF формат датотеке који је основни подржани формат. У нашем истраживању, коришћен је ARFF формат, а подаци који нису изворно у том формату, конвертовани су накнадно у ARFF формат. Овај алат подржава и остале формате, а то су: CVS, C4.5 или бинарни. Алат омогућава да се подаци такође преузму са URL адресе или из SQL базе података.

WEKA има три различита графичка корисничка окружења: *Explorer*, *Knowledge Flow* и *Experimenter*. Од ових окружења, најлакши начин за коришћење WEKA-е је *Explorer*. *Explorer* омогућава лаку и ефикасну примену свих функционалности алата путем изборних менија. Корисничко окружење *Knowledge Flow*, омогућава кориснику да самостално дефинише секвенцијалну обраду података. Главни недостатак *Explorer*-а је чињеница да све податке чува у главној меморији – по отварању датотеке или базе података цео садржај се учитава одмах, што доводи до тога да је примена могућа само на средњим и малим базама података. Ово окружење омогућава прецизно дефинисање обраде података повезивањем компоненти које представљају изворе података, методе за припрему, алгоритме, методе еваулације и графички приказ. Ако алгоритми имају могућности инкременталног учења, подаци ће бити учитани и обрађивани секвенцијално. Последње окружење, *Experimenter*, осмишљено је да помогне кориснику да одговори на основно питање при употреби класификације и регресије: које методе и параметре користити за дати проблем. *Experimenter* омогућава поређење различитих класификатора и филтера са различитим параметрима. Поређење може да се реализује и кроз *Explorer* али *Experimenter* нуди аутоматизацију целог процеса, тестове исправности и перформанси система. Такође, иза свих ових графичких окружења налази се основна функционалност WEKA-е, којима се може приступити и из командне линије. За проблеме које смо ми изучавали, проблем редукције димензионалности података, коришћена су корисничка окружења *Explorer* и *Experimenter*.

Главне предности овог алата су широк спектар метода за припрему података, избор атрибута и алгоритама интегрисаних у једном алату. Такође, било која од метода која је имплементирана у алату WEKA може бити позивана из корисничког кода, што за последицу има олакшан развој нових апликација за истаживање података уз минимум додатног кодирања. Овај алат је потпуно бесплатан, једноставно и лако се инсталира на свакој платформи, а GUI га чини једноставним за коришћење.

Недостатак WEKA-е је документација, јер WEKA константно расте и документација даје само листу расположивих алгоритама. Посебно је документација за графичко корисничко окружење ограничена. Скалабилност је други могући проблем при раду са WEKA-ом, јер се током рада са великим количинама података време за обраду драстично повећава. Такође, недостатак је и чињеница да у GUI окружењу нису

имплементиране све функционалности WEKA-е па је у раду неке опције потребно позивати из командне линије.

Приликом тражења модела који најбоље апроксимира циљну функцију, потребно је дати и мере квалитета модела, односно учења. Различите мере се могу користити у зависности од врсте проблема, али у случају проблема класификације се обично користи прецизност, односно број тачно класификованих инстанци подељен укупним бројем инстанци. У нашим експерименталним истраживањима користили смо тачност класификације као меру квалитета модела.

Да би добили поузданији начин евалуације наученог знања користили смо тзв. унакрсну валидацију, где смо цео скуп података којим смо располагали делили на *n* приближно једнаких подскупова. При томе смо један подскуп издвајали и тренинг вршили на осталих *n-1* подскупова, а након тренинга, квалитет наученог знања оцењивали на издвојеном подскупу. Описани поступак смо понављали за све остале издвојене подскупове и као финалну оцену квалитета узимали просек добијених оцена за сваки од подскупова. У нашем експерименталном истраживању смо за вредност *n* узимали број 10. Унакрсну валидацију смо користили у нашем експерименталном истраживању, јер описани поступак даје стабилнију оцену квалитета, а предност овог метода је и да се у сваком од *n* корака унакрсне валидације користи велика количина података при тренирању, а све расположиве инстанце у једном тренутку су искоришћене за тестирање.

За класификацију, за све реалне скупове података, коришћена је 10-струка унакрсна валидација, која је при томе била увек поновљена 10 пута. За вештачке скупове података *m1*, *m2* и *m3*, с обзиром да смо иницијално имали одвојене тест и тренинг податке, урадили смо спајање података тест и тренинг скупа, водећи рачуна да при тренирању користимо првих 22.3% података за *m1* проблем, за *m2* проблем првих 28.1% података, за *m3* проблем првих 22.0% података, како би организовали експеримент како је он оригинално замишљен. На овај начин добијамо упоредивост резултата са осталим истраживачима који су користили ове сетове података. Значи у новом скупу података који смо припремили за експеримент, на почетку скупа смештамо тренинг податке, па онда у наставку скупа тест податке. Податке који су иницијално у *arff* формату смо пребацили у *csv* формат, извршили спајање и поново вратили у *arff* формат. Током експерименталних истраживања најпре смо користили податке из ново добијеног сета за тренинг (првих 22.3% података за *m1* проблем, за *m2*

проблем првих 28.1% података, за *m3* проблем првих 22.0% података), па онда за тестирање, водећи рачуна да не радимо случајни избор података за тренинг и тестирање. Цео експеримент смо поновили 10 пута.

За неке алгоритме учења је неопходно да све вредности постоје за све атрибуте у свим инстанцама. У нашем случају, за SVM алгоритам је било неопходно да постоје све вредности свих атрибута. С обзиром да су постојали сетови података са недостајућим вредностима, да би могли да користимо SVM алгоритам, било је неопходно заменити недостајуће вредности са процењеним вредностима за дати скуп. Ову замену смо радили код свих сетова података који имају недостајуће вредности, пре коришћења SVM алгоритма. Остали алгоритми учења су могли да се сами изборе са недостајућим вредностима за поједине атрибуте у неким од истанци.

У експериманалном истраживању користили смо филтер методе, методе претходног учења и екстракцију атрибута ради смањења димензионалности података. Експерименталним истраживањем нису обухваћене уграђене методе, јер ове методе врше селекцију атрибута у склопу основног алгоритма индуктивног учења, односно као део процеса генерализације. Због тога није могуће вршити упоредне анализе ефеката редукције димензионалности података код ових метода. За разлику од ових метода, методе филтрирања, претходног учења и екстракције атрибута разматрају селекцију атрибута као спољашњи слој процеса индукције.

У многим случајевима тренинг скупови су оскудни и постоје међусобне интеракције атрибута. Избор оптималног подскупа атрибута врши се различитим естимацијама, које се заснивају на различитим статистичким претпоставкама, нпр. независност атрибута и довољан број тренинг примера, које нису увек задовољене. То је разлог због чега уграђене методе селекције атрибута, нису увек довољне, па се у многим практичним ситуацијама користе методе претходне селекције атрибута како би се перформансе побољшале.

За сваку методу која је коришћена у сврху смањења димензионалности података коришћен је скуп могућих решења, који је потом био пропуштен кроз класификаторе IBk, *Naïve Bayes*, SVM, J48 и RBF мреже. У свим експериментима, изабрано је оно решење за број атрибута који ће се даље користити у истраживању, који даје највећу тачност класификације.

У резулататима истраживања, када упоређујемо пар алгоритама, представићемо резултате тачности класификације за сваки алгоритам на сваком скупу података.

Важно је напоменути да кад смо користили 10-струку унакрсну валидацију за оцену тачности и прецизности, да је унакрсна валидација независна спољна петља, не иста као и унутрашња 5-струка унакрсна валидација која је део алгоритма за избор атрибута код методе претходног учења.

Користили смо 5-струку унакрсну валидацију која је део алгоритма за избор атрибута код методе претходног учења, јер се на тај начин избегава вишеструка унакрсна валидација за велике скупове података, како би избегли велико захтевано време за обраду података. Са друге стране нисмо користили мање вредности за унакрсну валидацију од 5, јер се у пракси показало да је за мале скупове података потребно урадити више пута унакрсну валидацију како би се превазишао проблем високе варијансе који је резултат мале количине података за обраду.

Наши резултати дају тачност и прецизност која је добијена као средња вредност десет понављања и сваки пут уз 10-струку унакрсну валидацију. Такође, приказујемо и стандардну девијацију. Да би смо утврдили да ли је разлика између два алгоритма значајна или не, ми приказујемо вредности *t*-теста, које указују на вероватноћу да је један алгоритам бољи од других. У наставку текста, укратко објаснићемо зашто смо користили у експерименталним резултатима и тачност и стандардну девијацију.

Слика 7.1: Тачност и прецизност, на основу [Taylor, 1999]

На слици 7.1. приказани су појмови тачности и прецизности. Тачност показује блискост резултата мерења са стварном вредношћу, а прецизност указује на поновљивост, односно репродуктивност мерења. Прецизност мереног система, која се назива и поновљивост, показује степен у којем поновно мерење под непромењеним условима даје исте резултате. Систем за мерење може бити тачан, али није прецизан, прецизан, али није тачан, нити тачан нити прецизан, или обоје и тачан и прецизан. Мерни систем је добро дизајниран ако је и тачан и прецизан. Упоредићемо појмове тачности и прецизности на примеру мете.

Слика 7.2: Висока тачност, али ниска прецизност, на основу BIMP и ISO 5725

Слика 7.3: Висока прецизност, али ниска тачност, на основу BIMP и ISO 5725

На слици 7.2. је приказана висока тачност, али ниска прецизност, док је на слици 7.3. приказана висока прецизност, али ниска тачност. Аналогија која се овде користи је у циљу да се објасни разлика између тачности и прецизности. У тој аналогији, поновљена мерења су упоређена са стрелицама које погађају мету. Тачност описује блискост стрелице ка циљном центру. Стрелице које погађају ближе центру сматра се да погађају тачније.

За наставак аналогије, ако је велики број стрелица испуцан, прецизност би била величина кластера стрелице. Када су све стрелице груписане заједно, кластер се сматра прецизним јер су све стрелице погодиле у близини истог места, чак и ако нужно не погађају у близини самог центра. Мерења су тада прецизна, иако нису нужно тачна. Идеални мерни уређај је и тачан и прецизан, са мерењима која су сва близу око познате вредности.

То вреди и када се мерења понављају и добију просечне вредности. У том случају, термин стандардна грешка се исправно примењује: прецизност просека је једнака познатој стандардној девијацији процеса подељена кореном броја мерења просека. То значи, да наша мерена стандардна девијација подељена са кореном из 10, што представља број мерења просека, одговара прецизности. Такође, централна гранична теорема показује да је расподела вероватноћа просечних мерења ближе нормалној расподели него код појединачних мерења.

У нашем експерименталном истраживању, кад год смо упоредили два или више алгоритама, у раду дајемо табелу тачности класификације, и приказујемо две врсте графова са стубићима. Један граф са стубићима приказује апсолутну разлику у

тачности класификације и други граф са стубићима приказује апсолутну разлику у стандардној девијацији за тачност класификације. Упоређивање ће углавном бити такво да други алгоритам је алгоритам код кога је урађена председелекција атрибута, а први алгоритам је стандардни алгоритам без председелекције атрибута. Кад је вредност стубића већа од нуле, други алгоритам са председелекцијом атрибута надмашује својом вредношћу први алгоритам који је стандардни алгоритам.

Када смо приказивали резултате за потребно време за тренинг података, они су изражавани у јединицама CPU секунди. Експеримент је рађен на AMD Phenom (tm) 9650 Quard-Core Pocessor 2.31 GHz са 4GB RAM-а. Такође, код упоређивања алгоритама, дајемо табелу потребног времена за тренинг и приказујемо две врсте графова са стубићима. Један граф са стубићима показује апсолутну разлику у потребном времену за тренинг и други граф са стубићима показује апсолутну разлику у стандардној девијацији за потребно време тренинга.

7.2. Статистички тестови (тестови значајности)

У експерименталном истраживању код извођења статистичких тестова постоје одређени кораци којих се треба придржавати да би закључак био поуздан, а то су: постављање нулте хипотезе, бирање нивоа поузданости, одређивање величине узорка, бирање статистичког теста за тестирање хипотезе, утврђивање критичне вредности за одабрани статистички тест, прикупљање података, израчунавање статистичке величине за одабрани статистички тест, доношење статистичког закључка и изражавање статистичког закључка.

У наставку текста биће речи о мерама централне тенденције, мерама варијабилности и тестовима хипотеза које смо у раду користили.

Централна тенденција је тежња ка окупљању података скупа око једне централне вредности, која је општа и репрезентативна за целу дистрибуцију. Њихова улога је да, занемарујући индивидуалне разлике између података скупа, истакну ону величину која је за све њих карактеристична и која може да служи као средство за упоређивање разних серија. У мере централне тенденције спада: аритметичка средина, медијана, мода, геометријска средина и хармонијска средина. Ми ћемо у нашим истраживањима користити аритметичку средину.

Аритметичка средина је средња вредност (процена параметра μ), чија се вредност добија дељењем суме експериментално добијених вредности са бројем мерења, што је дато у следећем изразу:

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{7.1}$$

Као мере варијабилности, које дају информацију о различитим одступањима у статистичком скупу, може да се користи интервал варијације (распон), стандардна девијација, варијанса и коефицијент варијансе.

Интервал варијације је размак од најмање до највеће вредности обележја посматрања. Представља најнетачнију меру груписања резултата око неке средње вредности.

$$R = x_n - x_1, \;\; x_1 < x_2 < \cdots x_n \tag{7.2}$$

где је *x* мерена величина.

Стандардна девијација је мера одступања вредности обележја од аритметичке средине, и дата је следећим изразом:

$$s = \sqrt{\frac{\sum(\bar{x}-x_i)^2}{n-1}} \tag{7.3}$$

Варијанса је просечно квадратно одступање од аритметичке средине, дато изразом:

$$s^2 = \frac{\sum(\bar{x}-x_i)^2}{n-1} \tag{7.4}$$

У нашим истраживањима смо користили стандардну девијацију као меру варијабилности, односно информацију о различитим одступањима у статистичком скупу.

За тестирање хипотезе користе се параметријски и непараметријски тестови. Параметријске методе користе се за упоређивање две или више група података и заснивају се на претпоставци да су подаци нормално расподељени. Ове методе се увек заснивају на теорији вероватноће и увек се у њима појављује потреба за оцењивањем појединих параметара (средње вредности, стандардне девијације или варијансе). Међутим, када не може са сигурношћу да се утврди да ли је расподела једне групе података нормална, израчунавање појединих параметара и примена параметријских метода дају врло непоуздане закључке. У тим случајевима се примењују

непараметријске методе, које се заснивају на претпоставци да постоји било која вероватноћа расподеле.

За елиминисање „спољних“ резултата, вредности које се издвајају у односу на остале, може се користити *Dixon*-ов тест (Q-тест) – за мале узорке, или *Grubbs*-ов тест (G-тест). *F*-тест служи да утврди да ли је разлика између варијанси два узорка значајна.

Ми ћемо у нашим истраживањима користити *t*-тест који се користи за утврђивање постојања систематских грешака. Користи се у следећим случајевима: (1) када се упоређује средња вредност групе података са правом вредношћу (одређивање тачности), (2) када се упоређују средње вредности две групе података, (3) код паралелних одређивања.

Код упоређивања експериментално одређене средње вредности са правом вредношћу, параметар *t* се израчунава према следећој једначини:

$$t = \frac{(\bar{x}-\mu)\times\sqrt{N}}{s} \tag{7.5}$$

$\bar{x}$ – аритметичка средина мерених вредности, μ - права вредност, N – број мерења, s - стандардна девијација.

Добијена вредност се упоређује са критичном *t*-вредношћу, која се за дати ниво поузданости и број степени слободе, очитава у табели. Ако вредност t прелази одређену критичну вредност нулта хипотеза се одбацује. У супротном не постоје докази за постојање систематске грешке (ово не значи да систематска грешка не постоји већ само да она није изражена).

У нашем експерименталном истраживању користили смо упоредни *t*-тест (енг. *Paired T-Test*), где је ниво значајности постављен на вредност 0.05. Ако имамо симултано одређивање тачности класификације у различитим сетовима података помоћу две методе, за утврђивање да ли се добијена вредност различитим методама значајно разликује користимо упоредни *t*-тест. Упоредним *t*-тестом се тестира значајност средње вредности разлике парова *d* према следећој једначини:

$$t = \frac{\bar{d}\sqrt{N}}{s_d} \tag{7.6}$$

где је s_d – стандардна девијација добијених разлика. Уколико је израчуната вредност параметра t већа од табличне (критичне вредности), нулта хипотеза се одбацује и каже се да се *d* значајно разликује од нуле, односно да је разлика у паровима статистички значајна.

У табелама које следе за тачност класификације различитих класификатора и у табелама за време потребно за тренинг података су приказане ознаке „+“ и „-“, које означавају да је одређени резултат статистички бољи (+) или лошији (-) од основног класификатора на нивоу значајности који је специфициран на вредност од 0,05.

У табелама за тачност класификације различитих класификатора ознака „+“ означава значајно већу вредност за тачност класификације, док „-“ означава значајно мању вредност за тачност класификације.

У табелама које садрже податке о времену потребном за тренинг података ознака „+“ означава значајно мању вредност за потребно време, што значи да се ради о статистички бољем резултату док „-“ означава значајно већу вредност за потребно време што значи да се ради о статистички лошијем резултату. С обзиром да време потребно за тренинг података може да се мења, ако применимо различите методе за редукцију димензионалности података, добро је да током експеримента можемо да добијемо мање вредности за потребно време тренирања, јер онда наш алгоритам ради брже, што је посебно значајно ако имамо проблем у реалном времену. Значи да су мање вредности у табелама за време боље, због чега се и смисао статистичких показатеља „+“ и „-“ мења у односу на тачност класификације.

ОСМИ ДЕО

8. ЕСТИМАЦИЈА ТАЧНОСТИ КЛАСИФИКАЦИЈЕ ЗА МЕТОДЕ ФИЛТРИРАЊА

У осмом делу дисертације, након разматрања поставки експерименталног истаживања, биће приказани резултати истраживања за различите методе филтрирања и то за сваки класификациони алгоритам посебно.

8.1. Поставке експерименталног истраживања

Методе филтрирања функционишу независно о изабраном алгоритму вештачког учења, за разлику од методе селекције претходним учењем. Вредност атрибута се хеуристички процењује анализом општих карактеристика података из скупа за учење. Ове методе користе више различитих техника избора атрибута, јер постоји више начина хеуристичког вредновања атрибута. Методе филтрирања се деле у две основне групе, зависно о томе вреднује ли коришћена хеуристика подскупове атрибута или појединачне атрибуте.

У овом раду, користимо следеће методе филтрирања за рангирање атрибута које су статистички и ентропијски засноване, а показују добре перформансе у различитим доменима: IG, GR, SU, RF, OR и CS.

За редукцију димензионалности података код класификационих проблема вештачке интелигенције, у овом раду смо код метода филтрирања за потребе рангирања атрибута за све скупове података, користили потпуни скуп података за тренирање, уместо 10-струке унакрсне валидације. Међутим, након редукције величине посматраног скупа података, за потребе класификације, користили смо 10-струку унакрсну валидацију.

Све методе филтрирања, IG, GR, SU, RF, OR и CS су одрадиле рангирање атрибута за сваки појединачни скуп података. С обзиром да методе рангирања приказују све атрибуте по оном редоследу какав је њихов значај за класификациони проблем, ове методе не врше аутоматски редукцију броја атрибута.

Да би се уз помоћ ових метода извршила редукција броја атрибута, постоје две могућности: (1) коришћење прага, или (2) коришћење одговарајућег броја атрибута за сваки сет података и сваку од метода филтрирања. У случају прве могућности, коришћења прага, за сваки сет података и за сваку од метода филтрирања, претражује се скуп свих могућих решења тако што се узима почетна вредност прага и она се у свакој даљој итерацији инкрементира за одређену вредност инкремента, и пропушта се кроз сваки од класификатора, док се недостигне унапред задата вредност прага. Почетна вредност прага нпр. може бити 0.00, а потом се са што мањом вредношћу за инкремент, достиже унапред задата горња граница прага, нпр. 0.05. Током овог инкрементирања за вредност прага, мери се тачност класификације за сваки класификатор, а најбоље решење за избор броја атибута је оно решење које даје највећу тачност класификације за изабрани праг.

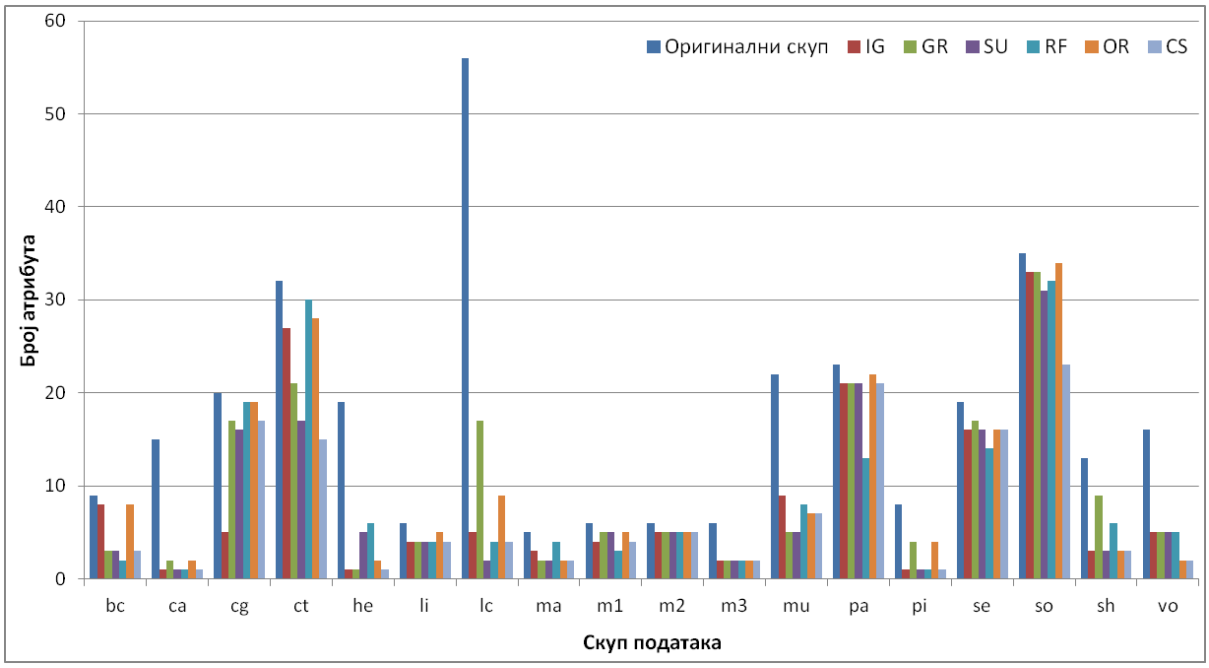
Табела 8.1. Број атрибута у оригиналном скупу података и број атрибута селектован уз помоћ метода филтрирања. Претраживањем скупа свих могућих решења за сваку методу је пронађен оптимални број атрибута.

Скуп	Ориг. скуп	IG	GR	SU	RF	OR	CS
bc	9	8	3	3	2	8	3
ca	15	1	2	1	1	2	1
cg	20	5	17	16	19	19	17
ct	23	18	12	8	21	19	6
he	19	1	1	5	6	2	1
li	6	4	4	4	4	5	4
lc	56	5	17	2	4	9	4
ma	5	3	2	2	4	2	2
m1	6	4	5	5	3	5	4
m2	6	5	5	5	5	5	5
m3	6	2	2	2	2	2	2
mu	22	9	5	5	8	7	7
pa	23	21	21	21	13	22	21
pi	8	1	4	1	1	4	1
se	19	16	17	16	14	16	16
so	35	33	33	31	32	34	23
sh	13	3	9	3	6	3	3
vo	16	5	5	5	5	2	2

Друга могућност за избор броја атрибута одређеног скупа података је претраживање скупа свих могућих решења за број атрибута који ће се користити у

класификатору. Поступак је сличан предходном. У овом случају, претражује се скуп свих могућих решења тако што се узимају све могуће вредности за број атрибута одговарајућег скупа података за сваку од метода филтрирања и пропушта се свако појединачно решење кроз сваки од класификатора. Током овог поступка, мери се тачност класификације за сваки класификатор, а најбоље решење за изабрани број атибута је оно које даје највећу тачност класификације.

У овом експерименталном истраживању коришћена је друга могућност, односно селектовање броја атрибута који ће се користити у класификатору, како би добили што већу тачност класификације за дати скуп података и посматрану методу филтрирања.



Слика 8.1: Број атрибута у оригиналном скупу података и оптималан број атрибута добијен методама филтрирања

У табели 8.1. приказан је оптималан број атрибута за потребе класификације, након претраживања скупа свих могућих решења за сваку од метода. У табели је приказана и оригинална величина скупа, како би се упоредили ефекти редукције димензионалности података. У десет сетова података, од 18 посматраних, тачно пола или више од пола метода је смањило оригинални број атрибута на пола. Ти сетови података су *bc*, *ca*, *he*, *lc*, *ma*, *m3*, *mu*, *pi*, *sh* и *vo*.

На слици 8.1. приказан је број атрибута у оригиналном скупу података и оптималан број атрибута добијен методама филтрирања. Највећу добробит од редукције димензионалности података има скуп података *lc*, где од 56 атрибута,

методом филтрирања смо издвојили мали број атрибута, чак мање од једне шестине, за сваку од метода, изузев методе GR, који су релевантни за посматрани проблем класификације. За скуп података *ca* уочавамо да су све методе филтрирања, показале да су највише два атрибута значајна за посматрани проблем класификације, а да остали атрибути не утичу на постизање веће поузданости класификације. За скуп података *he*, који оригинално има 19 атрибута, све методе филтрирања показују да је највише 6 атрибута значајно за изучавану појаву. Код вештачког скупа података *m3*, све методе филтрирања показују да су само два атрибута значајна за посматрани проблем класификације. Методе филтрирања за скуп података *pi*, показују да највише 4 атрибута су значајна за проблем класификације, а у случају сета података *vo* 5 атрибута.

Ако имамо симултано одређивање тачности класификације у различитим сетовима података помоћу две методе, за утврђивање да ли се добијена вредност различитим методама значајно разликује користимо упоредни *t*-тест. У експерименталном истраживању користили смо упоредни *t*-тест, где је ниво значајности постављен на вредност 0.05. Током експерименталног истраживања тестирали смо значајност средње вредности разлике парова d према изразу: $t = \frac{\bar{d}\sqrt{N}}{s_d}$ где је s_d – стандардна девијација добијених разлика. Ако је израчуната вредност параметра t већа од критичне вредности, нулта хипотеза се одбацује и каже се да се d значајно разликује од нуле, односно да је разлика у паровима статистички значајна.

У табелама које следе за тачност класификације различитих класификатора и у табелама за време потребно за тренинг података су приказане ознаке „+“ и „-“, које означавају да је одређени резултат статистички бољи (+) или лошији (-) од основног класификатора на нивоу значајности који је специфициран на вредност од 0,05.

У наставку експерименталног истраживања, за изабрани оптималан број атрибута, за сваки скуп података и методу филтрирања, проверавана је тачност класификације коришћењем различитих алгоритама, и то IBk, *Naïve Bayes*, SVM, J48 и RBF мреже. У наставку текста приказани су добијени резултати. Треба уочити да су приказане различите скале на сликама за апсолутну тачност класификације, стандардну девијацију за тачност класификације, време тренинга и стандардну девијацију за време тренинга, како би се боље уочиле разлике које постоје међу резултатима.

8.2. IBk

За сваки скуп података и методу филтрирања и за изабрани оптималан број атрибута, проверавана је тачност класификације коришћењем алгоритма IBk. Табела 8.2. приказује тачност класификације за IBk за оригинални скуп података и редуковани скуп података, добијен након примене метода филтрирања.

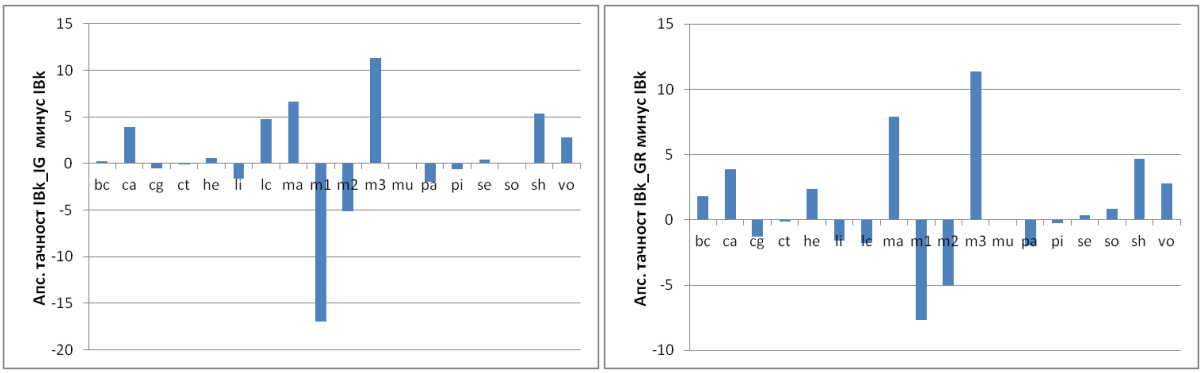
Можемо уочити да у шест сетова података (*ca*, *ma*, *m2*, *m3*, *se* и *vo*) имамо добијене резултате за бар једну од метода филтрирања који су статистички бољи од основног класификатора. Ни у једном сету података, немамо значајно лошије податке за све методе филтрирања, што значи да увек можемо изабрати методу за дати скуп података која има статистички боље резултате или резултате који су приближни оригиналном скупу података. Код три скупа података: *ca*, *m3* и *vo* све примењене методе филтрирања дају статистички боље резултате од основног класификатора.

Табела 8.2. Тачност класификације IBk алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

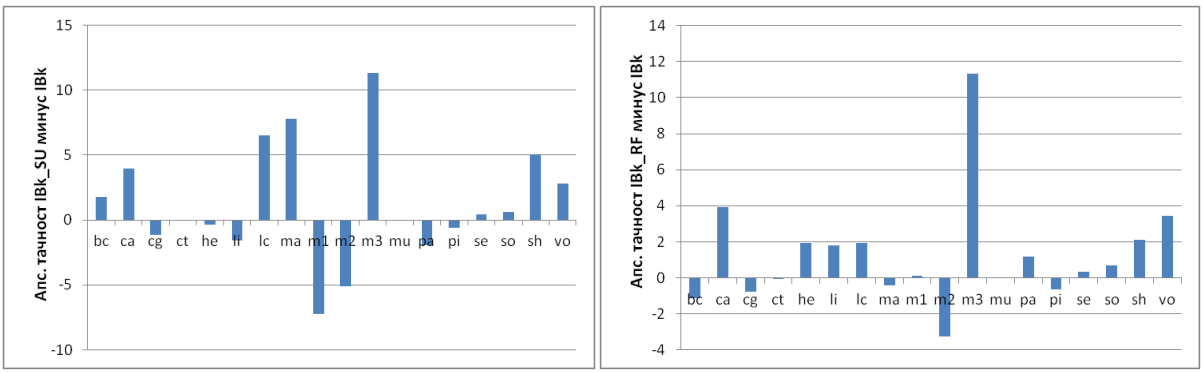
Скуп	IBk	IBk_IG	IBk_GR	IBk_SU	IBk_RF	IBk_OR	IBk_CS
bc	72.85	73.06	74.67	74.64	71.72	72.85	73.51
ca	81.57	85.51 +	85.46 +	85.51 +	85.51 +	85.45 +	85.51 +
cg	71.88	71.33	70.59	70.75	71.13	71.72	70.29
ct	98.85	98.79	98.74	98.81	98.79	98.77	98.76
he	81.40	81.97	83.78	81.02	83.33	83.65	81.91
li	62.22	60.62	60.62	60.62	64.02	60.29	60.62
lc	68.75	73.50	67.00	75.25	70.67	63.67	68.92
ma	75.60	82.27 +	83.49 +	83.38 +	75.18	82.75 +	83.36 +
m1	99.87	82.87 -	92.21	92.63	100.00	80.30 -	82.87 -
m2	72.22	67.13 -	67.13 -	67.13 -	68.98 -	75.00 +	67.13 -
m3	85.88	97.22 +	97.22 +	97.22 +	97.22 +	97.22 +	97.22 +
mu	100.00	100.00	100.00	100.00	100.00	100.00	100.00
pa	95.91	93.92	93.97	93.97	97.08	95.29	94.27
pi	70.62	69.99	70.39	69.99	69.99	71.21	69.99
se	97.15	97.57 +	97.49 +	97.57 +	97.48	97.57 +	97.57 +
so	91.20	91.26	92.06	91.79	91.89	91.11	91.68
sh	76.15	81.52	80.81	81.22	78.26	81.56	81.78
vo	92.58	95.38 +	95.36 +	95.36 +	96.04 +	95.63 +	95.63 +

На сликама 8.2, 8.3. и 8.4. приказана је апсолутна разлика у тачности класификације IBk алгоритма на основном скупу података и IBk алгоритма са различитим методама филтрирања. Примењени метод филтрирања IG је у скоро две трећине скупова података (11 скупова) показао исте или боље резултате од IBk

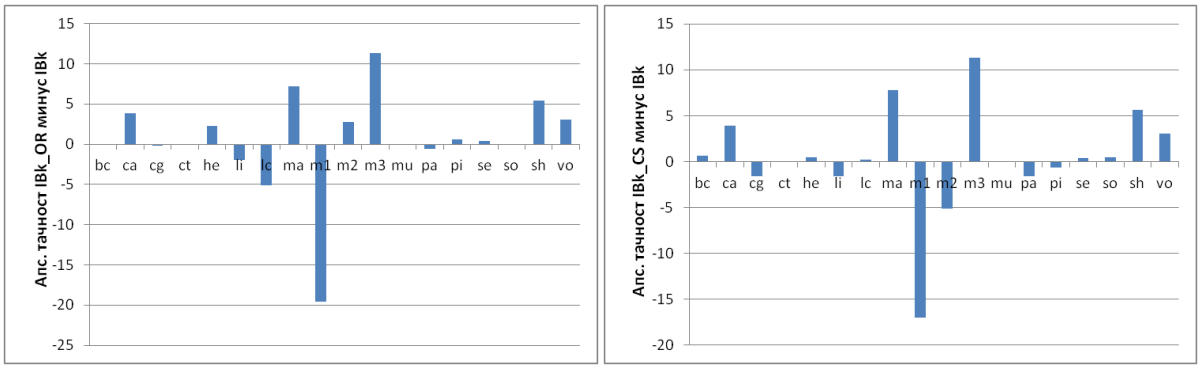
алгоритма на основном скупу података. У 5 скупова података резултати су били и статистички бољи. Метод филтрирања GR је у више од пола скупова података (10 скупова) показао исте или боље резултате од IBk алгоритма на основном скупу података. Такође, као и код методе IG, код 5 скупова података резултати су били и статистички бољи.



Слика 8.2: Апсолутна тачност класификације IBk_IG минус IBk и IBk_GR минус IBk



Слика 8.3: Апсолутна тачност класификације IBk_SU минус IBk и IBk_RF минус IBk



Слика 8.4: Апсолутна тачност класификације IBk_OR минус IBk и IBk_CS минус IBk

Примењени метод филтрирања SU је у више од пола скупова података (10 скупова) показао исте или боље резултате од IBk алгоритма на основном скупу података. У 5 скупова података резултати су били и статистички бољи. Метод филтрирања RF је у две трећине скупова података (12 скупова) показао исте или боље

резултате од IBk алгоритма на основном скупу података. У 3 скупа података, резултати су били и статистички бољи.

Метод филтрирања OR је у скоро две трећине скупова података (11 скупова) показао исте или боље резултате од IBk алгоритма на основном скупу података, а у 6 скупова података, резултати су били и статистички бољи. Примењени метод филтрирања CS је у скоро две трећине скупова података (11 скупова) показао исте или боље резултате од IBk алгоритма на основном скупу података, а у 5 скупова података резултати су били и статистички бољи.

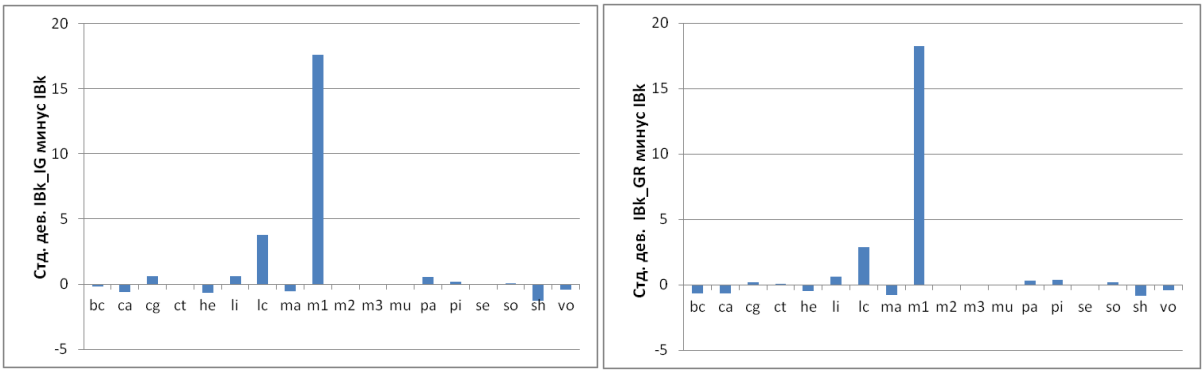
Коришћењем IBk класификатора, можемо да закључимо да је RF метода филтрирања у највећем броју случаја довела до статистички бољих резултата на посматраним скуповима података.

Табела 8.3. Стандардна девијација за тачност класификације IBk алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

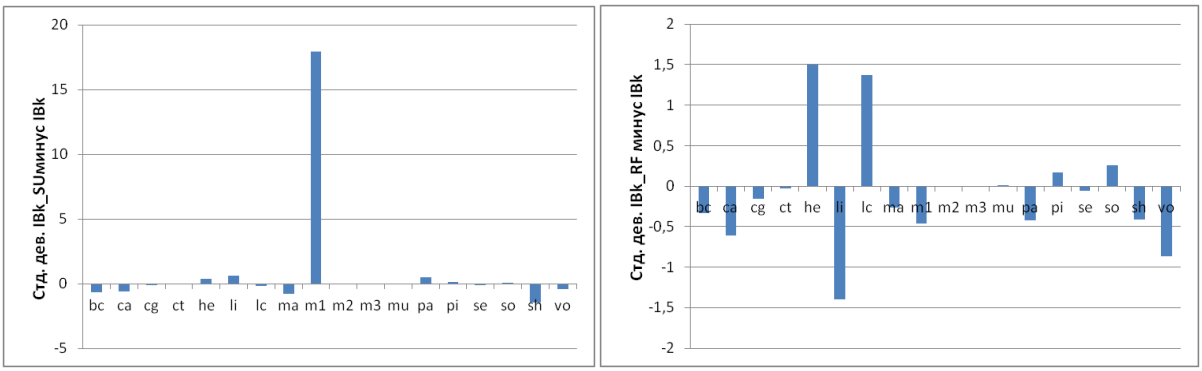
Скуп	IBk	IBk_IG	IBk_GR	IBk_SU	IBk_RF	IBk_OR	IBk_CS
bc	6.93	6.73	6.26	6.30	6.60	6.93	6.30
ca	4.57	3.96	3.93	3.96	3.96	3.94	3.96
cg	3.68	4.29	3.87	3.59	3.52	3.79	4.05
ct	0.77	0.74	0.78	0.73	0.74	0.82	0.72
he	8.55	7.86	8.10	8.93	10.05	8.88	7.99
li	8.18	8.80	8.80	8.80	6.78	8.55	8.80
lc	22.33	26.12	25.21	22.14	23.70	22.68	25.59
ma	3.90	3.37	3.13	3.10	3.64	3.08	3.09
m1	0.46	18.05	18.74	18.43	0.00	25.50	18.05
m2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mu	0.00	0.00	0.00	0.00	0.01	0.00	0.00
pa	4.52	5.07	4.84	5.00	4.10	4.78	4.69
pi	4.67	4.84	5.07	4.84	4.84	4.75	4.84
se	1.11	1.03	1.05	1.03	1.05	1.03	1.03
so	3.00	3.03	3.19	3.09	3.26	3.01	3.16
sh	8.46	7.21	7.65	7.01	8.05	6.74	6.86
vo	3.63	3.21	3.20	3.20	2.76	2.76	2.76

Стандардна девијација је у статистици апсолутна мера дисперзије у основном скупу, која нам говори колико у просеку елементи скупа одступају од аритметичке средине скупа. Најмања могућа вредност стандардне девијације је 0 и то се дешава када су сви резултати у дистрибуцији једнаки. Ова мера је осетљива на екстремне вредности, јер се базира на дистанци појединачних резултата од аритметичке средине.

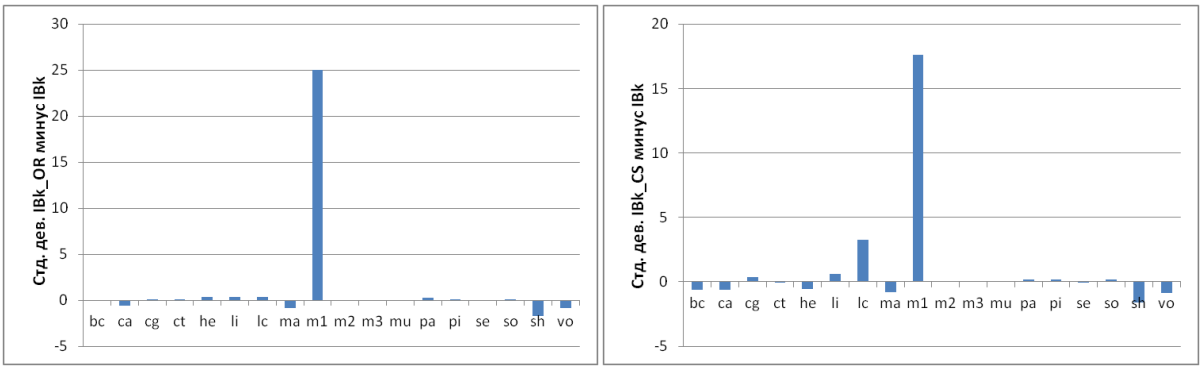
Вредност стандардне девијације је $s \in [0, +\infty)$. У овом експерименталном истраживању, као меру варијабилности, која даје информацију о различитим одступањима у статистичком скупу, користили смо стандардну девијацију. Стандардна девијација је мера одступања вредности обележја од аритметичке средине, и дата је следећим изразом: $s = \sqrt{\frac{\sum(\bar{x}-x_i)^2}{n-1}}$.



Слика 8.5: Стандардна девијација за тачност IBk_IG минус IBk и IBk_GR минус IBk



Слика 8.6: Стандардна девијација за тачност IBk_SU минус IBk и IBk_RF минус IBk



Слика 8.7: Стандардна девијација за тачност IBk_OR минус IBk и IBk_CS минус IBk

У поглављу 7 смо изнели тврдњу да је добар онај алгоритам који даје сличан резултат у свим случајевима, односно вредност стандардне девијације је минимална. Табела 8.3. приказује стандардну девијацију за тачност класификације IBk алгоритма

за оригинални и редуковани скуп података уз помоћ филтер метода. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута, осим у случају *m1* скупа података. У случају *m1* скупа података за све методе филтрирања добијамо велику вредност за стандардну девијацију, осим за методу RF.

На сликама 8.5, 8.6. и 8.7. приказана је апсолутна разлика у вредностима стандардне девијације за тачност класификације IBk алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, у позитивном и негативном смеру, то је и веће одступање између стандардних девијација. Најмање одступање у стандардној девијацији у односу на оригинални скуп података показује метода RF, која је код неких скупова података успела да смањи, а код неких да повећа стандардну девијацију.

Табела 8.4. Потребно време за тренинг (у секундама) IBk алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	IBk	IBk_IG	IBk_GR	IBk_SU	IBk_RF	IBk_OR	IBk_CS
bc	0.00	0.00	0.00	0.00	0.02 -	0.01 -	0.00
ca	0.00	0.00	0.00	0.00	0.19 -	0.03 -	0.00
cg	0.00	0.00	0.00	0.00	0.48 -	0.06 -	0.00
ct	0.00	0.03 -	0.03 -	0.03 -	4.17 -	0.24 -	0.03 -
he	0.00	0.00	0.00	0.00	0.01-	0.01 -	0.00
li	0.00	0.00	0.00	0.00	0.03 -	0.01-	0.00
lc	0.00	0.00	0.00	0.00	0.00	0.02 -	0.00
ma	0.00	0.00	0.00	0.00	0.17 -	0.02 -	0.00
m1	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m2	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m3	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
mu	0.00	0.02 -	0.03 -	0.02 -	30.20 -	0.76 -	0.02 -
pa	0.00	0.00	0.00	0.00	0.03 -	0.02 -	0.00
pi	0.00	0.00	0.00	0.00	0.16 -	0.02 -	0.00
se	0.00	0.06 -	0.06 -	0.06 -	3.13 -	0.18 -	0.05 -
so	0.00	0.00	0.00	0.00	0.42 -	0.07 -	0.00
sh	0.00	0.00	0.00	0.00	0.03 -	0.01 -	0.00
vo	0.00	0.00	0.00	0.00	0.06 -	0.02 -	0.00

У табели 8.4. приказано је потребно време за тренинг у секундама IBk алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода. Потребно време за тренинг података IBk класификатора за све оригиналне

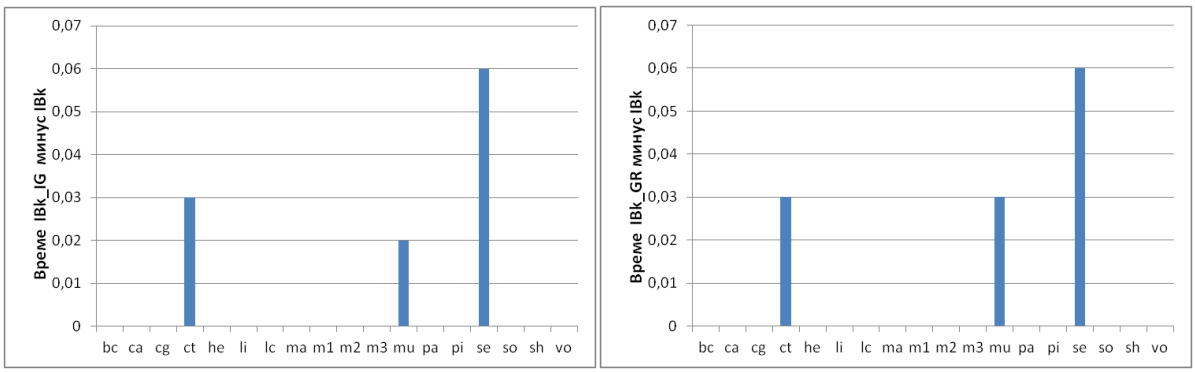
скупове података износи 0.00, док за филтер методе оно је нешто веће. Код само три скупа података (*ct*, *mu* и *se*), ни једна од метода не даје минимално потребно време за тренинг, док код свих осталих скупова података у једнако или више од пола случајева методе филтрирања дају минимално потребно време за тренинг.

На сликама 8.8, 8.9. и 8.10. приказана је апсолутна разлика у потребном времену за тренинг IBk алгоритма на основном скупу података и IBk алгоритма са различитим методама филтрирања. Примењени методи филтрирања IG, GR, SU и CS су само у три скупа података показали нешто лошије резултате за потребно време за тренинг и код тих скупова података, резултати су били и статистички лошији.

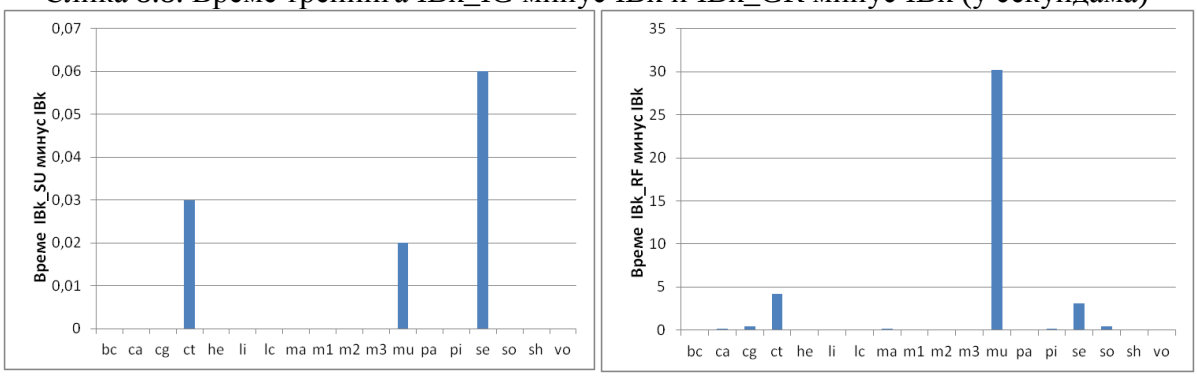
Метод филтрирања RF је у свим скуповима података, осим у једном, показао лошије резултате за потребно време за тренинг од IBk алгоритма на основном скупу података, а ти резултати су били и статистички лошији.

Метод филтрирања OR је у свим скуповима података показао лошије резултате од IBk алгоритма на основном скупу података, а ови резултати су били и статистички лошији.

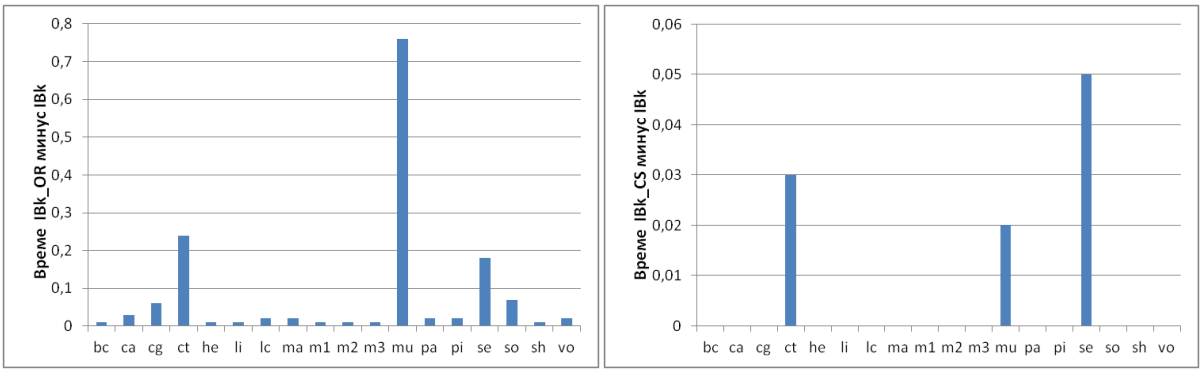
Коришћењем IBk класификатора, можемо да закључимо да су IG, GR, SU и CS методе филтрирања у најмањем броју случаја довеле до статистички лошијих резултата за потребно време за тренинг на посматраним скуповима података.



Слика 8.8. Време тренинга IBk_IG минус IBk и IBk_GR минус IBk (у секундама)



Слика 8.9: Време тренинга IBk_SU минус IBk и IBk_RF минус IBk (у секундама)



Слика 8.10: Време тренинга IBk_OR минус IBk и IBk_CS минус IBk (у секундама)

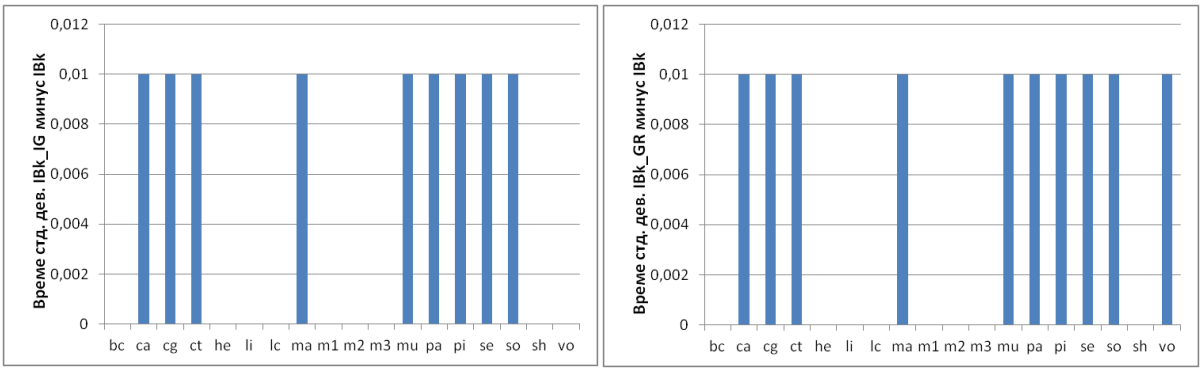
Табела 8.5. приказује стандардну девијацију за време тренинга IBk алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута. Нешто веће вредности за стандардну девијацију за време тренинга има RF метода филтрирања.

Табела 8.5. Стандардна девијација за време тренинга (у секундама) IBk алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

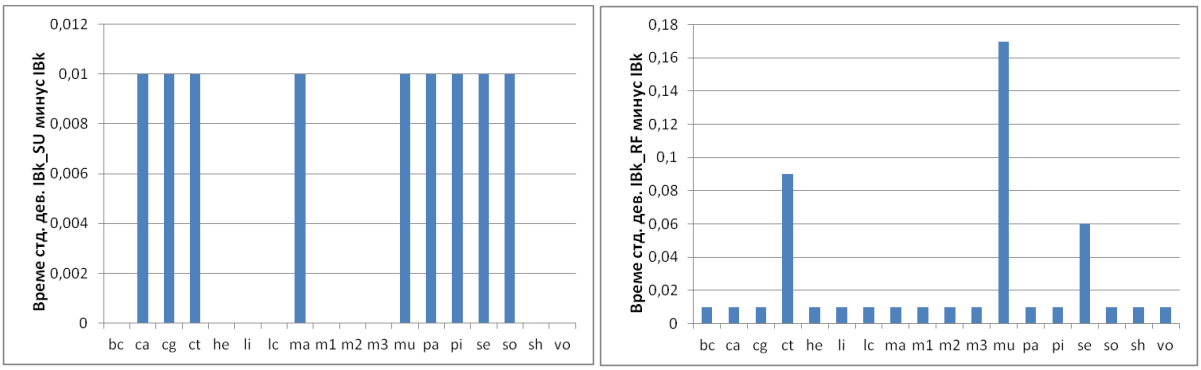
Скуп	IBk	IBk_IG	IBk_GR	IBk_SU	IBk_RF	IBk_OR	IBk_CS
bc	0.00	0.00	0.00	0.00	0.01	0.01	0.00
ca	0.00	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.00	0.01	0.01	0.01	0.01	0.01	0.01
ct	0.00	0.01	0.01	0.01	0.09	0.02	0.01
he	0.00	0.00	0.00	0.00	0.01	0.01	0.00
li	0.00	0.00	0.00	0.00	0.01	0.01	0.00
lc	0.00	0.00	0.00	0.00	0.01	0.01	0.00
ma	0.00	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.00	0.00	0.00	0.00	0.01	0.01	0.00
m2	0.00	0.00	0.00	0.00	0.01	0.01	0.00
m3	0.00	0.00	0.00	0.00	0.01	0.01	0.00
mu	0.00	0.01	0.01	0.01	0.17	0.01	0.01
pa	0.00	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.00	0.01	0.01	0.01	0.01	0.01	0.01
se	0.00	0.01	0.01	0.01	0.06	0.01	0.01
so	0.00	0.01	0.01	0.01	0.01	0.01	0.00
sh	0.00	0.00	0.00	0.00	0.01	0.01	0.00
vo	0.00	0.00	0.01	0.00	0.01	0.01	0.00

На сликама 8.11, 8.12. и 8.13. приказана је апсолутна разлика у вредностима стандардне девијације за време тренинга IBk алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Ако је вредност на сликама приближна нули,

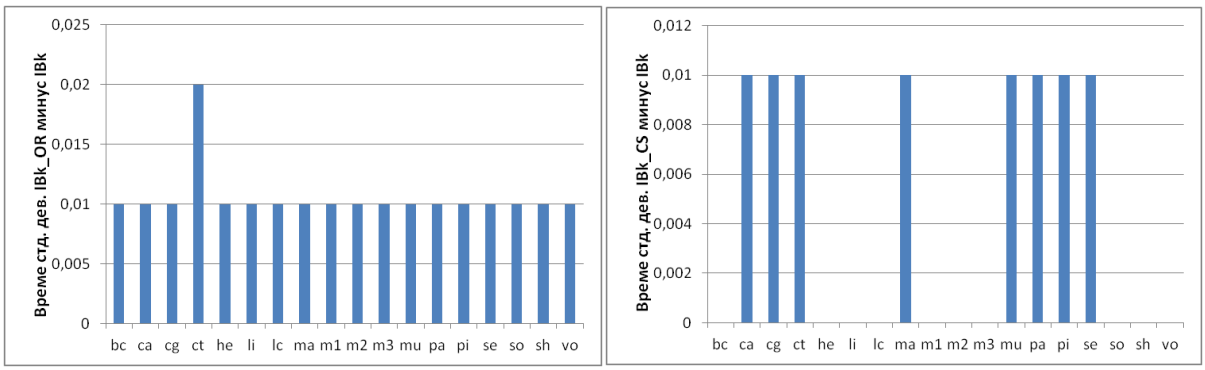
онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, у позитивном и негативном смеру, то је и веће одступање између стандардних девијација. Највеће одступање у стандардној девијацији у односу на оригинални скуп података показује метода RF, која је код свих скупова података успела да повећа стандардну девијацију.



Слика 8.11: Стандардна девијација за време IBk_IG минус IBk и IBk_GR минус IBk



Слика 8.12: Стандардна девијација за време IBk_SU минус IBk и IBk_RF минус IBk



Слика 8.13: Стандардна девијација за време IBk_OR минус IBk и IBk_CS минус IBk

8.3. Naïve Bayes

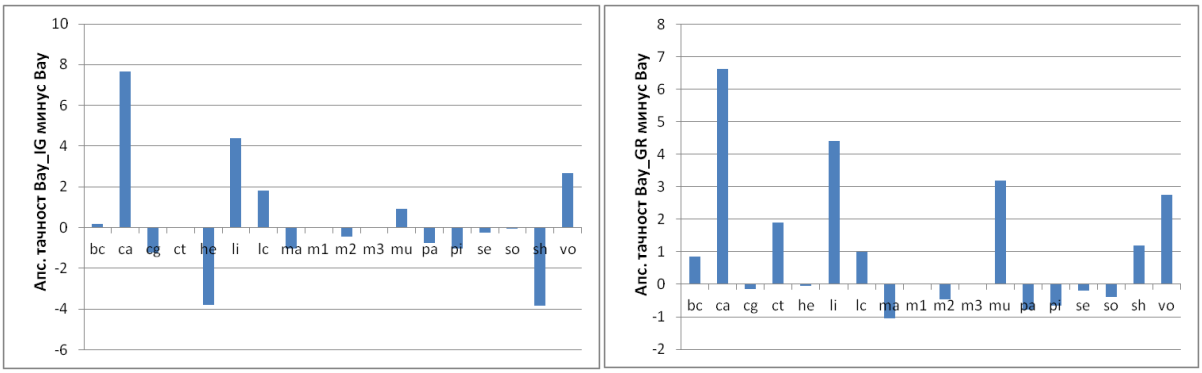
На основу приказаних података у табели 8.6. можемо уочити да у седам сетова података (*ca*, *ct*, *m2*, *mu*, *pa*, *se* и *vo*) имамо добијене резултате за бар једну од метода

филтрирања који су статистички бољи од основног класификатора. И поред смањења димензионалности података, ни у једном сету података, немамо значајно лошије податке за све методе филтрирања, што значи да увек можемо изабрати методу за дати скуп података која има статистички боље резултате или резултате који су приближни оригиналном скупу података. Све методе филтрирања имају статистички боље резултате од основног класификатора у случају три скупа података: *ca*, *mu* и *vo*.

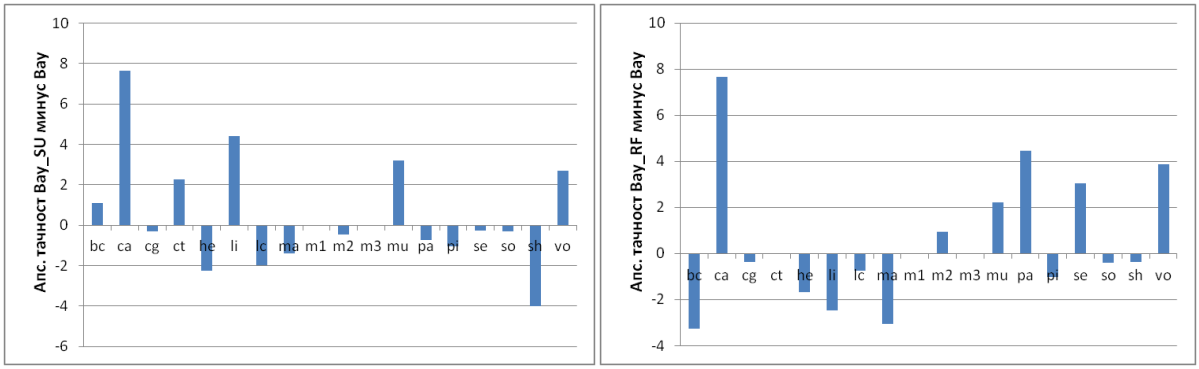
Табела 8.6. Тачност класификације *Naïve Bayes* алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	Bay	Bay_IG	Bay_GR	Bay _SU	Bay_RF	Bay_OR	Bay _CS
bc	72.70	72.87	73.54	73.78	69.45	72.70	72.59
ca	77.86	85.51 +	84.49 +	85.51 +	85.51 +	82.54 +	85.51 +
cg	75.16	73.95	75.01	74.87	74.79	74.88	74.94
ct	87.30	87.31	89.21 +	89.57 +	87.28	88.32 +	89.68 +
he	83.81	80.01	83.77	81.55	82.12	83.17	79.95
li	54.89	59.29	59.29	59.29	52.41	53.99	59.29
lc	78.42	80.25	79.42	76.42	77.67	71.83	75.33
ma	82.64	81.62	81.58	81.26	79.59 -	80.29 -	81.25
m1	74.64	74.64	74.64	74.64	74.64	74.64	74.64
m2	61.57	61.11 -	61.11 -	61.11 -	62.50 +	60.65 -	61.11 -
m3	97.22	97.22	97.22	97.22	97.22	97.22	97.22
mu	95.76	96.68 +	98.95 +	98.95 +	97.97 +	98.33 +	98.33 +
pa	69.98	69.21	69.21	69.26	74.44 +	69.99	69.26
pi	75.75	74.72	75.09	74.72	74.72	73.86	74.72
se	80.17	79.92	79.98	79.92	83.20 +	79.92	79.92
so	92.94	92.91	92.56	92.62	92.52	92.94	92.43
sh	83.59	79.74	84.78	79.59	83.22	79.81	79.85
vo	90.02	92.71 +	92.76 +	92.71 +	93.88 +	95.63 +	95.63 +

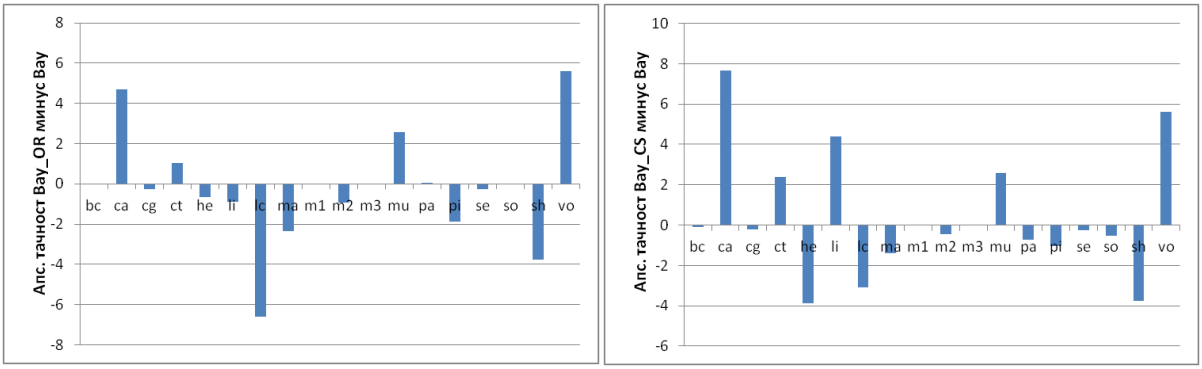
Апсолутна разлика у тачности класификације *Naïve Bayes* алгоритма на основном скупу података и *Naïve Bayes* алгоритма са различитим методама филтрирања приказана је на сликама 8.14, 8.15. и 8.16. Примењени метод филтрирања IG је у више од пола скупова података (10 скупова) показао исте или боље резултате од *Naïve Bayes* алгоритма на основном скупу података, а у 3 скупа података, резултати су били и статистички бољи. Код мерења тачности класификације, метод филтрирања GR је у више од пола скупова података (10 скупова) показао исте или боље резултате од *Naïve Bayes* алгоритма на основном скупу података. Код методе GR, у 4 скупа података резултати су били и статистички бољи.



Слика 8.14: Апсолутна тачност класификације Bay_IG минус Bay и Bay_GR минус Bay



Слика 8.15: Апсолутна тачност класификације Bay_SU минус Bay и Bay_RF минус Bay



Слика 8.16: Апсолутна тачност класификације Bay_OR минус Bay и Bay_CS минус Bay

Примењени метод филтрирања SU је у нешто мање од пола скупова података (8 скупова) показао исте или боље резултате од *Naïve Bayes* алгоритма на основном скупу података. У 4 скупа података, резултати су били и статистички бољи. Метод филтрирања RF је у нешто мање од пола скупова података (8 скупова) показао исте или боље резултате од *Naïve Bayes* алгоритма на основном скупу података, док у чак 6 скупова података, резултати су били и статистички бољи.

Приликом утврђивања тачности класификације, метод филтрирања OR је у пола скупова података (9 скупова) показао исте или боље резултате од *Naïve Bayes* алгоритма на основном скупу података, а у 4 скупа података, резултати су били и статистички бољи. Примењени метод филтрирања CS је у нешто мање од пола скупова

података (7 скупова) показао исте или боље резултате од *Naïve Bayes* алгоритма на основном скупу података, а у 4 скупа података резултати су били и статистички бољи.

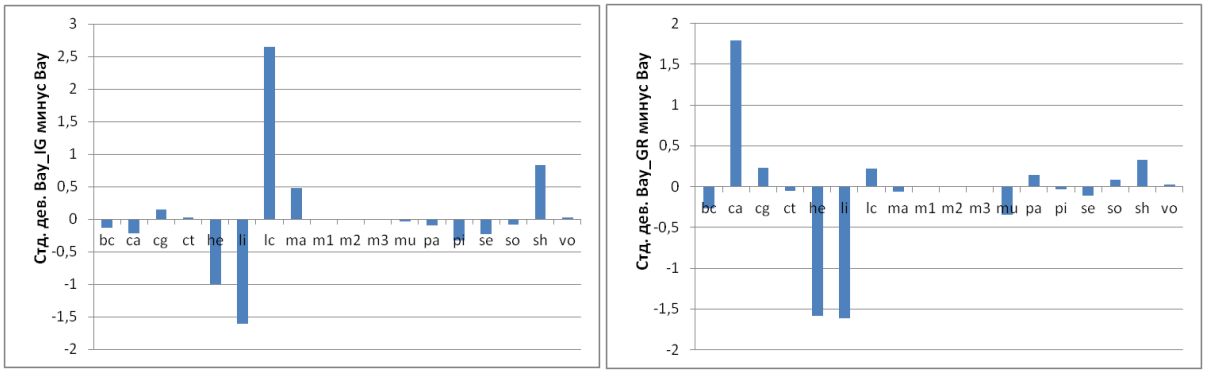
Коришћењем *Naïve Bayes* класификатора, можемо да закључимо да је RF метода филтрирања у највећем броју случаја довела до статистички бољих резултата на посматраним скуповима података.

Табела 8.7. Стандардна девијација за тачност класификације *Naïve Bayes* алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

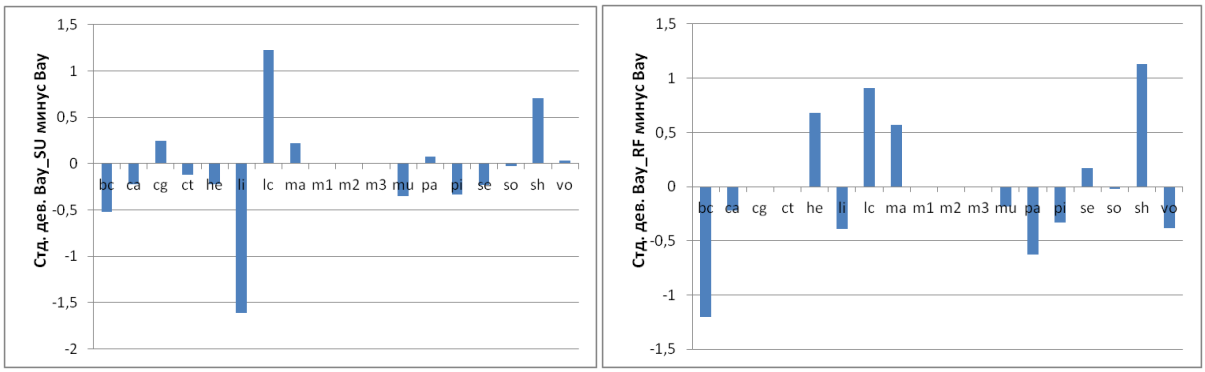
Скуп	Bay	Bay_IG	Bay_GR	Bay _SU	Bay_RF	Bay_OR	Bay _CS
bc	7.74	7.61	7.47	7.22	6.54	7.74	6.79
ca	4.18	3.96	5.97	3.96	3.96	5.53	3.96
cg	3.48	3.63	3.71	3.73	3.48	4.01	3.65
ct	2.21	2.23	2.16	2.09	2.21	2.21	1.99
he	9.70	8.70	8.12	9.48	10.38	8.84	8.80
li	8.83	7.22	7.22	7.22	8.44	9.74	7.22
lc	21.12	23.77	21.34	22.35	22.03	17.16	23.39
ma	3.11	3.59	3.05	3.33	3.68	3.54	3.33
m1	4.26	4.26	4.26	4.26	4.26	4.26	4.26
m2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mu	0.73	0.69	0.38	0.38	0.55	0.44	0.44
pa	9.51	9.41	9.65	9.59	8.88	9.44	9.71
pi	5.32	4.99	5.29	4.99	4.99	5.90	4.99
se	2.12	1.89	2.01	1.89	2.29	1.89	1.89
so	2.92	2.84	3.00	2.89	2.90	2.89	3.01
sh	5.98	6.81	6.31	6.69	7.11	6.56	6.75
vo	3.91	3.94	3.94	3.94	3.53	2.76	2.76

С обзиром да смо изнели тврдњу да је добар онај алгоритам који даје сличан резултат у свим случајевима, односно вредност стандардне девијације је минимална, разматраћемо у наставку текста вредности за стандардну девијацију за тачност класификације. Табела 8.7. приказује стандардну девијацију за тачност класификације *Naïve Bayes* алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Може се уочити да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута, у свим случајевима код свих скупова података. За разлику од IBk алгоритма, код *Naïve Bayes* алгоритма имамо мања одступања у вредностима стандардне девијације за тачност класификације.

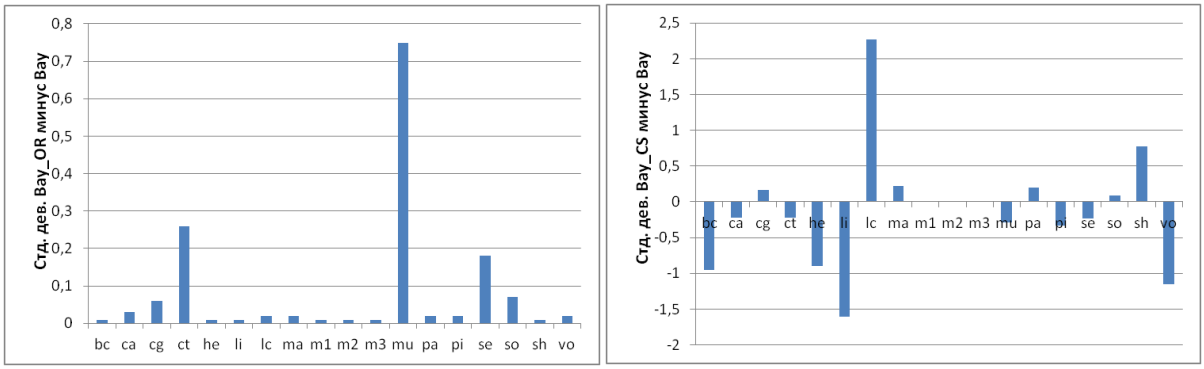
На сликама 8.17, 8.18. и 8.19. приказана је апсолутна разлика у вредностима стандардне девијације за тачност класификације *Naïve Bayes* алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода.



Слика 8.17: Стандардна девијација за тачност Bay_IG минус Bay и Bay_GR минус Bay



Слика 8.18: Стандардна девијација за тачност Bay_SU минус Bay и Bay_RF минус Bay



Слика 8.19: Стандардна девијација за тачност Bay_OR минус Bay и Bay_CS минус Bay

Вредности на скали са апсолутном разликом су мање на овим сликама у односу на IBk алгоритам, како би се уочиле разлике у вредностима између различитих метода, с обзиром да су оне заиста мале. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликују, а уколико она више одступа од нуле, у позитивном и негативном смеру, то је и веће одступање између стандардних девијација. Најмање одступање у стандардној девијацији у односу на оригинални скуп података, показује метода OR, док највеће одступање има метода IG и CS, које су код

неких скупова података успеле да смање, а код неких да повећају стандардну девијацију.

Потребно време за тренинг *Naïve Bayes* алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода приказано је у табели 8.8. Потребно време за тренинг података *Naïve Bayes* класификатора за све оригиналне скупове података је мало и износи највише 0.01, док за филтер методе оно је нешто веће. Код само три скупа података (*ct, mu* и *se*), ни једна од метода не даје минимално потребно време за тренинг, док код свих осталих скупова података у једнако или више од пола случајева методе филтрирања дају минимално потребно време за тренинг.

Табела 8.8. Потребно време за тренинг (у секундама) *Naïve Bayes* алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	Bay	Bay_IG	Bay_GR	Bay _SU	Bay_RF	Bay_OR	Bay _CS
bc	0.00	0.00	0.00	0.00	0.02 -	0.01 -	0.00
ca	0.00	0.00	0.00	0.00	0.19 -	0.03 -	0.00
cg	0.00	0.00	0.01	0.00	0.48 -	0.06 -	0.00
ct	0.01	0.04 -	0.04 -	0.04 -	4.26 -	0.27 -	0.04 -
he	0.00	0.00	0.00	0.00	0.01 -	0.01 -	0.00
li	0.00	0.00	0.00	0.00	0.03 -	0.01 -	0.00
lc	0.00	0.00	0.00	0.00	0.00	0.02 -	0.00
ma	0.00	0.00	0.00	0.00	0.17 -	0.02 -	0.00
m1	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m2	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m3	0.00	0.00	0.00	0.00	0.04 -	0.01 -	0.00
mu	0.01	0.02 -	0.02 -	0.03 -	30.15 -	0.76 -	0.02 -
pa	0.00	0.00	0.00	0.00	0.03 -	0.02 -	0.00
pi	0.00	0.00	0.00	0.00	0.16 -	0.02 -	0.00
se	0.01	0.06 -	0.06 -	0.06 -	3.13 -	0.19 -	0.06 -
so	0.00	0.00	0.00	0.00	0.42 -	0.07 -	0.00
sh	0.00	0.00	0.00	0.00	0.03 -	0.01 -	0.00
vo	0.00	0.00	0.00	0.00	0.06 -	0.02 -	0.00

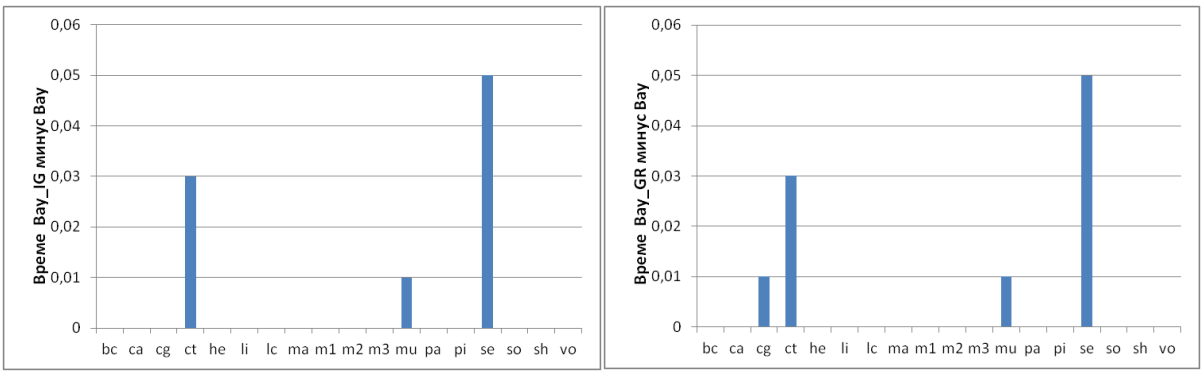
На сликама 8.20, 8.21. и 8.22. приказана је апсолутна разлика у потребном времену за тренинг *Naïve Bayes* алгоритма на основном скупу података и *Naïve Bayes* алгоритма са различитим методама филтрирања. Примењени метод филтрирања IG је у само три скупа података показао нешто лошије резултате за потребно време за тренинг и код тих скупова података, резултати су били и статистички лошији. Метод филтрирања GR је у четири скупа података показао нешто лошије резултате за

потребно време за тренинг, а код 3 скупа података резултати су били и статистички лошији.

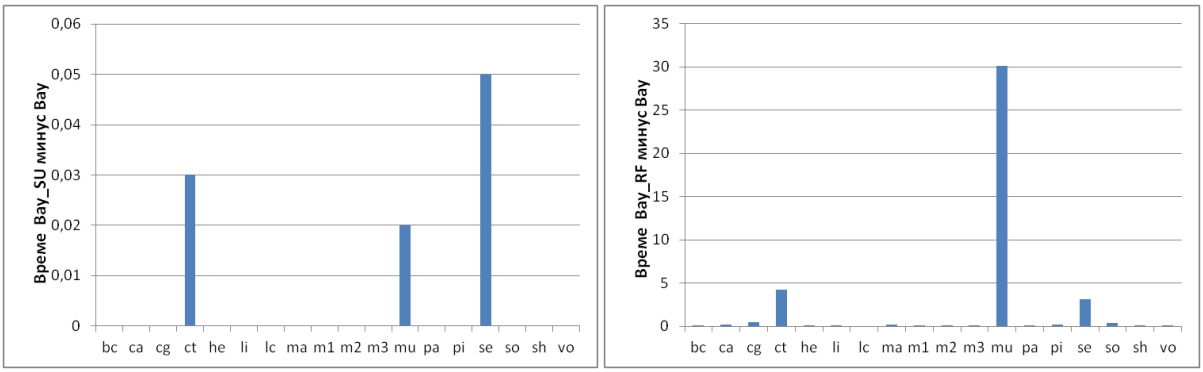
Примењени метод филтрирања SU је у 3 скупа података показао лошије резултате за потребно време за тренинг од *Naïve Bayes* алгоритма на основном скупу података, а у тим скуповима података, резултати су били и статистички лошији. Метод филтрирања RF је у скоро свим скуповима података (17 скупова) показао лошије резултате за потребно време за тренинг од *Naïve Bayes* алгоритма на основном скупу података, а у тим скуповима података, резултати су били и статистички лошији.

У свим скуповима података метод филтрирања OR је показао лошије резултате од *Naïve Bayes* алгоритма на основном скупу података, а ови резултати су били и статистички лошији. Примењени метод филтрирања CS је у само 3 скупа података показао лошије резултате од *Naïve Bayes* алгоритма на основном скупу података, а у 3 скупа података, резултати су били и статистички лошији.

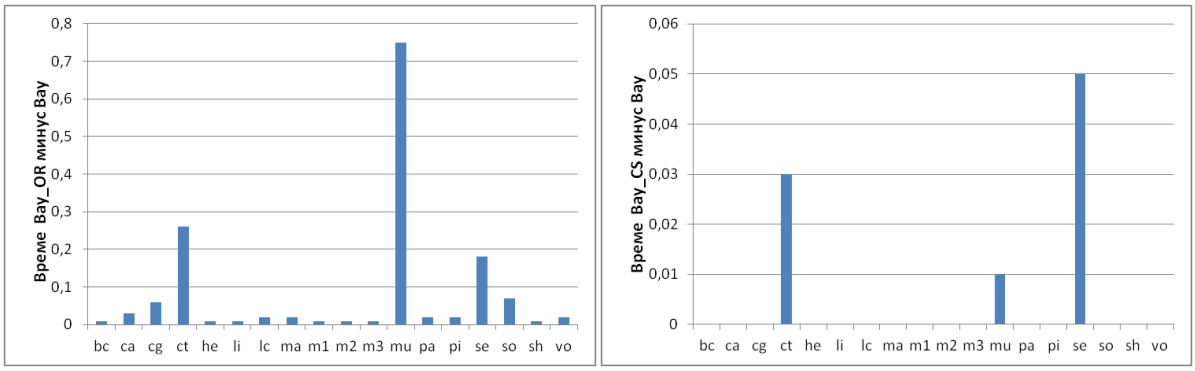
Коришћењем *Naïve Bayes* класификатора, можемо да закључимо да су IG, GR, SU и CS методе филтрирања у најмањем броју случаја довели до статистички лошијих резултата за потребно време за тренинг на посматраним скуповима података.



Слика 8.20: Време тренинга Bay_IG минус Bay и Bay_GR минус Bay (у секундама)



Слика 8.21: Време тренинга Bay_SU минус Bay и Bay_RF минус Bay (у секундама)



Слика 8.22: Време тренинга Bay_OR минус Bay и Bay_CS минус Bay (у секундама)

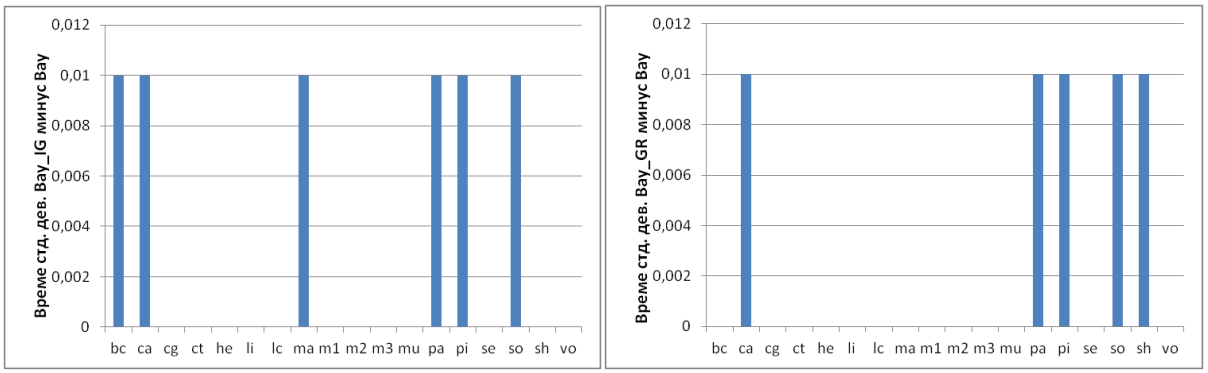
Табела 8.9. приказује стандардну девијацију за време тренинга *Naïve Bayes* алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута. Нешто веће вредности за стандардну девијацију за време тренинга има RF метода филтрирања.

Табела 8.9. Стандардна девијација за време тренинга (у секундама) *Naïve Bayes* алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

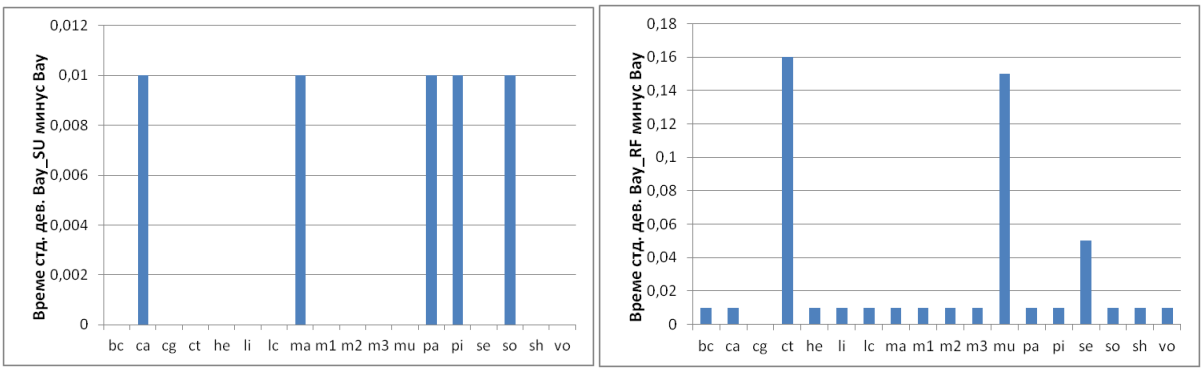
Скуп	Bay	Bay_IG	Bay_GR	Bay _SU	Bay_RF	Bay_OR	Bay _CS
bc	0.00	0.01	0.00	0.00	0.01	0.01	0.00
ca	0.00	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ct	0.01	0.01	0.01	0.01	0.17	0.02	0.01
he	0.00	0.00	0.00	0.00	0.01	0.01	0.00
li	0.00	0.00	0.00	0.00	0.01	0.01	0.00
lc	0.00	0.00	0.00	0.00	0.01	0.01	0.00
ma	0.00	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.00	0.00	0.00	0.00	0.01	0.01	0.00
m2	0.00	0.00	0.00	0.00	0.01	0.01	0.00
m3	0.00	0.00	0.00	0.00	0.01	0.01	0.00
mu	0.01	0.01	0.01	0.01	0.16	0.01	0.01
pa	0.00	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.00	0.01	0.01	0.01	0.01	0.01	0.01
se	0.01	0.01	0.01	0.01	0.06	0.01	0.01
so	0.00	0.01	0.01	0.01	0.01	0.01	0.01
sh	0.00	0.00	0.01	0.00	0.01	0.01	0.01
vo	0.00	0.00	0.00	0.00	0.01	0.01	0.00

На сликама 8.23, 8.24. и 8.25. приказана је апсолутна разлика у вредностима стандардне девијације за време тренинга *Naïve Bayes* алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Највеће одступање у стандардној

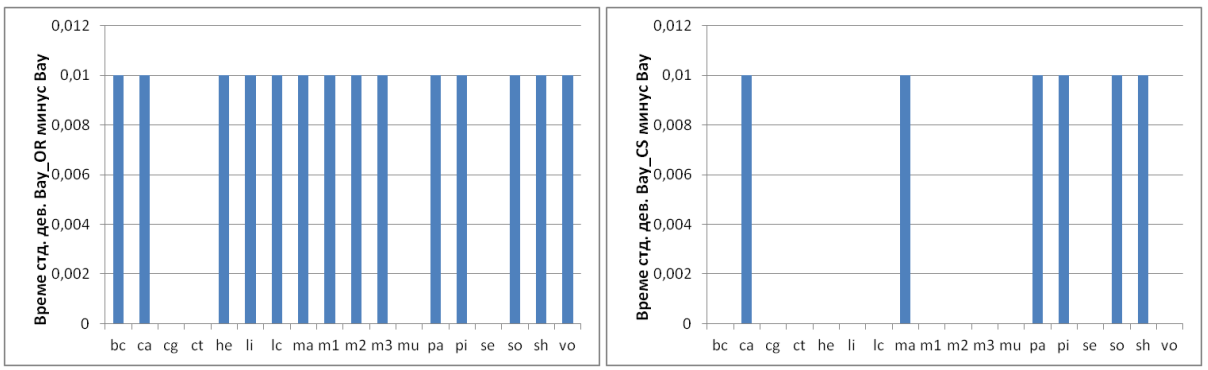
девијацији у односу на оригинални скуп података показује метода RF, која је код свих скупова података осим једног, успела да повећа стандардну девијацију.



Слика 8.23: Стандардна девијација за време Bay_IG минус Bay и Bay_GR минус Bay



Слика 8.24: Стандардна девијација за време Bay_SU минус Bay и Bay_RF минус Bay



Слика 8.25: Стандардна девијација за време Bay_OR минус Bay и Bay_CS минус Bay

8.4. SVM

Увидом у податке приказане у табели 8.10. можемо уочити да коришћењем SVM алгоритма у десет скупова података (*ca, cg, li, ma, m1, m3, pa, pi, so* и *sh*) имамо добијене резултате за бар једну од метода филтрирања који су статистички бољи од основног класификатора. Само у два сета података, имамо значајно лошије податке за

све методе филтрирања. Код три скупа података: *ca*, *m3* и *sh* све методе филтрирања су статистички боље од основног класификатора.

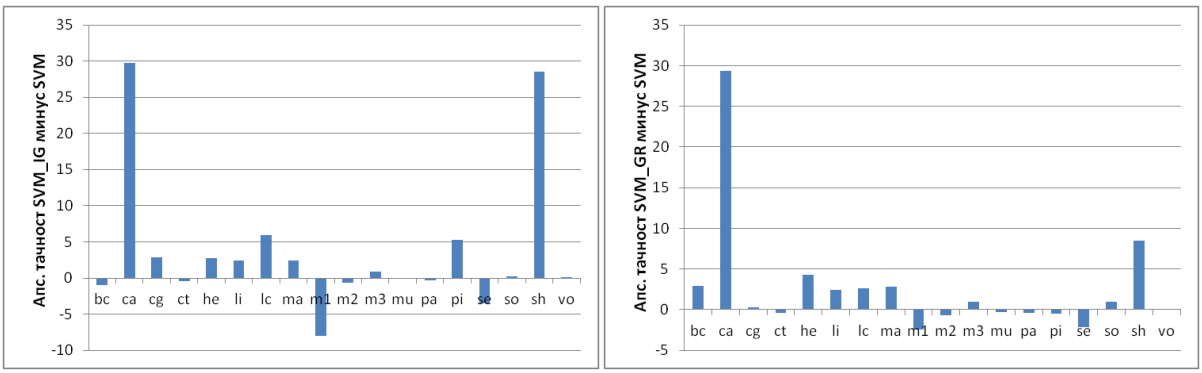
Табела 8.10. Тачност класификације SVM алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	SVM	SVM_IG	SVM_GR	SVM _SU	SVM _RF	SVM_OR	SVM_CS
bc	72.18	71.24	75.09	75.30	71.48	72.18	74.32
ca	55.80	85.51 +	85.19 +	85.51 +	85.51 +	85.43 +	85.51 +
cg	70.00	72.85 +	70.25	70.24	70.00	72.12 +	70.23
ct	81.01	80.57 -	80.58 -	80.50 -	80.88	80.90	79.93 -
he	79.38	82.15	83.70	83.31	84.09	84.49	81.97
li	59.37	61.77	61.77	61.77	63.64 +	60.79	61.77
lc	72.67	78.58	75.25	79.58	74.33	74.08	75.00
ma	80.29	82.68 +	83.15 +	83.06 +	79.95	82.46	83.03 +
m1	91.37	83.32 -	88.96	89.14	97.83 +	85.26 -	83.32 -
m2	67.82	67.13 -	67.13 -	67.13 -	66.67 -	67.59 -	67.13 -
m3	96.30	97.22 +	97.22 +	97.22 +	97.22 +	97.22 +	97.22 +
mu	100.00	99.98	99.72 -	99.72 -	99.99	99.98	99.98
pa	79.36	79.00	79.00	79.00	86.67 +	79.15	79.00
pi	65.11	70.36 +	64.59	70.36 +	70.36 +	64.49	70.36 +
se	63.98	60.52 -	61.85 -	60.52 -	60.36 -	60.52 -	60.52 -
so	93.63	93.88	94.55 +	94.20	94.44	93.59	93.62
sh	55.93	84.48 +	64.41 +	84.07 +	83.33 +	84.41 +	84.59 +
vo	95.61	95.63	95.63	95.63	95.63	95.63	95.63

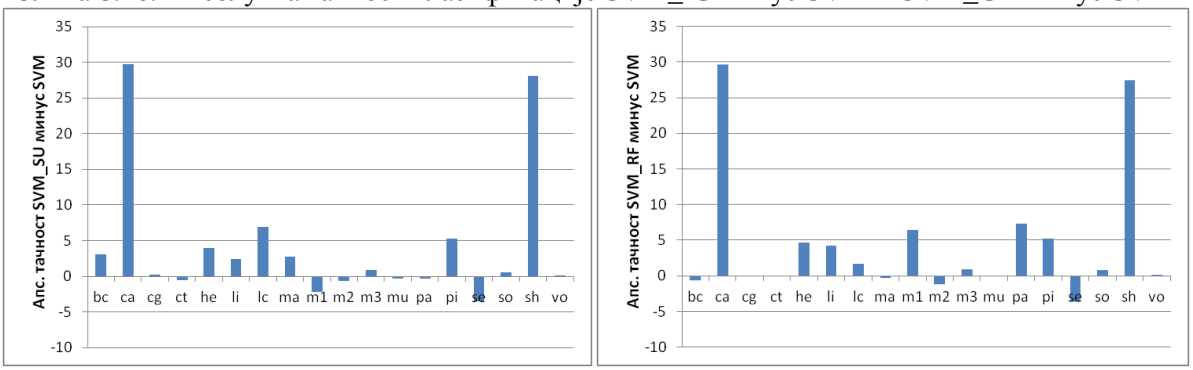
На сликама 8.26, 8.27. и 8.28. приказана је апсолутна разлика у тачности класификације SVM алгоритма на основном скупу података и SVM алгоритма са примењеним различитим методама филтрирања. Примењени метод филтрирања IG је у више од пола скупова података (11 скупова) показао исте или боље резултате од SVM алгоритма на основном скупу података, а у 6 скупова података резултати су били и статистички бољи. Метод филтрирања GR је у више од пола скупова података (11 скупова) показао исте или боље резултате од SVM алгоритма на основном скупу података. Код методе GR, у 5 скупова података резултати су били и статистички бољи.

Примењени метод филтрирања SU је у две трећине скупова података (12 скупова) показао исте или боље резултате од SVM алгоритма на основном скупу података, док су у 5 скупова података резултати били и статистички бољи. Метод филтрирања RF је у две трећине скупова података (12 скупова) показао исте или боље резултате од SVM алгоритма на основном скупу података. У 7 скупова података, резултати су били и статистички бољи.

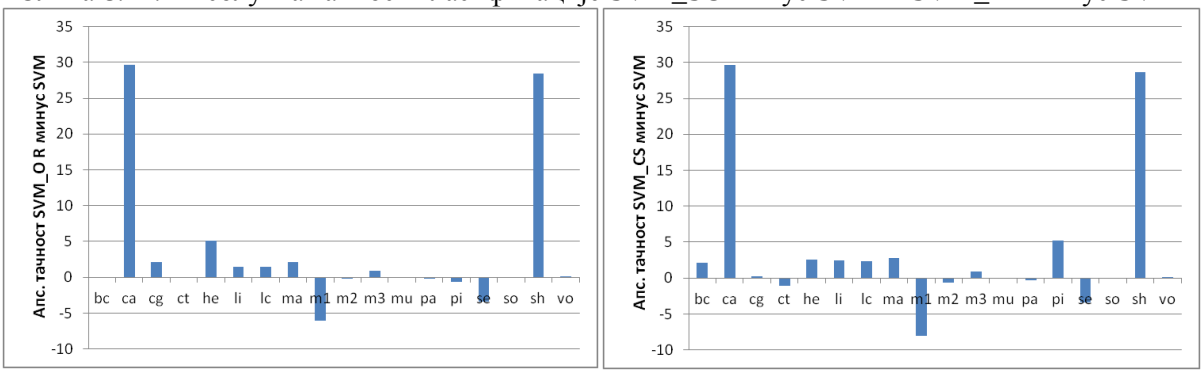
Метод филтрирања OR је у више од пола скупова података (10 скупова) показао исте или боље резултате од SVM алгоритма на основном скупу података, а у 4 скупа података резултати су били и статистички бољи. Примењени метод филтрирања CS је у више од пола скупова података (11 скупова) показао исте или боље резултате од SVM алгоритма на основном скупу података, а у 5 скупова података резултати су били и статистички бољи.



Слика 8.26: Апсолутна тачност класификације SVM_IG минус SVM и SVM_GR минус SVM



Слика 8.27: Апсолутна тачност класификације SVM_SU минус SVM и SVM_RF минус SVM



Слика 8.28: Апсолутна тачност класификације SVM_OR минус SVM и SVM_CS минус SVM

Коришћењем SVM класификатора, можемо да закључимо да је RF метода филтрирања у највећем броју случаја довела до статистички бољих резултата на посматраним скуповима података.

Табела 8.11. приказује стандардну девијацију за тачност класификације SVM алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Из

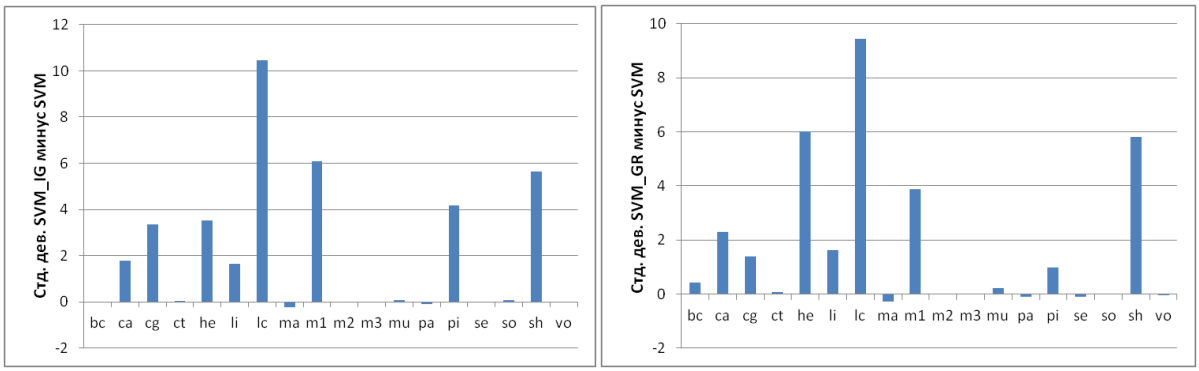
табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута. Ове разлике су нешто веће у односу на *Naïve Bayes* алгоритам, а нешто мање у односу на IBk алгоритам. У случају *lc* скупа података за све методе филтрирања добијамо највеће вредности за стандардну девијацију. Поред већих вредности у апсолутним износима, код SVM алгоритма и овог скупа података, применом свих метода филтрирања добијамо значајно веће вредности за стандардну девијацију.

Табела 8.11. Стандардна девијација за тачност класификације SVM алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

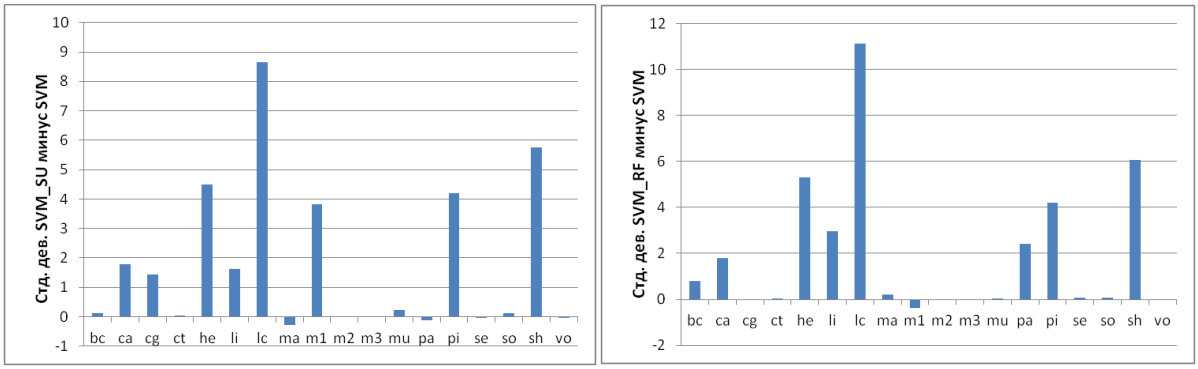
Скуп	SVM	SVM_IG	SVM_GR	SVM_SU	SVM _RF	SVM_OR	SVM_CS
bc	5.86	5.86	6.29	5.97	6.64	5.86	5.97
ca	2.18	3.96	4.47	3.96	3.96	4.01	3.96
cg	0.00	3.35	1.38	1.44	0.00	2.58	1.34
ct	0.99	1.02	1.05	1.04	1.02	1.04	0.93
he	2.26	5.80	8.29	6.75	7.57	8.10	6.13
li	2.28	3.91	3.91	3.91	5.23	3.55	3.91
lc	11.12	21.59	20.56	19.76	22.24	21.12	23.27
ma	3.41	3.18	3.14	3.12	3.61	3.20	3.10
m1	3.10	9.20	6.98	6.93	2.71	8.68	9.20
m2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mu	0.00	0.09	0.23	0.23	0.04	0.09	0.09
pa	4.46	4.35	4.35	4.35	6.87	4.33	4.35
pi	0.34	4.53	1.32	4.53	4.53	1.63	4.53
se	3.47	3.43	3.36	3.43	3.53	3.43	3.43
so	2.22	2.31	2.25	2.35	2.28	2.17	2.63
sh	1.12	6.76	6.94	6.87	7.19	6.56	6.61
vo	2.77	2.76	2.76	2.76	2.76	2.76	2.76

Слике 8.29, 8.30. и 8.31. приказују апсолутну разлику у вредностима стандардне девијације за тачност класификације SVM алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, у позитивном и негативном смеру, то је и веће одступање између стандардних девијација. Готово све методе показују одступање у стандардној девијацији у односу на оригинални скуп података у истој мери, с тим што можемо да запазимо у односу на претходне алгоритме, да су ово позитивна одступања, односно стандардна девијација

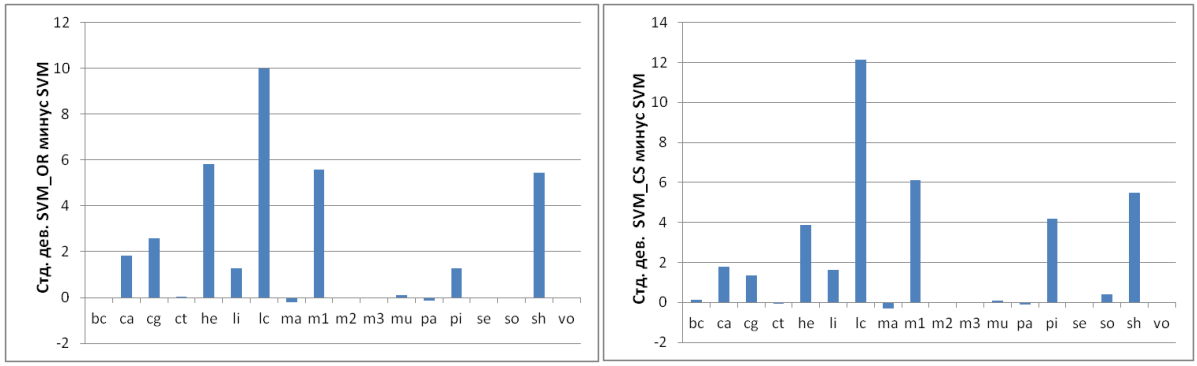
алгоритма SVM са претходном селекцијом атрибута различитим методама је већа у односу на стандардни SVM алгоритам.



Слика 8.29: Стандардна девијација за тачност SVM_IG минус SVM и SVM_GR минус SVM



Слика 8.30: Стандардна девијација за тачност SVM_SU минус SVM и SVM_RF минус SVM



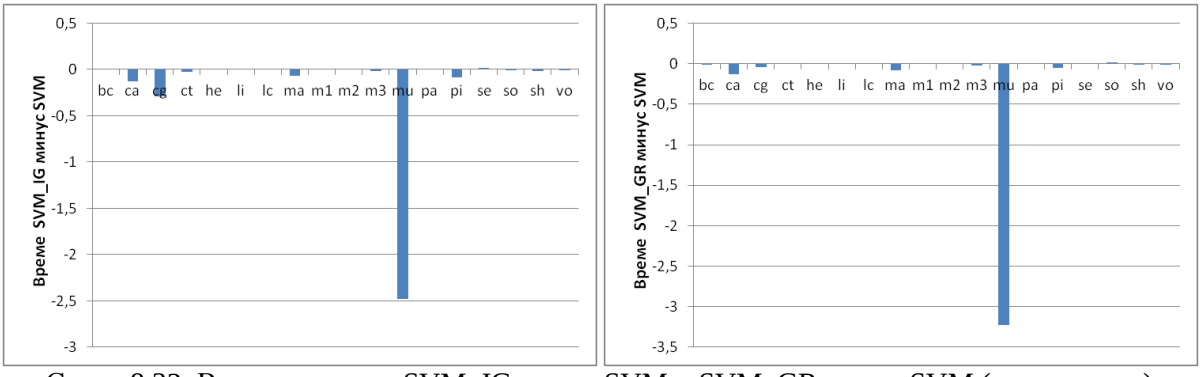
Слика 8.31: Стандардна девијација за тачност SVM_OR минус SVM и SVM_CS минус SVM

У табели 8.12. приказано је потребно време за тренинг SVM алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода. Потребно време за тренинг података SVM класификатора за све оригиналне скупове података је нешто веће него код IBk и *Naïve Bayes* алгоритма. Можемо уочити да неке методе филтрирања код SVM алгоритма смањују, а неке повећавају неопходно време за тренинг података. Код свих скупова података, изузев једног (*se*), потребно време за тренинг података бар једном од метода филтрирања је једнако или мање од основног класификатора.

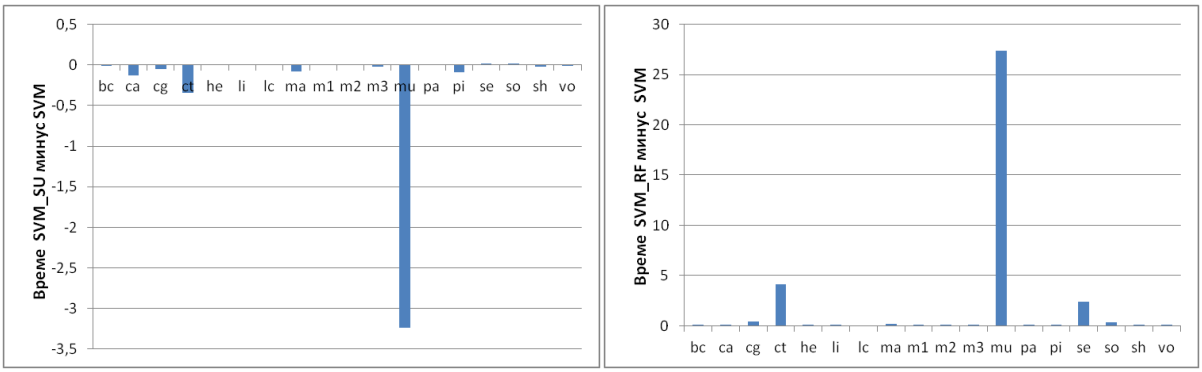
Табела 8.12. Потребно време за тренинг (у секундама) SVM алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	SVM	SVM_IG	SVM_GR	SVM _SU	SVM _RF	SVM_OR	SVM _CS
bc	0.02	0.02	0.01	0.01 +	0.03	0.03	0.01
ca	0.15	0.02 +	0.02 +	0.02 +	0.21 -	0.05 +	0.02 +
cg	0.42	0.13 +	0.38	0.37	0.83 -	0.50	0.38
ct	3.78	3.75	3.78	3.43 +	7.91 -	3.78	2.90 +
he	0.01	0.01	0.01	0.01	0.02	0.02	0.01
li	0.03	0.03	0.03	0.03	0.05 -	0.04 -	0.03
lc	0.01	0.01	0.01	0.01	0.01	0.03 -	0.01
ma	0.13	0.06 +	0.05 +	0.05 +	0.29 -	0.07 +	0.05 +
m1	0.04	0.04	0.04	0.04	0.07 -	0.05 -	0.04
m2	0.05	0.05	0.05	0.05	0.10 -	0.05	0.05
m3	0.03	0.01 +	0.01 +	0.01 +	0.06 -	0.02	0.01 +
mu	3.72	1.24 +	0.49 +	0.48 +	31.05 -	1.51 +	0.77 +
pa	0.01	0.01	0.01	0.01	0.03 -	0.03 -	0.01
pi	0.15	0.06 +	0.10 +	0.06 +	0.21 -	0.12 +	0.06 +
se	3.72	3.74	3.73	3.74	6.15 -	3.74	3.74
so	0.89	0.88	0.91	0.91	1.28 -	0.96	0.85
sh	0.03	0.01 +	0.02	0.01 +	0.04 -	0.03	0.01 +
vo	0.02	0.01	0.01	0.01	0.07 -	0.03 -	0.01 +

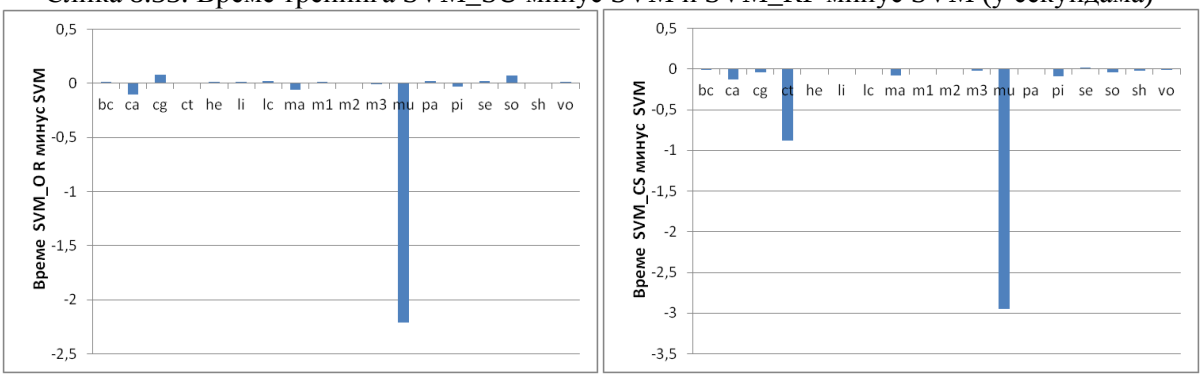
На сликама 8.32, 8.33. и 8.34. приказана је апсолутна разлика у потребном времену за тренинг SVM алгоритма на основном скупу података и SVM алгоритма са различитим методама филтрирања. Примењени метод филтрирања IG је у 10 скупова података показао нешто боље резултате за потребно време за тренинг, а у 7 скупова података резултати су били статистички бољи. Метод филтрирања GR је у 9 скупова података показао нешто боље резултате за потребно време за тренинг, а у 5 скупова података резултати су били статистички бољи.



Слика 8.32: Време тренинга SVM_IG минус SVM и SVM_GR минус SVM (у секундама)



Слика 8.33: Време тренинга SVM_SU минус SVM и SVM_RF минус SVM (у секундама)



Слика 8.34: Време тренинга SVM_OR минус SVM и SVM_CS минус SVM (у секундама)

Примењени метод филтрирања SU је у више од пола скупова података (10 скупова) показао боље резултате за потребно време за тренинг од SVM алгоритма на основном скупу података, а у 8 скупова података резултати су били и статистички бољи. Метод филтрирања RF је у свим скуповима података показао лошије или исте резултате за потребно време за тренинг од SVM алгоритма на основном скупу података, а у скоро свим скуповима података резултати су били и статистички лошији.

Метод филтрирања OR је у пет скупова података показао боље резултате од SVM алгоритма на основном скупу података, а у четири случаја резултати су били и статистички бољи. Примењени метод филтрирања CS је у 11 скупова података показао боље резултате од SVM алгоритма на основном скупу података, а у 8 скупа података резултати су били и статистички бољи.

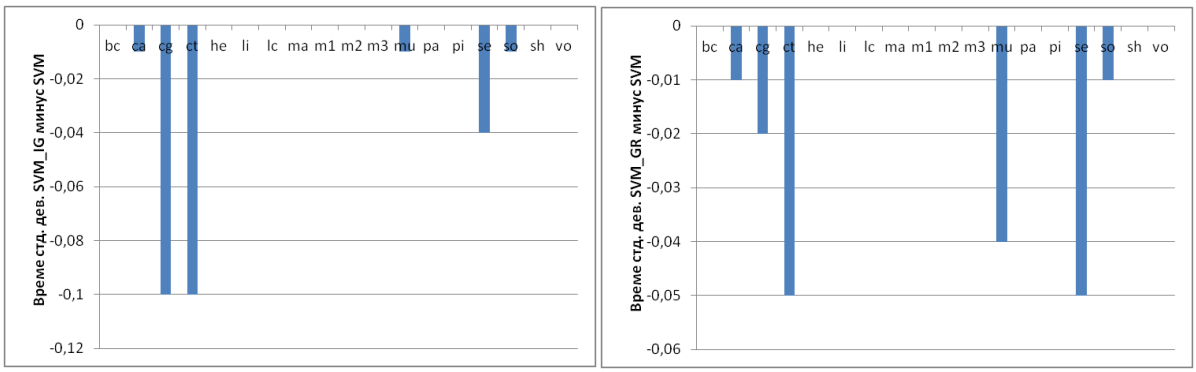
Коришћењем SVM класификатора, можемо да закључимо да су SU и CS методе филтрирања у највећем броју случаја довели до статистички бољих резултата за потребно време за тренинг на посматраним скуповима података.

Табела 8.13. приказује стандардну девијацију за време тренинга SVM алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Код GR и IG методе филтрирања вредности за стандардну девијацију за време тренинга су исте или мање у односу на основни алгоритам учења.

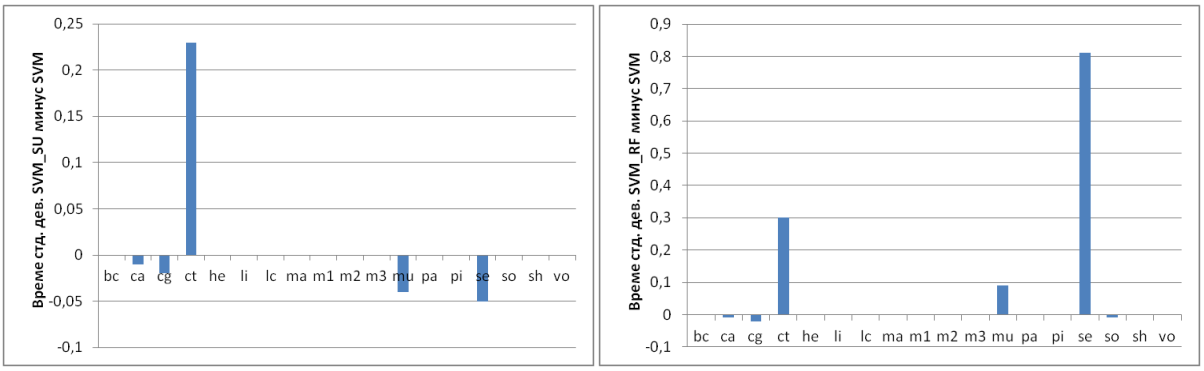
Табела 8.13. Стандардна девијација за време тренинга (у секундама) SVM алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	SVM	SVM_IG	SVM_GR	SVM _SU	SVM _RF	SVM_OR	SVM _CS
bc	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ca	0.02	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.11	0.01	0.09	0.09	0.09	0.09	0.09
ct	0.13	0.03	0.08	0.36	0.43	0.19	0.40
he	0.01	0.01	0.01	0.01	0.01	0.01	0.01
li	0.01	0.01	0.01	0.01	0.01	0.01	0.01
lc	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ma	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m2	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m3	0.01	0.01	0.01	0.01	0.01	0.01	0.01
mu	0.07	0.06	0.03	0.03	0.16	0.05	0.06
pa	0.01	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.01	0.01	0.01	0.01	0.01	0.01	0.01
se	0.08	0.04	0.03	0.03	0.89	0.05	0.04
so	0.10	0.09	0.09	0.10	0.09	0.09	0.10
sh	0.01	0.01	0.01	0.01	0.01	0.01	0.01
vo	0.01	0.01	0.01	0.01	0.01	0.01	0.01

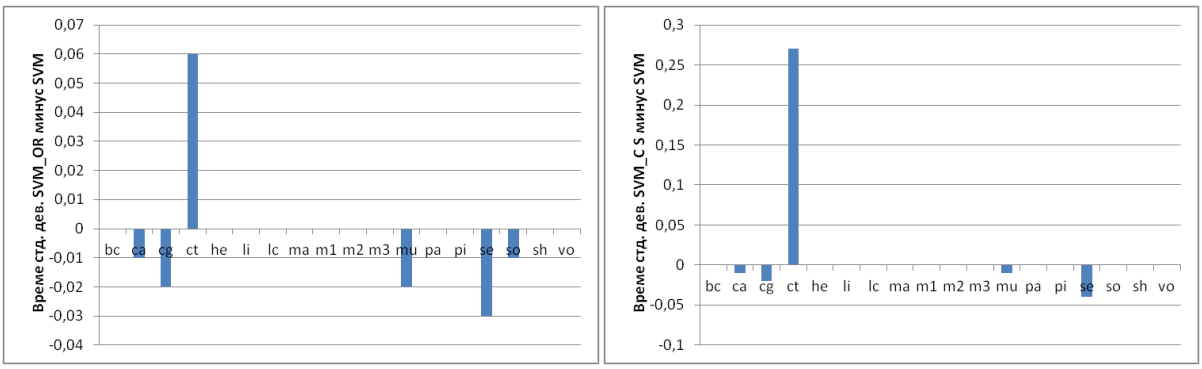
На сликама 8.35, 8.36. и 8.37. приказана је апсолутна разлика у вредностима стандардне девијације за време тренинга SVM алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, то је и веће одступање између стандардних девијација. Највеће одступање у стандардној девијацији у односу на оригинални скуп података показује метода RF.



Слика 8.35: Стандардна девијација за време SVM_IG минус SVM и SVM_GR минус SVM



Слика 8.36: Стандардна девијација за време SVM_SU минус SVM и SVM_RF минус SVM



Слика 8.37: Стандардна девијација за време SVM_OR минус SVM и SVM_CS минус SVM

8.5. J48

На основу приказаних података у табели 8.14. за тачност класификације J48 алгоритма можемо уочити да у четири сета података (*cg*, *ma*, *m2* и *sh*) имамо добијене резултате за бар једну од метода филтрирања који су статистички бољи од основног класификатора. У свим сетовима података осим једног сета података *m3*, немамо значајно лошије податке за све методе филтрирања, што значи да увек можемо изабрати методу за дати скуп података која има статистички боље резултате или резултате који су приближни оригиналном скупу података. Код само једног скупа података *m3* све методе филтрирања су статистички лошије од основног класификатора.

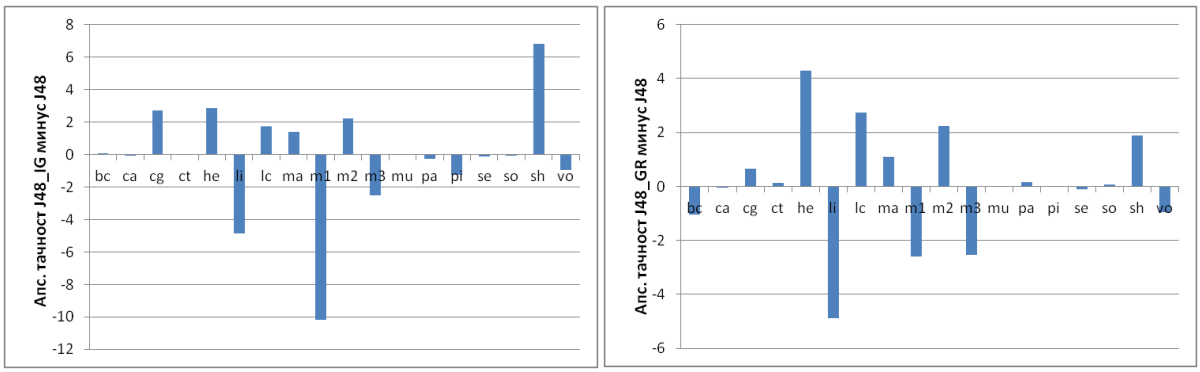
Слике 8.38, 8.39. и 8.40. приказују апсолутну разлику у тачности класификације J48 алгоритма на основном скупу података и J48 алгоритма са различитим методама филтрирања. Примењени метод филтрирања IG је у пола скупова података (9 скупова) показао исте или боље резултате од J48 алгоритма на основном скупу података, док су у 3 скупа података резултати и статистички бољи. Метод филтрирања GR је у више од пола скупова података (11 скупова) показао исте или боље резултате од J48 алгоритма

на основном скупу података, а у ни у једном скупу података резултат није био статистички бољи.

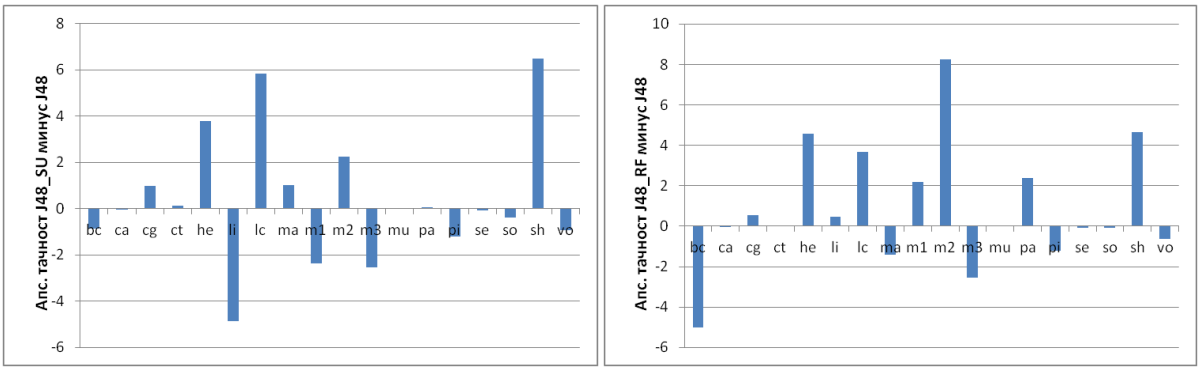
Табела 8.14. Тачност класификације J48 алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	J48	J48_IG	J48_GR	J48_SU	J48_RF	J48_OR	J48_CS
bc	74.28	74.35	73.23	73.40	69.27 -	73.10	72.84
ca	85.57	85.51	85.51	85.51	85.51	85.51	85.51
cg	71.25	73.95 +	71.91	72.22	71.78	71.49	71.94
ct	98.57	98.57	98.69	98.70	98.57	98.63	98.92
he	79.22	82.10	83.51	83.00	83.78	83.96	81.92
li	65.84	60.97	60.97	60.97	66.32	64.47	60.97
lc	79.25	81.00	82.00	85.08	82.92	80.58	81.33
ma	82.19	83.57 +	83.29	83.19	80.76	82.60	83.16
m1	97.80	87.63 -	95.21	95.43	100.00	87.63 -	89.99
m2	63.48	65.72	65.72	65.72	71.74 +	71.93 +	65.72
m3	98.92	96.39 -	96.39 -	96.39 -	96.39 -	96.39 -	96.39 -
mu	100.00	100.00	100.00	100.00	100.00	100.00	100.00
pa	84.74	84.49	84.90	84.79	87.14	84.48	84.74
pi	74.49	73.27	74.49	73.27	73.27	73.63	73.27
se	96.79	96.69	96.69	96.70	96.71	96.67	96.69
so	91.78	91.70	91.84	91.38	91.70	91.70	91.71
sh	78.15	84.96 +	80.04	84.63 +	82.81 +	85.07 +	85.22 +
vo	96.57	95.63	95.63	95.63	95.93	95.63	95.63

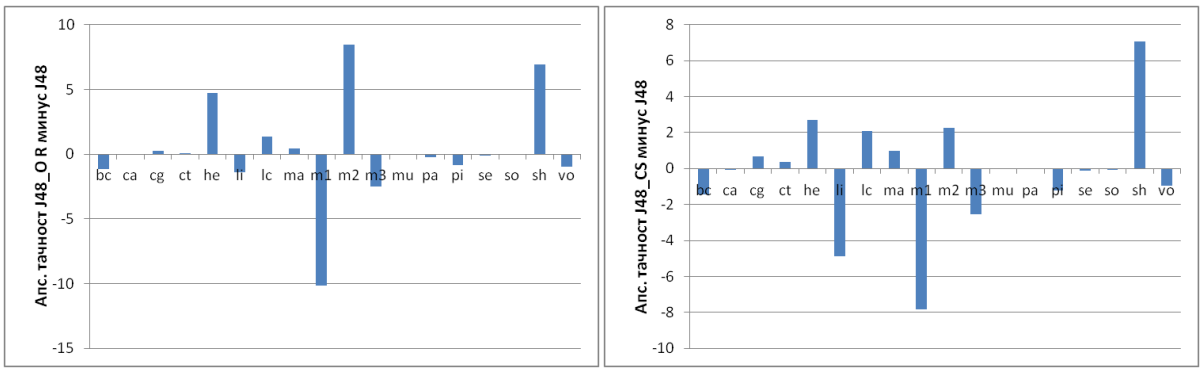
Примењени метод филтрирања SU је у пола скупова података (9 скупова) показао исте или боље резултате од J48 алгоритма на основном скупу података, док је у једном скупу података резултат био и статистички бољи. Метод филтрирања RF је у више од пола скупова података (10 скупова) показао исте или боље резултате од J48 алгоритма на основном скупу података, док је у 2 скупа података резултат био и статистички бољи.



Слика 8.38: Апсолутна тачност класификације J48_IG минус J48 и J48_GR минус J48



Слика 8.39: Апсолутна тачност класификације J48_SU минус J48 и J48_RF минус J48



Слика 8.40: Апсолутна тачност класификације J48_OR минус J48 и J48_CS минус J48

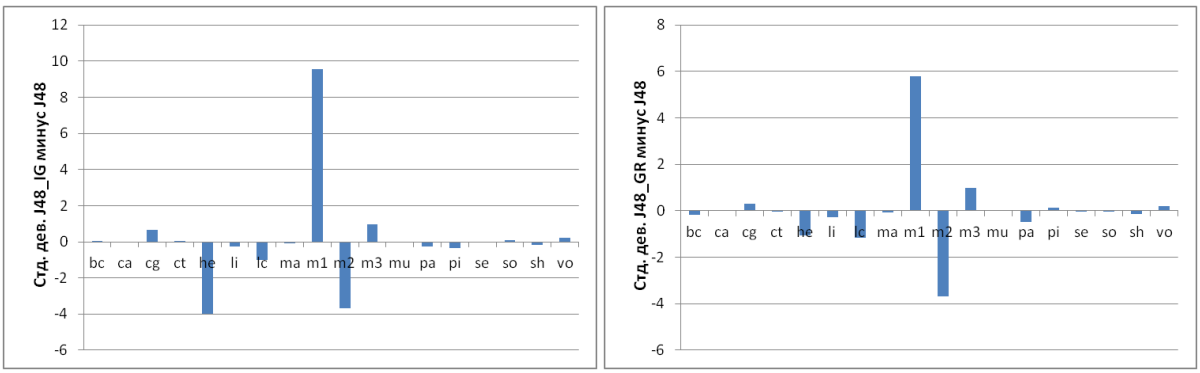
Табела 8.15. Стандардна девијација за тачност класификације J48 алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	J48	J48_IG	J48_GR	J48_SU	J48_RF	J48_OR	J48_CS
bc	6.05	6.07	5.88	5.43	5.22	5.55	5.55
ca	3.96	3.96	3.96	3.96	3.96	3.96	3.96
cg	3.17	3.83	3.48	3.42	3.40	3.50	3.48
ct	0.89	0.92	0.86	0.91	0.89	0.88	0.77
he	9.57	5.60	8.50	7.60	8.12	8.04	5.93
li	7.40	7.12	7.12	7.12	8.24	7.91	7.12
lc	21.50	20.48	20.33	16.76	17.10	21.02	19.96
ma	3.21	3.14	3.13	3.09	3.62	3.11	3.07
m1	3.45	13.03	9.25	9.10	0.00	13.03	12.07
m2	4.48	0.79	0.79	0.79	5.34	5.39	0.79
m3	1.23	2.20	2.20	2.20	2.20	2.20	2.20
mu	0.00	0.00	0.00	0.00	0.00	0.00	0.00
pa	8.01	7.73	7.54	7.57	7.31	7.62	7.74
pi	5.27	4.93	5.41	4.93	4.93	5.57	4.93
se	1.29	1.28	1.28	1.27	1.33	1.28	1.27
so	3.19	3.30	3.17	3.19	3.36	3.17	3.04
sh	7.42	7.23	7.29	7.21	6.95	6.67	6.75
vo	2.56	2.76	2.76	2.76	2.71	2.76	2.76

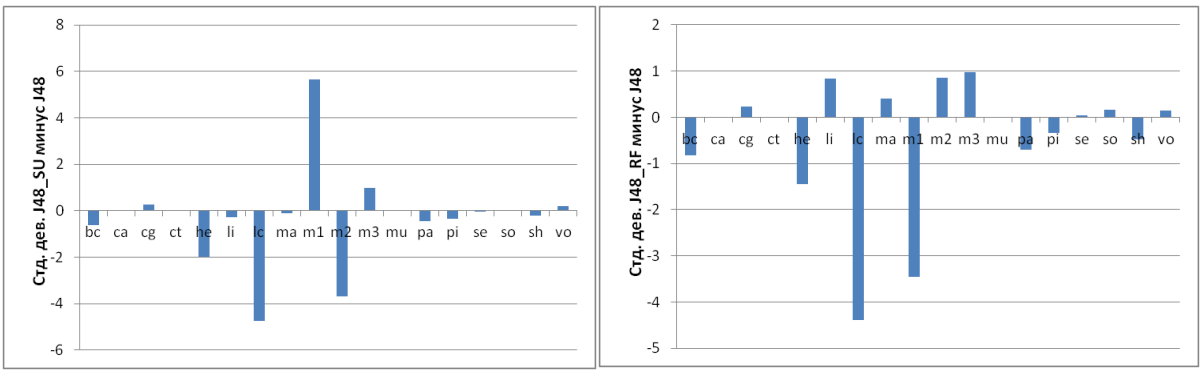
Метод филтрирања OR је у нешто мање од пола скупова података (8 скупова) показао исте или боље резултате од J48 алгоритма на основном скупу података, а у 2 скупа података резултати су били и статистички бољи. Примењени метод филтрирања CS је у пола скупова података (9 скупова) показао исте или боље резултате од J48 алгоритма на основном скупу података, а у 1 скупу података резултат је био и статистички бољи.

Коришћењем J48 класификатора, можемо да закључимо да је IG метода филтрирања у највећем броју случаја довела до статистички бољих резултата на посматраним скуповима података.

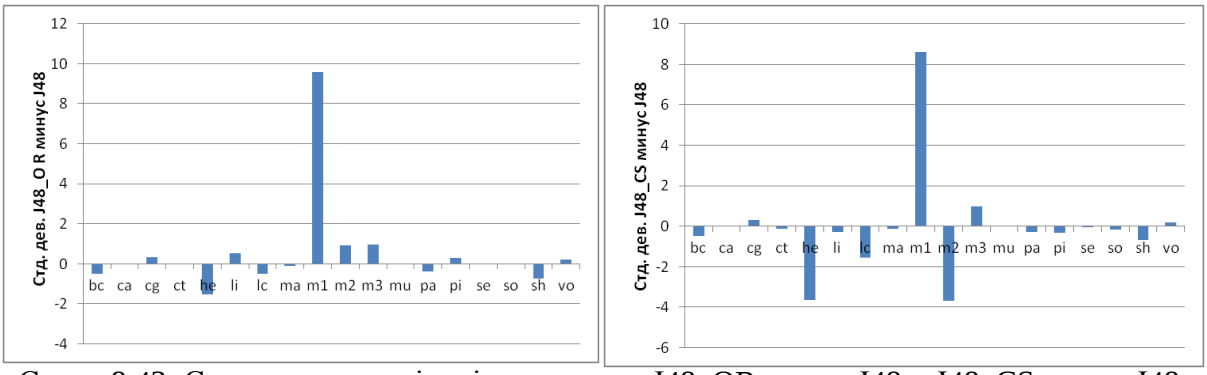
На сликама 8.41, 8.42. и 8.43. приказана је апсолутна разлика у вредностима стандардне девијације за тачност класификације J48 алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликују, а уколико она више одступа од нуле, у позитивном и негативном смеру, то је и веће одступање између стандардних девијација. Најмање одступање у стандардној девијацији у односу на оригинални скуп података, показује метода RF, док највеће одступање имају методе IG, OR и CS, које код неких скупова података су успеле да смање, а код неких да повећају стандардну девијацију.



Слика 8.41: Стандардна девијација за тачност J48_IG минус J48 и J48_GR минус J48



Слика 8.42: Стандардна девијација за тачност J48_SU минус J48 и J48_RF минус J48



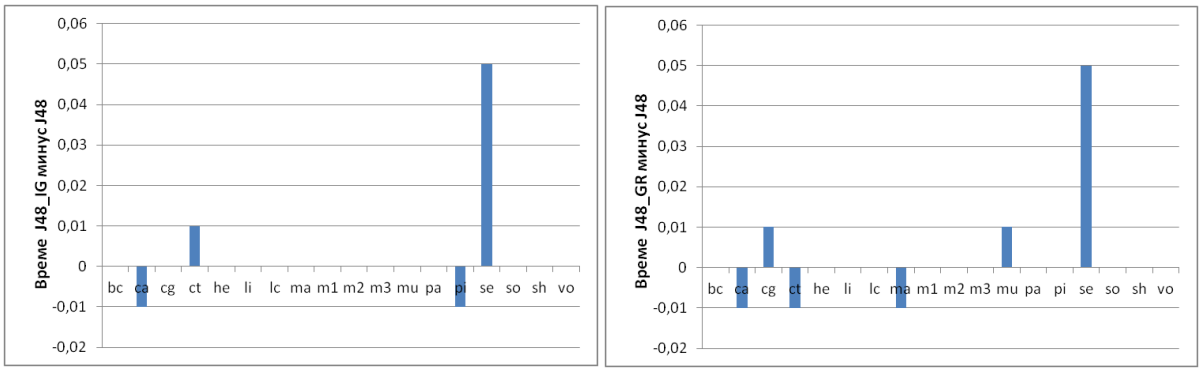
Слика 8.43: Стандардна девијација за тачност J48_OR минус J48 и J48_CS минус J48

Табела 8.16. Потребно време за тренинг (у секундама) J48 алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода

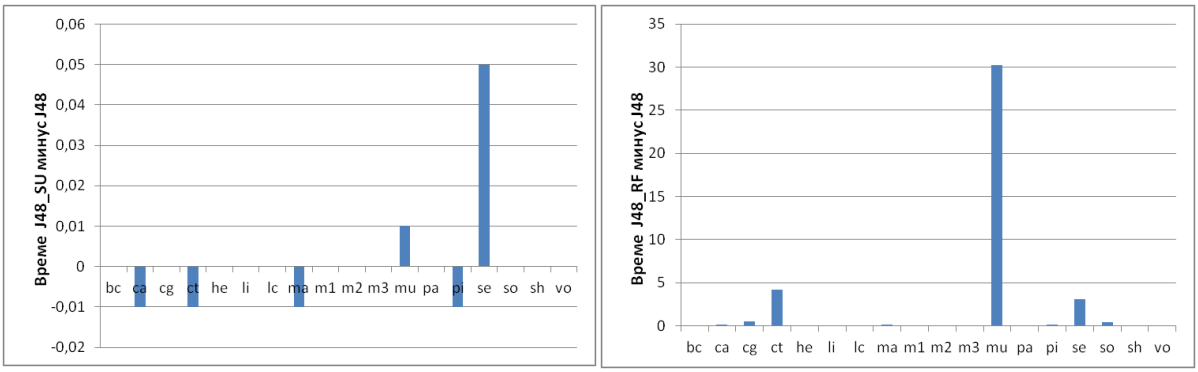
Скуп	J48	J48_IG	J48_GR	J48_SU	J48_RF	J48_OR	J48 _CS
bc	0.00	0.00	0.00	0.00	0.02 -	0.01 -	0.00
ca	0.01	0.00	0.00	0.00	0.19 -	0.03 -	0.00
cg	0.01	0.01 +	0.02	0.01	0.49 -	0.07 -	0.01
ct	0.11	0.12 -	0.10 +	0.10 +	4.33 -	0.34 -	0.07 +
he	0.00	0.00	0.00	0.00	0.01 -	0.01 -	0.00
li	0.00	0.00	0.00	0.00	0.03 -	0.01	0.00
lc	0.00	0.00	0.00	0.00	0.00	0.02 -	0.00
ma	0.01	0.01	0.00	0.00	0.18 -	0.02 -	0.00
m1	0.00	0.00	0.00	0.00	0.00	0.05 -	0.01 -
m2	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m3	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
mu	0.03	0.03	0.04	0.04	30.29 -	0.78 -	0.03
pa	0.01	0.01	0.01	0.01	0.03 -	0.03 -	0.01
pi	0.01	0.00	0.01	0.00	0.16 -	0.03 -	0.00
se	0.09	0.14 -	0.14 -	0.14 -	3.21 -	0.27 -	0.14 -
so	0.01	0.01	0.01	0.01	0.43 -	0.08 -	0.01
sh	0.00	0.00	0.00	0.00	0.03 -	0.01 -	0.00
vo	0.00	0.00	0.00	0.00	0.06 -	0.02 -	0.00

С обзиром на тврдњу коју смо изнели да је добар онај алгоритам који даје сличан резултат у свим случајевима, односно вредност за стандардну девијацију је минимална, разматраћемо стандардну девијацију за тачност класификације за J48 алгоритам. Табела 8.15. приказује стандардну девијацију за тачност класификације J48 алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута, осим у случају *m1* скупа података. У случају *m1* скупа података за све методе филтрирања

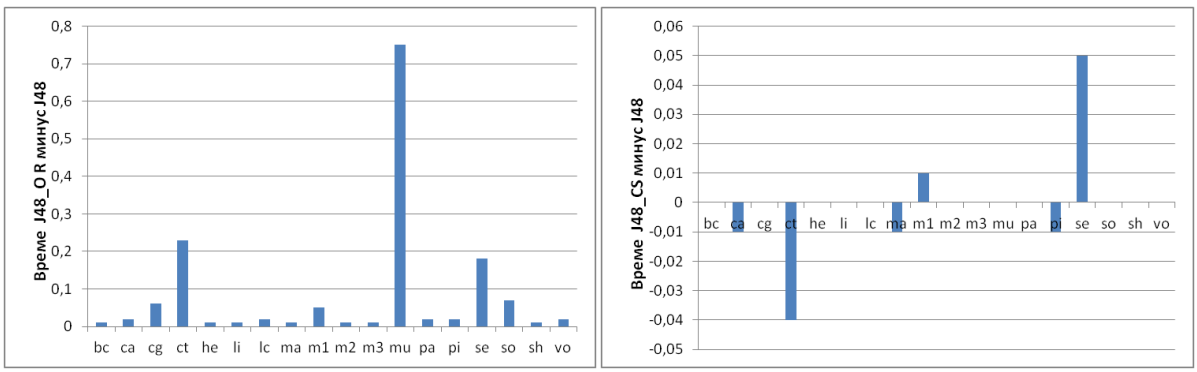
добијамо велику вредност за стандардну девијацију, осим методе RF. Најмања одступања у стандардној девијацији показује метода RF, у односу на друге методе, где је за поједине скупове података успела да добије како мање, тако и веће вредности за стандардну девијацију.



Слика 8.44: Време тренинга J48_IG минус J48 и J48_GR минус J48 (у секундама)



Слика 8.45: Време тренинга J48_SU минус J48 и J48_RF минус J48 (у секундама)



Слика 8.46. Време тренинга J48_OR минус J48 и J48_CS минус J48 (у секундама)

Табела 8.16. приказује потребно време за тренинг J48 алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода. Потребно време за тренинг података J48 класификатора за све оригиналне скупове података износи највише 0.11, док за филтер методе оно у неким случајевима веће, а у неким мање. За само један скуп података (*se*), ни једна од метода не даје исто или мање потребно време за тренинг података, док код свих осталих скупова података бар једна од метода

филтрирања даје исто или мање време за тренинг података као код оригиналног скупа података.

На сликама 8.44, 8.45. и 8.46. приказана је апсолутна разлика у потребном времену за тренинг J48 алгоритма на основном скупу података и J48 алгоритма са различитим методама филтрирања. Метод филтрирања IG је у само два скупа података показао нешто лошије резултате за потребно време за тренинг и код тих скупова података, резултати су били и статистички лошији. Метод филтрирања GR је у само три скупа података показао нешто лошије резултате за потребно време за тренинг и код 1 скупа података резултат је био и статистички лошији.

Примењени метод филтрирања SU је само у 2 скупа података показао лошије резултате за потребно време за тренинг од J48 алгоритма на основном скупу података, а у 1 скупу података резултат је био и статистички лошији. Метод филтрирања RF је у 16 скупова података показао лошије резултате за потребно време за тренинг од J48 алгоритма на основном скупу података, а у скоро свим скуповима података резултати су били и статистички лошији.

Метод филтрирања OR је у свим скуповима података показао лошије резултате од J48 алгоритма на основном скупу података, а ови резултати су били у готово свим случајевима и статистички лошији. Примењени метод филтрирања CS је у само 2 скупа података показао лошије резултате од J48 алгоритма на основном скупу података, а у 2 скупа података резултати су били и статистички лошији.

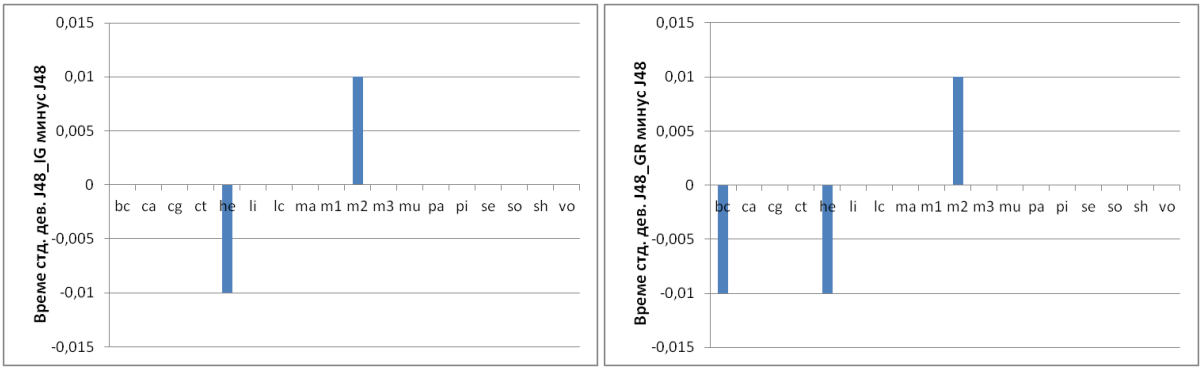
Коришћењем J48 класификатора, можемо да закључимо да су GR и SU методе филтрирања у најмањем броју случаја довеле до статистички лошијих резултата за потребно време за тренинг на посматраним скуповима података.

Табела 8.17. приказује стандардну девијацију за време тренинга J48 алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута. Нешто веће вредности за стандардну девијацију за време тренинга има RF метода филтрирања.

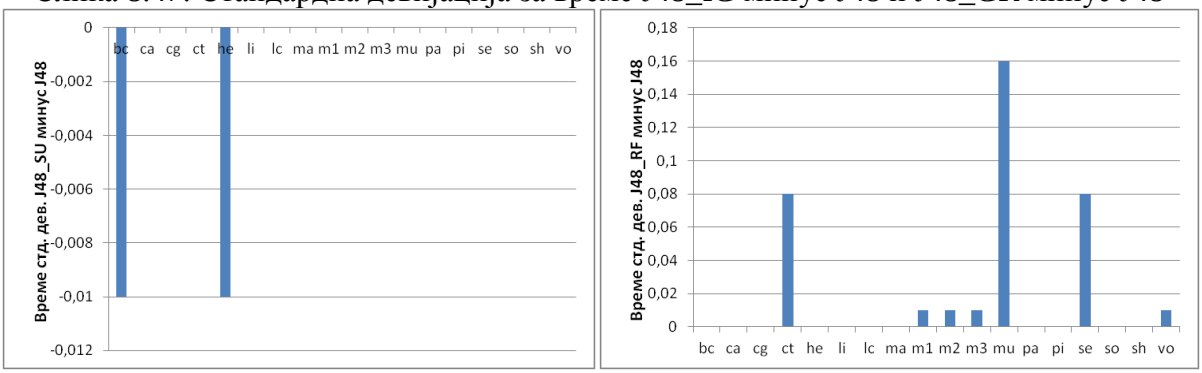
На сликама 8.47, 8.48. и 8.49. приказана је апсолутна разлика у вредностима стандардне девијације за време тренинга J48 алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Метода SU за све скупове података има мање или исте вредности за стандардну девијацију за време тренинга у односу на оригинални скуп података.

Табела 8.17. Стандардна девијација за време тренинга (у секундама) J48 алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

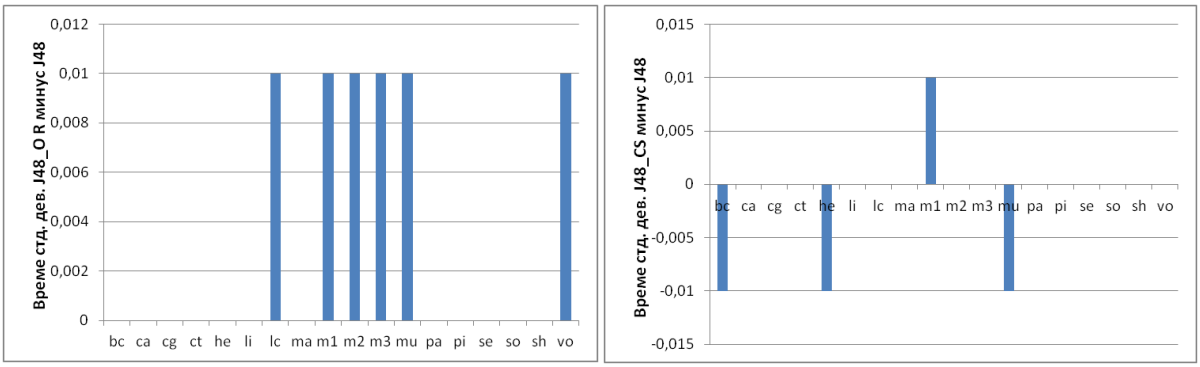
Скуп	J48	J48_IG	J48_GR	J48_SU	J48_RF	J48_OR	J48_CS
bc	0.01	0.01	0.00	0.00	0.01	0.01	0.00
ca	0.01	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ct	0.01	0.01	0.01	0.01	0.09	0.01	0.01
he	0.01	0.00	0.00	0.00	0.01	0.01	0.00
li	0.01	0.01	0.01	0.01	0.01	0.01	0.01
lc	0.00	0.00	0.00	0.00	0.00	0.01	0.00
ma	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.00	0.00	0.00	0.00	0.01	0.01	0.01
m2	0.00	0.01	0.01	0.00	0.01	0.01	0.00
m3	0.00	0.00	0.00	0.00	0.01	0.01	0.00
mu	0.01	0.01	0.01	0.01	0.17	0.02	0.00
pa	0.01	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.01	0.01	0.01	0.01	0.01	0.01	0.01
se	0.01	0.01	0.01	0.01	0.09	0.01	0.01
so	0.01	0.01	0.01	0.01	0.01	0.01	0.01
sh	0.01	0.01	0.01	0.01	0.01	0.01	0.01
vo	0.00	0.00	0.00	0.00	0.01	0.01	0.00



Слика 8.47: Стандардна девијација за време J48_IG минус J48 и J48_GR минус J48



Слика 8.48: Стандардна девијација за време J48_SU минус J48 и J48_RF минус J48



Слика 8.49: Стандардна девијација за време J48_OR минус J48 и J48_CS минус J48

8.6. RBF мреже

За тачност класификације код RBF алгоритма можемо уочити на основу табеле 8.18. да у три сета података (*ca*, *m1* и *se*) имамо добијене резултате за бар једну од метода филтрирања који су статистички бољи од основног класификатора.

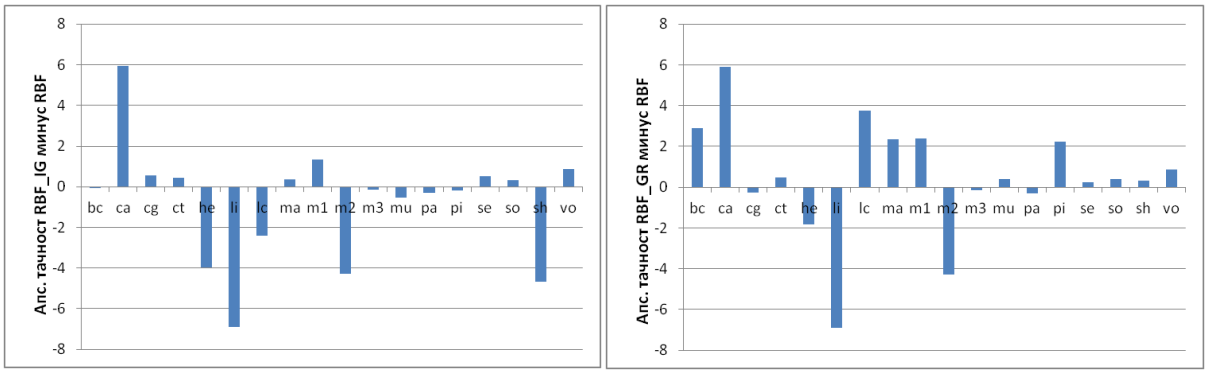
Табела 8.18. Тачност класификације RBF алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	RBF	RBF_IG	RBF_GR	RBF_SU	RBF_RF	RBF_OR	RBF_CS
bc	71.41	71.34	74.32	74.46	71.00	71.20	73.62
ca	79.55	85.51 +	85.43 +	85.51 +	85.51 +	85.10 +	85.51 +
cg	73.58	74.12	73.33	73.64	73.54	73.16	73.54
ct	97.93	98.35	98.41	97.65	98.13	96.90	96.27 -
he	85.29	81.31	83.45	83.05	80.49	82.69	81.25
li	65.06	58.16 -	58.16 -	58.16 -	57.33 -	60.96	58.16 -
lc	76.00	73.58	79.75	79.00	76.75	72.92	74.92
ma	77.31	77.66	79.67	79.24	77.07	77.51	79.16
m1	75.36	76.70	77.76	77.76	90.01 +	75.37	76.70
m2	67.82	63.53	63.54	63.53	64.77	64.77	63.53
m3	96.54	96.39	96.39	96.39	96.39	96.39	96.39
mu	98.61	98.06	98.99	98.99	98.43	98.55	98.55
pa	81.22	80.92	80.92	80.92	83.39	81.98	80.67
pi	74.04	73.84	76.28	73.84	73.84	75.32	73.84
se	87.31	87.84	87.56	87.84	88.88 +	87.84	87.84
so	90.79	91.11	91.20	91.59	91.29	90.57	91.42
sh	83.11	78.44	83.44	78.15	81.56	78.44	78.52
vo	93.73	94.60	94.60	94.60	94.92	95.63	95.63

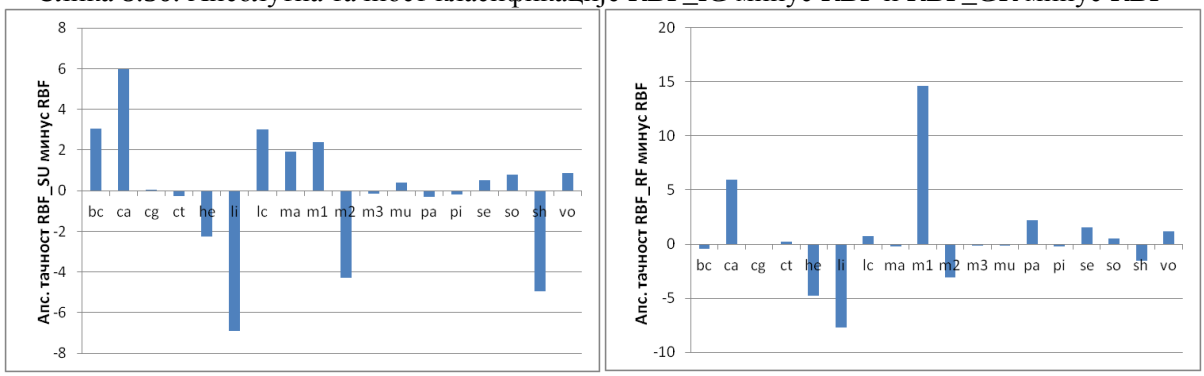
Ни у једном сету података, немамо значајно лошије податке за све методе филтрирања, што значи да увек можемо изабрати методу за дати скуп података која има статистички боље резултате или резултате који су приближни оригиналном скупу

података. Код једног скупа података (*ca*) све методе филтрирања су статистички боље од основног класификатора.

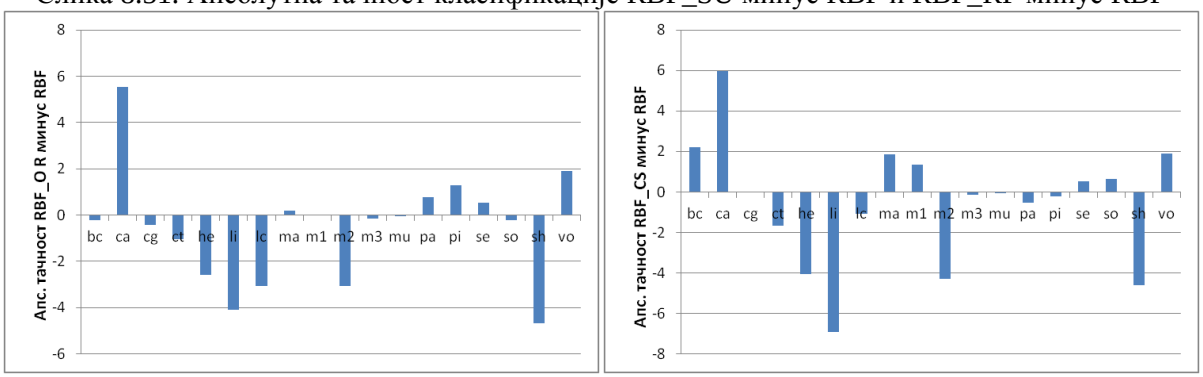
На сликама 8.50, 8.51. и 8.52. приказана је апсолутна разлика у тачности класификације RBF алгоритма на основном скупу података и RBF алгоритма са различитим методама филтрирања. Примењени метод филтрирања IG је у скоро пола скупова података (8 скупова) показао исте или боље резултате од RBF алгоритма на основном скупу података, док у 1 скупу података резултат је био и статистички бољи. Метод филтрирања GR је у две трећине скупова података (12 скупова) показао исте или боље резултате од RBF алгоритма на основном скупу података, док у само 1 скупу података резултат је био и статистички бољи.



Слика 8.50: Апсолутна тачност класификације RBF_IG минус RBF и RBF_GR минус RBF



Слика 8.51: Апсолутна тачност класификације RBF_SU минус RBF и RBF_RF минус RBF



Слика 8.52: Апсолутна тачност класификације RBF_OR минус RBF и RBF_CS минус RBF

Примењени метод филтрирања SU је у више од пола скупова података (10 скупова) показао исте или боље резултате од RBF алгоритма на основном скупу података, док у 1 скупу података резултат је био и статистички бољи. Метод филтрирања RF је у мање од пола скупова података (8 скупова) показао исте или боље резултате од RBF алгоритма на основном скупу података, док у 3 скупа података резултати су били и статистички бољи.

Метод филтрирања OR је у нешто мање од пола скупова података (7 скупова) показао исте или боље резултате од RBF алгоритма на основном скупу података, а у 1 скупу података резултат је био и статистички бољи. Примењени метод филтрирања CS је у нешто мање од пола скупова података (7 скупова) показао исте или боље резултате од RBF алгоритма на основном скупу података, а у 1 скупу података, резултат је био и статистички бољи.

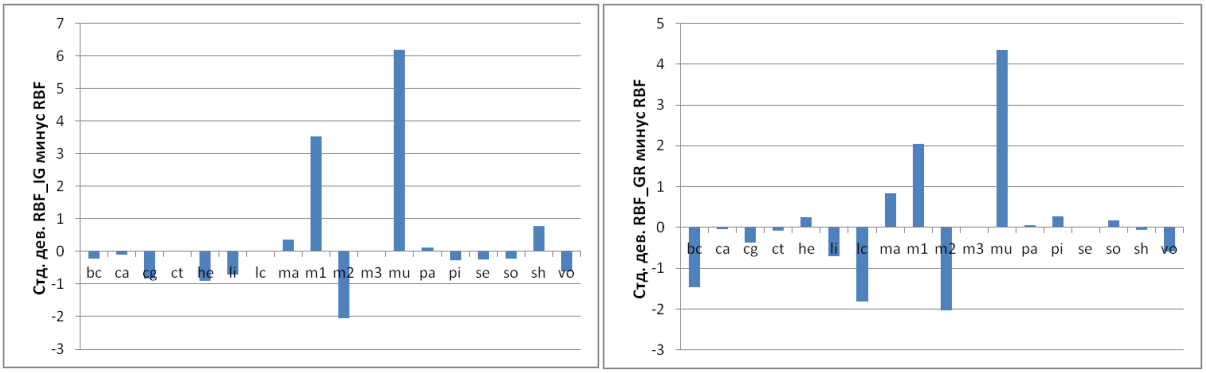
Коришћењем RBF класификатора, можемо да закључимо да је RF метода филтрирања у највећем броју случаја довела до статистички бољих резултата на посматраним скуповима података.

Табела 8.19. Стандардна девијација за тачност класификације RBF алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

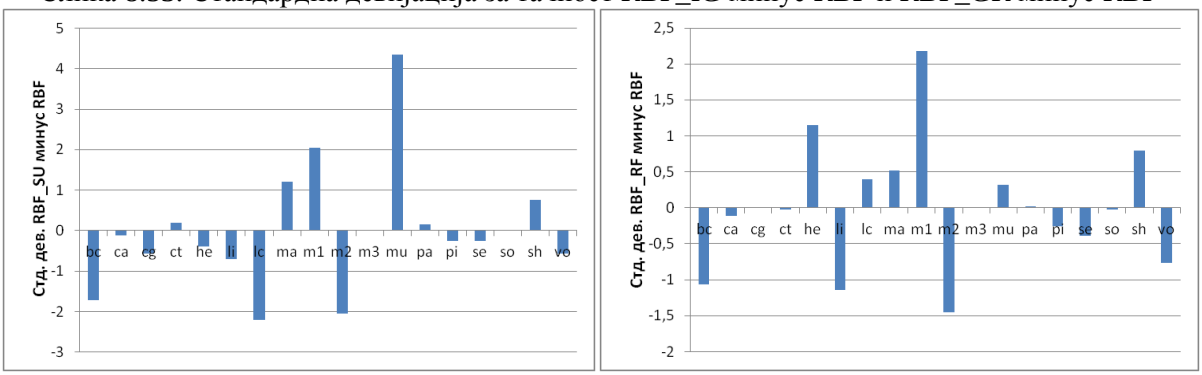
Скуп	RBF	RBF_IG	RBF_GR	RBF_SU	RBF_RF	RBF_OR	RBF_CS
bc	7.88	7.66	6.41	6.16	6.81	8.34	6.28
ca	4.07	3.96	4.03	3.96	3.96	4.14	3.96
cg	4.30	3.46	3.92	3.74	4.31	4.25	4.03
ct	1.02	1.02	0.94	1.21	1.00	2.19	1.29
he	8.29	7.38	8.54	7.90	9.44	8.25	7.51
li	8.80	8.10	8.10	8.10	7.66	9.62	8.10
lc	22.91	22.91	21.10	20.70	23.31	22.17	22.52
ma	3.31	3.67	4.14	4.51	3.83	4.35	4.50
m1	5.92	9.44	7.97	7.97	8.10	7.97	9.44
m2	6.24	4.19	4.21	4.19	4.79	4.79	4.19
m3	2.19	2.20	2.20	2.20	2.20	2.20	2.20
mu	0.58	6.77	4.93	4.93	0.90	4.86	4.86
pa	7.37	7.49	7.42	7.53	7.39	7.24	7.35
pi	4.91	4.65	5.18	4.65	4.65	5.31	4.65
se	2.15	1.91	2.15	1.89	1.76	1.91	1.90
so	2.92	2.69	3.09	2.93	2.90	2.92	3.24
sh	6.50	7.28	6.44	7.25	7.29	7.13	7.28
vo	3.87	3.25	3.28	3.30	3.10	2.76	2.76

Табела 8.19. приказује стандардну девијацију за тачност класификације RBF алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе предселекцију атрибута.

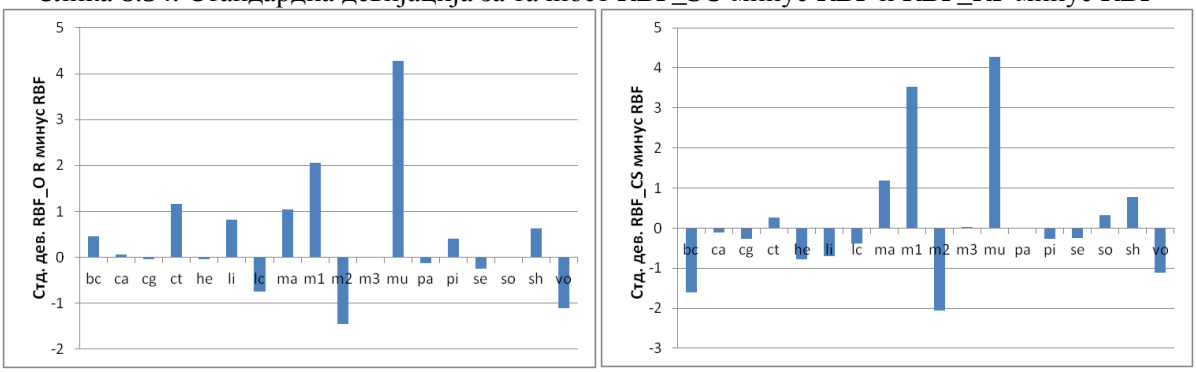
На сликама 8.53, 8.54. и 8.55. приказана је апсолутна разлика у вредностима стандардне девијације за тачност класификације RBF алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, то је и веће одступање између стандардних девијација. Најмање одступање у стандардној девијацији показује RF метода, тако што за поједине скупове података стандардна девијација има мању, али у неким случајевима и већу вредност.



Слика 8.53: Стандардна девијација за тачност RBF_IG минус RBF и RBF_GR минус RBF



Слика 8.54: Стандардна девијација за тачност RBF_SU минус RBF и RBF_RF минус RBF



Слика 8.55: Стандардна девијација за тачност RBF_OR минус RBF и RBF_CS минус RBF

Потребно време за тренинг RBF алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода приказано је у табели 8.20. Потребно време за тренинг података RBF класификатора за све оригиналне скупове података износи испод 1.00 секунде, осим за два сета података *se* и *so*, код којих је значајно веће. Потребно време за тренинг је код неких метода филтрирања веће, а код неких је мање у односу на оригинални скуп података. Код свих скупова података, бар једна од метода филтрирања даје исте или боље резултате за време потребно за тренирање у односу на оригинални скуп.

Табела 8.20. Потребно време за тренинг (у секундама) RBF алгоритма који користи оригинални и редуковани скуп података уз помоћ филтер метода

Скуп	RBF	RBF_IG	RBF_GR	RBF_SU	RBF_RF	RBF_OR	RBF_CS
bc	0.01	0.01	0.01	0.00	0.02 -	0.02	0.01
ca	0.03	0.01 +	0.01 +	0.01 +	0.20 -	0.04 -	0.01 +
cg	0.05	0.02 +	0.04	0.04	0.53 -	0.10 -	0.04
ct	0.39	0.44	0.34	0.34	4.57 -	0.55 -	0.34
he	0.01	0.00	0.00	0.00	0.01	0.02 -	0.00
li	0.01	0.01	0.01	0.01	0.03 -	0.02	0.01
lc	0.00	0.00	0.00	0.00	0.00	0.02 -	0.00
ma	0.02	0.02	0.02	0.02	0.19 -	0.03 -	0.02
m1	0.01	0.01	0.01	0.01	0.06 -	0.02	0.01
m2	0.01	0.01	0.01	0.01	0.06 -	0.02	0.01
m3	0.01	0.01	0.01	0.01	0.06 -	0.02	0.01
mu	0.49	0.33 +	0.34 +	0.33 +	30.38 -	1.08 -	0.33 +
pa	0.02	0.02	0.02	0.02	0.04 -	0.04 -	0.02
pi	0.03	0.01 +	0.02 +	0.01 +	0.17 -	0.04 -	0.01 +
se	4.04	4.09	3.77	4.20	7.41 -	3.96	4.28
so	248.83	214.28	238.50	242.58	266.20	249.31	248.23
sh	0.01	0.01	0.01	0.01	0.04 -	0.02	0.01
vo	0.01	0.01	0.01	0.01	0.07 -	0.03 -	0.01

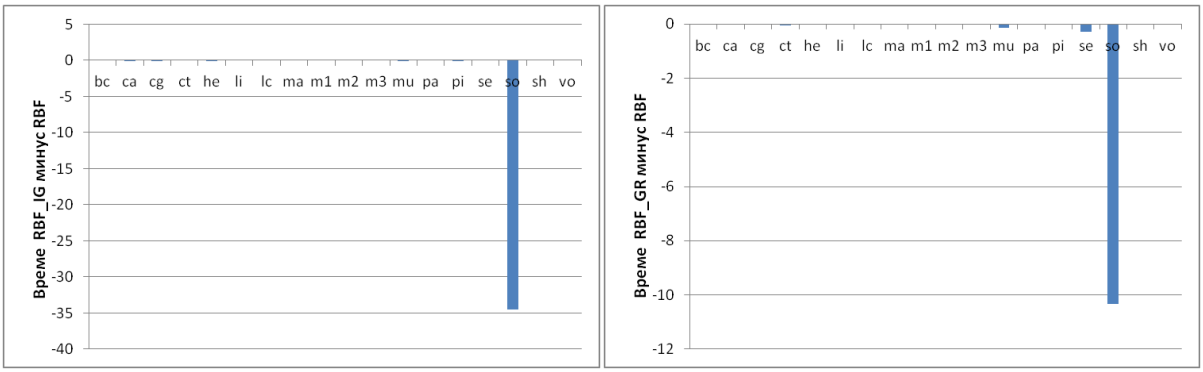
Слике 8.56, 8.57. и 8.58. приказују апсолутну разлику у потребном времену за тренинг RBF алгоритма на основном скупу података и RBF алгоритма са различитим методама филтрирања. Примењени метод филтрирања IG и GR није ни у једном скупу података показао лошије резултате за потребно време за тренинг; код 4, односно 3 скупа података респективно, резултати су били и статистички бољи.

Примењени метод филтрирања SU је само у једном скупу података показао лошије резултате за потребно време за тренинг од RBF алгоритма на основном скупу података, а у 3 скупа података резултати су били и статистички бољи. Метод филтрирања RF је у свим скуповима података показао исте или лошије резултате за

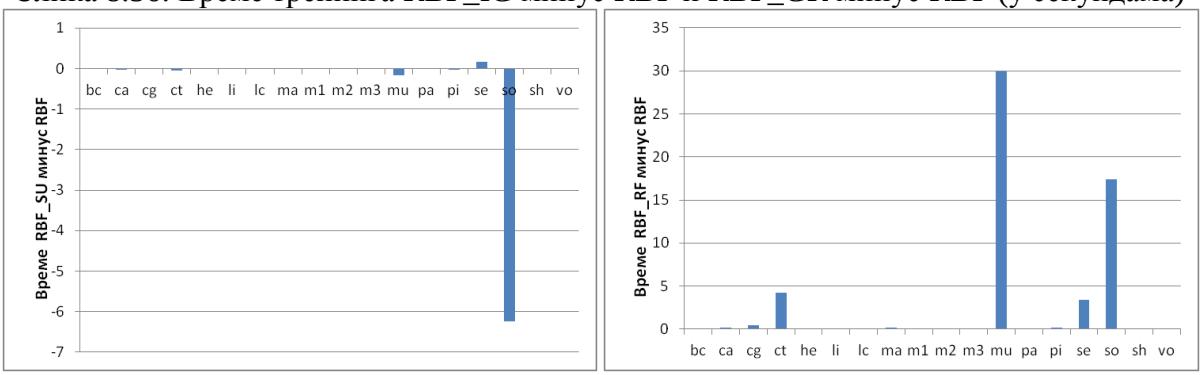
потребно време за тренинг од RBF алгоритма на основном скупу података, а у скоро свим скуповима података резултати су били и статистички лошији.

Метод филтрирања OR је у скоро свим скуповима података показао лошије резултате од RBF алгоритма на основном скупу података, а ови резултати су у већини случаја били и статистички лошији. Примењени метод филтрирања CS је у само 1 скупу података показао лошије резултате од RBF алгоритма на основном скупу података, а у 3 скупа података резултати су били и статистички бољи.

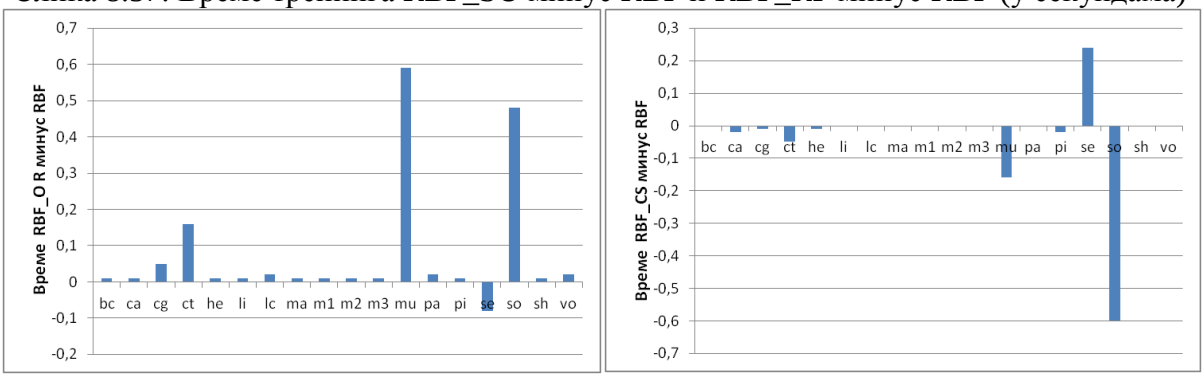
Коришћењем RBF класификатора, можемо да закључимо да је IG метода филтрирања у највећем броју случаја довела до статистички бољих резултата за потребно време за тренинг на посматраним скуповима података.



Слика 8.56: Време тренинга RBF_IG минус RBF и RBF_GR минус RBF (у секундама)



Слика 8.57: Време тренинга RBF_SU минус RBF и RBF_RF минус RBF (у секундама)

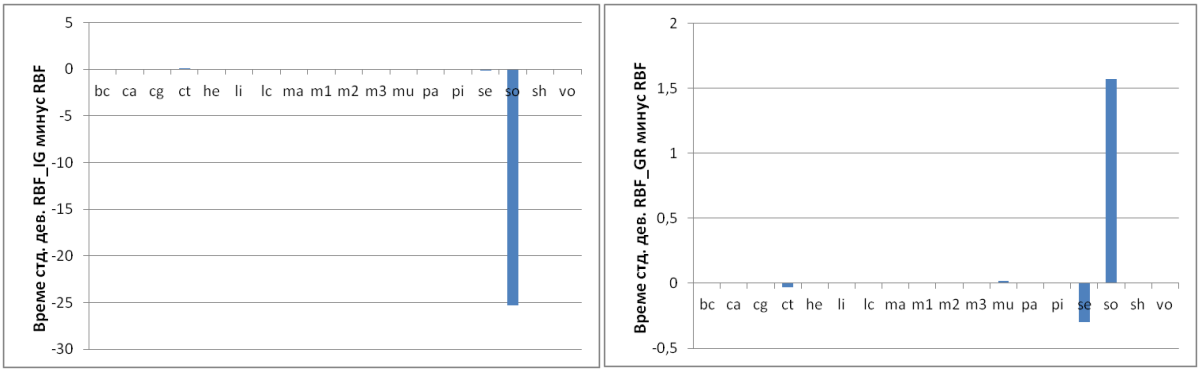


Слика 8.58: Време тренинга RBF_OR минус RBF и RBF_CS минус RBF (у секундама)

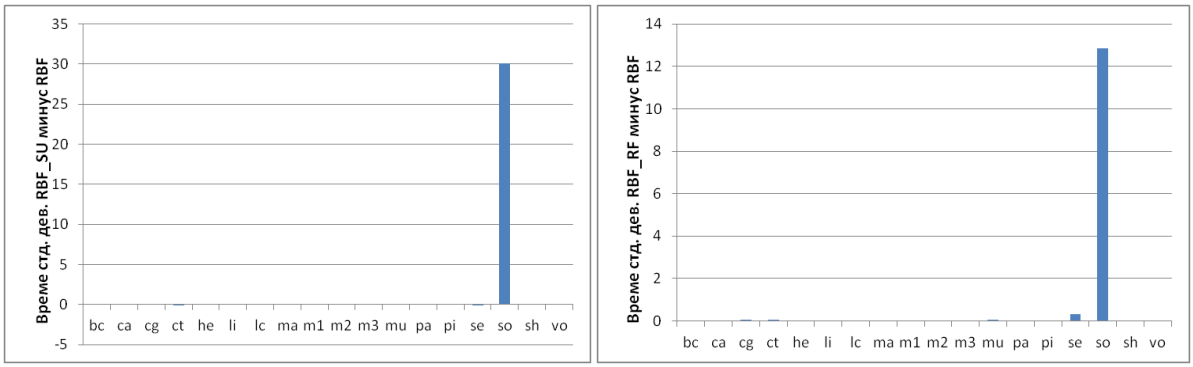
Табела 8.21. приказује стандардну девијацију за време тренинга RBF алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута, осим за скуп података *so* где је уз помоћ неких метода ова вредност знатно већа или знатно мања у односу на оригинални скуп.

Табела 8.21. Стандардна девијација за време тренинга (у секундама) RBF алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода

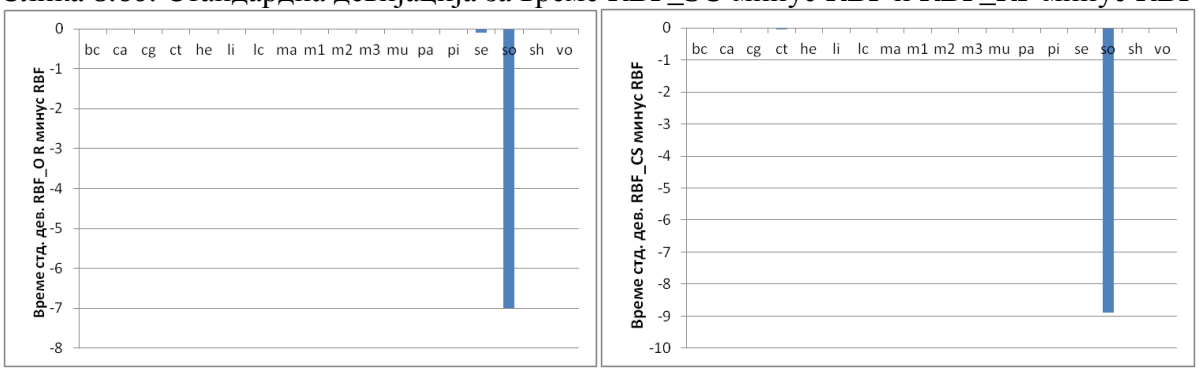
Скуп	RBF	RBF_IG	RBF_GR	RBF_SU	RBF_RF	RBF_OR	RBF_CS
bc	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ca	0.01	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.01	0.01	0.01	0.01	0.02	0.01	0.01
ct	0.09	0.11	0.06	0.06	0.10	0.07	0.05
he	0.01	0.01	0.01	0.01	0.01	0.01	0.01
li	0.01	0.01	0.01	0.01	0.01	0.01	0.01
lc	0.01	0.01	0.01	0.01	0.01	0.01	0.00
ma	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m2	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m3	0.01	0.01	0.01	0.01	0.01	0.01	0.01
mu	0.06	0.06	0.08	0.08	0.13	0.04	0.04
pa	0.01	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.01	0.01	0.01	0.01	0.01	0.01	0.01
se	1.30	1.19	1.00	1.27	1.61	1.21	1.30
so	129.82	104.56	131.39	159.87	142.67	122.82	120.93
sh	0.01	0.01	0.01	0.01	0.01	0.01	0.01
vo	0.01	0.01	0.01	0.01	0.01	0.01	0.01



Слика 8.59: Стандардна девијација за време RBF_IG минус RBF и RBF_GR минус RBF



Слика 8.60: Стандардна девијација за време RBF_SU минус RBF и RBF_RF минус RBF



Слика 8.61: Стандардна девијација за време RBF_OR минус RBF и RBF_CS минус RBF

На сликама 8.59, 8.60. и 8.61. приказана је апсолутна разлика у вредностима стандардне девијације за време тренинга RBF алгоритма за оригинални и редуковани скуп података уз помоћ филтер метода. Највеће одступање у стандардној девијацији у односу на оригинални скуп података показује метода SU за *so* скуп података.

ДЕВЕТИ ДЕО

9. ЕСТИМАЦИЈА ТАЧНОСТИ КЛАСИФИКАЦИЈЕ ЗА МЕТОДЕ ПРЕТХОДНОГ УЧЕЊА

У деветом делу дисертације, након разматрања поставки експерименталног истаживања, биће приказани резултати истраживања за различите методе претходног учења, и то за сваки класификациони алгоритам посебно.

Код метода претходног учења користе се одређени алгоритми за моделирање како би се оценили подскупови атрибута у односу на њихову класификацијску или предиктивну моћ. Код ових метода вредност одређеног скупа атрибута изражава се помоћу степена исправности класификације коју постиже модел конструисан уз коришћење тих атрибута. За класификацију, за све скупове података, коришћена је 10-струка унакрсна валидација, која је при томе била увек поновљена 10 пута. Упоређивана је тачност класификације IBk, *Naïve Bayes*, SVM, J48 и RBF мреже на оригиналном скупу података као и на редукованом скупу података добијеном са методом претходног учења.

Код ових метода за сваки посматрани подскуп атрибута изграђује се модел и оцењују се његове перформансе, тако да боље перформансе неког модела указују на бољи избор атрибута из којих је модел настао. Поступак избора атрибута је рачунски врло захтеван због учесталог извођења алгоритма машинског учења. Потребно је добити оцену перформанси одговарајућег модела за сваки посматрани подскуп атрибута, а методе оцене исправности модела углавном захтевају усредњавање резултата по већем броју изграђених модела. Код ових метода за сваки посматрани подскуп атрибута изграђује се више модела, а укупан број подскупова експоненцијално расте с повећањем броја атрибута.

У овом експерименталном истраживању метода претходног учења, као метода редукције димензионалности података је користила: различите класификаторе за селекцију атрибута, 5-струку унакрсну валидацију и праг за понављање унакрсне валидације ако стандардна девијација пређе ову вредност који је подешен на 0.01.

Исцрпно претраживање подскупова атрибута се може спровести само за мали број атрибута, будући да је тај проблем *NP*-тежак. Зато се користе разне технике

претраживања, као што су: најбољи први (енг. *best-first*), гранај-па-ограничи (енг. *branch-and-bound*), симулирано каљење (енг. *simulated annealing*) и генетски алгоритми.

Табела 9.1. Број атрибута у оригиналном скупу података и број атрибута селектован уз помоћ методе претходног учења за различите класификаторе

Скуп	Ориг. скуп	IBk	Bay	SVM	J48	RBF
bc	9	4	2	2	3	2
ca	15	5	9	6	8	4
cg	20	3	12	10	9	16
ct	23	9	3	5	7	3
he	19	6	3	2	2	13
li	6	5	3	3	6	5
lc	56	4	5	1	2	3
ma	5	2	3	4	4	1
m1	6	3	1	3	3	4
m2	6	3	1	4	6	6
m3	6	3	2	3	3	3
mu	22	5	3	5	5	8
pa	23	7	2	5	5	5
pi	8	1	5	2	4	4
se	19	10	9	11	9	7
so	35	17	17	22	14	19
sh	13	3	11	5	3	10
vo	16	7	3	1	5	4

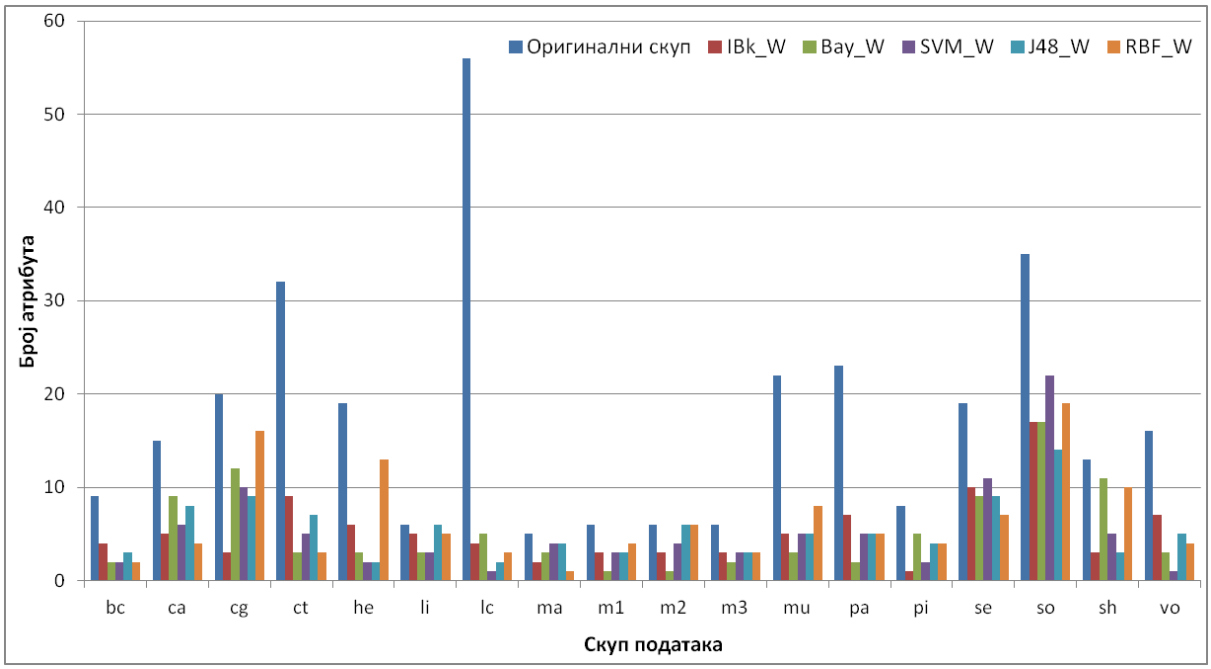
Код методе претходног учења, за претраживање простора решења користили смо хеуристику, како би убрзали претраживање. Хеуристика представља искуствена правила о природи проблема и особинама циља чија је сврха да се претраживање брже усмери ка циљу. Хеуристички или усмерени поступак претраживања је онај поступак претраживања који користи хеуристику како би сузио простор претраживања. У овом раду коришћен је хеуристички поступак „похлепног најбољег првог“ (енг. *greedy best-first*), који претражује подскуп атрибута користећи алгоритам успона на врх (енг. *hill climbing*). Постављање броја узастопних чворова са дозвољеним не-побољшањима контролише ниво праћења уназад. Најбољи први може започети са празним скупом атрибута и претраживати према унапред, односно почети са пуним скупом атрибута и претраживати уназад, или почети у било којој тачки и претраживати у оба смера (разматрања свих могућих појединачних атрибута за додавање или брисање у одређеној тачки). У раду смо користили смер претраживања унапред, што значи да смо

започели са празним скупом, а као критеријум за крај претраживања поставили смо 5 узастопних чворова са дозвољеним не-побољшањима. Главни разлог за избор смера претраживања унапред је рачунски, јер је изградња класификатора са неколико атрибута много бржа него када има више атрибута. Иако у теорији, претраживање уназад од пуног скупа атрибута, може лакше ухватити интеракцију атрибута, метода је изузетно рачунски скупа.

У експерименталном истраживању, као и код метода филтрирања користили смо упоредни *t*-тест, где је ниво значајности постављен на вредност 0.05.

С обзиром да су постојали сетови података са недостајућим вредностима, да би могли да користимо SVM алгоритам, било је неопходно заменити недостајуће вредности са процењеним вредностима за дати скуп, јер сам алгоритам SVM није могао да се избори са недостајућим вредностима за поједине атрибуте у неким од истанци.

У табели 9.1. приказан је оптималан број атрибута за потребе класификације, након претраживања скупа могућих решења за сваки од класификатора. Табела приказује и оригиналну величину скупа, како би се упоредили ефекти редукције димензионалности података. Од 18 посматраних сетова података, у 15 сетова података (сви осим *li*, *ma* и *m2*), тачно пола или више од пола класификатора је смањило оригинални број атрибута на пола.



Слика 9.1: Број атрибута у оригиналном скупу и оптималан број атрибута добијен методама претходног учења

Слика 9.1. приказује број атрибута у оригиналном скупу података и оптималан број атрибута добијен методама претходног учења. Највећу добробит од редукције димензионалности података има скуп података *lc*, где од 56 атрибута, методом претходног учења смо издвојили мали број атрибута релевантних за посматрани проблем класификације, чак исто или мање од пет, за сваки од класификатора.

Користећи методе претходног учења за чак 7 скупова података, сви класификатори смањују број атрибута на исто или више од пола. Ти скупови податак су: *bc, ct, lc, m3, mu, pa* и *vo*. Можемо уочити да су ове методе довеле до значајне редукције димензионалности података. Ако упоредимо податке приказане на сликама 8.1. и 9.1. можемо уочити да је редукција димнзионалности података у значајно већој мери урађена код метода претходног учења. За разлику од метода филтрирања, не постоје сетови података где су сви класификатори изабрали исти број значајних атрибута за дати скуп података.

Табела 9.2. Тачност класификације различитих класификатора за оригинални и редуковани скуп података уз помоћ метода претходног учења

Скуп	IBk	IBk_W	Bay	Bay_W	SVM	SVM_W	J48	J48_W	RBF	RBF_W
bc	72.85	69.81	72.70	72.26	72.18	73.47	74.28	72.95	71.41	74.01
ca	81.57	85.22 +	77.86	85.67 +	55.88	85.86 +	85.57	84.43	79.55	85.91 +
cg	71.88	71.70	75.16	74.00	70.00	72.62 +	71.25	71.72	73.58	73.93
ct	98.85	98.42	87.30	98.49 +	81.01	98.38 +	98.57	98.88	97.93	98.67 +
he	81.40	81.85	83.81	82.21	79.38	83.90	79.22	81.90	85.29	82.12
li	62.22	59.66	54.89	59.46	59.37	60.62	65.84	66.36	65.06	62.86
lc	68.75	70.67	78.42	79.33	72.67	77.42	79.25	78.83	76.00	76.08
ma	75.60	83.02 +	82.64	82.01	80.27	82.03	82.19	82.47	77.31	81.11 +
m1	99.87	100.00	74.64	74.64	91.37	97.83 +	97.80	100.00	75.36	88.16 +
m2	79.08	65.72 -	62.79	65.72 +	65.44	65.72	63.48	65.72	67.82	65.67
m3	97.46	98.92 +	96.39	96.39	96.39	98.92 +	98.92	98.92	96.54	97.49
mu	100.00	100.00	95.76	99.63 +	100.00	100.00	100.00	100.00	98.61	97.12
pa	95.91	93.40	69.98	82.04 +	79.36	97.64 +	84.74	86.24	81.22	87.47 +
pi	70.62	67.76	75.75	76.11	65.11	71.76 +	74.49	73.44	74.04	75.79
se	97.15	97.08	80.17	89.83 +	64.76	90.91 +	96.79	96.73	87.88	91.82
so	91.20	94.77 +	92.94	92.67	90.04	89.17	91.78	91.74	84.48	84.41
sh	76.15	78.56	83.59	84.30	55.93	81.74 +	78.15	81.74	83.11	82.59
vo	92.58	94.92 +	90.02	95.75 +	95.63	95.54	96.57	95.24 -	93.73	94.94

У наставку експерименталног истраживања, за изабрани оптималан број атрибута, за сваки скуп података и класификатор, проверавана је тачност класификације коришћењем различитих алгоритама, и то: IBk, *Naïve Bayes*, SVM, J48 и

RBF мреже. У наставку текста приказани су добијени резултати. Приказане су различите скале на сликама за апсолутну тачност класификације, стандардну девијацију за тачност, време тренинга и стандардну девијацију за време, како би се боље уочиле разлике које постоје међу резултатима.

У табелама које следе за тачност класификације различитих класификатора и у табелама за време потребно за тренинг података су приказане ознаке „+“ и „-“, које означавају да је одређени резултат статистички бољи (+) или лошији (-) од основног класификатора на нивоу значајности који је специфициран на вредност од 0,05.

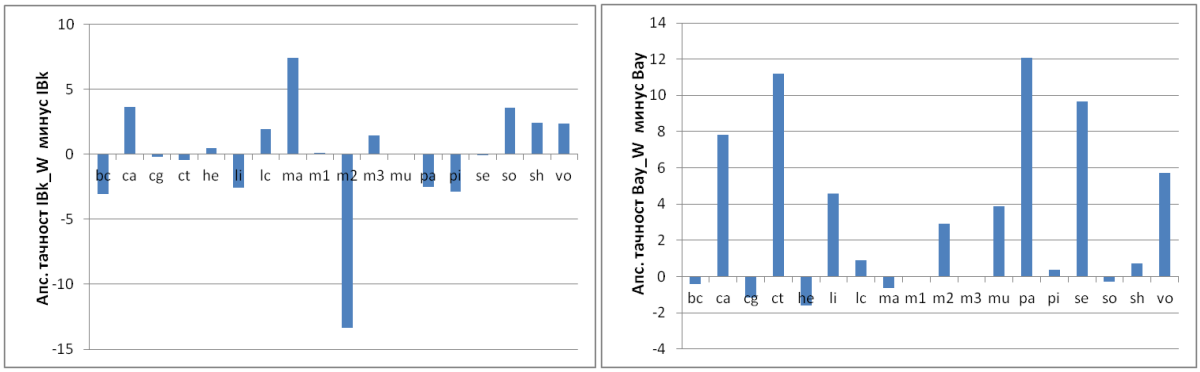
Табела 9.2. приказује тачност класификације различитих класификатора за оригинални и редуковани скуп података уз помоћ метода претходног учења. Можемо уочити да у свим сетовима података имамо добијене резултате за бар једну од метода претходног учења који су статистички бољи од основног класификатора. Само у два сета података *m2* и *vo*, имамо значајно лошије податке за неку од метода претходног учења.

На сликама 9.2, 9.3. и 9.4. приказана је апсолутна разлика у тачности класификације различитих алгоритма на основном скупу података и истих тих алгоритма са методама претходног учења. Метод претходног учења са IBk класификатором је у више од пола скупова података (10 скупова) показао исте или боље резултате од IBk алгоритма на основном скупу података, а у 5 скупова података резултати су били и статистички бољи. Метод претходног учења са *Naïve Bayes* класификатором је у више од две трећине скупова података (13 скупова) показао исте или боље резултате од *Naïve Bayes* алгоритма на основном скупу података, а у 7 скупова података резултати су били и статистички бољи.

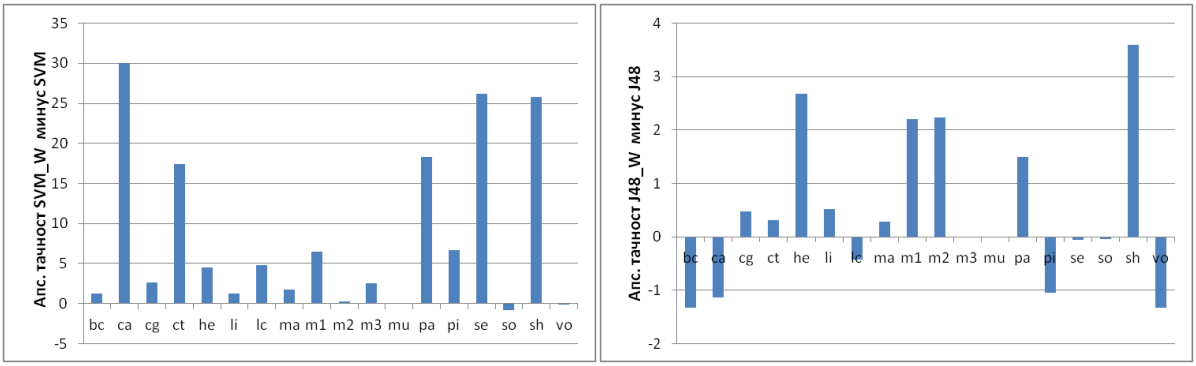
Метод претходног учења са SVM класификатором је у скоро свим скуповима података (16 скупова) показао исте или боље резултате од SVM алгоритма на основном скупу података. У 9 скупова података резултати су били и статистички бољи. Метод претходног учења са J48 класификатором је у више од пола скупова података (11 скупова) показао исте или боље резултате од J48 алгоритма на основном скупу података, али не постоји резултат који би био и статистички бољи.

Метод претходног учења са RBF класификатором је у две трећине скупова података (12 скупова) показао исте или боље резултате од RBF алгоритма на основном скупу података, а у 5 скупова података резултати су били и статистички бољи.

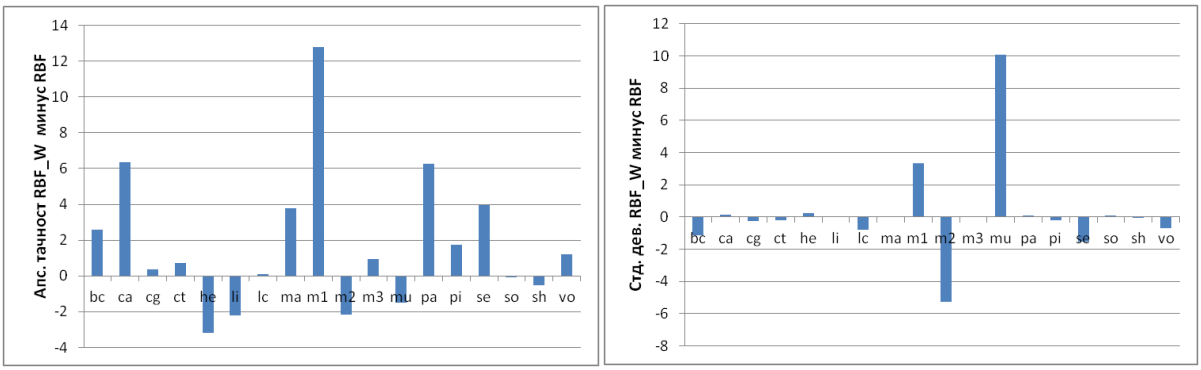
Коришћењем методе претходног учења можемо да закључимо да је SVM класификатор у највећем броју случаја довео до статистички бољих резултата на посматраним скуповима података.



Слика 9.2: Апсолутна тачност класификације IBk_W минус IBk и Bay_W минус Bay



Слика 9.3: Апсолутна тачност класификације SVM_W минус SVM и J48_W минус J48



Слика 9.4: Апсолутна тачност класификације RBF_W минус RBF и стандардна девијација за тачност RBF_W минус RBF

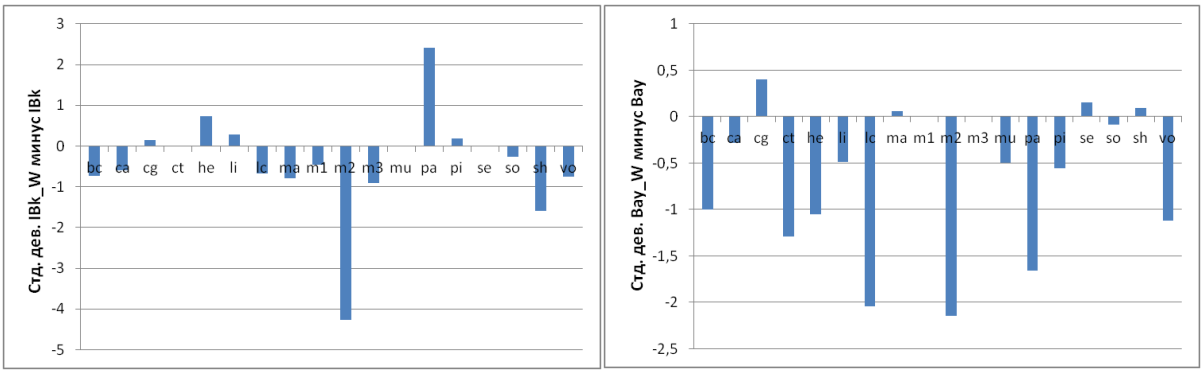
С обзиром на изнету тврдњу да је добар онај алгоритам који даје сличан резултат у свим случајевима, односно вредност за стандардну девијацију је минимална, разматраћемо вредности за стандардну девијацију за тачност класификације. Табела 9.3. приказује стандардну девијацију за тачност класификације различитих алгоритма за оригинални и редуковани скуп података уз помоћ метода претходног учења. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између

стандардног алгоритма и алгоритама који користе председелекцију атрибута, осим у случају SVM алгоритма, код кога су вредности стандардне девијације за тачност класификације значајно веће са методом претходног учења. Већа одступања у стандардној девијацији показује метода RBF мреже, у односу на друге алгоритме, али само за поједине скупове података. Генерално, вредности стандардне девијације за тачност класификације је код осталих алгоритама мања са методом претходног учења.

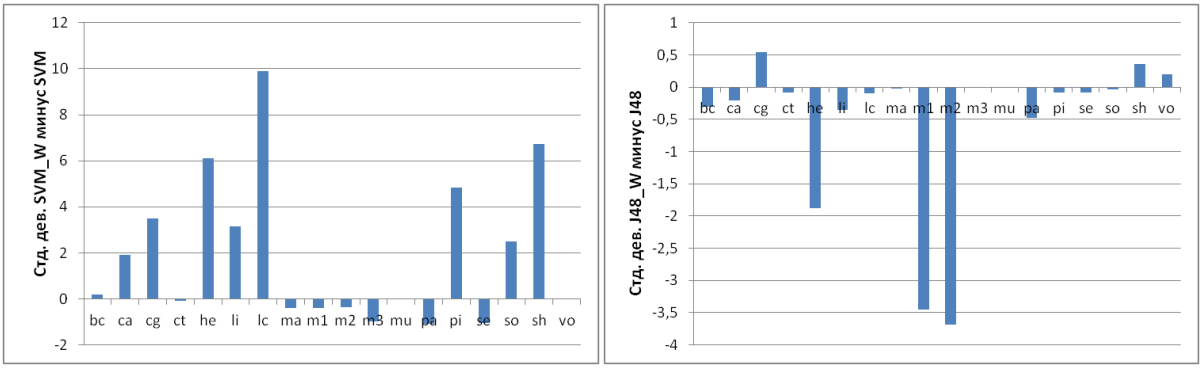
Табела 9.3.Стандардна девијација за тачност класификације различитих класификатора за оригинални и редуковани скуп података уз помоћ метода претходног учења

Скуп	IBk	IBk_W	Bay	Bay_W	SVM	SVM_W	J48	J48_W	RBF	RBF_W
bc	6.93	6.19	7.74	6.74	5.86	6.04	6.05	5.74	7.88	6.75
ca	4.57	3.98	4.18	3.90	2.14	4.05	3.96	3.75	4.07	4.20
cg	3.68	3.82	3.48	3.88	0.00	3.48	3.17	3.71	4.30	4.03
ct	0.77	0.77	2.21	0.92	0.99	0.89	0.89	0.80	1.02	0.82
he	8.55	9.29	9.70	8.65	2.26	8.38	9.57	7.69	8.29	8.53
li	8.18	8.47	8.83	8.34	2.28	5.42	7.40	7.05	8.80	8.81
lc	22.33	21.66	21.12	19.08	11.12	21.03	21.50	21.40	22.91	22.12
ma	3.90	3.10	3.11	3.17	3.41	3.01	3.21	3.19	3.31	3.35
m1	0.46	0.00	4.26	4.26	3.10	2.71	3.45	0.00	5.92	9.26
m2	5.06	0.79	2.94	0.79	1.14	0.79	4.48	0.79	6.24	0.97
m3	2.13	1.23	2.20	2.20	2.20	1.23	1.23	1.23	2.19	2.20
mu	0.00	0.00	0.73	0.23	0.00	0.00	0.00	0.00	0.58	10.65
pa	4.52	6.93	9.51	7.85	4.46	3.37	8.01	7.53	7.37	7.44
pi	4.67	4.85	5.32	4.76	0.34	5.18	5.27	5.18	4.91	4.72
se	1.11	1.11	2.12	2.27	2.66	1.62	1.29	1.20	2.57	1.04
so	3.00	2.74	2.92	2.83	0.39	2.88	3.19	3.16	0.86	0.94
sh	8.46	6.87	5.98	6.07	1.12	7.85	7.42	7.78	6.50	6.44
vo	3.63	2.87	3.91	2.79	2.76	2.77	2.56	2.76	3.87	3.19

На сликама 9.4, 9.5. и 9.6. приказана је апсолутна разлика у вредностима стандардне девијације за тачност класификације различитих алгоритма за оригинални и редуковани скуп података уз помоћ метода претходног учења. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, то је и веће одступање између стандардних девијација. Најмање одступање у стандардној девијацији у односу на оригинални скуп података, показује алгоритам *Naïve Bayes* и J48, док највеће одступање има алгоритам SVM и RBF мреже.



Слика 9.5: Стандардна девијација за тачност IBk_W минус IBk и Bay_W минус Bay



Слика 9.6: Стандардна девијација за тачност SVM_W минус SVM и J48_W минус J48

У табели 9.4. приказано је потребно време за тренинг у секундама различитих класификатора који користе оригинални и редуковани скуп података уз помоћ метода претходног учења. Потребно време за тренинг података IBk класификатора за све оригиналне скупове података износи 0.00 секунди, док за методе претходног учења оно је значајно веће. Потребно време за тренинг података *Naïve Bayes* класификатора за све оригиналне скупове података износи мање од 0.01 секунди, док за методе претходног учења оно је веће.

За потребно време тренинга података SVM класификатора за све оригиналне скупове података оно износи мање од 3.73 секунде, док за методе претходног учења оно је значајно веће. Потребно време за тренинг података J48 класификатора за све оригиналне скупове података износи мање од 0.1 секунде, док за методе претходног учења оно је веће. Потребно време за тренинг података RBF класификатора за све оригиналне скупове података износи мање од 179.02 секунде, док за методе претходног учења оно је значајно веће.

На сликама 9.7, 9.8. и 9.9. приказана је апсолутна разлика у потребном времену за тренинг различитих класификатора на основном скупу података и са редукованим бројем атрибута уз помоћ метода претходног учења. Можемо да уочимо на основу

слика и верикалне осе која приказује време у секундама, да класификатор RBF са методом претходног учења захтева највише времена за учење, у случају *so* скупа података и до 1389781.94 секунди, а потом класификатор SVM који за исти скуп података захтева време од 8806.81 секунди.

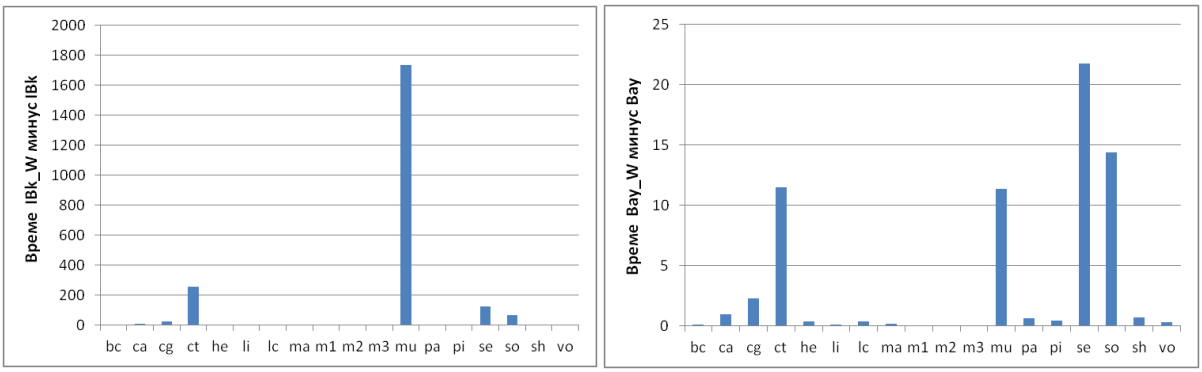
Табела 9.4. Потребно време за тренинг (у секундама) различитих класификатора за оригинални и редуковани скуп података уз помоћ метода претходног учења

Скуп	IBk	IBk_W	Bay	Bay_W	SVM	SVM_W	J48	J48_W	RBF	RBF_W
bc	0.00	0.96 -	0.00	0.12 -	0.01	6.28 -	0.00	0.31 -	0.01	3.69 -
ca	0.00	10.03 -	0.00	0.98 -	0.14	84.73 -	0.01	3.95 -	0.02	30.63 -
cg	0.00	23.08 -	0.00	2.30 -	0.50	312.81 -	0.01	12.20 -	0.05	67.23 -
ct	0.00	254.81 -	0.01	11.48 -	3.73	555.86 -	0.10	26.38 -	0.35	1189.54 -
he	0.00	1.30 -	0.00	0.41 -	0.01	18.07 -	0.00	0.57 -	0.01	11.10 -
li	0.00	0.59 -	0.00	0.15 -	0.02	6.95 -	0.00	0.47 -	0.01	2.59 -
lc	0.00	1.21 -	0.00	0.39 -	0.00	35.65 -	0.00	0.32 -	0.00	11.84 -
ma	0.00	3.52 -	0.00	0.18 -	0.13	16.72 -	0.00	0.68 -	0.02	4.33 -
m1	0.00	1.01 -	0.00	0.06 -	0.03	5.88 -	0.00	0.15 -	0.01	4.38 -
m2	0.00	0.92 -	0.00	0.06 -	0.04	6.30 -	0.00	0.07 -	0.01	1.81 -
m3	0.00	0.93 -	0.00	0.05 -	0.02	3.82 -	0.00	0.08 -	0.01	3.35 -
mu	0.00	1733.88-	0.00	11.33 -	3.73	1572.20 -	0.03	20.34 -	0.49	943.60 -
pa	0.00	3.09 -	0.00	0.67 -	0.02	62.12 -	0.01	3.87 -	0.02	19.36 -
pi	0.00	4.07 -	0.00	0.47 -	0.14	40.10 -	0.01	2.03 -	0.03	10.49 -
se	0.00	122.89 -	0.01	21.74 -	3.65	3012.16 -	0.08	90.07 -	2.61	11271.55
so	0.00	68.93 -	0.00	14.38 -	0.81	8806.81	0.01	29.48 -	179.02	1389781.94-
sh	0.00	1.82 -	0.00	0.72 -	0.02	19.20 -	0.00	1.66 -	0.01	11.02 -
vo	0.00	5.44 -	0.00	0.29 -	0.02	4.71 -	0.00	0.46 -	0.01	21.21 -

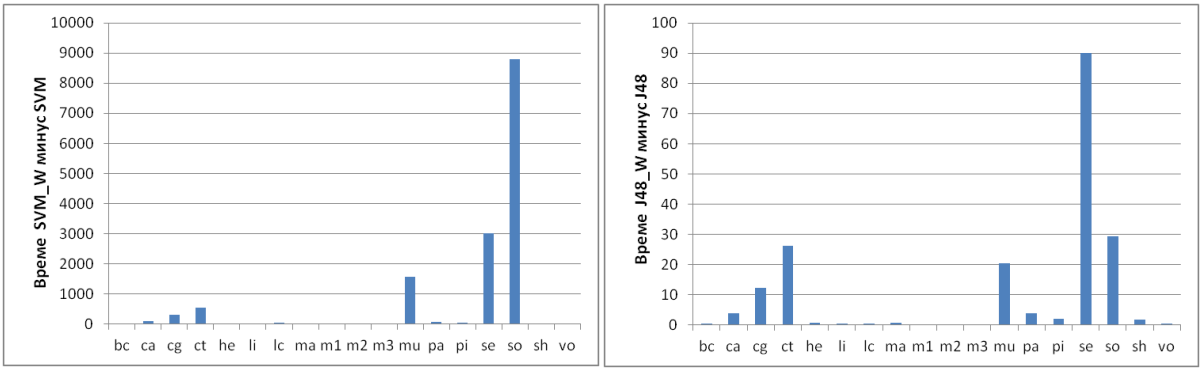
Код методе претходног учења и код великих сетова података, и то посебно код *mu*, *se* и *so* при коришћењу SVM и RBF алгоритма, задатак редукције димензионалности података је био захтеван и у погледу меморијских ресурса којих је било потребно обезбедити, као и у погледу захтеваног времена за извршавање алгоритма. Ово није био случај код метода филтрирања и екстракције, ни за једну од примењених метода на било ком скупу података.

Можемо уочити да су ове методе довеле до значајног повећања потребног времена за тренинг података. Ако упоредимо податке приказане на сликама који се односе на потребно време за тренинг коришћењем филтер метода и метода претходног учења, можемо уочити да је у значајно већој мери потребно времена методама претходног учења. Такође, доказ изнетог тврђења налази се и у табели 9.4, где за све

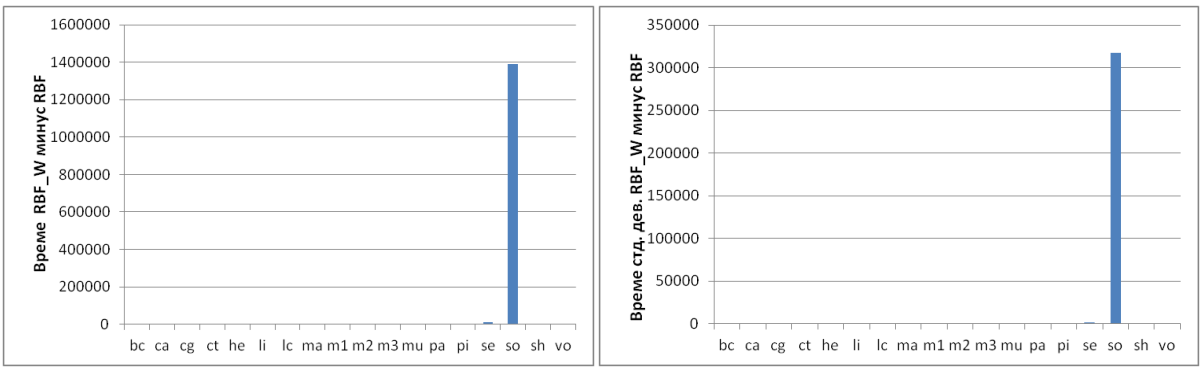
скупове података и за све коришћене алгоритме, резултати за потребно време тренинга су статистички лошији, односно захтевано је више времена за тренинг.



Слика 9.7: Време тренинга IBk_W минус IBk и Bay_W минус Bay (у секундама)



Слика 9.8: Време тренинга SVM_W минус SVM и J48_W минус J48 (у секундама)



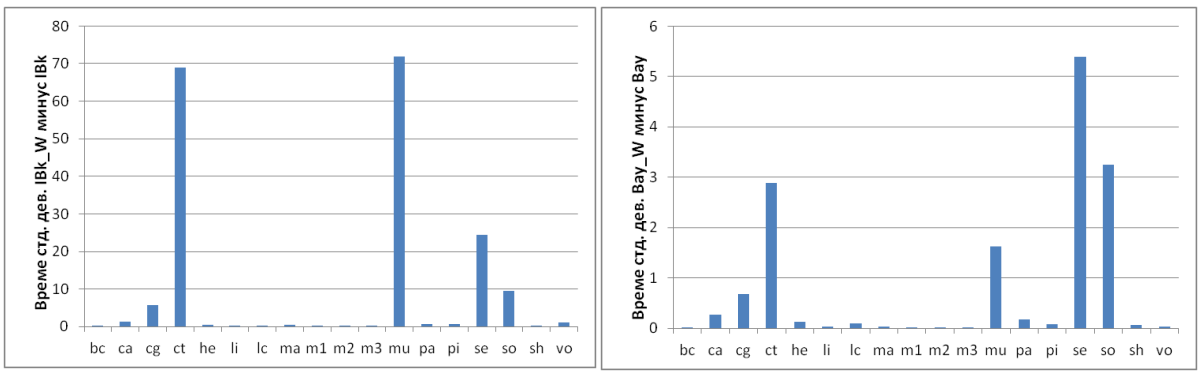
Слика 9.9: Време тренинга RBF_W минус RBF (у секундама) и стандардна девијација за време RBF_W минус RBF

Табела 9.5. приказује стандардну девијацију за време тренинга различитих алгоритама за оригинални и редуковани скуп података уз помоћ метода претходног учења. Из табеле се може видети да се стандардне девијације разликују доста између стандардног алгоритма и алгоритама који користе методе претходне редукције атрибута. У случају свих алгоритама, код свих скупова података, вредности за стандардну девијацију за време тренинга су веће код класификатора који користе методе претходне редукције атрибута, у односу на стандардни алгоритам.

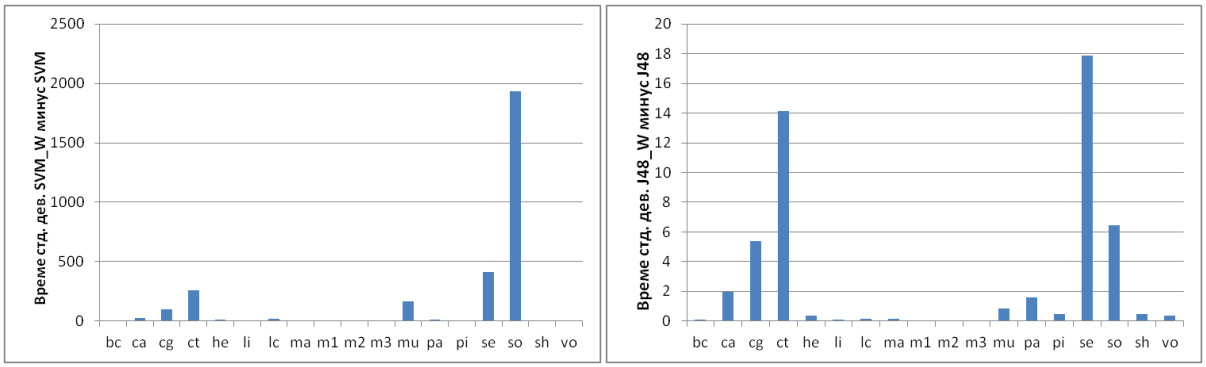
На сликама 9.9, 9.10. и 9.11. приказана је апсолутна разлика у вредностима стандардне девијације за време тренинга алгоритама за оригинални и редуковани скуп података уз помоћ метода претходног учења. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, у позитивном и негативном смеру, то је и веће одступање између стандардних девијација. Највеће одступање у стандардној девијацији у односу на оригинални скуп података показују методе претходног учења са RBF и SVM алгоритмом.

Табела 9.5. Стандардна девијација потребног времена за тренинг (у секундама) различитих класификатора за оригинални и редуковани скуп података уз помоћ метода претходног учења

Скуп	IBk	IBk_W	Bay	Bay_W	SVM	SVM _W	J48	J48_W	RBF	RBF_W
bc	0.00	0.17	0.00	0.02	0.00	1.21	0.01	0.09	0.01	0.64
ca	0.00	1.39	0.01	0.29	0.01	22.20	0.01	2.00	0.01	7.76
cg	0.00	5.76	0.00	0.68	0.06	101.38	0.00	5.38	0.01	12.99
ct	0.00	68.98	0.01	2.90	0.21	259.99	0.01	14.17	0.09	346.38
he	0.00	0.43	0.00	0.13	0.01	9.76	0.01	0.36	0.01	2.67
li	0.00	0.08	0.00	0.03	0.01	1.19	0.01	0.08	0.01	0.33
lc	0.00	0.32	0.00	0.10	0.01	20.60	0.00	0.14	0.01	3.05
ma	0.00	0.46	0.00	0.04	0.01	2.83	0.01	0.15	0.01	0.44
m1	0.00	0.08	0.00	0.01	0.01	0.61	0.00	0.01	0.01	0.60
m2	0.00	0.06	0.00	0.01	0.01	0.19	0.00	0.01	0.01	0.35
m3	0.00	0.07	0.00	0.01	0.01	0.35	0.00	0.01	0.01	0.29
mu	0.01	71.89	0.01	1.64	0.09	164.05	0.01	0.88	0.07	256.98
pa	0.00	0.77	0.00	0.18	0.01	15.40	0.01	1.58	0.02	4.52
pi	0.00	0.69	0.00	0.08	0.01	5.09	0.01	0.47	0.01	1.41
se	0.00	24.38	0.01	5.40	0.30	413.59	0.01	17.86	0.13	1370.49
so	0.00	9.60	0.00	3.25	0.09	1931.81	0.01	6.46	148.71	317542.22
sh	0.00	0.26	0.00	0.07	0.01	4.97	0.01	0.47	0.01	1.05
vo	0.00	1.09	0.00	0.04	0.01	0.68	0.00	0.34	0.01	4.79



Слика 9.10: Стандардна девијација за време IBk_W минус IBk и Bay_W минус Bay



Слика 9.11: Стандардна девијација за време SVM_W минус SVM и J48_W минус J48

ДЕСЕТИ ДЕО

10. ЕСТИМАЦИЈА ТАЧНОСТИ КЛАСИФИКАЦИЈЕ ЗА ЕКСТРАКЦИЈУ АТРИБУТА

У десетом делу дисертације, након разматрања поставки експерименталног истаживања, биће приказани резултати истраживања за екстракцију атрибута уз помоћ PCA методе, и то за сваки класификациони алгоритам посебно.

Проблем димензионалности се може превладати тако да се одабере само подскуп релевантних атрибута или стварањем нових атрибута које садрже највише информација о класи. Прва методологија се зове селекција атрибута, а друга се зове екстракција атрибута, а то укључује линеарну (PCA, ICA и сл.) и не-линеарну методу екстракције атрибута. У експерименталном истраживању користили смо PCA методу, као методу екстракције атрибута.

PCA представља технику формирања нових, синтетских варијабли које су линеарне сложенице - комбинације изворних варијабли. Овом техником се редукује димензионалност, а максимални број нових варијабли који се може формирати једнак је броју изворних, при чему нове варијабле нису међусобно корелисане. Очекује се да ће већина нових варијабли чинити шум, и имати тако малу варијансу да се она може занемарити. Већину информација ће понети првих неколико варијабли - главних компоненти, чије су варијансе значајне величине. На тај начин, из великог броја изворних варијабли креирано је тек неколико главних компоненти које носе већину информација и чине главни облик. Наравно, има ситуација када то није тако, и то у случају када су изворне варијабле некорелисане, тада анализа не даје повољне резултате.

У анализи главних компонената основни кораци су: стандардизација варијабли, израчунавање матрице корелације, проналажење својствених вредности главних компоненти и одбацивање компоненти. Најпре, потребно је стандардизовати варијабле тако да им је просек 0, а варијанса 1 како би све биле на једнаком нивоу у анализи, јер је већина сетова података конструисана из варијабли различитих скала и јединица мерења. Потом, потребно је израчунати матрице корелација између свих изворних стандардизованих варијабли, а након тога, пронаћи својствене вредности главних

компонената. На крају, потребно је одбацити оне компоненте које имају пропорционално мали удео варијансе (обично првих неколико компоненти има 80% - 90% укупне варијансе). У експерименталном истраживању користили смо за праг одбацивања вредност од 95%.

У табелама које следе за тачност класификације различитих класификатора и у табелама за време потребно за тренинг података су приказане ознаке „+“ и „-“, које означавају да је одређени резултат статистички бољи (+) или лошији (-) од основног класификатора на нивоу значајности који је специфициран на вредност од 0,05. Такође, приказане су различите скале на сликама за апсолутну тачност класификације, стандардну девијацију за тачност, време тренинга и стандардну девијацију за време, како би се боље уочиле разлике које постоје међу резултатима.

Табела 10.1. Тачност класификације различитих класификатора за оригинални и редуковани скуп података уз помоћ PCA

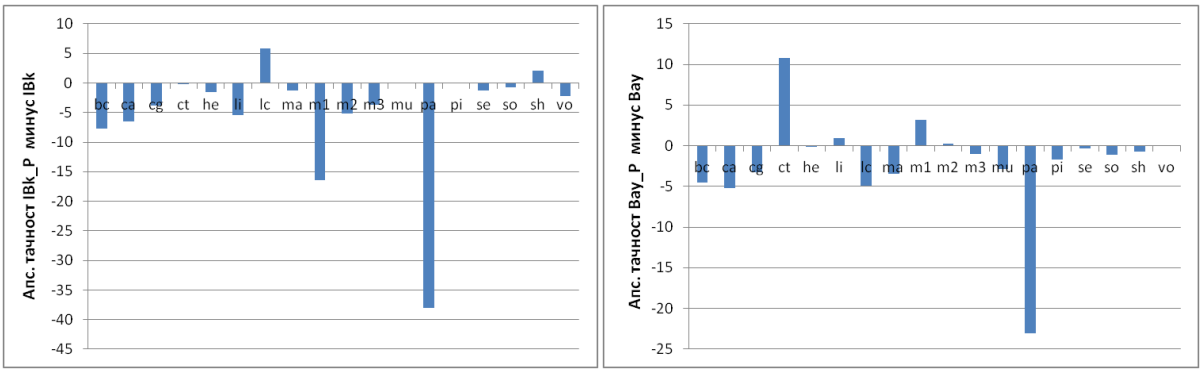
Скуп	IBk	IBk_P	Bay	Bay_P	SVM	SVM_P	J48	J48_P	RBF	RBF_P
bc	72.85	65.08 -	72.70	68.17	72.18	71.50	74.28	67.94 -	71.41	68.58
ca	81.57	75.12 -	77.86	72.67 -	55.88	85.49 +	85.57	79.91 -	79.55	77.09
cg	71.88	68.08 -	75.16	71.85 -	70.00	75.53 +	71.25	68.25	73.58	71.56
ct	98.85	98.59	87.30	98.11 +	81.01	98.97 +	98.57	98.14	97.93	98.28
he	81.40	79.79	83.81	83.68	79.38	85.90 +	79.22	80.31	85.29	83.71
li	62.22	56.81 -	54.89	55.85	59.37	62.29	65.84	56.36 -	65.06	61.46
lc	68.75	74.58	78.42	73.50	72.67	71.67	79.25	63.08	76.00	72.75
ma	75.60	74.27	82.64	79.18 -	80.28	81.97	82.19	80.46	77.31	79.08
m1	99.87	83.38 -	74.64	77.81	91.37	100.00 +	97.80	96.71	75.36	83.52 +
m2	79.08	73.89	62.79	63.06	65.44	61.20 -	63.48	76.34 +	67.82	67.73
m3	97.46	93.81	96.39	95.33	96.39	98.92 +	98.92	96.54 -	96.54	95.96
mu	100.00	100.00	95.76	92.88 -	100.00	99.94 -	100.00	99.75 -	98.61	98.31
pa	95.91	57.86 -	69.98	46.90 -	79.36	82.89	84.74	82.41	81.22	58.38 -
pi	70.62	70.54	75.75	74.08	65.11	76.38 +	74.49	70.92	74.04	73.63
se	97.15	95.93 -	80.17	79.82	63.98	91.68 +	96.79	91.43 -	87.31	87.04
so	91.20	90.48	92.94	91.83	93.63	92.94	91.78	86.76 -	93.63	92.97
sh	76.15	78.19	83.59	82.85	55.93	83.37 +	78.15	76.41	83.11	81.93
vo	92.58	90.31	90.02	90.09	95.63	94.25	96.57	90.27 -	93.73	92.50

Можемо уочити да у десет сетова података (*ca, cg, ct, he, m1, m2, m3, pi, se* и *sh*) имамо добијене резултате за тачност класификације за редуковани скуп података уз помоћ PCA методе за бар један од класификатора који су статистички бољи од основног класификатора (табела 10.1). Ни у једном сету података, немамо значајно лошије податке за све класификаторе, што значи да увек можемо изабрати

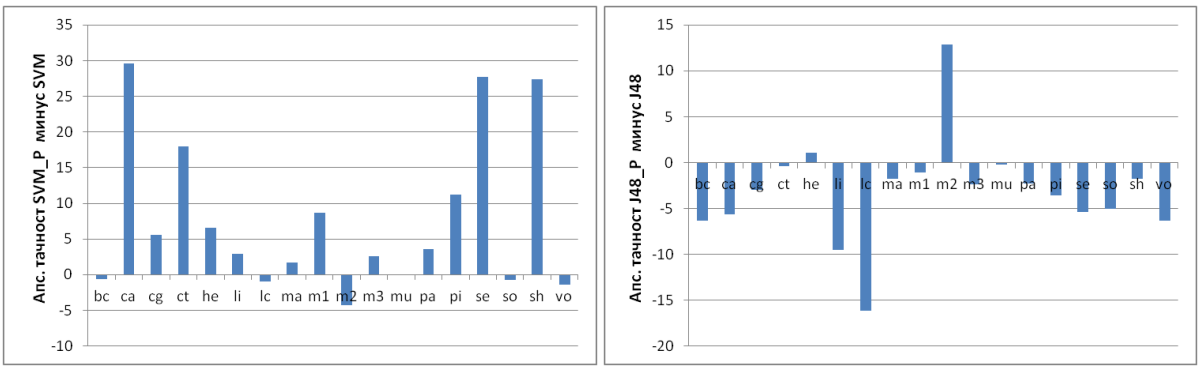
класификатор за дати скуп података која има статистички боље резултате или резултате који су приближни оригиналном скупу података.

На сликама 10.1, 10.2. и 10.3. приказана је апсолутна разлика у тачности класификације различитих алгоритма на основном скупу података и редукованом скупу података коришћењем PCA. Код IBk алгоритма уз коришћење PCA је у само 3 скупа података показао исте или боље резултате за тачност класификације од IBk алгоритма на основном скупу података, али ни у једном скупу података резултати нису били и статистички бољи. PCA је код *Naïve Bayes* алгоритма у мање од трећине скупова података (5 скупова) показао исте или боље резултате од *Naïve Bayes* алгоритма на основном скупу података, а само у 1 скупу података резултати су били и статистички бољи.

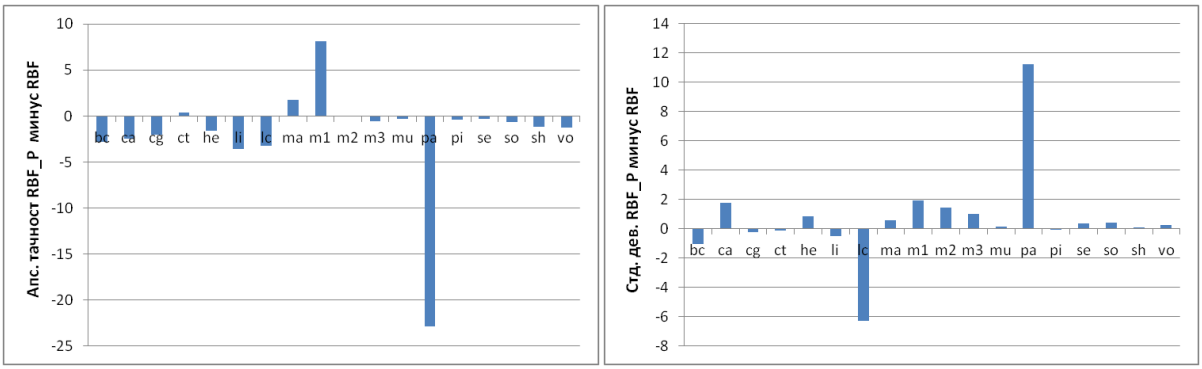
PCA је код SVM алгоритма у две трећине скупова података (12 скупова) показао исте или боље резултате од SVM алгоритма на основном скупу података. У 9 скупова података резултати за тачност класификације су били и статистички бољи. Код примене PCA на J48 алгоритам, у само 2 скупа података показао је исте или боље резултате од J48 алгоритма на основном скупу података. Само у 1 скупу података резултати су били и статистички бољи.



Слика 10.1: Апсолутна тачност класификације IBk_P минус IBk и Bay_P минус Bay



Слика 10.2: Апсолутна тачност класификације SVM_P минус SVM и J48_P минус J48



Слика 10.3: Апсолутна тачност класификације RBF_P минус RBF и стандардна девијација за тачност RBF_P минус RBF

PCA код RBF алгоритма је у само 3 скупа података показао исте или боље резултате од RBF алгоритма на основном скупу података, а у 1 скупу података, резултати су били и статистички бољи.

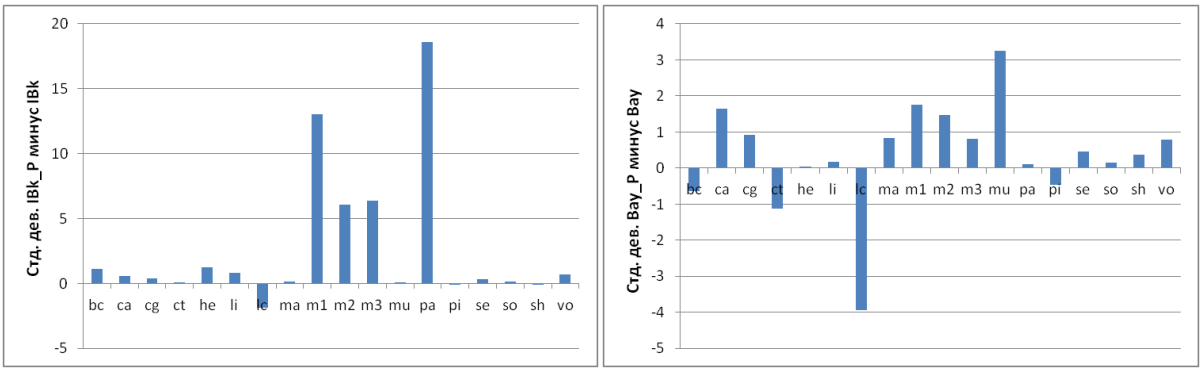
Коришћењем PCA за редукцију димензионалности података, можемо да закључимо да је SVM алгоритам у највећем броју случаја довео до статистички бољих резултата на посматраним скуповима података.

Табела 10.2. Стандардна девијација за тачност класификације различитих класификатора за оригинални и редуковани скуп података уз помоћ PCA

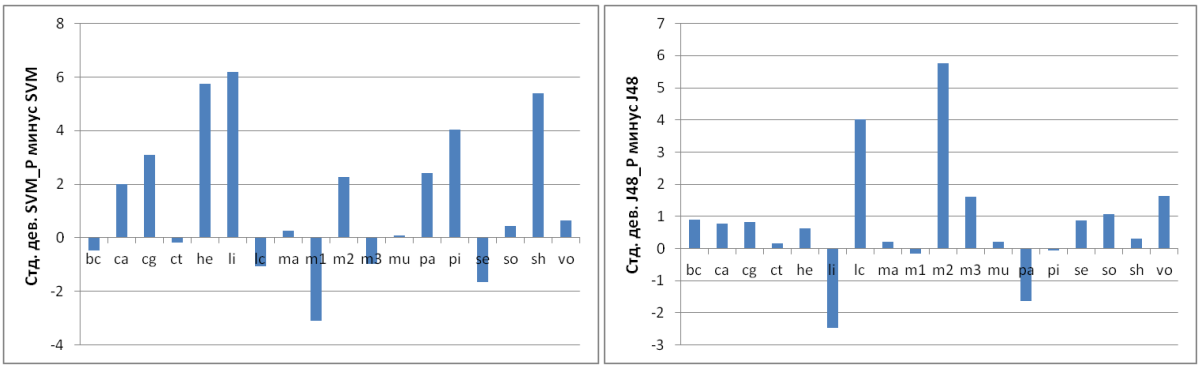
Скуп	IBk	IBk_P	Bay	Bay_P	SVM	SVM_P	J48	J48_P	RBF	RBF_P
bc	6.93	8.07	7.74	7.11	5.86	5.38	6.05	6.95	7.88	6.81
ca	4.57	5.15	4.18	5.83	2.14	4.13	3.96	4.73	4.07	5.83
cg	3.68	4.09	3.48	4.40	0.00	3.08	3.17	3.98	4.30	4.06
ct	0.77	0.85	2.21	1.09	0.99	0.80	0.89	1.06	1.02	0.91
he	8.55	9.81	9.70	9.75	2.26	8.01	9.57	10.20	8.29	9.12
li	8.18	8.99	8.83	9.00	2.28	8.47	7.40	4.93	8.80	8.31
lc	22.33	20.46	21.12	17.18	11.12	10.05	21.50	25.52	22.91	16.62
ma	3.90	4.04	3.11	3.95	3.42	3.67	3.21	3.41	3.31	3.86
m1	0.46	13.50	4.26	6.01	3.10	0.00	3.45	3.30	5.92	7.86
m2	5.06	11.11	2.94	4.40	1.14	3.40	4.48	10.24	6.24	7.70
m3	2.13	8.48	2.20	3.02	2.20	1.23	1.23	2.85	2.19	3.17
mu	0.00	0.02	0.73	3.98	0.00	0.08	0.00	0.20	0.58	0.70
pa	4.52	23.11	9.51	9.61	4.46	6.87	8.01	6.37	7.37	18.59
pi	4.67	4.58	5.32	4.86	0.34	4.37	5.27	5.20	4.91	4.88
se	1.11	1.44	2.12	2.58	3.47	1.80	1.29	2.15	2.15	2.50
so	3.00	3.12	2.92	3.08	2.22	2.67	3.19	4.26	2.22	2.64
sh	8.46	8.37	5.98	6.35	1.12	6.50	7.42	7.73	6.50	6.58
vo	3.63	4.35	3.91	4.69	2.76	3.40	2.56	4.20	3.87	4.11

Табела 10.2. приказује стандардну девијацију за тачност класификације различитих алгоритма за оригинални и редуковани скуп података уз помоћ PCA методе. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе председекцију атрибута, осим у случају IBk и RBF алгоритама. IBk и RBF алгоритми са методама претходног учења имају генерално веће вредности за стандардну девијацију за тачност класификације од стандардног алгоритма IBk.

На сликама 10.3, 10.4. и 10.5. приказана је апсолутна разлика у вредностима стандардне девијације за тачност класификације различитих алгоритма за оригинални и редуковани скуп података уз помоћ PCA методе. Ако је вредност на сликама приближна нули, онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, у позитивном и негативном смеру, то је и веће одступање између стандардних девијација. Најмање одступање у стандардној девијацији у односу на оригинални скуп података, показује *Naïve Bayes* алгоритам, док највеће одступање има IBk алгоритам.



Слика 10.4: Стандардна девијација за тачност IBk_P минус IBk и Bay_P минус Bay



Слика 10.5: Стандардна девијација за тачност SVM_P минус SVM и J48_P минус J48

У табели 10.3. приказано је потребно време за тренинг у секундама различитих алгоритма за класификацију који користе оригинални и редуковани скуп података уз

помоћ PCA методе. Потребно време за тренинг података IBk класификатора за све оригиналне скупове података износи 0.00 секунде, док је за IBk са PCA оно веће. Потребно време за тренинг података *Naïve Bayes* класификатора за све оригиналне скупове података износи мање од 0.01 секунде, док је за *Naïve Bayes* са PCA оно такође веће.

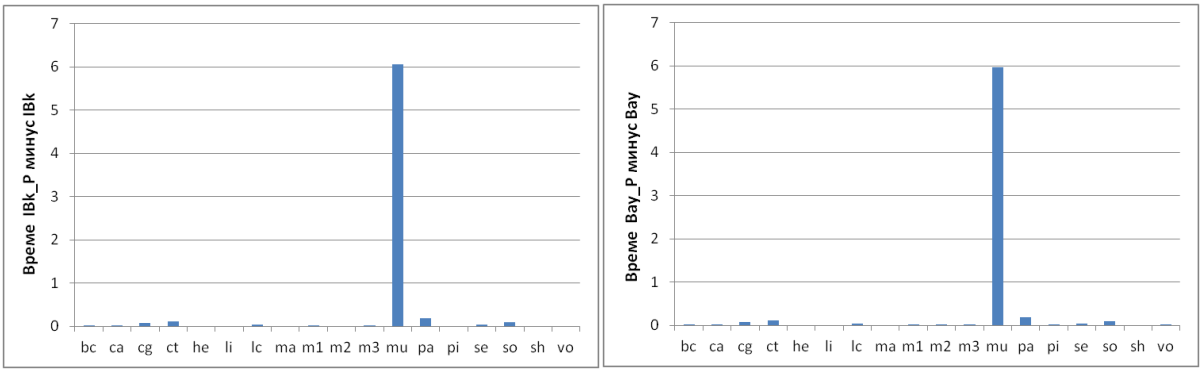
Табела 10.3. Потребно време за тренинг (у секундама) различитих класификатора за оригинални и редуковани скуп података уз помоћ PCA

Скуп	IBk	IBk_P	Bay	Bay_P	SVM	SVM_P	J48	J48_P	RBF	RBF_P
bc	0.00	0.01 -	0.00	0.01 -	0.02	0.07 -	0.00	0.03 -	0.01	0.03 -
ca	0.00	0.03 -	0.00	0.03 -	0.14	0.26 -	0.01	0.07 -	0.03	0.08 -
cg	0.00	0.07 -	0.00	0.08 -	0.51	0.89 -	0.01	0.19 -	0.05	0.25 -
ct	0.00	0.12 -	0.01	0.13 -	3.74	0.66 +	0.10	0.27 -	0.36	0.41
he	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.01	0.01	0.01
li	0.00	0.00	0.00	0.00	0.03	0.02	0.00	0.00	0.01	0.01
lc	0.00	0.04 -	0.00	0.04 -	0.01	0.04 -	0.00	0.04 -	0.00	0.04 -
ma	0.00	0.00	0.00	0.00	0.13	0.10 +	0.01	0.01	0.02	0.03 -
m1	0.00	0.01	0.00	0.01 -	0.04	0.14 -	0.00	0.02 -	0.01	0.03 -
m2	0.00	0.00	0.00	0.01 -	0.05	0.10 -	0.00	0.02 -	0.01	0.03 -
m3	0.00	0.01 -	0.00	0.01 -	0.03	0.12 -	0.00	0.02 -	0.01	0.03 -
mu	0.00	6.06 -	0.00	5.96 -	3.73	19.52 -	0.03	7.60 -	0.51	8.19 -
pa	0.00	0.18 -	0.00	0.19 -	0.03	0.36 -	0.01	0.23 -	0.02	0.23 -
pi	0.00	0.00	0.00	0.01	0.15	0.10 +	0.01	0.02 -	0.03	0.03
se	0.00	0.04 -	0.01	0.05 -	3.75	0.55 +	0.08	0.16 -	4.04	3.58
so	0.00	0.09 -	0.00	0.09 -	0.75	1.14 -	0.01	0.26 -	0.79	1.26 -
sh	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.01	0.01	0.02 -
vo	0.00	0.00	0.00	0.01 -	0.02	0.03 -	0.00	0.01 -	0.02	0.03 -

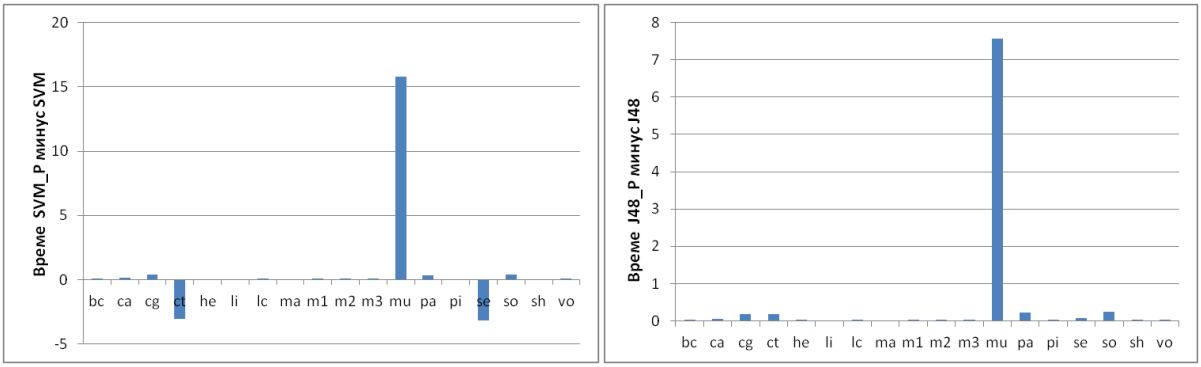
За потребно време тренинга података SVM класификатора за све оригиналне скупове података оно износи мање од 3.75 секунде, док је за SVM са PCA оно веће, осим у скуповима података *ct*, *li*, *ma*, *pi* и *se*. Потребно време за тренинг података J48 класификатора за све оригиналне скупове података износи мање од 0.10 секунди, док за J48 са PCA је оно такође нешто веће. Потребно време за тренинг података RBF класификатора за све оригиналне скупове података износи мање од 4.04 секунде, док је за RBF са PCA оно веће, осим у скупу података *se*.

На сликама 10.6, 10.7. и 10.8. приказана је апсолутна разлика у потребном времену за тренинг различитих алгоритма на основном скупу података и истих алгоритама са PCA методом за редукцију димензионалности података. Код три алгоритма, IBk, *Naïve Bayes* и J48 у свим скуповима података PCA метода је показала

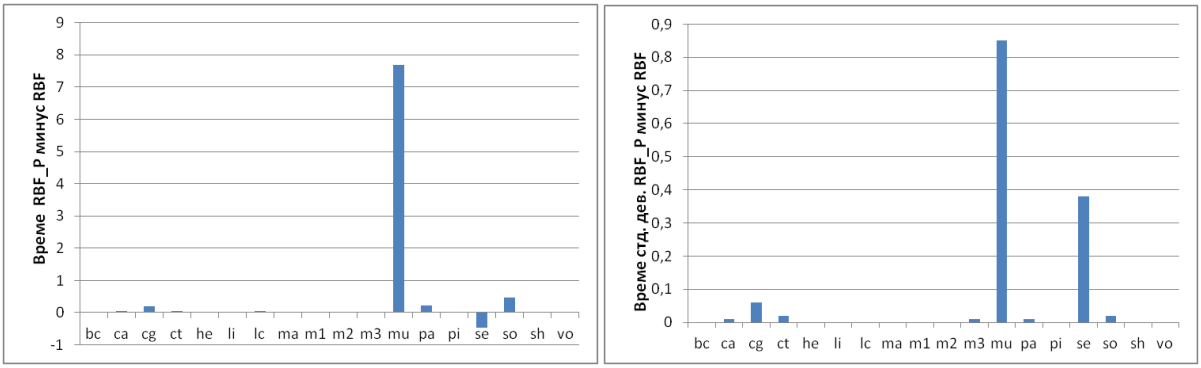
нешто лошије или исте резултате за потребно време за тренинг. Резултати су били и статистички лошији код ових алгоритама и то: за IBk у 10 скупова података, за *Naïve Bayes* у 13 скупова података и за J48 у 14 скупова података.



Слика 10.6: Време тренинга IBk_P минус IBk и Bay_P минус Bay (у секундама)



Слика 10.7: Време тренинга SVM_P минус SVM и J48_P минус J48 (у секундама)



Слика 10.8: Време тренинга RBF_P минус RBF (у секундама) и стандардна девијација за време RBF_P минус RBF

Алгоритам SVM је у 7 случаја показао исте или боље резултате за потребно време за тренинг од SVM алгоритма на основном скупу података, а у 4 скупа података резултати су били и статистички бољи. Алгоритам RBF је у 1 случају показао исте или боље резултате за потребно време за тренинг од RBF алгоритма на основном скупу података, али у том скупу података резултат није био и статистички бољи.

Коришћењем PCA методе, можемо да закључимо да је SVM алгоритам у највећем броју случаја довео до статистички бољих резултата за потребно време за тренинг на посматраним скуповима података. Такође, можемо уочити да PCA метода није довела до значајнијег повећања потребног времена за тренинг података, у односу на методу претходног учења.

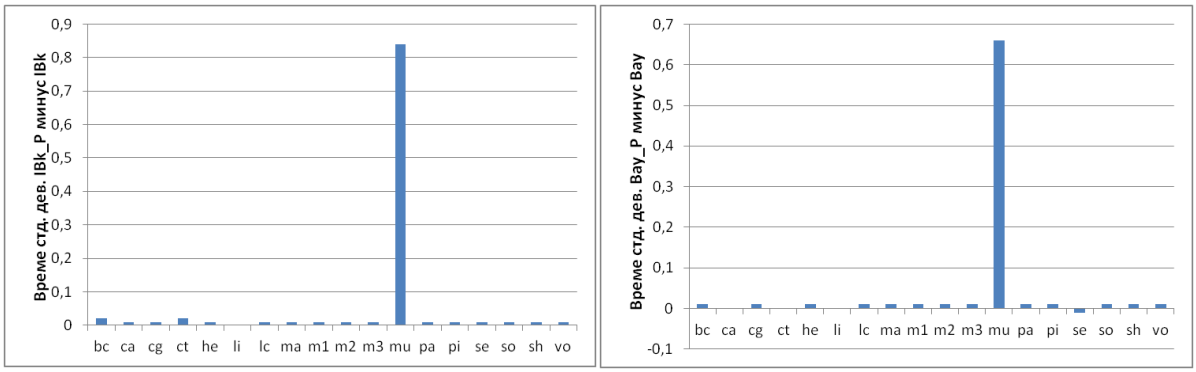
Табела 10.4. приказује стандардну девијацију за време тренинга различитих алгоритма за оригинални и редуковани скуп података уз помоћ PCA методе. Из табеле се може видети да се стандардне девијације генерално не разликују пуно између стандардног алгоритма и алгоритама који користе екстракцију атрибута. Нешто веће вредности за стандардну девијацију за време тренинга имају сви алгоритми за један сет података *mu*.

Табела 10.4. Стандардна девијација за потребно време за тренинг (у секундама) различитих класификатора за оригинални и редуковани скуп података уз помоћ PCA

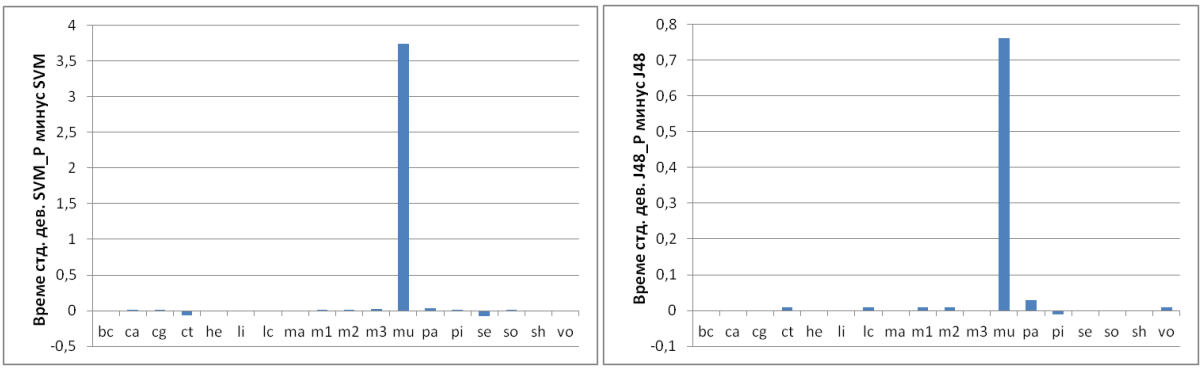
Скуп	IBk	IBk_P	Bay	Bay_P	SVM	SVM_P	J48	J48_P	RBF	RBF_P
bc	0.00	0.02	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ca	0.00	0.01	0.00	0.00	0.01	0.02	0.01	0.01	0.01	0.02
cg	0.00	0.01	0.00	0.01	0.02	0.03	0.01	0.01	0.01	0.07
ct	0.00	0.02	0.01	0.01	0.14	0.07	0.01	0.02	0.08	0.10
he	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
li	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01
lc	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.01	0.01	0.01
ma	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.00	0.01	0.00	0.01	0.01	0.02	0.00	0.01	0.01	0.01
m2	0.00	0.01	0.00	0.01	0.01	0.02	0.00	0.01	0.01	0.01
m3	0.00	0.01	0.00	0.01	0.01	0.03	0.00	0.00	0.00	0.01
mu	0.00	0.84	0.01	0.67	0.07	3.81	0.00	0.76	0.07	0.92
pa	0.00	0.01	0.00	0.01	0.01	0.04	0.01	0.04	0.01	0.02
pi	0.00	0.01	0.00	0.01	0.01	0.02	0.01	0.00	0.01	0.01
se	0.00	0.01	0.01	0.00	0.16	0.08	0.01	0.01	1.30	1.68
so	0.00	0.01	0.00	0.01	0.07	0.08	0.01	0.01	0.09	0.11
sh	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
vo	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.01	0.01	0.01

На сликама 10.8, 10.9. и 10.10. приказана је апсолутна разлика у вредностима стандардне девијације за време тренинга различитих алгоритма за оригинални и редуковани скуп података уз помоћ PCA методе. Ако је вредност на сликама

приближна нули, онда се стандардне девијације много не разликује, а уколико она више одступа од нуле, то је и веће одступање између стандардних девијација. Највеће одступање у стандардној девијацији у односу на оригинални скуп података показује SVM алгоритам.



Слика 10.9: Стандардна девијација за време IBk_P минус IBk и Bay_P минус Bay



Слика 10.10: Стандардна девијација за време SVM_P минус SVM и J48_P минус J48

ЈЕДАНАЕСТИ ДЕО

11. ДИСКУСИЈА РЕЗУЛТАТА И ДАЉА ИСТРАЖИВАЊА

У последњем делу докторске дисертације, биће дат резиме рада, а потом и закључци разматрања о утицају претходне селекције атрибута на класификацијске перформансе алгоритама надзираног учења. На крају, биће приказани правци могућих даљих истраживања у овој области.

11.1. Резиме

Основна хипотеза докторске дисертације је да је могуће знатно побољшати перформансе система за индуктивно учење правила у проблемима класификације, применом различитих метода и техника редукције димензионалности података. Да би се доказала постављена хипотеза, имплементиране су и емпиријски тестиране различите методе и технике редукције димензионалности података.

У дисертацији се разматра проблем редукције димензиналности података, где се селекција атрибута дефинише као процес који бира минимални подскуп M атрибута из изворног скупа N атрибута, тако да је простор атрибута оптимално смањен према одређеном критерију оцењивања. Проналажење најбољег подскупа атрибута је обично нерешив проблем и многи проблеми везани за избор атрибута су се показали да су NP-тешки. Сви атрибути могу бити важни за неке проблеме, али за нека циљана истраживања, само мали подскуп атрибута је обично релевантан. Алгоритме за селекцију атрибута смо поделили на филтере, методе претходног учења и уграђене приступе. Филтер методе оцењују квалитет одабраних атрибута независно од алгоритма за класификацију, док су методе претходног учења методе које захтевају примену класификатора (који би требао бити трениран на одређеном подскупу атрибута) за процену квалитета. Уграђене методе обављају избор атрибута током учења оптималних параметара (за на пример, неуронске мреже тежине између улазног и скривеног слоја).

Главни циљ овог рада је проверити утицај различитих филтер метода, метода претходног учења, уграђених метода и екстракције атрибута на тачност класификације.

Експериментална истраживања су спроведена уз коришћење вештачких и природних скупова података. Експериментални резултати показују да примењене методе ефикасно доприносе откривању и елиминисању небитних, редундатних података, као и шума у подацима. У многим случајевима описане методе претходне селекције атрибута одабирају релевантне атрибуте у скуповима података, и доприносе већој тачности класификације. Експерименталним истраживањем смо доказали следеће посебне хипотезе (докази се налазе у осмом, деветом и десетом делу дисертације):

- Прва посебна хипотеза која је доказана у раду гласи да примена претходне селекције атрибута уз помоћ метода филтрирања, претходног учења, уграђених метода и метода екстракције атрибута, код великог броја алгоритама за класификацију доводи до смањења негативних ефеката високе димензионалности података.
- Друга посебна хипотеза која је доказана у раду гласи да прецизно примењене методе претходне селекције атрибута доприносе повећању квалитета генерализације, јер се смањује вероватноћа претераног подешавања модела према тренирајућим подацима.
- Трећа посебна хипотеза која је доказана у раду гласи да претходна селекција атрибута уз помоћ метода филтрирања, претходног учења, уграђених метода и метода екстракције атрибута доводи у неким случајевима до значајног смањења времена за изградњу модела.
- Четврта посебна хипотеза која је доказана у раду гласи да применом метода филтрирања, претходног учења, уграђених метода и метода екстракције атрибута у оквиру система за индуктивно учење, могуће је у неким случајевима значајно побољшати тачност постојећих метода учења.

Генерално, од три групе метода, најбоље резултате везано за тачност класификације показале су методе претходног учења. Уопштено, недостатак метода филтрирања који вреднују појединачне атрибуте је немогућност детекције редундантних атрибута, због чега корелисаност неког атрибута с другим резултира сличном оценом вредности за оба атрибута, па ће по правилу оба атрибута бити прихваћена или одбачена. Следећи недостатак ових метода је да је увртавање атрибута у коначни подскуп препуштено спољним критеријима прага вредности или броја атрибута.

Методе филтрирања које вреднују подскупове атрибута су временски захтевније од филтера који вреднују појединачне атрибуте, јер постоји потреба претраживања подскупова атрибута. Међутим, ови захтеви су неупоредиво мањи у поређењу са методама претходног учења јер се у сваком кораку претраживања израчунава само вредност хеуристике вредновања, а није потребно више пута позивати алгоритам машинског учења.

Генерално, одабир атрибута методама филтрирања траје знатно краће у поређењу са методама претходног учења, посебно кад су у питању скупови података са већим бројем атрибута, због чега су методе филтрирања често практичније решење за анализу података од других метода. Методе филтрирања се због независности о алгоритму машинског учења могу користити у комбинацији са било којом техником моделирања података, за разлику од метода претходног учења који се морају поново изводити при свакој промени циљне технике моделирања.

Код метода претходног учења најважнији недостатак је спорост при извођењу условљена позивањем циљног алгоритма машинског учења више пута, због чега овим методама не одговарају обимни скупови података за учење са већим бројем атрибута.

Сматра се да методе претходног учења омогућују постизање нешто бољих перформанси класификације, због тесне повезаности с циљним алгоритмом машинског учења. Ово уједно може представљати и опасност јер претерано прилагођавање скупа за учење циљном алгоритму може нагласити његове недостатке.

Метода екстракције атрибута уз помоћ PCA методе очекује да ће већина нових варијабли чинити шум и имати тако малу варијансу да се она може занемарити, на основу чега се из великог броја изворних варијабли креира тек неколико главних компоненти које носе већину информација и чине главни облик. Међутим, има ситуација када то није тако, и то у случају када су изворне варијабле некорелисане, тада анализа не даје повољне резултате. Када су изворне варијабле високо позитивно или негативно корелисане могу се постићи најбољи резултати. Још један проблем код ове методе је немогућност смислене интерпретације главних компонената.

11.2. Закључци

У раду показујемо да нема једне најбоље методе за редукцију димензионалности података, и да избор зависи од особина посматраног скупа података и примењених класификатора. У практичним проблемима, једини начин како би били сигурни да је највиша прецизност добијена је тестирање датог класификатора са више различитих подскупова атрибута, добијених различитим методама за претходну селекцију атрибута. Спроведена истраживања у надгледаном учењу, покушавају дати увид у предности и ограничења различитих метода претходне селекције атрибута. Са оваквим увидом и предзнањем за одређени конкретни проблем, стручњаци могу одабрати које методе треба применити. Такав је случај са неким од метода претходне селекције атрибута, које могу да побољшају (или да не деградирају) извршење алгоритама машинског учења, док у исто време постижу смањење броја атрибута који се користе у учењу. Неке од приказаних метода, имале су проблем код избора релевантних атрибута, када у подацима постоји снажна интеракција између атрибута, или када имамо скупове података са оскудним бројем инстанци.

11.3. Даља истраживања

Уочено је да неке од метода предходне селекције атрибута имају проблем да изаберу атрибут који има локалне предиктивне могућности, јер се дешава да су засењени од стране атрибута који имају јаке, на глобалном нивоу предиктивне могућности. Ако је број таквих атрибута већи, онда се може појавити и кумулативни ефекат. У тим случајевима, може се допустити редунданса у скупу података, ако она нема негативне ефекте на алгоритме надзираног учења. Обично ова редунданса нема негативне ефекте на C4.5 и IB1, али може имати на *Naïve Bayes*.

Било би интересантно применити неке од техника у решавању проблема откривања локалног нивоа предиктивности атрибута. Те технике би могле побољшати класификацијске перформансе алгоритама учења. Такав приступ захтева коришћење одређеног алгоритма учења и одговарајуће технике која би у комбинацији са методама предходне селекције атрибута резултирала хибридним системом.

Атрибути изабрани од стране метода претходне селекције атрибута углавном представљају добру основу за формирање одговарајућег подскупа атрибута. Било би

интересантно истражити како методе претходног учења реагују, ако користе као почетни подскуп атрибута онај које је одабрала нека од метода филтрирања. То значи, да би се у овом случају, уместо претраживања унапред и у назад, када се користи у стартној позицији празан или потпун скуп атрибута, користио почетни подскуп атрибута које је изабрала одговарајућа метода филтрирања. На тај начин би се смањило потребно време за тражење одговарајућег подскупа методама претходног учења. У нашим експерименталним резултатима смо показали да се методама претходног учења добија у највећем броју случаја боља тачност класификације, али да је потребно време за извршење ове методе велико. Такође, овај приступ може побољшати перформансе класификације код метода претходног учења када методе раде са мањим скуповима где се добијају мање тачности класификације, јер се дешава да у тим случајевима постану заробљене у локалном максимуму.

ЛИТЕРАТУРА

N. Abe, M. Kudo, *Entropy criterion for classifier-independent feature selection*, Lecture Notes in Computer Science, Volume 3684/2005, 689-695, 2005.

D. Aha, *Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms*, International Journal of Man-Machine Studies, Volume 36, Issue 2, pp. 267–287, Academic Press Ltd, London, UK, Feb. 1992.

H. Almuallim, T. G. Dietterich, *Learning with many irrelevant features*, Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), pp. 547-552, Anaheim, CA: AAAI Press, 1991.

D. Ayres de Campos et al., *SisPorto 2.0 A Program for automated analysis of cardiotocograms*, J Matern Fetal Med 5:311-318, 2000.

D. Ayres de Campos, P. Sousa, A. Costa, J. Bernardes, *Omniview-SisPorto® 3.5—a central fetal monitoring station with online alerts based on computerized cardiotocogram ST event analysis*, J. Perinat. Med. 2008; 36:260-4.

M. Ben-Bassat, *Pattern recognition and reduction of dimensionality*, In P. R. Krishnaiah and L. N. Kanal, editors, Handbook of statistics-II, pp 773-791, North Holland, 1982.

A.L. Blum, R.L. Rivest, *Training a 3-node neural networks is NP-complete*, Neural Networks, 5:117-127, 1992.

A.I. Blum, P. Langley, *Selection of relevant features and examples in machine learning*, Artificial Intelligence, vol 97, 1997, 245-271.

L. Breiman, J.H. Friedman, R.H. Olshen, C.J. Stone, *Classification and regression trees*, Wadsworth and Brooks, Monterey, CA, 1984.

D.S. Broomhead, D. Lowe, *Multivariate functional interpolation and adaptive networks*, Complex Systems, 2:321-355, 1988.

R. Caruana, D. Freitag, *Greedy attribute selection*, In Proceedings of International Conference on Machine Learning (ICML-94), Menlo Park, California, 1994, AAAI Press/The MIT Press, 28–36.

G. Cestnik, I. Konenenko, I. Bratko,I., *Assistant-86: a knowledge-elicitation tool for sophisticated users*, In I.Bratko & N.Lavrac (Eds.) Progress in Machine Learning, 31-45, Sigma Press, 1987.

C. Chang, C. Lin, *LIBSVM: a Library for Support Vector Machines*, 2001. http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.

J.G. Cleary, L.E. Trigg, *K*: An instance-based learner using an entropic distance measure*, In Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, 1995.

S. Das, *Filters, wrappers and a boosting-based hybrid for feature selection*, In Proceedings of the Eighteenth International Conference on Machine Learning, 2001.

M. Dash, H. Liu, *Feature selection methods for classifications*, Intelligent Data Analysis: An International Journal, 1(3), 1997.

M. Dash, H. Liu, J. Yao, *Dimensionality reduction of unsupervised data*, In Proceedings of the Ninth IEEE International Conference on Tools with AI (ICTAI'97), November, 1997, Newport Beach, California, 1997, IEEE Computer Society, 532–539.

M. Dash, H. Liu, *Handling large unsupervised data via dimensionality reduction*, In Proceedings of 1999 SIGMOD Research Issues in Data Mining and Knowledge Discovery (DMKD-99) Workshop, 1999.

P. Diaconis, B. Efron, *Computer-intensive methods in statistics*, Scientific American, Volume 248, 1983.

J. Doak, *An evaluation of feature selection methods and their application to computer security*, Technical report, Davis CA: University of California, Department of Computer Science, 1992.

W. Duch, R. Adamczak, K. Grabczewski, *A new methodology of extraction, optimization and application of crisp and fuzzy logical rules*, IEEE Transactions on Neural Networks, vol. 12, pp. 277-306, 2001.

J. G. Dy, C.E. Brodley, *Feature subset selection and order identification for unsupervised learning*, In Proceedings of the Seventeenth International Conference on Machine Learning, 2000, 247–254.

B. Efron, R. Tibshirani, *An introduction to the bootstrap*, Chapman and Hall, London, 1993.

M. Elter, R. Schulz-Wendtland, T. Wittenberg, *The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process*, Medical Physics 34(11), pp. 4164-4172, 2007.

U.M. Fayyad, K.B. Irani, *The attribute selection problem in decision tree generation*, In AAAI-92, Proceedings of the Ninth National Conference on Artificial Intelligence, AAAI Press/The MIT Press, 1992, 104–110.

E. Fix, J.L. Hodges, *Discriminatory analysis; non-parametric discrimination: consistency properties*, Technical Report 21-49-004(4), USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

T. Fletcher, *Support Vector Machines Explained*, 2009, http://www.tristanfletcher.co.uk/SVM%20Explained.pdf.

A. Frank, A. Asuncion, *UCI Machine learning repository* [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science, 2010.

M.A. Hall, L.A. Smith, *Practical feature subset selection for machine learning*, Proceedings of the 21st Australian Computer Science Conference, 181–191, 1998.

Mark A. Hall, *Correlation-based feature selection for machine learning*, The University of Waikato, Doctoral disseration, 1999.

R.C. Holte, *Very simple classification rules perform well on most commonly used datasets*, Machine Learning, 11:63-91, 1993.

Z.Q. Hong, J.Y. Yang, *Optimal discriminant plane for a small number of samples and design method of classifier on the plane*, Pattern Recognition, Vol. 24, No. 4, pp. 317-324, 1991.

A. Hutchinson, *Algorithmic learning*, Clarendon Press, Oxford, 1993.

A. Jakulin, I. Bratko, *Analyzing attribute dependencies*, Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003), Cavtat-Dubrovnik, Croatia, September 22-26, 2003.

A. Jakulin, I. Bratko, *Testing the significance of attribute interactions*, Proceedings of the Twenty-first International Conference on Machine Learning (ICML-2004), Eds. R. Greiner and D. Schuurmans, pp. 409-416, Banff, Canada, 2004.

П. Јаничић, М. Николић, *Вештачка интелигенција*, Математички факултет у Београду, pp. 161, 2010.

M.V. Johns, *An empirical Bayes approach to non-parametric two-way classification*, Studies in item analysis and prediction, Stanford University Press, Palo Alto, 1961.

G.H. John, R. Kohavi, K. Pfleger, *Irrelevant feature and the subset selection problem*, In W.W. and Hirsh H. Cohen, editor, Machine Learning: Proceedings of the Eleventh International Conference, New Brunswick, N.J., 1994, Rutgers University, 121–129.

Y. Kim, W. Street, F. Menczer, *Feature selection for unsupervised learning via evolutionary search*, In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, 365–369.

K. Kira, L.A. Rendell, *The feature selection problem: tradional methods and a new algorithm*, In: Proc. AAAI-92, San Jose, CA, 1992, 122-126.

R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco, 1995.

R. Kohavi, G.H. John, *Wrappers for feature subset selection*, Artificial Intelligence - Special issue on relevance archive, Volume 97 Issue 1-2, Dec. 1997, pp. 273 – 324.

R. Kohavi, F. Provost, *Glossary of terms*, Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process (volume 30, Number 2/3, February/March 1998).

D. Koller, M. Sahami, *Toward optimal feature selection*, In International Conference on Machine Learning, 1996, 284-292.

I. Kononenko, *Esimating attributes: analysis and extensions of Relief*, In Proceeding of the European Conference on Machine Learning, 1994.

S.B. Kotsiantis, *Supervised machine learning: a review of classification techniques*, Informatica 31(2007) 249-268, 2007.

M. Kubat, R. Holte, S. Matwin, *Machine learning for the detection of oil spills in satellite radar images*, Machine Learning, 30, 195–215, 1998.

N. Lavrac, D. Gamberger, H. Blockeel, L. Todorovski (Eds.), *Lecture notes in artificial intelligence*, Vol. 2838, Springer, pp. 229-240, 2003.

M.A. Little, P.E. McSharry, S.J. Roberts, D.A.E. Costello, I.M. Moroz, *Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection*, BioMedical Engineering OnLine 2007, 6:23, 26 June 2007.

M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, L.O. Ramig, *Suitability of dysphonia measurements for telemonitoring of Parkinson's disease*, IEEE Transactions on Biomedical Engineering in 2009, 56(4):1015-1022.

H. Liu, R. Setiono, *Chi2: Feature selection and discretization of numeric attributes*, Proc. IEEE 7th International Conference on Tools with Artificial Intelligence, 338-391, 1995.

H. Liu, R. Setiono, *A probabilistic approach to feature selection - a filter solution*, In L. Saitta, editor, Proceedings of International Conference on Machine Learning (ICML-96), July 3-6, 1996, Bari, Italy, 1996, San Francisco: Morgan Kaufmann Publishers, CA, 319–327.

H. Liu, H. Motoda, *Feature selection for knowledge discovery and data mining*, Kluwer Academic Publishers, 1998.

B.J.F. Manly, *Multivariate methods*, Chapman & Hall, London, UK, 1986.

R.S. Marko, K. Igor, *Theoretical and empirical analysis of relief and ReliefF*, Machine Learning Journal, 53:23–69. doi: 10.1023/A:1025667309714, 2003.

R.S. Michalski, R.L. Chilausky, *Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis*, International Journal of Policy Analysis and Information Systems, Vol. 4, No. 2, 1980.

P. Mitra, C. A. Murthy, S. K. Pal, *Unsupervised feature selection using feature similarity*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3):301–312, 2002.

В. Мишковић, *Индуктивно учење разумљивог знања на основу оскудних обучавајућих скупова*, докторска дисертација, Универзитет Сингидунум, Београд, 2008.

В. Мишковић, М. Милосављевић, *Уграђени методи селекције атрибута у алгоритмима индуктивног учења*, LIII конференција „ЕТРАН 2010“, Доњи Милановац, Србија, 2010.

J. Moody, C. Darken, *Fast learning in networks of locally-tuned processing units*, Neural Computation, 1:281-294, 1989.

T. Nakagawa, T. Harab, H. Fujitab, T. Iwasec, T. Endod, K. Horitae, *Automated contour extraction of mammographic mass shadow using an improved active contour model*, Elsevier, International Congress Series, Volume 1268, June 2004, pp 882–885.

И. Петровић, *Основе интелигентног управљања (системи управљања засновани на умјетним неуронским мрежама)*, Загреб, 2009.

J.H. Piater, E.M. Riseman, P.E. Utgoff, *Interactively training pixel classifiers*, Published in the International Journal of Pattern Recognition and Artificial Intelligence 13(2), 1999.

G. Piatetsky-Shapiro, W.J.E. Frawley, *Knowledge discovery in databases*, MIT Press, Cambridge, Mass., 1991.

J.C. Platt, *Fast training of Support Vector Machines using sequential minimal optimization*, Advances in kernel methods, Pages 185-208, MIT Press Cambridge, MA, USA, 1999.

F. Provost, T. Fawcett, *Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions*, In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97).

S. Russell, P. Norvig, *Artificial intelligence: a modern approach*, Second edition, Prentice Hall, 2003.

J.S. Schlimmer, *Concept acquisition through representational adjustment* (Technical Report 87-19), Doctoral disseration, Department of Information and Computer Science, University of California, Irvine, 1987.

S. Sharma, *Applied multivariate techniques*, New York: John Wiley and Sons, Inc, 1996.

W. Siedlecki, J. Sklansky, *On automatic feature selection*, International Journal of Pattern Recognition and Artificial Intelligence, 2:197–220, 1988.

J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, R.S. Johannes, *Using the ADAP learning algorithm to forecast the onset of diabetes mellitus*, In Proceedings of the Symposium on Computer Applications and Medical Care, pp. 261-265, IEEE Computer Society Press, 1988.

J. Swets, *Measuring the accuracy of diagnostic systems*, Science 240, 1285-1293, 1988.

L. Talavera, *Feature selection as a preprocessing step for hierarchical clustering*, In Proceedings of Internationl Conference on Machine Learning (ICML'99), 1999.

J. R. Taylor, *An introduction to error analysis: the study of uncertainties in physical measurements*, University Science Books, Sausalito, CA, 1999, pp. 128–129.

S.B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. De Jong, S. Dzeroski, S.E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R.S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, J. Zhang, *The MONK's problems - a performance comparison of different learning algorithms*, Technical Report CS-CMU-91-197, Carnegie Mellon University in Dec. 1991.

Ф. Уjевић, *Поступци анализе података у изградњи профила корисника услуга*, магистарски рад, Свеучилиште у Загребу, Факултет електротехнике и рачунарства, Загреб, 2004.

Д. Врањеш, *Моделирање трења примјеном RBF неуронских мрежа*, дипломски рад бр. 1364, Свеучилиште у Загребу, Факултет електротрхнике и рачунарства, Загреб, српањ 2003.

S.M. Weiss, C.A. Kulikowski, *Computer systems that learn*, Morgan Kaufmann Publishers, San Mateo, California, 1991.

J.R. Quinlan, *Induction of decision trees*, Machine Learning 1, 1986.

J.R. Quinlan, *Simplifying decision trees*, Int J Man-Machine Studies 27, Dec 1987, pp. 221-234.

J.R. Quinlan, *C4.5: Programs for machine learning*, San Mateo, Morgan Kaufman, 1993.

J.R. Quinlan, *Comparing connectionist and symbolic learning methods*, In Computational Learning Theory and Natural Learning Systems, Vol.1, MIT Press, Cambridge, 1994.

I.H. Witten, E. Frank, *Data mining: practical machine learning tools and techniques*, Second Edition, Morgan Kaufmann, San Francisco, 2005, pp. 17.

N. Wyse, R. Dubes, A.K. Jain, *A critical evaluation of intrinsic dimensionality algorithms*, In E.S. Gelsema and L.N. Kanal, editors, Pattern Recognition in Practice, pp 415–425, Morgan Kaufmann Publishers, Inc., 1980.

E. Xing, M. Jordan, R. Karp, *Feature selection for high-dimensional genomic microarray data*, In Proceedings of the Eighteenth International Conference On Machine Learning, 2001.

J. Yang, V. Honavar, *Feature subset selection using a genetic algorithm*, IEEE Intelligent Systems 13:44-49, 1998.

БИОГРАФИЈА СА ПУБЛИКАЦИЈАМА КАНДИДАТА

1. Биографија кандидата

Јасмина Новаковић је рођена 07.07.1965. године у Лозници. Електротехнички факултет Универзитета у Београду, одсек електроника, смер телекомуникације, завршила је 13.07.1989. године, са средњом оценом у току студија 8.15. Магистарске студије завршила је на Економском факултету, Универзитет у Београду, студијска група, Информациони системи и управљање. Магистраску тезу под називом "Систем информација за управљање међународним маркетингом" одбранила је 11.07.2000. године. Докторску дисертацију: "Информациони систем у функцији управљања јавним приходима", одбранила је на Факултету за пословне студије, Мегатренд универзитета, 2003. године.

Од 1990. до 1997. године ради у Центру за везу Савезног министарства иностраних послова. У Центру за везу је радила као програмер, програмски језик C, учествујући у пројекту пријема и слања криптообрађених података. У периоду 1997-2002. године ради у Управи за информатику Савезног министарства иностраних послова, као пројектант информационог система министарства. Јула 2002. године бива унапређена у шефа Одсека за пројектовање Дипломатско-конзуларног информационог система. Током периода 1997-2002. године, као предавач на Дипломатској академији Савезног министарства иностраних послова одржала је велики број предавања из области информатике, као и практичну обуку дипломатско-конзуларног особља за рад на рачунару. На Факултету за пословне студије у Београду ради од 2003. године, на Катедри за компјутерски инжењеринг, као доцент на предметима Пословна информатика, Пословне рачунарске апликације и Електронско пословање. Од 2005. године у сталном радном односу је на Факултету за државну управу и администрацију у Београду, као доцент на предметима Правна информатика, Канцеларијско пословање и Е-влада (магистарске студије). На поменутом факултету обављала је и послове продекана за наставу од октобра 2005. године. На Вишој школи за компјутерске науке, а потом Високој школи за компјутерске науке, ради као директор од 2007. године, до њеног прерастања у Факултет за компјутерске науке, где је била стално запослена као предавач до октобра 2009. године. На Факултету за компјутерске науке покрива следеће предмете: Увод у рачунарске апликације, Напредне рачунарске апликације и

Социјална и професионална питања. Запослена је на Факултету за државну управу и администрацију у Београду, од октобра 2009. године, а месец дана касније поново је бирана за продекана за наставу, на истоименом факултету. Поред поменутих факултета, где је била у сталном радном односу, и држала наставу из напред поменутих предмета, у периоду од 2003. године, до данас одржала је велики број предавања из области рачунарских наука на различитим институцијама Мегатренд универзитета. На Вишој пословној школи "Мегатренд" држала је наставу из предмета Пословна информатика у Београду, као и центрима ван Београда: Лозница, Чачак, Сомбор, Зрењанин и Суботица. На Факултету за пословне студије - Пожаревац, држала је наставу из следећих предмета: Пословна информатика, Пословне рачунарске апликације, Интелигентни системи у пословном одлучивању и Електронско пословање. Наставу је држала и на Факултету за пословне студије - Вршац, и то из следећих предмета: Анализа информационих система, Менаџмент информациони системи, Пројектовање информационих система, Послова информатика, Основе менаџмент информационих система, Информационе мреже у пословном окружењу и Електронско пословање. На Факултету за културу и медије - Београд, држала је наставу из предмета Пословна информатика, а на Високој пословној школи "Мегатренд" - Београд, Електронско пословање. Учествовала је у поступку акредитације Високе школе за компјутерске науке (као директор школе) и Факултета за компјутерске науке.

2. Публикације кандидата

Јасмина Новаковић је ванредни професор рачунарских наука, која се и у свом досадашњем научно-истраживачком раду бавила проблемима вештачке интелигенције, машинског учења, као и надзираним и ненадзираним учењима.

Објавила је велики број научних радова из научне области вештачке интелигенције, од чега су четири рада на SCI листи.

1. Референце међународног нивоа (публикације у међународним часописима):

1. Новаковић, J.; Ранков С.: *Classification Performance Using Principal Component Analysis and Different Value of the Ratio R*, International Journal of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. VI (2011), No. 2 (June), pp. 317-327, (М23 за 2009 и 2010 годину. 2009 IF=0,373; 2010 IF=0,650) Извор: http://www.kobson.nb.rs.proxy.kobson.nb.rs:2048/servisi.131.html?jid=386187
2. Новаковић, J.; Вељовић, А.: *Classifier Ensembles with Asymmetric Misclassification Costs in Medical Diagnosis*, Metalurgia International, ISSN 1582-2214, vol. XVII (2012), no. 1, pp.

114-122, (M23 за 2009 и 2010 годину. 2009 IF=0,173; 2010 IF=0,154) Извор: http://www.kobson.nb.rs.proxy.kobson.nb.rs:2048/servisi.131.html?jid=391820

3. Новаковић, Ј.; Вељовић, А.: *Classification of Human Tissue by the Electrical Bio-impedance with Multilayer Perceptron*, Metalurgia International, ISSN 1582-2214, vol. XVI (2011), no. 12, pp 140-146, (M23 за 2009 и 2010 годину. 2009 IF=0,173; 2010 IF=0,154) Извор: http://www.kobson.nb.rs.proxy.kobson.nb.rs:2048/servisi.131.html?jid=391820
4. Новаковић, Ј.; Вељовић, А.: *Credit Risk Evaluation Based on Supervised Learning Algorithms,* Metalurgia International, ISSN 1582-2214, vol. XVII (2012), no. 5, pp. 195-203, (M23 за 2009 и 2010 годину. 2009 IF=0,173; 2010 IF=0,154) Извор: http://www.kobson.nb.rs.proxy.kobson.nb.rs:2048/servisi.131.html?jid=391820
5. Бутиган, В.; Јанић, Е.; Новаковић, Ј.; Вељовић, А.: *Representation of Molecular Orbitals of C_2H_4 by Application of Cascade Symmetry,* Metalurgia International, ISSN 1582-2214, vol. XVII (2012), no. 5, pp. 66-72, (M23 за 2009 и 2010 годину. 2009 IF=0,173; 2010 IF=0,154) Извор: http://www.kobson.nb.rs.proxy.kobson.nb.rs:2048/servisi.131.html?jid=391820
6. Новаковић, Ј.; Штрбац, П.; Булатовић, Д.: *Toward Optimal Feature Selection Using Ranking Methods and Classification Algorithms***,** Yugoslav Journal of Operations Research, ISSN: 0354-0243, 21 (2011), Number 1, 119-135, DOI: 10.2298/YJOR1101119N, pp. 119-135**,** http://yujor.fon.rs/index.php/journal/issue/archive.
7. Новаковић, Ј.; Минић, М.; Вељовић, А.: *Classification Accuracy of Neural Networks with PCA in Emotion Recognition,* Theory and Applications of Mathematics & Computer Science, ISSN: 2067-2764, Vol 1, No 1 (2011), pp. 11-16, http://www.uav.ro/applications/se/journal/index.php/TAMCS/issue/view/1.
8. Кулић, Р.; Туба, М.; Новаковић, Ј.: *The Control Based on Lyapunov Adaptation Law to be Improved by Modified Kohonen Rule*, International Review of Aerospace Engineering (IREASE), ISSN 1973-7459, Vol. 1, N. 4, pp. 422-429, Naples, August 2008.

2. Референце националног нивоа у другим државама (публикације у страним националним часописима):

1. Штрбац, П.; Туба, М.; Новаковић, Ј.: *Suitability of an Upgraded Petri Net for Modeling of Systolic Architecture for Solving Differential Equations*, Bulletins for Applied Mathematics (BAM), CXIII, Nr 2396, pp. 121-128, Budapest, 2008. [Саопштено на конференцији Panonnian Applied Mathematical Meeting PC-155 Balatonalmadi, 28th May – 1st June 2008].
2. Новаковић, Ј.; Штрбац, П.; Туба, М.: *Criterion Selection for Ranking the Importance of Each Feature in Data Dimensionality Reduction*, Bulletins for Applied Mathematics (BAM), CXIII, Nr 2398, pp. 140-145, Budapest, 2008. [Саопштено на конференцији Panonnian Applied Mathematical Meeting PC-155 Balatonalmadi, 28th May – 1st June 2008].

3. Референце националног нивоа (публикације у домаћим часописима):

1. Туба, М., Станаревић, Н., Штрбац, П., Новаковић, Ј.: *Impact of Hash Function Non-uniformity on Digital Signature Security*, J. Math., Vol. 38, No. 3, 2008, pp 201-208.
2. Станојевић, Љ.; Вељовић, А.; Новаковић, Ј.; Еремија, З.: *Развој информационог система факултета,* Техника, YU ISSN 0040-2176, UDC:316.776:378.6=861, година LXII 2007, број 2, pp 14-18, Београд, 2007.
3. Његуш, А.; Новаковић, Ј.: *Утицај електронског пословања на индустрију осигурања*, Финансије, банкарство, ревизија, осигурање, часопис за теорију и праксу, Универзитет Сингидунум, ISSN 1820-0702, година III, број 1, pp 107-113, Београд, 2006.
4. Новаковић, Ј.: *Information System of Revenue Autorities in Serbia*, Објављено у Мегатренд ревији, Међународном часопису за примењену економију, број 1/04, Година I, ISSN 1820-4570, pp 197-218, Београд, 2004.

5. Глувачевић, Д.; Новаковић, Ј.: *Европске интеграције: Хармонизација фитосанитарне регулативе и модела институција у Србији*, Саопштено на међународном научном скупу - Радикалне промене у предузећима и привреди у условима глобализације, 28.11.2003, Београд. Објављено у Зборнику радова са међународног научног скупа - Радикалне промене у предузећима и привреди у условима глобализације, Мегатренд универзитет примењених наука у Београду, ISBN 86-7747-119-7, pp 489-498, 489-498, Београд, 2003.

4. Саопштења на међународним научним скуповима:

1. Новаковић, Ј.: *Bagging Algorithm for Pixel Classification*, 19 Телекомуникациони форум, Зборник радова Телфор 2011, ИСБН 978-1-4577-1498-6, pp 1348-1351, IEEE Catalog Number CFP 1198P-CDR, Београд, 22-24 новембар.
2. Новаковић, Ј.: *Speaker Identification in Smart Environments with Multilayer Perceptron*, 19 Телекомуникациони форум, Зборник радова Телфор 2011, ИСБН 978-1-4577-1498-6, pp 1418-1421, IEEE Catalog Number CFP 1198P-CDR, Београд, 22-24 новембар.
3. Новаковић, Ј.; Вељовић А.: *C-Support Vector Classification: Selection of Kernel and Parameters in Medical Diagnosis*, 9th IEEE International Symposium on Intelligent Systems and Informatics (SISY 2011), September 8-10, 2011, Subotica, Serbia.
4. Новаковић, Ј.; Вељовић А.: *Interpretation of Mammograms with Rotation Forest and PCA*, 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, May 19–21, 2011, Timişoara, Romania, pp. 571-575.
5. Новаковић, Ј.: *Accuracy and Performance of RBF Network Classifier with Wrapper Approach*, (Invited Talk), ISREIE 2010, ISSN 2065 2569, pp 3-12, Arad, Romania.
6. Новаковић, Ј.; Минић М.; Вељовић А.: *Neural Network and PCA for Feature Selections in Facial Expression Analysis*, ISREIE 2010, ISSN 2065 2569, pp 50-56, Arad, Romania.
7. Новаковић, Ј.; Минић, М., Вељовић, А.: *A Wrapper Approach for Selecting Features in Supervised Learning*, 11-th European Conference E-COMM-LINE 2010, ISBN-10: 973-1704-18-3, ISBN-13: 978-973-1704-18-0, September 27-28, 2010, pp 10, Bucharest, Romania.
8. Новаковић, Ј.: *Rule Induction Algorithms in breast cancer diagnosis*, 11-th European Conference E-COMM-LINE 2010, ISBN-10: 973-1704-18-3, ISBN-13: 978-973-1704-18-0, September 27-28, 2010, pp 15, Bucharest, Romania.
9. Новаковић, Ј.: *Generating Decision Rules for Credit risk Evaluation by C4.5 Decision Tree with Genetic Algorithms*, 10-th European Conference E-COMM-LINE 2009 September 28-29, 2009, pp 6, Bucharest, Romania.
10. Новаковић, Ј.: *Searching Collision for the Birthday Attack and Comparing Results for SHA Hash Functions*, 10-th European Conference E-COMM-LINE 2009 September 28-29, 2009, pp 5, Bucharest, Romania.
11. Кулић, Р.; Туба, М.; Новаковић, Ј.: *An Algorithm to Hold a Desired Robot Vehicle Distance from the Closest Obstacle,* 8th WSEAS International Conference on Applied Computer Science (ACS'08), Venice, November 21-23, 2008.
12. Туба, М.; Курдулија, Н.; Новаковић, Ј.; Simian, D.: *Modeling of the Hash Function Irregularity*, Conference Sibiu Proceedings, 2008.
13. Новаковић, Ј.; Бачанин Џакула, Н.: *Analyzing Static and Dynamic Content Centric Applications and Variations Model in Content Management System,* Proceedings of the XIII JISA DICG – ICT, фајл 04. html, pp 263-268, Херцег Нови, 8-14.06.2008.
14. Новаковић, Ј.; Суботић, М.: *Optimizing Data Warehouse Performance with Logical Data Model Changes*, Proceedings of the VII Southeast Europe Forum ICT (SEFICT), фајл 10.html, pp 70-76, Дубровник, 11-12.06.2008.
15. Новаковић, Ј.; Суботић, М.; Бачанин Џакула, Н.: *Content Management System*, 8 - th

European Conference E-COMM-LINE 2007, ISBN-10: 973-88046-6-3, ISBN-13: 978-973-88046-6-1, 53 / 4 pag, Bucharest, September 20, 21, 2007.

16. Бачанин Џакула, Н.; Новаковић, Ј.; Суботић, М.: *E-business Infrastructure*, 8 - th European Conference E-COMM-LINE 2007, Bucharest, ISBN-10: 973-88046-6-3, ISBN-13: 978-973-88046-6-1, 5 / 4 pag, September 20, 21, 2007.
17. Новаковић, Ј.; Бачанин Џакула, Н.; Суботић, М.: *OLAP in e-business*, 8 - th European Conference E-COMM-LINE 2007, Bucharest, ISBN-10: 973-88046-6-3, ISBN-13: 978-973-88046-6-1, 8 / 6 pag, September 20, 21, 2007.
18. Љумовић, И.; Бачанин Џакула, Н.; Новаковић, Ј.: *Risks of e-banking in Serbia*, 8 - th European Conference E-COMM-LINE 2007, Bucharest, ISBN-10: 973-88046-6-3, ISBN-13: 978-973-88046-6-1, 10 / 6 pag, September 20, 21, 2007.
19. Новаковић, Ј.; Станојевић, Љ.; Вељовић, А.: *OLAP for Taking Exams on Faculty*, ВИРТ-2007, ISSN 1993-405X, pp 128-135, Јалта, 17-21.09.2007.
20. Новаковић, Ј.; Његуш, А.; Бачанин Џакула, Н.: *Web Mining*, Proceedings of the VI Southeast Europe Forum ICT (SEFICT), фајл 25.html, pp 123-128, Дубровник, 5-10.6.2007.
21. Новаковић, Ј.; Бачанин Џакула, Н.: *Примена технике вештачке неуронске мреже у data mining-u*, Саопштено на Фестивалу информатичких достигнућа - ИНФОФЕСТ 2007, 23-29.09.2007, Будва. Објављено у Каталогу XIV Фестивала информатичких достигнућа, pp 276-281, Подгорица, 2007.
22. Његуш, А.; Новаковић, Ј.; Миланов, Г.: *Using Rijndael Symetric Algorythm for Data Encryption in Development of Custom Solutions for Authentication of Web Service Users that is a Part of the Subsystem for Activation and Licensing of the Pauk System*, IPSI 2005, Venice, Italy, November 10-13, 2005.

5. Саопштења на домаћим научним скуповима:

1. Новаковић, Ј.: *The Impact of Feature Selection on the Accuracy of Naïve Bayes Classifier*, 18 Телекомуникациони форум, Зборник радова Телфор 2010, ИСБН 978-86-7466-392-9, pp 1113-1116, Београд, 23-25 новембар.
2. Новаковић, Ј.; Минић, М.; Вељовић, А.: *Genetic Search for Feature Selection in Rule Induction Algorithms*, 18 Телекомуникациони форум, Зборник радова Телфор 2010, ИСБН 978-86-7466-392-9, pp 1109-1112, Београд, 23-25 новембар.
3. Новаковић, Ј.: *Emotion Recognition Using Facial Expressions in e-Learning System Based on Affective Computing*, Електронско учење на путу ка друштву знања, Универзитет Метрополитан, ИСБН 978-86-912685-3-4, pp 176-180, Београд, 7. октобар 2010.
4. Новаковић, Ј.; Штрбац, П.: *Impact of setting individual parameters of genetic algorithm on IB1 classifier*, 54. Конференција ЕТРАН-а, pp. 83, Доњи Милановац, 7 – 10. јуна 2010.
5. Штрбац, П.; Новаковић, Ј.: *Modeling, simulation and analysis based on Monte Carlo method implemented as an upgraded Petri Net*, 54. Конференција ЕТРАН-а, pp. 73, Доњи Милановац, 7 – 10. јуна 2010.
6. Новаковић, Ј.: *Using Information Gain Attribute Evaluation to Classify Sonar Targets*, 17 Телекомуникациони форум, Зборник радова Телфор 2009, pp 1351-1354, Београд, 24-26 новембар.
7. Новаковић, Ј.: *RBF Network with Genetic Algorithm for Feature Selection*, 17 Телекомуникациони форум, Зборник радова Телфор 2009, pp 1347-1350, Београд, 24-26 новембар.
8. Новаковић, Ј.; Штрбац, П.; Булатовић Д.: *Clasification Accuracy Using Entropy-based Indices For Feature Ranking And Selection*, 53. Конференција ЕТРАН-а, pp. 83, Врњачка Бања, 15-18. јуна 2009.
9. Штрбац, П.; Новаковић, Ј.; Булатовић Д.: *An Upgraded Petri Net Model Simulation And*

Analysis Of An Image Compression, 53. Конференција ЕТРАН-а, pp. 72, Врњачка Бања, 15-18. јуна 2009.

10. Туба, М.; Курдулија, N.; Штрбац, П.; Новаковић, Ј.; *Digital Signature Security and the Hash Function Irregularity*, 12th Serbian Mathematical Congress, University of Novi Sad, Faculty of Science and Mathematics, Department of Mathematics and Informatics, Book of Abstracts, pp 79, Novi Sad, August 28th - Septembar 2nd, 2008.
11. Новаковић, Ј.; Бачанин Џакула Н.: *Middle Tier Stratification in Multi-tier Client Server Model Architecture of Web Portal,* YU INFO Симпозијум о рачунарским наукама и информационим технологијама, Копаоник, 9-12.03.2008, Зборник радова на CD-у, ISBN 978-86-85525-03-2, фајл 147.pdf, Зборник апстраката, pp 101, Београд, 2008.
12. Бачанин Џакула, Н.; Новаковић, Ј.; Суботић, М.: *Системи online плаћања*, VII Међународна конференција Е-трговина 2007, Палић, 18-20.04.2007.
13. Новаковић, Ј.; Његуш, А.: *Пројектовање пословне интелигенције*, YUPMA, ISBN 978-86-86385 02-4, pp 225- 229, Златибор, 6-8.06.2007.
14. Његуш, А.; Новаковић, Ј.: *Потпуни животни циклус пројекта развоја апликација пословне интелигенције*, XXXIV међународном симпозијуму SYM-OP-IS 2007, pp 139-142, Златибор, Септембар 16-19, 2007.
15. Новаковић, Ј.; Бачанин Џакула Н.: *Примена виртуелне реалности у CIM – у*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, Копаоник, 11-14.03.2007, Зборник радова на CD-у, ISBN 978-86-85525-02-5, фајл 172.pdf, Зборник апстраката, pp 101, Београд, 2007.
16. Вељовић, А.; Новаковић, Ј.; Захорјански, М.: *Моделирање као основ развоја подсистема рачуноводства производног предузећа,* XXI INFOTEH, јуни 2006, Врњачка Бања.
17. Вељовић, А.; Новаковић, Ј.; Захорјански, М.: *Реинжењеринг информационог система у области књиговодства,* DQM, јуни 2006, Београд.
18. Бачанин Џакула, Н.; Суботић, М.; Новаковић, Ј.: *Смернице у формулисању стратегија е-пословања*, SymOrg X Међународни симпозијум Изазови европских интеграција, Златибор, 7-10 јуна 2006, ISBN 86-7680-086-3, pp 103, Београд, 2006.
19. Новаковић, Ј.: *Проблеми и њихово решење у ланцу понуда у е-пословању*, Саопштено на Фестивалу информатичких достигнућа - ИНФОФЕСТ 2006, 24-30.09.2006, Будва. Објављено у Каталогу XIII Фестивала информатичких достигнућа, pp 296-303, Подгорица, 2006.
20. Вељовић, А.; Његуш, А.; Новаковић, Ј.: *Развој софтвера за вредновање наставе на Мегатренд универзитету*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, 6-10.03.2006, Зборник радова на CD-у, ISBN 86-85525-01-2, Зборник апстраката , pp 36, Beograd, 2006.
21. Радивојевић, М.; Новаковић, Ј.; Његуш, А.: *Моделирање послова обрачуна зарада запослених*, Саопштено на Фестивалу информатичких достигнућа - ИНФОФЕСТ 2005, 25.09.-01.10.2005, Будва. Објављено у Каталогу XII Фестивала информатичких достигнућа, pp 73-81, Подгорица, 2005.
22. Новаковић, Ј.: *Развојне стратегије за апликације е-пословања*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, Копаоник, 7-11.03.2005, Зборник радова на CD-у, ISBN 86-85525-00-4, фајл 056.pdf, Зборник апстраката , pp 43, Beograd, 2005.
23. Новаковић, Ј.: *Електронска влада и унапређење услуга Пореске управе Србије*, Proceedings of the IX Conference JISA, pp 28-33, Херцег Нови, and Proceedings of the III Southeast Europe Forum ICT (SEFICT), pp 4-9, Дубровник, 14-19.2004.
24. Новаковић, Ј.: *Информациони систем конзуларних прихода*, Саопштено на Фестивалу информатичких достигнућа - ИНФОФЕСТ 2004, 26.09.-02.10.2004, Будва. Објављено у

Каталогу XI Фестивала информатичких достигнућа, pp 252-259, Подгорица, 2004.
25. Новаковић, Ј.; *Информациони систем у функцији управљања јавним приходима*, SymOrg IX Међународни симпозијум Менаџмент – кључни фактори успеха, Златибор, 6-10 јуна 2004, ISBN 86-7680-022-7, Београд, 2004.
26. Новаковић, Ј.: *Основне функције и подсистеми информационог система пореске управе*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, Копаоник, 8-12.03.2004, Зборник радова на CD-у, фајл 053.pdf, Зборник апстраката, pp 46, Београд, 2004.
27. Миленковић, М.; Новаковић, Ј.; Марјановић, В.: *ДКИС – Дипломатско - конзуларни информациони систем - Подсистем Дипломатске академије*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, 10-14.03.2003, Копаоник, 2003.
28. Вратоњић, Б.; Поповић, Н.; Новаковић, Ј.: *Концепт модела е-МИП-а заснован на моделу е-владе*, Proceedings of the VII Conference JISA, Херцег Нови, 03-08.06.2002.
29. Марјановић, В.; Делевић-Ђилас, М.; Миленковић, М.; Новаковић, Ј.: *Примена Е-Учења у савременој дипломатији*, Саопштено на Фестивалу информатичких достигнућа - ИНФОФЕСТ 2002, 22.09.-28.9.2002, Будва. Објављено у Каталогу XI Фестивала информатичких достигнућа, pp 85-90, Подгорица, 2002.
30. Новаковић, Ј.; Тамбурић, И.; Аврам, С.; Крстић, Д.: *ДКИС – Модел подсистема Дипломатског протокола*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, Копаоник, 11-15.03.2002, Зборник радова на CD-у, фајл 17.pdf, Београд, 2002.
31. Глувачевић, Д.; Новаковић, Ј.: *Предлог аутоматизације издавања фитосертификата у пољопривреди*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, Копаоник, 11-15.03.2002, Зборник радова на CD-у, фајл 31.pdf, Beograd, 2002.
32. Вратоњић, Б.; Миленковић, М.; Новаковић, Ј.: *Дипломатско-конзуларни информациони систем – Модел и концепт реализације*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, Копаоник, 19-23.03.2001, Зборник радова на CD-у, фајл 504_380.pdf, Зборник апстраката, pp 33, Ниш, 2001.
33. Новаковић, Ј.; Вратоњић, Б.: *Дипломатски и конзуларни информациони систем – Подсистем економских информација (ЕКИС)*, YU INFO Симпозијум о рачунарским наукама и информационим технологијама, Копаоник, 19-23.03.2001, Зборник радова на CD-у, фајл 507_383.pdf, Зборник апстраката, pp 35, Ниш, 2001.
34. Глувачевић, Д.; Новаковић, Ј.: *Модел Информационог система извештајно-прогнозних послова у области заштите биља Југославије*, 9. Саветовање о заштити биља, Златибор, 2000.