УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

UNIVERZITET U BEOGRADU

FAKULTET ORGANIZACIONIH NAUKA

Sanja P. Vuković

Model poslovne inteligencije zasnovan na zaključivanju na osnovu slučajeva i izboru mera sličnosti

doktorska disertacija

Beograd, avgust 2013.

Јове Илића 154, 11000 Београд, Србија, Тел.: (011) 3950-800, Факс: (011) 2461-221
ПИБ: 100383934, Матични број: 07004044, Текући рачун: 840-1344666-69
Е пошта: dekanat@fon.bg.ac.rs; Посетите: www.fon.bg.ac.rs

UNIVERZITET U BEOGRADU

FAKULTET ORGANIZACIONIH NAUKA

Sanja P. Vuković

Model poslovne inteligencije zasnovan na zaključivanju na osnovu slučajeva i izboru mera sličnosti

doktorska disertacija

Beograd, avgust 2013.

Mentor:	Prof. dr Boris Delibašić Vanredni profesor Univerzitet u Beogradu Fakultet organizacionih nauka
Članovi komisije:	Prof. dr Milija Suknović Redovni profesor Univerzitet u Beogradu Fakultet organizacionih nauka
	Prof. dr Dragan Radojević Naučni savetnik Institut Mihajlo Pupin
Datum odbrane:	_____

Zahvalnost

Zahvaljujem se profesoru Borisu Delibašiću za brojne savete, za razumevanje i strpljenje, za stručnu i uvek prisutnu pomoć tokom celokupnog studiranja, a naročito prilikom izrade doktorske disertacije.

Zahvaljujem se i svima koji su mi tokom proteklih godina bili izvor snage, a naročito mojim roditeljima, majci Vidi i ocu Peri. Pokušaću da moja dela budu odraz bar malog dela onoga što su oni učinili za mene, stoga ovu disertaciju posvećujem upravo njima.

Sanja Vuković

Dok ne pokušate ne znate šta možete učiniti.
Henry James

Biti poražen je često samo prolazno stanje. Odustajanje ga čini trajnim.
Marilyn vos Savant

Stvarna razlika između ljudi je u njihovoj energiji. Jaka volja, definisani cilj, nepobediva upornost, mogu da ostvare skoro sve.
Thomas Fuller

Model poslovne inteligencije zasnovan na zaključivanju na osnovu slučajeva i izboru mera sličnosti

REZIME

Predmet istraživanja ove doktorske disertacije je model za klasifikaciju, sa ciljem razvoja modela zasnovanog na zaključivanju na osnovu slučajeva i inteligentnom izboru mera sličnosti, koji će pomoći što precizniju klasifikaciju novih slučajeva.

Cilj je razmotriti da li domensko znanje, koje se može izraziti preko funkcija preferencija, može biti bolje iskorišćeno na takav način da poboljša prediktivne performanse sistema zaključivanja na osnovu slučajeva (ZOS). Dodatno, cilj je utvrditi da li model sa korišćenjem određenih mera sličnosti za kategoričke atribute, u kombinaciji sa korišćenjem funkcija preferencija za numeričke atribute, može dati bolje rezultate od tradicionalnog modela ZOS-a.

Merenje sličnosti između slučajeva je važan deo svakog ZOS modela. Tradicionalno, za merenje sličnosti u modelima ZOS se koristi Euklidova metrika, za numeričke varijable, odnosno funkcija preklapanja za kategoričke atribute. Teorija preferencija se takođe može iskoristiti za merenje sličnosti između slučajeva, naročito što ona pruža više mogućnosti u izražavanju preferencija donosilaca odluka. Dodatno, pokazaće se i rezultati klasifikacije modela u kojima su korišćene određene mera sličnosti za kategoričke atribute, a u kombinaciji sa korišćenjem funkcija preferencija za numeričke atribute.

Rad obuhvata i optimizaciju težina atributa, kao i rezultate predikcione moći modela u zavisnosti od broja najbližih suseda koji se uzimaju u obzir.

Genetski algoritam je korišćen za podešavanje parametara funkcija preferencija, kao i za optimizaciju težina atributa.

Model je testiran na tri različita skupa podataka o klijentima kojima treba razmotriti kreditni zahtev, dok se za procenu učinka prediktivnog modela koristi metodologija

desetostruke unakrsne validacije. Particije unakrsne validacije su se generisale korišćenjem takozvanog semena za slučajni izbor podataka.

Rezultati istraživanja pokazuju da predloženi pristupi sa primenom teorije preferencija, u svakom slučaju, bez obzira na primenjene mere sličnosti za kategoričke atribute, mogu nadmašiti rezultate tradicionalne K-NN klasifikacije. Sa druge strane, što se tiče modela u kojima se koriste funkcije preferencije, ali različite mere sličnosti za kategoričke atribute, pokazalo se da za svaki pojedinačni skup podataka postoji njemu najviše odgovarajuća tehnika u kojoj se pored funkcija preferencije koristi i određena mera sličnosti za kategoričke varijable.

Ključne reči: Zaključivanje na osnovu slučajeva; Funkcije preferencija; Mere sličnosti za kategoričke atribute; Klasifikacija; Genetski algoritam; Kredit skoring.

Naučna oblast: Menadžment

Uža naučna oblast: Modeliranje poslovnih sistema

UDK broj:

A model of business inteligence based on case-based reasoning and similatiry measures selection

ABSTRACT

The main subject of interest in this doctoral thesis is model of classification, with the common goal of developing a model based on case-based reasoning (CBR) and intelligent selection of similarity measures, which will contribute to a more precise classification of new cases.

It has been interesting to consider whether the domain knowledge, which can be expressed through preference functions, could be better exploited in such a way to improve the predictive performance of a CBR system.

Additionally, the goal is to consider whether the model that using certain similarity measures for categorical data in combination with preference functions for similarity measuring between numerical data, can show better results than the traditional k-NN CBR model.

Similarity measuring between the cases is an important part of each CBR model. Traditionally, in CBR models, for similarity measuring between numerical variables Euclidean metric is used, while for similarity measuring between categorical attributes overlap function is used. Preference theory functions could also be used for similarity measuring between the cases, particularly as they provide more opportunities to express preferences of the decision makers. Additionally, the results of classification models which use certain similarity measures for categorical attributes combined with the use of preference functions for numerical attributes, will be presented in this thesis.

This thesis includes the optimization of attribute weights, as well as the results of models' predictive power depending on the number of nearest neighbors which are taken into account.

A genetic algorithm is used for setting the parameters of each preference function, as well as to set the attribute weights.

The model has been evaluated on three different benchmark datasets of clients who require their credit application to be considered. Models' accuracy have been measured with 10-fold cross-validation test. Cross-validation folds are generating by using the seed for randomizing the data.

The experiment results show that the proposed approaches with preference theory functions can, in every case, outperform the traditional k-NN classifier, regardless of the applied similarity measures for categorical attributes. On the other hand, models which use preference theory functions, but different similarity measures for categorical data, showed that for each particular data set there was always a best suited technique, in which, besides the preference theory functions, an appropriate similarity measure for categorical attributes is also used.

Keywords: Case-based reasoning; Preference functions; Similarity measures for categorical attributes; Classification; Genetic algorithm; Credit scoring.

Scientific field: Management

Scientific subfield: Business system modelling

UDK:

Sadržaj

1. Uvod

Broj rizika sa kojima se banke suočavaju svaki dan je u stalnom porastu. Banke su finansijske organizacije koje pretvaraju rizik u profit i suštinski većina prihoda banaka se generiše iz kredita, odnosno kreditnih operacija. Kreditni plasmani su stoga jedan od najvažnijih generatora profita banke. Naravno, najveći rizik prilikom odobravanja kredita je da klijenti neće biti u stanju da ispune svoje obaveze prema banci i da će banka izgubiti sredstva.

Tokom poslednjih nekoliko decenija, beleži se brz rast i u dostupnosti i u korišćenju kredita. Nekada je odluka da se odobri kredit bila zasnovana na ljudskoj proceni kada je u pitanju ocena rizika nenaplativosti (Thomas, 2000). Međutim, rast tražnje za kreditima doveo je do većeg interesovanja za upotrebu formalnijih i objektivnijih metoda (opšte poznatih kao ”credit scoring”, tj. metode kreditnog bodovanja). Njihov cilj je da pomognu davaocima kredita da se odluče da li da odobre kredit podnosiocu zahteva (Akhavein i ostali, 2005;. Chye i ostali, 2004), a akcenat je kako na adekvatnosti odluke, tako i na brzini donošenja iste.

Kreditni skoring je problem klasifikacije. Kredit skoring modeli pomažu u odluci da li odobriti kredit novim kandidatima, a s obzirom na njihove karakteristike, kao što su starost, prihod i bračni status (Chen & Huang, 2003). Odobravanje kredita je veoma važan deo bankarskih aktivnosti, naročito što može doneti velike profite, iako sa druge strane, postoji značajan rizik uključen u donošenje odluka u ovoj oblasti, gde greške mogu biti veoma skupe za finansijske institucije (Zakrzewska, 2007).

Zbog gore navedenih razloga, donošenje odluka u vezi sa davanjem kredita jedan je od ključnih elemenata u politici svake banke. Ključni problem je da se napravi razlika između dobrih (onih koji će sigurno otplaćivati) i loših kreditnih kandidata (koji verovatno neće izvršavati svoje obaveze). To znači da se procena kreditnog rizika sastoji od izgradnje klasifikacionih pravila koja pravilno definišu klijente banke kao dobre ili loše otplatioce (Zakrzewska, 2007).

Dugi niz godina, odluka da li da se odobri kredit donosila se od strane kreditnih analitičara. Analitičari su obično morali da napišu pravila koja su koristili za procenu

kredibiliteta podnosioca zahteva za kredit, a koji se tiče otplate kredita. Kreditne odluke su potom donošene korišćenjem ovih pravila, pri čemu su neka od njih bila veoma restriktivna.

Sama metodologija kreditnog bodovanja se može koristiti za različite namene, kao što su: odobrenje kreditnih kartica, potrošačkih, stambenih kredita, manjih poslovnih kredita, kao i zahteva za osiguranje ili za ponovno kreditiranje. Osim toga, ova metodologija može da se koristi za povećanje odziva kada su u pitanju reklamne kampanje, itd. (Thomas, 2000). Ono što je bitno jeste da se pronađe način da se izgradi efikasan model klasifikacije kupaca/ klijenata, koji će što preciznije da predvidi njihovo ponašanje.

Postoji niz metoda koje se mogu koristiti za razvoj modela za ocenu kreditne sposobnosti. Neke od korišćenih metoda su statističke, dok se neke od njih oslanjaju na pristupe u kojima se primenjuje veštačka inteligencija. Statističke metode, često korišćene za kredit skoring, su: višestruka regresija (npr. Meyer & Pifer, 1970), diskriminaciona analiza (npr. Altman, 1968; Banasik i ostali, 2003), i logistička regresija (npr. Dimitras i ostali, 1996; Martin, 1977; Lee i ostali, 2002; Desai i ostali,1996), dok metode veštačke inteligencije uključuju induktivno učenje (npr. Han i ostali, 1996; Shaw & Gentry, 1998), veštačke neuronske mreže (npr. Boritz & Kennedy, 1995; Coakley & Brown, 2000; Jo & Han, 1996; Zhang i ostali, 1999; Lee & Chen, 2005; West, 2000), genetske algoritme (npr. Desai i ostali, 1997; Yobas i ostali, 2000; Huang i ostali, 2006, 2007), i veštačke imune sisteme (npr. Leung i ostali, 2007).
Od prvobitnog rada Schanka i Abelsona (1977), zaključivanje na osnovu slučajeva (ZOS) se uspešno primenjuje u mnogim oblastima, uključujući i ocenu kreditne sposobnosti (npr. Bryant, 1997; Buta, 1994; Wheeler & Aitken, 2000; Shin & Han, 2001).
Pored toga što se može uspešno primeniti na polju finansija, ZOS se može koristiti i u mnogim drugim oblastima, kao što su medicina i proizvodna industrija (npr. Hsu i ostali, 2004; Im & Park, 2007; Tseng i ostali, 2005), na probleme segmentacije (npr. Chen i ostali, 2010; Changchien & Lin, 2005; Chiu, 2002; Chun & Park, 2006), itd.

Zaključivanje na osnovu slučajeva (ZOS) podrazumeva da se novi problem (slučaj) rešava iskustvom, odnosno uzimanjem u obzir rešenja prethodno rešenih sličnih slučajeva. Nalaženje sličnih slučajeva novom slučaju je osnovi korak u paradigmi ZOS-a. Metod određivanja stepena sličnosti između slučajeva utiče na izbor sličnih slučajeva. Normalizacija indeksa (atributa koji opisuju slučaj) je od koristi u potrazi sa odgovarajućom merom sličnosti. ZOS-om se rešavaju problemi klasifikacije, odnosno ZOS podrazumeva proces odlučivanja koji ima za cilj da posmatranom objektu ili situaciji dodeli ispravnu alokaciju, odnosno jednu od predefinisanih kategorija ili klasa.

Postoji nekoliko razloga zbog kojih se veruje da je ZOS značajan metod za klasifikaciju. Prvo, ZOS se smatra ne-parametarskom metodom koja ne zahteva nikakvu pretpostavku distribucije podataka za ulazni slučaj, odnosno podaci ne moraju imati bilo kakve specifične osobine. Ova karakteristika omogućava ZOS-u da se primenjuje na većem broju problema u odnosu na statističke metode, kao što su regresija ili diskriminaciona analiza. Drugo, ZOS je tehnika učenja koja može da zadrži novi slučaj bez prerade, a ukoliko se proceni da je novi slučaj vredan pamćenja. Dodavanjem novog slučaja ažurira se prethodna baza slučajeva. Pored toga, ZOS je veoma jednostavan za primenu i može da barata i sa kompleksnim i nestrukturiranim odlukama veoma efektivno (Ahn i ostali, 2007).

Uprkos mnogim prednostima, postoje neki problemi koji se moraju rešavati u cilju projektovanja efikasnog sistema ZOS-a (Ahn & Kim, 2008):

- Kako izabrati odgovarajuću funkciju sličnosti koja će generisati odgovarajuću klasifikaciju na osnovu sačuvanih, prethodnih slučajeva iz baze?
- Kako izabrati odgovarajuće atribute i reprezentativne slučajeve?
- Kako odrediti težinu, tj. značaj svakog atributa, što je takozvani problem ponderisanja atributa?
- Kako odrediti optimalnu vrednost parametra k, ako se koristi algoritam k-najbližih suseda (k-NN)?
- Kako izračunati sličnost za kategoričke varijable koje, pored numeričkih atributa, takođe mogu da opisuju slučajeve?

Rađeno je dosta studija u kojima je pokušavano da se reše ovi problemi. Izbor odgovarajućih mera sličnosti, i izbor atributa, kao i određivanje njihovih pondera, bila su najpopularnija istraživačka pitanja kada je u pitanju faza pronalaženja najsličnijih slučajeva (npr. Wang & Ishii, 1997; Shin & Han, 1999; Kim & Han, 2001; Chiu i ostali, 2003; Ahn i ostali, 2007; Liao i ostali, 1998).

Određivanje sličnosti između slučajeva je važan deo svakog ZOS modela.

Sličnost između para objekata (slučajeva) predstavljenih sa dva indeksna vektora se obično određuje na bazi Euklidove funkcije rastojanja, ako se radi o numeričkim atributima. Za dva objekta se očekuje da su slični ako je vrednost Euklidove funkcije udaljenosti dva indeksna vektora mala. Euklidova norma predstavlja samo jedan od načina na koji je moguće meriti sličnost. Postoje, teorijski, beskonačno mnogo načina za računanje sličnosti, a što zavisi od situacije u kome se primenjuje (Suknović & Delibašić, 2010). Za pretraživanje baze slučajeva mogu se koristiti i funkcije preferencije (Delibašić, 2004). Pojedine mere sličnosti mogu se koristiti samo za numeričke podatke, dok se za kategoričke podatke koriste druge mere sličnosti. U ovom radu za merenje sličnosti između kategoričkih atributa predlažu se mere iz rada (Boriah i ostali, 2008).

Kao što je napomenuto, teorija preferencija takođe može da se koristi za merenje sličnosti između slučajeva, naročito što pruža više mogućnosti u iskazivanju preferencija donosilaca odluka. Li, Sun i Sun (2009) i Li i Sun (2010) predložili su kombinovanje ZOS-a i funkcija preferencija (funkcija višeg ranga) za predviđanje finansijskih nevolja i poslovnog neuspeha, respektivno.

Kada se govori o primeni funkcija preferencija u okviru ZOS, osnovni cilj je da se razmotri da li domensko znanje, izraženo preko funkcija preferencija, može biti bolje iskorišćeno na takav način da poboljša prediktivne performanse ZOS sistema.
Osnovna razlika između ZOS sistema sa funkcijama preferencija i tradicionalnog ZOS sistema je u mehanizmu računanja sličnosti. U ovom radu, polazna pretpostavka je da upotreba funkcija teorije preferencija u ZOS-u može da pokaže bolje rezultate nego

tradicionalni k-NN model, zasnovan na meri Euklidskog rastojanja, a kada je u pitanju problem odobravanja kredita.

Li, Sun i Sun (2009) su koristili odnose višeg ranga zasnovane na funkcijama preferencija iz metode Electre III, dok su Li i Sun (2010) projektovali hibridni ZOS sistem za predviđanje koji koristi sve raspoložive funkcije preferencije u pristupima višeg ranga, kao što su Electre, Promethee, i Oreste. U ovom radu će se za merenje sličnosti između slučajeva koristiti funkcije preferencije predložene metodom Promethee.

Li i Sun (2010) su koristili iterativan proces pokušaja i pogrešaka da bi identifikovali optimalni hibridni ZOS modul sa odgovarajućim funkcijama preferencija i njihovim parametrima. U ovom radu, genetski algoritam je korišćen za te svrhe.

U ovom radu će se takođe analizirati broj suseda koji se uzimaju u obzir za klasifikaciju, kao i uticaj atributa (karakteristika) i njihovih težina na tačnost predviđanja.

K-najbliži sused (k-NN) je jedna od najjednostavnijih tehnika za klasifikaciju. Prema k-NN pravilu klasifikacije (odlučivanja) novom objektu se dodeljuje klasa kojoj pripada većina njemu najbližih k suseda. Konkretnije, 1-NN pravilo klasifikacije novom objektu dodeljuje klasu kojoj pripada najbliži sused (najsličniji slučaj). K-NN pravilo odlučivanja je veoma blisko ZOS shemi (Bobrowski, 2012).

Problemi kredit skoringa mogu biti rešeni ZOS ili k-NN paradigmom. Karakteristike novog komitenta koji aplicira za kredit se koriste za identifikaciju (iz postojeće baze klijenata) najsličnijih aplikanata. Novom slučaju se dodeljuje klasa (odobren ili ne) kojoj pripada većina njemu najbližih suseda.

1.1. Definicija problema i predmet istraživanja

Predmet istraživanja u ovom radu je model za klasifikaciju, sa ciljem razvoja modela zasnovanog na zaključivanju na osnovu slučajeva i inteligentnom izboru mera sličnosti, koji će pomoći što precizniju klasifikaciju novih slučajeva. Model je testiran na klijentima kojima treba razmotriti kreditni zahtev. Kreditna funkcija je jedna od najvažnijih funkcija za banku, jer se najveći deo prihoda banke stvara kreditnim poslovima, odnosno pozajmljivanjem. Glavna opasnost pri davanju kredita je da klijent neće biti u stanju da ispuni svoje obaveze prema banci i da će banka time izgubiti sredstva. Rast potražnje za kreditima vremenom je doveo do povećane zainteresovanosti za korišćenjem formalnijih i objektivnijih metoda, opšte poznatih kao kredit skoring modeli. Time bi se institucijama koje plasiraju kredite pomoglo u odlučivanju da li da odobre kredit podnosiocu zahteva.

Problem odlučivanja da li podnosiocu zahteva odobriti kredit je tipični problem klasifikacije, čiji je zadatak da predvidi da li potencijalni klijent nosi povoljan ili nepovoljan kreditni rizik. Ovo predviđanje se bazira na karakteristikama klijenta i zavisi od prethodnog iskustva, npr. primeri poznatog ishoda prethodnih slučajeva. Bazična pretpostavka ovog procesa je da se istorijsko ponašanje reflektuje na buduće ponašanje.

Kredit skoring modeli pomažu u odlučivanju da li plasirati kredit podnosiocima zahteva, a razmatrajući različite karakteristike klijenata koji apliciraju za kredit, kao što su godine, bračni status i svrha kredita, finansijski podaci (npr. prihodi koje pojedinac ostvaruje, podaci o postojećim kreditima, itd.), lične karakteristike zajmotražioca, kao što su stalnost zaposlenja, rezidencijalni status, stalnost adrese stanovanja ili firme, itd. Od niza varijabli koje se spominju treba odrediti one koji najviše utiču na odluku o odobrenju kredita.

Postoji niz metoda koje se mogu koristiti za razvoj modela za ocenu kreditne sposobnosti. Neke od metoda su statističke, dok se neke od njih oslanjaju na pristupe u kojima se primenjuje veštačka inteligencija. Statističke metode, često korišćene za kredit skoring, su: višestruka regresija (npr. Meyer & Pifer, 1970), diskriminaciona analiza (npr. Altman, 1968; Banasik i ostali, 2003), i logistička regresija (npr. Dimitras i ostali, 1996; Martin, 1977; Elliott & Filinkov, 2008; Lee i ostali, 2002; Desai i

ostali,1996), dok metode veštačke inteligencije uključuju induktivno učenje (npr. Han i ostali, 1996; Shaw & Gentry, 1998), veštačke neuronske mreže (npr. Boritz & Kennedy, 1995; Coakley & Brown, 2000; Jo & Han, 1996; Zhang i ostali, 1999; Lee & Chen, 2005; West, 2000), genetske algoritme (npr. Desai i ostali, 1997; Yobas i ostali, 2000; Huang i ostali, 2006, 2007), i veštačke imune sisteme (npr. Leung i ostali, 2007).

Počev od prvobitnog rada Schanka i Abelsona (1977), zaključivanje na osnovu slučajeva (ZOS) se uspešno primenjuje u mnogim područjima, uključujući i ocenu kreditne sposobnosti.

ZOS je analitički metod rasuđivanja koji rešava probleme povezujući prethodno rešene probleme i iskustvo sa trenutno nerešenim problemom, kako bi se kreirali analitički zaključci bitni za rešavanje problema (Kolodner, 1991).

ZOS pronalazi slične slučajeve uskladištene u bazi slučajeva i adaptira ih tako da odgovaraju trenutnom problemu. Funkcija za dobro uparivanje i pronalaženje najsličnijih slučajeva bi trebalo da uzme u obzir atribute slučajeva i njihove osobine. Slučaj iz baze koji odgovara trenutnom problemu po važnim atributima, ali ne odgovara po onim manje važnim, će sigurno biti bolji za uparivanje u odnosu na slučaj koji odgovara po manje važnim, ali ne i po značajnim osobinama. Iz ovog razloga, integracija domenskog znanja u funkciju uparivanja i pronalaženja slučajeva je izuzetno preporučljiva prilikom modelovanja uspešnog ZOS sistema (Park & Han, 2002).

Uprkos brojnim prednostima, postoje određeni problemi koji se moraju rešiti u cilju projektovanja efektivnog ZOS sistema (Ahn & Kim, 2008). Brojne studije se bave rešavanjem ovih problema, pri čemu su najpopularnije istraživačke teme: odabir odgovarajućih mera sličnosti, izbor atributa i njihovo ponderisanje (npr. Wang & Ishii, 1997; Shin & Han, 1999; Kim & Han, 2001; Chiu i ostali, 2003; Ahn i ostali, 2007; Liao i ostali, 1998).

Postojeći primeri primene ZOS-a uglavnom klasifikuju slučajeve u jednu od dve grupe i zaključuju kojoj od njih trenutni slučaj pripada. Tako da primer, ocenjuje se da li će kompanija bankrotirati (Min i ostali, 2006) ili da li pacijent ima rak (Ahn & Kim, 2009), da li će klijent otplaćivati kredit (Lee, 2007; Vuković i ostali, 2012); da li će neko postati kupac ili ne (Ahn i ostali, 2007).

U praksi, klasifikacija klijenata u više klasa može ponekad bolje da istakne razlike u ponašanju, zbog čega se predlažu pojedine modifikovane ZOS metode najbližeg suseda (Chen i ostali, 2010).

Važan deo svakog ZOS modela je merenje sličnosti. Tradicionalno, za merenje sličnosti u modelima ZOS se koristi Euklidova metrika, za numeričke varijable, odnosno funkcija preklapanja (overlap) za kategoričke atribute. Teorija preferencija se takođe može iskoristiti za merenje sličnosti između slučajeva, naročito što ona pruža više mogućnosti u izražavanju preferencija donosilaca odluka. Li, Sun i Sun (2009) i Li i Sun (2010) su predložili kombinovanje ZOS-a i funkcija koje izražavaju odnose „višeg ranga“ za predviđanje finansijskih nevolja i neuspeha.

U ovom radu, za merenje sličnosti između slučajeva, za numeričke atribute, koristiće se funkcije preferencije iz metode Promethee (Brans & Vincke, 1985; Mareschal, 1986, 1988; Mareschal & Brans, 1988, Brans i ostali, 1986). Cilj je razmotriti da li domensko znanje, koje se može izraziti preko funkcija preferencija, može biti bolje iskorišćeno na takav način da poboljša prediktivne performanse ZOS sistema. Dodatno, hipoteza je i da model sa korišćenjem određenih mera sličnosti za kategoričke atribute, u kombinaciji sa korišćenjem funkcija preferencija za numeričke atribute, može dati bolje rezultate od tradicionalnog modela ZOS-a.

Rad bi obuhvatio i optimizaciju težina atributa, kao i rezultate predikcione moći modela u zavisnosti od broja najbližih suseda koji se uzimaju u obzir.

Modeli ocene kreditne sposobnosti se prevashodno kreiraju da bi olakšali posao bankama. Prednosti primene ovih modela sa aspekta banaka su višestruke, naročito zbog toga što podrazumevaju davanje podrške procesu odlučivanja i dobar alat za planiranje (npr. planiranje budućih kamatnih stopa po kategorijama korisnika), zbog doslednosti i tačnosti, zbog uključivanja svih neophodnih faktora u proces odlučivanja, zbog smanjenja gubitka usled loših kredita, brzine,.... Ne treba zaboraviti i drugu stanu. Modeli donose određene prednosti i klijentima, koje se ogledaju u sledećem: laka procedura aplikacije za kredit, dobijanje odgovora u mnogo kraćem vremenskom intervalu, smanjenje potrebnih informacija za definisanje kreditne sposobnosti, manji troškovi obrade kreditnog zahteva. Često se klijenti koji nemaju dovoljno dugu kreditnu istoriju odbijaju, iako mogu biti dobri korisnici kredita. Upravo u tim slučajevima,

modeli mogu biti od pomoći da se po automatizmu ne odbijaju komitenti koji u potpunosti ne ispunjavaju predefinisane uslove.

Sve navedeno doprinosi boljim odnosima sa klijentima i njihovom zadržavanju.

1.2. Ciljevi istraživanja u doktorskoj disertaciji

Cilj istraživanja u doktorskoj disertaciji je razvoj modela zasnovanog na zaključivanju na osnovu slučajeva i inteligentnom izboru mera sličnosti, koji će pomoći što precizniju klasifikaciju novih slučajeva.

Cilj je i razmotriti da li domensko znanje, koje se može izraziti preko funkcija preferencija, može biti bolje iskorišćeno na takav način da poboljša prediktivne performanse ZOS sistema. Dodatno, cilj je i utvrditi da li model sa korišćenjem određenih mera sličnosti za kategoričke atribute, u kombinaciji sa korišćenjem funkcija preferencija za numeričke atribute, može dati bolje rezultate od tradicionalnog modela ZOS-a.

1.3. Polazne hipoteze

Polazne hipoteze u doktorskoj disertaciji, su sledeće:

Opšta hipoteza:

- Tačnost klasifikacije ZOS metodologije se može poboljšati za rešavanje problema klasifikacije u realnim aplikacijama.

Pojedinačne hipoteze:

- Domensko znanje, koje se može izraziti preko funkcija preferencija, može biti iskorišćeno na takav način da poboljša performanse ZOS modela.
- Optimizacijom parametara odabrane funkcije preferencije, kao mere sličnosti za numeričke atribute, mogu se dobiti bolji rezultati od tradicionalnog modela ZOS-a.

- Inteligentan izbor mera sličnosti za kategoričke atribute može dati bolje rezultate od tradicionalnog modela ZOS-a.

1.4. Metode istraživanja

Osnovne metode istraživanja koje će se koristiti pri rešavanju postavljenog problema su sledeće:

- metoda deskripcije će se koristiti za opisivanje pojava i procesa od interesa, uz objašnjenja važnih obeležja opisivanih pojava i procesa, uočavanje zakonitosti i uzročnih veza i odnosa,
- metoda analize će se upotrebljavati kroz postupak naučnog istraživanja raščlanjivanjem složenih pojmova, sudova i zaključaka na njihove jednostavnije sastavne delove i elemente, odnosno kroz postupak mišljenja od posebnoga ka opštem,
- primena metoda sinteze će se ogledati putem sinteze jednostavnih sudova u složenije i kroz proces uopštavanja, čime će se doći do sistematizovanog znanja, odnosno do izgradnje teorijskog znanja u pravcu od posebnog ka opštem.
- metoda kompilacije će biti primenjena u smislu preuzimanja tuđih rezultata naučnoistraživačkog rada, odnosno tuđih opažanja, stavova, zaključaka i spoznaja, pri čemu će se ova metoda upotrebiti i u kombinaciji s drugim metodama u naučnoistraživačkom radu, a kako bi disertacija u najvećoj meri nosila lični pečat autora, koji će, uz lični pristup pisanju naučnog dela korektno i na uobičajen način citirati sve ono što je od drugih preuzeo,
- komparativna metoda će se koristiti kroz postupak upoređivanja rezultata modela, a radi utvrđivanja njihove sličnosti u ponašanju i razlika među njima,
- metoda modeliranja se sastoji u razvoju modela koji treba da predstavlja stvarnu pojavu, a kojeg eksperimentalno istražujemo sa ciljem da se dobijeni rezultati i unapređenja modela mogu preneti i na realnu pojavu,
- metoda merenja se koristi sa ciljem da se dobiju rezultati predloženih rešenja,

- statistička metoda će se primeti radi utvrđivanja statističke značajnosti dobijenih rezultata.

1.5. Očekivani doprinos rada

Najznačajniji doprinos disertacije je razvoj modela poslovne inteligencije zasnovanog na zaključivanju na osnovu slučajeva i izboru mera sličnosti.

Pored toga, naučni doprinos rada ogleda se i u sledećem:

- Pregled savremenih modela koji mogu da se koriste za ocenu kreditne sposobnosti, kao i postojećih modela zaključivanja na osnovu slučajeva.

- Razvoju modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na zaključivanju na osnovu slučajeva, ali uključuje i domensko znanje, izraženo preko funkcija preferencija.

- Razvoju modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na zaključivanju na osnovu slučajeva, uključuje domensko znanje, izraženo preko funkcija preferencija, ali i odabrane mere sličnosti za kategoričke atribute.

Rad na disertaciji rezultovaće i nizom dodatnih doprinosa, kao što su:

- Podizanje nivoa opšte stručne svesti o mogućnostima predloženih metoda.

- Razvoj klasifikacionog modela, koji je primenjiv u velikom broju konteksta izvan date oblasti ovog istraživanja.

1.6. Plan istraživanja i struktura rada

Plan istraživanja je dat u tabeli 1:

Tabela 1. Dinamika istraživanja

FAZA	ZADACI	METODE, TEHNIKE, STANDARDI
1. Analiza postojećih modela i rešenja iz oblasti	Prikupljanje informacija i analiza metoda	– Pretraživanje stručne literature – Pretraživanje elektronskih baza naučnih radova – Studije slučaja
2. Obezbeđivanje skupova podataka za analizu	Pronalaženje skupova podataka iz oblasti istraživanja	– Pretraživanje Internet resursa – Studije slučaja
3. Definisanje modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na zaključivanju na osnovu slučajeva uz primenu tradicionalnih mera sličnosti	Analiza tradicionalnog modela ZOS-a i razvoj istog za oblast istraživanja	– Stručna literatura – Microsoft Excel 2003/2010 – Palisade Software's Evolver Version 5.5.

FAZA	ZADACI	METODE, TEHNIKE, STANDARDI
4. Projektovanje modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na ZOS-u, ali uključuje i domensko znanje, izraženo preko funkcija preferencija	Analiza i razvoj modela za oblast istraživanja	– Stručna literatura – Microsoft Excel 2003/2010 – Palisade Software's Evolver Version 5.5.
5. Projektovanje modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na ZOS-u, uključuje domensko znanje, izraženo preko funkcija preferencija, ali i odabrane mere sličnosti za kategoričke atribute	Analiza i razvoj modela za oblast istraživanja	– Stručna literatura – Microsoft Excel 2003/2010 – Palisade Software's Evolver Version 5.5.
6. Testiranje i evaluacija rezultata	Testiranje i ocena rezultata metodom komparacije i analiza rezultata	– SPSS for Windows

2. Osnovni koncepti

Onoliko koliko možeš da sagledaš prošlost, toliko možeš da spoznaš budućnost!
Winston Churchill

Budućnost pripada onima koji se najdalje i najduže sećaju prošlosti.
Fridrih F. Nietzsche

Mudro je učiti se na svojim greškama, ali još mudrije na tuđim.
Hillel Segal

Predmet istraživanja u ovom radu je model za klasifikaciju, dok je cilj istraživanja razvoj modela zasnovanog na zaključivanju na osnovu slučajeva i inteligentnom izboru mera sličnosti, koji će pomoći što precizniju klasifikaciju novih slučajeva. U ovoj studiji će se razmotriti korisnost hibridnog ZOS modela, u kome će se kombinovati funkcije preferencije i odabrane funkcije za ocenu sličnosti kategoričkih atributa, uz korišćenje genetskog algoritma (GA). Prva dva dela ovog odeljka predstavljaju pregled osnovnih koncepata ZOS-a. Posebno je istaknuta faza pronalaženja slučajeva, odnosno metoda najbližeg suseda, a s obzirom na značaj iste za ovo istraživanje. Treći deo opisuje osnovne vrste funkcija preferencija. Četvrti deo ukazuje na vrste podataka i njihove osobine, i daje moguće mere sličnosti kvantitativnih i kategoričkih podataka. Peti deo ovog poglavlja sadrži osnovne koncepte GA, kao i opis i podešavanje GA koji je korišćen u ovom radu.

2.1. Zaključivanje na osnovu slučajeva

Zaključivanje na osnovu slučajeva (ZOS) je metodologija za rešavanje problema i odlučivanje u složenom i promenljivom poslovnom okruženju. To je metodologija koja rešava nove probleme koristeći rešenja starih, tačnije nove probleme rešava povezujući neke ranije rešene probleme i iskustva sa novim problemom, formirajući tako analogijske zaključke bitne za rešavanje problema (Kolodner & Mark, 1992). Suočavajući se sa novim problemom, ZOS pronalazi slične slučajeve uskladištene u

bazi slučajeva i prilagođava ih novom problemu. Ključni faktori koji utiču na performanse mehanizma preuzimanja ZOS-a su predstavljanje slučaja, indeksiranje slučaja i pokazatelj sličnosti (Buta, 1994).

2.1.1. Zaključivanje na osnovu slučajeva kao metoda klasifikacije

ZOS-om se rešavaju problemi klasifikacije, odnosno ZOS podrazumeva proces odlučivanja koji ima za cilj da posmatranom objektu ili situaciji dodeli ispravnu alokaciju, odnosno jednu od predefinisanih kategorija ili klasa.

Sam postupak klasifikacije se može opisati na sledeći način:

Postoje ulazni podaci koji predstavljaju skup za obučavanje. Ovaj skup sadrži veći broj zapisa (slučajeva), od kojih svaki sadrži:
- poznato obeležje klase kojoj pripada,
- više promenljivih (osobine, prediktori), čije vrednosti bi, po svoj prilici, trebalo da sadrže dovoljno informacija da se napravi razlika između klasa (Yang L., 2002).

Skup za obučavanje se koristi da bi se izgradio model dodeljivanja klasne promenljive na osnovu drugih varijabli. Model se zatim koristi za predviđanje klasa budućih slučajeva (ili verovatnoća da oni pripadaju određenoj klasi). Problem klasifikacije se bavi izgradnjom modela (klasifikatora) koji će biti primenjen na nizu slučajeva, gde svakom novom slučaju treba dodeliti jednu od unapred definisanih klasa na osnovu posmatranih varijabli ili osobina slučaja (Yang L., 2002).

Bez obzira na sve težnje da se dobije što precizniji model klasifikacije, ipak postoje jaki praktični razlozi da se očekuje da apsolutno tačno klasifikovanje ne postoji. Postoje tri osnovna problema u praksi koja sprečavaju savršeno predviđanje klasifikacije (Yang L., 2002):

- Slučajevi: S jedne strane, slučajevi sa poznatom klasom su ograničeni. Klasifikacija se zasniva na algoritmima učenja na prethodnim slučajevima, uzorcima sa poznatom klasom/ ishodom, i ovi podaci se zovu podacima za učenje.
Kada bismo znali sve moguće slučajeve i klase koje im odgovaraju, svi podaci bi se mogli smestiti u tabelu, i za neki novi slučaj bismo jednostavno mogli da pogledamo u tabelu i tražeći odgovarajuću klasu na osnovu prethodnog iskustva. Međutim, ovo je,

nažalost, gotovo nemoguće u realnosti, uzorci sa poznatom klasom su često sasvim ograničeni u praksi. S druge strane, čest je slučaj da prikupljeni uzorci nisu reprezentativni za populaciju. Performanse naučenog klasifikatora sa nereprezentativnim uzorcima ne mogu biti dobre (Yang L., 2002).

- Karakteristike/ osobine slučaja: Prediktivna sposobnost osobina slučaja je od fundamentalnog značaja za uspeh bilo kog sistema učenja. U realnom svetu odlučivanja, granice klasa se često preklapaju. Iz analitičke perspektive, to znači da je sasvim moguće da slični ili čak identični slučajevi spadaju u različite klase, odnosno, može biti nedoumica u uzorcima. Ako su mnogi uzorci dvosmisleni za dati skup osobina, to može da sugeriše da te osobine imaju lošu prediktivnu moć, i da dobro rešenje za problem klasifikacije nije moguće ako se zadrže samo te osobine. Nekad se može desiti da neka važna karakteristika nije bila dimenzija razmatrana u slučajevima, pa samim tim nije ni uključena u model odlučivanja (Yang L., 2002).

- Definicija klasa: neizvesna definicija pripadnosti klasi slučajeva iz uzorka doprinosi još većoj složenosti algoritma klasifikacije. Osnovni zahtev metoda za klasifikaciju je da su podaci predstavljeni u obliku slučajeva sastavljenih od odgovarajućih osobina sa ispravnom klasifikacijom. U mnogim primenama, ispravna klasifikacija nije apsolutno poznata. U modelima kreditnog bodovanja, na primer, tačna klasa rizika može biti poznata tek nakon određenog vremenskog perioda, koji se naziva „period ishoda“ (Yang L., 2002).

Ukratko, klasifikacija ne može dati savršena predviđanja zbog gore navedenih praktičnih problema. Tačnije, ono što se može direktno naučiti iz prošlih slučajeva je ograničeno, naročito ako se ignoriše kontekst u kojem je rešavanje problema sprovedeno. Potencijal za izgradnju uspešnog modela klasifikacije zavisi ne samo od tehnika klasifikacije, već i od podataka koji su izabrani za analizu, gde, takođe, stručnjaci iz domena, mogu da igraju važnu ulogu (Yang L., 2002).

U algoritmima učenja koji su zasnovani na slučajevima, pamte se slučajevi za učenje i kada treba da se donese odluka za novi slučaj, pretražuju se sačuvani primeri kako bi se

pronašao onaj koji najviše podseća na novi slučaj. „Učenje“ se dešava u trenutku primene, zbog čega se ovaj algoritam naziva i lenjo učenje ili učenje zasnovano na pamćenju (Yang L., 2002).

ZOS metod ima nekoliko atraktivnih svojstava koje ga čine pogodnim za problem kreditnog bodovanja. Neparametarska priroda metoda omogućava modeliranje nelinearnosti funkcije rizika. Takođe je prilično intuitivan postupak i kao takav može se lako objasniti poslovnim menadžerima od čije konačne odluke će i zavisiti primena metode. Drugim rečima, ZOS je konceptualno jednostavan i lak za implementaciju. Može se koristiti dinamično i postepeno, uz automatsko ažuriranje modela, kako slučajevi evoluiraju (Yang L., 2002).

2.1.2. Induktivne i deduktivne metode i ZOS kao induktivna metoda klasifikacije

Kao što je u prethodnom poglavlju napomenuto, ZOS metodologija će se u ovom radu korisititi kao metoda klasifikacije. Klasifikacija je proces kojim se objekat svrstava u neku od predefinisanih kategorija, a u zavisnosti od svojstava koja ga karakterišu. Da bi se donela odluka o klasifikaciji, ljudi se uglavnom uvek oslanjaju na prošlo iskustvo. Prethodna iskustva se mogu dobiti na dva načina – indukcijom (zaključivanje iz pojedinačnog o opštem) ili dedukcijom (zaključivanje iz opšteg o posebnom), što vodi do dva pristupa računarskog odlučivanja za rešavanje problema klasifikacije (Yang L., 2002).

Deduktivno odlučivanje je moguće preko implementacije baze znanja, odnosno takozvanog ekspertnog sistema. Odlučivanje je ovde automatizovano preko kompjuterskog sistema zasnovanog na znanju, do koga se došlo intervjuišući relevantne eksperte (Yang L., 2002).

Induktivno odlučivanje podrazumeva modele klasifikacije koji su izgrađeni induktivno, polazeći od brojnih pojedinačnih prethodno sačuvanih primera i idući ka opštem, na primer otkrivajući i analizirajući paterne pronađene u prethodno rešenim slučajevima.

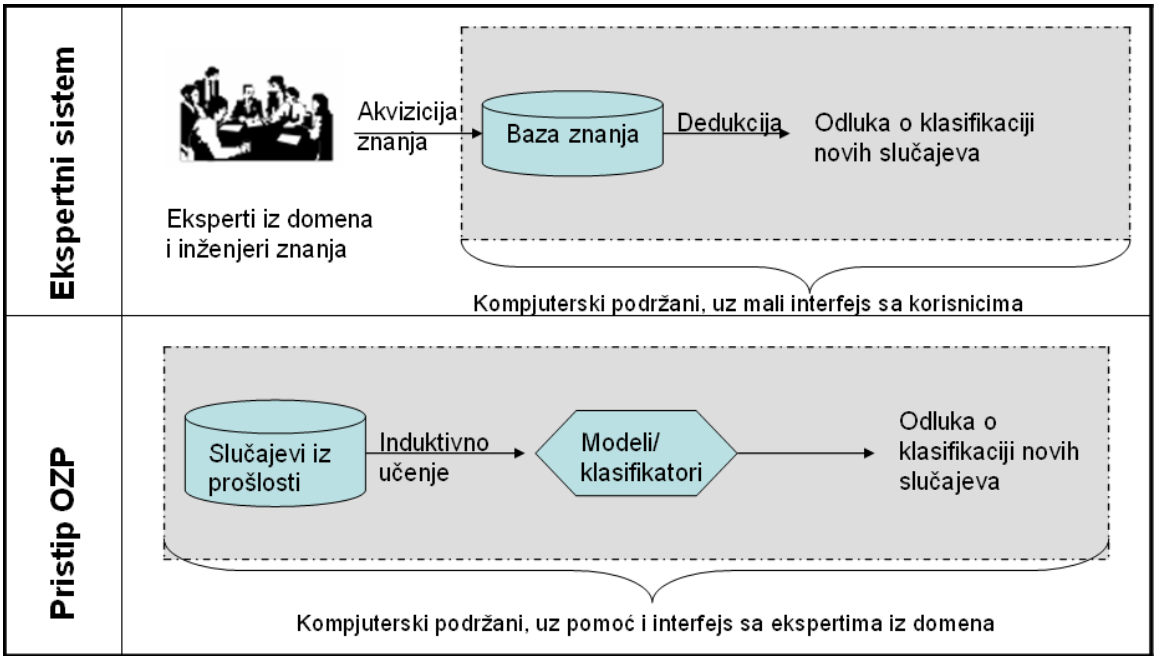
Ovi modeli mogu potom biti korišćeni za odlučivanje kojoj klasi treba da pripadne novi nepoznati slučaj. Jedan od glavnih zadataka otkrivanja zakonitosti u podacima (OZP, eng. data mining) je upravo izgradnja modela klasifikacije na induktivan način (Yang L., 2002).

Osnovne razlike između ova dva pristupa su prikazane na slici 1. (Yang L., 2002). Znanje u ekspertnom sistemu potiče od takozvanih „inženjera znanja“ i sačuvano je u sistemu, tako da se odluka za nove slučajeve pravi deduktivno u skladu sa znanjem koje je sačuvano u sistemu, a koje je obično u formi niza „ako-onda“ pravila. Donošenje odluka bi trebalo da bude automatsko sa veoma malim interakcijama sa korisnikom u posebnim slučajevima. Nasuprot tome, OZP pristup otkriva znanje induktivno iz podataka, forma znanja su modeli ili klasifikatori, koji mogu biti u različitim oblicima, kao što su model ZOS-a, stabla odlučivanja ili obučene neuronske mreže, u zavisnosti od toga koja tehnika klasifikacije se koristi. Ceo proces OZP-a je kompjuterski podržan, ali je najčešće daleko od potpuno automatskog otkrivanja znanja, odnosno, interfejs i pomoć stručnjaka iz domena su tokom procesa OZP-a često potrebni u značajnoj meri (Yang L., 2002).

Oba pristupa klasifikacije imaju svoja ograničenja. Ekspertni sistemi su često kritikovani zbog ograničenih sposobnosti da prevaziđu nivo postojećih stručnjaka.
Drugi, često navođeni problem, je veliki napor potreban da se izgradi i održava baza znanja, kao i nedostatak obučenih inženjera znanja koji bi intervjuisali eksperte i sakupili njihovo znanje u skupu pravila odlučivanja ili drugih reprezentativnih elemenata. Ovaj proces, poznat kao sticanje znanja, veoma je dugotrajan, što vodi do dugog razvoja, koji mora biti kontinualan ukoliko sistem treba da se održi u rutinskoj upotrebi sa visokim nivoom učinka (Yang L., 2002).

Induktivna klasifikacija može biti ograničena tačnošću predviđanja zbog nekih nedostataka u podacima za učenje. Pored toga, često su neophodne podrške stručnjaka kada je u pitanju odabir uzoraka, kao i tumačenje rezultata (Yang L., 2002). Automatski otkrivena pravila nisu uvek razumna i stoga bi nekad trebalo da budu revidirana od strane stručnjaka iz domena. Argument u korist induktivnog učenja iz baze uzoraka

rešenih slučaja je da bi se tako mogle prevazići performanse stručnjaka jer postoji potencijal da se otkriju novi odnosi ispitivanjem slogova uspešno rešenih slučajeva. Pored toga, proces učenja automatski podržava uključivanje znanja u sistem bez potrebe za znanjem inženjera (Yang L., 2002).



Slika 1. Deduktivno i induktivno odlučivanje o klasifikaciji

Obe opcije imaju sebi svojstvene karakteristike i mogu biti odgovarajuće za različite situacije. Za probleme odlučivanja kod kojih su stručnjaci iz domena potvrdili iskustva koja su identifikovana i izražena preko formalnih pravila, i kod kojih su pravila odlučivanja stabilna tokom vremena, ekspertni sistem, odnosno sistem zasnovan na znanju može biti implementiran i održavan efikasno (Yang L., 2002). Nasuprot tome, induktivni pristup se može odnositi na probleme klasifikacije kod kojih je dostupan veliki broj ranijih slučajeva sa poznatim ishodima – klasama. Pored toga, primena ovog pristupa je pogodna ako su iskustva stručnjaka nepotpuna ili nepotvrđena, ili ta iskustva nije lako izraziti formalnim jezikom (Yang L., 2002).

Oba navedena pristupa klasifikacije se primenjuju na problem kreditne procene. Neki ekspertni sistemi su napravljeni da podrže ispitivanje boniteta i kreditnog rejtinga, dok je primena induktivnih metoda obično prisutna kod rešavanja problema kreditnog

bodovanja (eng. credit scoring), koji je popularan u oblasti potrošačkih kredita i kredita malim preduzećima (Yang L., 2002).

Induktivna klasifikacija je u velikoj meri proučavana od strane statističara, kao i grupa koje se bave bazama podataka i veštačkom inteligencijom. U statistici problem klasifikacije se ponekad naziva problemom predviđanja, dok se u oblasti mašinskog učenja često naziva konceptom nadgledanog učenja, jer se parametri modela podešavaju prema poznatim vrednostima izlaza, odnosno proces učenja je vođen obezbeđenim primerima. Ovi algoritme treba razlikovati od nenadgledanog učenja ili klasterovanja, gde se do klasa dolazi iz podataka.
Izgradnja modela klasifikacije iz skupa podataka za koje se zna pripadajuća klasa, se imenuje različitim terminima kao što su: otkrivanje paterna ili diskriminacija, dok drugi autori, posebno oni koji se bave mašinskim učenjem, nazivaju ove tehnike induktivno učenje, empirijsko učenje, ili zaključivanje na osnovu slučajeva (Yang L., 2002).

2.1.3. Osnovne vrste ZOS metoda

ZOS je naziv dat metodi zaključivanja koja koristi specifična iskustva iz prošlosti, pre nego opšte znanje. ZOS podrazumeva rešavanje problema po analogiji, gde se novi problem rešava prepoznavanjem njegove sličnosti sa određenim već poznatim problemom, prenoseći rešenje poznatog problema na novi, do tad nerešeni, problem.

ZOS paradigma obuhvata niz različitih metoda za organizovanje, pronalaženje, korišćenje i indeksiranja zapamćenog znanja iz prošlih slučajeva (Aamodt & Plaza, 1994). Slučajevi mogu da se čuvaju kao konkretna iskustva, ili kao skup sličnih slučajeva koji može da formira jedan uopšten slučaj. Slučajevi se mogu čuvati kao posebne iskustvene celine ili mogu biti podeljene na podceline i raspoređene unutar strukture znanja/iskustva. Slučajevi mogu biti indeksirani pomoću reči u kojima se koriste prefiksi ili slobodnim rečima, i to u okviru ravne ili hijerarhijske indeksne strukture. Rešenje iz prethodnog slučaja može biti direktno primenjeno na trenutni problem, ili može biti modifikovano u skladu sa razlikama koje postoje između dva

slučaja. Poklapanje slučajeva, prilagođavanje rešenja, i učenje iz iskustva mogu biti vođeni i podržani snažnim modelom uopštenog domenskog znanja, ili više površnim i sakupljenim iskustvom/znanjem ili se mogu zasnivati samo na oćiglednoj sintaksičkoj sličnosti. ZOS metode mogu biti sasvim samostalne i automatske, ili mogu u velikoj meri podrazumevati međusobnu interakciju sa korisnikom po pitanju podrške i smernica za određene izbore. Neke ZOS metode pretpostavljaju prilično veliki broj široko rasprostranjenih slučajeva u svojoj bazi slučajeva, dok se druge zasnivaju na više ograničenoj grupi tipičnih slučajeva. Prethodni slučajevi mogu se pronalaziti ili procenjivati redom ili paralelno (Aamodt & Plaza, 1994).

U suštini, „zaključivanje na osnovu slučajeva" je samo jedan u nizu pojmova koji se koriste da označe sisteme ove vrste. Ovo je dovelo do izvesne konfuzije, pogotovo što se termin zaključivanje na osnovu slučajeva koristi i kao opšti termin za nekoliko vrsta specifičnijih pristupa, kao i za samo jedan takav pristup. U izvesnoj meri, ovo sve se može reći i za analogno zaključivanje. Pokušaj razjašnjavanja termina koji se odnose na zaključivanje na osnovu slučajeva dat je u nastavku teksta (Aamodt & Plaza, 1994).

Zaključivanje na osnovu primera (Exemplar-based reasoning) – podrazumeva definisanje koncepta na osnovu primera. ZOS metode koje se bave konceptom učenja se ponekad nazivaju i metodama zasnovanim na primerima. Primeri se mogu naći u ranim radovima pojedinih autora (više u Aamodt & Plaza, 1994). U okviru ovog pristupa, rešavanje problema je zadatak klasifikacije, tj. određivanje prave klase za neklasifikovane primere. Klasa najsličnijeg prethodnog slučaja postaje rešenje problema klasifikacije. Skup klasa čini grupu mogućih rešenja. Modifikacija otkrivenog rešenja je izvan okvira ove metode.

Zaključivanje na bazi instanci (Instance based reasoning). Ovo je specijalizovana vrsta zaključivanja na osnovu primera. Opis problema iz domena često može rezultirati standardizovanim opisom slučaja, koji se može predstaviti vektorom obeležja koji sadrži numeričke ili simboličke vrednosti sa jednostavnom unutrašnjom strukturom. Rešeni zadaci veoma često imaju vrlo jednostavnu definiciju prostora konačnog rešenja.

Zaključivanje na bazi instance je specijalizacija zaključivanja na osnovu primera, koje definiše svoje koncepte prošireno kao skup svih primeraka.
Zaključivanje na bazi instanci je sintaksička specijalizacija, koja se zasniva na jednostavnom predstavljanju slučajeva. Osim toga, ovo zaključivanje ima za cilj da proučava automatizovano učenje bez korisnika u petlji. Za pristupe zaključivanja na osnovu primera se smatra da zahtevaju više znanja u odnosu na pristupe zaključivanja na bazi instanci. Nedostatak smernica iz domena opšteg znanja kompenzuje korišćenjem velikog broja primera. To je negeneralizujući pristup problemu konceptualnog učenja, koji se rešava klasičnim, induktivnim metodama mašinskog učenja..

Zaključivanje na bazi sećanja/ memorije (Memory based reasoning). Ovaj pristup naglašava kolekciju slučajeva kao veliku memoriju, i zaključivanje kao proces pristupanja memoriji i pretraživanja memorije. Organizacija memorije i pristup istoj su u fokusu metoda zasnovanih na slučajevima. Korišćenje paralelnih tehnika obrade je karakteristika ovih metoda, i ono što ovaj pristup razlikuje od drugih. Metode pristupa i skladištenja mogu da se oslone na čisto sintaksičke kriterijume, ili mogu pokušati da iskoriste opšte domensko znanje (Aamodt & Plaza, 1994).

Zaključivanje na osnovu slučajeva (Case based reasoning). Iako se zaključivanje na osnovu slučajeva u ovom radu koristi kao generički pojam, tipične metode zaključivanja na osnovu slučajeva imaju neke karakteristike koje ih razlikuju od drugih pristupa navedenih u ovom poglavlju. Prvo, pretpostavlja se da tipičan slučaj obično ima određeni stepen bogatstva informacija sadržanih u njemu, kao i određenu složenost. Na taj način, vektor osobina koji sadrži određene vrednosti i odgovarajuću klasu nije ono što bismo nazvali tipičnim opisom slučaja. Ono što bismo nazvali tipičnim metodama zasnovanim na slučajevima takođe imaju još jedno karakteristično svojstvo: one mogu da modifikuju, ili prilagođavaju preuzeto rešenje kada se primenjuje u rešavanju problema u drugačijem kontekstu. Tipične metode zasnovane na slučajevima takođe koriste opšte pozadinsko znanje - iako njegovo bogatstvo, stepen eksplicitnog predstavljanja i uloga u ZOS procesu variraju. Osnovne metode tipičnih ZOS sistema pozajmljuju dosta od teorije kognitivne psihologije (Aamodt & Plaza, 1994).

U odnosu na Zaključivanje na bazi instanci, Zaključivanje na bazi slučajeva se najviše razlikuje po sledećem:
- koristi se i za ostale zadatke, ne samo za probleme klasifikacije,
- uopšteno, slučaj ima kompleksnu strukturu, ne predstavlja se samo vektorom karakteristika,
- pronađeni slučaj se uopšteno modifikuje, kada se primenjuje za rešavanje novog problema,
- koristi opšte domensko znanje.

U odnosu na Zaključivanje na bazi analogije, Zaključivanje na bazi slučajeva se najviše razlikuje po tome što su svi slučajevi iz istog domena, pa se stoga može smatrati i specijalnom vrstom analitičkog zaključivanja.

Zaključivanje na bazi analogije (Analogy based reasoning). Ovaj termin se ponekad koristi kao sinonim za zaključivanje na osnovu slučajeva, da bi se opisao tipičan pristup zasnovan na slučajevima. Međutim, ovaj termin se takođe često koristi i da okarakteriše metode koje rešavaju nove probleme na osnovu slučajeva iz prošlosti iz drugog domena, dok se tipična metoda zasnovana na slučajevima fokusira na indeksiranje i strategije uparivanja za slučajeve pojedinačnih domena. Istraživanje zaključivanja na bazi analogije je stoga podoblast koja se bavi mehanizmima za identifikaciju i korišćenje analogija kroz različite domene. Glavni fokus je na ponovnoj upotrebi prethodnog slučaja, što se zove mapiranje problema: pronalaženje načina za prenos, ili mapiranje, rešenja identifikovane analogije na novi, ciljni problem (Aamodt & Plaza, 1994).

Termin zaključivanje na osnovu slučajeva se obično koristi u opštem smislu (Aamodt & Plaza, 1994).

2.1.4. Osnovni pojmovi i aktivnosti kod Zaključivanja na osnovu slučajeva

Kao što se iz samog naziva naslućuje u osnovi ZOS-a su slučajevi. Slučaj je uređeni par problem-rešenje u određenom kontekstu. Slučaj se pamti jer se pretpostavlja da će to iskustvo moći da se iskoristi u trenutku kada se ponovo javi sličan problem.
Svaki slučaj se sastoji iz dva dela i to problema, u kome se krije kontekst, i rešenja za problem u datom kontekstu. Slučaj se sastoji od niza kriterijuma u čijim vrednostima su smeštene informacije o slučaju. U vrednostima kriterijuma mogu biti i smešteni i tekstualni zapisi, fotografije, video zapisi, itd. Može se reći da se slučaj sastoji iz tzv. indeksiranih i neindeksiranih kriterijuma. Indeksirani kriterijumi služe za pretraživanje baze znanja. Neindeksirani kriterijumi služe za opisivanje slučaja i često se na osnovu njih donosi konačna odluka.
U nekim sistemima ZOS metodologija se može koristiti i samo za pretraživanje baze i izbor slučajeva, a kako bi se našli slučajevi koji su najsličniji određenom željenom slučaju. U takvim situacijama, slučajevi ne moraju imati izlazne atribute.

Kada se suočimo sa novim problemom koji treba rešiti, ZOS metodologijom će se pretražiti baza prethodnih slučajeva, naći će se slučaj iz baze koji je najsličniji novom slučaju. Moguće je izvojiti i više najsličnijih slučajeva iz prošlosti, a ako je potrebno, moguće je i prilagoditi staro rešenje, ili više njih, novom problemu i sačuvati novo (modifikovano staro) rešenje u bazi slučajeva. Novo rešenje se generiše pronalaženjem i eventualnim prilagođavanjem starog slučaja koji približno odgovara trenutno datoj situaciji – problemu. S vremena na vreme, dobro je proveriti da li je baza slučajeva postala redudantna, tj. da li ima slučajeva koji su jednaki ili „dovoljno slični“ da je moguće izvršiti redukciju baze slučajeva i time podići efikasnost rada (Suknović & Delibašić, 2010).

Shodno navedenom, može se reći da ZOS uključuje: prihvatanje opisa novog problema, pronalaženje relevantnih slučajeva iz baze slučajeva, prilagođavanje pronađenih slučajeva da odgovaraju trenutnom problemu i generisanje rešenja za postojeći problem, i ocenjivanje rešenja.

Oko osnovnih aktivnosti ZOS metodologije se slaže većina autora, pa tako na primer, Lopez de Mantaras i ostali (2005) ističu da rešavanje problema primenom ZOS-a uključuje dobijanje opisa problema, merenje sličnosti aktuelnog problema sa prethodnim problemima čija su rešenja poznata, a koji su uskladišteni u bazi slučajeva (ili memoriji), zatim preuzimanje jednog ili više sličnih slučajeva i pokušavanje da se ponovo iskoriste njihova rešenja ili iskustva, nekad i uz moguće prilagođavanje na račun razlike u opisima problema. Rešenje predloženo od strane sistema se zatim ocenjuje (npr. tako što primenjuje na početnom problemu ili procenjuje od strane stručnjaka iz date oblasti). Zatim, ako je predloženo rešenje adekvatno opisu problema, rešenje treba da bude zadržano kao novi slučaj, a sistem je naučio da reši novi problem (Lopez de Mantaras i ostali, 2005).

Prema Kolodner (1993), ZOS obuhvata četiri značajna koraka: (1) predstavljanje slučaja, (2) indeksiranje slučaja, (3) pronalaženje slučaja, i (4) adaptacija slučaja. Predstavljanje slučaja podrazumeva karakteristike vezane za prošli slučaj; indeksiranje slučaja ima za cilj da olakša pretragu i pronalaženje sličnih slučajeva; pronalaženje slučaja preuzima slučajeve iz baze najsličnije posmatranom slučaju, dok je adaptacija slučaja proces menjanja postojećeg slučaja ili izgradnja novog ako nijedan pronađeni slučaj nije u skladu sa trenutno nerešenim (Chen i ostali, 2010).

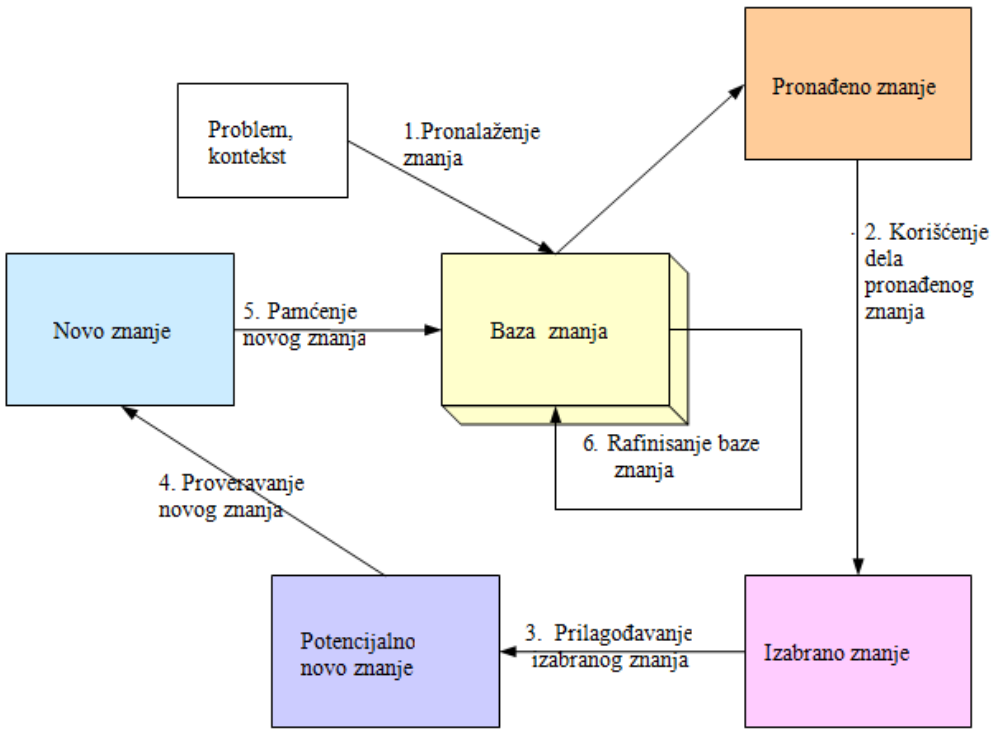
Aamodt & Plaza (1994) su predložili model ciklusa rešavanja problema u ZOS-u, koji uključuje četiri zadatka, poznata kao 4RE[1] (tj. retrieve - pronaći najsličniji slučaj ili slučajeve, reuse - ponovo iskoristiti informacije i znanje iz tih slučajeva da bi se rešio trenutni slučaj, revise – pregledati predloženo rešenje i retain - zadržati delove iskustva koji će verovatno biti korisni za buduće rešavanje problema). Reinartz i ostali (2001) su preradili ovaj model, proširujući ga uključivanjem dva nova koraka, a to su: review - korak čiji je cilj praćenje kvaliteta sistema znanja i restore - korak koji podrazumeva biranje i primenu operacija održavanja.
Taksativno, njihov model uključuje sledećih šest aktivnosti, takozvanih šest RE (Reinartz i ostali, 2001):

[1] retrieve, reuse, revise, retain

1. Pronalaženje znanja (slučaja) koji odgovara, koji je sličan trenutnom zahtevu /problemu (eng. *retrieve*);
2. Korišćenje dela pronađenog znanja, tj. onih slučajeva koji su dovoljno slični novom problemu (eng. *reuse*);
3. Prilagođavanje, ukoliko ima potrebe, dela pronađenog znanja novom zahtevu (eng. *revise*);
4. Proveravanje da li je novo rešenje vredno pamćenja, odnosno merenje korisnosti novorešenog slučaja (eng. *review*);
5. Pamćenje novog znanja, ukoliko je tako odlučeno pod 4 (eng. *retain*);
6. Periodično prečišćavanje baze slučajeva kako bi se ista tokom vremena unapredila, da ne bi bilo redudantnih slučajeva i da bi zauzimala manje memorije u cilju efikasnije pretrage prilikom rešavanja novih problema (eng. *refine*).

Ove aktivnosti su prikazane i na slici 2. u nastavku.

Može se reći da prve tri aktivnosti najviše zanimaju donosioca odluke, dok se preostale pre svega tiču analitičara, odnosno osobe koja je zadužena za održavanje ZOS sistema.



Slika 2. Životni ciklus ZOS

Metodologija ZOS-a je zasnovana na određenim pretpostavkama (Watson, 2003):
- svet funkcioniše po principu reda, a ne haosa,
- situacije (problemi) se ponavljaju, pa iz vredi pamtiti,
- slični problemi imaju slična rešenja.

ZOS je generalno veoma jednostavan za primenu i često može da barata i sa kompleksnim i nestrukturiranim odlukama veoma efektivno (Ahn i ostali, 2007).

Jedna od ključnih tema u procesu ZOS-a je pronalaženje sličnih slučajeva u bazi slučajeva, tj. merenje sličnosti slučajeva da bi se našao najbolji odgovarajući slučaj. Dakle, mera uspešnosti ZOS sistema u ogromnoj meri zavisi od njegove sposobnosti da pronađe najrelevantnije slučajeve iz prošlosti u cilju podrške rešavanju novog slučaja. Cilj pronalaženja je da se preuzmu iskustva najkorisnijih prethodnih slučajeva u odnosu na novi slučaj i da se ignorišu oni prethodni slučajevi koji su irelevantni (Montazemi & Gupta, 1997).

Funkcija za dobro uparivanje i pronalaženje najsličnijih slučajeva bi trebalo da uzme u obzir atribute slučajeva i njihove osobine. Slučaj iz baze koji odgovara trenutnom problemu po važnim atributima, ali ne odgovara po onim manje važnim, će sigurno biti bolji za uparivanje u odnosu na slučaj koji odgovara po manje važnim, ali ne i po značajnim osobinama. Iz ovog razloga, integracija domenskog znanja u funkciju uparivanja i pronalaženja slučajeva je izuzetno preporučljiva prilikom modelovanja uspešnog ZOS sistema (Park & Han, 2002).

Jedan od najvažnijih koraka u ciklusu ZOS-a, koji će biti i u fokusu ovog rada, je pronalaženje prethodnih slučajeva koji mogu da se iskoriste za rešavanje trenutno posmatranog problema. Osnovna pretpostavka na kojoj se bazira pronalaženje znanja na bazi sličnosti jeste da su upravo najsličniji slučajevi najkorisniji za rešavanje ciljnog problema.

U cilju sticanja opšte slike o fazama odlučivanja i potrebnim aktivnostima u ciklusu ZOS-a, u poglavlju koje sledi iste će biti detaljnije opisane.

2.1.5. Faze odlučivanja u okviru ZOS metodologije

Može se reći da ZOS čine četiri faze odlučivanja (Delibašić, 2007):
1. Strukturiranje slučajeva,
2. Normalizacija indeksa,
3. Agregacija slučajeva i
4. Analiza sistema (evaluacija).

2.1.5.1. Strukturiranje slučajeva

Slučajevi se pamte u bazi slučajeva, tj. znanja. Da bi slučajevi mogli da se sačuvaju i koriste u procesu ZOS-a, potrebno je da imaju jasno definisanu, mašinski čitljivu, strukturu (Suknović & Delibašić, 2010). Definisanje tabela, skupa atributa i tipova atributa koji će se čuvati u bazi podataka, predstavlja fazu strukturiranja.

Najčešće se za opisivanje slučajeva može koristiti mnoštvo atributa, ali se zbog efikasnosti ZOS sistema teži izdvajanju samo određenog podskupa atributa i to onih atributa koji nose dovoljnu količinu informacija da najbolje opišu slučaj tako da pretraživanje baze da najbolje moguće rezultate. Skup atributa za pretraživanje se kod ZOS sistema naziva skup indeksa (Suknović & Delibašić, 2010).

Izbor skupa atributa koji mogu da opišu problem može biti jedan od bitnih momenata u procesu odlučivanja. To je postupak u kome se određuju težine, odnosno značaj atributa (ako težina teži nuli, atribut verovatno nema uticaja na rešenje) i odnosi među njima.

Indeksiranje, tj. izbor indeksa, može da se radi ručno, ali i automatski. Ručna metoda traži ekspertsko znanje, da se razume slučaj i da se odrede uslovi, izraženi preko kriterijuma, pod kojim će se slučaj koristiti. Sa druge strane, automatska metoda koristi neke od tehnika i metoda otkrivanja zakonitosti u podacima.

Organizacija baze slučajeva zavisi od vrste podataka i namene sistema (Suknović & Delibašić, 2010). U ovom radu korišćene su baze slučajeva kod kojih se slučajevi čuvaju u formi redova u tabeli.

2.1.5.2. Normalizacija slučajeva

Nakon faze strukturiranja problema, sledi faza normalizacije.

Normalizacija podataka predstavlja svođenje podataka na isti raspon vrednosti, odnosno dovođenje podataka različitih dimenzija u takav oblik da mogu međusobno da se upoređuju. Normalizacija indeksa u bazi slučajeva je neophodna jer se njome sve vrednosti dovode u isti rang, najčešće u rasponu [0,1] što dalje omogućava međusobnu uporedivost slučajeva i izračunavanje sličnosti slučajeva iz baze sa novim slučajem, primenom odgovarajućih mera sličnosti.
Od načina normalizacije podataka zavisi i način ocenjivanja slučajeva i njihovo poređenje, pa se ova faza smatra izuzetno bitnom u celokupnom procesu ZOS.

Indeksi mogu biti predstavljeni kako numeričkim tako i kategoričkim podacima, pa se stoga razlikuju metrike za svaki od ovih tipova podataka.

Za numeričke podatke se za normalizaciju najčešće koriste sledeće metrike:
- L1 metrika (Menhetn ili metrika gradskog bloka),
- L2 (vektorska ili Euklidska),
- L∞ (Čebiševljeva) itd.

Da bi se kategorički podaci pripremili za pretraživanje, analitičar treba da vodi računa o (Boriah i ostali, 2008):
- broju kategoričkih atributa,
- broju kategorija koje postoje u atributu,
- distribuciji kategorija u atributu.

U zavisnosti od navedenog, potrebno je odrediti adekvatnu meru sličnosti. S obzirom da ne postoji mera sličnosti koja se može koristiti za sve podatke i sve sisteme, u ovom radu je primenjeno nekoliko mera sličnosti za kategoričke atribute predloženih u radu (Boriah i ostali, 2008).

Vrsta normalizacije ima neograničeno, i nije jasno *zašto* je bolje koristiti nekad jednu, a ne drugu metriku, već se iskustvenom proverom dolazi do saznanja o upotrebljivosti metrika.

2.1.5.3. Agregacija slučajeva

Pošto se izvrši strukturiranje sistema i normalizacija, prelazi se na ocenjivanje kvaliteta slučajeva u bazi, odnosno na fazu agregacije.

ZOS metodologija koristi koncept sličnosti (udaljenosti) kojim se računa sličnost slučaja za koji se traži rešenje i slučajeva koji postoje u bazi slučajeva (Suknović & Delibašić, 2010). Svakom slučaju se tokom procesa pretraživanja baze slučajeva dodeljuje mera koja govori koliko je slučaj iz baze blizak slučaju za koji se traži rešenje.

Najpoznatije metode koje se koriste za pronalaženje sličnih slučajeva iz prošlosti su: metoda najbližeg suseda, indukcija, indukcija vođena znanjem i otkrivanje paterna (Buta, 1994; Delibašić, 2007). Ove metode se mogu koristiti samostalno ili kombinovane u tzv. hibridne strategije pronalaženja. Sve metode pretraživanja pretvaraju vrednosti više indeksa u jednu izvedenu vrednost (agregaciju), na osnovu koje se vrši sortiranje i izbor najbližih, odnosno najsličnijih slučajeva (Suknović & Delibašić, 2010).

2.1.5.3.1. Metoda najbližeg suseda

Metoda najbližeg suseda predstavlja najčešće korišćenu metodu za računanje blizine između novog problema i starih problema.
Podrazumeva da se sličnost postojećeg slučaja sa novim slučajem određuje na osnovu težinskog zbira indeksa.

Najveći problem kod ove metode je određivanje težina.
Problem težina (pondera) je i inače jedan od najvećih problema nauke o odlučivanju, pre svega zbog toga što se često dešava da male promene težina mogu imati veliki uticaj na rešenje. Ova pojava čini određivanje pondera još značajnijim. Za definisanje pondera koriste se sve raspoložive tehnike otkrivanja zakonitosti u podacima, statistika, itd. Često se konsultuje i ekspert iz oblasti, ali i dalje problem težina ostaje kritična tačka nauke o odlučivanju.

Ograničenje ove metode je i sporost pronalaženja rešenja. Brzina metode je linearno zavisna od broja slučajeva u bazi znanja. Ova metoda je, prema tome, korisnija kada je baza slučajeva relativno mala.

Za funkciju sličnosti numeričkih podataka se obično koristi Euklidsko rastojanje, ali treba imati u vidu da Euklidova norma predstavlja samo jedan od načina na koji je moguće meriti sličnost. Euklidsko rastojanje se može koristiti kao mera sličnosti samo za numeričke podatke, dok se za kategoričke primenjuju posebne mere sličnosti. Neke od njih će biti predstavljene u nastavku. Teorijski postoji beskonačno mnogo načina na koje može da se računa sličnost, kako kategoričkih, tako i numeričkih podataka. U ovom radu, kao i u radovima (Delibašić, 2004; Vuković i ostali, 2012) je prikazano kako teorija preferencije može da se koristi za merenje sličnosti između numeričkih podataka.

S obzirom na značaj metode najbližeg suseda za ovaj rad, ovoj temi će se u nastavku posvetiti dodatna pažnja.

2.1.5.3.2. Indukcija

Induktivni metod podrazumeva organizaciju postojećih slučajeva u strukturu stabla po atributima (osobinama), tako da slični slučajevi mogu biti brzo indeksirani i pronađeni. Uopšteno, algoritmi indukcije, među kojima je najpoznatiji ID3, određuju koje osobine najbolje diskriminišu slučajeve i generišu drvo odlučivanja za organizaciju slučajeva u memoriji. Ovaj pristup je koristan kada se koristi leksikografski način razmišljanja ili kada na osnovu pravila tipa IF-THEN može da se izmeri sličnost.
Kada su indeksi kategoričkog tipa i kada postoji samo jedan izlazni atribut moguće je nad bazom slučajeva sprovesti algoritam stabla odlučivanja ID3, koji određuje strukturu kako će slučajevi biti pronađeni.
Kada se pojavi novi problem, sistem ZOS pronalazi odgovarajući slučaj krećući se niz stablo odlučivanja. Svaki put kada se doda novi slučaj u bazu slučajeva, potrebno je ponovo sprovesti induktivni algoritam stabla odlučivanja (Suknović & Delibašić, 2010).

2.1.5.3.3. Indukcija vođena znanjem

Indukcija vođena znanjem je hibridni metod pretraživanja baze slučajeva, u kome se od korisnika traži da zada osobine koje ima novi slučaj, a koje nisu eksplicitno navedene u samom opisu novog slučaja. Ovaj algoritam zahteva znanje, koje korisnik mora da poseduje, i primenjuje to znanje u procesu indukcije znanja, određujući ručno osobine slučajeva koji su poznati ili za koje se misli da rešavaju postavljeni problem. Ovaj pristup se često koristi zajedno sa drugim pristupima, zato što pravo znanje nije uvek na raspolaganju, pogotovu kod velikih baza znanja. Kod ove metode korisnik se navodi da kroz niz pitanja dođe do slučaja koji ga zanima. Ovaj pristup je zasnovan na konceptima ekspertnih sistema i na IF-THEN pravilima (Delibašić, 2007).

2.1.5.3.4. Otkrivanje paterna

Otkrivanje paterna ima za cilj da otkrije sve slučajeve koji se uklapaju u okviru određenih parametara. Ovo podrazumeva definisanje opsega u okviru koga, numerički, mogu da se nalaze sve vrednosti slučajeva, najčešće izraženih preko minimalne i maksimalne vrednosti. Ova metoda se često koristi pre ostalih metoda, kao što je metoda najbližeg suseda, da bi se pretraživanje ograničilo na bitni deo baze slučajeva (Delibašić, 2007).

Kod svih navedenih načina agregacije više informacija se sliva u jednu, na osnovu koje može da se odredi najbliži (najbolji, najprihvatljiviji) slučaj za rešenje problema. Drugim rečima, sve metode za pretraživanje baze slučajeva imaju zadatak da iz skupa slučajeva nađu podskup onih slučajeva koji na najtačniji način mogu da reše novi problem.

Nakon izvršene agregacije slučajeva, koja podrazumeva da je sistem donosiocu odluke ponudio rešenja na osnovu kojih može da odlučuje, prelazi se na sledeću, poslednju fazu - analizu rešenja.

2.1.5.4. Analiza rešenja

Analiza rešenja podrazumeva proveru mogućnosti primene starog rešenja za rešavanje novog problema. Staro rešenje je rešenje pronađenog sličnog slučaja, ili više njih, do kojih se došlo primenom izabranih mera sličnosti.

Moguće je da se pronađeni korisni slučajevi na različite načine prilagode novom problemu i za to postoje posebni mehanizmi (više u: Watson i Farhi, 1994). Prilagođavanje se zasniva na uočavanju bitnih razlika starog i novog problema i izmena starog rešenja da bi se odgovorilo zahtevima novog problema.

U ovom radu će se koristiti samo takozvana *Nulta adaptacija*, što je najjednostavnija tehnika jer ne vrši nikakvu adaptaciju. Donosilac odluke dobija slučajeve sortirane po sličnosti sa novim problemom. Ova tehnika se inače koristi za probleme koji imaju složeno zaključivanje, ali jednostavno rešenje. Npr, davanje kredita je vrlo složeno ali je rešenje jednostavno: odobriti ili odbiti zahtev za kredit.

Tehnike adaptacije kod ZOS-a mogu da pomognu da se sagleda da li je slučaj dobro strukturiran, da li je izvršena kvalitetna normalizacija, da li je agregacija urađena korektno. Dodatno, omogućava da se postojeća baza slučajeva efikasnije iskoristi za rešavanje novog problema. Ukoliko analitičar proceni da je novi slučaj vredan pamćenja, isti je moguće sačuvati u bazi slučajeva.
Ovakvim načinom rada, sistem ZOS napreduje tokom vremena i postaje korisniji.

ZOS je disciplina koja pomaže izgradnji tzv. organizacionog znanja. Za ZOS je bitno što baze slučajeva čuvaju upravo onakvo znanje kakvo je potrebno za odlučivanje, to je znanje sa dodatkom - akcijom. Kada se javi određeni problem i kada se reši, jednostavno je zapamtiti taj slučaj u bazi podataka. Prilikom rešavanja novog problema, mogu se naći sva rešenja koja su rešavala probleme slične postojećem. Zatim se dobija uvid kako bi novo rešenje trebalo da izgleda i ono se pamti za buduće potrebe. Ako se dva puta rešava isti problem u istom kontekstu (svi parametri su isti ili dovoljno slični), pamti se rešenje koje na bolji način rešava problem. Ovim se postiže poboljšavanje određenog poslovnog procesa.

2.1.6. Prednosti i nedostaci ZOS metodologije

ZOS pruža mnoge prednosti, među kojima su i (Kolodner, 1992):

- ZOS omogućava brzo predlaganje rešenja za probleme, skraćujući vreme potrebno da se odgovori izvedu polazeći od nule.

Na primer, lekari mogu imati koristi ukoliko se sećaju starih dijagnoza ili iskustava lečenja. Sistem ZOS-a dobija prednost prilikom rešavanja problema jer lako može da generiše predloge. Postoji značajna prednost u tome što ne moraju iznova da se prave dugotrajni proračuni i zaključci. Ova prednost je korisna za skoro sve faze zaključivanja, uključujući rešavanje problema, planiranje, objašnjenje i dijagnozu.

- ZOS dozvoljava donosiocu odluke da predloži rešenja u oblastima koje ne razume u potpunosti.

Mnoge oblasti je nemoguće u potpunosti razumeti, često i zato što mnogo zavisi od nepredvidivog ljudskog ponašanja. Takva je, na primer, ekonomija. Isto tako, dešava se da se ne razume uvek, na primer, kako neki lekovi i bolesti deluju. Ili, jednostavno se nađemo u situacijama koje ne razumemo dobro, ali u kojima svakako moramo da delujemo. ZOS omogućava da se daju pretpostavke i predviđanja na osnovu onoga što se desilo u prošlosti bez potpunog razumevanja.

- ZOS donosiocu odluke daje sredstvo za procenu rešenja kada ne postoji algoritamski metod dostupan za evaluaciju.

Slučajevi se mogu koristiti kao pomoć prilikom ocene i kada postoji mnogo nepoznanica, tj. kada su druge vrste evaluacije nemoguće ili teške. Umesto njihove primene, rešenja se kod ZOS-a ocenjuju u kontekstu prethodnih sličnih slučajeva, odnosno zaključivanje se radi na osnovu onoga što je rađeno u prošlosti.

- Slučajevi su naročito korisni za upotrebu u tumačenju otvorenih i nedovoljno definisanih koncepata.

Ovo je jedan od razloga zbog koga advokati intenzivno koriste slučajeve prilikom zaključivanja. Takođe, ovo je bitan razlog korišćenja slučajeva i u svakodnevnim situacijama. Kada se zna veoma malo informacija i kada je problem nedovoljno definisan, tumačenja ZOS metodologijom mogu biti tačnija od metoda klasifikacije zasnovanih na generalizaciji.

- Sećanje na prethodno iskustvo može biti posebno korisno kao upozorenje na potencijalne probleme koji su se dogodili u prošlosti.

Korišćenjem ZOS-a donosilac odluke se upozorava da preduzme odgovarajuće akcije kojima bi se izbeglo ponavljanje grešaka iz prošlosti. Iskustva iz prošlosti mogu biti kako uspesi tako i neuspesi, odnosno, situacije u kojima se stvari nisu odvijale baš kako je planirano.

- Slučajevi omogućavaju izgradnju alata za učenje, tako što dobri primeri iz prošlosti predstavljaju osnovu za učenje kako se određeni problem uspešno rešava.

- Kao metod za izgradnju inteligentnog sistema zaključivanja, ZOS je privlačan jer se čini relativno jednostavnim i prirodnim.

Veoma je teško da eksperti mogu da prenesu svoje celokupno znanje koje koriste za rešavanje problema, ali im je zato relativno lako da ispričaju svoje iskustvene priče. Veliki problem prilikom zaključivanja u stručnim oblastima je visok stepen neizvesnosti i nepotpuna znanja. ZOS sistem rešava te probleme tako što zaključuje oslanjajući se na ono šta je rađeno u prošlosti. ZOS sistemi takođe pružaju efikasnost. Ukoliko bi se problem svaki put rešavao iznova, često bi se trošilo značajno vreme kako na celokupno rešavanje problema, tako i na nalaženje prvih principa za rešavanje problema. U tom smislu, ZOS sistemi mogu pomoći bržem nalaženju rešenja.

- Slučajevi pomažu da se zaključivanje usredsredi na važne delove problema, ističući one karakteristike problema koje su važne.

ZOS metodologija podrazumeva težnju da ono što je bilo važno u prethodnim situacijama da bude važno i u novim. Dakle, ako je u prethodnom slučaju, neki skup karakteristika prouzrokovao neuspeh, zaključivanje treba da se usredsredi na one karakteristike koje će osigurati da se propust neće ponoviti. Slično tome, ako su neke karakteristike doprinele uspehu, bitno je izdvojiti te osobine. Ovakvo usredsređivanje je bitno i u rešavanju problema i za tumačenja ZOS metodologijom.

Uprkos navedenim prednostima, postoje i određeni nedostaci koje ZOS metodologija nosi sa sobom.

(Kolodner, 1992) ističe da je to pre svega činjenica da donosilac odluke može pasti u iskušenje da slepo koristi iskustva iz prošlosti za zaključivanje, oslanjajući se na prethodna iskustva bez validacije istih u novoj situaciji. Donosilac odluke može dozvoliti da slučajevi imaju loš uticaj na njega, ukoliko se isključivo vodi njima u rešavanju novog problema. Često se ljudi, naročito početnici, ne podsete najpogodnijih skupova slučajeva za zaključivanje. Upravo iz ovakvih razloga je korisno izgraditi sistem za podršku odlučivanju koji bi pomogao ljudima da pronađu najbolje slučajeve iz prošlosti. Psiholozi su otkrili da su ljudi naviknuti da koriste slučajeve prilikom odlučivanja, ali da se ne sete uvek odgovarajućih situacija iz prošlosti. Upravo iz ovog razloga, računar može da se koristi kao sredstvo koje bi pomoglo pretraživanje (Kolodner, 1992).

Zaključivanje na osnovu slučajeva je metodologija i zaključivanja i učenja. ZOS metodologijom se uči na dva načina. Prvo, zaključivanje može postati efikasnije prisećajući se starih rešenja i prilagođavajući ih umesto da se svaki put od početka izvode potpuno novi odgovori. Ako se slučaj adaptira na nov način, ako je rešen korišćenjem nekih novih metoda, ili ako je rešen kombinovanjem rešenja nekoliko slučajeva, prilikom podsećanja na njega, tokom kasnijeg zaključivanja, koraci potrebni da se problem reši neće morati da se ponavljaju za novi problem. Drugo, sistem

zaključivanja zasnovan na slučajevima postaje kompetentniji vremenom, proizvodeći bolje odgovore nego što je mogao sa manje iskustva. Jedan od aduta ZOS-a je i u tome što pomaže da se u procesu zaključivanja predvide i time izbegnu greške koje su napravljene u prošlosti. To je moguće jer se beleže problematične situacije, i indeksiraju se preko osobina koje predviđaju stare greške. Sećanje na ove slučajeve tokom kasnijeg zaključivanja upozorava na probleme koji bi mogli da se pojave, a mogu se izbeći (Kolodner, 1992).

ZOS postiže mnogo od svog učenja na dva načina: kroz prikupljanje novih slučajeva i kroz dodelu indeksa. Novi slučajevi daju ZOS sistemu poznatiji kontekst za rešavanje problema ili procenu situacije. Sistem zaključivanja čiji slučajevi pokrivaju veći deo oblasti zaključivanja će biti bolji od onog čiji slučajevi pokrivaju manji deo. Sistem čiji slučajevi pokrivaju kako neuspešne, tako i uspešne slučajeve će biti bolji od onog čiji slučajevi pokrivaju samo uspešne situacije (Kolodner, 1992).

ZOS predstavlja jedan način izgradnje sistema menadžmenta znanja, odnosno može pomoći u procesu upravljanja usvajanja znanja i njegovog ponovnog korišćenja (Watson, 2003). U savremenom dobu, kada se znanje vrednuje više od fizičkih resursa, organizacije koje efikasno koriste znanje, odnosno prethodno iskustvo, postižu bolje rezultate i veću konkurentnost.

2.1.7. Primena ZOS metodologije

ZOS može značiti prilagođavanje starih rešenja kako bi se zadovoljili novi zahtevi; korišćenje slučajeva iz prošlosti da bi se objasnile nove situacije; korišćenje prethodnih slučajeva da bi se kritikovala nova rešenja; ili zaključivanje iz ranijih da bi se protumačila nova situacija ili kreiranje adekvatnog rešenja za novi problem (Kolodner, 1992).

Kada se govori o primeni ZOS metodologije, može se govoriti o primeni od strane ljudi koji zaključuju na osnovu slučajeva, ili pak o mašinskom učenju koje podrazumeva primenu ove metode za zaključivanje.

Ako malo bolje sagledamo način na koji ljudi rešavaju probleme, verovatno ćemo shvatiti da je zaključivanje na osnovu slučajeva u upotrebi svuda oko nas.
ZOS je svesno ili nesvesno u osnovi mnogih naših izbora i rešenja, kada planiramo svakodnevne aktivnosti, prisećamo se šta je već obavljeno i šta je preostalo da se uradi i to koristimo za pravljenje novih planova; u izboru hrane u restoranu uglavnom se oslanjamo na prethodna iskustva i biramo ono što volimo; kada se nađemo u nekoj konfliktnoj situaciju, često se prisetimo koje smo uspešne argumente ranije koristili za smirivanje situacije (Kolodner, 1992).

Drugo suočavanje sa nekim problemom ili zadatkom je najčešće lakše nego prvo, upravo iz razloga što pamtimo i ponavljamo prethodno iskustvo. Drugi put smo stručniji, prisećamo se grešaka i tražimo način da ih izbegnemo.

Kvalitet rešenja onog ko zaključuje na osnovu prethodnog iskustva, zavisi od četiri stvari (Kolodner, 1992):
- iskustva koje je imao,
- sposobnosti da razume nove situacije u kontekstu starih iskustava,
- njegove veštine da se prilagodi i
- njegove veštine da izvrši dobru procenu.

Većina ljudi koristi prethodno iskustvo smatrajući to efikasnim načinom da se reše problemi.

Postoje dva stila ZOS-a: rešavanje problema i tumačenje (Kolodner, 1992).

ZOS rešavanje problema podrazumeva da se rešenja novih problema dobijaju korišćenjem starih rešenja kao vodiča. Stara rešenja mogu da pruže uglavnom ispravna rešenja za nove probleme i mogu da upozore na eventualne greške ili propuste.
Kod ZOS tumačenja, nove situacije se procenjuju u kontekstu starih situacija.

Generalno, ZOS može da se koristi za bolje sagledavanje i razumevanje problema, kao i za njegovo rešavanje.

Činjenica je, međutim, da za oba stila ZOS-a važi da u velikoj meri zavise od mehanizma za pretraživanje slučajeva, koji navodi na korisne slučajeve u odgovarajuće vreme, i u oba stila, skladištenje novih situacija u memoriji omogućava učenje iz iskustva. Stil rešavanja problema koristi prilagođavajuće procese za generisanje rešenja, dok stil tumačenja iste koristi za procenu izvedenih rešenja. Drugim rečima, interpretativni stil koristi slučajeve da pruži obrazloženja za rešenja, što omogućava procenu rešenja kada nema dostupnih jasnih metoda, kao i za tumačenje situacije kada su definicije granica situacije otvorene ili rasplinute (Kolodner, 1992).

ZOS rešavanje problema je korisno za rešavanje širokog spektra zadataka, uključujući planiranje, dijagnosticiranje i osmišljavanje (projektovanje ili dizajn). U svakom od njih, slučajevi su korisni u predlaganju rešanja i za upozoravanje o mogućim problemima koji mogu nastati (Kolodner, 1992).

Planiranje je proces dolaska do niza koraka ili rasporeda za postizanje nekog stanja. Stanje koje treba postići može da se odredi u konkretnom smislu, ili opisujući krajnji rezultat isporuke. Ili može biti određeno u smislu ograničenja koja moraju biti zadovoljena.

U prvom slučaju, krajnji proizvod procesa planiranja je skup koraka. U drugom, krajnji proizvod je raspored ili stanje, ali se proces planiranja mora koristiti za kreiranje istog (Kolodner, 1992).

Postoje određeni problemi koji se mogu javiti u procesu planiranja. Prvi je problem zaštite da koraci iz plana koji kasne ne ponište rezultate ranijih koraka. To zahteva da se efekti koraka iz plana projektuju u budućnost (ostatak plana). Drugi je problem preduslova - planer mora biti siguran da su preduslovi za bilo koji korak iz plana ispunjeni pre raspoređivanja za taj korak. Dva navedena problema zajedno, kada se rešavaju tradicionalnim metodama, zahtevaju dosta računanja. Kako broj planskih koraka raste, računska složenost projektovanja efekata i poređenje preduslova se povećava eksponencijalno.

ZOS se bavi ovim problemima obezbeđujući planove koji su već korišćeni i na kojima su ovi problemi već razmatrani. Planer je potreban samo da bi napravio relativno manje popravke, pre nego da za potpuno planiranje od nule (Kolodner, 1992).

Učenost obezbeđuje skup pravila i može da se koristi za kreiranje približnih planova. Ali učenost daje pravila za situacije u celini, a ne za određene situacije. Na polju planiranja, kao i u mnogim drugim situacijama, mali detalji situacije su važni za adekvatnost i vrednost plana. Postoji mnogo detalja koji mogu da budu prisutni, a samo su neki važni. Postoje brojne situacije planiranje gde je nemoguće znati sve o celokupnoj situaciji, ali ipak dobar plan mora biti kreiran, a mora se ocenjivati na osnovu projektovanih efekata (Kolodner, 1992).

Slučajevi omogućavaju projektovanje posledica sadašnjih planova na osnovu onoga što se dešavalo u prošlosti. Slučajevi sličnih planova koji su se u prošlosti pokazali kao propusti mogu da ukažu na potencijalne probleme plana. Ako prethodni plan sličan trenutno predloženom nije uspeo iz nekog razloga, donosilac odluke se upozorava da proceni trenutnu situaciju sa tog aspekta i da, na osnovu toga, eventualno revidira svoju strategiju. Slučajevi sličnih planova koji su se u prošlosti pokazali kao uspesi daju kredibilitet trenutnom planu. Pored toga, kada su delovi plana ocenjuju, slučajevi mogu pomoći u tome (Kolodner, 1992).

Dijagnostika podrazumeva davanje na uvid skupa simptoma i traženje objašnjenja za iste. Kada postoji mali broj mogućih objašnjenja, neko može dijagnostiku videti kao

problem klasifikacije. Kada se skup objašnjenja ne može lako nabrojati, na dijagnostiku se može gledati kao problem stvaranja objašnjenja. Dijagnostika na osnovu slučajeva može koristiti slučajeve da predloži objašnjenja za simptome i da upozori na objašnjenja koja su pronađena kao neprimerena u prošlosti (Kolodner, 1992).
Generisanje dijagnoze od nule je dugotrajan zadatak. U gotovo svim dijagnostičkim domenima, međutim, postoji dovoljan broj pravilnosti za pristup zasnovan na slučajevima kojim bi se generisala dijagnoza na efikasan način. Sugestija na osnovu slučajeva treba da bude potvrđena. Često je, ipak, provera mnogo lakša nego celokupno generisanje. U ovakvim oblastima, ZOS zaista može da doprinese mnogo (Kolodner, 1992).

Tokom **projektovanja** (osmišljavanja), problemi su definisani preko skupa ograničenja, a onaj ko rešava problem je dužan da obezbedi konkretna rešenja koja zadovoljavanju ograničenja problema. Obično data ograničenja bliže određuju problem, tj. postoji mnogo mogućih rešenja. Ponekad, međutim, ograničenja previše ograničavaju problem, odnosno, ne postoji rešenje koje ispunjava sva ograničenja. U tom slučaju, rešavanje problema zahteva ponovno određivanje problema, tako da su najvažnija ograničenja ispunjena, dok je za ostala nađeno kompromisno rešenje.
Ograničenja obično daju smernice, ali ne usmeravaju onog ko rešava problem ka nekom određenom odgovoru. Pored toga, najčešće je pretraživački prostor ogroman, tako da postoje mnogi odgovori koji bi bili moguća rešenja, ali oni su dovoljno retki u prostoru pretrage da bi standardne metode pretrage provele dugo vremena u pronalaženju jednog od njih. Često se dešava i da je problem preveliki da bi mogao da se reši u jednom komadu i da delovi problema komuniciraju jedni sa drugima značajno. Rešavanje pojedinačnih, manjih delova problema u izolaciji i njihovo ponovno vraćanje u celinu će skoro uvek narušavati interakcije između delova.
Za ove vrste problema, koji su teško rastavljivi na podprobleme, slučajevi mogu biti traženo rešenje. Umesto rešavanja problema rastavljajući ga na delove, rešavanja parcijalnih delova, i ponovnog sastavljanja, što može da se uradi samo sa nekim problemima, slučaj ukazuje na celokupno rešenje, a delovi koji se ne uklapaju u novu situaciju se prilagođavaju (Kolodner, 1992).

Dok god je moguća adaptacija da se staro rešenje uklopi u novu situaciju, ova metodologija je skoro uvek bolja od generisanja potpuno novog rešenja, „od nule", a naročito kada postoji veliki broj ograničenja i kada rešenja delova problema ne mogu lako da se sastave ponovo (Kolodner, 1992).

Rešavanje problema prilagođavajući staro rešenje omogućava da se izbegne bavljenje mnogim ograničenjima, a ujedno ne postoji potreba da se problem razlaže na podprobleme kojima je nakon rešavanja potrebno prekomponovanje.

Tumačenje zasnovano na slučajevima je proces procene situacija i rešenja u kontekstu prethodnog iskustva. Ono uzima situaciju ili rešenje kao ulaz, a izlaz je klasifikacija situacije, argument koji podržava klasifikaciju ili rešenje, i/ili opravdanja koja podupiru ovaj argument ili rešenje. To je korisno za klasifikaciju, ocenu rešenja, argumentaciju, pravdanje rešenja, tumačenje, ili plan, i projekciju efekata odluke ili plana .
Realno gledano, tumačenja bazirana na slučajevima koristimo svakodnevno, pravdajući svoje ponašanje i postupke prošlim događajima, poredeći se sa drugima, ali takođe i za sagledavanje dobrih i loših strana rešenja problema. Advokati u velikoj meri koriste tumačenja zasnovana na slučajevima, pre svega kada koriste slučajeve da obrazlože argumente (Kolodner, 1992).

Generalno, često se ističu tri opšta zadatka gde je tumačenje bazirano na slučajevima korisno: opravdavanje, tumačenje, i projekcija. Kod opravdanja, treba da se pokaže uzrok ili dokaz o čvrstini argumenta, pozicije ili rešenja. U tumačenju se pokušava da se nova situacija postavi u kontekst. Projekcija znači predviđanje efekata rešenja. Sve ove zadatke deli zajednička nit argumentacije. Neki slučajevi će podržati jednu interpretaciju ili efekat. Ostali će podržati neku drugu. Onaj koji zaključuje mora da uporedi i suprotstavi slučajeve jedan naspram drugog da bi došao do konačnog rešenja (Kolodner, 1992).

Opravdavanje i obrazlažuće zaključivanje znači kreiranje važnih argumenata koji treba da ubede druge da smo mi ili naši stavovi ispravni. Pri donošenju ubedljivih argumenata, moramo navesti stav i podržati ga, ponekad sa čvrstim činjenicama, a

ponekad i sa važećim zaključcima. Često je jedini način da se opravda pozicija upravo navođenje relevantnih dosadašnjih iskustava ili slučajeva. Primena ove vrste zaključivanja na osnovu slučajeva je najčešća u oblasti prava. Uopšteno, slučajevi su korisni pri izboru argumenata i opravdavanju pozicije /stava kada ne postoje konkretni principi ili postoji samo nekoliko njih, ako su principi u suprotnosti, ili ako značenja principa nisu dobro precizirana (Kolodner, 1992).

Uopšteno, **klasifikacija i tumačenje** u kontekstu zaključivanja na osnovu slučajeva znači odlučiti da li neki koncept odgovara klasifikaciji sa otvorenim ili rasplinutim granicama. Klasifikacija može biti izvedena u hodu, ili može biti dobro poznata, ali ne i dobro definisana u smislu neophodnih, i dovoljnih uslova. Mnoge klasifikacije pretpostavljamo da su definisane kao klasifikacije otvorene vrste (Kolodner, 1992).
Jedan od načina da se uradi klasifikacija zasnovana na slučajevima jeste da se zapita da li je novi slučaj dovoljno sličan prethodnim. Umesto klasifikacije novih slučajeva koja se radi korišćenjem neophodnih i dovoljnih uslova, klasifikaciju je moguće uraditi i pokušavajući da se pronađe slučaj iz baze koji najviše odgovara novom slučaju. Novi slučaj se klasifikuje na osnovu pripadajućih klasa njemu najsličnijih slučajeva iz prošlosti.

ZOS objašnjavanje, tj. **objašnjavanje zasnovano na slučajevima** (Schank & Abelson, 1977) podrazumeva da se neki fenomen može objasniti prisećajući se sličnog fenomena, pozajmljujući njegovo objašnjenje, i prilagođavajući ga da se uklopi, da odgovara novom fenomenu.
Objašnjavanje zasnovano na slučajevima zahteva mehanizam pretrage koji može da pronađe slične slučajeve, mehanizam adaptacije koji mora biti sasvim kreativan, i mehanizam validacije koji može da odluči da li predloženo objašnjenje ima neku vrednost (Kolodner, 1992).

Mnogo rada na tumačenju je usmereno na domen prava i opravdavanja argumenata za ili protiv nekih tumačenja zakona. Međutim, tumačenje zasnovano na slučajevima ne mora biti usmereno samo za interpretativne probleme. Naprotiv, ono može doprineti i kao deo vrednosne ili kritičke komponente rešavanja problema i odlučivanja kad god

nedostaju jaki uzročni modeli. Procesi uključeni u tumačenje zasnovano na slučajevima imaju potencijal da odigraju nekoliko važnih uloga za rešavanje problema. Prvo, ako je okvir rešenja poznat, ili ako su poznata prisutna ograničenja, ove metode mogu da se koriste za odabir slučajeva koji će omogućiti takvo rešenje. Drugo, kreiranje argumenata i opravdanja rezultuju znanjem koje su to karakteristike, osobine slučaja koje su važne i na koje se treba fokusirati. Znanje na šta se fokusirati je takođe važno u rešavanju problema. Treće, sporedni efekat je da se ukaže na osobine koje, ako postoje, bi mogle da daju bolje rešenje. Na kraju, metode tumačenja se mogu koristiti i za predviđanje upotrebljivosti, kvaliteta, ili rezultata rešenja (Kolodner, 1992).

Iako su neke primene već navedene, generalno ZOS metodologija se može uspešno primenjivati u mnogim oblastima, pre svega kao podrška odlučivanju. Do sada su se sistemi ZOS-a koristili kod pravljenja baze znanja korisničkih servisa (npr. mobilni operatori). Pored toga, može se uspešno primeniti i u mnogim drugim oblastima, kao što su medicina i proizvodna industrija (npr. Hsu i ostali, 2004; Im & Park, 2007; Tseng i ostali, 2005), na probleme segmentacije (npr. Chen i ostali, 2010; Changchien & Lin, 2005; Chiu, 2002; Chun & Park, 2006), na polju finansija, uključujući i ocenu kreditne sposobnosti (npr. Bryant, 1997; Buta, 1994; Wheeler & Aitken, 2000; Shin & Han, 2001).

Ciljevi primene ZOS-a kao dela sistema menadžmenta znanja mogu biti:

- ✓ povećanje konkurentnosti poboljšavajući procese, kao i kvalitet proizvoda/ usluge,
- ✓ povećanje konkurentnosti povećavajući efikasnost, efektivnost, i smanjujući troškove,
- ✓ povećanje konkurentnosti skraćivanjem procesa razvoja proizvoda,
- ✓ fokus na rezultate i ciljeve,
- ✓ smanjenje vremena proizvodnje/ procesa odlučivanja,
- ✓ korišćenje informacija i tehnologije,...

Znajući trenutne procese i probleme koje treba rešiti, u bazi znanja se mogu pronaći slični slučajevi sa ciljem pronalaženja najboljih praksi koje su već bile primenjene.

Sa tehničke strane gledano, postoje brojni argumenti koji podržavaju korišćenje ZOS-a pre nego druge na znanju zasnovane metodologije. Istraživači tvrde da ZOS obezbeđuje potencijal za razvoj sistema menadžmenta znanja mnogo lakše nego pristupi koji su zasnovani na pravilima ili modelima. Oni tvrde da konkretni primeri opisani slučajevima doprinose lakšem razumevanju i primeni od strane korisnika, u različito struktuiranim kontekstima rešavanja problema, i to mnogo lakše od kompleksnih stabala zaključivanja generisanih pravilima ili modelima. Mogućnost validacije i ažuriranja rešenja obezbeđuje okvir za učenje iz iskustva, na taj način što se usvojeno znanje uključuje kao deo svakodnevne primene ZOS aplikacije (Allen, 1994).

Zašto bi u postupku donošenja logičkih odluka, lekar, ili bilo ko drugi stručno obučen, zaključivao na osnovu slučajeva? Na primer, lekar je obučen da koristi činjenice i znanje, ali on je obučen da prepozna poremećaje u izolaciji, kao i uobičajene kombinacije poremećaja. On takođe zna uzroke i posledice poremećaja, odnosno kako poremećaji napreduju. Međutim, on ne može biti obučen da prepozna baš sve kombinacije poremećaja, a znanje koje ima o patološkim procesima zahteva dosta vremena da bi se ustanovila moguća dijagnoza. Ako je već koristio znanje o bolesti da na jednom slučaju reši teži problem, ima smisla da se to rešenje sačuva na takav način da se može ponovo koristiti (Kolodner, 1992).

2.1.8. Primena ZOS metodologije na probleme kredit skoringa

Problem odlučivanja u oceni kreditne sposobnosti i merenje rizika su veoma značajni i teški zadaci komercijalnih banaka i finansijskih institucija zbog visokog rizika pogrešnog odlučivanja.

Metodologija kredit skoringa zahteva iskustvo na bazi stručnosti. Prilikom rešavanja novih problema, stručnjaci se oslanjaju na prethodne scenarije. Oni moraju da znaju koji kreditni plasmani su bili uspešni, a koji su propali. Oni takođe treba da znaju kako da se izmeni stari slučaj i da se prilagodi novoj situaciji. ZOS je opšta paradigma iskustveno zasnovanog rasuđivanja. Ona pretpostavlja memorijski model za predstavljanje, indeksiranje i organizovanje prethodnih slučajeva i procesni model za pronalaženje i menjanje prethodnih slučajeva i njihovo prilagođavanje novim (Slade, 1991).

Moguće primene modela bodovanja u praksi mogu se generalizovati kao sledeće (Yang L., 2002):

- Modeli bodovanja se koriste kao metod analize za automatsko donošenje kreditne odluke. Ovako nešto se može primenjivati na kreditne proizvode koji se daju u većem broju i koji podrazumevaju malo novca, kao na primer u slučaju kreditnih kartica. U ovakvim situacijama, ljudska intervencija na kreditnim odlukama je potrebna samo na pojedinačnim spornim i važnim slučajevima.

- Modeli bodovanja se koriste kao jedna od alatki za analizu i integrisani su u proces kreditnog odlučivanja ili u sistem kreditnog odlučivanja. Sistem kreditnog odlučivanja se može sastojati od modela bodovanja i drugih alata, kao i iskusnih stručnjaka. Tako na primer, kreditni proces može biti tako organizovan da se kreditni zahtev prvo ocenjuje od strane modela bodovanja. Ako se slučaj klasifikuje kao „dobar“, kredit je odobren. Slučajevi klasifikovani kao „loši“ se pojedinačno preispituju od strane iskusnih kreditnih analitičara.

- Rezultati modela bodovanja mogu da se koriste i kao ulazni podaci nekog rejting sistema ili portfolio modela. Ponekad se modeli bodovanja ponašanja ne koriste samo za direktne odluke o kreditu, već se takođe koriste kao deo drugih sistema ili modela za upravljanje kreditnim portfoliom. Na primer, rezultati modela bodovanja mogu da se koriste za utvrđivanje rejtinga za kreditni rizik. Kompanije se svrstavaju u nekoliko nivoa rizika shodno dobijenim ocenama, eventualno uz prilagođavanja na osnovu mišljenja eksperata prema drugim faktorima, kada je to neophodno.

U fazi primene modela u praksi, sistematski bi trebalo pratiti stvarno ponašanje komitenata koji otplaćuju odobrene kredite. Ovi podaci, s jedne strane, treba koristiti za potvrdu ispravnosti modela bodovanja, a kako bi se osigurala njegova efikasnost - da li se i u kojoj meri kreditne odluke slažu sa stvarnim ponašanjem kreditnih klijenata. Validacija rezultata generiše povratnu informaciju za reviziju modela. S druge strane, pravi kreditni primeri treba da budu novi uzorci koji se koriste za obnovu modela.
Novoizgrađeni model bodovanja se obično zasniva na ograničenim prošlim slučajevima, dok je u procesu obnove na raspolaganju više prethodnih slučajeva, možda sa više novih informacija. Dakle, sa povećanjem nagomilanih slučajeva iz prošlosti i sa relevantnijim informacijama, kreditni model će postati još pouzdaniji. Uspostavljanje modela kreditnog bodovanja je iterativni proces. Za dobar model su nekada potrebne godine da se prikupi dovoljno podataka i da se uspostavi dobar sistem, koji nakon toga treba uvek da se redovno revidira i da se prilagođava uvek novim varijacijama iz okruženja (Yang L., 2002).

2.1.9. Poređenje metoda za rešavanje problema kredit skoringa

Postoji niz metoda koje se mogu koristiti za razvoj modela za ocenu kreditne sposobnosti. Neke od metoda su statističke, dok se neke od njih oslanjaju na pristupe u kojima se primenjuje veštačka inteligencija. Statističke metode, često korišćene za kredit skoring, su: višestruka regresija (npr. Meyer & Pifer, 1970), diskriminaciona analiza (npr. Altman, 1968; Banasik i ostali, 2003), i logistička regresija (npr. Dimitras i ostali, 1996; Martin, 1977; Elliott & Filinkov, 2008; Lee i ostali, 2002; Desai i ostali,1996), dok metode veštačke inteligencije uključuju induktivno učenje (npr. Han i ostali, 1996; Shaw & Gentry, 1998), veštačke neuronske mreže (npr. Boritz & Kennedy, 1995; Coakley & Brown, 2000; Jo & Han, 1996; Zhang i ostali, 1999; Lee & Chen, 2005; West, 2000), genetske algoritme (npr. Desai i ostali, 1997; Yobas i ostali, 2000; Huang i ostali, 2006, 2007), i veštačke imune sisteme (npr. Leung i ostali, 2007).

U nastavku će se ukratko uporediti najpopularnije metode, i to Linearno diskriminacionu analizu (LDA), Logističku regresiju (LR) i Veštačke neuronske mreže (VNM).

Tehnike Diskriminacione analize se koriste da bi se jedinka klasifikovala u jednu od dve ili više alternativnih grupa (ili populacija) na bazi niza merenja, odnosno diskriminacionom analizom se identifikuju varijable kojima se vrše diskriminacija (razgraničenje, razlikovanje) između jedinica posmatranja deleći ih u dve ili više grupa. Cilj diskriminacione analize je da se pronađe linearna kombinacija nezavisnih varijabli kojom se maksimizira diskriminacija između dve ili više grupa i minimizira verovatnoća pogrešnog klasifikovanja u odgovarajuće grupe.

Logistička regresija je poznata tehnika iz grupe prediktivnih tehnika, gde je zavisna promenljiva diskretna, najčešće binarna (ima dva modaliteta, odnosno kategorije), a nezavisne promenljive (jedna ili više) mogu biti kvantitativne ili kvalitativne, pri čemu u logističkoj regresiji ne postoje pretpostavke o raspodeli za ove promenljive. Pored toga što ovaj model daje odgovor na pitanje pripadnosti svake opservacije jednoj od dve grupe, takođe, daje i odgovarajuće verovatnoće pripadnosti. Cilj regresione analize je da

oceni populacijsku srednju vrednost zavisne varijable na osnovu poznatih vrednosti nezavisnih varijabli.

Smatra se da modeli logističke regresije imaju određenih prednosti nad modelima diskriminacione analize i to pre svega zbog manje restriktivnih pretpostavki modeliranja, koje se tiču linearnosti, uslova normalnosti, kao i nezavisnosti između nezavisnih promenljivih. Pristup logističke regresije ne zahteva ispunjenost svih ovih uslova, pa stoga ostavlja više fleksibilnosti u radu sa stvarnim podacima.

Veštačka neuronska mreža (VNM) je model koji imitira biološke neuronske mreže, a koristi se za rešavanje problema predviđanja, procene i klasifikacije. VNM može da modeluje proizvoljnu funkciju zavisnosti, kako linearnu, tako i nelinearnu. Takođe, VNM se mogu primeniti i u oblastima gde su podaci multivarijantni sa visokim stepenom međuzavisnosti između atributa.

Poređenje na osnovu ulaznih varijabli:

Postoje određeni preduslovi za primenu LDA. Standardni model pretpostavlja da za svaku od populacija važi normalna distribucija sa više promenljivih. Dalje se pretpostavlja da je matrica kovarijanse jednaka u obe populacije. Ipak, srednje vrednosti za date promenljive mogu biti različite u dve populacije.

Dakle, jedan od osnovnih zahteva za primenu LDA, što je ujedno i glavna zamerka za LDA, jeste pretpostavka da su varijable multivarijaciono normalno distribuirane i da su matrice kovarijansi jednake za grupe. Pravilo linearne diskriminacije ne može biti optimalno ako se pretpostavke nisu zadovoljile. Ako se LDA posmatra kao fleksibilna linearna kombinacija varijabli koja maksimizira određeni kriterijum razdvajanja, onda je ova metoda široko primenljiva. Iskustveno posmatranje kreditnog bodovanja je takođe pokazalo da narušavanje pretpostavke o normalnoj raspodeli u LDA, ipak nije sprečilo njegovu uspešnu primenu (Yang L., 2002).

Funkcija klasifikacije do koje se dolazi primenom LDA je linearna funkcija ulaznih promenljivih.

Višestruka kolinearnost između ulaznih varijabli kod LDA i LR ima negativne efekte na rezultate jer ona znači da je jedna promenljiva linearna kombinacija ostalih varijabli. S tim u vezi, uvek prethodno treba ispitati ulazne promenljive i obezbediti da nema snažno povezanih ulaznih promenljivih preostalih u setu podataka (Yang L., 2002). Dakle, bilo bi idealno da su prediktorske promenljive jako povezane sa zavisnom promenljivom, ali ne i međusobno.

Druga mana LDA i LR je da ove metode ne mogu da podrazumevaju interakcije. Interakcija se javlja kada korelacija između varijable i zavisne promenljive zavisi od vrednosti drugih varijabli. Da bi se rešio ovaj problem, interakcijska promenljiva, koji je proizvod dve ili više promenljivih može biti uključena u model (Yang L., 2002).

Jedan od uslova za primenu VNM je normalizacija ulaznih varijabli. Ulazne varijable mogu imati različite skale vrednosti, te bi njihovi uticaji na izlaz bili nejednako razmatrani, što dovodi do pristrasnih modela. Dakle, vrednosti svake ulazne promenljive treba da se transformišu da budu u istoj skali (npr. između 0 i 1). Neuronsko računanje inače može da procesira samo brojeve, što bi značilo da pre tretiranja od strane VNM, kvalitativni atributi ili slike, ako postoje u problemu, moraju prethodno biti procesirani na numeričke ekvivalencije.

Tvrdnje ljudi iz prakse ukazuju na to da performanse neuronskih mreža, stabala odlučivanja (DT - decision trees), i ZOS-a, u odnosu na konvencionalne metode zavise od udela loših kredita u skupu podataka. Obično, iste prethodne verovatnoće „dobrih“ i „loših“ zahteva/ slučajeva u uzorku za učenje mogu da poboljšaju performanse modela. Nasuprot tome, statistički algoritmi mogu da se izbore, u većoj ili manjoj meri, sa različitim proporcijama klasa u učećem uzorku (Yang L., 2002).

Iz uporednog prikaza zahteva za ulazne varijable za pet algoritama (Tabela 2.) može se videti da oštriji zahtevi ograničavaju upotrebljivost tradicionalnih metoda (LDA i LR) i da bi trebalo da se preduzmu dodatni predobradni zadaci. Sa druge strane, savremene metode (VNM, ZOS i stabla odlučivanja) mogu doći do znanja iz podataka sa malim pretpostavkama ili zahtevima za ulazne podatke.

Poređenje na osnovu kvaliteta generisanih modela:

Izlazi modela LDA i LR su u obliku bodova. Ove dve tehnike su najčešće korišćene za izradu tabeli bodovanja. Obično se koeficijenti i numeričke vrednosti atributa kombinuju da bi dali jedinstvene doprinose, čijim objedinjavanjem se dobija ukupan skor, broj bodova (rezultat).

Model LDA, kao model klasifikacije, podrazumeva da se za svaki slučaj (individuu) dobija tzv. Z score, što je njegova prosečna ocena posmatrajući sve karakteristike, nakon čega se poređenjem (stavljanjem u odnos) diskriminacionog Z score-a svakog pojedinačnog slučaja i granične vrednosti (cutting score-a) vrši klasifikovanje u unapred definisane grupe.

Pored izračunavanja Z-score-a za pojedinačni slučaj, potrebno je izračunati srednje ocene za svaku grupu (populaciju) po pojedinim varijablama, nakon čega se formira srednja ocena, tj. centroid na nivou čitave grupe. Centroid je praktično vrednost oko koje se koncentrišu Z score-ovi u jednoj grupi. Na osnovu centroida obe grupe određuje se Cutting score (granična tačka) – vrednost koja razdvaja jednu od druge grupe. Na osnovu Cutting scora i Z score-a pojedinačnog slučaja može se odrediti kojoj populaciji pripada slučaj (Z score>Cutting scora označava populaciju I, dok Z score<Cutting score-a označava populaciju II).

Kod modela logističke regresije za ocenjivanje regresionih koeficijenata najčešće se koristi metod maksimalne verodostojnosti. LR pokušava da proceni šanse da se vrednosti zavisne promenljive predvide korišćenjem nezavisnih promenljivih i to tako što se počinje sa slučajnim skupom koeficijenata, a zatim se isti iterativno unapređuje u zavisnosti od poboljšanja evidentirane verovatnoće uspešnosti. Nakon nekoliko iteracija, proces se zaustavlja kada dalja poboljšanja postanu neznatna, na osnovu nekih utvrđenih kriterijuma.

Rezultat modela LR, kao modela klasifikacije, je takodje određeni skor, ali on ima univerzalno značanje - na primer, skor od 0,75 za bilo koji od modela znači istu stvar, a to je da podrazumeva 75% šanse da slučaj spada u grupu sa vrednošću 1 (nasuprot grupi sa vrednošću 0) za taj model. Sa tačke gledišta korisnika, lakše je tumačiti rezultat

modela LR nego rezultate modela LDA, kod koga za dva različita modela isti rezultati mogu da znače različite stvari.

Izlazi metode NN mogu da budu kontinualne vrednosti ili pripadnost klasi.

Izlazi stabla odlučivanja su obično pripadnosti klasama, ali ima i stabala koja mogu dati numerička previđanja (kontinualne izlaze).

Izlaz ZOS metoda mogu biti bodovi ili klase. K-NN metod kao rezultat daje pripadnost određenoj klasi, dok numerička predviđanja ZOS metodom sa lokalno ponderisanom regresijom daju kontinualne rezultate.

Za metode sa bodovnim rezultatima, u slučaju da oni podrazumevaju dve klase, granična vrednost se može utvrditi odabirom željenog kompromisa između dobrih i loših rizika. Ovo povećava fleksibilnost donošenja odluka. Kada se koncept rizika davaoca kredita menja, ono što treba da se uradi jeste da se samo promene granični bodovi. Suprotno tome, metode koje za izlazni rezultat određuju pripadnost klasi nemaju ovu fleksibilnost, iako su generalno jasne. Kada se promeni definicija rizika, ovi modeli treba ponovo da se izgrade.

Tabela 2. Poređenja metoda za klasifikaciju

		Metode klasifikacije				
		LDA	LR	NN	DT	ZOS
Zahtevane ulazne vrednosti	kvantitativne i kvalitativne varijable	✓	✓	✓	✓	✓
	normalna distribucija vrednosti varijabli, jednake matrice kovarijansi	✓				
	problem interakcije	✓	✓			
	problem višestruke kolinearnosti	✓	✓			
	normalizacija varijabli			✓		✓
	osetljivost na učešće klasa			✓	✓	✓
Forma izlaza	bodovi	✓	✓	✓	✓	✓
	klase			✓	✓	✓
Učinak	tačnost klasifikacije	ne postoje jedinstveni rezultati na osnovu sprovedenih istraživanja				

Mali je broj javno dostupnih poređenja različitih tehnika za oblast kreditnog bodovanja. Objavljena literatura predlaže različito: neki smatraju da tradicionalne tehnike daju bolje rezultate od novih, dok drugi tvrde suprotno (Yobas i ostali, 2000). Neki autori pak zaključuju da postoji samo mala razlika u tačnosti klasifikacije između svake od metoda (Thomas, 2000).

Kada se predstavlja novi algoritam, autori uglavnom uspeju da dokažu njegovu superiornost nad stvarno uzetim istraživačkim skupovima podataka. Međutim, često je diskutabilno da li može da se dokaže superiornost istog algoritma nad drugim skupovima podataka. Istraživanje poređenja je pokazalo da nijedan metod ne može uvek da rezultatski nadmaši druge. Drugim rečima, za određeni skup podataka, postoji određeni optimalni algoritam (Yang L., 2002).

Učinak algoritma klasifikacije zavisi od toga kako projektant modela tretira skup podataka, na primer u smislu izbora ulaznih varijabli, ponašanja u slučaju nedostajućih

vrednosti, i sl., i kako se biraju parametri modela. Da bi se izabrao optimalni metod za određeni skup podataka, ovi faktori treba da budu uzeti u obzir (Yang L., 2002).

2.2. Metoda najbližeg suseda

Cilj pronalaženja slučajeva u okviru ZOS metodologije je da se nađu najkorisniji prethodni slučajevi koji će voditi optimalnom rešavanju novog slučaja, ignorišući prethodne slučajeve koji su irelevantni. Pronalaženje slučajeva u ZOS sistemu se odvija na sledeći način: na osnovu opisa novog slučaja traže se prethodni slučajevi koji imaju potencijal da pruže podršku odlučivanju. Pretraga može biti uz ograničenja, čime se obično pronalazi veliki broj prethodnih slučajeva. Ipak, moguće je filtrirati prethodne slučajeve i na osnovu kriterijuma isključivosti, što podrazumeva upoređivanje i filtriranje. Prethodni slučajevi koji ostanu posle filtriranja se uparuju i rangiraju prema opadajućem stepenu sličnosti. Uparivanje je proces koji procenjuje stepen sličnosti potencijalno korisnih prethodnih slučajeva sa novim slučajem (Montazemi & Gupta, 1997).

Slučaj se može smatrati šemom koja se sastoji od skupa atributa (deskriptora). Uparivanje podrazumeva utvrđivanje sličnosti šeme novog slučaja sa šemama prethodnih slučajeva. Ono uključuje dva koraka:
(1) procenu sličnosti novog i prethodnog slučaja preko svakog od deskriptora, i
(2) procenu ukupne sličnosti preko odgovarajuće funkcije.

Sličnost dva slučaja duž deskriptora može biti procenjena preko pravila podudaranja specifičnih za određeni domen. Tako na primer, pravilo podudaranja može da utvrdi da je deskriptor "boja objekta" sa vrednošću narandžasta veoma sličan deskriptoru sa vrednošću crvena (Montazemi & Gupta, 1997). Međutim, najčešće bi bio potreban izuzetno veliki broj odgovarajućih pravila da bi se utvrdila sličnost svih mogućih parova vrednosti za deskriptore. Samim tim, prihvatanje odgovarajućih pravila može biti veoma težak zadatak.

Ukupna sličnost novog slučaja prethodnom se procenjuje preko agregirane sličnosti duž deskriptora korišćenjem odgovarajuće funkcije uparivanja. U ovom radu, za procenu ukupne sličnosti je korišćena funkcija najbližeg suseda (nearest neighbour - NN). Uparivanje metodom najbližeg suseda je najčešće korišćeno u ZOS sistemima (Montazemi & Gupta, 1997). Ukupna sličnost novog slučaja T i prethodnog slučaja S korišćenjem NN funkcije se dobija na sledeći način:

$$Sličnost\ (T,S) = \frac{\sum_{i=1}^{F} w_i sim(a_i^T, a_i^S)}{\sum_{i=1}^{F} w_i} \quad (1)$$

gde je w_i težina (značaj) osobine (deskriptora) *i*, *T* je ciljni (trenutni) slučaj za koji se traži rešenje, *S* je izvorni (postojeći, pronađeni) slučaj, *F* je broj atributa u svakom slučaju, *i* je pojedinačna karakteristika (osobina, atribut), počev od *1* do *F*, dok je a_i vrednost samih atributa za posmatrane slučajeve.

NN funkcija podudaranja procenjuje ukupnu sličnost preko ponderisane linearne kombinacije sličnosti duž atributa. Ovo je slično metodama koje se koriste kod višeatributivnog odlučivanja. Ponderisanje ukazuje na stepen značaja atributa za rešenje problema odlučivanja. NN funkcija uparivanja je usvojena iz literature o otkrivanju paterna (obrazaca). U svrstavanju u paterne, svi prethodni slučajevi se prestavljaju preko istog skupa atributa - deskriptora i njihov značaj se određuje pomoću induktivne tehnike mašinskog učenja koja minimizira grešku klasifikacije. Ovaj pristup je donekle izvodljiv u ZOS sistemima. Uvek treba imati u vidu da broj deskriptora koji može da opiše prethodne slučajeve može biti izuzetno veliki, i da se kod ZOS sistema samo podskup deskriptora može koristiti da se opiše određeni prethodni slučaj.

Atributi- deskriptori u ZOS sistemima imaju značaj na dva nivoa, globalnom i lokalnom. Na globalnom nivou, značaj atributa je isti bez obzira na prethodni slučaj u kome se koristi, a na lokalnom nivou, značaj deskriptora je specifičan za prethodni slučaj. Globalni nivo je grub i neosetljiv na kontekst. Nasuprot tome, lokalni nivo je detaljniji i osetljiv na kontekst. U nekim sistemima ZOS-a, stepen važnosti atributa na

lokalnom nivou se usvaja od strane stručnjaka iz domena preko inženjera znanja. Međutim, procene važnosti deskriptora dobijene od strane eksperta iz domena mogu biti „neobjektivni“. Osim toga, značaj deskriptora dobijenih od stručnjaka iz domena je statična i nezavisna od prethodnih slučajeva iz baze slučajeva. Alternativni pristup je da se utvrdi stepen značaja, dinamički, prilikom pretrage. Na primer, pravila specifična za oblast se mogu koristiti za određivanje važnosti deskriptora tokom pretrage. Tokom pretrage, ovaj pristup uzima u obzir kontekst novog slučaja. Ipak, potreba da se odrede pravila za ovu metodologiju ograničava njenu primenu.

Nepravilna primena rešenja prethodnog slučaja na novi slučaj može da rezultira takozvanom preteranom generalizacijom. Preterana generalizacija može da se desi kada rešenje prethodnog slučaja nije primenljivo zbog određenih uslova koji postoje u novom slučaju. Da bi se sprečila preterana generalizacija, produkciona pravila se koriste da se proceni valjanost prethodnog prema novom slučaju. Produkciona pravila imaju dva ograničenja. Prvo, ona pretpostavljaju dobro definisano domensko znanje, i drugo, njihovo sticanje od stručnjaka iz domena je puno poteškoća. Zato, umesto pravila, često se predlaže korišćenje ograničenja koja uzimaju u obzir nesavršenosti domenskog znanja i pružaju metod učenja zasnovan na objašnjenjima kako bi se usvojila ova ograničenja.

Jedna od najkorisnijih mogućnosti ZOS-a je njegova sposobnost da efikasno pronalazi relevantne slučajeve iz baze slučajeva. Korisnost slučajeva iz prošlosti određena je njihovom sličnošću sa novim slučajem. Uspešnost mehanizma pronalaženja zavisi od uspešnosti opisa slučaja, indeksiranja i pokazatelja sličnosti (Park & Han, 2002). Kao što je već istaknuto, metode koje se najčešće koriste za pronalaženje slučajeva su: metoda najbližeg suseda (nearest neighbour - NN), induktivno učenje i vođenje znanjem, ili kombinacija svih ovih (Buta, 1994). U ovom radu se za pronalaženje slučajeva koristi metod najbližeg susreda (NN), što je ujedno i najčešće korišćen metod. NN može da se koristi za klasifikaciju novog slučaja, pronalazeći slučajeve iz baze koji imaju najsličnije vrednosti atributa, i dodeljujući novom slučaju klasu najbližeg suseda, ili više njih.

Svaki slučaj iz baze podataka ima svoju klasu, što je kategorička promenljiva od interesa (npr. kreditno sposoban ili ne), kao i veliki broj dodatnih prediktorskih varijabli (npr. starost, prihod, bračni status, itd.). Za novi slučaj, za koji treba da se uradi klasifikacija, NN algoritam pronalazi najbliži, novom slučaju najsličniji postojeći slučaj iz baze podataka i dodeljuje kategoriju tog najbližeg suseda novom slučaju. NN funkcija je neparametarski algoritam klasifikacije zasnovan na pretpostavci o nezavisnosti atributa u prethodnim slučajevima i na dostupnosti pravila i procedura za uparivanje (Park & Han, 2002)..

Najveći nedostatak klasične NN funkcije je osetljivost na prisustvo irelevantnih osobina u predstavljanju slučaja. To je zato što njena funkcija sličnosti, koja se obično predstavlja preko Euklidove funkcije odstojanja, pretpostavlja da su sve osobine jednako značajne. To znači da svaka osobina ima pojednak uticaj na izračunavanje sličnosti.

Značajan napredak u preciznosti klasifikacije može biti postignut algoritmima ponderisanja atributa. To znači da najrelevantnije osobine (atributi) imaju najveće težine. Sveobuhvatna sličnost je određena ponderisanom NN funkcijom sličnosti, koja se matematički može prikazati na sledeći način (Kolodner, 1993):

$$Sličnost\ (T,S) = \sqrt{\sum_{i=1}^{F} w_i (T_i - S_i)^2} \qquad (2)$$

gde je w_i težina (značaj) osobine *i*, *T* je ciljni (trenutni) slučaj za koji se traži rešenje, *S* je izvorni (postojeći, pronađeni) slučaj, *F* je broj atributa u svakom slučaju, dok je *i* pojedinačna karakteristika (osobina, atribut), počev od *1* do *F*.

Adekvatno postavljanje težina u funkciju sličnosti može poboljšati performanse NN algoritma. Važnijim atributima treba dodeliti veće težine nego manje bitnim, dok nebitnim atributima treba dodeliti težine jednake nuli (Park & Han, 2002).

Svakom kriterijumu je potrebno dodeliti obeležje kojim će se istaći njegova važnost i uticaj na konačno rešenje. Veličina važnosti kriterijuma (težina) utiče značajno na rešenje, ali je najčešće za njegovo definisanje potrebno ekspertsko znanje. Često se dešava da male promene težina utiču na velike promene u rešenju.

Dosta istraživača se bavilo empirijskim radom na postavljanju težina k-NN algoritama. Mnogi istraživači predlažu da se težine karakteristika prihvate na osnovu znanja od stručnjaka iz domena (Kolodner, 1993), ili da se dobiju tehnikama mašinskog učenja kao što su genetski algoritmi (Shin & Han, 1999) ili pak statističkim metodama kao što su višestuka diskriminaciona analiza i regresija. Pojedini istraživači su predlagali pristup kombinovanja algoritma stabla odlučivanja i k-NN metode (Kibler & Aha, 1987). Wettschreck i Aha (1995) su predstavili pristup dodele kontinualnih težina atributima u k-NN algoritmu jednostavno po njihovim vrednostima informacionih dobiti, tj. po količini dodatnih informacija koju unosi svaki od atributa u sistem. Park & Han (2002) prilagođavaju AHP metodologiju za ponderisanje atributa kako bi ono bilo zasnovano na domenskom znanju, a što je u skladu sa mišljenjem mnogih autora da je znanje iz oblasti bitno za proces zaključivanja.
Nekoliko studija je pokazalo da se skup težina atributa može odrediti koristeći razne algoritme učenja. U ovom radu, težine su određivane koristeći genetske algoritme.

K-NN metoda je modifikovana, proširena verzija NN metode, koja daje bolje rezultate klasifikacije jer traži k najbližih slučajeva novom slučaju (k je obično neparan broj) i predlaže rešenje koje se javlja najčešće u tih k najbližih, najsličnijih slučajeva (Chen i ostali, 2010). Na primer, za problem kredit skoringa, gde bi trebalo doneti odluku da li odobriti kredit ili ne, k-NN metod treba da pomogne u donošenju odluke – da li zahtev za kredit treba da bude odobren ili odbijen. Drugim rečima, pošto se novi problem (slučaj) pojavi, ZOS sistem će (na osnovu funkcije sličnosti) pronaći k najsličnijih slučajeva iz baze (k-NN) i zaključiti kojoj klasi pripada novi slučaj. Grupa (klasa) kojoj većina od k slučajeva pripada je i grupa kojoj pripada novi slučaj (Chen i ostali, 2010).

K-NN metod koristi funkciju sličnosti za generisanje klasifikacije od sačuvanih slučajeva. Nekoliko studija je pokazalo da je učinak k-NN algoritma veoma osetljiv na

izbor ove funkcije, tako da se u cilju smanjenja ove osetljivosti predlažu mnoge k-NN metode.

Glavni nedostaci K-NN metode su:

(1) niska efikasnost - kao lenja metoda učenja ne preporučuje se za mnoge aplikacije kao što su dinamički veb mining sa ogromnim skladištima podataka, i
(2) zavisnost od izbora adekvatne vrednosti za k.

Kako bi se donekle prevazišao problem adekvatnog izbora vrednosti za k, u ovom radu je razmatrano ponašanje modela na nekoliko različitih vrednosti.

2.3. Funkcije preferencija

Osnovna slabost tradicionalnog (tzv. čistog) ZOS-a je proces validacije preko Euklidove norme, gde se svi kriterijumi posmatraju na jednak način, a jedini način da se izrazi određena preferencija prema nekom kriterijumu predstavljaju ponderi, što je za rešavanje praktičnih problema, često, nedovoljno pouzdano.

Proces validacije se donekle može unaprediti korišćenjem teorije preferencije za pretraživanje baze znanja i za vrednovanje znanja. Teorije preferencije može da pruži više mogućnosti pri izražavanju preferencija donosioca odluke u traženju odgovarajućeg slučaja.

Normalizacija ne može kvalitetno da se uradi preko metrika ili preko koncepata dominacije. Nije logično da se alternative čije se vrednosti „malo“ razlikuju, tretiraju kao i one koje se razlikuju „mnogo“. Potrebno je uvesti određene kriterijume za dominaciju i poređenje. Uvođenjem tipova preferencije se može od donosioca odluke zahtevati ekspertsko znanje.

Mogu da se koriste funkcije preferencije preko kojih se izražava kada postoji dominacija i kolika je ona. Koristi se upoređivanje u parovima (Pair-wise comparison).

Uvođenjem funkcija preferencije definišu se pragovi i oblici funkcija koji određuje preferencije jednih alternativa u odnosu na druge, kao i njihove konkretne vrednosti. Time se javlja se potreba za ekspertskim znanjem i aktivnijem učešću DO u odlučivanju.

U tom slučaju, može se reći da postoji određena vrsta preterane generalizacije u računanju udaljenosti, tako da nije moguće potpuno izraziti ekspertovu indiferentnost ili striktnu preferenciju prema razlikama u vrednostima atributa između dva slučaja. Sa druge strane, pravilna ograničenja su uslovi (tj. domensko znanje) koji sprečavaju prekomernu generalizaciju. Pravilna ograničenja ili specijalizuju prethodni slučaj ili ograničavaju njegovu primenljivost (Montazemi & Gupta, 1997).

ZOS je efektivan metod koji integriše metodologiju zaključivanja sa predstavljanjem domenskog znanja. Domensko znanje može biti inkorporirano u izgradnju ZOS modela, bilo preko relativnog značaja kriterijuma ili preko odnosa višeg ranga (outranking

relations - OR), uključujući striktne, slabe razlike, ili indiferentnost, između vrednosti atributa slučajeva za svaki pojedinačni atribut.
Parametri za odnose višeg ranga mogu biti određeni od strane stručnjaka iz oblasti, ili se mogu utvrditi, na primer, pristupom genetskog algoritma (GA) , kao što se predlože u ovom radu.
Teško je odrediti ekspertova pravilna ograničenja, u slabo strukturiranim problemima odlučivanja. Problem je u izboru pravilnog ograničenja i njegovoj parametrizaciji (prag ravnodušnosti i/ili prag stroge preferencije) za svaki atribut.
Ovo predstavlja priličan problem, jer ekspert, od velikog broja mogućih vrednosti, mora izabrati granice. Stoga se u radu predlaže korišćenje GA za objektivnu izbor i parametrizaciju odgovarajućih ograničenja.
Da bi se donekle ispravio proces validacije, za merenje sličnosti između slučajeva, kada su u pitanju numerički atributi, u radu se koriste funkcije preferencija predložene u čuvenoj metodi višekriterijumskog odlučivanja Promethee (Brans & Vincke, 1985; Mareschal, 1986, 1988; Mareschal & Brans, 1988). Za svaki atribut može biti definisana drugačija funkcija preferencije.

U Promethee metodi, funkcije preferencije se zasnivaju na poređenju parova alternativa duž svakog prepoznatog kriterijuma (u problemu kredit skoringa, postoje poređenja vrednosti atributa ciljnog, novog slučaja i svakog istorijskog slučaja iz baze). Za svaki kriterijum, funkcija preferencije prevodi razliku između vrednosti atributa iz dve alternative (dva slučaja) u stepen preferencije u rasponu od nula do jedan.
Za svaki atribut i svaki postojeći slučaj u bazi, u klasičnom pristupu, rastojanje između novog slučaja, odnosno vrednosti njegovih atributa i postojećih slučaja iz baze, odnosno vrednosti njihovih atributa, meri se na sledeći način:

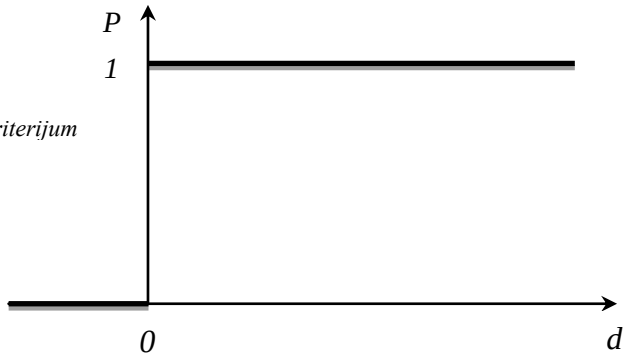
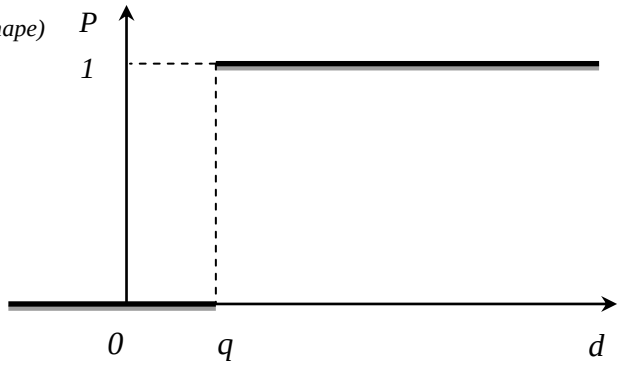
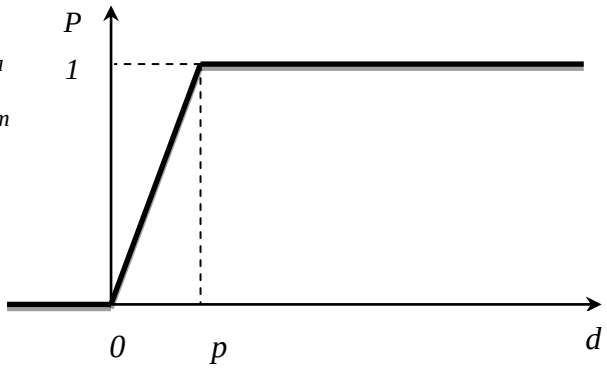
$$d_i = T_i - S_i \quad (3)$$

gde je d_i razlika u vrednostima na *i*-tom atributu između *T* - ciljnog (trenutnog) slučaja i *S* - izvornog (postojećeg) slučaja iz baze.

Promethee metod proširuje klasičan pristup, dozvoljavajući različite funkcije preferencija $p_i(d_i)$ za svaki kriterijum, tako da se za svaki atribut odražava nivo

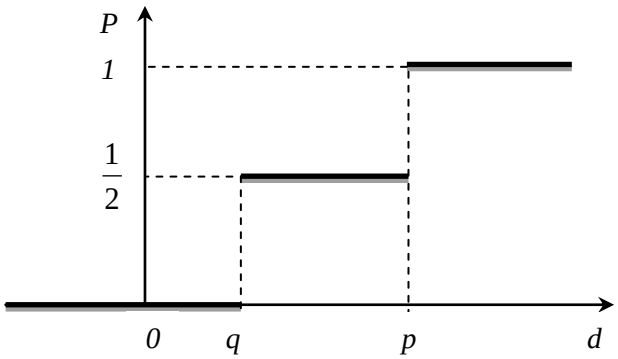
preferencije S_i preko T_i koji se kreće od 0 do 1. Ukoliko je $p_i(d_i) = 0$, alternative se smatraju indiferentnim jedna prema drugoj na *i*-tom atributu. Ako je $p_i(d_i) = 1$, jedna alternativa je striktno preferirana u odnosu na drugu na *i*-tom atributu. Neke funkcije preferencija imaju interval indiferentnosti, definisan pragom q_i koji omogućava da se pokaže indiferentnost između dve alternative na nekom atributu ne samo kada one imaju iste vrednosti tog *i*-tog atributa (tj. kada je $d_i = 0$), već i kada su te vrednosti samo slične jedna drugoj ($0 < d_i < q_i$). Osim toga, ove funkcije mogu da imaju blagu tranziciju između indiferentnosti i stroge preferencije, što omogućava pravljenje ocena preferencija sa različitim intenzitetima. Vrednosti između 0 i 1 izražavaju intenzitet preferencija na takav način da $p_i(d_i) \sim 0$ ukazuje na slabu preferenciju, dok $p_i(d_i) \sim 1$ ukazuje na snažnu sklonost (Parreiras & Vasconcelos, 2007).

Neki autori (Brans i ostali, 1986) predlažu glavne vrste funkcija preferencija, koje pokrivaju većinu praktičnih situacija (slika 3).

Funkcija preferencije	*Definicija*	*Parametri za definisanje*
<u>Tip 1:</u> *Običan kriterijum* 	$P(d) \begin{cases} 0 & d \le 0 \\ 1 & d > 0 \end{cases}$	-
<u>Tip 2:</u> *Kvazi (U-shape) kriterijum* 	$P(d) \begin{cases} 0 & d \le q \\ 1 & d > q \end{cases}$	q
<u>Tip 3:</u> *Kriterijum sa linearnom preferencijom (V-shape)* 	$P(d) \begin{cases} 0 & d \le 0 \\ \frac{d}{p} & 0 \le d \le p \\ 1 & d > p \end{cases}$	p

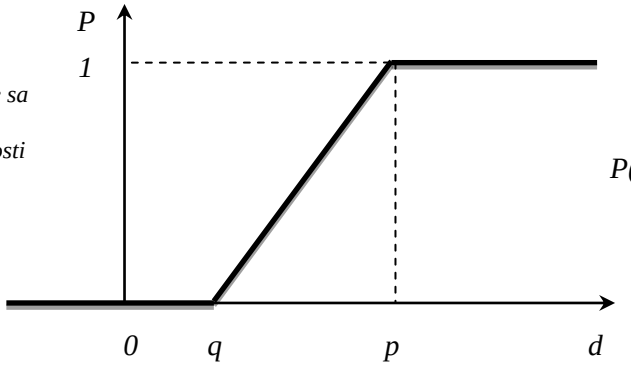
Funkcija preferencije	Definicija	Parametri za definisanje

<u>Tip 4:</u>
Nivo kriterijum



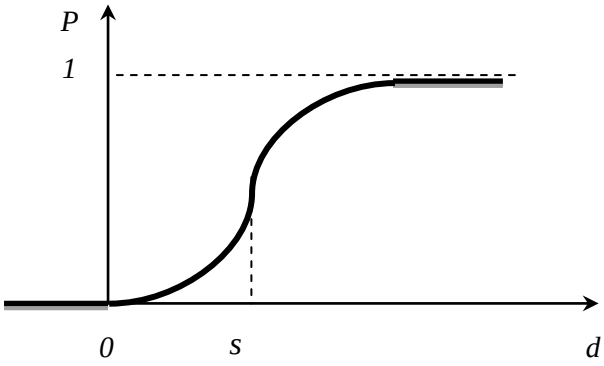
$$P(d)=\begin{cases}0 & d\le q\\ \frac{1}{2} & q< d\le p\\ 1 & d>p\end{cases}\qquad p,\ q$$

<u>Tip 5:</u>
Kriterijum linearne preferencije sa područjem indiferentnosti



$$P(d)=\begin{cases}0 & d\le q\\ \frac{d-q}{p-q} & q< d\le p\\ 1 & d>p\end{cases}\qquad p,\ q$$

<u>Tip 6:</u>
Gaussov kriterijum



$$P(d)=\begin{cases}0 & d\le 0\\ 1-e^{-\frac{d^2}{2s^2}} & d>0\end{cases}\qquad s$$

Slika 3. Vrste funkcija preferencija

Nula, jedan ili dva parametra moraju da budu definisani za svaku funkciju preferencija. Značenje ovih parametara je:

- *q* je prag ravnodušnosti, indiferencije;
- *p* je prag stroge preferencije;
- *s* je srednja vrednost između *p* i *q*.

Prag indiferencije *q* je najveće odstupanje, koje se smatra zanemarljivim, dok je prag stroge preferencije *p* najmanje odstupanje koje se smatra dovoljnim za generisanje pune prednosti, odnosno preferencije.

Funkcija preferencije **običan kriterijum** može da se koristi samo u slučajevima kada donosilac odluke ne može da izdvoji značaj razlike između vrednosti kriterijuma, tako da što je veća razlika veča je i preferencija. Ova funkcija ne zavisi od parametara, odnosno za ovu vrstu funkcije preferencije nije potrebno definisani donju i gornju graničnu vrednost. Ova funkcija može biti predložena samo u slučajevima kada je jedino važno da razlika d_i između vrednosti T_i i S_i pozitivna (p(d)= 1) ili negativna (p(d)= 0), a vrednost razlike nije bitna.

Tako na primer, jedna ponuda za posao je bolja od druge, ako je ponuđena plata veća bez dodeljivanja bilo kakve važnosti samoj vrednosti razlike; zatim važno je ako je udaljenost do kancelarije veća ili manja; ako su kamatne stope koje banke nude na oročene depozite veće ili manje; ako je dužina radnog iskustva između dva kandidata za posao veća ili manja; ako je cena između dva investiciona projekta veća ili manja; ako kandidat za posao zna više jezika od drugog kandidata; ako je brzina procesora jednog računara veća ili manja od drugog i slično.

Drugi oblik funkcije preferencija **kvazi kriterijum** razlikuje se od prethodnog, običnog kriterijuma, po postavljanju vrednosti donje granice *q*, počev od koje se razlika vrednosti primenjenog kriterijuma smatra da podstiče strogu opredeljenost za jednu alternativu u odnosu na drugu. Dakle, kada je razlika d_i veća od *q*, vrednost funkcije preferencije iznosi jedna, dok je p(d) = 0 kada je $d_i \leq q$.

Ova funkcija ima veći praktični značaj u odnosu na prvu običnu funkciju preferencija. Gore navedeni primeri se lako mogu prilagoditi da odgovaraju slučaju funkcije kvazi kriterijuma. Tako na primer, novi posao će imati strogu preferenciju (p(d)=1) u odnosu na drugi u slučaju da se plata razlikuje za ne manje od 100 novčanih jedinica (q=100), odnosno određeni posao nije preferiranog značaja za zaposlenog (p(d)=0), ako je plata veća za manje od 100 novčanih jedinica u odnosu na drugu ponudu. Ponuda kamatne stope na depozit koju banka nudi može, na primer, biti od interesa u slučaju da kamatna stopa na oročene depozite prelazi 1% u odnosu na ponudu druge banke (q=1); određeni

kandidat će biti preferiran u slučaju da njegovo radno iskustvo prelazi radno iskustvo drugog kandidata za tri godine i više (q=3), ili ako je ispravno odgovorio na najmanje tri pitanja više od drugog kandidata i tako dalje.

Treći oblik funkcije preferencija - **kriterijum sa linearnom preferencijom** se razlikuje od prethodnog kriterijuma u intervalu od nula do *p*, gde je veza između tačke ravnodušnosti između alternativa (p(d)=0), do koje se alternative smatraju ravnopravnim, i tačke stroge preferencije jedne alternative nad drugom (p(d)=1) nije u obliku smene, već je ta veza linearna. Još jedna razlika je u postavljanju gornje granice parametra *p*, od koje jedna alternativa ima strogu preferenciju nad drugom, umesto postavljanja donje granice parametra *q*, do koje se dve alternative smatraju indiferentnim.
Opet, prethodni primeri se mogu blago korigovati kako bi se pokazalo delovanje ovog kriterijuma funkcije preferencije. Na primer, određena ponuda za posao će imati strogu prednost nad drugom u slučaju razlike u plati od 100 novčanih jedinica ili više, ponuda niće biti uopšte od interesa u slučaju da je plata koja se nudi niža (p(d)=0, kada je d_i negativno), dok će se interesovanje za ponudu postepeno povećavati u slučaju da je razlika između alternativa do 100 novčanih jedinica ($0< d_i \leq 100$). Vrednost funkcije preferencije onda može biti izražena po formuli: p(d)= $\frac{d_i}{100}$. Ostali primeri se mogu lako modifikovati na sličan način.

Četvrti oblik funkcije preferencija se naziva **nivo kriterijum**, a zavisi od dva parametra *q* i *p*, čime su obe granične vrednosti postavljene: granica ravnodušnosti *q* i stroge preferencije granica *p*. Dakle, u slučaju kada razlika vrednosti dve alternative d_i nije veća od *q*, onda je odnos prema alternativama indiferentan (p(d)=0); kada je razlika d_i veća od *p*, onda je jedna alternativa strogo preferirana u odnosu na drugu, dok kada god je razlika d_i između vrednosti *q* i *p* ili $d_i \in [q, p]$, tada vrednost funkcije preferencije iznosi 0,5. U ovom slučaju jedna alternativa ima srednju preferenciju nad drugom.

Na primeru posmatrano, može se reći da kandidat za posao neće imati nikakvu prednost ako zna manje stranih jezika od drugog kandidata (p(d)=0, d_i je negativno), imaće određenu prednost u slučaju ako zna jedan jezik više od drugog kandidata (p(d)=0,5), i imaće striktnu prednost u odnosu na drugog kandidata u slučaju da zna dva i više strana jezika u odnosu na drugog kandidata (p(d)=1). Slična funkcija preferencija, ali sa više nivoa gradacije može da se koristi u slučaju većeg broja diskretnijih opcija. Što se broj nivoa povećava, to se više aproksimira linearna funkcija.

Peti oblik funkcije preferencija je **kriterijum linearne preferencije sa područjem indiferentnosti**. Ovaj kriterijum podrazumeva oba parametra q i p, koji postavljaju granice ravnodušnosti i stroge preferencije. Kada kriterijumska razlika vrednosti dve alternative pada u interval od q do p ili $d_i \in [q, p]$, funkcija preferencije se ravnomerno linearno povećava od nule do jedan u skladu sa formulom $\frac{d-q}{p-q}$ i njena vrednost ukazuje na nivo preferencije jedne alternative nad drugom. U slučaju kada je q=0 ova funkcija postaje treći oblik funkcije preferencije.

Prethodno uzimani primer se ponovo može lako transformisati u ovom konkretnom slučaju. Zaposleni će indirefentan ako se plate između dve ponude za posao razlikuju od manje od 100 novčanih jedinica (p(d)=0). Jedan posao će biti strogo preferiran u slučaju da plata na tom radnom mestu premašuje 500 novčanih jedinica (p(d)=1), dok će određeni posao biti od neke prednosti u odnosu na drugi posao u slučaju da je referentna plata veća za iznos između 100 i 500 novčanih jedinica. Tada se nivo preferencija izračunava po formuli $p(d) = \frac{d-100}{500-100} = \frac{d-100}{400}$.

Ostali primeri se mogu lako transformisati na sličan način.
Ova funkcija se ističe kao najvrednija i privlači najveći broj teorijskih i praktičnih primena za ocene koje se izvode primenom Promethee metodama.

Šesti oblik funkcije preferencija - **Gaussov kriterijum** se koristi u slučaju kada se početni statistički podaci sastoje od slučajnih vrednosti sa normalnom raspodelom.

Prednost pri malim razlikama u vrednostima kriterijuma polako se povećava sa povećanjem d_i, počevši od nule. Isto se odnosi i na velike razlike d_i vrednosti kriterijuma, funkcija preferencije se u ovom slučaju postepeno približava vrednosti 1. Ova funkcija zahteva parametar σ što je standardna devijacija datih slučajnih podataka, a raste najbrže u onim vrednostima razlike d_i koje su blizu vrednosti σ.

Projektantu baze znanja mogu da posluže sledeće rečenice da bi se lakše snašao sa upotrebom funkcija preferencije (Brans i ostali, 1986):

Kriterijum 1. Dve alternative su različite po određenom kriterijumu, ako između njih postoji i najmanja razlika.

Kriterijum 2. Dve alternative su različite po određenom kriterijumu, ako su različite za više od *q*.

Kriterijum 3. Dve alternative su linearno (srazmerno) različite po određenom kriterijumu, ako je razlika između njih manja ili jednaka *p*, dok su totalno različite ako je razlika veća od *p*.

Kriterijum 4. Dve alternative su jednake ukoliko je razlika između njih manja ili jednaka *q*, upola različite (ili slične) ukoliko je razlika između njih veća od *q*, a manja ili jednaka od *p*, a totalno različite ukoliko je razlika veća od *p*.

Kriterijum 5. Dve alternative su jednake ukoliko je razlika između njih manja ili jednaka *q*, linearno različite ukoliko je razlika između njih veća od *q*, a manja ili jednaka od *p*, a totalno različite ukoliko je razlika veća od *p*.

Kriterijum 6. Razlika između dve alternative se meri Gausovim kriterijumom, pri čemu je standardna devijacija σ (S).

Pojedini autori predlažu nove funkcije preferencije, sa ciljem da se poveća mogući izbor funkcija koje bi bile adekvatne za različite praktične primene (Podvezko & Podviezko, 2010).

Kao slaba tačka PROMETHEE metode ističe se nepreciznost u proceni prioritetnih funkcija, koja se vrši pomoću relativno jednostavnih i krutih matematičkih operacija (Radojevic i ostali, 1997). S tim u vezi, navedeni autori predlažu teoriju fazi skupova i povezano rasuđivanje, kao odgovarajući okvir da se oponaša ljudsko rasuđivanje u izražavanju strukture preferencija. Fazi skupovi se posmatraju kao odgovarajuće sredstvo za tretiranje nepreciznih informacija. Na taj način, donosilac odluke bi za svaki kriterijum koristio pojmove kao što su „mala razlika“, „srednja razlika“ ili „velika razlika“, što bi mu omogućilo da izrazi svoju strukturu preferencija jezički i na prirodniji način. Pomenuti deskriptori ukazuju na sistem vrednosti donosioca odluke i zavisni su od konteksta. Prof. Radojević i ostali autori su uveli fazi AKO-ONDA pravila koja se odnose na razliku u vrednosti kriterijuma u funkcijama preferencija.

U disertaciji, izabran je peti tip funkcije preferencije, jer se smatra da je ova funkcija najvažnija i da privlači najveći broj teorijskih i praktičnih primena za evaluacije koje se sprovode metodom Promethee (Podvezko & Podviezko, 2010). Funkcija preferencije tip 5 koristi prag indiferencije i prag stroge preferencija, koje treba definisati. U ovoj studiji to je urađeno pomoću GA.

2.4. Merenje sličnosti podataka

Merenje sličnosti ili rastojanja između dva entiteta je ključni korak u nekoliko metoda otkrivanja zakonitosti u podacima, kao što su na primer: klasterovanje (k-means), otkrivanje izuzetaka na osnovu udaljenosti, klasifikacija (K-NN, SVM), itd. Ovi algoritmi obično tretiraju računanje sličnosti kao nezavisan korak koji može da podrazumeva korišćenje bilo koje mere sličnosti, tj. udaljenosti.

Sličnost između dve instance, dva objekata je numerička mera koja opisuje koliko su ti objekti slični. Što dva objekta više liče jedan na drugi sličnost im je veća. Obično se uzima da sličnost bude nenegativna vrednost i to u opsegu [0,1] gde 0 označava najmanju sličnost, a 1 najveću.
Različitost je suprotno od sličnosti, dakle numerička mera koja opisuje koliko su dva objekta različita. Što dva objekta više liče jedan na drugi različitost im je manja. Najmanja različitost je često 0, dok najveća nekad 1, ali nekad se dopušta i da ide u $+\infty$. Kao sinonim za različitost koristi se i termin rastojanje.

Ono što je suštinsko jeste kako se sličnost između dva objekta zaista računa. Tu ima raznih varijacija. Sličnost između dve instance direktno zavisi od sličnosti između vrednosti njihovih atributa.

Jedan od pristupa da se računa sličnost, a analogno i različitost, između instanci koji imaju veći broj atributa, gde ne moraju svi atributi biti istog tipa, jeste računanje sličnosti između svakog od atributa, a zatim njihovim kombinovanjem dobijamo neku vrednost u intervalu [0,1].

To kombinovanje može jednostavno biti prosek sličnosti pojedinačnih parova atributa. Mada, računanje proseka tretira sve parove atributa podjednako, što nekad nije poželjno. Nekad su određeni atributi mnogo bitniji od nekih drugih. Tada se koriste tzv. težine, parametri kojima kontrolišemo bitnost atributa u računanju proseka svih sličnosti. Za svaki atribut se određuje težina (broj) kojim se množi sličnost u izračunavanju proseka, ali sve težine su iz intervala [0,1] i zbir svih težina je 1.

2.4.1. Vrste podataka

Računanje sličnosti između pojedinačnih atributa zavisi od njihove vrste.

Promenljive mogu biti kvantitativne (numeričke) ili kvalitativne (atributivne).

Kvantitativne ili numeričke promenljive mogu brojčano da se iskažu. Promenljive koje se ne mogu brojčano izraziti, ali se mogu razvrstati u različite kategorije jesu kvalitativne (atributivne) ili kategorijske promenljive.
Svi kvantitativni podaci mogu da se urede. Postoje i kategorički podaci kod kojih je moguća uređenost, ali ne i tačno merenje razlike među njima, odnosno ne možemo govoriti ni o apsolutnoj ni o relativnoj veličini ovih razlika. To su redni (ordinarni) podaci. Kod imenskih (nominalnih) kategoričkih podataka nije moguće uspostaviti poredak.

Kvantitativne promenljive mogu da se klasifikuju kao prekidne (diskretne) promenljive ili neprekidne (kontinualne) promenljive.

Prekidna (diskretna) promenljiva je ona slučajna promenljiva čije vrednosti su prebrojive i konačne. Prekidna promenljiva može uzeti samo izolovane – diskontinualne vrednosti, ali ne i vrednosti u intervalu između njih. Diskretni atributi imaju konačan ili prebrojivo beskonačan skup vrednosti, pri čemu su binarni atributi specijalan slučaj diskretnih atriubuta.

Promenljiva koja može uzeti bilo koju numeričku vrednost u određenom intervalu ili intervalima jeste neprekidna (kontinualna) promenljiva. Skup vrednosti kontinualnih atributa čine realni brojevi.

Podela atributa je moguća i prema osobinama i operacijama koje mogu da se primene za podelu. Ako se za podelu koriste operacije: različitosti (= i ≠), uređenja (<, ≤, > i ≥), aditivnosti (+ i –) i multiplikativnosti (* i /), podaci mogu biti:
- imenski (nominalni),

- redni (ordinarni),
- intervalni i
- razmerni (racio).

U tabeli 3. su prikazane osobine ovih tipova podataka.

Tabela 3. Tipovi podataka

Tip atributa	Opis	Primeri	Operacije
Imenski (eng. Nominal)	Vrednost imenskog atributa su upravo različita imena, tj. imenski atributi pružaju samo mogućnost razlikovanja jednog od drugog objekta (=, ≠)	poštanski kodovi, identifikacije zaposlenih, boja očiju, pol (muški, ženski), vrste muzike, vrste industrija,...	način, entropija, korelacija kontingenata, Hi2 test
Redni (eng. Ordinal)	Vrednosti rednih atributa pružaju dovoljno informacija za uređenje objekata (<, >)	tvrdoća minerala, poređenje atributa (lep, lepši, najlepši), socijalna klasa (niža, srednja, viša), stanje pacijenta (dobro, srednje, ozbiljno i kritično), stručna sprema zaposlenih (osnovna, srednja,...)	procenat, korelacija ranga, izvršavanje testova, oznake testova
Intervalni (eng. Interval)	Za intervalne atribute, ima smisla razlika između vrednosti, tj. postoji jedinica mere takvih atributa (+, -)	datumi u kalendaru, temperatura u stepenima Celizijusa	srednja vrednost, standardna devijacija
Razmerni (eng. Ratio)	Kod razmernih atributa ima smisla i proizvod i količnik (*, /) tih atributa	temperatura u Kelvinima, količina novca, godine, masa, dužina	geometrijska sredina, harmonijska sredina, procenat varijacije

Kvalitativni podaci su oni koji su imenski ili redni, dok su kvantitativni oni koji su intervalni ili razmerni.

Sličnost slučajeva koji se sastoje i od kvantitativnih i od kvalitativnih atributa, u ovom radu se računa tako što se izračuvana sličnost odvojeno po svakom od atributa, nakon čega se pristupa izračunavanju agregatne ponderisane sličnosti.

Pojam sličnosti za kvantitativne podatke je relativno dobro razumljiv, ali za kategoričke podatke, računanje sličnosti nije nimalo jednostavno. U literaturi je za sada za računanje sličnosti između dva kategorička podataka predloženo nekoliko mera sličnosti vođenih podacima. U ovom radu se, između ostalog, proučavaju performanse različitih mera sličnosti u kontekstu određenog zadataka zaključivanja na osnovu slučajeva. Rezultati na različitim skupovima podataka uglavnom pokazuju da ne postoji mera koja je univerzalno dobra za sve vrste problema, ali primećuje se da pojedine mere imaju konstantno visoke performanse.

2.4.2. Merenje sličnosti kvantitativnih podataka

Za kvantitativne podatke, rastojanje Minkovskog je opšti metod koji se koristi da se izračuna udaljenost između dve tačke, a predstavlja maksimum razlike između odgovarajućih komponenti vektora.

$$Udaljenost\ (T,S) = (\sum_{i=1}^{F} |T_i - S_i|^r)^{\frac{1}{r}} \quad (4)$$

gde su: r parametar, F broj dimenzija (atributa), a T_i i S_i su vrednosti i-tih atributa objejata T i S.

Za r=1 dobija se Minkovski udaljenost prvog reda, tj. Menhetn (L1 norma) rastojanje ili Hamingovo rastojanje.
Za r = 2, dobija se Euklidsko rastojanje.
Kada r → ∞ dobija se „supremum“ (Lmax norma) rastojanje

Dve najčešće korišćene mere udaljenosti za kontinualne podatke su Minkovski udaljenost prvog reda (Manhattan udaljenost) i kao i Minkovski udaljenost drugog reda (Euklidska udaljenost). Za ove mere sličnosti je ključno da su one nezavisne od osnovnih podataka kojima se opisuju dve tačke. Nekoliko mera vođenih podacima, kao što je Mahalanobis udaljenost, su takođe ispitivane za kontinualne podatke.

2.4.3. Merenje sličnosti kategoričkih podataka

Pojam sličnosti ili udaljenosti kategoričkih podataka nije tako jednostavan kao za kvantitativne podatke. Ključna karakteristika kategoričkih podataka je da različite vrednosti koje kategorički atribut može da ima nisu suštinski uređene, odnosno vrednosti kategoričkih atributa ne mogu prirodno da se rangiraju i urede na određenoj skali. Iz tog razloga, nije moguće direktno uporediti dve različite kategoričke vrednosti (Boriah i ostali, 2008). Ova karakteristika čini da mnoge tehnike i analize istraživanja kvantitativnih podataka ne mogu da se primenjuju na kategoričke varijable. Najjednostavniji način određivanja sličnosti između vrednosti dva kategorička atributa je da se dodeli sličnost 1 ako su vrednosti identične i sličnost 0 ako te vrednosti nisu identične. Ova najjednostavnija mera je takođe poznata kao mera podudaranja (overlap). Za dva multivarijaciona kategorička slučaja, sličnost između njih će biti direktno proporcionalna broju atributa u kojima se poklapaju.

Osnovni nedostatak mere preklapanja je da ne pravi razliku između različitih vrednosti koje uzima atribut. Sva poklapanja, kao i nepoklapanja, se tretiraju jednako – ili su vrednosti jednake ili nisu. Učestalost pojave, tj. raspodela frekvencija, se u okviru ove mere sličnosti ne uzima u obzir i to je ono što čini meru preklapanja isuviše pojednostavljenom davajući jednaki značaj svim poklapanjima i neusklađenostima.
Iako ne postoje suštinska uređenost kod kategoričkih podataka, postoje druge informacije u skupovima kategoričkih podataka koje se mogu iskoristiti za definisanje onoga što bi se trebalo smatrati većom sličnošću i onoga što bi se trebalo smatrati manjom sličnošću. Takve mere sličnosti obično uzimaju u obzir raspodelu frekvencija različitih vrednosti atributa u datom skupu podataka da bi se definisala sličnost između

vrednosti dva kategorička atributa. Svaka od mera za definisanje sličnosti jedinstveno koristi informacije iz skupa podataka (Boriah i ostali, 2008). Pošto se radi o proceni mera sličnosti koje se izvode iz samih podataka, jasno je da je njihov učinak veoma vezan za sam skup podataka koji se analizira.

Proučavanje sličnosti između objekata sa kategoričkim varijablama ima dugu istoriju. Pirson (Pearson) je predložio hi-kvadrat statistiku krajem 1800.-tih što je često korišćeno za testiranje nezavisnosti između kategoričkih varijabli u tabeli kontigencije. Pirsonova hi-kvadrat statistika je kasnije modifikovana i proširena, što je dovelo do nekoliko drugih mera (Boriah i ostali, 2008). U novije vreme, međutim, mera preklapanja (overlap) je postala najčešće korišćena mera za sličnost kategoričkih podataka. Njena popularnost je možda pre svega posledica njene jednostavnosti i lakog korišćenja.

Kada je u pitanju merenje sličnosti kategoričkih atributa, pojedini autori su, baveći se klaster analizom, preporučivali da se vrednosti atributa svedu na binarne podatke, pa da se potom koriste binarne mere sličnosti. Sa druge strane, Wilson i Martinez (1997) su upravo na primeru zaključivanja na osnovu slučajeva ispitivali različite funkcije udaljenosti, za podatke kako sa kategoričkim, tako i sa kontinualnim vrednostima. Oni su mere u svojoj studiji zasnovani na nadgledanom pristupu gde svaki slučaj, pored skupa kategoričkih i kontinualnih vrednosti, ima i informaciju o pripadnosti određenoj klasi (Boriah i ostali, 2008).

U poslednjih nekoliko godina je predložen priličan broj novih tehnika otkrivanja zakonitosti u podacima i za kategoričke podatke. Neke od njih koriste pojmove sličnosti koji su bazirani na susedstvu ili uključuju izračunavanje sličnosti u algoritam učenja. Pristupi zasnovani na susedstvu koriste pojam sličnosti (obično meru preklapanja) da definišu susedstvo za slučaj. Postoje i tehnike koje ugrađuju mere sličnosti u algoritam, pri čemu se eksplicitno ne definiše opšta mera sličnosti za kategoričke atribute (Boriah i ostali, 2008). U ovom radu se koriste mere koje uključuju izračunavanje sličnosti u algoritam učenja, a koje direktno utvrđuju sličnost između parova podataka iz dva slučaja.

Kao što je ranije pomenuto, izračunavanje sličnosti između kategoričkih podataka u slučajevima nije jednostavno zbog činjenice da ne postoji eksplicitno shvatanje uređenosti kategoričkih vrednosti. Da bi se prevazišao ovaj problem, za kategoričke atribute se predlaže nekoliko mera sličnosti zasnovanih na podacima. Ponašanje takvih mera direktno zavisi od podataka.

Pošto se u ovom radu koriste mere sličnosti kategoričkih podataka koje su vođene samim podacima, sledi pregled ključnih karakteristika kategoričkog skupa podataka koje potencijalno mogu da utiču na ponašanje takvih mera sličnosti (Boriah i ostali, 2008):

- Veličina skupa podataka, *N*. Većina mera je tipično nepromenljiva od veličine skupa podataka, iako postoje neke mere (npr. Smirnov) koje koriste ovu informaciju.

- Broj atributa, *d*. Većina mera je nazavisna od ove osobine, jer se uglavnom radi normalizacija sličnosti preko broja atributa. Međutim, pojedini eksperimentalni rezultati (Boriah i ostali, 2008.) su pokazali da broj atributa utiče na učinak algoritama za otkrivanje izuzetaka (autlajera, anomalija).

- Broj vrednosti koje može imati svaki od atributa, n_k. Skup podataka može da sadrži atribute koji uzimaju nekolicinu vrednosti i atribute koji uzimaju svega nekoliko vrednosti. Na primer, jedan atribut može uzeti nekoliko stotina mogućih vrednosti, dok drugi atribut može da ima samo nekoliko vrednosti. Mera sličnosti može dati veći značaj drugom atributu, gotovo ignorišući prvi. Jedna od posmatranih mera u ovom radu (Eskin) se ponaša upravo na ovaj način.

- Distribucija $f_k(x)$ se odnosi na distribuciju frekvencija vrednosti koje uzima atribut u datom skupu podataka. U pojedinim skupovima podataka atribut može biti ravnomerno raspoređen po određenom skupu A_k , dok u drugim skupovima podataka distribucija može biti iskrivljena. Pojedine mere sličnosti mogu dati

veći značaj vrednostima atributa koje se javljaju retko, dok druge mere sličnosti mogu dati veći značaj čestim vrednostima atributa.

Pretpostavimo da kategorički skup podataka *D* sadrži *N* objekata, koji su definisani preko skupa *d* kategoričkih atributa gde A_k označava *k*-ti atribut. Neka atribut A_k uzima n_k mogućih vrednosti u datom skupu podataka, koji je označen sa A_k . Tada se može reći da su karakteristike skupa sledeće (Boriah i ostali, 2008):

- $f_k(x)$: ukazuje na to koliko puta atribut A_k uzima vrednost *x* u skupu podataka *D*. Ako $x \notin A_k$, onda je $f_k(x) = 0$

- $\hat{p}_k(x)$ označava uzoračku verovatnoću da atribut A_k uzima vrednost *x* u skupu podataka *D,* a izračunava se na sledeći način:

$$\hat{p}_k(x) = \frac{f_k(x)}{N} \qquad (5)$$

- $p_k^2(x)$ označava procenu verovatnoće da atribut A_k uzima vrednost *x* u skupu podataka, a izračunava se na sledeći način:

$$p_k^2(x) = \frac{f_k(x)(f_k(x)-1)}{N(N-1)} \qquad (6)$$

Statistika koja se tiče učestalosti poklapanja vrednosti između slučaja i podataka referentnog skupa ukazuje na to da će posmatrani slučaj biti sličniji slučaju iz baze na određenom atributu ako podudaranja vrednosti postoje na učestalim vrednostima atributa, dok poklapanja na retkim vrednostima atributa ne moraju značiti i veću sličnost. Ovo je naročito značajno u situacijama kada atribut u referentom setu podataka može uzimati izuzetno veliki broj podataka (Chandola i ostali, 2009).

Statistika koja uzima u obzir frekvenciju nepodudarajućih vrednosti atributa između posmatranog slučaja i slučaja iz referentnog skupa podataka govori o tome da ako na određenom atributu postoji neslaganje vrednosti atributa posmatranog slučaja i učestalih vrednosti slučaja iz referentnog skupa podataka, manja je sličnost posmatranih slučajeva (Chandola i ostali, 2009).

Statistika koja uzima u obzir broj mogućih vrednosti atributa se može smatrati funkcijom broja argumenata neusaglašenih atributa između posmatranog slučaja i podataka referentnog skupa. Konkretno, vrednost statistike je veća kada atributi koji se ne podudaraju uzimaju manji broj vrednosti. Ideja je da ako postoje neslaganja posmatranog slučaja i referentnog slučaja iz baze na atributima koji uzimaju vrlo malo vrednosti posmatrajući sve slučajeve iz baze, onda je malo verovatno da posmatrani slučaj pripada istoj klasi, odnosno manje je sličan referentnom slučaju iz baze - jednostavno zato što postoji samo nekoliko mogućnosti da se vrednosti ne podudaraju (Chandola i ostali, 2009).

Za konverziju mera udaljenosti u mere sličnosti, Boriah i ostali (2008.) predlažu sledeću formulu:

$$sličnost = \frac{1}{1 + udaljenost} \qquad (7)$$

Skoro sve mere sličnosti polaze od toga da se vrednost sličnosti između dve instance podataka X i Y, koje pripadaju skupu podataka D, izračunava na sledeći način (Boriah i ostali, 2008):

$$S(X,Y) = \sum w_k S_k(X_k, Y_k) \qquad (8)$$

gde je $S_k(X_k, Y_k)$ sličnost između dve vrednosti po svakom kategoričkom atributu A_k, pri čemu $X_k, Y_k \in$ A$_k$., dok w_k ukazuje na težinu, odnosno značaj dodeljen atributu A_k.

Mere sličnosti između kategoričkih atributa se mogu klasifikovati na nekoliko načina, a podela se uglavnom svodi na sledeće grupe mera (Boriah i ostali, 2008):

- mere koje svakom neslaganju vrednosti atributa dodeljuju sličnost 0, a poklapanjima eventualno dodeljuju različite vrednosti. Primeri ovih mera su sledeće: Overlap, Goodall, Goodall1, Goodall2, Goodall3, Goodall4, Gambaryan,

- mere koje svakom poklapanju daju maksimalnu vrednost sličnost, tj. 1 i eventualno daju različite vrednosti za nepodudaranja vrednosti atributa. Primeri ovih mera su sledeće: Eskin, Inverse Occurrence Frequency – IOF, Occurrence Frequency – OF, Burnaby.

- mere koje daju različite vrednosti i poklapanjima i nepoklapanjima vrednosti atributa. Primeri ovih mera su sledeće: Lin, Lin1, Smirnov, Anderberg.

Mere sličnosti za kategoričke varijable se mogu klasifikovati i na osnovu argumenata koji se koriste u predloženim merama (Boriah i ostali, 2008):

1. Probabilistički pristupi – uzimaju u obzir verovatnoću dešavanja određenog poklapanja. Sledeće mere su mere verovatnoće: Goodall, Smirnov, Anderberg.

2. Informaciono-teorijski pristupi - uključuju sadržaj informacija o određenoj vrednosti/promenljivoj u odnosu na skup podataka. Sledeće mere su informaciono-teorijske: Lin, Lin1, Burnaby.

Za izračunavanje sličnosti između kategoričkih podataka, u ovoj disertaciji će se koristiti sledeće mere udaljenosti između dva slučaja: Overlap, Goodall1, Eskin i OF. Ove statistike su korišćene jer se iste smatraju standardima i druge mere za utvrđivanje

sličnosti (Boriah i ostali, 2008) mogu se smatrati izvedenim funkcijama jedne ili više predloženih statistika (Chandola i ostali, 2009).

U nastavku će se ukratko predstaviti osnovno o merama sličnosti korišćenim u ovom radu, a za više informacija o svim pomenutim merama čitalac se upućuje na autore Boriah i ostali (2008).

2.4.3.1. Mera podudaranja (Overlap)

Mera preklapanja ili podudaranja (overlap) jednostavno prebrojava broj kategoričkih atributa koji se podudaraju posmatrajući dva slučaja, ponderisući ih pritom željenim težinama. Obim međuatributske sličnosti, kada se koristi mera preklapanja je u granicama [0, 1], pri čemu se vrednost sličnosti jednaka 0 javlja kada ni po jednom atributu ne postoji podudaranje vrednosti, dok se vrednost 1 javlja kada se vrednosti po svakom posmatranom kategoričkom atributu podudaraju (Boriah i ostali, 2008).

Matematički, mera podudaranja se može izraziti na sledeći način (Boriah i ostali, 2008):

$$S_k(X_k, Y_k) = \begin{cases} 1, \text{ ako je } X_k = Y_k \\ \\ 0, \text{ ako je } X_k \neq Y_k \end{cases} \qquad (9)$$

2.4.3.2. Goodall 1

Goodall je 1966. predložio meru koja pokušava da normalizuje sličnost između dva objekta preko verovatnoće da posmatrana vrednost sličnosti može da se uoči na slučajnom uzorku od dve tačke (Boriah i ostali, 2008). Ova mera dodeljuje veću sličnost poklapanjima, ako je vrednost retka pre nego ako je vrednost atributa učestala. Goodall-ova originalna mera ukazuje na postupak kombinovanja sličnosti u multivarijaciono okruženje koje uzima u obzir zavisnosti između atributa. Pošto je ovaj postupak

računski skup, koristi se jednostavnija verzija mere, nazvana Goodall1 (Boriah i ostali, 2008).

Mera Goodall1 je ista kao mera Goodall na međuatributskoj osnovi. Međutim, umesto da kombinuje sličnosti uzimajući u obzir zavisnosti između atributa, Goodall1 mera uzima u obzir prosečnu međuatributsku sličnost između dva objekta. Opseg vrednosti sličnosti $S_k(X_k, Y_k)$ za podudarajuće vrednosti, korišćenjem mere Goodall1 je $\left[0{,}1 - \frac{2}{N(N-1)}\right]$, sa minimumom koji se postiže kada atribut A_k uzima samo jednu vrednost, dok se maksimalna vrednost postiže kada se vrednost X_k javlja dva puta, dok se sve ostale moguće vrednosti atributa A_k javljaju više od 2 puta (Boriah i ostali, 2008).

Matematički, mera Goodall1 se može izraziti na sledeći način (Boriah i ostali, 2008):

$$S_k(X_k, Y_k) = \begin{cases} 1 - \sum_{q \in Q} p_k^2(q), \text{ ako je } X_k = Y_k \\ 0, \text{ ako je } X_k \neq Y_k \end{cases} \qquad (10)$$

Goodall1 (poput mera kao što su: Lin, Lin1, Goodall3, Smirnov, Anderberg) je mera kojom se veća sličnost dodeljuje podudaranjima kada je vrednost atributa retka (f_k je malo), za razliku od mera kao što su Goodall2 i Goodall4, koje veću sličnost dodeljuju podudaranjima kada je posmatrana vrednost atributa česta (f_k je veliko).

2.4.3.3. Eskin

Eskin i ostali su 2002. predložili normalizaciju jezgra podataka koji su bazirani na slogovima, a služe za otkrivanje upada u mrežu (Boriah i ostali, 2008). Originalna mera je zasnovana na udaljenosti i dodeljuje vrednost $\frac{2}{n_k^2}$ za neslaganja; kada se prilagodi za

merenje sličnosti, postaje $\frac{n_k^2}{n_k^2+2}$. Ova mera daje veću vrednost sličnosti nepoklapanjima koja se javljaju kod atributa koji uzimaju veliki broj mogućih vrednosti. Opseg vrednosti sličnosti $S_k(X_k,Y_k)$ za nejednake vrednosti, korišćenjem mere Eskin je $\left[\frac{2}{3},\frac{N^2}{N^2+2}\right]$, pri čemu se minimalna vrednost postiže kada atribut A_k uzima samo dve vrednosti, dok se maksimalna vrednost postiže kada atribut ima sve jedinstvene vrednosti (Boriah i ostali, 2008).

Matematički, mera Eskin se može izraziti na sledeći način (Boriah i ostali, 2008):

$$S_k(X_k,Y_k) = \begin{cases} 1, \text{ ako je } X_k = Y_k \\ \\ \frac{n_k^2}{n_k^2+2}, \text{ ako je } X_k \neq Y_k \end{cases} \qquad (11)$$

2.4.3.4. OF

Frekvencija pojavljivanja (Occurrence Frequency - OF) podrazumeva takav način merenja sličnosti da se nepoklapanjima na manje frekventnim vrednostima dodeljuje manja sličnost, dok se neslaganjima na češćim vrednostima atributa dodeljuje veća sličnost. Opseg vrednosti sličnosti $S_k(X_k,Y_k)$ za nejednake vrednosti, korišćenjem mere OF je $\left[\frac{1}{(1+(\log N)^2)},\frac{1}{(1+(\log 2)^2)}\right]$, pri čemu se minimalna vrednost postiže kada se vrednosti atributa X_k i Y_k pojavljuju samo jednom u skupu podataka, dok se maksimalna vrednost postiže kada se vrednosti atributa X_k i Y_k pojavljuju $\frac{N}{2}$ puta (Boriah i ostali, 2008).

Matematički, mera OF se može izraziti na sledeći način (Boriah i ostali, 2008):

$$S_k(X_k, Y_k) = \begin{cases} 1, \text{ ako je } X_k = Y_k \\ \\ \dfrac{1}{1 + \log \dfrac{N}{f_k(X_k)} \, xog \, \dfrac{N}{f_k(Y_k)}}, \text{ ako je } X_k \neq Y_k \end{cases} \qquad (12)$$

Mera OF (kao i mera Anderberg) dodeljuje veću sličnost za nepodudaranja čestih vrednosti, za razliku od mera kao što su IOF, Lin1, Smirnov i Burnaby, koje dodeljuju veću sličnost kada dođe do nepodudaranja između retkih vrednosti (Boriah i ostali, 2008).

2.5. Genetski algoritmi

Poslednjih nekoliko decenija razvijaju se i neki opšti heuristički pristupi- metaheuristike koji su se pokazali vrlo efikasni u praksi. Najčešće korišćene metaheuristike su: tabu pretraživanje (tabu search), simulirano kaljenje (simulated annealing), Lagranžova relaksacija (Lagrangean relaxation), genetski algoritmi (genetic algorithms), metoda promenljivih okolina (variable neighborhood search), neuronske mreže (neural networks), mravlji sistemi (ant systems) i druge. Ove metode uglavnom daju dobra rešenja, neretko se dobijaju optimalna rešenja, iako se u većini slučajeva optimalnost ne može dokazati. Čak i u slučajevima kada problem ima više lokalnih ekstrema, ove heuristike nam često pronalaze globalni optimum. Moguće je i njihovo međusobno kombinovanje u cilju korišćenja dobrih strana svake od njih, kao i hibridizacija sa egzaktnim metodama čime se povećava efikasnost pri nalaženju optimalnog rešenja.

GA pristup je metod optimizacije koji podrazumeva stohastičku tehniku pretrage, koja ima sposobnost da pretražuje velike i komplikovane prostore. Ona poboljšava rezultate pretrage stalno pokušavajući različita moguća rešenja sa genetskim operacijama (Ahn i ostali, 2007) sličnim kao kod prirodne selekcije i principa evolucije (Ahn & Kim, 2008). GA u osnovi istražuje složeni prostor na adaptivan način preko operatora GA, kao što su: selekcija, ukrštanje i mutacija. Ovaj algoritam koristi prirodnu selekciju - opstanak najbolje prilagođenih jedinki, u cilju rešavanja problema optimizacije (Kim, 2004). Pristup GA je posebno pogodan za probleme više parametarske optimizacije sa funkcijom cilja koja se odnosi na razna tvrda i meka ograničenja (Shin & Han, 1999).
GA se dosta razlikuju od mnogih konvencionalnih algoritama pretrage i na sledeće načine: GA razmatraju ne samo jednu tačku već mnoge tačke u prostoru pretrage istovremeno, smanjujući mogućnost za približavanje lokalnim optimumima; GA pretražuju direktno sa nizovima karaktera koji predstavljaju skup parametara, a ne samo sa pojedinačnim parametrima; i GA koriste pravila verovatnoće, a ne deterministička pravila za obavljanje potrage (Min i ostali, 2006).

Prve ideje o genetskim algoritmima izložene su u radu J. Holland-a 1975. godine. Javile su se u okviru tzv. teorije adaptivnih sistema, koja proučava modele efikasnog

adaptivnog ponašanja nekih bioloških, specijalno genetskih, sistema. Iako su i ranije postojali radovi sa sličnim idejama, Holland se smatra tvorcem ove metaheuristike i postavke iz njegovih najranijih radova još uvek važe.
Genetski algoritmi su prvobitno kreirani da simuliraju proces genetske evolucije jedne populacije jedinki pod dejstvom okruženja i genetskih operatora, dok se danas koriste za rešavanje široke klase problema kombinatorne optimizacije.

Genetski algoritam se primenjuje na konačnom skupu jedinki koji se naziva populacija. Svaka jedinka u populaciji je predstavljena nizom karaktera (genetskim kodom) i odgovara nekom rešenju u pretraživačkom prostoru. One jedinke iz populacije koje su u većoj meri prilagođene okruženju, međusobno se dalje reprodukuju i tako stvara nova generacija jedinki. Ovaj proces se ponavlja pri čemu se iz generacije u generaciju prosečna prilagođenost članova populacije povećava. Nakon nekog broja generacija čitav postupak se zaustavlja do zadovoljenja jednog ili više kriterijuma zaustavljanja.
Najbolji član trenutne populacije predstavlja rešenje genetskog algoritma.
Svaka jedinka u populaciji je predstavljena genetskim kodom nad određenom konačnom azbukom. Najčešće se koristi binarno kodiranje, gde se genetski kod sastoji od niza bitova. Binarno kodiranje je najpogodnije za implementaciju zbog svoje jednostavnosti, a nekada je pogodno koristiti azbuke veće kardinalnosti. Kodiranje rešenja je bitan korak genetskog algoritma jer se neadekvatnim izborom koda može doći do loših rezultata bez obzira na ostalu strukturu genetskog algoritma.

Početna populacija se obično generiše na slučajan način, što obezbeđuje raznovrsnost genetskog materijala. Moguće je korišćenje neke heuristike za generisanje početne populacije, ili jednog njenog dela, uz uslov da se ta heuristika relativno brzo izvršava i da značajno ne smanjuje raznovrsnost genetskog materijala. Svakoj jedinki u populaciji (u praksi ih je najviše do nekoliko stotina) se na određen način dodeljuje funkcija prilagođenosti ili funkcija cilja (fitness function) koja je merilo kvaliteta jedinke, odnosno odgovarajućeg rešenja. Standardni pristup konceptu GA smatra da je cilj algoritma da se iz generacije u generaciju poboljšava prilagođenost svake jedinke u populaciji, kao i srednja prilagođenost cele populacije uzastopnom primenom genetskih operatora: selekcije, ukrštanja i mutacije (Stanimirović, 2004).

Genetski operator selekcije vrši izbor jedinki iz populacije koje učestvuju u stvaranju nove generacije. Selekcija se primenjuje u skladu sa vrednostima funkcije prilagođenosti. Standardni pristup smatra da će bolje prilagođene jedinke preneti dobar genetski materijal na svoje potomke, pa smanjuje šansu prolaska loših jedinki u narednu generaciju, tako da one postepeno nestaju iz populacije (Stanimirović, 2004).

Operator ukrštanja predstavlja postupak razmene delova genetskog koda dve (ili više) jedinke-roditelja, tako da se dobijaju kodovi novih (jedne ili više) jedinki-potomaka. Razmena genetskog materijala daje mogućnost da dobro prilagođene jedinke generišu još bolje potomke, ali i da neki dobri geni relativno loših jedinki dobiju svoju šansu za dalju reprodukciju (Stanimirović, 2004).

Mutacijom se vrši promena koda jedinke zamenom pojedinih simbola koda nekim drugim simbolima azbuke kodiranja. Operator mutacije se koristi da bi se unela raznovrsnost među jedinkama populacije jer one vremenom mogu postati jako slične. Mutacija takođe sprečava gubitak dela genetskog materijala do kojeg može doći višestrukom primenom operatora selekcije i ukrštanja. Svaki gen genetskog koda može mutirati sa datom malom verovatnoćom (Stanimirović, 2004).

Operatori genetskog algoritma se uzastopno primenjuju do zadovoljenja nekog od kriterijuma zaustavljanja: maksimalan broj generacija, dostignut optimum, nepromenjen kvalitet rešenja posle unapred zadatog broja generacija itd (Stanimirović, 2004).

Shematski zapis osnovnih elemenata GA (Stanimirović, 2004) može se predstaviti na sledeći način:

```
Unošenje_Ulaznih_Podataka();
Generisanje_ Početne _Populacije();
while (! Kriterijum_Zaustavljanja_GA() )
{
for(i=0;i< N_pop ; i++) p_i = Vrednosna_Funkcija();
Funkcija_Prilagođenosti();
Selekcija();
Ukrštanje();
```

```
Mutacija();
}
Štampanje_Izlaznih_Podataka();
```

Uopšteno govoreći, proces GA se odvija na sledeći način (Ahn i ostali, 2007):
Prvo, GA nasumično generiše skup rešenja koji predstavlja početnu populaciju. Svako rešenje u populaciji se zove hromozom i obično se daje u obliku binarnog stringa. Nakon generisanja početne populacije, GA izračunava funkciju prilagođenosti za svaki hromozom. Funkcija prilagođenosti je korisnički definisana funkcija koja vraća rezultate ocene svakog hromozoma, tako da veće vrednosti prilagođenosti znače da je bolji hromozom. Tačnost klasifikacije (učinka) se obično koristi kao funkcija prilagođenosti za probleme klasifikacije.

Nakon generisanja početne populacije, primenom genetskih operatora se generišu potomci. Između različitih genetskih operatora, može se reći da su selekcija, ukrštanje i mutacija najosnovniji i najpopularniji operatori. Operator selekcije određuje koji hromozom će preživeti. Ukrštanjem se potomci iz parova hromozoma razmenjuju i tako stvaraju nove parove hromozoma. Mutacijom, gde je stopa mutacije obično mala, proizvoljno izabranih bitovi u hromozomu se obrću. Ovi koraci evolucije se nastavljaju sve dok se zadovolje uslovi za zaustavljanje. U većini slučajeva, kriterijum zaustavljanja je postavljen na maksimalnom broju generacija (Chiu, 2002; Fu & Shen, 2004; Han & Kamber, 2001).

Pristup GA se u ovoj studiji koristi za utvrđivanje odgovarajućih parametara funkcija preferencija, kao i da se pronađu težine atributa. Tačnost klasifikacije modela je postavljena kao funkcija prilagođenosti GA.

Implementacija GA podrazumeva korišćenje raznih parametara: veličina početne populacije, nivo mutacije, nivo ukrštanja, itd. Nedostatak genetskog algoritma je u tome što ne postoji jedinstvena kombinacija parametara koja je najbolja za sve probleme, ili bar za različite instance istog problema, već ih u svakom konkretnom slučaju moramo podesiti eksperimentalnim putem.

Evolver je napredni, ali istovremeno i jednostavan za korišćenje alat za optimizaciju koji je dodatak Microsoft Excel-u. Evolver koristi inovativni genetski algoritam (GA). Može pomoći u rešavanju problema u oblasti finansija, distribucije, planiranja, raspodele sredstava, proizvodnje, budžetiranja, inženjeringa, korišćenja energije, maloprodaje, itd. Evolver se može koristiti za rešavanje mnogih složenih nelinearnih problema, ukoliko ih je moguće modelovati u Excelu-u. Evolver-om se može doći do sveobuhvatno najboljeg "globalnog" rešenja problema – rešenja do koga bi tradicionalnim metoda bilo veoma teško doći.

Standardni programi za optimizaciju, poput Excel-ovog Solvera, su dobri u pronalaženju najboljeg „lokalnog" rešenja, ili za pronalaženje kombinacije vrednosti kojima se postiže maksimalna ili minimalna vrednost ishoda modela koji je predstavljan jednostavnom tabelom sa određenim ograničenjima. Ovi modeli pronalaze neko rešenje koje izgleda da daje povoljne rezultate i nastavljaju da rade na toj osnovi, bez pokušaja generisanja novih rešenja. Ovo je poznato kao „penjanje uz brdo". Metode „planinarenja" će početi sa početnom pretpostavkom i odatle onda nastaviti da sa nalaženjem najbližeg optimalnog rešenja (maksimuma ili minimuma). Može se početi u nekoj tački duž krive i kretati levo ili desno dok se ne dostigne vrhunac ili korito. Planinarenje će uvek naći najbolji odgovor ako je (a) funkcija koja se istražuje glatka, i (b) početne vrednosti promenljivih su blizu optimalnog rešenja. Ako neki od ovih uslova nije ispunjen, metodama planinarenja će se pre doći do lokalnog, nego do globalnog rešenja. Stoga ovi programi nisu podešeni za rešavanje komplikovanijih, nelinearnih problema, gde najbolje lokalno rešenje ne mora biti globalno najbolje rešenje.
Evolver ne radi na ovim osnovama, već koristi genetske algoritme, stohastički usmerene tehnike pretrage. Koristeći inovativne „mutacije" i kombinacije rešenja, ili „jedinke", veoma je pogodan za pronalaženje najboljeg globalnog rešenja pretražujući celokupan prostor mogućih rešenja. Iz tog razloga je pogodan i za složene i nelinearne probleme.

Pre korišćenja Evolver-a mora se napraviti model postojećeg problema koristeći Microsoft Excel. Model u Excelu može podrazumevati primenu bilo koje Excel funkcije, a čak i pozivanje VBA macroa.

Model treba da zadovolji tri osnovna kriterijuma:

1. da sadrži varijable koje u određenoj interakciji proizovde krajnji rezultat,
2. da se preračun radi ispravno za sve moguće kombinacije varijabli,
3. da je rezultat tačan prikaz koliko je neko rešenje zaista dobro.

Dok god su ova tri kriterijuma ispunjena, Evolver može da traži kombinacije vrednosti varijabli koje daju najbolji odgovor, bilo da su to najbolje vrednosti, odgovarajući redosled, ili optimalno grupisanje.

Kako Evolver radi?

Evolver rešava probleme optimizacije koristeći genetske algoritme (GA). Kod GA, svaka jedinka ili moguće rešenje datog problema postaje „nezavistan“ organizam koji može da se „ukršta“ da drugim organizmima. Model u radnom listu Excela predstavlja okruženje za organizme, određujući koji od tih organizama su dovoljno „prilagođeni“ da bi preživali, a u zavisnosti od rezultata koji daju.
Ukratko, proces podrazumava:
1. Generisanje na slučajan način većeg broja organizama (mogućih rešenja), i izračunavanje rezultata koji proizvodi svaki od njih. Celokupna ova „populacija“ organizama se potom rangira od najboljeg do najgoreg.
2. Izabrati dobre organizme i razmeniti njihove varijable (gene) koristeći ukrštanje i mutaciju, a da bi se proizveli „potomci“. Ako potomak ne daje dobar rezultat, biraju se još dva roditelja.
3. Ako je organizam potomka dobar, on se ubacuje u populaciju.

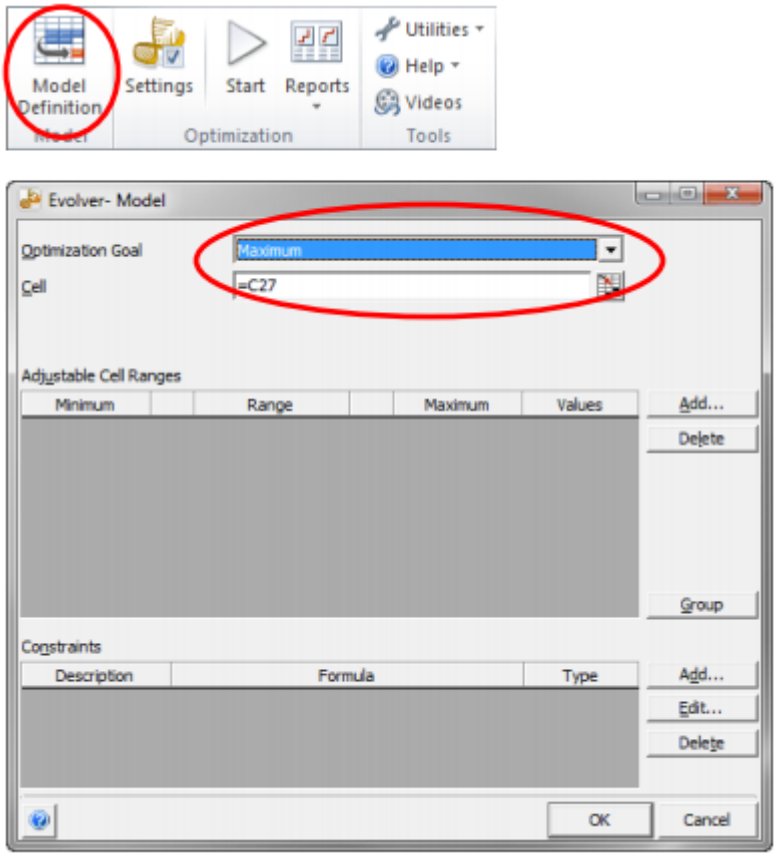
Pošto Evolver ponavlja korake 2 i 3, populacija se razvija, odnovno „evoluira“ povećavajući optimalna rešanja.

Korišćenje Evolver-a podrazumeva tri jednostavna koraka:

1. Podešavanje modela.

Prozor Evolver-Model, prikazan na slici 4. i slici 5., obezbeđuje jednokratno podešavanje za problem optimizacije. U ovom prozoru se i identifikuju opsezi ćelija u kojima su vrednosti koje treba podesiti i definišu se ograničenja. Ovde se navodi ćelija u kojoj je smeštena funkcija cilja, pri čemu se navodi da li se želi njena minimizacija, maksimizacija ili dostizanje ciljne vrednosti.

Zadatak definisanja funkcije prilagođenosti je uvek veoma značajan. U slučaju izgradnje ZOS sistema, cilj je da se pronađu najrelevantniji slučajevi iz baze koji mogu da dovedu do ispravnog rešenja za trenutni problem.
Sposobnost ZOS sistema za postizanje ovog cilja se može predstaviti preko funkcije prilagođenosti koja određuje koliko dobro funkcija podudaranja povećava tačnost klasifikacije. U ovom radu se kao funkcija prilagođenosti primenjuje stopa tačnosti klasifikacije testnog skupa slučajeva. Testni skup slučajeva se sastoji od slučajeva čija su rešenja poznata, ali za koje se određuje rezultat klasifikacije kako bi se procenila podobnost različitih metoda pronalaženja i vrednosti atributa. Cilj optimizacije GA je da se pronađe vektor najboljih pondera i najbolja metoda koja će dovesti do najveće stope tačnosti klasifikacije.



Slika 4. Podešavanje modela - Prozor Evolver-Model

Definisanje opsega i uslova zaustavljanja

U svakom modelu optimizacije mora postojati bar jedna, a najčešće je više, „podesiva“ ćelija kojoj se traži vrednost koja će voditi ga optimizaciji funkcije cilja. Stoga, podesive ćelije moraju biti povezane sa, direktno ili indirektno, preko Excel formula, sa funkcijom cilja, odnosno ćelijom čiji optimalnu vrednost želimo. U samom modelu, prilikom definisanja ćelija u kojima treba podesiti vrednosti, mogu da se odrede maksimalne i minimalne granične vrednosti ćelija određenog opsega, što u velikoj meri pojednostavljuje podešavanje i pravljenje promena.

Nekada je potrebno definisati i ograničenja u modelu (na primer, mogu biti ograničena sredstva ili model mora da zadovolji određene uslove kada su neke varijable u pitanju). Prilikom definisanja ograničenja (koja mogu biti teška ili meka), takođe se mogu odrediti minimalne i maksimalne vrednosti ćelija izabranog opsega.

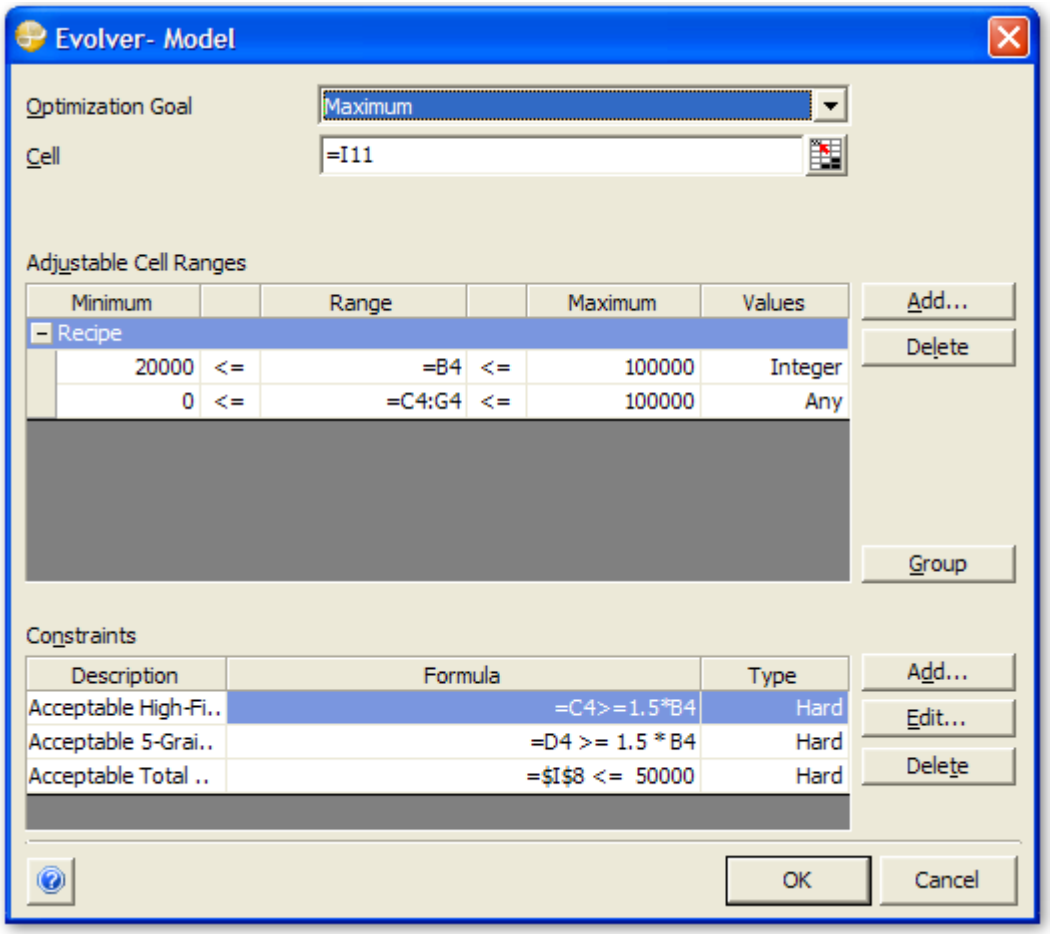
Na kraju, potrebno je da se podese uslovi zaustavljanja optimizacije.

Genetski algoritmi su u osnovi stohastičke metode pretrage dopustivog prostora rešenja, tako da mogu raditi beskonačno dugo, ukoliko im ne nametnemo kriterijum zaustavljanja. Postoji nekoliko kriterijuma završetka GA: maksimalni broj generacija, sličnost jedinki u populaciji, najbolja jedinka je ponovljena maksimalni broj puta, algoritam je dostigao optimalno rešenje (ukoliko je ono unapred poznato), dokazana optimalnost najbolje jedinke (ukoliko je to moguće), ograničeno vreme izvršavanja GA, prekid od strane korisnika itd. Svaki od navedenih kriterijuma ima dobre i loše aspekte, tako da se u praksi najbolje pokazalo njihovo kombinovanje, jer se tako smanjuje mogućnost loše procene prekida GA.

Ispunjenje definisanog kriterijuma zaustavljanja je u suštini signal Evolveru da treba da završi optimizaciju.

Metode rešavanja

Evolver koristi šest različitih metoda rešavanja koje se mogu odabrati, a radi pronalaženja optimalne kombinacije vrednosti u ćelijama koje su namenjene podešavanju.



Slika 5. Evolver Model prozor – podešavanje modela.

Šest mogućih metoda su:

Recept – podrazumeva skup varijabli koje mogu nezavisno da se menjaju. Kao kod sastojaka za kulinarski recept- potrebno je pronaći optimalne vrednosti svih sastojaka da bi se dobio najbolji miks.

Grupisanje – podrazumeva kolekcije elemenata koje treba rasporediti u grupe. Ovo se na primer koristi za probleme tipa raspoređivanja resursa u određene grupe, npr. radnike treba rasporediti u grupe radi obavljanja različitih poslova.

Poredak - spisak elemenata sa određenim poretkom. Ovaj metod se koristi, na primer, u slučaju da je potrebno odrediti optimalni redosled izvršenja nekoliko nezavisnih zadataka.

Budžet - recept algoritam, ali se total vodi kao konstanta, ovaj metod se koristi kada zbir vrednosti u podešavajućim ćelijama treba da ostane konstanta.

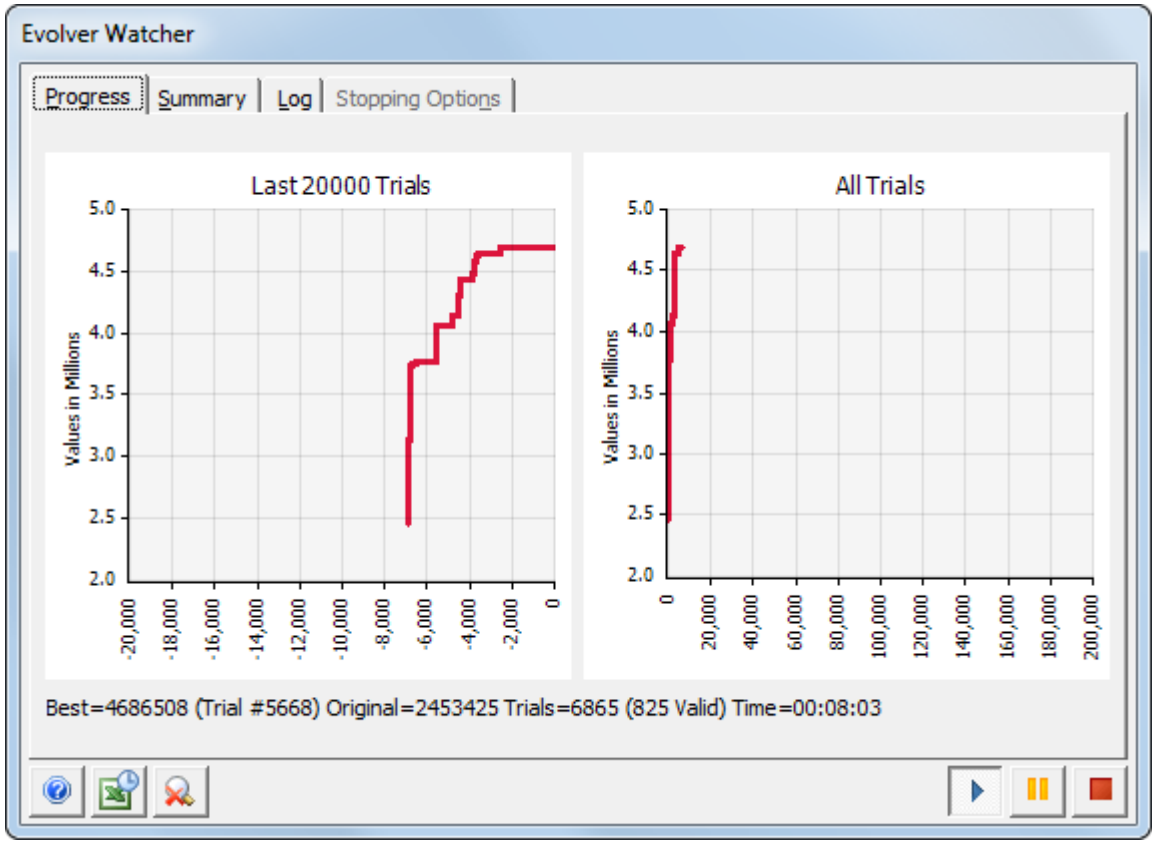
Projekat – algoritam poretka, ali je definisano da neki elementi prethode drugima.

Raspored – algoritam grupisanja, ali dodeljuje elemente blokova vremena dok zadovoljava ograničenja.

Evolver takođe omogućava veliki stepen kontrole nad tim kako se sama optimizacija obavlja. Tako se mogu podesiti parametri za optimizaciju, izvršiti podešavanja, kontrolisati upotreba makroa i još više u Evolver Settings dijalogu.

2. Pokretanje optimizacije.

Optimizacija se započinje klikom na Start ikonicu. Evolver će početi generisanje probnih rešenja u nastojanju da se postigne cilj postavljen u koraku 1. Rezime se pojavljuje u prozoru Evolver Progress, pokazujući status optimizacije i najbolje postignuto rešenje do tog trenutka. Ovaj prozor omogućava da se načini pauza, da se zaustavi, i pokrene optimizacija pomoću odgovarajućih kontrola. Sve pojedinosti napretka se mogu videti preko Evolver Watcher-a, čiji je primer prikazan na slici 6. Izveštaji na tabovima pokazuju ažuriranja najbolje postignutih rešenja, sva pokušana rešenja, raznovrsnost rešenja koja su probana, i još mnogo toga.



Slika 6. Evolver Watcher.

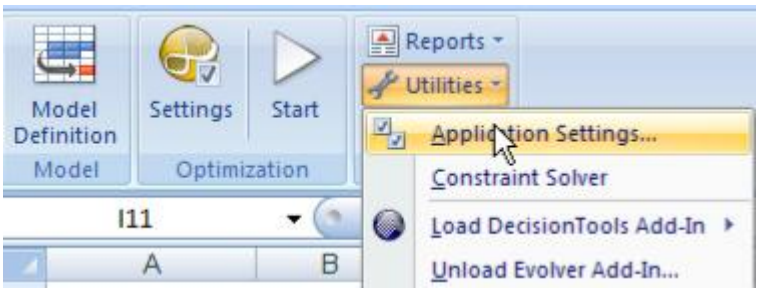
Šta radi optimizacija?

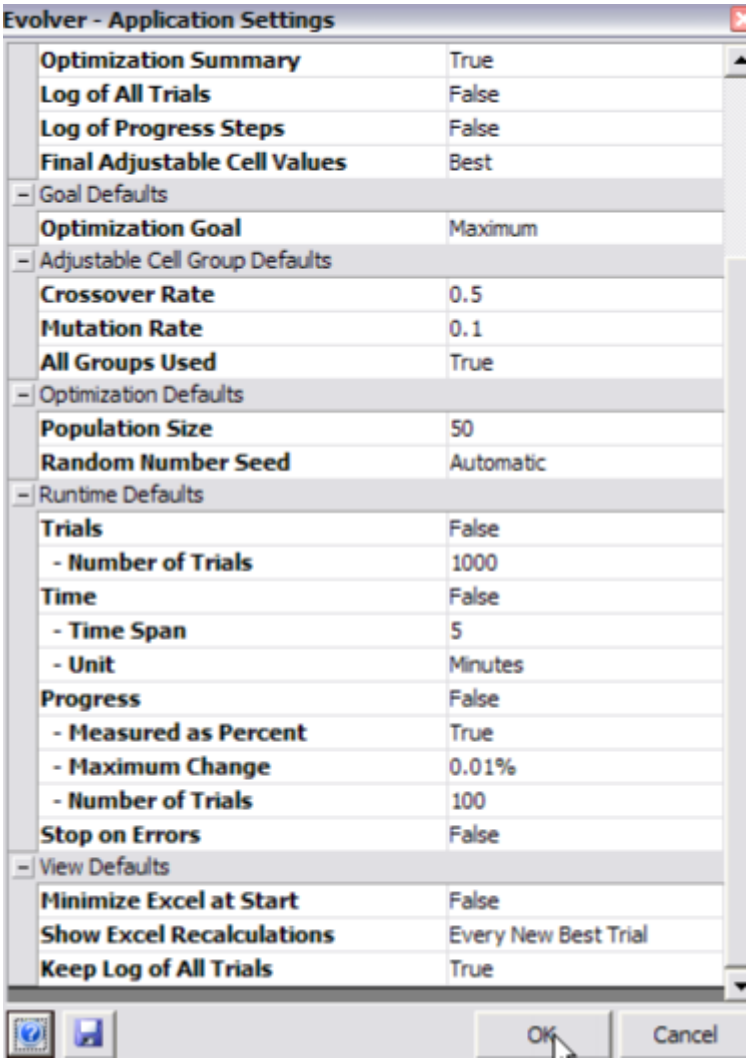
Tokom optimizacije, Evolver generiše veliki broj probnih rešenja i koristi genetske algoritme da stalno poboljšava rezultate svakog pokušaja. Sa genetskim algoritmima, svako moguće rešenje postaje nezavisna "jedinka" koja može da "množi" sa drugim jedinkama. Model u tabeli Excel-a se ponaša kao okruženje jedinki, određujući koje su dovoljno "prilagođene" da prežive, a na osnovu njihovih rezultata, pri čemu se povremeno pokušavaju "mutacije", ili pak potpuno nova rešenja.

3. Pogledati rezultate optimizacije.

Nakon optimizacije, Evolver može da prikaže rezultate originalnog, najboljeg, i poslednjeg rešenja za ceo model. Ovo čini da je se lako odlučiti za najbolji nastavak akcije. Takođe se mogu generisati izveštaji direktno u Excel-u za rezime optimizacije, evidenciju svih simulacija, i evidenciju svi koraka gde je postojao određeni napredak.

Dijalog ”Podešavanja” omogućava podešavanja aplikacije, odnosno sadrži prilagodljiva podešavanja kao što su izveštaji, kriterijumi zaustavljanja, podešavanje cilja, trajanja optimizacije, i još mnogo više. Podešene vrednosti se primenjuju potom na sve modele do eventualne potrebe za nekim izmenama. Vrednosti se menjaju na opciji ”Alati (Utilities)”, što je ilustrovano u nastavku, slika 7.





Slika 7. Alati/ Utilities – podešavanje aplikacije

3. Projektovanje modela

Ko uči na tuđim greškama, štedi vlastitu školarinu.

Nemačka poslovica

Ne postoji tako moćan učitelj kao što je iskustvo.

Anonimni autor

Znanje koje imamo je samo mrvica onoga što nemamo.

Platon

Tema trećeg poglavlja ove disertacije je Projektovanje modela. U ovom delu rada se daje prikaz razvoja predloženih modela za klasifikaciju i to Modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na zaključivanju na osnovu slučajeva uz primenu tradicionalnih mera sličnosti (bazni model), Modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na ZOS-u, ali uključuje i domensko znanje, izraženo preko funkcija preferencija, kao i Modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na ZOS-u, uključuje domensko znanje, izraženo preko funkcija preferencija, ali i odabrane mere sličnosti za kategoričke atribute.

3.1. Razvoj modela za klasifikaciju

U disertaciji će biti predloženi modeli razvijeni uz korišćenje programa Microsoft Excel 2003/ 2010 i Palisade Software's Evolver Version 5.5. (Palisade Software, www.palisade.com). Algoritam je opšti i može biti korišćen za bilo koji skup podataka, nezavisno od broja slučajeva u bazi i broja i tipova indeksa, tj. atributa koji opisuju slučajeve.

Proces izgradnje modela klasifikacije sastoji se iz nekoliko koraka (Vukovic i ostali, 2012), što će u nastavku biti detaljnije objašnjeno za svaki od modela.

Predloženi izvedeni model, skraćenog naziva CBR-PF-GA, predstavlja kombinaciju ZOS-a i funkcija teorije preferencije uz upotrebu GA. Model u kome su sve

karakteristike jednake važnosti (jednako ponderisane) nazvan je čist (tradicionalni) model, dok je model sa karakteristikama, atributima različitih vrednosti težina, tj. značaja nazvan ponderisani. Za numeričke atribute, svi predloženi modeli koriste funkciju preferencije tipa 5. Za kategoričke atribute, model CBR-PF-GA koristi funkciju preklapanja (overlap), dok modeli CBR-PF-GA-GOODALL 1, CBR-PF-GA-OF i CBR-PF-GA-ESKIN koriste različite mere za određivanje sličnosti kod kategoričkih podataka – Goodall1, OF, Eskin.

Za pronalaženje slučajeva, koristi se k-NN metod. Svi predloženi modeli su testirani na nekoliko k vrednosti, sa variranjem od 1 do 9 na svakoj neparnoj vrednosti. NN model se koristi za merenje sličnosti između slučajeva. Za svaki slučaj iz skupa slučajeva za testiranje, model treba da dodeli jednu od dve moguće klase, i to merenjem udaljenosti tog slučaja u odnosu na svaki slučaj iz skupa namenjenog za treniranje. Za procenu učinka prediktivnog modela, koristi se metodologija 10-ostruke unakrsne validacije (engl. 10-fold cross validation). Particije (fold) unakrsne validacije se generišu korišćenjem takozvanog semena za slučajni izbor podataka.

3.2. Model za ocenu kreditne sposobnosti klijenata koji je zasnovan na zaključivanju na osnovu slučajeva uz primenu tradicionalnih mera sličnosti (bazni model)

Bazni modeli od kojih se krenulo u postupku istraživanja i čiji rezultati se porede sa kasnije razvijanim modelima je model za ocenu kreditne sposobnosti klijenata koji je zasnovan na zaključivanju na osnovu slučajeva uz primenu tradicionalnih mera sličnosti.

Pod tradicionalnim merama sličnosti se podrazumava primena Euklidove norme za merenje sličnosti numeričkih podataka, dok se za merenje sličnosti kategoričkih indeksa koristi funkcija preklapanja (overlap funkcija).

U postupku istraživanja razmatrani su rezultati baznog modela kod koga svi atributi imaju podjednak značaj, kao i rezultati modela koji uključuje ponderisane vrednosti atributa.

Varijable od značaja za sprovođenje baznog modela ilustrovane su u tabeli 4.

Tabela 4. Varijable od značaja za sprovođenje baznog modela

Broj slučajeva	Korisnik treba da unese ili sistem može da prepozna na osnovu broja redova u bazi
Broj atributa	Korisnik treba da unese ili sistem može da prepozna na osnovu broja kolona u bazi
Broj setova za unakrsnu validaciju	Korisnik treba da unese željeni broj partija za unakrsnu validaciju
	atribut 1 atribut 2 … ….atribut n-1 atribut n
Značaj (težine)	Korisnik može da upiše značaj (težinu) svakog od indeksa, iste ili različite za svaki od njih, ili težine mogu biti određene GA, kao što je u spovedenim istraživanjima

U nastavku su izdvojeni najznačajniji delovi koda za bazni model i to:

- **Normalizacija numeričkih atributa**, koja ima za cilj svođenje podataka na isti raspon vrednosti, tako da mogu međusobno da se upoređuju. Za normalizaciju indeksa korišćena je L∞ (Čebiševljeva) metrika koja podrazumeva da se maksimalna vrednost kolone uzima za normu. Normalizovana tabela odlučivanja se dobija deljenjem kolona baze podataka sa odgovarajućim normama.
Novi slučajevi bi se takođe delili normama, pri čemu se njihovim vrednostima dodeljuje 1, u slučaju da imaju vrednost indeksa veću od vrednosti određene norme.
Za kategoričke varijable, vrednosti u okviru ovog modela ostaju iste.

Zbog različitih mera sličnosti za numeričke i kategoričke atribute, u kodu je bilo bitno razdvojiti ove tipove podataka, i to tako što se čuvaju i različitim pomoćnim matricama (D – ukazuje na numeričke (double) i S – ukazuje na kategoričke (string) varijable).

```
For i = 1 To BrojAtributa
  If VarType(Worksheets("OriginalData").Cells(3, i + 1)) = vbDouble Then
    MaxVrednostAtributa(i) = 0
    For j = 1 To BrojSlucajeva
    If MaxVrednostAtributa(i) < Worksheets("OriginalData").Cells(2 + j, i + 1).Value Then
MaxVrednostAtributa(i) = Worksheets("OriginalData").Cells(2 + j, i + 1).Value
    Next j
    Worksheets("Sumarno").Cells(8, 1 + i).Value = MaxVrednostAtributa(i)
    End If
Next i

For i = 1 To BrojAtributa
If VarType(Worksheets("OriginalData").Cells(3, i + 1)) = vbDouble Then

For j = 1 To BrojSlucajeva
  NormalizovanAtribut(j, i) = Worksheets("OriginalData").Cells(2 + j, i + 1).Value /
MaxVrednostAtributa(i)
  PomocnaMatrica(j, i) = "D"
Next j

ElseIf VarType(Worksheets("OriginalData").Cells(3, i + 1)) = vbString Then
  For j = 1 To BrojSlucajeva
```

```
  NormalizovanAtribut(j, i) = Worksheets("OriginalData").Cells(2 + j, i + 1).Value
  PomocnaMatrica(j, i) = "S"
  Next j
End If

Next i
```

- **Normalizacija težina atributa** – formula za Euklidovu normu može dati preciznije rezultate klasifikacije ako se kriterijumi ponderišu adekvatnim ponderima, tj. težinama koje treba da ukažu na bitnost kriterijuma za odluku o davanju kredita.
Tokom istraživanja analizirana su dva bazna modela – jedan kod koga svi kriterijumi imaju podjednak značaj i drugi model kod koga su ponderi različiti.
Ponderi mogu uzimati vrednosti od 1 do 9 (9 – utiče mnogo, 1 – zanemarljivo utiče). Normalizacija težina atributa se vrši L1 metrikom, koja podrazumeva deljenje sa sumom svih pondera.

```
SumaTezinaAtributa = 0

For i = 1 To BrojAtributa
SumaTezinaAtributa = SumaTezinaAtributa + TezinaAtributa(i)
Next i

For i = 1 To BrojAtributa
NormalizovaneTezineAtributa(i) = TezinaAtributa(i) / SumaTezinaAtributa
Next i
```

- **Generisanje particija** (fold-ova) za unakrsnu validaciju. Particije unakrsne validacije se generišu korišćenjem takozvanog semena (seed) za slučajni izbor podataka. Broj particija za unakrsnu validaciju definiše sam korisnik, preko varijable BrojDataSetova.

```
BrojZaTestiranje = (BrojSlucajeva * (100 / BrojDataSetova)) / 100
```

* BrojDataSetova se odnosi na broj particija za unakrsnu validaciju
* BrojZaTestiranje – broj slučajeva iz baze koji treba da posluži za testiranje modela, ovi slučajevi se u kodu čuvaju u tzv. Pomocnoj matrici noseći oznaku „T“, za razliku od slučajeva iz baze koji imaju za cilj „učenje“, a koji nose oznaku „U“

```
BrojZaUcenje = BrojSlucajeva – BrojZaTestiranje

For s = 1 To BrojDataSetova
  pocetniSeed = -(s * s)
  For j = 1 To BrojZaTestiranje
    NizSlucajnihTestnih(s, j) = 0
  Next j

  For j = 1 To BrojZaTestiranje
    slucajanBroj = Int(BrojSlucajeva * Rnd(pocetniSeed) + 1)
    postoji = False
    For k = 1 To BrojZaTestiranje
     If slucajanBroj = NizSlucajnihTestnih(s, k) Then
      postoji = True
     End If
    Next k

    If Not postoji Then
      PomocnaMatricaSet(slucajanBroj, s) = "T"
      NizSlucajnihTestnih(s, j) = slucajanBroj
      pocetniSeed = -slucajanBroj
    Else
      pocetniSeed = -slucajanBroj - 1
      j = j - 1
    End If
  Next j

  For l = 1 To BrojSlucajeva
    If PomocnaMatricaSet(l, s) <> "T" Then
      PomocnaMatricaSet(l, s) = "U"
    End If
  Next l
Next s
```

- **Računanje sličnosti** slučajeva se opisuje preko linija koda koje slede.

U ovom delu nam je bitno da se za svaki „testni“ slučaj odredi sličnost sa svim slučajevima iz baze koji nose oznaku da su „učeći“. Sličnost se računa preko

Euklidskog rastojanja za numeričke, odnosno preko funkcije preklapanja za kategoričke varijable.

Ono na šta takođe treba obratiti pažnju je činjenica da moramo da čuvamo informacije o ishodima slučajeva koji nam služe za učenje. U tu svrhu je formiran PomocniNiz niz koji čuva podatak o odluci, a određuju ga sledeći članovi: particija unakrsne validacije, broj slučaja za testiranje i broj slučaja iz baze.

Postoje dva moguća ishoda: odobren („good“) ili odbijen („bad“).

```
For s = 1 To BrojDataSetova
  For i = 1 To BrojZaTestiranje
    For j = 1 To BrojSlucajeva

          If PomocnaMatricaSet(j, s) = "U" Then
      Udaljenost(s, NizSlucajnihTestnih(s, i), j) = 0
      If Worksheets("OriginalData").Cells(2 + j, BrojAtributa + 2).Value = "good" Then
       PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 1
     Else: PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 0
     End If


      For k = 1 To BrojAtributa
       If PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) =
NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then
         Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
       ElseIf PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) <>
NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then
         Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
       End If
       If PomocnaMatrica(j, k) = "D" And PomocnaMatrica(i, k) = "D" Then
         Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) +
((NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) ^ 2) *
NormalizovaneTezineAtributa(k)
       End If
      Next k
    Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) ^ (1 / 2)
```

```
    End If
   Next j
  Next i

Next s
```

- **Sortiranje** po blizini se vrši da bi se videlo koji slučaj, ili više njih, najviše odgovara slučaju za koji se traži rešenje. Sličnost se dobija kada se od 1 oduzme udaljenost.

```
For s = 1 To BrojDataSetova
  For i = 1 To BrojZaTestiranje
    For j = 1 To BrojSlucajeva
      For k = j + 1 To BrojSlucajeva - 1
        If PomocnaMatricaSet(k, s) = "U" Then
          If Udaljenost(s, NizSlucajnihTestnih(s, i), j) > Udaljenost(s, NizSlucajnihTestnih(s, i), k) Then
            PomProm = Udaljenost(s, NizSlucajnihTestnih(s, i), j)
            Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), k)
            Udaljenost(s, NizSlucajnihTestnih(s, i), k) = PomProm
            PomProm2 = PomocniNiz(s, NizSlucajnihTestnih(s, i), j)
            PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = PomocniNiz(s, NizSlucajnihTestnih(s, i), k)
            PomocniNiz(s, NizSlucajnihTestnih(s, i), k) = PomProm2
          End If
        End If
      Next k
    Next j
  Next i
Next s
```

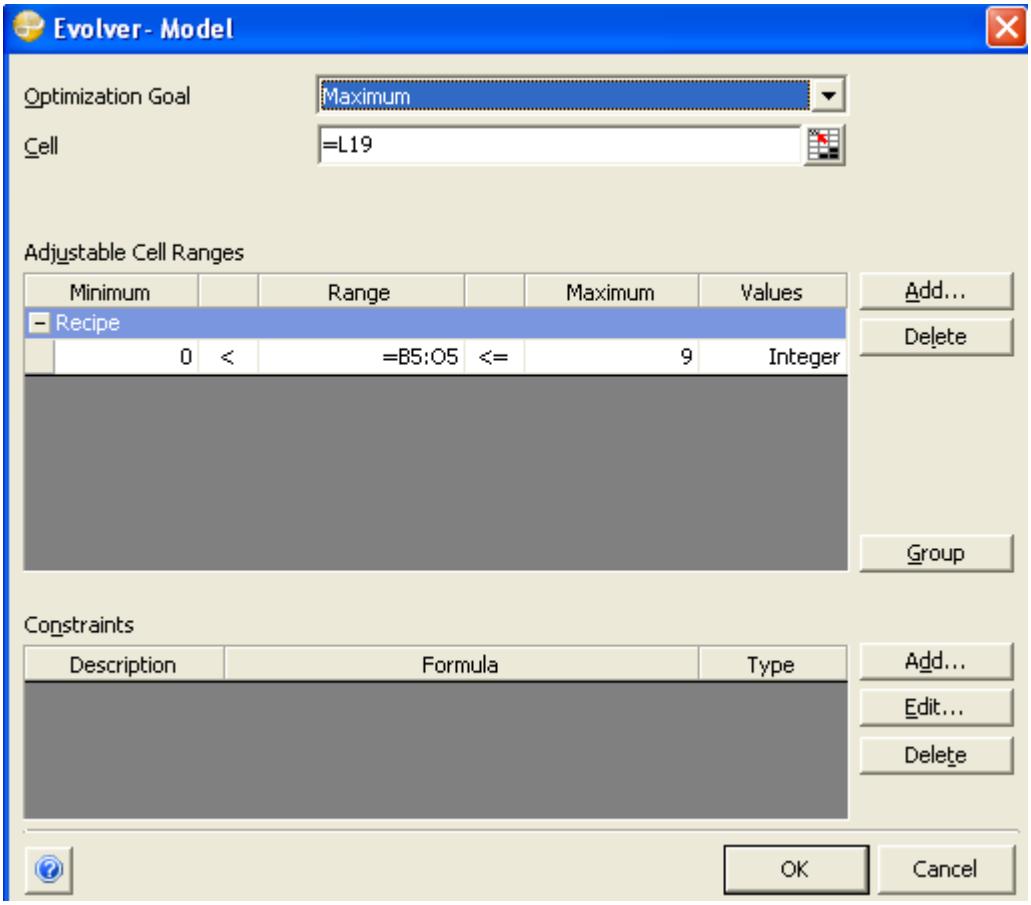
- **Određivanje odluke** za testni slučaj uzimajući u obzir ishode njemu najbližih suseda. BrojMinimuma je od strane korisnika definisan broj najbližih suseda koje treba uzimati u obzir. Radi uporedivosti podataka i analize, u ovom radu se razmatrao rezultat za 9 najbližih suseda, uzimajući u obzir samo neparne vrednosti.

Nakon toga, sledi poređenje da li je preporučena odluka za „testni“ slučaj ispravna, odnosno da li odgovara stvarno donetoj odluci.

```
For m = 1 To BrojMinimuma

For s = 1 To BrojDataSetova
  For i = 1 To BrojZaTestiranje
    OdlukaK = 0
    For k = 1 To m step 2
      OdlukaK = OdlukaK + PomocniNiz(s, NizSlucajnihTestnih(s, i), k)
    Next k
    If k = 1 And OdlukaK = 1 Then OdlukaMinK(s, NizSlucajnihTestnih(s, i)) = "good"
    If k = 1 And OdlukaK = 0 Then OdlukaMinK(s, NizSlucajnihTestnih(s, i)) = "bad"
    If k <> 1 Then
      If OdlukaK >= m / 2 Then
        OdlukaMinK(s, NizSlucajnihTestnih(s, i)) = "good"
      Else
        OdlukaMinK(s, NizSlucajnihTestnih(s, i)) = "bad"
      End If
    End If
  Next i
Next s

For s = 1 To BrojDataSetova
BrojacKOK = 0

For i = 1 To BrojZaTestiranje
  If OdlukaMinK(s, NizSlucajnihTestnih(s, i)) = Worksheets("OriginalData").Cells(2 +
NizSlucajnihTestnih(s, i), BrojAtributa + 2).Value Then
  BrojacKOK = BrojacKOK + 1
  End If
Next i

ProcenatKOk(s) = BrojacKOK / BrojZaTestiranje

Worksheets("Sumarno").Cells(9 + m, s + 1).Value = ProcenatKOk(s)


Next s
Next m
```

Za razliku od modela kod koga su svi atributi podjednako značajni i koji ne podrazumeva dodatne optimizacije, bazni model sa ponderisanim težinama atributa podrazumeva i optimizaciju GA kada su značaji atributa u pitanju.

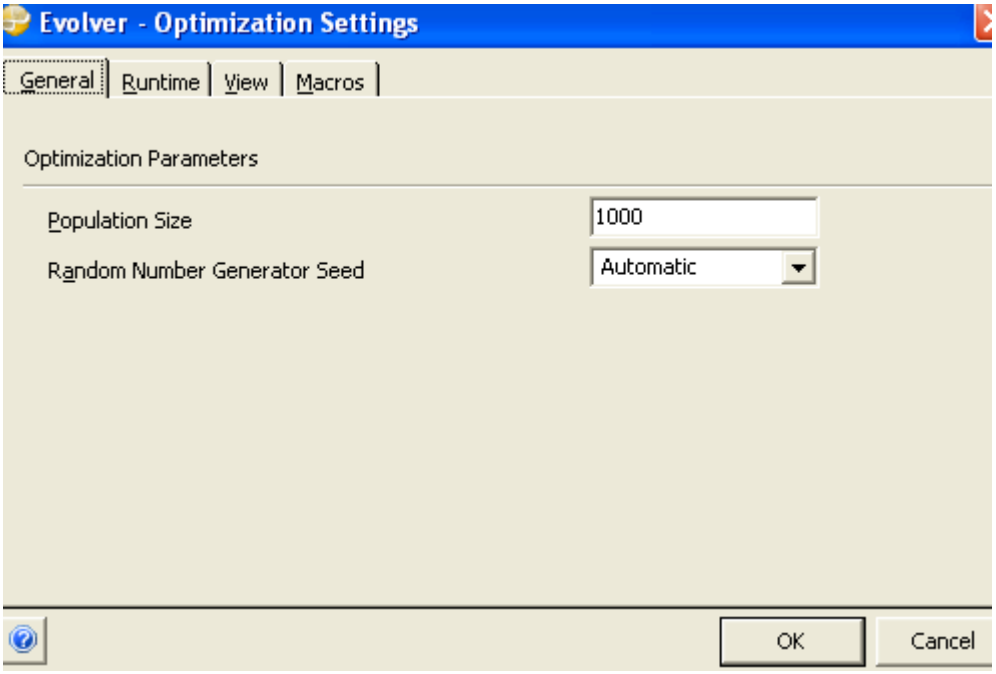
Optimizacija se radi GA primenom Evolvera. Podešavanje parametara GA – definisanje cilja i parametara za optimizaciju u baznom modelu su prikazani na slici 8. Cilj optimizacije je maksimizacija funkcije prilagođenosti (u slučaju u prilogu definisana je u ćeliji L19), koja nije ništa drugo do tačnost klasifikacije.
Maksimizacija tačnosti klasifikacije treba da se postigne menjajući težine atributa koje mogu uzimati celobrojne vrednosti od 1 do 9.



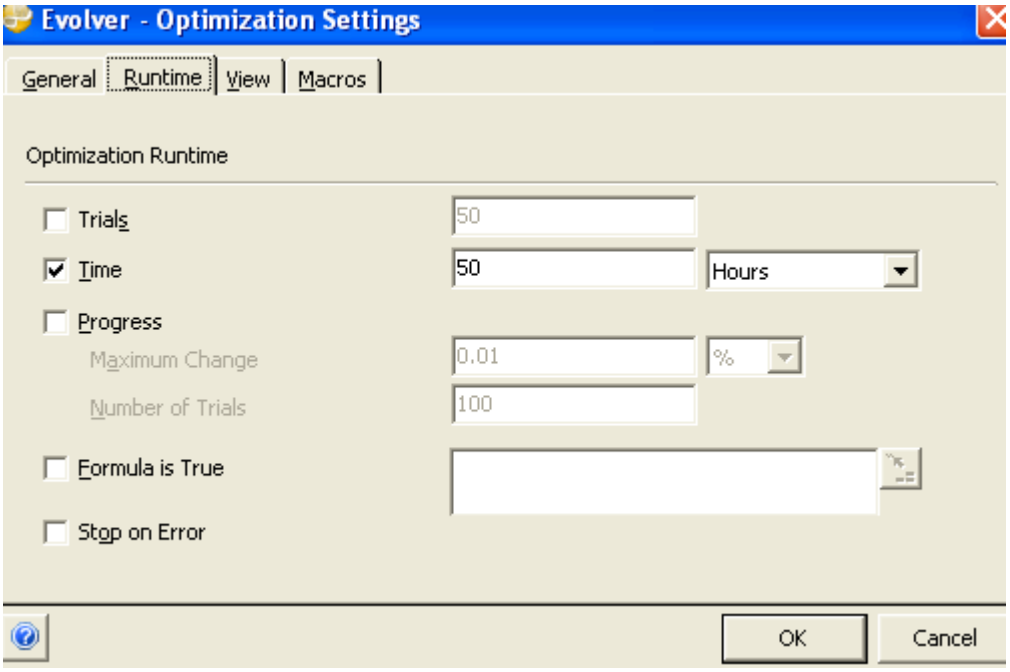
Slika 8. Podešavanje parametara GA – definisanje cilja i parametara za optimizaciju u baznom modelu

Slike u nastavku (slika 9, slika 10, slika 11, slika 12) ukazuju na parametre GA koji se mogu podestiti pomoću Evolvera. Kao kontrolni parametri pretrage GA u

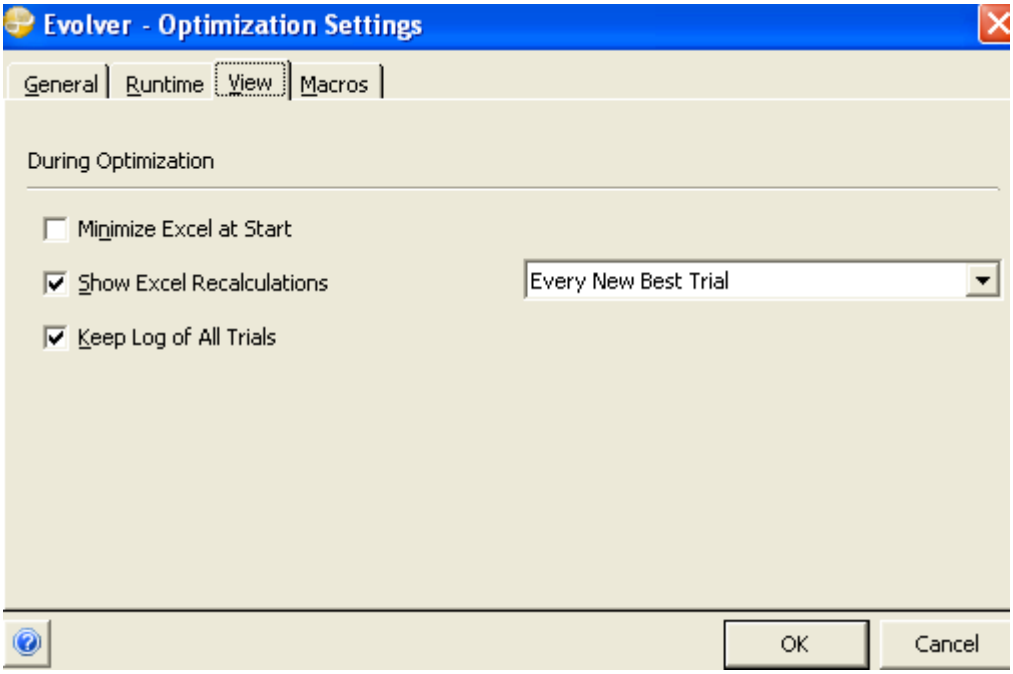
eksperimentima, podešene su sledeće vrednosti: 1000 organizama u populaciji, postavka ukrštanja na 0.5 i stope mutacije na 0.1. Kao kriterijum zaustavljanja koristiće se vreme, i to period od 50 sati, s obzirom da je ustanovljeno da je to vremenski interval nakon koga rešenje postaje stabilno.



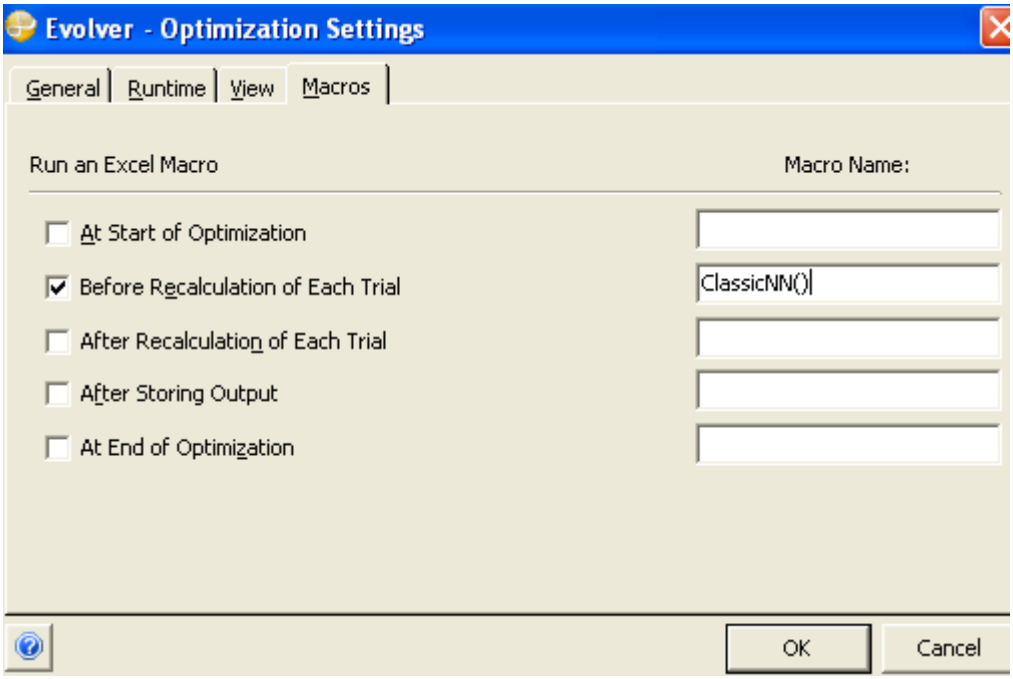
Slika 9. Podešavanje parametara GA – definisanje veličine populacije



Slika 10. Podešavanje parametara GA – definisanje vremena optimizacije



Slika 11. Podešavanje parametara GA – definisanje željenog izlaza



Slika 12. Podešavanje parametara GA – definisanje makroa koji stoji u pozadini

3.3. Projektovanje modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na ZOS-u, ali uključuje i domensko znanje, izraženo preko funkcija preferencija

Proces izgradnje modela klasifikacije CBR-PF-GA je sledeći:

Korak 1. Unos osnovnih informacija o skupu podataka.
Za svaki atribut treba da se definišu težine, kao i tip funkcije preferencije sa podrazumevanim parametrima za taj tip funkcije (na primer, vrednosti p i q sa slike 3). Za određivanje adekvatnih težina i parametara funkcije preferencije, za ovo istraživanje je korišćen GA sproveden pomoću Evolvera.

Varijable od značaja za sprovođenje modela CBR-PF-GA prikazane su u tabeli 5.

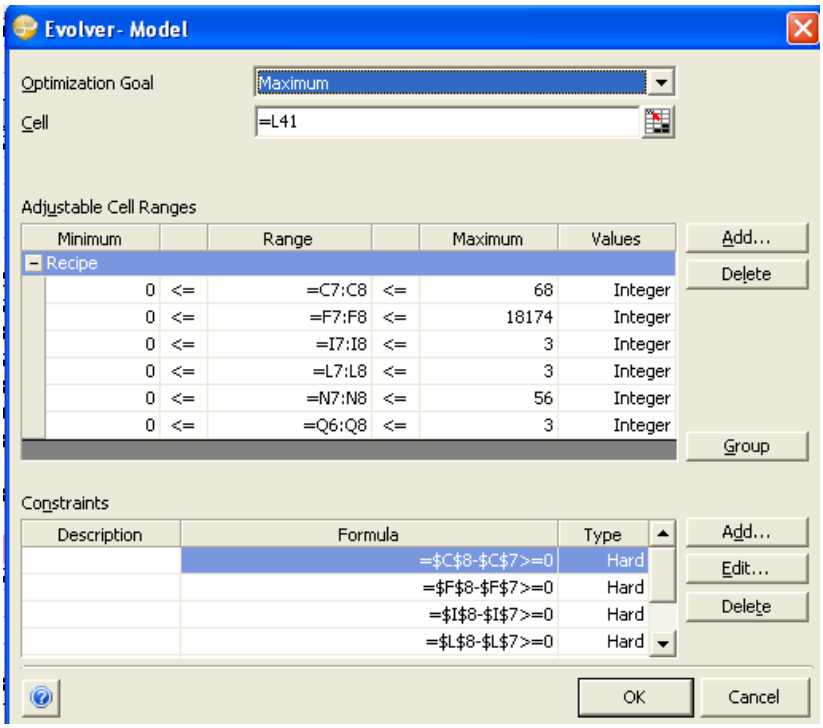
Tabela 5. Varijable od značaja za sprovođenje modela CBR-PF-GA

Broj slučajeva	Korisnik treba da unese ili sistem može da prepozna na osnovu broja redova u bazi
Broj atributa	Korisnik treba da unese ili sistem može da prepozna na osnovu broja kolona u bazi
Broj setova za unakrsnu validaciju	Korisnik treba da unese
	atribut 1 atribut 2 … ….atribut n-1 atribut n
Značaj (težine)	Korisnik može da upiše značaj (težinu) svakog od indeksa, iste ili različite za svaki od njih, ili težine mogu biti određene GA, kao što je u spovedenim istraživanjima
Tip funkcije preferencije	Za numeričke atribute, indekse, korisnik može definisati upisati tip funkcije preferencije za koji smatra da je najadekvatniji za taj atribut
Prag indiferencije	Za svaki numerički atribut, korisnik ili algoritam, kao u sprovedenim istraživanjima u ovom radu, određuje prag indiferencije, a uzimajući u obzir definisani tip funkcije preferencije za posmatrani atribut
Prag izričite preferencije	Za svaki numerički atribut, korisnik ili algoritam, kao u sprovedenim istraživanjima u ovom radu, određuje prag striktne preferencije, a uzimajući u obzir definisani tip funkcije preferencije za posmatrani atribut

Korak 2. Unos parametara za GA (veličina populacije, kriterijum zaustavljanja, itd.) i sprovoljenje ZOS postupka.

Pristup GA se u ovoj studiji koristi za utvrđivanje odgovarajućih parametara funkcija preferencija, kao i da se pronađu težine atributa. Tačnost klasifikacije modela je postavljena kao funkcija prilagođenosti GA.

Podešavanje parametara GA, tj. definisanje cilja i parametara za optimizaciju u modelu CBR-PF-GA, prikazani su na slici 13.



Slika 13. Podešavanje parametara GA – definisanje cilja i parametara za optimizaciju u modelu CBR-PF-GA

Za merenje sličnosti između slučajeva koristi se metoda najbližeg suseda.

U ovom modelu za merenje sličnosti kategiričkih atributa koristi se funkcija preklapanja, tj. udaljenost je 1 ako su vrednosti koje se porede različite, odnosno udaljenost je 0 ako su vrednosti iste.

Za numeričke atribute, model CBR-PF-GA koristi funkciju preferencije tipa 5.

Za pronalaženje slučajeva, koristi se k-NN metod. Pošto se različitim istraživanjima pokazalo da optimalna vrednost k zavisi od većeg broja parametara, svi predloženi modeli su testirani uzimajući u obzir nekoliko k najbližih suseda, koji zapravo variraju od 1 do 9, uzimajući u obzir samo neparne vrednosti.

Za svaki slučaj iz skupa podataka za testiranje, model treba da dodeli jednu od dve moguće klase, i to mereći udaljenost svakog posmatranog slučaja sa svakim slučajem iz baze slučajeva za učenje. Za procenu učinka prediktivnog modela, koristi se metodologija 10-ostruke unakrsne validacije (engl. 10-fold cross validation). Particije (fold) unakrsne validacije se generišu korišćenjem takozvanog semena za slučajni izbor podataka.

Korak 3. Izračunavanje funkcije prilagođenosti za svaki hromozom.

GA funkcioniše tako da nasumično generiše skup rešenja. Taj skup rešenja predstavlja početnu populaciju. Svako rešenje u populaciji se zove hromozom i obično se daje u obliku binarnog stringa. Nakon generisanja početne populacije, GA izračunava funkciju prilagođenosti za svaki hromozom. Funkcija prilagođenosti u ovom slučaju nije ništa drugo do tačnost klasifikacije.

Funkcija prilagođenosti je korisnički definisana funkcija. Ona vraća rezultate ocene svakog hromozoma (generisanog rešenja), tako da veće vrednosti prilagođenosti znače da je bolji hromozom.

Korak 4. Primena genetskih operatora i generisanje potomaka (nove generacije).
Ovaj korak podrazumeva primenu osnovnih operatora, kao što su: selekcija, ukrštanje i mutacija. Njihovom primenom se proizvodi nova generacija populacije. Operator selekcije određuje koji hromozom će preživeti. Ukrštanjem se potomci iz parova hromozoma razmenjuju i tako stvaraju nove parove hromozoma. Mutacijom, gde je stopa mutacije obično mala, proizvoljno izabrani bitovi u hromozomu se obrću. Ovi koraci evolucije se nastavljaju sve dok se zadovolje uslovi za zaustavljanje. U većini

slučajeva, kriterijum zaustavljanja je postavljen na maksimalnom broju generacija (Chiu, 2002; Fu & Shen, 2004; Han & Kamber, 2001).

U petom koraku, proces evolucije genetskog algoritma se nastavlja u smeru maksimizacije vrednosti funkcije prilagođenosti.

Korak 5. Koraci 3-5 se ponavljaju sve dok se ne zadovolje kriterijumi zaustavljanja. Po završetku procesa evolucije, u skladu sa definisanim kriterijumom zaustavljanja, najbolji parametri do kojih se došlo se pamte. U ovom slučaju to su podaci o parametrima funkcija preferencija za sve numeričke indekse i težine svih indeksa, kako numeričkih tako i kategoričkih.

Najznačajniji deo koda, po kome se ovaj model bitno razlikuje od baznog modela, je računanje sličnosti za numeričke indekse preko funkcija preferencija.
Upravo je taj deo koda prizakan u nastavku. Radi konzistentnosti nazivi promenjivih u kodu su ostale iste kao i za bazni model, uz napomenu da u ovom modelu nije potrebno vršiti normalizaciju vrednosti atributa, pošto se primenom funkcija preferencije vrednosti svode na opseg vrednosti od 0 do 1. Na taj način, NormalizovanAtribut nije ništa drugo do originalna vrednost atributa.

Normalizacija težina atribuda je zadržana kao i u prethodnom modelu.

Kod modela sa primenom funkcija preferencije, kao i kod baznog modela, postoje dve varijante – sa neponderisanim težinama atributa (svi indeksi su podjednako značajni) i sa ponderisanim težinama atributa (indeksi većeg značaja imaju veći ponder).

Sam model korisniku daje slobodu da za svaki indeks može da ukuca tip funkcije preferencije koji smatra da je najadekvatniji. U zavisnosti od izabrane funkcije treba definisati odgovarajuće parametre. Parametri funkcija se mogu definisati od strane korisnika ili mogu biti, kao u ovom radu, određeni od strane GA.

```
For s = 1 To BrojDataSetova
  For i = 1 To BrojZaTestiranje
    For j = 1 To BrojSlucajeva
      If PomocnaMatricaSet(j, s) = "U" Then
       Udaljenost(s, NizSlucajnihTestnih(s, i), j) = 0
       If Worksheets("OriginalData").Cells(2 + j, BrojAtributa + 2).Value = "good" Then
        PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 1
      Else: PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 0
      End If


  For k = 1 To BrojAtributa

  If PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) =
NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then
        Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
  ElseIf PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) <>
NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then
        Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  End If
  If PomocnaMatrica(j, k) = "D" Then

  If TipPreferencije(k) = 1 Then
  If Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <> 0 Then
    Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
    Else: Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
  End If

  ElseIf TipPreferencije(k) = 2 Then
    If (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) >
MinRazlika(k) Then
    Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
    Else: Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
  End If

  ElseIf TipPreferencije(k) = 3 Then
```

```
   If (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) = 0 Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > 0) And
(Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <= MaxRazlika(k))
Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) +
((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) / MaxRazlika(k))
* NormalizovaneTezineAtributa(k)
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MaxRazlika(k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  End If

  ElseIf TipPreferencije(k) = 4 Then
   If ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <=
MinRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
   ElseIf ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MinRazlika(k)) And ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j,
k))) <= MaxRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 / 2 *
NormalizovaneTezineAtributa(k)
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MaxRazlika(k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  End If

  ElseIf TipPreferencije(k) = 5 Then
   If ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <=
MinRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
   ElseIf ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MinRazlika(k)) And ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j,
k))) <= MaxRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) +
(Abs(((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) -
MinRazlika(k))) / (MaxRazlika(k) - MinRazlika(k))) * NormalizovaneTezineAtributa(k)
```

```
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MaxRazlika(k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  End If


End If
End If

  Next k


     End If
    Next j
  Next i

Next s
```

3.4. Projektovanje modela za ocenu kreditne sposobnosti klijenata koji je zasnovan na ZOS-u, uključuje domensko znanje, izraženo preko funkcija preferencija, ali i odabrane mere sličnosti za kategoričke atribute

Model za ocenu kreditne sposobnosti klijenata koji je zasnovan na ZOS-u i uključuje domensko znanje izraženo preko funkcija preferencija, kao i odabrane mere sličnosti za kategoričke atribute, je sličan prethodno predstavljenom modelu, ali je i različit u delu koji se tiče merenja sličnosti za kategoričke varijable.

Napravljene su tri moguće verzije ovog modela u zavisnosti od izabrane mere sličnosti za kategoričke atribute, a to su:
- ponderisani model zasnovan na ZOS-u, koji uključuje domensko znanje izraženo preko funkcija preferencija, kao i GOODALL1 meru sličnosti za kategoričke atribute (Weighted CBR-PF-GA Goodall1),
- ponderisani model zasnovan na ZOS-u, koji uključuje domensko znanje izraženo preko funkcija preferencija, kao i OF meru sličnosti za kategoričke atribute (Weighted CBR-PF-GA OF),
- ponderisani model zasnovan na ZOS-u, koji uključuje domensko znanje izraženo preko funkcija preferencija, kao i ESKIN meru sličnosti za kategoričke atribute (Weighted CBR-PF-GA-ESKIN).

S obzirom da su istraživanja pokazala da modeli sa ponderisanim težinama indeksa uvek daju bolje rezultate klasifikacije od modela sa jednakim značajem atributa, nisu razmatrane varijante modela jednakih težina varijabli.

Proces izgradnje modela svakog od ova tri modela klasifikacije sadrži korake opisane kod modela iz prethodnog poglavlja, koji uključuje funkcije preferencije, odnosno podrazumeva sledeće:

Korak 1. Unos osnovnih informacija o skupu podataka.

Za svaki atribut treba da se definišu težine, kao i tip funkcije preferencije sa podrazumevanih parametrima za taj tip funkcije (na primer, vrednosti p i q sa slike 3). Za određivanje adekvatnih težina i parametara funkcije preferencije, za ovo istraživanje je korišćen GA sproveden pomoću Evolvera.

Korak 2. Unos parametara za GA (veličina populacije, kriterijum zaustavljanja, itd.) i sprovoljenje ZOS postupka.

Pristup GA se u ovoj studiji koristi za utvrđivanje odgovarajućih parametara funkcija preferencija, kao i da se pronađu težine atributa. Tačnost klasifikacije modela je postavljena kao funkcija prilagođenosti GA.

Za merenje sličnosti između slučajeva koristi se metoda najbližeg suseda.

Za merenje sličnosti kategiričkih atributa u svakom od modela koristi se neka od odabranih funkcija – GOODALL 1, OF ili ESKIN.
Za numeričke atribute, model CBR-PF-GA koristi funkciju preferencije tipa 5.

Za pronalaženje slučajeva, koristi se k-NN metod. Pošto se različitim istraživanjima pokazalo da optimalna vrednost k zavisi od većeg broja parametara, svi predloženi modeli su testirani uzimajući u obzir nekoliko k najbližih suseda, koji zapravo variraju od 1 do 9, uzimajući u obzir samo neparne vrednosti.

Za svaki slučaj iz skupa podataka za testiranje, model treba da dodeli jednu od dve moguće klase, i to mereći udaljenost svakog posmatranog slučaja sa svakim slučajem iz baze slučajeva za učenje. Za procenu učinka prediktivnog modela, koristi se metodologija 10-ostruke unakrsne validacije (engl. 10-fold cross validation). Particije (fold) unakrsne validacije se generišu korišćenjem takozvanog semena za slučajni izbor podataka.

Korak 3. Izračunavanje funkcije prilagođenosti za svaki hromozom.

GA funkcioniše tako da nasumično generiše skup rešenja. Taj skup rešenja predstavlja početnu populaciju. Svako rešenje u populaciji se zove hromozom i obično se daje u obliku binarnog stringa. Nakon generisanja početne populacije, GA izračunava funkciju prilagođenosti za svaki hromozom. Funkcija prilagođenosti u ovom slučaju nije ništa drugo do tačnost klasifikacije.
Funkcija prilagođenosti je korisnički definisana funkcija. Ona vraća rezultate ocene svakog hromozoma (generisanog rešenja), tako da veće vrednosti prilagođenosti znače da je bolji hromozom.

Korak 4. Primena genetskih operatora i generisanje potomaka (nove generacije).
Ovaj korak podrazumeva primenu osnovnih operatora, kao što su: selekcija, ukrštanje i mutacija. Njihovom primenom se proizvodi nova generacija populacije. Operator selekcije određuje koji hromozom će preživeti. Ukrštanjem se potomci iz parova hromozoma razmenjuju i tako stvaraju nove parove hromozoma. Mutacijom, gde je stopa mutacije obično mala, proizvoljno izabrani bitovi u hromozomu se obrću. Ovi koraci evolucije se nastavljaju sve dok se zadovolje uslovi za zaustavljanje. U većini slučajeva, kriterijum zaustavljanja je postavljen na maksimalnom broju generacija (Chiu, 2002; Fu & Shen, 2004; Han & Kamber, 2001).

U petom koraku, proces evolucije genetskog algoritma se nastavlja u smeru maksimizacije vrednosti funkcije prilagođenosti.

Korak 5. Koraci 3-5 se ponavljaju sve dok se ne zadovolje kriterijumi zaustavljanja. Po završetku procesa evolucije, u skladu sa definisanim kriterijumom zaustavljanja, najbolji parametri do kojih se došlo se pamte. U ovom slučaju to su podaci o parametrima funkcija preferencija za sve numeričke indekse i težine svih indeksa, kako numeričkih tako i kategoričkih.

Najznačajniji deo koda, po kome se ovaj model bitno razlikuje od prethodno predstravljenih, je računanje sličnosti za kategoričke indekse preko odabranih funkcija sličnosti za kategoričke atribute.
Ti delovi koda su prikazani u nastavku. Radi konzistentnosti nazivi promenjivih u kodu su ostale iste kao i za bazni model, uz napomenu da u ovom modelu nije potrebno vršiti normalizaciju vrednosti atributa, pošto se primenom funkcija preferencije vrednosti svode na opseg vrednosti od 0 do 1.

Sam model korisniku daje slobodu da za svaki indeks može da ukuca tip funkcije preferencije koji smatra da je najadekvatniji. U zavisnosti od izabrane funkcije treba definisati odgovarajuće parametre. Parametri funkcija se mogu definisati od strane korisnika ili mogu biti, kao u ovom radu, određeni od strane GA.

Ponderisani model CBR-PF-GA OF

Za ponderisani model CBR-PF-GA OF je bilo neophodno kreirati poseban niz koji bi sadržao različite atribute, s obzirom da u zavisnosti od skupa podataka broj kategoričkih atributa može varirati. Niz koji sadrži različite kategoričke atribute u posmatranom skupu podataka je nazvan NizRazlicitihAtributa. U skladu sa prethodno navenim karakteristikama kategoričkih podataka, a za izračunavanje željenih mera sličnosti, neophodno je definisati i niz FrAtr koji govori o učestalosti pojavljivanja određenih vrednosti svakog atributa, kao i niz koji bi prebrojavao broj mogućih vrednosti atributa – brojRazlicitihVrednostiAtributa.

```
For s = 1 To BrojDataSetova

For i = 1 To BrojAtributa
t = 0
For l = 1 To BrojSlucajeva
  If PomocnaMatricaSet(l, s) = "U" Then
    If PomocnaMatrica(l, i) = "S" Then
```

```
      pronasao = False
      For m = 1 To t
        If NizRazlicitihAtributa(s, i, m) = NormalizovanAtribut(l, i) Then pronasao = True
      Next m

      If pronasao = False Then
        t = t + 1
        NizRazlicitihAtributa(s, i, t) = NormalizovanAtribut(l, i)
      End If
    End If
  End If
  Next l
   brojRazlicitihVrednostiAtributa(i) = t
 Next i

 For i = 1 To BrojAtributa
  For k = 1 To brojRazlicitihVrednostiAtributa(i)
    FrAtr(s, i, k) = 0

    For l = 1 To BrojSlucajeva
      If PomocnaMatricaSet(l, s) = "U" Then
        If PomocnaMatrica(l, i) = "S" Then

          If NizRazlicitihAtributa(s, i, k) = NormalizovanAtribut(l, i) Then
            FrAtr(s, i, k) = FrAtr(s, i, k) + 1
          End If

        End If
      End If
    Next l

   Next k
 Next i

 Next s

 For s = 1 To BrojDataSetova
   For i = 1 To BrojZaTestiranje
     For j = 1 To BrojSlucajeva
```

```
    If PomocnaMatricaSet(j, s) = "U" Then
     Udaljenost(s, NizSlucajnihTestnih(s, i), j) = 0
     If Worksheets("OriginalData").Cells(2 + j, BrojAtributa + 2).Value = "good" Then
      PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 1
    Else: PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 0
    End If


  For k = 1 To BrojAtributa

  If PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) = NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then

    Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0 * NormalizovaneTezineAtributa(k)

  ElseIf PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) <> NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then

   For g = 1 To k

    For p = 1 To brojRazlicitihVrednostiAtributa(k)

      If NormalizovanAtribut(j, k) = NizRazlicitihAtributa(s, g, p) Then
      pomfk1 = p
      End If

      If NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) = NizRazlicitihAtributa(s, g, p) Then
      pomfk2 = p
      End If
    Next p

   Next g


    Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + (1 / (1 / (1 + (Log(BrojZaTestiranje / FrAtr(s, k, pomfk1)) * Log(BrojZaTestiranje / FrAtr(s, k, pomfk2))))) - 1) * NormalizovaneTezineAtributa(k)
```

```
End If

If PomocnaMatrica(j, k) = "D" Then

If TipPreferencije(k) = 1 Then
If Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <> 0 Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  Else: Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
End If

ElseIf TipPreferencije(k) = 2 Then
  If (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) >
MinRazlika(k) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  Else: Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
End If

ElseIf TipPreferencije(k) = 3 Then
  If (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) = 0 Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
  ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > 0) And
(Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <= MaxRazlika(k))
Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) +
((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) / MaxRazlika(k))
* NormalizovaneTezineAtributa(k)
  ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MaxRazlika(k)) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
End If

 ElseIf TipPreferencije(k) = 4 Then
  If ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <=
MinRazlika(k))) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
```

```
   ElseIf ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MinRazlika(k)) And ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) <= MaxRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 / 2 * NormalizovaneTezineAtributa(k)
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MaxRazlika(k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 * NormalizovaneTezineAtributa(k)
  End If

 ElseIf TipPreferencije(k) = 5 Then
   If ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <= MinRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
   ElseIf ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MinRazlika(k)) And ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) <= MaxRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + (Abs(((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) - MinRazlika(k))) / (MaxRazlika(k) - MinRazlika(k))) * NormalizovaneTezineAtributa(k)
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MaxRazlika(k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 * NormalizovaneTezineAtributa(k)
 End If


End If
End If

 Next k


    End If

  Next j

 Next i
```

```
Next s
```

Ponderisani model CBR-PF-GA-ESKIN

Za ponderisani model CBR-PF-GA ESKIN je potrebno uvesti niz koji bi sadržao različite atribute, jer u zavisnosti od skupa podataka broj kategoričkih atributa može varirati. Ovaj niz je nazvan NizRazlicitihAtributa. Da bi se pristupilo izračunavanju mere Eskin potrebna je i informacija o broju mogućih vrednosti svakog kategoričkog atributa, pa je otuda uveden niz brojRazlicitihVrednostiAtributa.

```
For s = 1 To BrojDataSetova

For i = 1 To BrojAtributa
t = 0
For l = 1 To BrojSlucajeva
  If PomocnaMatricaSet(l, s) = "U" Then
    If PomocnaMatrica(l, i) = "S" Then

      pronasao = False
      For m = 1 To t
        If NizRazlicitihAtributa(s, i, m) = NormalizovanAtribut(l, i) Then pronasao = True
      Next m

      If pronasao = False Then
        t = t + 1
        NizRazlicitihAtributa(s, i, t) = NormalizovanAtribut(l, i)
      End If
    End If
  End If
  Next l
   brojRazlicitihVrednostiAtributa(i) = t 'mozda ne treba
 Next i
```

```
Next s

For s = 1 To BrojDataSetova
 For i = 1 To BrojZaTestiranje
  For j = 1 To BrojSlucajeva
   If PomocnaMatricaSet(j, s) = "U" Then
    Udaljenost(s, NizSlucajnihTestnih(s, i), j) = 0
    If Worksheets("OriginalData").Cells(2 + j, BrojAtributa + 2).Value = "good" Then
     PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 1
    Else: PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 0
    End If


 For k = 1 To BrojAtributa

 If PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) =
NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0 *
NormalizovaneTezineAtributa(k)

 ElseIf PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) <>
NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then

   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + (1 /
((brojRazlicitihVrednostiAtributa(k) * brojRazlicitihVrednostiAtributa(k)) /
((brojRazlicitihVrednostiAtributa(k) * brojRazlicitihVrednostiAtributa(k)) + 2)) - 1) *
NormalizovaneTezineAtributa(k)

 End If


 If PomocnaMatrica(j, k) = "D" Then

 If TipPreferencije(k) = 1 Then
 If Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <> 0 Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  Else: Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
 End If
```

```
ElseIf TipPreferencije(k) = 2 Then
  If (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) > MinRazlika(k) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 * NormalizovaneTezineAtributa(k)
  Else: Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
 End If

 ElseIf TipPreferencije(k) = 3 Then
  If (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) = 0 Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
  ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > 0) And (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <= MaxRazlika(k)) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) / MaxRazlika(k)) * NormalizovaneTezineAtributa(k)
  ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MaxRazlika(k)) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 * NormalizovaneTezineAtributa(k)
 End If

  ElseIf TipPreferencije(k) = 4 Then
  If ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <= MinRazlika(k))) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
  ElseIf ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MinRazlika(k)) And ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) <= MaxRazlika(k))) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 / 2 * NormalizovaneTezineAtributa(k)
  ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MaxRazlika(k)) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 * NormalizovaneTezineAtributa(k)
  End If
```

```
 ElseIf TipPreferencije(k) = 5 Then
  If ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <=
MinRazlika(k))) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
  ElseIf ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MinRazlika(k)) And ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j,
k))) <= MaxRazlika(k))) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) +
(Abs(((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) -
MinRazlika(k))) / (MaxRazlika(k) - MinRazlika(k))) * NormalizovaneTezineAtributa(k)
  ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MaxRazlika(k)) Then
  Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
 End If


End If
End If

 Next k


    End If
  Next j
 Next i

Next s


For s = 1 To BrojDataSetova
 For i = 1 To BrojZaTestiranje
    For j = 1 To BrojSlucajeva
     For k = j + 1 To BrojSlucajeva
      If PomocnaMatricaSet(k, s) = "U" Then
      If Udaljenost(s, NizSlucajnihTestnih(s, i), j) > Udaljenost(s, NizSlucajnihTestnih(s, i), k) Then
          PomProm = Udaljenost(s, NizSlucajnihTestnih(s, i), j)
          Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), k)
          Udaljenost(s, NizSlucajnihTestnih(s, i), k) = PomProm
```

```
            PomProm2 = PomocniNiz(s, NizSlucajnihTestnih(s, i), j)
            PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = PomocniNiz(s, NizSlucajnihTestnih(s, i), k)
            PomocniNiz(s, NizSlucajnihTestnih(s, i), k) = PomProm2
          End If
        End If
      Next k
    Next j
  Next i
Next s
```

Ponderisani model CBR-PF-GA- GOODALL1

Ponderisani model CBR-PF-GA Goodall1 za izračunavanje sličnosti između kategoričkih varijabli zahteva da se prethodno napravi niz različitih kategoričkih atributa, kod koga u zavisnosti od skupa podataka broj kategoričkih atributa može varirati. Taj niz je nazvan NizRazlicitihAtributa. Dodatno, za izračunavanje Goodall1 mere sličnosti, neophodno je definisati i niz FrAtr koji govori o učestalosti pojavljivanja određenih vrednosti svakog atributa, kao i niz koji bi prebrojavao broj mogućih vrednosti atributa – brojRazlicitihVrednostiAtributa.

```
For s = 1 To BrojDataSetova

For i = 1 To BrojAtributa
t = 0
For l = 1 To BrojSlucajeva
  If PomocnaMatricaSet(l, s) = "U" Then
    If PomocnaMatrica(l, i) = "S" Then

      pronasao = False
      For m = 1 To t
        If NizRazlicitihAtributa(s, i, m) = NormalizovanAtribut(l, i) Then pronasao = True
      Next m

      If pronasao = False Then
        t = t + 1
```

```
        NizRazlicitihAtributa(s, i, t) = NormalizovanAtribut(l, i)
      End If
    End If
  End If
  Next l
   brojRazlicitihVrednostiAtributa(i) = t 'mozda ne treba
 Next i

 For i = 1 To BrojAtributa
  For k = 1 To brojRazlicitihVrednostiAtributa(i)
    FrAtr(s, i, k) = 0

    For l = 1 To BrojSlucajeva
      If PomocnaMatricaSet(l, s) = "U" Then
        If PomocnaMatrica(l, i) = "S" Then

          If NizRazlicitihAtributa(s, i, k) = NormalizovanAtribut(l, i) Then
            FrAtr(s, i, k) = FrAtr(s, i, k) + 1
          End If

        End If
      End If
    Next l

   Next k
 Next i



 Next s

 For s = 1 To BrojDataSetova
   For i = 1 To BrojZaTestiranje
     For j = 1 To BrojSlucajeva
       If PomocnaMatricaSet(j, s) = "U" Then
        Udaljenost(s, NizSlucajnihTestnih(s, i), j) = 0
        If Worksheets("OriginalData").Cells(2 + j, BrojAtributa + 2).Value = "good" Then
         PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 1
       Else: PomocniNiz(s, NizSlucajnihTestnih(s, i), j) = 0
```

```
    End If


 For k = 1 To BrojAtributa

 If PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) =
NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then

      If FrAtr(s, k, brojRazlicitihVrednostiAtributa(k)) > 1 Then
      Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + (1 / (1 -
((FrAtr(s, k, brojRazlicitihVrednostiAtributa(k)) * (FrAtr(s, k, brojRazlicitihVrednostiAtributa(k)) - 1)) /
(BrojZaTestiranje * (BrojZaTestiranje - 1)))) - 1) * NormalizovaneTezineAtributa(k)
      End If

      If FrAtr(s, k, brojRazlicitihVrednostiAtributa(k)) = 1 Then
      Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0 *
NormalizovaneTezineAtributa(k)
      End If


 ElseIf PomocnaMatrica(j, k) = "S" And (NormalizovanAtribut(j, k) <>
NormalizovanAtribut(NizSlucajnihTestnih(s, i), k)) Then
      Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
 End If


 If PomocnaMatrica(j, k) = "D" Then

 If TipPreferencije(k) = 1 Then
 If Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <> 0 Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
   Else: Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
 End If

 ElseIf TipPreferencije(k) = 2 Then
   If (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) >
MinRazlika(k) Then
```

```
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
   Else: Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
  End If

  ElseIf TipPreferencije(k) = 3 Then
   If (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) = 0 Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > 0) And
(Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <= MaxRazlika(k))
Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) +
((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) / MaxRazlika(k))
* NormalizovaneTezineAtributa(k)
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MaxRazlika(k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  End If

  ElseIf TipPreferencije(k) = 4 Then
   If ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <=
MinRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
   ElseIf ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MinRazlika(k)) And ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j,
k))) <= MaxRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 / 2 *
NormalizovaneTezineAtributa(k)
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) >
MaxRazlika(k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 *
NormalizovaneTezineAtributa(k)
  End If

  ElseIf TipPreferencije(k) = 5 Then
   If ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) <=
MinRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 0
```

```
   ElseIf ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MinRazlika(k)) And ((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) <= MaxRazlika(k))) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + (Abs(((Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k))) - MinRazlika(k))) / (MaxRazlika(k) - MinRazlika(k))) * NormalizovaneTezineAtributa(k)
   ElseIf (Abs(NormalizovanAtribut(NizSlucajnihTestnih(s, i), k) - NormalizovanAtribut(j, k)) > MaxRazlika(k)) Then
   Udaljenost(s, NizSlucajnihTestnih(s, i), j) = Udaljenost(s, NizSlucajnihTestnih(s, i), j) + 1 * NormalizovaneTezineAtributa(k)
  End If


End If
End If

  Next k


     End If
   Next j
  Next i

Next s
```

4. Sprovođenje istraživanja

Ljudi nisu mudri u srazmeri sa svojim iskustvom, već sa svojom sposobnosti da iskustvo prime.

Bernard Shaw

Nema ništa teže, i zbog toga ništa vrednije, od sposobnosti donošenja odluka!

Napoleon Bonaparte

Najlakše je doneti ispravnu odluku kada nema loših opcija.

Robert Half

U ovom poglavlju disertacije se daje opis sprovedenog istraživanja, uz prikaz korišćenih baza slučajeva i rezultata performansi izvedenih modela ZOS-a za svaki od razmatranih skupova podataka. Ovo poglavlje sadrži i rezultate testiranja statističke značajnosti otkrivenih razlika u rezultatima modela.

Da bi se ispitale mogućnosti predloženih modela poslovne inteligencije koji su zasnovani na zaključivanju na osnovu slučajeva i izboru različitih mera sličnosti, u ovom istraživačkom radu za potrebe testiranja modela korišćeno je nekoliko baza podataka koje se tiču postupka odobravanja kredita stanovništvu u bankama.

Kreditna funkcija je jedna od najvažnijih za banku, iz razloga što se najveći prihod banke stvara upravo kreditnim poslovima, odnosno pozajmljivanjem. Glavna opasnost pri davanju kredita je da klijent neće biti u stanju da ispuni svoje obaveze prema banci i da će banka time izgubiti sredstva. Rast potražnje za kreditima vremenom je doveo do povećane zainteresovanosti za korišćenjem formalnijih i objektivnijih metoda, čime bi se institucijama koje plasiraju kredite pomoglo u odlučivanju da li da odobre kredit podnosiocu zahteva.

Problem odlučivanja da li podnosiocu zahteva odobriti kredit je tipični problem klasifikacije, čiji je zadatak da predvidi da li potencijalni klijent nosi povoljan ili

nepovoljan kreditni rizik. Ovo predviđanje se bazira na karakteristikama klijenta i zavisi od prethodnog iskustva, npr. primeri poznatih ishoda prethodnih slučajeva.

Prilikom postupka odobravanja kredita, razmatraju se različite karakteristike klijenata koji apliciraju za kredit, kao što su godine, bračni status i svrha kredita, finansijski podaci (npr. prihodi koje pojedinac ostvaruje, podaci o postojećim kreditima, itd.), lične karakteristike zajmotražioca, kao što su stalnost zaposlenja, rezidencijalni status, stalnost adrese stanovanja ili firme, itd. Od niza varijabli koje se spominju treba odrediti one koji najviše utiču na odluku o odobrenju kredita.

Činjenica je da banke svakodnevno rešavaju veliki broj zahteva stanovništva za kreditima. Da bi se utvrdilo da li je neko sposoban da vrati kredit koriste se razne tehnike i metode koje zahtevaju ekspertsko znanje. Aplikacija za odobravanje kredita može da se primeni ako banka ima dokumentaciju o prethodno odobrenim kreditima koju su bili uspešni. Takvi podaci služe kao osnova za bazu znanja.
Tehnološki napredak omogućava kreiranje bržih i boljih predikcionih sistema. Ovi sistemi mogu biti zasnovani na tehnikama otkrivanja zakonitosti u podacima i metodama veštačke inteligencije. Njihova primena može doprineti kako smanjenju troškova i rizika kojima se banka izlaže, tako i povećanju profita.

Bazična pretpostavka za primenu predloženih modela i procesa ZOS-a je da se istorijsko ponašanje reflektuje na buduće ponašanje.

4.1. Baze slučajeva

U diserticiji će se prikazani rezultati istraživačkog rada nad tri skupa podataka ilustrovanih u tabeli 6. Skupovi podataka Australian i German su dostupni na "UCI Repository of Machine Learning Databases" (Frank & Asuncion, 2010), dok je skup podataka SPSS Credit iz priručnog dela SPSS-a (SPSS za Windows 13.0, 2004).

Svaka baza slučajeva nad kojom je spovođeno istraživanje sadrži slučajeve kod kojih je broj mogućih izlaznih atributa dva, što je u skladu sa samom prirodom postupka odobravanja kredita, zahtev za kredit je ili odobren ili odbijen. Broj slučajeva u bazi, kao i broj indeksa, atributa koji opisuju slučaj i na osnovu kojih se radi pretraživanje, se razlikuje od baze do baze, ali sva tri skupa podataka sadrže i numeričke i kategoričke atribute.

Za sve tri baze slučajeva, za čuvanje tabele slučajeva odabrana je struktura tabele odlučivanja. Ova tabela se sastoji od atributa i redova, pri čemu su neki od atributa izlaznog karaktera, tj. kada se desi slučaj opisan neizlaznim (ulaznim) atributima, tada se donela odluka opisana u izlaznom atributu.

Struktura tabele odlučivanja je jednostavna za implementaciju, omogućava relativno brzo rešavanje problema, a sama izgradnja je moguća u softveru Microsoft Excel.
Tabele odlučivanja imaju različit broj ulaznih atributa, tačnije broj ulaznih atributa je 14 (Australian Credit dataset), 20 (German Credit dataset) ili 8 (SPSS Credit dataset), dok je broj izlaznih atributa jedan.

Tabela 6. Skupovi podataka.

Baza slučajeva	#Broj klasa	#Broj slučajeva	Kategorički atributi	Numerički atributi	Ukupan broj ulaznih atributa
Australian Credit	2	690	8	6	14
German Credit	2	1000	13	7	20
SPSS Credit	2	700	1	7	8

4.2. Rezultati istraživanja

Radi testiranja uspešnosti klasifikacije, u disertaciji će biti analizirano nekoliko različitih izvedenih modela ZOS-a, kao što su:

1. Tradicionalni ZOS (čist ZOS ili bazni model) – svi atributi su podjednake važnosti, funkcija Euklidove metrike se koristi za merenje sličnosti između numeričkih varijabli u slučajevima, dok se funkcija preklapanja koristi za kategoričke podatke.

2. Tradicionalni ZOS sa funkcijama preferencija optimizovanim GA za numeričke atribute (tzv. čist ZOS-PF-GA), u kome su svi atributi jednako ponderisani, tj. podjednako su značajni. Za merenje sličnosti između kategoričkih podataka koristi se funkcija preklapanja ili podudaranja.

3. Ponderisani bazni ZOS sa funkcijom Euklidove metrike i ponderisanim atributima. Za merenje sličnosti između kategoričkih podataka koristi se funkcija preklapanja.

4. Ponderisani ZOS-PF-GA sa funkcijama preferencija i ponderisanim atributima. Funkcija preklapanja se koristi za merenje sličnosti između kategoričkih podataka.

5. Ponderisani ZOS-PF-GA-GOODALL1 – za merenje sličnosti između kategoričkih podataka koristi se funkcija GOODALL1, dok se za numeričke varijable koriste funkcije preferencije. Atributi su ponderisani. Optimizacija se postiže korišćenjem GA.

6. Ponderisani ZOS-PF-GA-OF - za merenje sličnosti između kategoričkih podataka koristi se OF funkcija, a za numeričke varijable funkcije preferencije.

7. Ponderisani ZOS-PF-GA-ESKIN – za merenje sličnosti između kategoričkih podataka koristi se ESKIN funkcija, dok se za numeričke varijable koriste funkcije preferencije.

Kao kontrolni parametri pretrage GA u eksperimentima, biće korišćene sledeće vrednosti: 1000 organizama u populaciji, postavka ukrštanja na 0.5 i stope mutaciji na 0.1. Kao kriterijum zaustavljanja koristiće se vreme, i to period od 50 sati, s obzirom da je ustanovljeno da je to vremenski interval nakon koga rešenje postaje stabilno.

U svakom od navedenih ZOS modela, za pronalaženje slučajeva je korišćen k-NN metod. Svi predloženi modeli su testirani na nekoliko k vrednosti, sa variranjem od 1 do 9 na svakoj neparnoj vrednosti. Za svaki slučaj iz skupa slučajeva za testiranje, model treba da dodeli jednu od dve moguće klase, i to merenjem udaljenosti tog slučaja u odnosu na svaki slučaj iz skupa namenjenog za treniranje. Za procenu učinka prediktivnog modela, koristi se metodologija 10-ostruke unakrsne validacije (engl. 10-fold cross validation). Particije (fold) unakrsne validacije se generišu korišćenjem takozvanog semena za slučajni izbor podataka. Tačnosti 10-ostruke unakrsne validacije su prikazane u Tabeli 7, Tabeli 8. i Tabeli 9. Može se primetiti da kombinacija ZOS-PF-GA uvek daje bolje performanse nego tradicionalni ZOS model. Ovo se naročito odnosi na ponderisane modele.

Tabela 7. Poređenje performansi modela - Australian credit dataset

Model	Tačnost klasifikacije - Australian Credit dataset				
K	1	3	5	7	9
Čist ZOS (neponderisani bazni, tradicionalni model)	83.19%	87.68%	88.55%	88.84%	90.58%
Neponderisani ZOS-PF-GA	89.57%	90.00%	92.17%	91.59%	91.74%
Ponderisani tradicionalni ZOS	87.54%	88.26%	92.90%	92.75%	91.59%
Ponderisani ZOS-PF-GA	85.80%	91.45%	93.48%	93.77%	92.46%
Ponderisani ZOS-PF-GA- Goodall 1	88.12%	92.46%	93.77%	93.91%	94.06%
Ponderisani ZOS-PF-GA- OF	85.80%	93.33%	93.48%	93.48%	92.17%
Ponderisani ZOS-PF-GA –ESKIN	86.23%	92.90%	93.91%	94.78%	93.77%

Tabela 8. Poređenje performansi modela - German credit dataset

Model	Tačnost klasifikacije – German Credit dataset				
K	1	3	5	7	9
Čist ZOS (neponderisani bazni, tradicionalni model)	63.90%	61.10%	69.40%	68.20%	66.60%
Neponderisani ZOS-PF-GA	72.30%	73.30%	74.80%	76.70%	76.20%
Ponderisani tradicionalni ZOS	74.00%	66.00%	70.10%	70.80%	69.60%
Ponderisani ZOS-PF-GA	74.30%	73.90%	77.70%	80.30%	77.90%
Ponderisani ZOS-PF-GA- Goodall 1	66.00%	71.70%	75.10%	73.80%	70.80%
Ponderisani ZOS-PF-GA- OF	65.90%	72.20%	70.50%	73.90%	72.90%
Ponderisani ZOS-PF-GA –ESKIN	73.60%	72.30%	76.40%	77.90%	77.40%

Tabela 9. Poređenje performansi modela - SPSS credit dataset

Model	Tačnost klasifikacije – SPSS Credit dataset				
K	1	3	5	7	9
Čist ZOS (neponderisani bazni, tradicionalni model)	70.71%	69.14%	72.29%	72.86%	74.43%
Neponderisani ZOS-PF-GA	78.43%	74.71%	78.43%	77.57%	77.43%
Ponderisani tradicionalni ZOS	73.14%	73.86%	76.71%	78.57%	76.00%
Ponderisani ZOS-PF-GA	77.29%	75.14%	78.57%	78.57%	78.00%
Ponderisani ZOS-PF-GA- Goodall 1	78.00%	76.57%	79.86%	80.14%	77.57%
Ponderisani ZOS-PF-GA- OF	75.00%	75.86%	76.57%	78.29%	77.29%
Ponderisani ZOS-PF-GA –ESKIN	76.86%	75.71%	81.29%	78.43%	78.00%

Sledeći korak je bio da se pristupi testiranju - da li su otkrivene razlike u rezultatima modela statistički značajne. Wilcoxonov test ranga za znakom (Wilcoxon signed ranks test) se koristi da bi se ispitalo da li je učinak klasifikacije hibridnog pristupa znatno veći u odnosu na rezultate klasifikacije tradicionalnim modelom. Tabele 10.-12. pokazuju rezultate Wilcoxonovog testiranja sposobnosti klasifikacije različitih modela.

Za svaki skup podataka postoje vrednosti za k gde model ZOS-PF-GA ima značajno bolje rezultate od čistog ZOS modela na nivou od 5%. Ovo ukazuje da se razlike između rezultata modela u nekim slučajevima mogu smatrati statistički značajnim. Slična situacija je i sa ponderisanim modelima.

Tabela 10. Wilcoxonov test ranga za znakom - Australian credit dataset

Wilcoxonov test ranga za znakom - Australian credit dataset					
Neponderisani/ čist ZOS-PF-GA – Neponderisani/ čist ZOS					
K	1	3	5	7	9
Z	-2.84400	-2.82500	-2.87100	-2.81600	-2.41400
Asymp. Sig. (2-tailed)	**0.00400**	**0.00500**	**0.00400**	**0.00500**	**0.01600**
Ponderisani ZOS-PF-GA – Ponderisani bazni, tradicionalni ZOS					
K	1	3	5	7	9
Z	-1.40900	-2.82900	-0.94800	-2.03200	-1.51000
Asymp. Sig. (2-tailed)	0.15900	**0.00500**	0.34300	**0.04200**	0.13100
Ponderisani ZOS-PF-GA-GOODALL 1 – Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-0.59700	-2.82900	-1.08400	-2.41400	-2.67500
Asymp. Sig. (2-tailed)	0.55000	**0.00500**	0.27900	**0.01600**	**0.00700**
Ponderisani ZOS-PF-GA-OF – Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-1.23400	-2.81400	-0.73400	-1.50800	-1.07800
Asymp. Sig. (2-tailed)	0.21700	**0.00500**	0.46300	0.13200	0.28100
Ponderisani ZOS-PF-GA-ESKIN - Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-0.56100	-2.69200	-1.97700	-2.86900	-2.53600
Asymp. Sig. (2-tailed)	0.57500	**0.00700**	**0.04800**	**0.00400**	**0.01100**
Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-GOODALL 1					
K	1	3	5	7	9
Z	-2.37500	-1.80700	-1.41400	-0.17100	-2.23300
Asymp. Sig. (2-tailed)	**0.01800**	0.07100	0.15700	0.86500	**0.02600**

Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-OF					
K	1	3	5	7	9
Z	0.00000	-2.31900	-0.21300	-0.12100	-0.40600
Asymp. Sig. (2-tailed)	1.00000	**0.02000**	0.83100	0.90400	0.68400
Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-ESKIN					
K	1	3	5	7	9
Z	-0.05100	-2.31300	-0.63900	-2.28000	-1.60400
Asymp. Sig. (2-tailed)	0.95900	**0.02100**	0.52300	**0.02300**	0.10900

Tabela 11. Wilcoxonov test ranga za znakom - German credit dataset

Wilcoxonov test ranga za znakom - German Credit dataset					
Neponderisani/ čist ZOS-PF-GA – Neponderisani/ čist ZOS					
K	1	3	5	7	9
Z	-2.805	-2.142	-2.296	-2.655	-2.821
Asymp. Sig. (2-tailed)	**0.00500**	**0.03200**	**0.02200**	**0.00800**	**0.00500**
Ponderisani ZOS-PF-GA – Ponderisani bazni, tradicionalni ZOS					
K	1	3	5	7	9
Z	-0.172	-2.814	-2.807	-2.812	-2.807
Asymp. Sig. (2-tailed)	0.86300	**0.00500**	**0.00500**	**0.00500**	**0.00500**
Ponderisani ZOS-PF-GA-GOODALL 1 – Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-2.677	-2.812	-2.203	-1.379	-0.765
Asymp. Sig. (2-tailed)	**0.00700**	**0.00500**	**0.02800**	0.16800	0.44400
Ponderisani ZOS-PF-GA-OF – Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-2.816a	-2.814	-0.239	-1.958	-2.016
Asymp. Sig. (2-tailed)	**0.00500**	**0.00500**	0.81100	**0.05000**	**0.04400**

Ponderisani ZOS-PF-GA-ESKIN - Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-0.56600	-2.67000	-2.81000	-2.80700	-2.80700
Asymp. Sig. (2-tailed)	0.57200	**0.00800**	**0.00500**	**0.00500**	**0.00500**
Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-GOODALL 1					
K	1	3	5	7	9
Z	-2.812	-1.249	-1.786	-2.712	-2.705
Asymp. Sig. (2-tailed)	**0.00500**	0.21200	0.07400	**0.00700**	**0.00700**
Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-OF					
K	1	3	5	7	9
Z	-2.812	-1.134	-2.814	-2.809	-2.515
Asymp. Sig. (2-tailed)	**0.00500**	0.25700	**0.00500**	**0.00500**	**0.01200**
Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-ESKIN					
K	1	3	5	7	9
Z	-0.89300	-2.01000	-1.69100	-2.67100	-0.68200
Asymp. Sig. (2-tailed)	0.37200	**0.04400**	0.09100	**0.00800**	0.49500

Tabela 12. Wilcoxonov test ranga za znakom - SPSS credit dataset

Wilcoxonov test ranga za znakom - SPSS Credit dataset					
Neponderisani/ čist ZOS-PF-GA – Neponderisani/ čist ZOS					
K	1	3	5	7	9
Z	-2.668	-2.657	-2.668	-2.608	-1.865
Asymp. Sig. (2-tailed)	**0.00800**	**0.00800**	**0.00800**	**0.00900**	0.06200
Ponderisani ZOS-PF-GA – Ponderisani bazni, tradicionalni ZOS					
K	1	3	5	7	9
Z	-1.962	-0.307	-0.983	-0.102	-0.847
Asymp. Sig. (2-tailed)	**0.05000**	0.75900	0.32600	0.91900	0.39700
Ponderisani ZOS-PF-GA-GOODALL 1 – Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-2.143	-1.897	-1.856	-0.306	-0.419
Asymp. Sig. (2-tailed)	**0.03200**	0.05800	0.06300	0.75900	0.67500
Ponderisani ZOS-PF-GA-OF – Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-1.433	-2.043	-0.677	-0.773	-0.476
Asymp. Sig. (2-tailed)	0.15200	**0.04100**	0.49800	0.44000	0.63400
Ponderisani ZOS-PF-GA-ESKIN - Ponderisani bazni ZOS					
K	1	3	5	7	9
Z	-1.83800	-0.84300	-2.65500	-0.41500	-0.46500
Asymp. Sig. (2-tailed)	0.06600	0.39900	**0.00800**	0.67800	0.64200
Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-GOODALL 1					
K	1	3	5	7	9
Z	-1.807	-1.897	-2.209	-1.975	-0.520
Asymp. Sig. (2-tailed)	0.07100	0.05800	**0.02700**	**0.04800**	0.60300

Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-OF					
K	1	3	5	7	9
Z	-2.201	-0.769	-2.207	0.000	-1.282
Asymp. Sig. (2-tailed)	**0.02800**	0.44200	**0.02700**	1.00000	0.20000
Ponderisani ZOS-PF-GA – Ponderisani ZOS-PF-GA-ESKIN					
K	1	3	5	7	9
Z	-0.53100	-1.12700	-2.45600	-0.51400	0.00000
Asymp. Sig. (2-tailed)	0.59500	0.26000	**0.01400**	0.60700	1.00000

5. Zaključak

Suptilna umetnost odlučivanja se sastoji u tome da ne odlučujete o nevažnim stvarima, i o onome o čemu mogu da odluče drugi.

Chester Bernard

Nikada ne možete da imate sve informacije potrebne za odlučivanje. Ukoliko bi ih imali, odluke ne bi bile ni potrebne. Umesto njih imali bi očigledne činjenice.

David Mahoney

Mnogi ljudi gube vreme u donošenju odluka, a pritom stalno odlučuju isto, jer ne analiziraju rezultate prethodnih odluka.

Philip Marvin

5.1. Kritički osvrt na sprovedeno istraživanje

U želji da se unaprede performanse tradicionalnog sistema zaključivanja na osnovu slučajeva, u ovom radu je predložen novi model klasifikacije zasnovan na zaključivanju na osnovu slučajeva i inteligentnom izboru mera sličnosti.

Cilj je bio da se pronađe efikasan način izgradnje modela klasifikacije kupaca/ klijenata, koji će što preciznije da predvidi njihovo ponašanje. U ovom radu akcenat je bio na problemu ocene kreditne sposobnosti, odnosno na kreiranju modela kojima bi se prevashodno olakšao posao bankama. Prednosti primene ovih modela sa aspekta banaka mogu biti višestruke, naročito zbog toga što podrazumevaju davanje podrške procesu odlučivanja, zbog doslednosti i tačnosti, zbog uključivanja svih neophodnih faktora u proces odlučivanja, zbog smanjenja gubitka usled loših kredita, brzine,.... Modeli takođe donose određene prednosti i klijentima, koje se ogledaju u sledećem: laka procedura aplikacije za kredit, dobijanje odgovora u mnogo kraćem vremenskom intervalu, smanjenje potrebnih informacija za definisanje kreditne sposobnosti, manji troškovi obrade kreditnog zahteva. Često se klijenti koji nemaju dovoljno dugu kreditnu istoriju odbijaju, iako mogu biti dobri korisnici kredita. Upravo u tim slučajevima, modeli mogu

biti od pomoći da se po automatizmu ne odbijaju komitenti koji u potpunosti ne ispunjavaju predefinisane uslove.
Sve navedeno doprinosi boljim odnosima sa klijentima i njihovom zadržavanju.

Modeli predstavljeni u ovom radu podrazumevaju kombinaciju ZOS-a, funkcija teorije preferencija, pojedinih mera sličnosti za kategoričke atribute i pristupa genetskog algoritma.

Pošlo se od toga da funkcije preferencija pružaju više mogućnosti u izražavanju preferencija donosioca odluke, odnosno više mogućnosti da se iskoristi domensko znanje, i to na takav način da se poboljša proces pronalaženja najkorisnijih slučajeva iz baze.
Prilikom sprovođenja istraživanja, korišćen je peti tip funkcija preferencije jer je zaključeno da je ovaj tip funkcije najviše koristan u smislu da pokriva najveći broj situacija. Eksperimentalni rezultati jasno pokazuju da predloženi model može po prediktivnoj moći prevažići tradicionalni model ZOS-a.

Dodatno, pretpostavilo se i da će se korišćenjem određenih mera sličnosti za kategoričke atribute, u kombinaciji sa korišćenjem funkcija preferencija za numeričke atribute, doći do modela boljih prediktivnih performansi od tradicionalnog modela ZOS-a.

Izračunavanje sličnosti između kategoričkih atributa se analiziralo u različitim kontekstima i većem broju naučnih radova.
U ovom radu je korišćeno nekoliko takvih mera, i to onih koje se smatraju osnovnim, a kako bi se postigla što veća tačnost klasifikacije klijenata koji apliciraju za kredit. Eksperimentalni rezultati su pokazali da nijedna mera nije najefikasnija, odnosno nijedna nije dominantna u odnosu na druge kod svih korišćenih baza podataka. Isto tako, pokazalo se da neke mere mogu da imaju stalne visoke učinke. Potrebno je razumeti kako mera sličnosti zavisi od različitih karakteristika skupa kategoričkih podataka, i to je nešto što će tek biti tema budućih istraživanja.

U ovom radu je razmatrano ponašanje modela sa pojedinačnim merama sličnosti, odnosno za posmatrani model je korišćena po jedna mera sličnosti za svaki od kategoričkih atributa. Pošto različiti atributi u skupu podataka mogu biti različitog karaktera, alternativni način je da se koristite različite mere za različite atribute. Ovo posebno može izgledati obećavajuće s obzirom na komplementarnu prirodu nekoliko mera sličnosti.

GA je korišćen za optimizaciju parametara funkcije preferencija, ali i za optimizaciju značaja (težina) atributa.

5.2. Budući pravci istraživanja

U disertaciji se preporučuju novi ZOS modeli sa željom da se unapredi učinak tradicionalnog ZOS sistema. Novi model predstavlja kombinaciju ZOS-a, funkcije teorije preferencije, GA pristupa i različitih mera sličnosti za kategoričke podatke.
Budući da funkcije preferencije pružaju više mogućnosti u izražavanju preferencija donosilaca odluka, one se mogu koristiti za unapređivanje procesa pronalaženja najsličnijih slučajeva iz prošlosti. GA se mogu koristiti radi optimizacije parametara funkcija preferencije, ali takođe i za optimizaciju važnosti atributa. Dosadašnji rezultati eksperimenta jasno pokazuju da predloženi modeli mogu da nadmaše tradicionalni ZOS model. Tehnike koje koriste različite mere za određivanje sličnosti između kategoričkih podataka pokazale su da nijedna mera nije dominantna u odnosu na druge kod svih tipova problema, ali neke mere mogu da imaju stalne visoke učinke.

Sa druge strane, ovo istraživanje ima svoja ograničenja. Selekcija atributa i selekcija slučajeva su faktori koji mogu biti optimizovani u ZOS sistemu. Izbor parametra k, koji predstavlja broj slučajeva najsličnijih novom slučaju, može se takođe koristiti kao parameter za optimizaciju. U disertaciji, predloženi model se procenjuje za ograničen broj različitih vrednosti k, ali je takođe moguće da neke druge vrednosti k mogu poboljšati ukupni učinak ZOS sistema. Naposletku, u budućnosti bi trebalo dodatno testirati uopštavanje predloženog modela njegovom primenom na probleme koji imaju veći broj klasa, i generalno na probleme u drugim oblastima.

6. Literatura

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approach. AI Communications, 7 (1), pp. 39-59.

Ahn, H. & Kim, K.-J. (2009). Global optimization of case-based reasoning for breast cytology diagnosis. Expert Systems with Applications, 36 (1), pp. 724 -734.

Ahn, H., & Kim, K.-J. (2008). Using genetic algorithms to optimize nearest neighbors for data mining. Annals of Operations Research, 263 (1), pp. 5-18.

Ahn, H., Kim, K.-J., & Han, I. (2007). A case-based reasoning system with the two-dimensional reduction technique for customer classification. Expert Systems with Applications, 32 (4), pp. 1011-1019.

Akhavein, J. D., Frame, W. S., & White, L. J. (2005). The diffusion of financial innovations: An examination of the adoption of small business credit scoring by large banking organisations. The Journal of Business, 78 (2), pp. 577-596.

Allen, B.P. (1994). Case-based reasoning: business applications, Communications of the ACM, 37 (3), pp. 40-42.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, 23 (4), pp. 589-609.

Banasik, J., Crook, J., & Thomas, L. (2003). Sample selection bias in credit scoring models. Journal of the Operational Research Society, 54 (8), pp. 822-832.

Bobrowski L. (2012.) Class Separating Measures of Similarity for the CBR Scheme. Industrial Conference on Data Mining - Workshops 2012: pp. 5-18.

Boriah, S., Chandola, V., & Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. In Proceedings of the eighth SIAM International Conference on Data Mining. pp. 243–254.

Boritz, J. E., & Kennedy, D. B. (1995). Effectiveness of neural network types for prediction of business failure. Expert Systems with Applications, 9 (4), pp. 503-512.

Brans, J. P., & Vincke, Ph. (1985). A preference ranking organization method: the promethee method for multiple criteria decision-making. Management Science, 31 (6), pp. 647-656.

Brans, J. P., Vincke, Ph., & Mareschal, B. (1986). How to select and how to rank projects: The Promethee method. European Journal of Operational Research, 24, pp. 228-238.

Bryant, S. M. (1997). A case-based reasoning approach to bankruptcy prediction modeling. International Journal of Intelligent Systems in Accounting, Finance and Management, 6 (3), pp. 195-214.

Buta, P. (1994). Mining for financial knowledge with CBR. AI Expert, 9 (2), pp. 34-41.

Chandola V., Boriah S., & V. Kumar V. (2009). A framework for exploring categorical data. In Proceedings of the SIAM International Conference on Data Mining, pp 187–198.

Changchien, S. W., & Lin, M. C. (2005). Design and implementation of a case-based reasoning system for marketing plans. Expert Systems with Applications, 28 (1), pp. 43–53.

Chen, M. C., & Huang, S. H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. Expert Systems with Applications, 24, pp. 433-441.

Chen, Y.-K., Wang, C.-Y., & Feng, Y.-Y. (2010). Application of a 3NN+1 based CBR system to segmentation of the notebook computers market. Expert Systems with Applications, 37 (1), pp. 276-281.

Chiu, C. (2002). A case-based customer classification approach for direct marketing. Expert Systems with Applications, 22 (2), pp. 163-168.

Chiu, C., Chang, P. C., & Chiu, N. H. (2003). A case-based expert support system for due-date assignment in a water fabrication factory. Journal of Intelligent Manufacturing, 14 (3-4), pp. 287–296.

Chun, S. H., & Park, Y. J. (2006). A new hybrid data mining technique using a regression case based reasoning: Application to financial forecasting. Expert Systems with Applications, 31 (2), pp. 329-336.

Chye, K. H., Chin, T. W., & Peng, G. C. (2004). Credit scoring using data mining techniques. Singapore Management Review, 26 (2), pp. 25-47.

Coakley, J. R., & Brown, C. E. (2000). Artificial neural networks in accounting and finance: Modeling issues. International Journal of Intelligent Systems in Accounting, Finance and Management, 9 (2), pp. 119-144.

Delibašić B. (2004). Projektovanje i implementacija sistema menadžmenta znanja, magistarska teza, FON, Beograd.

Delibašić B. (2007). Formalizacija procesa poslovnog odlučivanja preko paterna, doktorska disertacija, FON, Beograd.

Desai, V. S., Convay, D. G., Crook, J. N., & Overstreet, G. A. (1997). Credit scoring models in the credit union environment using neural networks and genetic algorithms. IMA Journal of Mathematics Applied in Business and Industry, 8, pp. 323-346.

Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. European Journal of Operational Research, 95 (1), pp. 24-37.

Dimitras, A. I., Zanakis, S. H., & Zopounidis, C. (1996). A survey of business failure with an emphasis on prediction methods and industrial applications. European Journal of Operational Research, 90 (3), pp. 487-513.

Elliott, R., & Filinkov, A. (2008). A self tuning model for risk estimation. Expert Systems with Application, 34 (3), pp. 1692-1697.

Frank, A., & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Fu, Y., & Shen, R. (2004). GA based CBR approach in Q&A system. Expert Systems with Applications, 26 (2), pp. 167-170.

Han, I., Chandler, J. S., & Liang, T. P. (1996). The impact of measurement scale and correlation structure on classification performance of inductive learning and statistical methods. Expert Systems with Applications, 10 (2), pp. 209-221.

Han, J., & Kamber, M. (2001). Datamining: Concepts and techniques. San, Francisco, CA: Morgan Kaufmann Publishers.

Hsu, C. I., Chiu, C., & Hsu, P. L. (2004). Predicting information systems outsourcing success using a hierarchical design of case-based reasoning. Expert Systems with Applications, 26 (3), pp. 435-441.

Huang, C., Chen, M., & Wang, C. (2007). Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications, 33 (4), pp. 847-856.

Huang, J., Tzeng, G., & Ong, C. (2006). Two-stage genetic programming (2SGP) for the credit scoring model. Applied Mathematics and Computation, 174 (2), pp. 1039-1053.

Im, K. H., & Park, S. C. (2007). Case-based reasoning and neural network based expert system for personalization. Expert Systems with Applications, 32 (1), pp. 77-85.

Jo, H., & Han, I. (1996). Integration of case-based forecasting, neural network, and discriminant analysis for bankruptcy prediction. Expert Systems with Applications, 11 (4), pp. 415-422.

Kibler, D., & Aha, D.W., (1987). Learning representative exemplars of concepts: An initial case study. Proceedings of the Fourth International Workshop on Machine Learning. Irvine, CA: Morgan Kaufmann, pp. 24-30.

Kim, K. (2004). Toward global optimization of case-based reasoning systems for financial forecasting. Applied Intelligence, 21 (3), pp. 239-249.

Kim, K., & Han, I. (2001). Maintaining case-based reasoning systems using a genetic algorithms approach. Expert Systems with Applications, 21 (3), pp. 139-145.

Kolodner, J. L. (1992). An Introduction to Case-Based Reasoning. Artificial Intelligence Review 6, pp. 3-34.

Kolodner, J. L. (1993). Case-based reasoning. San Mateo, CA: Morgan.

Kolodner, J. L., & Mark, W. (1992). Case-based reasoning. IEEE Expert, 7, pp. 5-6.

Lee, T., & Chen, I. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. Expert Systems with Applications, 28 (4), pp. 743-752.

Lee, T., Chiu, C., Lu, C., & Chen, I. (2002). Credit scoring using the hybrid neural discriminant technique. Expert Systems with Applications, 23 (3), pp. 245-254.

Lee, Y.-C. (2007). Application of support vector machines to corporate credit rating prediction. Expert Systems with Applications, 33, pp. 67–74.

Leung, K., Cheong, F. and Cheong, C. (2007). Consumer credit scoring using an artificial immune system algorithm. Proceedings of the IEEE International Conference on Evolutionary Computation (CEC 2007), pp. 3377-3384, IEEE Press.

Li, H., & Sun, J. (2010). Business failure prediction using hybrid2 case-based reasoning (H2CBR). Computers and Operations Research, 37 (1), pp.137-151.

Li, H., Sun, J., & Sun, B.-L. (2009). Financial distress prediction based on OR-CBR in the principle of k-nearest neighbors. Expert Systems with Applications, 36 (1), pp. 643-659.

Liao, T. W., Zhang, Z., & Mount, C. R. (1998). Similarity measures for retrieval in case-based reasoning systems. Applied Artificial Intelligence, 12, pp. 267-288.

Lopez de Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. L., Cox, M. T., Forbus, K., Keane, M., Aamodt, A., & Watson, I. (2005), Retrieval, reuse, revision, and retention in case-based reasoning. Knowledge Engineering Review, 20 (3), pp. 215-240.

Mareschal, B. (1986). Stochastic PROMETHEE multiple criteria decision making under uncertainty. European Journal of Operational Research, 26, pp. 58-64.

Mareschal, B. (1988). Weight stability intervals in the PROMETHEE multicriteria decision aid method. European Journal of Operational Research, 33, pp. 54-64.

Mareschal, B., & Brans, J. P. (1988). Geometrical representations for MCDM (GAIA). European Journal of Operational Research, 34, pp. 69-77.

Martin, D. (1977). Early warning of bank failure: A logit regression approach. Journal of Banking and Finance, 1 (3), pp. 249-276.

Meyer, P. A., & Pifer, H. (1970). Prediction of bank failures. The Journal of Finance, 25, pp. 853-868.

Min, S.H., Lee, J., & Han, I. (2006). Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert Systems with Applications, 31, pp. 652–660.

Montazemi, A. R., & Gupta, K. M. (1997). A framework for retrieval in case-based reasoning systems. Annals of Operations Research, 72 , 51-73.

Park, C. S., & Han, I. (2002). A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction. Expert Systems with Applications, 23 (3), pp. 255-264.

Parreiras, R. O., & Vasconcelos, J. A. (2007). A multiplicative version of Promethee II applied to multiobjective optimization problems. European Journal of Operational Research, 183, pp. 729-740.

Podvezko, V., & Podviezko, A. (2010). Dependence of multi-criteria evaluation result on choice of preference functions and their parameters. Technological and Economic Development of Economy, 16 (1), pp. 143-158.

Radojevic, A., Petrovic, S. & Radojevic, D. (1997). A Fuzzy Approach to Preference Structure in Multicriteria Ranking. International Transactions in Operational Research, 4, pp. 419-430.

Reinartz, T., Iglezakis, I., & Roth-Berghofer, T. (2001). Review and restore for case-based maintenance. Computational Intelligence, 17 (2), pp. 214-234.

Schank, R. C., & Abelson, R. P. (1977). Scripts, plans, goals and understanding. Hillsdale, NJ: Lawrence Erlbaum.

Shaw, M., & Gentry, J. (1998). Using an expert system with inductive learning to evaluate business loans. Financial Management, 17 (3), pp. 45-56.

Shin, K. S., & Han, I. (2001). A case-based approach using inductive indexing for corporate bond rating. Decision Support Systems, 32 (1), pp. 41-52.

Shin, K.-S., & Han, I. (1999). Case-based reasoning supported by genetic algorithms for corporate bond rating. Expert Systems with Applications, 16 (2), pp. 85-95.

Slade, S. (1991). Case-based reasoning: A research paradigm. AI Magazine, 12 (1), pp. 42-55.

SPSS for Windows, Rel. 13.0. 2004. Chicago: SPSS Inc.

Stanimirović Z. (2004). Rešavanje nekih diskretnih lokacijskih problema primenom genetskih algoritama, *magistarska teza*, Matematički fakultet, Beograd

Suknović M., Delibašić B. (2010). Poslovna inteligencija i sistemi za podršku odlučivanju, Fakultet organizacionih nauka, Beograd

Thomas, L. C. (2000). A survey of credit and behavioral scoring - forecasting financial risk of lending to consumers. International Journal of Forecasting, 16 (2), pp. 149-172.

Tseng, H. E., Chang, C. C., & Chang, S. H. (2005). Applying case-based reasoning for product configuration in mass customization environments. Expert Systems with Applications, 29 (4), pp. 913-925.

Vukovic S., Delibasic B., Uzelac A. & Suknovic M. (2012). A case-based reasoning model that uses preference theory functions for credit scoring, Expert Systems with Applications, 39 (9), pp. 8389-8395.

Wang, Y., & Ishii, N. (1997). A method of similarity metrics for structured representations. Expert Systems with Applications, 12 (1), pp. 89-100.

Watson, I., & Farhi, M. (1994). Case-based reasoning: A review. The Knowledge Engineering Review, 9, pp. 327-354.

Watson, I., *Applying Knowledge Management – Techniques for Building Corporate Memories*, Morgan Kaufmann Publishers, 2003.

West, D. (2000). Neural network credit scoring models. Computers & Operations Research, 27 (11-12), pp. 1131-1152.

Wettschreck, D., & Aha, D.W. (1995). Weighting features. *In Proceedings of the First International Conference on Case-Based Reasoning (ICCBR-95).*

Wheeler, R., & Aitken, S. (2000). Multiple algorithms for fraud detection. Knowledge-Based Systems, 13 (2), pp. 93-99.

Wilson D. R. & T. R. Martinez (1997). Improved heterogeneous distance functions. Journal of Artificial Intelligence Research (6), pp. 1-34.

Yang L. (2002). A framework of data mining application process for credit scoring. Arbeitsbericht, No. 01.

Yobas, M. B., Crook, J.N., & Ross, P. (2000). Credit scoring using neural and evolutionary techniques. IMA Journal of Mathematics Applied in Business & Industry, 11 (2), pp. 111-125.

Zakrzewska, D. (2007). On integrating unsupervised and supervised classification for credit risk evaluation. Information technology and control, 36, pp. 98-102.

Zhang, G., Hu, M. Y., Patuwo, B. E.,& Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: General framework and crossvalidation analysis. European Journal of Operational Research, 116 (1), pp. 16-32.

Biografija

Sanja Vuković je rođena u Beogradu, 25.09.1982. godine. Diplomirala je u februaru 2006. godine, na Fakultetu organizacionih nauka u Beogradu, na odseku za Menadžment, sa opštim uspehom 9,60 u toku studija i ocenom 10 na diplomskom ispitu. Time je stekla zvanje diplomiranog inženjera organizacionih nauka – odsek za menadžment. Tema diplomskog rada je bila „Mogućnosti upravljanja portfoliom hartija od vrednosti“, mentor dr Vesna Bogojević Arsić. Sa navedenim diplomskim radom je učestvovala na Konkursu za najbolji diplomski studentski rad na temu finansijskih tržišta u 2006., organizovanim od strane brokerske kuće Senzal, i osvojila treću nagradu.

Nakon završenih osnovnih studija, u oktobru 2006. upisuje master studije, takođe na Fakultetu organizacionih nauka u Beogradu. Master studije završava u martu 2008. sa opštim uspehom 10,00 u toku studija i ocenom 10 na master ispitu. Tema master rada je bila „Mogućnosti primene strategija portfolio menadžmenta“, mentor dr Vesna Bogojević Arsić. Time je stekla zvanje master inženjera organizacionih nauka.

Doktorske studije na Fakultetu organizacionih nauka u Beogradu je upisala 2008. godine, smer Menadžment. U toku studija položila je predviđenih 9 ispita sa prosečnom ocenom 10,00. Lista položenih ispita: Nauka o menadžmentu; Menadžment e-poslovanja; Elektronsko poslovanje; Marketing informacioni sistem; Sistemi za podršku odlučivanju; Odlučivanje - izabrana poglavlja; Upravljanje lancima snabdevanja, Poslovna inteligencija - izabrana poglavlja; Sistem menadžmenta životnom sredinom.

Radno iskustvo započela je u aprilu 2006. godine u Meridian banci (Credit Agricole Group ad) na poziciji menadžer filijale - pripravnik. U julu 2007. godine prelazi u Raiffeisen banku, gde i danas radi. Od 2007. do danas radila je u nekoliko odeljenja, počev od pozicije Saradnika za koordinaciju i analizu prodaje Sektora za poslove sa malim preduzećima i preduzetnicima, preko pozicije Specijaliste za razvoj proizvoda Odeljenja za upravljanje tokovima novca u Sektoru za finansiranje pravnih lica, da bi u novembru 2012. prešla na poziciju Višeg saradnika za razvoj proizvoda malim preduzećima i preduzetnicima.

Pohađala je niz obuka organizovanih od strane Raiffeisen banke (Cash Management and Payments, Osnove ekonometrije, Data mining: otkrivanje znanja iz baza podataka u

bankarstvu, Osobna učinkovitost, Engleski jezik), kao i Meridian banke (uglavnom kursevi vezani za pospešivanje prodaje u bankarstvu). Pored obuka, bila je učesnik i većeg broja radionica, namenjenih razvoju bankarskih proizvoda.
Tečno govori, čita i piše engleski jezik. Poseduje pasivno znanje francuskog jezika.

Učešće na konferencijama:
1. 12th Industrial Conference on Data Mining ICDM'2012, July 13-20, 2012, Berlin/Germany
2. XXXVI Simpozijum o operacionim istraživanjivma, SYM-OP-IS 2009, Septembar 22-25, 2009, Ivanjica/Srbija

Spisak objavljenih radova

1. Vukovic S., Delibasic B., Uzelac A., Suknovic M. (2012). A case-based reasoning model that uses preference theory functions for credit scoring, Expert Systems with Applications, 39 (9), pp. 8389-8395
2. Vukovic S., Delibasic B., Suknovic M. (2012)., How do various categorical similarity measures influence the CBR credit scoring model?, Advances in Data Mining, Workshop Proceedings, 12th Industrial Conference on Data Mining ICDM 2012, Berlin, Germany, pp. 19-30
3. Gligorić N., Uzelac A., Vuković S. (2011). Uticaj ERP sistema na upravljanje lancima snabdevanja. Singidunum revija, 8 (2), str. 168-172
4. Uzelac A., Zoranović D., Gligorić N., Vučetić M., Vuković S. (2011). Unapređenje zdravstvenog sistema zemalja u razvoju primenom mobilnih tehnologija, Arhiv za tehničke nauke, 5(1), str. 63-70
5. Vuković S., Vujošević M. (2009). Modeli optimizacije portfolia hartija od vrednosti, XXXVI Simpozijum o operacionim istraživanjivma, SYM-OP-IS 2009, Zbornik radova, str. 745-748

Prilog 1.

Izjava o autorstvu

Potpisana Sanja P. Vuković, broj indeksa 15/08

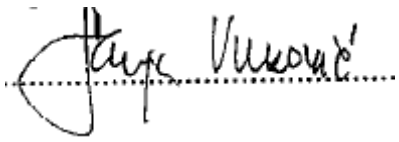
Izjavljujem

da je doktorska disertacija pod naslovom

Model poslovne inteligencije zasnovan na zaključivanju na osnovu slučajeva i izboru mera sličnosti

- rezultat sopstvenog istraživačkog rada,
- da predložena disertacija u celini ni u delovima nije bila predložena za dobijanje bilo koje diplome prema studijskim programima drugih visokoškolskih ustanova,
- da su rezultati korektno navedeni i
- da nisam kršila autorska prava i koristila intelektualnu svojinu drugih lica.

U Beogradu, 15.08.2013. **Potpis doktoranta**



Prilog 2.

Izjava o istovetnosti štampane i elektronske verzije doktorskog rada

Ime i prezime autora: Sanja P. Vuković

Broj indeksa: 15/08

Studijski program: Menadžment

Naslov rada: Model poslovne inteligencije zasnovan na zaključivanju na osnovu slučajeva i izboru mera sličnosti

Mentor: Prof. dr Boris Delibašić

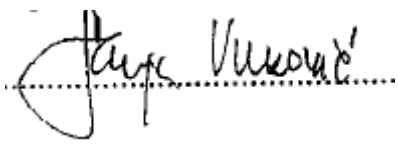
Potpisana Sanja Vuković

Izjavljujem da je štampana verzija mog doktorskog rada istovetna elektronskoj verziji koju sam predala za objavljivanje na portalu **Digitalnog repozitorijuma Univerziteta u Beogradu.**

Dozvoljavam da se objave moji lični podaci vezani za dobijanje akademskog zvanja doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada.

Ovi lični podaci mogu se objaviti na mrežnim stranicama digitalne biblioteke, u elektronskom katalogu i u publikacijama Univerziteta u Beogradu.

U Beogradu, 15.08.2013. **Potpis doktoranta**



Prilog 3.

Izjava o korišćenju

Ovlašćujem Univerzitetsku biblioteku „Svetozar Marković“ da u Digitalni repozitorijum Univerziteta u Beogradu unese moju doktorsku disertaciju pod naslovom:

Model poslovne inteligencije zasnovan na zaključivanju na osnovu slučajeva i izboru mera sličnosti

koja je moje autorsko delo.

Disertaciju sa svim prilozima predala sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju pohranjenu u Digitalni repozitorijum Univerziteta u Beogradu mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučila.

1. Autorstvo
2. Autorstvo - nekomercijalno
3. Autorstvo – nekomercijalno – bez prerade
4. Autorstvo – nekomercijalno – deliti pod istim uslovima
5. Autorstvo – bez prerade
6. Autorstvo – deliti pod istim uslovima

U Beogradu, 15.08.2013. **Potpis doktoranta**