



Univerzitet „Union –Nikola Tesla” u Beogradu
Fakultet za informacione tehnologije i inženjerstvo u Beogradu

Ivica B. Stanković

Primena modela inteligentnih agenata u procesu donošenja strategijskih odluka

Doktorska disertacija

Beograd, 2019



University "Union –Nikola Tesla", Belgrade
Faculty of Information Technology and Engineering, Belgrade

Ivica B. Stanković

Implementation of a model of intelligent agents for strategic decisions making processes

Doctoral dissertation

Belgrade, 2019

Mentor:

prof. dr Radoje Cvejić, redovni profesor
Fakultet za informacione tehnologije i inženjerstvo, Beograd
Univerzitet „UNION – Nikola Tesla”, Beograd

Datum odbrane: 07.05.2019. godine

Zahvaljujem svima koji su mi pomogli u toku izrade doktorske disertacije, pre svega, prof. dr Radomiru Mihajloviću, mentoru prof. dr Radoju Cvejiću, emeritusu prof. dr Životi Radosavljeviću, prof. dr Vladimiru Brusiću, prof. dr Petru Kočoviću, prof. dr Mileni Kukrić.

Veliku zahvalnost za razumevanje, pomoć i podršku tokom rada na disertaciji dugujem supruzi, porodici, kolegama i prijateljima.

Primena modela inteligentnih agenata u procesu donošenja strategijskih odluka

Sažetak

Savremeno poslovno okruženje je dinamično, konkurentno i globalno-orijentisano. Poslovno odlučivanje predstavlja složen proces koji u sebe uključuje analizu velikog broja podataka što za posledicu ima da je menadžmentu poslovnog sistema, često, teško da efikasno i pravovremeno donese kvalitetnu strategijsku odluku. Podaci, čijom obradom će donosioc odluke ekstrahovati informaciju, moraju biti: pouzdani, tačni, relevantni, sažeti, pravovremeni i u obliku razumljivom donosiocu odluke.

Pomoć u savremenom poslovanju mogu da pruže mašine sa određenom autonomijom u procesu donošenja odluke. Inteligentni agenti predstavljaju efikasan alat koji može pomoći u procesu donošenja strategijske odluke.

U okviru doktorske disertacije biće izvršen eksperiment u kome će biti korišćeni inteligentni agenti kao alati koji mogu da na osnovu obrade velike količine prikupljenih nestruktuiranih podataka i analize istih, daju relevantnu podlogu za donošenje strategijske odluke. Inteligentni agenti bi na taj način trebalo da poboljšaju kvalitet donete strategijske odluke od strane menadžmenta, a time i poslovnu uspešnost.

Date činjenice daju osnovu da u doktorskoj disertaciji, sa naučnog stanovišta, budu proučeni problemi koji se javljaju u toku procesa kreiranja i primene inteligentnih mašina za strateško odlučivanje.

Ključne reči: veštačka inteligencija, neuronske mreže, pajton, predviđanje ponašanja, finansijska pismenost dece

Implementation of a model of intelligent agents for strategic decisions making processes

Summary

The modern business environment is dynamic, competitive and global-oriented. Business decision-making is a complex process that involves analyzing a large number of data, which results in the management of the business system. This process is often difficult. The data from which the decision maker extracts the information must be: reliable, accurate, relevant, concise, timely and in the form of an understandable decision-maker.

Help in modern business can be provided by machines with a certain autonomy in the decision-making process. Intelligent agents are an effective tool that can help in the process of making a strategic decision.

In this doctoral dissertation, an experiment will be conducted in which intelligent agents will be used as tools that can, based on processing a large amount of collected unstructured data and analyzes, provide a relevant basis for making a strategic decision. Intelligent agents should in that way improve the quality of the strategic decision made by the management, and thus to give better business success.

These facts give a basis in the doctoral dissertation, from a scientific point of view, to study problems that arise during the process of creating and applying intelligent machines for strategic decision making.

Key words: artificial intelligence, neural networks, python, behavior prediction, financial literacy of children

Sadržaj

1 Predmet i ciljevi istraživanja	17
1.1 Glavni cilj	17
1.2 Naučni ciljevi	17
1.3 Ciljevi eksperimenta	18
1.4 Značaj i aktuelnost istraživanja (naučni i društveni)	19
1.5 Metodologija istraživanja, hipoteze i faze realizacije eksperimenta	20
1.6 Očekivani naučni doprinos istraživanja	22
1.7 Sažet opis sadržaja po poglavljima	23
2 Strategijsko odlučivanje u poslovnim sistemima	25
3 Tehnike pretraživanja i kompresije velike količine podataka	27
3.1 Primena Data Mining-a i Big Data u praksi	29
4 Veštačka inteligencija.....	32
4.1 Pojam veštačke inteligencije.....	32
4.2 Periodi razvoja veštačke inteligencije.....	36
5 Neuronske mreže	48
5.1 Pojam neurona	48
5.2 Perceptron	51
5.3 Sigmoid.....	56
5.4 Osobine neuronskih mreža.....	63
5.5.1 Rekurentni modeli neuronskih mreža	72
5.5.2 Podela neuronskih mreža u zavisnosti od metoda učenja	73
5.5.3 Modularne neuronske mreže	75
5.5.4 Neuronske mreže sa osnovnim radijalnim funkcijama	77
5.5.5 Primena evolutivnih algoritama u neuronskim mrežama.....	86
5.6 Neuronske mreže i vremenske serije	89
5.6.1 Učenje neuronskih mreža na osnovu postojećih vremenskih serija.....	95
5.7 Podela neuronskih mreža u zavisnosti od načina njihove implementacije	95
5.7.1 Implementacija neuronskih mreža korišćenjem koda napisanog u pajtonu ...	97
6 Eksperimentalni deo rada	127
6.1 Uvod.....	127
6.2 Pokretanje koda.....	128

6.3 Anketa.....	129
6.4 Analiza opštih anonimnih podataka o ispitanicima	129
6.5 Analiza prvog segmenta pitanja.....	140
6.5.1 Povezivanje pitanja prvog segmenta u jednu celinu	145
6.5.2 Povezivanje pitanja prvog segmenta i pitanja opšteg tipa u jednu celinu	148
6.6 Analiza drugog segmenta pitanja.....	153
6.6.1 Povezivanje pitanja drugog segmenta u jednu celinu	159
6.6.2 Povezivanje pitanja drugog segmenta i pitanja opšteg tipa u jednu celinu	161
6.7 Analiza trećeg segmenta pitanja	165
6.7.1 Povezivanje pitanja trećeg segmenta u jednu celinu.....	170
6.7.2 Povezivanje pitanja trećeg segmenta i pitanja opšteg tipa u jednu celinu ...	172
6.8 Analiza četvrtog segmenta pitanja	176
6.8.1 Povezivanje pitanja četvrtog segmenta u jednu celinu.....	182
6.8.2 Povezivanje pitanja četvrtog segmenta i pitanja opšteg tipa u jednu celinu	183
6.9 Analiza petog segmenta pitanja	188
6.9.1 Povezivanje pitanja petog segmenta u jednu celinu.....	192
6.10 Grupisanje pitanja iz segmenata u klastere.....	199
6.11 Predviđanje	201
7. Zaključna razmatranja	203
Literatura	208
Prilog A	215
Prilog B.....	232
Prilog C.....	233
Biografija autora.....	237

Spisak tabela

- Tabela 5.2.1. Paralela između biološkog i veštačkog neurona
- Tabela 5.2.2. Koeficijenti težine
- Tabela 5.2.3. Odgovori studenta
- Tabela 5.2.4. Rezultati ankete
- Tabela 5.5.1. Ulazi u dati neuron i izlaz iz datog neurona
- Tabela 5.5.4.1. Tablica istinitosti za I (AND) logičko kolo
- Tabela 5.5.4.2. Tablica istinitosti za ILI (OR) logičko kolo
- Tabela 5.5.3.3. Tablica istinitosti za NI (NAND) logičko kolo
- Tabela 5.5.4.4. Tablica istinitosti za XILI (XOR) logičko kolo
- Tabela 5.5.4.4. Tablica istinitosti za XILI (XOR) logičko kolo
- Tabela 6.4.1. Udeo ispitanika u zavisnosti od pola ispitanika.
- Tabela 6.4.2. Nivo obrazovanja majke ispitanika
- Tabela 6.4.3. Nivo obrazovanja oca ispitanika
- Tabela 6.4.4. Međusobna zavisnost opštih podataka od pola ispitanika
- Tabela 6.5.1. Udeo odgovara na Pitanje 1, Segment 1
- Tabela 6.5.2. Udeo odgovara na Pitanje 2, Segment 1
- Tabela 6.5.3. Udeo odgovara na Pitanje 3, Segment 1
- Tabela 6.5.2.1. Opšti podaci za ispitanike sa prosečnom ocenom pet iz matematike
- Tabela 6.6.1. Udeo odgovara na Pitanje 1, Segment 2
- Tabela 6.6.2. Udeo odgovara na Pitanje 2, Segment 2
- Tabela 6.6.3. Udeo odgovara na Pitanje 3, Segment 1
- Tabela 6.7.1. Udeo odgovara na Pitanje 1, Segment 3
- Tabela 6.7.2. Udeo odgovara na Pitanje 2, Segment 3
- Tabela 6.7.3. Udeo odgovara na Pitanje 3, Segment 3
- Tabela 6.8.1. Udeo odgovara na Pitanje 1, Segment 4
- Tabela 6.8.2. Udeo odgovara na Pitanje 2, Segment 4
- Tabela 6.8.3. Udeo odgovara na Pitanje 3, Segment 4
- Tabela 6.9.1. Udeo odgovara na Pitanje 1, Segment 5
- Tabela 6.9.2. Udeo odgovara na Pitanje 2, Segment 5
- Tabela 6.9.3. Udeo odgovara na Pitanje 3, Segment 5

Spisak slika

- Slika 5.1.1. Izgled i povezanost neurona u ljudskom mozgu (izvor: [34])
- Slika 5.2.1. Jednoslojni perceptron
- Slika 5.2.1. Višeslojna mreža perceptrona
- Slika 5.3.1. Sigmoid
- Slika 5.3.2. Pokretanje koda preko CMD shell-a Windows OS-a
- Slika 5.3.3. Sigmoidna funkcija
- Slika 5.3.4. Pokretanje koda preko CMD shell-a Windows OS-a
- Slika 5.3.5. Hevisajdova step funkcija
- Slika 5.5.1. Jednoslojna veštačka neuronska mreža
- Slika 5.5.2. Ulaz kod Delta učenja (izvor: autor)
- Slika 5.5.3. Funkcija greške u zavisnosti od razlike između željene i stvarne vrednosti izlaza
- Slika 5.5.4. Delta učenje
- Slika 5.5.5. Višeslojna veštačka neuronska mreža
- Slika 5.5.1.1. Princip rada rekurentne neuronske mreže
- Slika 5.5.2.1. Kohonenova ili samoorganizujuća mapa (izvor: [47])
- Slika 5.5.2.2. Izgled Kohonenove dvodimenzionalne mreže ([88])
- Slika 5.5.4.1. Linerna skalabilnost
- Slika 5.5.4.2. Linearna skalabilnost dvodimenzionalnog prostora ([89])
- Slika 5.5.4.3. Nelinearna skalabilnost dvodimenzionalnog prostora
- Slika 5.5.4.4. Raspored ulaznih parova I (AND) logičkog kola
- Slika 5.5.4.5. Raspored ulaznih parova ILI (OR) logičkog kola
- Slika 5.5.4.6. Raspored ulaznih parova NI (NAND) logičkog kola
- Slika 5.5.4.7. Rezultat rada koda pokrenut preko CMD shell-a Windows OS-a
- Slika 5.5.4.8. Raspored ulaznih parova XILI (XOR) kola bez mogućnosti linearne podele
- Slika 5.6.1. Vremenske serije kontinualne veličine
- Slika 5.6.2. Predviđanje jedan korak unapred
- Slika 5.6.3. Rekurzivno predviđanje sa dva koraka unapred
- Slika 5.6.4. Direktno predviđanje dva koraka unapred
- Slika 5.6.5. Heterogeno predviđanje sa dva izlaza

Slika 5.6.6. Heterogeno predviđanje sa jednim izlazom

Slika 5.7.1.1. Pokretanje programa preko CMD shell-a

Slika 5.7.1.2 Određivanje koeficijenta težine nasumičnim odabirom iz skupa uzoraka

Slika 5.7.1.3 Pokretanje programa preko CMD shell-a

Slika 5.7.1.4. Skup uzoraka kada je noise parametar jednak 0.25

Slika 5.7.1.5. Skup uzoraka kada je noise parametar jednak 10 (cmap='jet')

Slika 5.7.1.6 Skup uzoraka kada je noise parametar jednak 0.000001

Slika 5.7.1.7. Pokretanje koda iz CMD sela (shell)

Slika 5.7.1.8. Logistička regresija

Slika 5.7.1.9. Parametar d ima vrednost 0.1 i parametar d ima vrednost 1

Slika 5.7.1.10 Promena položaja linije granice

Slika 5.7.1.11 ReLU funkcija (Izvor: [79])

Slika 5.7.1.12 Učenje neuronske mreže sa tri neurona u skrivenom sloju-kod pokrenut preko CMD

Slika 5.7.1.13 Dijagram učenja neuronske mreže sa tri neurona u skrivenom sloju

Slika 5.7.1.14. Učenje neuronske mreže sa deset neurona u skrivenom sloju-kod pokrenut preko CMD šela

Slika 5.7.1.15. Dijagram klasifikacije uzoraka neuronske mreže sa deset neurona u skrivenom sloju-kod pokrenut preko CMD šela

Slika 5.7.1.16. Učenje neuronske mreže sa sto neurona u skrivenom sloju-kod pokrenut preko CMD šela

Slika 5.7.1.17. Dijagram klasifikacije uzoraka neuronske mreže sa sto neurona u skrivenom sloju-kod pokrenut preko CMD šela

Slika 5.7.1.18. Učenje neuronske mreže sa hiljadu neurona u skrivenom sloju-kod pokrenut preko CMD šela

Slika 5.7.1.19. Dijagram klasifikacije uzoraka neuronske mreže sa hiljadu neurona u skrivenom sloju-kod pokrenut preko CMD šela

Slika 6.1.1. Anketa1.csv otvoren pomoću notepad tekst editora

Slika 6.2.1. Pokrenut kod predvidjanje_fin_pismenosti_dece1.py gde su sve ćelije popunjene

Slika 6.2.2. Pokrenut kod predvidjanje_fin_pismenosti_dece1.py gde nisu sve ćelije popunjene

Slika 6.2.3. Rezultat pozivanja info() funkcije i redirektovanje rezultata te funkcije u fajl podaci_info.txt

Slika 6.4.1. Preosečna ocena na kraju prethodnog razreda i prosečna ocena iz matematike na kraju prethodnog razreda

Slika 6.4.2. Međusobna zavisnost opštih podataka od pola ispitanika

Slika 6.4.3. Međusobna zavisnost opštih podataka od pola ispitanika u procentima

Slika 6.5.1. Statistički podaci za kolonu pojam_inflacije kada je u pitanju tip podataka string i tip podataka int.

Slika 6.5.2. Statistički podaci za kolonu ocena_poznavanja_inflacije

Slika 6.5.3. Statistički podaci za kolonu pitanje_inflacija kada je u pitanju tip podataka string i int.

Slika 6.5.1.1 Međusobna zavisnost prva tri pitanja u anketi od datih odgovora i broja ispitanika

Slika 6.5.1.2. Tabelarni prikaz dobijenih rezultata koji su eksportovani u Anketa1_rezultati.xlsx fajl

Slika 6.5.1.3. Međusobna zavisnost prva tri pitanja u anketi od datih odgovora i broja ispitanika

Slika 6.5.1.4. Tabelarni prikaz dobijenih rezultata u procentima koji su eksportovani u Anketa2_rezultati.xlsx fajl

Slika 6.5.2.1. Pitanja prvog segmenta povezana sa opštim pitanjima

Slika 6.5.2.2. Pitanja prvog segmenta povezana sa opštim pitanjima (rezultat je dat u procentima)

Slika 6.5.2.3. Pitanja prvog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.5.2.4. Pitanja prvog segmenta povezana sa opštim pitanjima_procentulano (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.5.2.5. Pitanja prvog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.5.2.6. Pitanja prvog segmenta povezana sa opštim pitanjima_procentulano (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.6.1. Statistički podaci za kolonu kamatna_stopa kada je u pitanju tip podataka string i int.

Slika 6.6.2. Statistički podaci za kolonu ocena_poznavanja_kamatne_stope

Slika 6.6.3. Statistički podaci za kolonu pitanje_kamatna_stopa kada je u pitanju tip podataka string i int.

Slika 6.6.4. Statistički podaci za kolonu pitanje_kamatna_stopa

Slika 6.6.1.1. Grafik odnosa pitanja drugog segmenta ankete

Slika 6.6.1.2. Grafik odnosa pitanja drugog segmenta ankete u procentima

Slika 6.6.3. Podaci eksportovani u spread-šit

Slika 6.6.2.1. Pitanja drugog segmenta povezana sa opštim pitanjima

Slika 6.6.2.2. Pitanja drugog segmenta povezana sa opštim pitanjima_procentulano

Slika 6.6.2.3. Pitanja drugog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.6.2.4. Pitanja drugog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.6.2.5. Pitanja drugog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.6.2.6. Pitanja drugog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.7.1. Statistički podaci za kolonu osiguranje kada je u pitanju tip podataka string i int.

Slika 6.7.2. Statistički podaci za kolonu ocena_poznavanja_osiguranja

Slika 6.7.3. Statistički podaci za kolonu pitanje_osiguranje kada je u pitanju tip podataka string i int.

Slika 6.7.1.1. Grafik odnosa pitanja trećeg segmenta ankete

Slika 6.7.1.2. Grafik odnosa pitanja trećeg segmenta ankete_procentualno

Slika 6.7.1.3. Podaci eksportovani u spread-šit

Slika 6.7.2.1. Pitanja trećeg segmenta povezana sa opštim pitanjima

Slika 6.7.2.2. Pitanja trećeg segmenta povezana sa opštim pitanjima_procentualno

Slika 6.7.2.3. Pitanja trećeg segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.7.2.4. Pitanja trećeg segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.7.2.5. Pitanja trećeg segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.7.2.6. Pitanja trećeg segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.8.1 Statistički podaci za kolonu studentski_kredit kada je u pitanju tip podataka string i int.

Slika 6.8.2 Statistički podaci za kolonu ocena_poznavanja_studentskog_kredita

Slika 6.8.3. Statistički podaci za kolonu pitanje_studentski_kredit kada je u pitanju tip podataka string i int.

Slika 6.8.1.1. Grafik odnosa pitanja četvrtog segmenta ankete

Slika 6.8.1.2. Grafik odnosa pitanja četvrtog segmenta ankete

Slika 6.8.1.3. Podaci eksportovani u spread-šit

Slika 6.8.2.1. Pitanja četvrtog segmenta povezana sa opštim pitanjima

Slika 6.8.2.2. Pitanja četvrtog segmenta povezana sa opštim pitanjima_procentualno

Slika 6.8.2.3. Pitanja četvrtog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.8.2.4. Pitanja četvrtog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.8.2.5. Pitanja četvrtog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.8.2.6. Pitanja četvrtog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.9.1. Statistički podaci za kolonu pitanje_studentski_kredit kada je u pitanju tip podataka string i int.

Slika 6.9.2 Statistički podaci za kolonu ocena_poznavanja_kriptoaluta

Slika 6.9.3. Statistički podaci za kolonu pitanje_kriptoalute kada je u pitanju tip podataka string i int.

Slika 6.9.1.1. Grafik odnosa pitanja petog segmenta ankete

Slika 6.9.1.2. Grafik odnosa pitanja petog segmenta ankete_procentualno

Slika 6.9.1.3. Podaci eksportovani u spread-šit

Slika 6.9.2.4. Pitanja petog segmenta povezana sa opštim pitanjima

Slika 6.9.2.5. Pitanja petog segmenta povezana sa opštim pitanjima_procentualno

Slika 6.9.2.6. Pitanja petog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.9.2.7. Pitanja petog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu)

Slika 6.9.2.8. Pitanja petog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.9.2.9. Pitanja petog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Slika 6.10.1 Grupisanje odgovora iz kolona pitanje_inflacija i pitanje_kamatna_stopa od ženskog pola ispitanika

Slika 6.10.2 Grupisanje odgovora iz kolona pitanje_inflacija i pitanje_kamatna_stopa od prosečne ocene iz matematike čija je vrednost pet

Slika 6.11.1. Predviđanje koda

Slika 7.1. Primarni vizuelni korteks ([91])

Slika 7.2. Arapske cifre pisane rukom

Slika 7.3.3. Izgled neuronske mreže

1 Predmet i ciljevi istraživanja

Ciljevi koji su postavljeni u okviru ovog istraživanja proizilaze iz samog naziva i predmeta disertacije i odnose se na sledeće grupe:

1.1 Glavni cilj

Glavni cilj doktorske disertacije je izrada (projektovanje) modela baziranih na upotrebi veštačkih inteligentnih agenata koji će samostalno, a na osnovu prikupljenih i obrađenih podataka, dati predlog odluke. Podaci će biti organizovani na način koji će obezbediti njihovu brzu i laku dostupnost, mogućnost manipulacije i analizu, a sve u cilju dobijanja kvalitetnih informacija od strane autonomnog alata.

1.2 Naučni ciljevi

Naučni cilj je da se na osnovu rezultata eksperimenta uoče i definišu određene zakonitosti u ponašanju veštačkih inteligentnih agenata. Na osnovu uočenih zakonitosti biće urađena analiza uspešnosti primenjenih metoda u konkretnom sistemu. Analiza bi trebalo da da preporuke u načinima primene veštačkih inteligentnih agenata u različitim poslovnim informacionim sistemima. U ovoj disertaciji, istraživanje, kreiranje i modeliranje će se izvršiti na primerima srednjoškolskih ustanova.

Savremeni poslovni procesi zahtevaju primenu najnovijih tehnologija. Menadžmenti obrazovnih ustanova u Srbiji treba da shvate da će primenom savremenih tehnologija kreirati bolje uslove poslovanja sa jedne, a sa druge strane će školovati kvalitetan i odmah nakon školovanja upotrebljiv kadar.

Jedan od naučnih ciljeva je da se pokaže da je ograničavajući faktor u primeni savremenih alata još uvek veliko prisustvo klasičnog menadžmenta klasične organizacije koji nije u stanju da primeni nove tehnologije u procesu donošenja stratejskih odluka. Da bi to bilo izbegnuto nužno je pre ili za vreme uvođenja veštačkih inteligentnih agenata upoznati strateški menadžment sa njegovim mogućnostima i prednostima.

1.3 Ciljevi eksperimenta

Cilj eksperimenta je da pokaže da alati bazirani na inteligentnim agentima mogu pomoći menadžmentu određene kompanije ili ustanove u procesu donošenja kvalitetnih i pravovremenih strateških odluka. Informacije ekstrahovene iz podataka kao objektivnih nosioca istih, bi od strane inteligentnih agenata bile predstavljene menadžmentu u odgovarajućem obliku. Inteligentni agenti (koji će biti kreirani u eksperimentalnom delu doktorske disertacije) će činiti osnovu autonomne mašine koja će imati sposobnost davanja predloga odluke.

Podaci koji će biti korišćeni za analizu u okviru doktorske disertacije su prikupljeni u anketi pod nazivom: „Finansijska pismenost dece“. Sprovedena anketa je urađena u okviru projekta „Finansijska pismenost dece“ koju sprovodi Fakultet za poslovne studije i pravo Univerziteta Union – „Nikola Tesla“. Dati projekat obuhvata procese prikupljanja, selektovanja, obrade i analizu podataka. Projekat se sprovodi i tri države: Republika Srbija, Republika Slovenija i Republika Crna Gora. Cilj projekta je utvrđivanje stepena finansijske pismenosti srednjoškolske dece završnih godina u navedene tri države i upoređivanje dobijenih rezultata sa rezultatima razvijenih zemalja. Osnovna ideja ovog projekta je u činjenici da od 2018. godine na PISA testovima se proverava i finansijska pismenost dece.

Anketa, koja je sprovedena u okviru projekta, je dobila podršku Ministarstva za prosvetu, nauku i tehnološki razvoj. Pozivno pismo za učestvovanje srednjih škola u anketi koje je uputilo Ministarstvo za prosvetu, nauku i tehnološki razvoj, se nalazi u prilogu doktorske disertacije.

U eksperimentalnom delu doktorske disertacije je razvijen kod u programskom jeziku Pajton (Python) koji omogućava brzu analizu velike količine podataka i davanje složenih međuzavisnosti različitih parametara koji se javljaju u anketi. Pored toga razvijeni kod omogućava i određena predviđanja budućih ponašanja ispitanika. U doktorskoj disertaciji podaci koji su obrađeni i analizirani odnose se na populaciju srednjoškolske dece u Republici Srbiji.

1.4 Značaj i aktuelnost istraživanja (naučni i društveni)

Primena veštačkih inteligentnih agenata, kao pomoć u procesu donošenja strategijskih odluka, je jedna od najznačajnijih i najaktuelnijih tema u IKT oblasti. Sa druge strane i tema primene inovacija u obrazovnom sistemu je značajna i aktuelna. Ova tema je, kako je konstatovano, u Srbiji nedovoljno obrađena. Značaj upotrebe inovacija u obrazovnom sistemu se ogleda u tome da kvalitetnim ulaganjem kako u tehnološke inovacije, tako i u sam nastavni kadar se može postati lider u obrazovanju. Uvođenjem savremenih tehnologija se može ostvariti visok nivo efikasnosti u obrazovanju. Mladi ljudi su preokupirani sadržajima sa društvenih mreža i teško ih je motivisati da uče i napreduju. U nedostatku motivacije treba tražiti odgovore gde se greši u nastavnom procesu. Svet je globalno selo i ti problemi nisu karakteristični samo za Srbiju, već i za sve zemlje na svetu. U razvijenim zemljama se tom problemu pristupa ozbiljno. Otuda i potreba za jednim ovakvim istraživanjem koje će omogućiti efikasno uočavanje problema i dati predloge za preduzimanje odgovarajućih akcija.

Koncept inteligentnih agenata je odabran, jer isti mogu da budu kreirani u bilo kom okruženju nezavisno od njegovih karakteristika. Ovi agenti bi trebalo da se upotrebljavaju u kontekstu koje je dinamično, nepredvidivo i nepouzđano. Prikupljeni podaci predstavljaju takvo okruženje. Stalne promene operacionog konteksta isti čine dinamičnim. Ova okruženja su nepredvidiva, jer je nemoguće znati unapred dobijene odgovore. Primarni razlog zašto će biti primenjeni inteligentni agenti jeste autonomna reakcija agenta, odnosno mogućnost da agent može da daje relevantni predlog odluke na osnovu algoritama koji su u njega implementirani. Upravo ta samostalnost u donošenju odluka razlikuje inteligentne agente od dinamičnih objekata.

Ukoliko se sistem sastoji od nekoliko agenata, tada ova mogućnost dovodi do toga da je sistem distribuiran i decentralizovan, a dobijeno sistemsko rešenje kvalitetnije. Svrha upotrebe inteligentnih agenata leži u njihovoj proaktivnosti, odnosno sposobnosti da prevaziđu stanja koja ih mogu sprečiti da ispune cilj. U tom smislu inteligentni agenti će kreirati nova rešenja koja će ih dovesti do unapred specificiranog cilja.

Glavni cilj u procesu kreiranja inteligentnih agenata, u okviru doktorske disertacije, biće usklađivanje njihove reaktivnosti sa njihovom proaktivnošću. Sa jedne strane

kreirani agent mora da bude sposoban da reaguje na promene okruženja, a sa druge mora da bude proaktivan i dostigne unapred specificirani cilj. Saznanja do kojih će u doktorskoj disertaciji inteligentni agenti doći, bi pomogli Ministarstvu prosvete, nauke i tehnološkog razvoja u davanju smernica za finansijsko opismenjavanje dece. Iz tog razloga će u doktorskoj disertaciji biti data tumačenja dobijenih rezultata do kojih su došli veštački inteligentni agenti tokom procesa obrade i analize skupa prikupljenih podataka.

1.5 Metodologija istraživanja, hipoteze i faze realizacije eksperimenta

Izrada doktorske disertacije, postavka i razrada, kao i opšta istraživanja baziraće se na određenim naučno-istraživačkim metodama:

- Sistemskom pristupu i kompletnoj analizi delova i operacija.
- Metoda indukcije će se koristiti da bi se na osnovu rezultata eksperimentalnog dela rada definisao skup pravila koji bi mogao da bude primenjen na različite tipove okruženja na koje mogu da se primene inteligentni agenti.
- Metoda dedukcije će naći svoju primenu u istraživanju tokom teorijskog razmatranja alata koji se zasnivaju na veštačkoj inteligenciji. Deduktivnom metodom bi se formulisali opšti zaključci i zakonitosti o autonomnim alatima za samostalno donošenje odluka. Tako definisani zaključci i zakonitosti bi pomogli u eksperimentalnom delu doktorske disertacije. Na taj način bi na primerima razmatranim u okviru poglavlja *Neuronske mreže* doktorske disertacije bile izbegnute „zamke” koje se mogu javiti tokom kretanja inteligentnih agenata.
- Metode konceptualnog i logičkog modeliranja podataka.
- Komparativna analiza koja će biti urađena u doktorskoj disertaciji će pokazati razlike u brzini izvršavanja procesa obrade i analize podataka između čoveka i inteligentnih agenata.
- Kvantitativne metode prikupljanja podataka, konkretno u doktorskoj disertaciji: anketa. Ovom metodom će biti prikupljeni podaci o jednom od aspekata srednjoškolskog obrazovnog sistema.
- Metode semantičke obrade podataka od strane inteligentnih agenata.

U istraživanju će se koristiti veći broj navedenih metoda, jer se pokazalo da nijedna metoda i tehnika nisu dovoljne da bi precizno bila definisana pravila pri kreiranju inteligentnih agenata koji bi mogli semantički da interpretiraju prikupljene podatke.

U skladu sa predmetom rada, postavljenih ciljeva i zadataka istraživanja, formulisana je osnovna hipoteza i nekoliko posebnih hipoteza.

Osnovna hipoteza:

H: Eksperimentalni deo doktorske disertacije bi trebalo da pokaže da inteligentni agenti mogu brže da reše kompleksni nestruktuirani problem u odnosu na čoveka i da na osnovu toga predlože strategijsku odluku koja će biti istog ili boljeg kvaliteta u odnosi na odluku koju bi doneo ekspert iz date oblasti.

Posebne hipoteze:

h1: Inteligentni agenti koji koriste tzv. pojačano učenje, tj. učenje koje se zasniva na nagradama i kaznama uče brže (brže dolaze do boljih rešenja nego inteligentni agenti koji koriste klasično učenje.

h2: Intelligenti agent najbolje uči kroz istoriju sopstvenih iskustava sa okolinom u kojoj deluje.

h3: Predloženo rešenje će biti kvalitetnije ukoliko se koristi više nezavisnih inteligentnih agenata u okviru datog okruženja.

Doktorska disertacija, pod navedenim naslovom, treba da potvrdi ili ospori navedene hipoteze.

Eksperimentalni deo, kao glavni deo doktorske disertacije, će sadržati sledeće faze istraživanja:

- Analiza realnog problema
- Kreiranje modela realnog problema i predložena konceptualna rešenja
- Definisavanje inputa za inteligentne agente
- Dizajn agenata sa posebnim fokusom na autonomnost agenata i sposobnost njihovog donošenja odluka

- Dizajn i testiranje algoritama donošenja odluka pojedinačnih agenata
- Dizajn sistema agenata i definicija komunikacionih protokola između agenata
- Softverska implementacija pojedinačnih agenata
- Sistemska integracija agenata
- Prikupljanje podataka i testiranje sistema agenata
- Predlog korektivne optimizacije sistema agenata.

1.6 Očekivani naučni doprinos istraživanja

Nakon obrade i istraživanja na temu doktorske disertacije pod naslovom: „Primena modela inteligentnih agenata u procesu donošenja strategijskih odluka“, realno je očekivanje da se nauka o menadžmentu upotpuni novim pristupima u procesu donošenja strategijskih odluka uz korišćenje inteligentnih agenata sa posebnim osvrtom na obrazovne ustanove.

Okruženje obrazovnog sistema u Srbiji se menjalo brže od samog obrazovnog sistema što za posledicu ima da je obrazovni sistem u Srbiji spor u prihvatanju inovacija koje mogu da pospeše sam proces obrazovanja. U okviru navedenog je potrebno ukazati na opravdanost korišćenja savremenih tehnologija u obrazovnom sistemu, pri čemu je potrebno obrazovni sistem sagledati sa novog stanovišta, odnosno potrebno ga je posmatrati kao savremeni poslovni sistem. Postoji velika potreba za konceptualnim rešenjima korišćenja inteligentnih agenata u obrazovanju, a predlozi strategijskih odluka od strane inteligentnih agenata u okviru ove doktorske disertacije će omogućiti fokusiranje na poboljšanje obrazovanja.

Da bi sistem obrazovanja u Srbiji napredovao neophodno je da se primene iskustva razvijenih zemalja uz određeni stepen adaptacije. U doktorskoj disertaciji će, na osnovu analize posmatranih sistema, biti predložena rešenja koja su u suštini jednostavna. Problem, na koji će biti pokušano da se da odgovor u okviru disertacije, je problem koji se javlja prilikom odabira optimalnog rešenja iz velikog broja mogućnosti.

U eksperimentalom delu doktorske disertacije će biti razmatrani podaci koji se prikupljaju anketom. Ankete su standardizovane, pa je pouzdanost ankete, koja se meri Kronbahovim alfa koeficijentom (Cronbach's alpha) visoka. Anketa koja će biti

sprovedena u okviru doktorske disertacije će biti zasnovana i adaptirana prema standardizovanim pitanjima:

- Sveučilišta u Zagrebu (dostupno na: <http://www.unizg.hr/ankete-zavrsne/>),
- Samovrednovanja Fakulteta za streteški i operativni menadžment (dostupno na: <http://www.fsom.edu.rs/samovrednovanje/>),
- Reinforcement of the Framework for Experiential Education in Healthcare in Serbia –ReFEEHS koji se realizuje u okviru projekta ERASMUS+Izgradnja kapaciteta u visokom obrazovanju (dostupno na: <http://www.mfub.bg.ac.rs/dotAsset/99774.pdf>).

Disertacija treba da dokaže da veštački inteligentni agenti mogu da daju predloge odluke u veoma dinamičnim uslovima. Svi prikupljeni podaci će biti ponovo analizirani od strane veštačkih inteligentnih agenata. Ovakav pristup u doktorskoj disertaciji će ukazati da prilikom analize izmenjenih podataka agent koristi sopstvena prethodna iskustva. Ovo je značajno, jer će u disertaciji biti predložen koncept koji može biti primenjen na različite poslovne sisteme uz određene modifikacije algoritama inteligentnih agenata.

1.7 Sažet opis sadržaja po poglavljima

Doktorska disertacija je koncipirana tako da ima teoretski i eksperimentalni deo. Napisana je u sedam poglavlja.

Poglavlje I – Predmet i ciljevi istraživanja

U ovom delu će biti dat pregled relevantne naučno-tehničke literature, određen predmet i navedeni ciljevi istraživanja.

Poglavlje II - Strategijsko odlučivanje u poslovnim sistemima

Fokus će biti stavljen na pojmovno-kategorijalno određenje načina donošenja poslovnih odluka od strane menadžmenta. U ovom poglavlju će biti predočene prednosti korišćenja savremenih tehnologija u procesu odlučivanja menadžmenta.

Poglavlje III - Tehnike pretraživanja i kompresije velike količine podataka

U okviru doktorske disertacije će biti izveden eksperiment u čijoj osnovi će ležati prikupljanje i obrada podataka. Iz tog razloga će u ovom poglavlju biti definisani pojmovi *big data* i *data mining* kao savremenih metoda i tehnika pretraživanja velikih skupova nestruktuiranih podataka. Cilj primene ovih tehnika je dobijanje korisne informacije iz, naizgled, nepovezanih skupova podataka.

Poglavlje IV- *Veštačka inteligencija*

U ovom poglavlju će kroz istorijski pregled razvoja veštačke inteligencije biti detaljno opisani svi koncepti iste. Za svaki od koncepata će biti date njihove prednosti i mane. Biće pojmovno određena semantika kao neizostavni deo koncepta veštačke inteligencije. Primena semantike u oblasti veštačke inteligencije daje uvod u definisanje primene veštačke inteligencije u oblasti odlučivanja. Sam eksperimentalni deo će se zasnivati na kreiranju modela baziranih na veštačkim inteligentnim agentima u čijoj osnovi se nalazi veštačka inteligencija.

Poglavlje V – *Neuronske mreže*

Ovde će biti definisani pojmovi neuronskih mreža i kroz originalne primere će biti prikazana upotreba veštačkih inteligentnih ageanta kao autonomnih mašina koje mogu da daju predlog za donošenje kvalitetne odluke. Biće opisan detaljan postupak klasifikacije podataka u cilju analize podataka. Pored toga će biti data podela neuronskih mreža i matematički koncepti principa rada veštačkih neurona i neuronskih mreža. Biće data komparativna analiza između biološkog, tj. ljudskog neurona i veštačkog neurona.

Poglavlje VI – *Empirijsko istraživanje i eksperiment*

U okviru ovog poglavlja će biti date polazne postavke eksperimenta i biće analizirani podaci prikupljeni anketom. Biće kreirani i implementirani algoritmi veštačkih inteligentnih agenata koji će vršiti pretragu podataka.

Poglavlje VII – *Zaključna razmatranja*

U ovom poglavlju će biti razmatrani rezultati eksperimenta i na kraju će biti date preporuke i smernice za primenu utvrđenih rezultata.

2 Strategijsko odlučivanje u poslovnim sistemima

U ovom poglavlju je dat kratak osvrt na to šta je poslovni sistem, koje su karakteristike i ciljevi poslovnog sistema, kao i ko upravlja poslovnim sistemom. Na osnovu ovog poglavlja će, u daljem delu doktorske disertacije, biti moguće naći primenu veštačkih inteligentnih agenata u poslovnim sistemima.

Poslovni sistem je vrsta organizovanog dinamičkog sistema. Osnove karakteristike poslovnog sistema su: otvorenost, povezanost, dinamičnost, složenost i stalna težnja za razvojem. Težnja za razvojem poslovnog sistema je uslovljena spoljnim uticajima na sistem. Razvoj poslovnog sistema se ogleda u stalnoj težnji sistema da bude što efikasniji.

Svaki sistem u toku svog postojanja i rada se ponaša prema određenim pravilima. Ova činjenica važi za sve sisteme, počev od ljudskog organizma koji se može posmatrati kao sistem. Poslovni sistem nije uravnotežen ako se ne poštuju pravila po kojima sistem treba da postoji i da radi, a to može dovesti do smanjenja efikasnosti samog sistema. U dužem vremenskom periodu nepoštovanje pravila po kojima treba da posluje dati sistem može dovesti do propadanja istog.

Osnovne jedinice poslovnog sistema su: elementi sistema i relacije između elemenata sistema. Elementi sistema imaju svoje karakteristike i svojim ponašanjem utiču na ostale elemente sistema, što dovodi do postojanja određenih zavisnosti između elemenata sistema. Elementi sistema primaju određene signale, tj. ulazne veličine koje dovode do izvođenja elementa iz stanja ravnoteže i samim tim izvođenja sistema, kome dati element pripada, iz ravnoteže. Nakon obrade signala, tj. ulazne veličine sistem proizvodi izlaznu veličinu kojom može da utiče na stanja drugih elemenata sistema.

Poslovni sistemi imaju dve vrste cilja: osnovni cilj i posebne ciljeve. Dostizanjem osnovnog cilja, poslovni sistem ostvaruje svrhu sopstvenog postojanja i obezbeđuje svoje dalje postojanje. Najčešći osnovni cilj poslovnog sistema je ostvarivanje ekonomske koristi, čime se podmiruju potrebe pojedinca i društva.

Posebni ciljevi se odnose na pojedinačne elemente posmatranog poslovnog sistema, a donose se na osnovu osnovnog cilja poslovnog sistema.

Za ostvarivanje ciljeva poslovnog sistema potrebni su dodatni elementi. Ti dodatni elementi su menadžeri poslovnog sistema. Menadžeri poslovnog sistema donose optimalne odluke koje omogućavaju postojanje, rad i napredak poslovnog sistema.

Poslovne odluke predstavljaju određene odabrane akcije koje menadžer preduzima radi ostvarivanja poslovnih ciljeva posmatranog preduzeća. U izveštaju Gartner grupe (Gartner, Inc. vodeća svetska konsultantska i istraživačka kompanija na polju informacionih tehnologija) *Predicts 2017: Artificial Intelligence* [94], predviđa se sledeće:

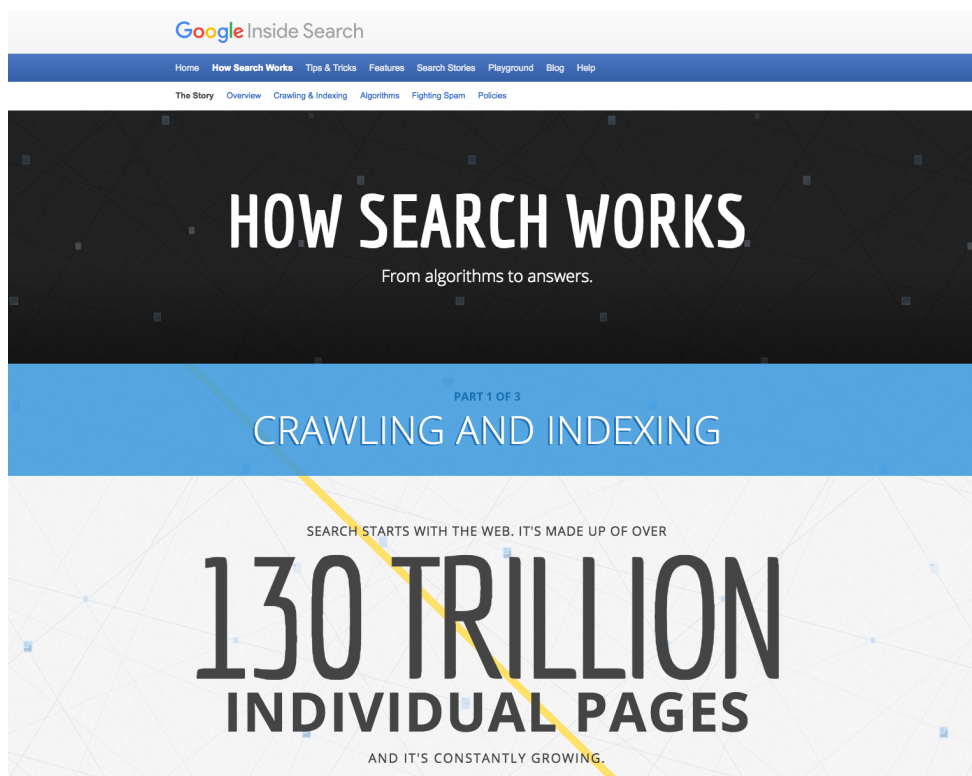
- Do 2019. godine startup kompanije će prestići IT gigante (Amazon, Google, IBM i Microsoft) u oblasti koja nudi usluge veštačke inteligencije u procesu rešavanja poslovnih problema.
- Do 2019. godine, usluge platforme za veštačku inteligenciju će oboriti prihode za 30% vodećih kompanija na tržištu.
- Do 2020. godine organizacije koje koriste kognitivnu ergonomiju i dizajn sistema u novim projektima veštačke inteligencije će postizati četiri puta dugoročniji uspeh u odnosu na kompanije koje taj pristup ne koriste.
- Do 2020. godine u 20% kompanija će deo radnika biti zadužen za kontrolu i vođenje neuronskih mreža.

Na osnovu analize datog izveštaja Gartner grupe može se zaključiti da:

- Veliki broj kompanija i pojedinaca već razvija odgovarajuće alate za autonomno donošenje odluka.
- Razvijanje alata za autonomno donošenje odluka predstavlja IKT oblast koja je u brzom ekspanziji.

3 Tehnike pretraživanja i kompresije velike količine podataka

Pojava računara dovodi do pitanja kvalitetne i brze obrade podataka od strane automatizovanih sistema. Količina podataka koja se obrađuje je u stalnom porastu. Danas na vidljivom delu Interneta i to samo na guglu (eng. Google), ima preko 130 triliona (130×10^{12}) indeksiranih Veb stranica [1]. Gugl je ovaj podatak poslednji put učinio dostupnim javnosti u novembru 2016. godine te sledi slika 1. koja potvrđuje ovu činjenicu, jer na navedenoj Veb adresi taj podatak više nije dostupan.



Slika 3.1.1. Broj indeksiranih Veb stranica na guglu (izvor: [1])

Pored gugla postoje i drugi Veb pretraživači. Da ne bi došlo do preklapanja istih Veb stranica koriste se algoritmi koji upoređuje reči koje su navedene u pretrazi za tri dominantna pretraživača: gugl, jahu (eng. Yahoo) i bing (eng. Bing) [2], [3]. Ovaj broj Veb stranica na Internetu nije konačan broj. Vidljiv deo Interneta čine samo indeksirane Veb stranice, a to je svega oko 5% Interneta, jer postoji i nevidljivi deo Interneta, tzv. Deep Web, koji se sastoji od neindeksiranih veb stranica čija je ekstenzija .oi. Pretraživač koji se koristi za pristup tim Veb stranicama je TOR (The Onion Routing). Broj neindeksiranih strana je 400 do 500 puta veći od broja indeksiranih strana [4]. Ovo je prikaz količine podataka koja se trenutno nalazi samo na Internetu. Pored toga postoje i

drugi izvori podataka kao što su to različiti uređaji. Trenutni trend je razvoj oblasti IoT (Internet of Things) u kojoj dolazi do umrežavanja uređaja, kreiranja i razmene podataka, odnosno signala.

Pod terminom *Big Data* se podrazumeva skup negrupisanih podataka kojima zbog svoje veličine i složenosti ne može biti upravljano klasičnim metodama. Termin Big Data je pomenut prvi put u prezentaciji Džona Mešija (John Mashey) “*Big Data and the Next Wave of InfraStress Problems, Solutions, Opportunities*” [95] 1998. godine.

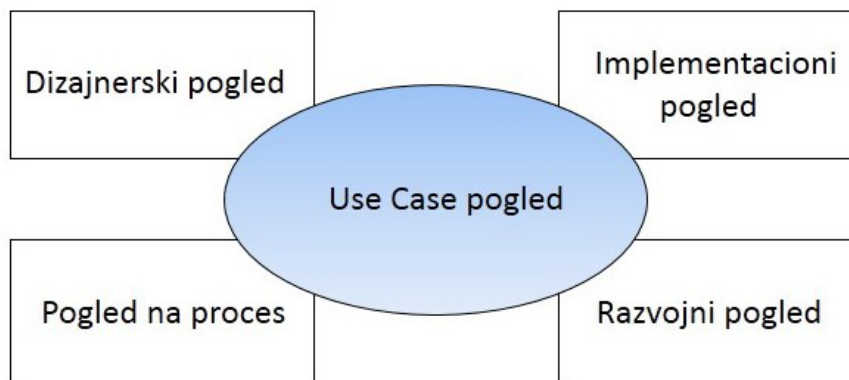
Da bi iz ogromne količine podataka bili izvučeni značajni podaci, potrebno je koristiti tehnike *Data Mining*-a (DM), odnosno tehnike efikasne obrade podataka. Sve više organizacija uviđa prednosti obrade uskladištenih podataka. Osnovna prednost se ogleda u tome da tehnike DM omogućuju kvalitetnije strateško planiranje. Primena DM tehnika se značajno isplaćuje organizacijama koje je koriste. Glavne tehnike koje se koriste radi obrade ogromne količine podataka su: statistika, veštačka inteligencija, grafovi zasnovani na strukturi drveta i vizuelizacija.

Mnogi sistemi baza znanja imaju kompleksne podatke, kao što su to: hipertekst, multimedijalni podaci i prostorni podaci (predstavljaju objekte koji se nalaze u geometrijskom prostoru). Ako je tehnika DM dovoljno dobro razvijena, tada bi ona mogla da na efektivan način pretražuje i obrađuje te različite vrste podataka. Da bi DM tehnika bila dobro razvijena, algoritmi koji je pokreću moraju da budu efikasni u pogledu pretraživanja date baze, odnosno vreme koje ti algoritmi koriste za pretragu date baze mora da bude unapred poznato. Vreme pretrage date baze treba da bude srazmerno veličini te baze (odatle potiče činjenica da vreme pretrage treba da bude unapred poznato).

Da bi primena DM tehnike imala smisla potrebno je da podaci, koji su rezultat pretrage i obrade, imaju smisla krajnjem korisniku. To dovodi do toga da ti dobijeni podaci, nakon primene DM tehnika, moraju biti u formi koju će krajnji korisnik da razume i što je bitnije, dobijeni podaci treba imaju upotrebnu vrednost za krajnjeg korisnika.

Vrlo je teško odlučiti koji od pogleda na bazu znanja (posmatranu kao softverski sistem) koristiti ili kako izvući korisne podatke iz date baze. Sa druge strane vrednost dobijenog podatka/podataka zavisi od procene krajnjeg korisnika. Jedan te isti dobijeni

podatak za različite korisnike ima različitu vrednost. Odatle proističe potreba za DM-om na različitim nivoima apstrakcije. To je razlog da DM alati podržavaju “rudarenje” podatka na različitim nivoima apstrakcije.



Slika 3.1.2. Različiti pogledi na softverski sistem

Postoje tri glavna koraka u procesu primene DM tehnika:

1. Pripremanje podataka.
2. Smanjenje polazne količine podataka na podatke koji imaju značaj za dati upit.
3. Potraga za korisnom informacijom i to obradom dobijenih (korisnih) podataka.

3.1 Primena Data Mining-a i Big Data u praksi

Potpuno razumevanje ponašanja potrošača, odnosno naći odgovore na pitanja: zašto kupuju, na koji način više vole da kupuju, zašto su prešli kod konkurencije, šta će sledeće kupiti, šta ih je navelo da preporučie kompaniju drugima, je strateško pitanje za svaku kompaniju. Da bi top menadžment velikih korporacija mogao da dobije odgovor na ova pitanja, potrebno je iz velike količine podataka izvući i obraditi korisne podatke. Ovo je jedino moguće DM tehnikama. Sa druge strane princip pretrage podataka ne može biti zasnovan samo na podacima koji su uzeti od kupca. Na taj način će kupac biti tretiran kao informacioni objekat. To nije dobar pristup, jer bez obzira na prikupljene podatke postoji faktor nesigurnosti koji proizilazi iz nepoznavanja emocija kupca. Korporacije su u poslednjih 10 do 15 godina primenjivale dva sistema za prikupljanje i obradu podatka i to: *Customer Relationship Management (CRM)* sistem i *Master Data Management (MDM)* sistem. Uočena je potreba za primenom boljeg i efikasnijeg sistema. To je razlog primene DM tehnika. Organizacije su primenom ove tehnike uspele da predvide

ponašanja kupaca (u odgovarajućoj meri) i da na osnovu toga postignu dublju vezu sa kupcem. Ovaj pristup omogućava kvalitetnije iskustvo koje će kupac imati.

Izvori podataka kao što su: transakcije, višekanalne interakcije, društveni mediji, prikupljeni podaci kroz različite izvore kao što su to kartice odanosti (eng. loyalty cards), su značajno povećali sposobnost organizacija da kreiraju kompletnu sliku o sklonostima i zahtevima klijenata. Kreiranje te slike je bio krajnji cilj odeljenja marketinga, prodaje i podrške kupcima. Ovaj princip poslovanja nije više karakterističan samo za prodaju, već je primenjiv i u medicini, telekomunikacijama, državnoj upravi i bankarstvu.

Društveni mediji predstavljaju jedan od novih izvora velike količine podataka. U društvene medije se ubrajaju, pre svega, društvene mreže: Fejsbuk (eng. Facebook) i Tviter (eng. Twitter). Ukoliko neki proizvod ili usluga nije dobar/dobra kupci će odmah preko društvenih mreža obavestiti svoje prijatelje. Na taj način ove mreže imaju značajnu ulogu u formiranju mišljenja kupaca. Zahvaljujući ovim podacima, marketinška odeljenja mogu da ciljaju različite grupe kupaca i to na osnovu prethodno prikupljenih i obrađenih velikih skupova podataka.

Korišćenje DM tehnika u poslovanju omogućava da obrađeni podaci budu korisni i zaposlenima koji ne rade u marketingu. Ukoliko jedna organizacija ima veliki broj radnji u različitim zemljama, zaposleni koji su zaduženi za nabavku nove robe u tim radnjama moraju stalno da znaju šta im je od robe potrebno. Najefikasniji način za to jeste upravo upotreba DM tehnika. Podaci šta ima na stanju (odakle se vidi koja roba se najviše prodaje) se stalno updejtuju (eng. update). Iz te ogromne količine podataka pomoću DM tehnika će biti određeno koju robu i u kojoj količini treba poslati u datu radnju. Na taj način primenjena DM tehnika će top menadžmentu dati sledeće poglede na ukus kupaca u datoj radnji:

- Istorijski pogled: ogleda se u obrascima prodaje koji se kreiraju na osnovu vrste prodane robe, regiona prodaje robe i cena prodane robe.
- Lokalni pogled: iz prikupljenih podataka se otkriva sentimentalna informacija o lokalnim trendovima.

Obradom negrupisanih podataka pomoću DM tehnika top menadžment date organizacije će znati:

- kako da odgovori na zahteve kupaca u što kraćem vremenu,
- da prati trend kupaca,
- da poboljša taktiku u različitim oblastima poslovanja,
- kako kompanija da bude ispred konkurencije.

Na osnovu prethodnih razmatranja može se uočiti veliki broj promenljivih koji utiču na donošenje kvalitetne poslovne odluke. Problem se dodatno usložnjava što je skup podataka raznovrsniji i veći. Današnje poslovanje u sebe uključuje veliki broj raznovrsnih podataka koje je potrebno analizirati. To je razlog primene različitih tehnika u rešavanju tako složenog problema.

4 Veštačka inteligencija

4.1 Pojam veštačke inteligencije

Da bi u radu na odgovarajući način bio prikazan razvoj veštačkih neuronskih mreža, potrebno je dati kratku retrospektivu razvoja veštačke inteligencije povezujući koncepte evolucije i veštačke inteligencije.

Čovečanstvo je u toku sopstvene evolucije (koja i dalje traje i koja predstavlja jedan neprekidan proces) spoznalo načine izrade i korišćenja alata. Ti alati nastaju kao posledica primene određene i trenutno dostupne tehnologije. U početku je primenjena tehnologija, kao i samo čovečanstvo, bila primitivna. Početni cilj te primitivne tehnologije je bio ubrzavanje radnih procesa (skupljanje hrane, lov, izgradnja prvobitnih naseobina,...) koje je čovek izvodio. Daljim razvojem čovečanstva svrha tehnologije nije više samo u kreiranju alata koji omogućavaju olakšavanje izvršenja manuelnih poslova koje je čovek obavljao, već se teži da se pomoću, trenutno dostupne i aktuelne tehnologije kreiraju alati koji će moći da oponašaju čoveka kako u fizičkom tako i u intelektualnom smislu. Jedna od osnovnih odlika ljudskog intelekta je pravovremeno donošenje odluka. Donošenje odluka predstavlja jednu od posledica procesa evolucije. Pojam evolucije je prvi definisao Čarls Darwin (Charles Darwin) u knjizi „*Poreklo vrsta*” izdatoj 1859. godine. Po Darwinu evoluciju treba posmatrati kao proces razvoja (kako u fizičkom, tako i u intelektualnom smislu) svih živih organizama u toku veoma dugog perioda vremena (milioni godina). Evolucija nastaje kao posledica promena u okruženju (kako nežive tako i žive prirode) u kome data vrsta živi. Kao rezultat te promene svako buduće pokolenje date vrste se menja i nadograđuje određene nasleđene osobine [5]. Sa druge strane u savremenom dobu, zahvaljujući napretku čovečanstva koje se ogleda i u tehnologiji koju je čovečanstvo razvilo, evolucija ne može biti razmatrana samo za živa bića, već se mora razmotriti i evolucija samih mašina koje je čovek osmislio i konstruisao. Brian Artur (Brian Arthur), jedan od pionira naučne discipline Teorija kompleksnosti, u knjizi „*The Nature of Technology: What It Is and How It Evolves*” razmatra evoluciju mašina smatrajući da su „... mašine započele kao različiti delovi naizgled nepovezanih tehnologija, ali kao i ljudi, mašine takođe imaju poreklo i proces evolucije” [6].

Definisanjem pojma evolucije na ovakav način uvodi se novi pojam: Veštačka inteligencija (eng. Artificial intelligence) [7].

Kako je navedeno u prethodnom delu evolucija vrste nastaje kao posledica reakcije promene te vrste na okolinu.

Termin *Inteligencija* se može definisati na sledeće načine:

- Inteligencija je sposobnost pojedinca da razume i uči nove stvari [8].
- Inteligencija je sposobnost razmišljanja u toku donošenja odluka, tj. ne raditi stvari na osnovu instinkta i automatizma [8].

Iz prve definicije sledi da pojam inteligencije važi samo za živa bića, jer u sebe uključuje reč pojedinac koja se odnosi na živa bića. U drugoj definiciji nije naglašeno da se radi o živim bićima. Ta definicija uvodi novi pojam „razmišljanje”. Isti izvor [8] navodi da je razmišljanje aktivnost u kojoj se koristi mozak da bi dati problem bio rešen ili da bi određene ideje bile kreirane. Problem kod druge definicije je u tome što je na osnovu termina „razmišljanje” uveden pojam mozak koji predstavlja ljudski organ. Samim tim i ova definicija je teško primenjiva kada su u pitanju mašine.

Osnovni cilj oblasti Veštačka inteligencija je da pomoću naučnih principa utvrdi da li je moguće napraviti mašinu koja bi rešila zadati problem i to na način na koji bi čovek rešio taj problem (uz upotrebu procesa razmišljanja, tj. mozga). Problem koji se javlja kod ove definicije je u tome što ne može biti na jednostavan način utvrđeno kada mašina pokazuje znake samostalnog razmišljanja koje odudara od algoritama koji su u nju ugrađeni. Odgovor da li neka mašina poseduje inteligenciju je „zamagljen”, nejasan, odnosno fazi (eng. fuzzy - nejasan, maglovit, nerazumljiv u celini). Termin fazi objašnjava i postupke samih ljudi, jer se može desiti da će u jednoj te istoj situaciji (situaciji koja se ponavlja više puta) isti čovek reagovati različito. U jednoj te istoj situaciji koja se ponavlja, čovek će doneti bolje ili lošije odluke u odnosu na odluke koje je donosio ranije kada je bio izložen toj istoj situaciji. Tada se postavlja pitanje da li čovek izložen jednoj te istoj situaciji ima isti stepen inteligencije ili ne. Postoji veliki broj primera ljudi koji poseduju inteligenciju da rešavaju vrlo složene zadatke, ali su zato u pogledu socijalne inteligencije na vrlo niskom nivou. Sve ovo navodi da je i u ovim slučajevima (kada su u pitanju ljudi) u pogledu inteligencije sve „zamagljeno”, tj. fazi.

Jedan od prvih članaka u kome su povezani pojmovi inteligencije i mašina jeste rad Alana Tjuringa (Alan Turing) „*Computing machinery and intelligence*” [9]. Značaj ovog rada je u tome da je on izdržao test vremena i da i danas pokazuje svoj univerzalni karakter. Tjuring je u svom radu pokušao da da odgovore na pitanja kao što su:

- Da li postoji misao bez iskustva?
- Da li postoji razum bez komunikacije?
- Da li postoji govorni jezik ukoliko nema živih ljudi koji pričaju taj jezik?
- Da li postoji inteligencija bez života koji čovečanstvo zna?

Tjuring u radu nije dao definicije, ali je izvršio misaoni eksperiment kojim je uprostio složenu semantiku koja se javlja u oblasti istraživanja veštačke inteligencije. Taj misaoni eksperiment se sastojao u kreiranju igre koju je nazvao: Igra imitacije (eng. Imitation game). Osnova ove igre jeste zamena pitanja „Da li mašine mogu da misle?” pitanjem „Da li mašina može da oponaša inteligentan oblik života?”. Tjuring je predviđao da će do 2000-te godine postojati računar koji će biti tako programiran da će moći u 30 posto slučajeva da obmane ljudskog ispitivača i da ga uveri da je u pitanju čovek, a ne mašina. Tjuring je definisao inteligentno ponašanje računara kao njegovu sposobnost da dostigne ljudski nivo u izvršavanju kongnitivnih zadataka, tj. zadatka u kojima se koristi misaoni proces. Mašina je uspešno položila test ukoliko ljudski ispitivač ne može da, na osnovu dobijenih odgovora, razlikuje čoveka od mašine.

Tjuringov test se sastoji od dve faze. Karakteristika obe faze je da se ljudski ispitivač nalazi u prostoriji koja razdvaja ispitanike. Ispitivač ne zna ko su ispitanici niti može da stupi u kontakt sa njima. Sve odgovore koje dobija (u idealnom slučaju) dobija preko terminala. Na taj način ispitivač nije u mogućnosti da na osnovu svojih čula predvidi pol ispitanika. U prvoj fazi testa, Tjuring posmatra troje ljudi i to: muškarca (A), ženu (B) i ispitivača (C, nije bitan pol). Ispitivač mora da na osnovu pitanja koja postavlja pogodi ko je od ispitanika muškarac, a ko žena. Po pravilima igre koje je definisao Tjuring, ispitanik A mora da ubedi ispitivača da je žena (laž), dok ispitanik B takođe tvrdi da je žena (istina). Tjuring je predložio da prvo pitanje koje bi bilo upućeno ka ispitaniku A bude:

C: Kolika Vam je dužina kose?

Odgovor koji na ovo pitanje daje ispitanik B glasi:

A: Kosa mi je kratko potšišana sa pojedinim krajevima od dvadesetak centimetara.

Iz ovog odgovora ispitivač ne može da utvrdi da li se radi o muškarcu ili ženi. Ispitivač bi morao da postavi još pitanja i da čeka na grešku muškog ispitanika da bi mogao sa sigurnošću da tvrdi ko je ko u ovoj igri.

U drugoj fazi ove igre muškarac je zamenjen računatom. Računar poseduje program koji mu omogućava da oponaša odgovore muškarca. Računar će praviti iste greške kao čovek, čak će davati i nejasne odgovore. Ukoliko računar uspe da prevari isti ili veći broj puta ispitivača, u odnosu na broj uspešnih obmana od strane muškarca kao subjekta ispitivanja, tada se može reći da je računar položio test inteligentnog ponašanja.

Oponašanje muškarca od strane mašine nije presudno za inteligenciju. Kako, po uslovu igre, ispitivač ne vidi i ne čuje ispitanike, njemu je dozvoljeno da postavi bilo koje pitanje ispitanicima. U cilju utvrđivanja ko je mašina u igri, ispitivač može zahtevati od oba ispitanika da izvrše veoma složene matematičke operacije koje im je zadao. Osnov za ovakvo pitanje leži u činjenici da ispitivač smatra da će računar dati brže odgovor u odnosu na čoveka, a ujedno ispitivač pretpostavlja da će računar dati i tačan odgovor. Takav pristup ispitivača za posledicu ima kreiranje algoritama na osnovu kojih će računar da daje odgovore na određena pitanja sa zakašnjenjem, da daje netačne odgovore ili oboje. Pored toga ispitivač može postavljati pitanja čiji će odgovori zahtevati emocionalnu inteligenciju.

Tjuringov test pokazuje svoju univerzalnost koja se ogleda u sledećem:

- Koristeći posrednike u komunikaciji (kao što su terminali) ispitivač nije u mogućnosti da fizičkim opažanjem utvrdi da li je ispitanik mašina ili čovek. Na taj način postiže se objektivni pristup tokom testiranja, odnosno ispitivač ne favorizuje ni jednu stranu tokom ispitivanja.
- Test je u potpunosti nezavisan u odnosu na detalje eksperimenta. Testiranje može da obuhvata obe navedene faze ili može da obuhvata samo drugu fazu. Ispitivač može postaviti bilo koje pitanje i na osnovu dobijenih odgovora može staviti fokus na oblast koja mu može pomoći da utvrdi ko je ko u ovoj igri.

Nijedna mašina nije dosada prošla Tjuringov test, ali uprkos tome ovaj test predstavlja osnovu za verifikaciju i validaciju sistema koji se baziraju na znanju (eng. Knowledge

Based Systems). Da bi neki program bio smatran inteligentnim u nekoj usko-specijalizovanoj oblasti, performanse tog programa moraju biti poređene sa performansama ljudskog eksperta iz te oblasti. Za sistem sa veštačkom inteligencijom za određenu usko-specijalizovanu oblast potrebno je skupiti, skladištiti i obraditi prikupljene podatke koji se tiču ljudskog znanja o datoj oblasti.

4.2 Periodi razvoja veštačke inteligencije

Razvoj veštačke inteligencije obuhvata sledeće periode koji se delimično preklapaju i to u zavisnosti od tehnika koje su razvijene u posmatranom periodu :

1943-1956

Ovaj period je period u kome su se javile prve ideje o veštačkoj inteligenciji. Prvi rad koji dotiče temu veštačke inteligencije i koji otvara novo polje nauke jeste rad: „*A logical calculus of the ideas immanent in nervous activity*” autora Mekkalaha i Pitsa (Warren McCulloch i Walter Pitts) [10]. Ovaj rad je značajan jer je u njemu definisan model kojim su predstavljeni neuroni u mozgu. Autori rada su posmatrali neurone u mozgu kao model mreže. U toj neuronskoj mreži svaki neuron je predstavljen sa dva stanja, odnosno kao bit (ili neuron u stanju uključenosti ili je neuron u stanju isključenosti) [10]. Njihov model je u stvari ekvivalent Tjuringovoj mašini (Turings machine) [11]. Na osnovu ovako definisane neuronske mreže, autori rada su dokazali da bilo koja matematički izračunjiva funkcija može biti izračunata pomoću mreže povezanih neurona. Iz njihovog rada sledi zaključak da proste strukture mreže mogu da uče. Model neuronske mreže je doveo, kako do teorijskog, tako i do eksperimentalnog rada na razumevanju ljudskog mozga. Naposljetku su ti eksperimenti dokazali da neuroni ne mogu biti smatrani kao bitovi, jer se oni odlikuju nelinearnim karakteristikama, pa ne mogu biti smatrani prostim uređajima sa dva stanja. Uprkos tome Mekalik je jedan od tvoraca veštačke inteligencije. On je nakon Tjuringa postavio temelje oblasti Veštačkih Neuronskih Mreža (eng. Artificial Neural Networks - ANN). Treći osnivač oblasti veštače inteligencije je bio Džon fon Nojman (John von Neumann). Nojman je imao ključnu ulogu u projektu „Menhetn“ u toku drugog sv. rata. Pored toga nadgledao je projekat Electronic Numerical Integrator and Calculator (ENIAC) univerziteta Pensilvanija (University of Pennsylvania) i ujedno je pomagao u dizajniranju Electronic Discrete Variable Automatic Computer (EDVAC)

mašine, odnosno mašine za skladištenje programa. Kao profesor na Princeton univerzitetu (Princeton University) je ohrabrivao svoja dva svršena studenta: Marvinina Minskog (Marvin Minsky) i Dina Edmondsa (Dean Edmonds) u izgradnji prvog računara baziranog na neuronskim mrežama (1951. godina).

Klod Šenon (Claude Shannon) je još jedan od pionira oblasti veštačke inteligencije. Nakon završenog MIT-a (Massachusetts Institute of Technology) pridružio se laboratorijama Bel (Bell Telephone Laboratories). Šenon je delio ideje Tjuringa o veštačkoj inteligenciji. Objavio je 1950. godine rad o mašinama koje umeju da igraju šah. U njemu je naglasio da obična partija šaha ima oko 10^{120} poteza. U tom trenutku bi računaru baziranom na fon Nojmanovoj tehnologiji bilo potrebno 3×10^{106} godina da odigra prvi potez. On je zato predložio heuristički pristup ovom problemu u cilju traženja rešenja [12].

Dalji razvoj oblasti veštačke inteligencije predstavlja trenutak kada je Džon Mekarti (John McCarthy) ubedio Minskog i Šenona da na Dartmout koledžu (Dartmouth College) oforme grupu istraživača koja će proučavati inteligenciju mašina, veštačke neuronske mreže i teoriju automata. Tu grupu je sponzorovala firma IBM. Iako je ova grupa imala samo deset istraživača, ona će postati dominantna u oblasti veštačke inteligencije u narednih dvadeset godina. Pored toga Mekarti je tvorac izraza Veštačka Inteligencija koji je prvi put upotrebio 1956. godine tokom akademske konferencije na kojoj je pričao o inteligentnim mašinama [7].

1956 – kasne šezdesete XX veka

U početku razvoja oblasti veštačke inteligencije naučnici su bili puni entuzijazma, jer su uspeali da prevaziđu stereotipe u pogledu računara. Oni su pokazali da računari nisu samo prosti uređaji koji služe kao pomoćno sredstvo pri računanju, već da mogu biti osnova nečeg mnogo značajnijeg.

Mekarti je 1958. godine je objavio rad pod imenom: „*Programs with Common Sense*” u kome je predstavio hipotetički program *Advice Taker* [13]. Ovaj program se smatra prvim sistemom koji se zasniva na znanju (eng. Knowledgebased system). Advice Taker je program koji je omogućavao rešavanje problema iz stvarnog sveta, kao što je to npr. izrada plana vožnje do aerodroma (Mekarti u radu daje baš ovaj primer). Program je bio

u mogućnosti da prihvati nova saznanja iz različitih oblasti i na taj način da proširi svoju bazu znanja, pri čemu nije bilo potrebe za reprogramiranjem koda.

Frenk Rozenblat (Frank Rosenblatt) je 1962. godine predstavio rad „*Perceptron convergence theorem*”, odnosno predstavio je algoritam koji daje mašini sposobnost da oponaša ljudski mozak u procesu prihvatanja novih ideja i odbacivanja neodgovarajućih ideja, tj. definisao je perceptron [14]. Perceptron predstavlja osnovu na kojoj se baziraju savremeni veštački neuroni. Rozenblatov Perceptron se zasniva na radu Mekkalka i Pitsa [10].

Jedan od najambicioznijih projekata u ovom periodu je projekat pod imenom *General Problem Solver* (GPS) koji su razvili Alen Njuel (Alen Newell) i Herbert Simon (Herbert Simon). U ovom projektu je razvijen program opšte namene koji omogućava računaru da oponaša ljudski pristup u rešavanju problema. Ovaj program je, verovatno, prvi program u kome je pokušano da se napravi razlika između tehnika rešavanja problema i samih podataka koji se obrađuju [15]. Tehnika koju su istraživači koristili za kreiranje programa je danas poznata kao means-ends analiza. To je analiza u kojoj onaj ko rešava problem zamišlja šta bi bio krajnji cilj rešenog problema, pa tek onda pristupa izradi koraka koji će dovesti do rešavanja tog problema [15]. Ovo je heuristički pristup rešavanju problema. Njuel i Simon su definisali postulat u kome su naveli da problem koji je u postupku rešavanja se može definisati pomoću stanja kroz koja taj problem prolazi da bi se dobilo krajnje rešenje tog problema. Na taj način se može utvrditi razlika između trenutnog stanja problema sa željenim stanjem problema. Ukoliko željeno stanje ne može da se dobije iz trenutnog stanja, tada se pristupa novom stanju koje je korak bliže željenom stanju. Proces bi se odvijao sve dok problem ne bi došao u željeno stanje, tj. u unapred definisano stanje po metodi means-ends. Skup operatora je određivao plan rešavanja problema. GPS nije bio u mogućnosti da rešava složene probleme. Program je bio baziran na formalnoj logici i kao takav je mogao da generiše beskonačan broj mogućih operatora. To dovodi do paradoksa: postojaće beskonačno mnogo mogućnosti za rešavanje datog problema, a samim tim problem neće nikada biti rešen.

U ovom periodu su autori kreirali mehanizme, odnosno opšte metode kojima bi mašina koja poseduje inteligenciju rešila neki problem na način kako bi čovek to uradio. Ti mehanizmi su bili slabi, jer su davali malo podataka o samom domenu problema.

U tom periodu je Lotfi Zadeh (Lotfi Zadeh) objavio rad „*Fuzzy sets*”. Ovaj rad se smatra osnovom teorije Fazi skupova, odnosno fazi logike koja je sredinom osamdesetih omogućila izgradnju stotina pametnih mašina i inteligentnih sistema [16].

Do početka sedamdesetih godina prošlog veka većina vlada je povukla resurse sa projekata koji su se ticali oblasti veštačke inteligencije.

Kasne 60 XX veka – rane 70 XX veka

Dva osnovna problema sa kojima su se susretali naučnici i istraživači u procesu kreiranja mašina koje bi imale veštačku inteligenciju, u ovom periodu su:

- Previše generalan pristup u rešavanju problema. Dati pristupi nisu odgovarali većini problema koji su se rešavali, jer su ti problemi bili iz različitih oblasti. Iz tog razloga je često bio pogrešno definisan sam domen problema. Rešavanje problema se izvodilo u malim „koracima”, odnosno deo po deo problema je rešavan sve dok se ne bi došlo do pravog rešenja. To je skup i dugotrajan proces čija primena je bila samo kod jednostavnijih problema.
- Većina problema koje su naučnici pokušali da reše primenom veštačke inteligencije je bila previše složena. Jedan od tih problema je bio i prevođenje naučnih i stručnih tekstova sa ruskog na engleski jezik uz pomoć veštačke inteligencije. Ovaj projekat je finansirala vlada Sjedinjenih Američkih Država i to nakon što je 1957. godine SSSR lansirao prvi veštački satelit Sputnik (rus. Sputnik) u orbitu Zemlje. Američka vlada je od ovog projekta odustala 1966. godine, jer se pokazao neefikasnim sa jedne, a sa druge strane nije mogao da „uhvati” nit misli autora, tj. fali ljudska inteligencija u tumačenju teksta koji se prevodi (emocionalan pristup problemu).

Rane 70 XX veka – sredina 80-tih XX veka

Najznačajniji doprinos ovog razdoblja veštačke inteligencije je shvatanje naučnika da domen problema mora biti dobro definisan, odnosno da moraju postojati ograničenja u zadavanju problema inteligentnim mašinama.

Program DENDRAL [17] je primer u kome je dobro definisan domen problema. Ovaj program se koristio za analizu hemijskih jedinjenja. DENDRAL je razvijen na univerzitetu Stanford (Stanford University) 1969. godine i to za podršku jednom od

programa NASA-e. NASA je želela da bude razvijen sistem koji bi bio u mogućnosti da analizira sastav soli planete Mars, jer je postojao svemirski program na osnovu koga je trebalo da bude poslata letelica na Mars radi sakupljanja uzoraka sa tla. Edvard Fajgenbaum (Edward Feigenbaum) je kao bivši student Herberta Simona, zajedno sa Brusom Bjukenenom (Bruce Buchanan) i Džošom Lederbergom (Joshua Lederberg, osvojio Nobelovu nagradu za genetiku 1958. godine), oformio tim koji je pokušao da reši problem hemijske analize uzoraka. Da su primenili tradicionalne metode u procesu rešavanja problema onda bi prvo morali da u program unesu molekulske mase svih mogućih molekularnih struktura (za koje se znalo), a tek onda bi prikupljeni uzorci bili upoređivani sa unapred unešenim vrednostima. Ovaj pristup nije mogao da da rezultat, jer je moguće generisati milione i milione mogućih molekularnih struktura i samim tim potrebno je dosta vremena da bi se izvršila analiza (čak i relativno prostih jedinjenja). Dodatni problem je što nije postojao algoritam koji je mogao da mapira molekulske mase jedinjenja i poveže ih sa molekularnom strukturom jedinjenja. Lederberg je mogao da reši ovaj problem korišćenjem svog znanja i stručnosti iz date oblasti i iskustva. Na taj način je data mogućnost svođenja ogromnog broja mogućih struktura jedinjenja, jer je Lederberg mogao iz spektrograma analiziranih uzoraka da uoči vrhove/pikove (eng. peak) koji su se podudarali sa postojećim jedinjenjima. Zadatak Fajgenbauma je bio da tu stručnost koju poseduje Lederberg unese u računarski program koji bi tada radio na nivou ljudskog eksperta. Takvi programi se nazivaju Ekspertni sistemi. U toku pisanja takvog programa Fajgenbaum je naišao na problem koji je nazvao: usko grlo tokom sticanja znanja (eng. knowledge acquisition bottleneck), a koji se ogledao u tome kako izvući svo potrebno znanje iz ljudskog eksperta i implementirati ga u program za računar. Zato je bilo potrebno da Fajgenbaum nauči nešto o hemijskoj analizi, a Lederberg da nauči nešto o računarima. Na taj način trojica naučnika su kreirala prvi Ekspertni sistem DENDRAL koji se zasnivao na sticanju znanja u toku samog rada. DENDRAL dobija još više na značaju ako se ima u vidu da je ovo prvi sistem u kome se ne polazi od opštih pretpostavki problema, već se zahteva precizno i jasno definisanje domena problema, odnosno ovo je program koji važi za usko-specijalizovanu oblast.

Sledeći značajan korak u razvoju iz oblasti ekspertnih sistema je kreiranje sistema MYCIN [18]. U ovom programu, koji je nastao u toku izrade doktorske teze, Edvarda Šortlifa (Edward Shortliffe) glavnu ulogu je, opet, imao Fajgenbaum. MYCIN je

prvenstveno bio dizajniran za otkrivanje zaraznih krvnih bolesti. Neke od karakteristika MYCIN -a su:

- MYCIN kao ekspertni sistem je mogao da da analizu krvi koja je bila na nivou doktora-početnika.
- Baza znanja koju je posedovao MYCIN se sastojala od 450 nezavisnih IF-THEN pravila koja su nastala na osnovu podataka dobijenih od ljudskih eksperata.

Pravila koja su integrisana u MYCIN su uvela i pojam nesigurnosti u pogledu postavljanja dijagnoze. MYCIN je testirao stanja tih pravila (IF deo) njihovim upoređivanjem sa dostupnim podacima. Kada je bilo potrebno MYCIN je koristio tzv. račun faktora sigurnosti (eng. certainty factors). Srž ovog računa leži u činjenici da je program do istine o nekom stanju dolazio kroz račun koji je u sebe obuhvato sve nepouzdanosti koje su mogle da se jave. Tek nakon obavljenog računa program je donosio zaključak.

Još jedan ekspertni sistem koji se pokazao kao vrlo uspešan je bio PROSPECTOR (Computer-Based Consultation System For Mineral Exploration) [19]. To je bio sistem za određivanje mineralnog sastava uzorka. PROSPECTOR je razvijen od strane istraživačkog centra Stanford univerziteta (Stanford Research Institute). Sistem je bio u upotrebi od 1974 do 1983. godine. Karakteristika ovog sistema je što je kombinovao podatke o strukturi minearala, koji su bili ugrađeni u njega, sa semantičkom mrežom. PROSPECTOR je prvo zahtevao da mu istraživač da podatke o uzorku: geološke karakteristike, strukturu, vrstu kamena ili minerala. Onda bi sistem te podatke upoređivao sa karakteristikama minerala koje ima u svojoj bazi. Ukoliko je bilo potrebe PROSPECTOR bi postavio još neko pitanje istraživaču. Na kraju bi sistem dao svoj sud o uzorku. Pored toga sistem je bio u mogućnosti da prikaže (na zahtev) sve korake koji su ga doveli do datog zaključka. Kako u geologiji postoji veliki broj pretpostavki i nesigurnosti o uzorku koji se ispituje (u pogledu porekla, sastava, strukture) PROSPECTOR primenjuje Bajesova pravila (eng. Bayes's rules) o statičkom zaključivanju koja se zasnivaju na Bajesovoj teoremi [20], da bi sagledao sve nesigurnosti koje se javljaju kod ispitivanja uzorka. U toku 1980. godine PROSPECTOR je u okolini planine Tolman (Tolman) u državi Vašington (Washington State), SAD (USA) potvrdio postojanje rude molibdena (Mo) koja je u tom trenutku vredela preko 100 miliona dolara.

U ovom periodu su ekspertni sistemi razvijani uz pomoć specijalnih programskih jezika kao što su: LISP, PROLOG, OPS. Programiranje ekspertnih sistema ovim jezicima zahteva što bolji hardver, a u vreme početka razvoja ekspertnih sistema takav hardver je bio skup. Iz tog razloga su se samo određene istraživačke grupe bavile razvojem ekspertnih sistema i veštačke inteligencije. To su bile grupe na univerzitetu Stanford, MIT-u i Karnegi-Melon univerzitetu (Carnegie-Mellon University).

Tek osamdesetih godina XX veka kada je došlo do pojave PC-jeva i alata koji su se lako koristili (eng. easy-to-use) i koji su skrivali svu kompleksnost rada i izrade ekspertnih sistema, tzv. šelova/ljuski (eng. shell), je omogućeno svim zainteresovanim istraživačima iz različitih oblasti da učestvuju u razvoju ekspertnih sistema.

Ekspertni sistemi ne mogu biti smatrani inteligentnim mašinama koje bi mogle da polože Turingov test, jer:

- su oni kreirani samo za usko-specijalizovane oblasti.
- ekspertni sistemi nisu fleksibilni u meri koju bi korisnik želeo, jer se fokusiraju samo na određeni domen problema.
- Ovi sistemi mogu delimično da prikažu korake koji su doveli do rešenja, ali ne mogu da prikažu heuristički pristup u rešavanju problema.
- Ekspertni sistemi teško mogu da se koriste za verifikaciju i validaciju obrađenih podataka. Još uvek nije razvijena nijedna tehnika koja može da potvrdi u potpunosti njihovu tačnost i konzistentnost u davanju rezultata.
- Ekspertni sistemi (pogotovo prve generacije) poseduju vrlo malu sposobnost učenja iz dobijenih rezultata, odnosno iz iskustva.

Uprkos svim nedostavima, Ekspertni sistemi su napravili proboj u oblasti veštačke inteligencije i imaju veliku primenu. Tako je 1986. godine postojalo 200 ekspertnih sistema [21] , a već 1994. godine već 2500 [22] ekspertnih sistema.

Sredina 80-ih XX veka – sadašnjost

Sredinom osamdesetih godina prošlog veka, istraživači su zaključili da izgradnja ekspertnih sistema zahteva dosta resursa, pa su čak predviđali da će doći do obustave projekata koji su se ticali veštačke inteligencije. To je bio glavni razlog što su se

istraživači okrenuli novom-starom principu, a to su neuronske mreže. Činjenica je da su skoro sve ideje o neuronskim mrežama kreirane još 60-ih godina XX veka, ali tek sredinom 80-ih godina XX veka su pronađena tehnička rešenja koja su mogla da omogućće primenu tih ideja. Glavni razlog za to je pojava PC računara koji su bili dovoljno jaki da omogućće modeliranje i rad sa veštačkim neuronskim mrežama.

Grozberg (Grossberg) je ustanovio nove principe o samoorganizaciji, što je predstavljalo osnovu za novu klasu neuronskih mreža [23].

Hopfield (Hopfield) je predstavio neuronske mreže sa povratnom spregom/fidbek (eng. feedback) [24]. Kod ovih mreža signali putuju u oba smera stvarajući petlje, što znači da su ovo dinamične mreže. Kao dinamičke mreže njihovo stanje se stalno menja i to sve dok ne dostignu tzv. tačku ekvilibrijuma/ravnoteže (eng. equilibrium). Neuronska mreža ostaje u ravnoteži sve dok na ulaz (eng. input) ne dođu signali koji će je izbaciti iz ravnoteže. Tada neuronska mreže ponovo pokušava da dostigne stanje ravnoteže. Ovo su veoma moćne mreže koje mogu, u zavisnosti od složenosti problema, da postanu i previše kompikovane.

Pravi proboj na polju veštacke inteligencije u ovom periodu se desio kada su Rumelhart (Rumelhart) i Meklelend (McClelland) 1986. godine podsetili naučnu zajednicu na algoritam pod nazivom: back-propagation learning algorithm [25]. Ovaj algoritam su 1969. godine prvi predstavili Brajson (Bryson) i Ho (Ho) [26]. Otprilike u isto vreme (1988. godine) Kun (Cun) takođi uvodi na velika vrata ovaj isti algoritam [27]. Od tada je primena ovog algoritma najzastupljenija kod programiranja višeslojnih Perceptrona.

Brumhed (Broomhead) i Louv (Lowe) su 1988. godine su pronašli proceduru koja je omoguććavala dizajn slojevitih fidforvard (eng. feedforward) mreža [28]. Ove mreže se sastoje od velikog broja prostih neurona (jedinica za obradu podataka) koji su organizovani u slojeve. Svaka od jedinica sloja je povezana sa svim jedinicama prethodnog sloja. Ove veze nisu međusobno jednake: svaka veza može imati različitu važnost. Upravo ta različitost u važnosti veza omoguććava unošenje znanja u mrežu. Veoma često se ove jedinice u ovoj vrsti neuronske mreže nazivaju čvorovi (eng. nodes). Podatak ulazi kao input i prosleđuje se kroz mrežu i to sloj po sloj, sve dok ne dođe do izlaza. Tokom normalne operacije, odnosno kada se ova mreža ponaša kao klasifikator (podela datog skupa elemenata u dve grupe i to na osnovu zadatog pravila klasifikovanja),

ne postoji povratna sprega između slojeva. Od te činjenice potiče i ime ove mreže. Skriveni (eng. hidden) sloj je sloj u kome se podaci sa ulaza, tj. inputi transformišu u pogodan oblik koji izlazni sloj (eng. output layer) može da obradi. Operacije koje se odigravaju u ovoj fazi mogu biti podeljene u dve faze:

- fazu klasifikovanja
- fazu učenja

U fazi klasifikovanja se ulazni signali razvrstavaju prema međusobnim sličnostima u grupe manjih veličina. To omogućava samoj neuronskoj mreži da utvrdi konačan broj podcelina (koje zajedno čine jednu celinu primljenih ulaznih signala u datom vremenskom trenutku). Novoprimljeni ulazni signali će biti, na osnovu svojih karakteristika, pridodati jednoj od već formiranih grupa ulaznih signala.

U fazi učenja je moguće izvršiti delimičnu ili potpunu rekonstrukciju primljenog ulaznog signala na osnovu pretpostavki koje su utvrđene u fazi klasifikovanja ulaznih signala. U dinamičnom okruženju same neuronske mreže može doći do promene primljenog i obrađenog ulaznog signala, pa će taj signal biti preusmeren ka nekoj drugoj izlaznoj grupi signala.

Rane 70-te XX veka – sadašnjost

Prirodna inteligencija je proizvod evolucije. Ovim pristupom je pokušana simulacija biološke evolucije u cilju shvatanja kako živi sistemi stiču visok stepen inteligencije. U prirodi živa stvorenja uče radeći određene stvari. Živim stvorenjima nije rečeno kako da se prilagođavaju određenom okruženju, ona se bore za opstanak. Vrste koje se lakše prilagođavaju imaju više šansi za opstanak, a samim tim imaju mogućnost za prosljeđivanje svog genetskog materijala budućim generacijama.

Evolucionni pristup pri kreiranju veštačke inteligencije je zasnovan na izračunljivim modelima prirodne selekcije i genetike. Evoluciono izračunavanje funkcioniše tako što se simulira populacija pojedinaca i daje se evaluacija njihovih karakteristika, zatim se opet generiše neka nova populacija i posmatraju se karakteristike pojedinaca te populacije. Proces se odvija određeni broj puta. Evoluciono izračunavanje u sebe uključuje tri tehnike: genetičke algoritme, strategije evolucije i genetičko programiranje. Ovaj koncept je predložen od strane Holanda (Holland) 1975. godine [29]. Holand je

napravio algoritam za manipulaciju veštačkih „hromozoma”, odnosno niza koji je sastavljen od binarnih cifara, koristeći genetičke operacije kao što su: selekcija, odstupanja i mutacije. Odstupanja u ovom algoritmu predstavljaju genetički operator koji se koristi radi programiranja varijacija u hromozomima (od generacije do generacije). Evolucionarne strategije su razvijene specijalno za problem optimizacije parametara. Ove strategije u stvari predstavljaju alternativu inženjerskoj intuiciji. One koriste postupke numeričke optimizacije.

Prednost ovog pristupa u kreiranju mašina sa veštačkom inteligencijom je u činjenici što su ovi algoritmi primenjivi na širok spektar problema. Rešenja genetskog modela su često obimna, ali i vrlo pouzdana. Genetsko programiranje predstavlja aplikaciju genetskog modelovanja koja ima sposobnost da uči tokom izvršenja zadatka. Cilj ove vrste modelovanja je da genetsko programiranje dovede do generisanja računarskog programa koji će sam dalje da uči, tj. da se razvija. Veliki doprinos genetskom programiranju dao je Džon Koza (John Koza) . On je koristio genetske operacije za manipulaciju simboličnim kodom koji se koristi u LISP programima [30].

Genetsko programiranje nudi rešenje za glavni problem u kompjuterskoj nauci, a to je: kreiranje kompjutera koji mogu da rešavaju novonastale i nepredvidive probleme.

Kasne 80-te XX veka – sadašnjost

Tehnologija neuronskih mreža daje znatnu interakciju sa realnim svetom u odnosu na sisteme koji su zasnovani na simboličnom rezonovanju. Neuronske mreže mogu da uče, da se prilagođavaju promenama u okruženju problema, da uspostavljaju obrasce u situacijama gde ne postoje jasno definisana pravila i da koriste nejasne i nepotpune podatke. Sa druge strane razvoj neuronskih mreža sa trenutnom tehnologijom je dosta spor.

Veštačke neuronske mreže i ekspertni sistemi su tehnologije koje se međusobno nadopunjuju, odnosno ne postoji međusobna konkurentnost između ovih tehnologija. Tamo gde ne postoji dovoljno znanja o načinu rešavanja problema ili o domenu problema, neuronske mreže mogu da iz datih podataka dobiju odgovarajuće znanje koje ekspertni sistemi mogu da iskoriste.

Oblast koja može da se suoči sa nejasnim, nepreciznim i nepouzdanim znanjem je fazi logika [24]. Većina metoda koja se u klasičnim ekspertskim sistemima primenjuje, kada

je u pitanju nepreciznost, se zasniva na konceptu verovatnoće. Stručnjaci iz nematematičkih oblasti, uglavnom, ne koriste termine i izraze koji se koriste u teoriji verovatnoće, već koriste termine i izraze iz svakodnevne upotrebe, kao što su: često, uopšteno, ponekad, sa vremena na vreme i retko. Fazi logika se zasniva na upotrebi nejasnih vrednosti koje zadržavaju izvorno značenje upotrebljenih reči koje ljudi koriste tokom donošenja odluka. Upravo zahvaljujući toj činjenici fazi logika izbacuje usko grlo koje se javlja kod tradicionalnih ekspertnih sistema.

Srž fazi logike je u uspostavljanju koncepta jezičke/lingvističke varijabile. Vrednosti ove varijabile su mnogo češće reči, a ne brojevi. Slično ekspertnim sistemima, fazi logika koristi IF-THEN pravila da bi ugradila ljudsko znanje, ali upotreba tih pravila je „zamašnjak”, tj. fazi, kao na primer:

- AKO (IF) je brzina velika, TADA JE (THEN) je zaustavna putanja duga.
- AKO (IF) je brzina mala, TADA JE (THEN) je zaustavna putanja kratka.

U ovom primeru nije precizno definisan prag brzine kada će biti smatrano da auto ide prebrzo/sporo.

Koncept fazi logike je bio zanemaren na zapadu, ali su zato japanci njemu pridali značaj. Fazi logika se u Japanu primenjuje još od 1987. godine i osnova njenog koncepta se nalazi u automobilima, veš-mašinama, klima uređajima, televizorima.

Prednosti fazi logike su:

- Poboljšana moć izračunavanja: sistemi zasnovani na fazi logici su znatno brži i zahtevaju manje pravila od konvencionalnih ekspertnih sistema. Fazi ekspertni sistemi integrišu pravila, čineći na taj način sam sistem znatno jačim.
- Poboljšano kognitivno modelovanje: Fazi sistemi omogućavaju ugrađivanje znanja u formi u kojoj su to znanje predstavili ljudski eksperti. Ljudski eksperti veoma često misle i izražavaju se na nekonvencionalan i nematematički način.
- Tradicionalni ekspertni sistemi predstavljaju sisteme koji zahtevaju dobro definisan domen problema, odnosno primenjivi su u usko-specijalizovanoj oblasti. Potreban je jedan ljudski ekspert iz date oblasti koji će na svoj način definisati domen datog problema. To je veoma skupo, jer izabrani stručnjak mora biti jedan od najboljih u toj oblasti. Za razliku od ekspertnih sistema, fazi sistemi

mogu da koriste više ljudskih eksperata u definisanju domena problema, iako oni neće istim rečnikom definisati taj domen problema niti će, možda, usaglasiti stavove povodom datog problema. Ovo je naročito bitno kod složenih problema kao što su problemi koji se tiču poslovanja i upravljanja poslovnim procesima.

Glavni nedostatak fazi sistema je što u potpunosti zavisi od pravila koja definišu eksperti. Ta pravila mogu biti dobro definisana, ali sa druge strane mogu biti i loše definisana što može dovesti do zastoja i problema u radu samog sistema. Fazi sistem može biti napravljen relativno brzo i relativno jednostavno, ali zato može da prođe dosta vremena dok se definisana pravila ne podese odgovarajućem problemu. To je proces koji traje i za čije odvijanje moraju biti isprobane razne varijacije tih definisanih pravila, tj. mora biti izvršeno fino podešavanje sistema.

5 Neuronske mreže

5.1 Pojam neurona

Koncept veštačkih neuronskih mreža datira od sredine pedesetih godina XX veka. Osnovu veštačkih neuronskih mreža čine algoritmi dubokog učenja (eng. Deep learning). Duboko učenje je skup tehnika u okviru mašinskog učenja (eng. Machine learning) koje omogućavaju mašinama ljudski pristup rešavanju problema, tj. omogućavaju učenje kroz primere. Neki od primera algoritama dubokog učenja su: su prepoznavanje slika, obrada prirodnog jezika i prepoznavanje glasa.

Neuroni u mozgu sisara, tj. biološki neuroni predstavljaju vrstu nervne ćelije, koja za razliku od ostalih ćelija, ima sposobnost da prima (i šalje) podatke i obrađuje ih u informacije. U ljudskom mozgu ima oko $86 \cdot 10^9$ neurona [31]. Glavni delovi biološkog neurona sisara su [32]:

- telo neurona ili soma u kome se nalazi nukleus,
- nukleus koji predstavlja osnovni deo neurona, a njegova svrha je da pokreće sve aktivnosti u okviru neurona u kome se nalazi.
- dendriti (starogrčki: δένδριτης) koji služe za primanje podataka od drugih neurona,
- aksoni (starogrčki: axon) koji služe za slanje podataka ka drugim neuronima.
- terminali koji predstavljaju tačke gde se spajaju aksoni jednog neurona sa aksonima drugog neurona.

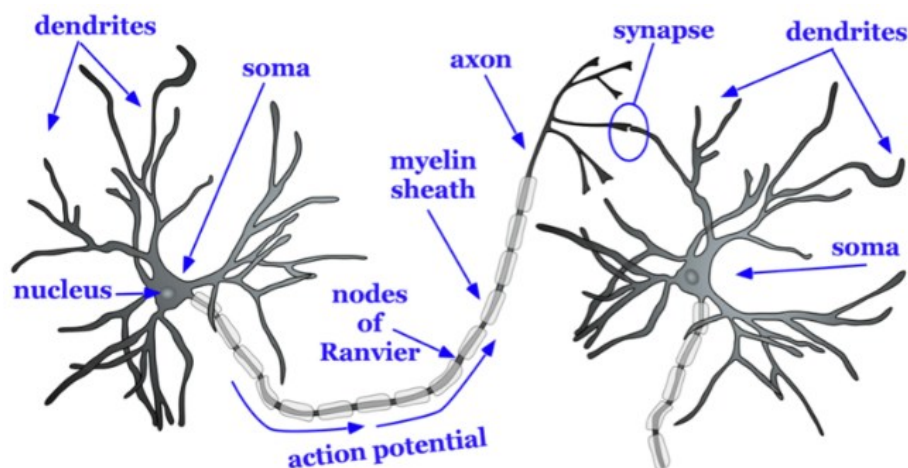
Na osnovu gore navedenog neuroni se mogu smatrati samostalnim procesorima za obradu podataka.

Pored ova tri osnovna dela neurona, postoje još tri jako bitna dela neurona [32]:

- neuronska membrana: okružuje neuron, a lipidi i proteini koji čine tu membranu određuju koje supstance mogu da pristupe neuronu,
- sinapse: veoma mali prostor između spojenih neurona. Sinapse omogućuju prenos električnih ili hemijskih signala ka drugom neuronu ili mišićnoj ćeliji, pa sinapse nisu ništa drugo do specijalizovani međućelijski komunikacioni kanali.

- mijelinski omotač koji se sastoji od masti i on okružuje aksone (služi kao izolator). Masti koje čine mijelinski omotač nastaju u procesu lučenja oligodendro ćelija (starogrčki: ολίγος + δένδρον + κτύιον) koje se još nazivaju i Švon (nazvane po nemačkom psihologu Theodor Schwann-u) ćelije. Mijelinski omotač je periodično prekinut malim prostorima koji se nazivaju Ranvier čvorovima. Ovi čvorovi ubrzavaju signal kroz akson. Mijelinski omotač je odličan izolator, ali će kao takav dovesti do usporenja signala koji se kreće kroz akson.

Kada je biološki neuron neaktivan razlika u električnom potencijalu preko membrane neurona je u granicama od -60 mV do -80 mV [33]. Taj potencijal se naziva potencijal mirovanja. Ulazi u biološke neurone se menjaju u vremenu, pa ljudski mozak ima sposobnost učenja na osnovu prethodnih iskustava. Kod veštačkih neuronskih mreža se teži principu oponašanja ljudskog mozga u pogledu učenja na sopstvenom iskustvu.



Slika 5.1.1. Izgled i povezanost neurona u ljudskom mozgu (izvor: [34])

Osnovna razlika koja se javlja između ljudskog mozga i računara koji se zasniva na arhitekturi Džona fon Nojmana (John von Neumann) [35] je u načinu tretiranja podataka.

Ljudski mozak nije u stanju, za razliku od računara, da:

- prima ogromne količine nestruktuiranih podataka koji naizgled nemaju nikakvu međusobnu povezanost,
- brzo obrađuje veliku količinu podataka i da osnovu tako obrađenih podataka donese relevantnu i kvalitetnu odluku.

Sa druge strane ljudski mozak može da:

- izvede neke nove zaključke na osnovu nekog prethodnog iskustva,
- izvede, sasvim spontano, neke nove zaključke bez prethodnog iskustva.
- da radi iako njegov dobar deo može da bude oštećen, što kod računara nije slučaj (računar ne može da radi ako je CPU oštećen)

Aktivnostima bioloških neurona u ljudskom mozgu koordiniraju oscilacije samih neurona [36]. Zahvaljući prostorno-vremenskim karakteristikama oscilacija neurona može biti utvrđeno na koji način neuroni međusobno koordiniraju u različitim regijama ljudskog mozga. Na taj način neuroni kreiraju ponašanje ljudskog bića u datom dinamičnom okruženju [37]. Frekvencija gama talasa se, u proseku, kreće u opsegu od 30 Hz do 100 Hz. Gama talasi koji se beleže u ljudskom mozgu su talasi koji predstavljaju brzinu kojom neuron obrađuje primljeni stimulans i vreme potrebno za davanje odgovora na taj ulazni stimulans. Ovi talasi omogućavaju izvršavanje određenih kognitivnih, tj. motoričkih funkcija. Značaj gama talasa se ogleda u tome što oni omogućuju davanje izlaza na ulazne emocionalne stimulanse [38]. Frekvencija prosečnog PC računara (u GHz) je višetrukostruko veća u odnosu na frekvenciju gama talasa bioloških neurona ljudskog mozga. Vreme odziva računara na ulazni signal je nekoliko nanosekundi u odnosu na vreme odziva ljudskog mozga na ulazni stimulans koje se meri u mikrosekundama. Prednost bioloških neurona je u zajedničkom dejstvu prilikom obrade složenog stimulanse što znatno ubrzava proces obrade primljenog signala i davanja odgovora na njega. Biološki neuroni su u stanju da paralelno obrađuju isti stimulans različitim pristupom tom problemu tokom procesa njegove obrade, tj, moguće je izvršiti paralelizam odvijanja različitih procesa tokom obrade jednog primljenog signala. Kod računara se samo prividno ostvaruje proces paralelizacije obrade primljenih signala.

Na osnovu prethodnih razmatranja može se uočiti potreba da rad kreiranih veštačkih neuronskih mreža treba da bude sličan radu ljudskog mozga, tj. treba omogućiti gusto pakovanje veštačkih neurona koji će međusobno biti povezani. Ovaj postupak stvaranja milijardi veštačkih neuronskih ćelija koje su međusobno povezane po principu rada ljudskog mozga je skup. Zato se u svrhu kreiranja veštačkih neuronskih mreža koriste savremeni računari, odnosno preciznije, koriste se logička kola koja su serijski povezana. Savremena nauka još uvek ne može da da odgovore na puno pitanja koje se tiču samih principa rada ljudskog mozga, ali je i to znanje koje postoji bilo dovoljno za kreiranje modela veštačkih neuronskih mreža.

Za razliku od ljudskog mozga računar radi na osnovu unapred zadatih algoritama koje je kreirao ljudski mozak, te nema sposobnost sopstvene kreativnosti u toku procesa donošenja odluka, tj, računar radi po determinističkim principima. Iz navedenog sledi da veštačke neuronske mreže predstavljaju skup programa, tj. simulaciju koja u određenim segmentima oponaša rad ljudskog mozga.

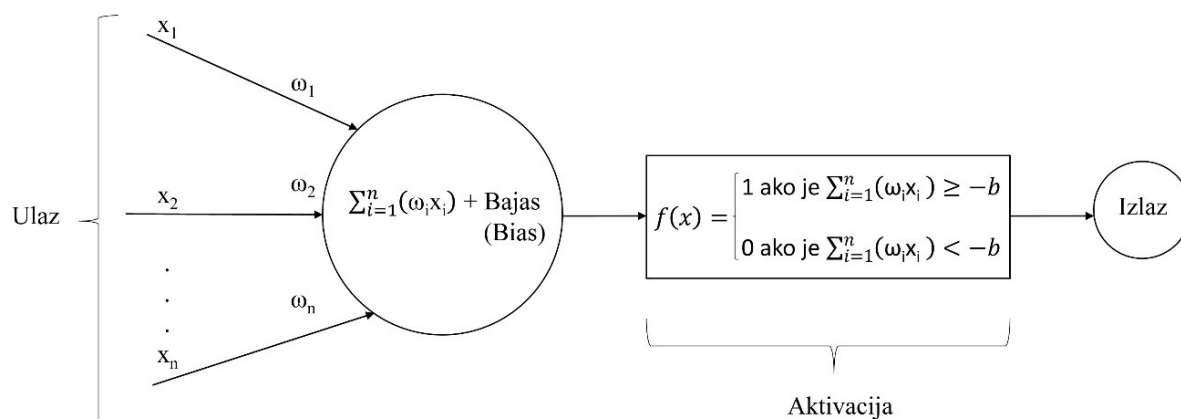
5.2 Perceptron

Princip rada Perceptrona se zasniva na principu rada neurona u ljudskom mozgu. Perceptron ima nekoliko binarnih ulaza: x_1, x_2, \dots, x_n , a kao rezultat operacija koje on izvodi dobija se jedan binarni izlaz. Ukoliko se ulazni signali direktno prevode u izlazne signale korišćenjem samo jednog sloja veštačkih neurona, onda se takav perceptron naziva jednoslojni perceptron. Sledi tabela koja prikazuje paralelu između biološkog neurona i veštaškog neurona.

Tabela 5.2.1. Paralela između biološkog i veštačkog neurona

Neuron kod ljudskog mozga	Veštački neuron
Dendriti	Ulaz koji prima signale
Soma (u kome se nalazi jezgro ćelije)	Čvorovi
Koeficijenti težine ili međuveze	Prenos ulaznih podataka ka neuronu
Aksoni	Izlaz

Na slici koja sledi je predstavljen jednoslojni perceptron.



Slika 5.2.1. Jednoslojni perceptron

Rozenblat [14] je predložio jednostavno pravilo za izračunavanje rezultata tako što je koristio koeficijente težine za definisanje značajnosti ulaznih signala koje je označavao sa: $\omega_1, \omega_2, \dots, \omega_n$. Ti koeficijenti težine ulaznih signala su realni brojevi čija vrednost odgovara značaju primljenih ulaznih podataka koji će biti obrađeni, a rezultati te obrade prosleđeni ka izlazu.

Izlaz neurona može biti 0 ili 1 i to u zavisnosti da li je $\sum_{i=1}^n (\omega_i x_i) + b$ veće od granične vrednosti (dozvoljenog praga prelaza) ili ne. Granična vrednost ili prag prelaza je, takođe, deo neurona. Izlaz neurona se predstavlja formulom:

$$f(x) = \begin{cases} 1 & \text{ako je } \sum_{i=1}^n (\omega_i x_i) \geq \text{prag prolaza} & (1a) \\ 0 & \text{ako je } \sum_{i=1}^n (\omega_i x_i) < \text{prag prolaza} & (1b) \end{cases}$$

Bajas (Bias) ili stav je unapred definisani broj. Prag prolaza je negativna vrednost Bajasa. Iz formule (1a) sledi da ukoliko je suma proizvoda ulaza i njihovih težinskih koeficijenata veća ili jednaka pragu prolaza dolazi do primene funkcije aktivacije. U suprotnom neće doći do primene funkcije aktivacije (formula (1b)). Funkcija aktivacije može biti step funkcija, sigmoidna funkcija ili neka druga funkcija.

Perceptron polazi od pretpostavki koje su unapred definisane određenim koeficijentom težine, što je dato sledećim primerom:

- Ukoliko neki fakultet želi da vidi da li će krenuti u proces akreditacije studiranja na daljinu (eng. Distance learning) potrebno je izvršiti analizu tržišta. Neophodno je dati odgovore na veliki broj pitanja koja su međusobno povezana. U ovoj analizi je dati postupak pojednostavljen i postoje tri ulazna signala. Kako odgovor na svako od postavljenih pitanja može biti Da ili Ne, to su vrednosti tih ulaznih signala binarne, tj. 1 ili 0. Nakon ovako određenih vrednosti ulaznih signala, potrebno je utvrditi za svaki od ulaznih signala njegov koeficijent težine ($\omega_1, \omega_2, \omega_3$).

Tabela 5.2.2. Koeficijenti težine

Pitanje	Da li željeni fakultet studiranje na daljinu		Da li je student zainteresovan da uči na daljinu		Da li postoji adekvatna oprema za učenje na daljinu	
Odgovor	Da	Ne	Da	Ne	Da	Ne
Binarna vrednost	1	0	1	0	1	0
Koeficijent težine	5		4		3	

Neka je, po pretpostavci, prag prolaza 6, odnosno vrednost bajasa je -6.

Dobijeni su odgovori koji su prikazani u tabeli koja sledi:

Tabela 5.2.3. Odgovori studenta

Redni broj pitanja	Student 1	Student 2	Student 3
1	1	1	0
2	1	1	0
3	1	0	1

Iz formule $\sum_{i=1}^n (\omega_i x_i)$ sledi za svakog od studenata ponaosob rezultat ankete:

Tabela 5.2.4. Rezultati ankete

	Student 1	Student 2	Student 3
$\sum_{i=1}^n (\omega_i x_i)$	$1 * 5 + 1 * 4 + 1 * 3 = 12$	$1 * 5 + 1 * 4 + 0 * 3 = 9$	$0 * 5 + 0 * 4 + 1 * 3 = 3$
$\sum_{i=1}^n (\omega_i x_i) + b$	$12 - 6 = 6$	$9 - 6 = 3$	$3 - 6 = -3$

Student 1 i Student 2 su prošli prag prolaznosti, dok student 3 nije. Odavde se uočava značajnost bajasa kao unapred zadatog broja čija vrednost ne zavisi od ulaznih veličina koje će se obrađivati. Od vrednosti bajasa će zavisiti i prag prolaznosti. Ukoliko je prag prolaznosti nizak, verovatnoća, da će izlaz biti 1 i da će doći do pokretanja funkcije aktivnosti, raste. Što bajas ima veću pozitivnu vrednost to će perceptron lakše da kao izlaz

da vrednost jedan, u suprotnom što bajas ima negativniju vrednost verovatnoća da će perceptron kao izlaz dati nulu se povećava.

Menjanjem koeficijenata težine i praga prolaznosti mogu se dobiti različiti modeli za odlučivanje. Da je u gornjem primeru uzet prag prolaznosti čija je vrednost četiri, tada bi porasla mogućnost dobijanja pozitivnog ishoda.

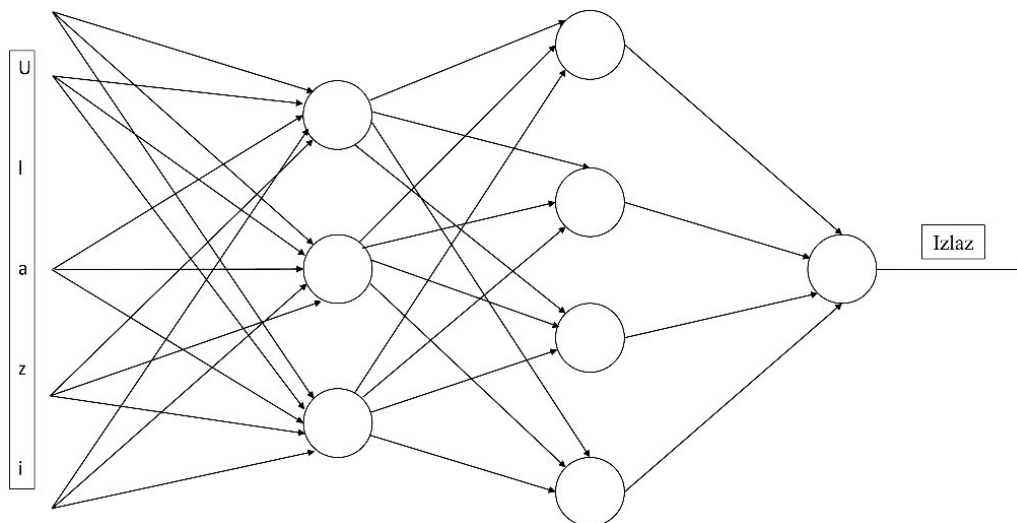
Postoje dva tipa mreža Perceptrona:

- Jednoslojne mreže
- Višeslojne mreže

Mreže perceptrona sa jednim slojem mogu da uče samo po linearnom obrascu.

Mreže perceptrona sa višeslojeva ili napredne neuronske mreže sa dva ili više slojeva perceptrona imaju veću procesorsku snagu.

Na osnovu gore urađenog primera se može zaključiti da perceptron ne predstavlja kompletan model ljudskog odlučivanja, ali sa druge strane ovaj primer pokazuje kako perceptron može da izmeri različite vrste dokaza kako bi donosio odluke. Da bi odluka koju perceptron donese bila približna odluci koju donosi ljudsko biće, potrebna je složena mreža perceptrona koja će donositi visoko-rafinirane odluke.



Slika 5.2.1. Višeslojna mreža perceptrona

Sa gornje slike se uočava da svaki perceptron donosi jednu odluku na osnovu obrađenog ulaza određenog koeficijenta težine, ali ta odluka može biti poslata većem

broju perceptrona koji se nalaze u sledećem sloju mreže perceptrona. U ovoj mreži, prva kolona perceptrona, odnosno prvi sloj perceptrona donosi tri jednostavne odluke u zavisnosti od koeficijenta težine ulaznih veličina. Perceptroni drugog sloja razmatrane mreže perceptrona donose odluke na osnovu procenjivanja rezultata prvog sloja perceptrona. Na ovaj način perceptroni drugog sloja mogu doneti složenije odluke koje su na apstraktnijem nivou u odnosu na odluke koje donose perceptroni u prvom sloju. U posmatranoj mreži perceptrona najsloženije odluke će doneti treći sloj perceptrona u kome se nalazi jedan perceptron. Na ovaj način je omogućeno donošenje fino-rafinisanih odluka.

Perceptroni se mogu koristiti i za izračunavanje elementarnih logičkih funkcija:

- I (AND)
- ILI (OR),
- NI ili NE-I (NAND).

NI kolo je univerzalno logičko kolo, jer su svi izlazi tačni, sem u slučaju kada na ulaz dolaze signali kao dve logičke jedinice. U tom slučaju da bi izlaz iz perceptrona bio jedinica koristiće se I kolo. Ovo sve važi u slučaju kada je potrebno da funkcija aktivacije ima vrednost jedan, odnosno potrebno je da ta funkcija aktivira određeni sistem.

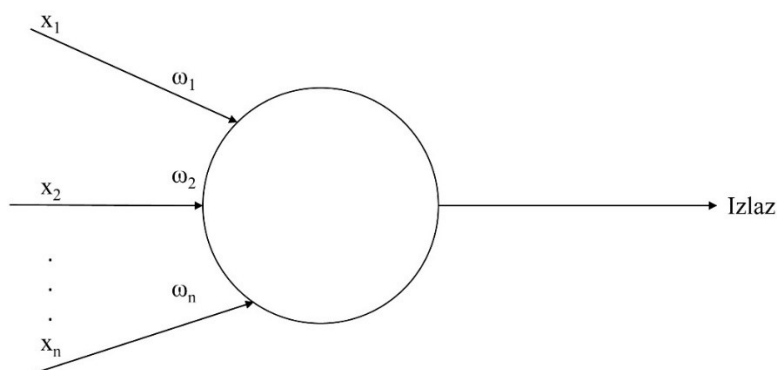
Mrežom perceptrona je moguće izvršavati bilo koje računске operacije, ali isti ne treba mešati sa NAND kolima i tretirati ga poboljšanu verziju NAND kola.

Kreirani algoritmi za učenje mogu automatski podesiti koeficijente težina i bajas za mrežu perceptrona, tj. mreže veštačkih neurona. Mreža perceptrona može sopstvenim automatskim podešavanjima da reaguje na nove stimulanse okoline koje prima, pri čemu programer(i) ne učestvuje(u) u novim podešavanjima mreže, već sama mreža perceptrona se prilagođava novoprimitljenim stimulansima. To je jedna od glavnih odlika mreže perceptrona i predstavlja osnovnu razliku u odnosu na NAND kola. NAND logička kola za primljene nove stimulanse zahtevaju dodavanje novih logičkih kola, dok kod mreže perceptrona, odnosno veštačke neuronske mreže, je moguće prevazići eventualnu pojavu problema u toku rešavanja zadatka metodom učenja i prilagođavanja neuronske mreže dinamičnom okruženju u kome se nalazi.

5.3 Sigmoid

Na osnovu poslednjeg pasusa prethodnog podpoglavlja može se uočiti problem u promenama koje se mogu javiti ukoliko dođe do promene koeficijenta težine stimulanasa koje veštačka neuronska mreža prima ili ako dođe do promene bajasa. Potrebno je odrediti promenu koeficijenta težine (oznaka: $\Delta\omega$), odnosno promenu bajasa, kako bi veštačka neuronska mreža mogla i dalje da reaguje u skladu sa unapred zadatim zahtevima koji su definisani u algoritmu ugrađenom u posmatranu veštačku neuronsku mrežu. Najmanja promena ulaznog signala, tj. stimulansa u perceptrone, koji čine veštačku neuronsku mrežu, će biti detektovana i to će dovesti do promene izlaza u vrednosti akumuliranih promena ulaznih signala koje primaju perceptroni u odnosu na prethodno stanje. Primer je ljudski rukopis, gde i najmanja promena u stilu pisanja može dovesti do toga da veštačka neuronska mreža neće prepoznati alfanumerik ili određeni, unapred definisani, simbol ili se može desiti da ih pogrešno prepozna. Potrebno je napraviti veštačku mrežu koja može samostalno da se prilagodi promenama u okruženju. Mreža perceptrona, na primer, treba da prepozna rukom pisan izraz $2 + 3$ i da kao rezultat da zbir ova dva broja. Može se javiti greška u očitavanju broja dva, koji može biti napisan slično kao broj pet, pa kao rezultat rada mreže perceptrona može se dobiti zbir koji iznosi pet. Ukoliko bi se postupak ponovio moguće je kao zbir dobiti i broj osam, što zavisi od toga kako je prvi perceptron u datoj mreži perceptrona prepoznao broj dva.

Da ne bi došlo do ove pojave mreža veštačkih neurona mora da uči i to na sopstvenim greškama koje pravi. U tu svrhu je potrebno koristiti SIGMOID neurone, gde reč sigmoid znači „u obliku razvučenog slova S”. Sigmoid se šematski predstavlja isto kao i perceptron (Slika 7).



Slika 5.3.1. Sigmoid

Za razliku od perceptrona sigmoid može imati izlaze u rasponu od 0 do 1, tj. izlaz može imati vrednosti u opsegu: $0 < Izlaz < 1$. Kod perceptrona je moguće imati samo dve vrednosti izlaza: nula ili jedan. Sigmoid ima i koeficijente težine i bias. Različiti oblici jedne iste formule, koja definiše izlaz kod sigmoida, su:

$$\sigma(z) = \frac{1}{1+e^{-(\sum_j^n (x_j * \omega_j) + b)}} \quad (2a)$$

$$\sigma(z) = \frac{1}{1+e^{-\sum_j^n (x_j * \omega_j) - b}} \quad (2b)$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (2c)$$

Sigmoid se naziva neuron logistike, a sigmoidna funkcija funkcija logistike.

Ukoliko $z \rightarrow +\infty$, po formuli (2a) biće:

$$\sigma(z) = \frac{1}{1+e^{-(+\infty)}} \quad (3a)$$

$$\sigma(z) = \frac{1}{1+e^{-\infty}} \quad (3b)$$

$$\sigma(z) = \frac{1}{1+\frac{1}{e^{\infty}}} \rightarrow 0 \quad (3c)$$

$$\sigma(z) = \frac{1}{1} \quad (3d)$$

$$\sigma(z) = 1 \quad (3e)$$

Radi lakšeg razmatranja u prethodnim formulama (3a – 3e) je uzeto da $z \rightarrow +\infty$. U konačnim, tj. diskretnim stanjima mora biti definisana konačna vrednost broja kome će z težiti, tako da se posmatra dovoljno veliki pozitivan broj. Na osnovu te aproksimacije iz formule (3e) sledi da $\sigma(z) \approx 1$. Ova izlazna vrednost je slična izlaznoj vrednosti koju može da da perceptron, a to je jedan.

Sa druge strane ako $z \rightarrow -\infty$, po formuli (2a) biće:

$$\sigma(z) = \frac{1}{1+e^{-(-\infty)}} \quad (4a)$$

$$\sigma(z) = \frac{1}{1+e^{+\infty}} \quad (4b)$$

$$\sigma(z) = \frac{1}{1+\infty} \quad (4c)$$

$$\sigma(z) = \frac{1}{\infty} \quad (4d)$$

$$\sigma(z) = 0 \quad (4e)$$

Radi lakšeg razmatranja u prethodnim formulama (4a – 4e) je uzeto da $z \rightarrow -\infty$. Kod konačnih tj. diskretnih stanja mora biti definisana konačna vrednost broja kome će z težiti, tako da se posmatra dovoljno veliki negativan broj. Na osnovu te aproksimacije iz formule (4e) sledi da $\sigma(z) \approx 0$. Ova izlazna vrednost je slična izlaznoj vrednosti koju može da da perceptron, a to je nula. Slede kod (sa komentarima) i grafik sigmoidne funkcije. Kao okruženje za pisanje koda je korišćen Sublime Text 3 editor. Ovaj editor je korišćen za sve kodove u okviru doktorske disertacije. Kodovi koji su prikazani u doktorskoj disertaciji su kopirani direktno iz editora.

```

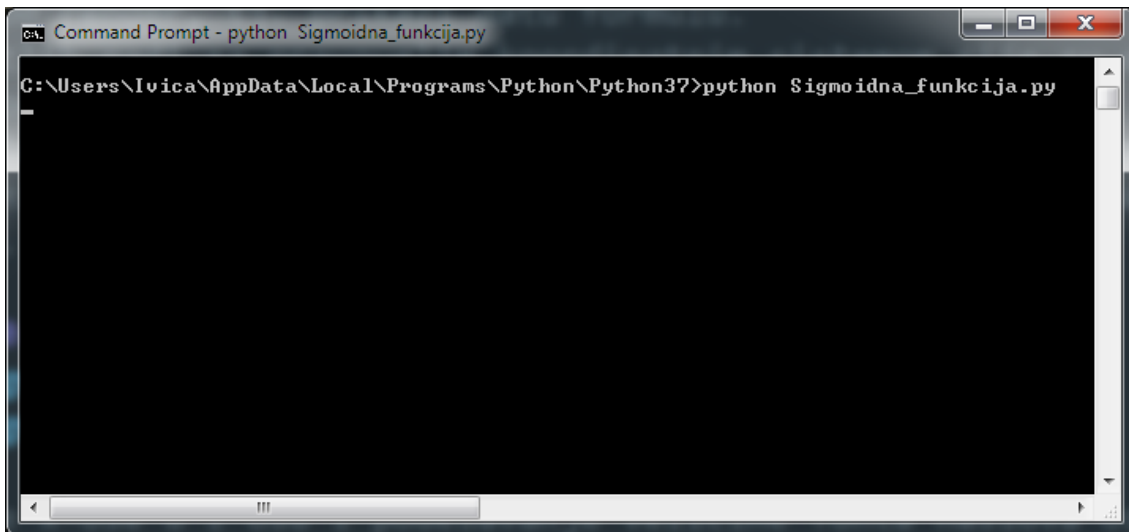
"""Kreiranje sigmoidne funkcije koriscenjem paketa pylab i numpy."""
# Iz paketa matplotlib se uvoze biblioteke paketa pylab
from matplotlib import pylab
import pylab as plt
import numpy as np
# Paketima pylab i numpy mogu biti dodeljeni bilo kakvi pseudonimi, ali je, ipak, najbolje
# koristiti skracenice plt i np , jer su te skracenice opste prihvacene.
# Definicija Sigmoidne funkcije je:  $\sigma(z) = 1/(1+e^{(-z)})$ , gde je  $z = \sum(x_j * w_j) - b$ 
# Funkciji sigmoidna_funkcija() je dodeljen argument z.
# Komandom return se kao rezultat vraca rezultat rada sigmoidne funkcije.
# Iz paketa numpy se poziva eksponencijalna funkcija kojoj se dodeljuje vrednost
argumenta
# z.
def sigmoidna_funkcija(z):
    return (1 / (1 + np.exp(-z)))
# Naziv grafika pozivanjem title() metode
plt.title('Sigmoidna funkcija')
# Iz paketa pylab se poziva funkcija linspace koja kreira ravnomerno rasporedjen
numericki
# niz duz x-ose. Prvi broj definise pocetnu vrednost niza, drugi broj definise krajnju
vrednost
# niza, dok treci broj definise za koliko tacaka ce se racunati funkcija. Sto je treci broj
veci
# to ce funkcija biti preciznije nacrtana.

```

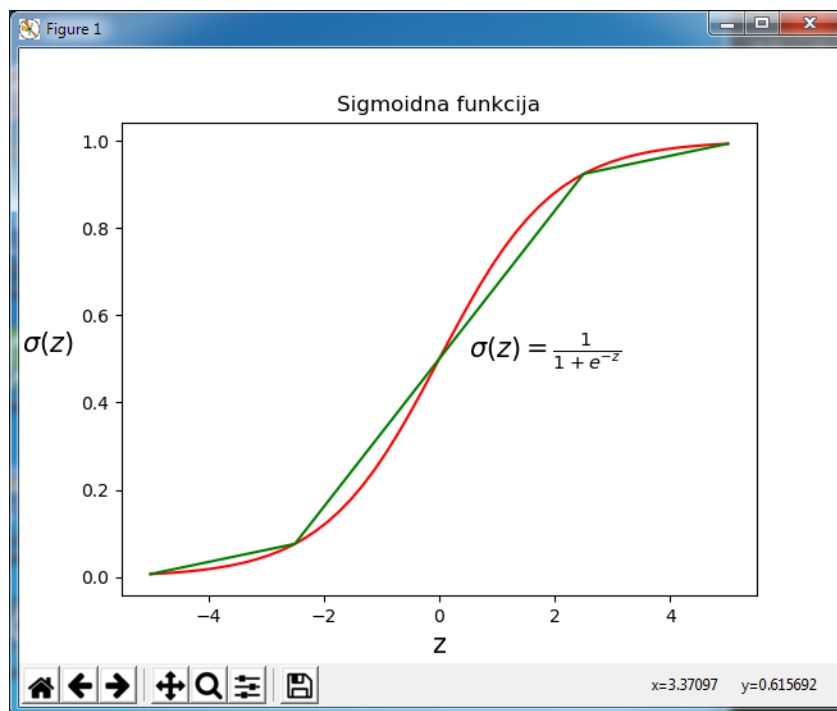
```

# Definisane su varijabile x i y da bi bio prikazan na koji nacin utice broj uzoraka koji ce
# biti
# obradjen (treći broj u zagradi). Da ne bi bilo zabune nije koriscen simbol z za
# varijabilu.
x = plt.linspace(-5,5,50)
y = plt.linspace(-5,5,5)
# Pozivanjem text() koja pripada paketu pylab bice ispisana formula sigmoidne formule
# na grafiku. Parametri koji su definisani ovom metodom jesu polozaj u odnosu na x-osu,
# polozaj u odnosu na y-osu, sama formula i atribut fontzize kojim je odredjena velicina
# karaktera u formuli. Formula ce se nalaziti kod broja 0.5 (x-osa) i broja 0.5 (y-osa).
# Atribut r j od render, tj. prikazi datu formulu.
# Paket matplotlib radi sa pravouglim koordinatnim sistemom cije su ose x - y.
# U kasnijem kodu ce biti promenjena imena ove dve ose
# Pozivanjem plot() metode koja pripada paketu pylab bice iscrtan grafik.
# Grafik ce ciniti povezane vrednosti varijabile x (z-osa) i izracunate funkcije  $\sigma(z)$  za
# varijabilu x. Atribut 'r' ce krivu obojiti u crveno.
plt.text(0.5, 0.5, r'$\sigma(z)=\frac{1}{1+e^{-z}}$', fontsize=15)
plt.plot(x, sigmoidna_funkcija(x), 'r')
plt.plot(y, sigmoidna_funkcija(y), 'g')
# Preimenovanje x-ose u z osu i podesavanje velicine fonta naslova ose
# Preimenovanje y-ose u  $\sigma(z)$  osu, podesavanje velicine fonta naslova ose, ispisi ose
# horizontalno i
# podesavanje naslova ose skroz uz desnu ivicu
plt.xlabel('z', fontsize=15)
plt.ylabel('$\sigma(z)$', fontsize=15, rotation='horizontal', horizontalalignment='right')
# Pozivanjem metode show() dolazi do prikazivanja grafa.
plt.show()

```



Slika 5.3.2. Pokretanje koda preko CMD shell-a Windows OS-a



Slika 5.3.3. Sigmoidna funkcija

Sa slika 5.3.3 se vidi da Sigmoid funkcija sa jedne strane teži jedan, a sa druge strane teži nuli. Oblik sigmoidne funkcije je sličan obliku Hevisajdove step funkcije čiji kod i slika sledi.

"""Kreiranje Hevisajdove funkcije pomocu numpy paketa"""

Hevisajdova funkcije ili jedinica step funkcija ima sledece vrednosti:

```

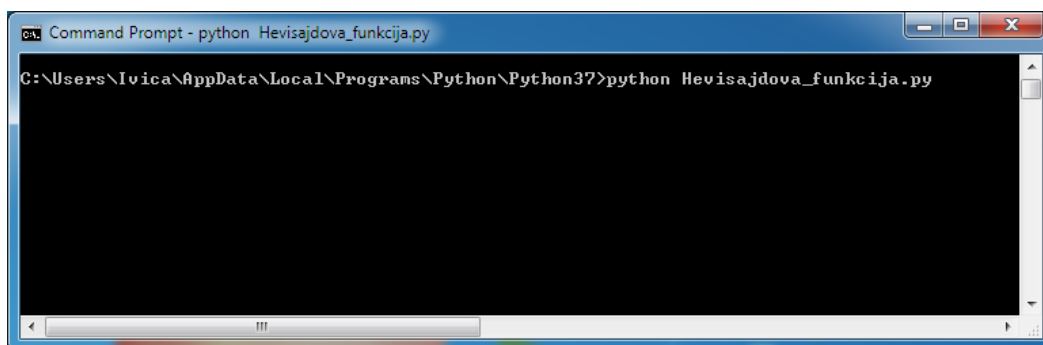
# 0 ako je  $x < 0$ 
# 1/2 ako je  $x = 0$ 
# 1 ako je  $x > 0$ 
# Kako je potrebno kreiranje multidimenzionalnog objekta bice koriscen numpy paket
# Graf ne moze biti nacrtan za sve uslove ukoliko se ne koristi multidimenzionalni objekat.
# Moguce je nacrtati graf ukoliko se posmatra uslov da je  $x > 0$ . Tada ce se vratiti samo
taj
# broj iz skupa.
# Bice koriscena kljucna rec return i kod ce biti: return 1 ( $x > 0$ )
# Graf ce biti tacan, ali pitanje je sta ce biti sa ostalim uslovima.
from matplotlib import pylab
import pylab as plt
import numpy as np
# Definise se nova funkcija hevisajdova_funkcija.
# Pomocu metode arrange() iz paketa numpy se kreira sekvenca ravnomerno
rasporedjenih
# vrednosti koje je se skladiste kao niz, tj. kao objekat.
# np.arange(x.shape[0]) ce metodi arange () dati prvi skalar u novokreiranom nizu koji
# obezbedjuje
# argument x. x.shape[0] je pridodat, jer je moguce imati vise vrednosti (radi se o
true/false).
# Time je definisano tacno jedno mesto u multidimenzionom objektu u koje ce biti upisana
# vrednost.
# Kreira se varijabila pozicija i ispituje njena vrednost kroz petlje.
# Ukoliko je u nizu hf na mestu varijabile pozicija stavljen argument x cija je vrednost 0,
# tada ce rezultat biti 0.5.
def hevisajdova_funkcija(x):
    hf = np.arange(x.shape[0])
    for pozicija in hf:
        if x[pozicija] == 0:
            hf[pozicija] = 0.5
        elif x[pozicija] > 0:

```

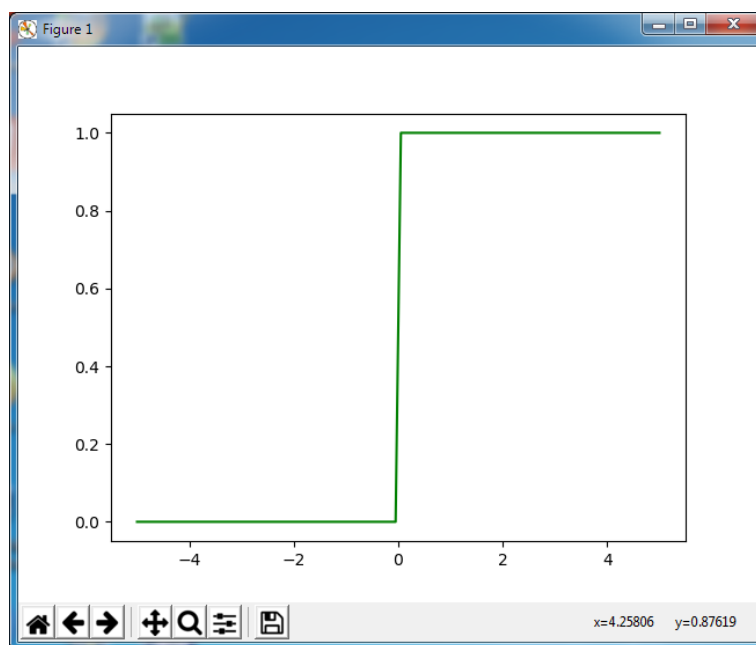
```

    hf[pozicija] = 1
elif x[pozicija] < 0:
    hf[pozicija] = 0
return hf
z = plt.linspace(-5,5,100)
plt.plot(z, hevisajdova_funkcija(z),'g')
plt.show()

```



Slika 5.3.4. Pokretanje koda preko CMD shell-a Windows OS-a



Slika 5.3.5. Hevisajdova step funkcija

Ukoliko bi sigmodna funkcija bila Hevisajdova stepena funkcija, tada bi sigmoid bio perceptron. Postoji razlika u izlazima sigmoidne funkcije i Hevisajdove stepene funkcije,

jer kada je $x * \omega + b = 0$, perceptron će kao izlaz dati vrednost jedan, dok bi kod Hevisajdove stepene funkcije za ulaz nula dobili izlaz jedan. To znači da bi trebalo modifikovati Hevisajdovu stepenu funkciju da kao izlaz da jedan.

Razlika koja se javlja između ove dve funkcije u tome što sigmoidna funkcija uzima i druge vrednosti za izlaz sem jedinice i nule, odnosno može se uvesti pojam fine-rafiniranosti sigmoidne funkcije. Pod pojmom fine-rafiniranosti sigmoidne funkcije se smatraju male promene u pogledu promene kako težinskih koeficijenata, tako i bajasa, pri čemu dolazi do fino-rafiniranih promena izlaza sigmoid neurona.

$$\Delta izlaza \approx \sum_j \frac{\partial izlaz}{\partial \omega_j} \Delta \omega_j + \frac{\partial izlaz}{\partial b} \Delta b \quad (5)$$

Iz formule (5) se vidi da dolazi do upotrebe parcijalnih izvoda tokom određivanja ukupnih promena funkcije izlaza. Ukupna promena izlaza, tj. Δb predstavlja linearnu zavisnost u odnosu na $\Delta \omega_j$ i Δb . Spomenuta linearnost omogućava lakši odabir malih promena u koeficijentima težine i bajasa. Na taj način je moguće ostvariti željenu malu promenu u izlazu. Korisnost sigmoidne funkcije se može sagledati u davanju izlaza koji će predstavljati npr. broj piksela (eng. pixel), odnosno gustinu piksela na određenoj slici.

5.4 Osobine neuronskih mreža

Kao što je u ovom radu već navedeno definicija neurona, po kojoj se biološki neuron može predstaviti kao element sa dva stanja, a koju su predložili Mekkalah i Pits [10] je odbačena. Predložena definicija predstavlja polaznu osnovu koju je trebalo prilagoditi i dopuniti. Prvi je to uradio američki neuropsiholog Donald Heb (Donald Hebb) 1949. godine u knjizi *The organization of Behavior*. On je proširio definiciju neurona i uspostavio postulat koji je nazvao *A Neurophysiological Postulate* koji glasi: „Kada je akson ćelije A dovoljno blizu da pobudi ćeliju B učestalim slanjem električnih impulsa, dolazi do odigravanja nekog procesa rasta ili metaboličkih promena u jednoj ili u obe ćelije, što dovodi do porasta efikasnosti uticaja ćelije A ka ćeliji B” [39]. Na osnovu ovog postulate Heb je zaključio da je veza između neurona „plastična”, tj. da je jačina ove veze u direktnoj korelaciji sa aktivnostima koje se javljaju između dva neurona [39]. Postulat je omogućio Hebu da uvede pojam koeficijenata težine sinapsi. Hebov postulat je važeći i danas, a jedna od matematičkih formulacija postulata je:

$$\Delta w_i = \epsilon * y(x) * x_i \quad (6)$$

Parametri koji se javljaju u jednačini (6) su:

Δw_i – promena koeficijenta težine i -te sinapse koju jedan neuron (presinaptički) šalje ka drugom neuronu (postsinaptički)

x_i – električni impuls koji neuron šalje drugom neuronu sinapsom, odnosno ulaz u i -tu sinapsu

$y(x)$ – funkcija pobude koju neuron prima od drugog neurona, tj. signal obučavanja

ϵ – parametar koji meri veličinu pojedinačnog koraka pri učenju, tj. koeficijent obučavanja

Osnovne osobine neuronskih mreža su [40]:

- mogućnosti linearnog i nelinearnog rešavanja problema
- jednom obučena neuronska feedforward mreža može da izvrši nelinearnu multivarijantnu transformaciju u konačnom i relativno malom broju koraka
- Režim može da izvrši multivarijantnu nelinearnu transformaciju u fiksnom (i generalno veoma malom) broju operacija
- kreiranje težinskih koeficijenata za različite ulazne signale
- da je neuron posmatranog sloja neuronske mreže pobuđen aktivnošću neurona prethodnog sloja te neuronske mreže
- da neuronska mreža ima sposobnost učenja na osnovu prethodnog iskustva, tj. prethodno obrađenih ulaznih signala
- neuronska mreža kreira relacije između ulaznih i izlaznih signala
- paralelnost obrade ulaznih signala
- decentralizacija smeštanja i čuvanja podataka u memoriju neuronske mreže.

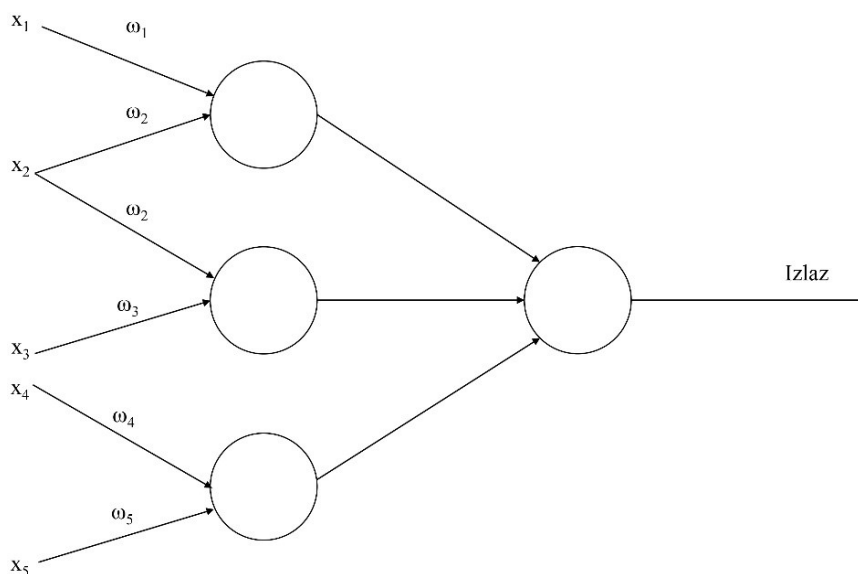
Decentralizacija smeštanja i čuvanja podataka u memoriju približava rad veštačke neuronske mreže radu biološke neuronske mreže ljudskog mozga. Delimično oštećenje veštačke neuronske mreže neće onemogućiti mrežu da obrađuje podatke. Ona će obrađivati podatke, ali će rezultati takve obrade biti manje precizni nego u slučaju potpune funkcionalnosti mreže (slično ponašanje ljudskom mozgu). Ova osobina veštačkih neuronskih mreža predstavlja znatan napredak u odnosu na klasične računarske sisteme.

5.5 Podela veštačkih neuronskih mreža

Postoji više podela neuronskih mreža. U ovom delu rada će biti date osnovne podele neuronskih mreža. U kasnijem delu rada će biti definisane metode učenja, tj. treniranja neuronskih mreža, pa će biti predstavljane i druge podele neuronskih mreža. Neuronske mreže se u osnovi mogu podeliti u zavisnosti:

- od broja slojeva:
 - jednoslojne veštačke neuronske mreže
 - višeslojne veštačke neuronske mreže
- smeru prostiranja podataka kroz neuronsku mrežu

Jednoslojne veštačke neuronske mreže se sastoje od jednog sloja neurona koji primaju stimulanse i obrađuju ih. Ove mreže su poznate pod imenom ADALINE (ADaptive LInear NEuron). Kod ovih mreža nema skrivenih slojeva veštačkih neurona, i postoji samo jedna vrednost izlaza. Sledi slika jednoslojne veštačke neuronske mreže.



Slika 5.5.1. Jednoslojna veštačka neuronska mreža

Najpoznatije strukture veštačkih jednoslojnih neuronskih mreža su mreže čije učenje zasniva na:

- perceptronskom pravilu učenja koje je definisao Rozenblat [14]
- mreže čije se učenje zasniva na Vidrov-Hofovom pravilu (Widrow-Hoof) [41]

U najjednostavnijem obliku matematička formulacija pravila učenja je [42]:

$$\Delta\omega_{ij} = \epsilon * e_i * a_j \quad (7)$$

Parametri koji se javljaju u jednačini (7) su:

$\Delta\omega_{ij}$ – promena koeficijenta težine j -te sinapse za i -ti neuron

ϵ – pozitivna vrednost koraka učenja

a_j – jačina j -te sinapse

e_i – razlika između željene i dobijene vrednosti izlaza, tj.

$$e_i = (y_{\text{željeno}} - y_{\text{dobijeno}}) \quad (8)$$

Problem koji se javlja kod pravila učenja koje je definisano formulom (8) u odnosu na formulu koja definiše Hebov postulat (formula (6)) je mogućnost pojave greške usled „podele radne snage” pri obradi ulaznih signala, odnosno pri određivanju koeficijenata težine veza između neurona. Sledi tabela 5.5.1. u kojoj su date vrednosti ulaza u dati u neuron i vrednost izlaza iz datog neurona.

Tabela 5.5.1. Ulazi u dati neuron i izlaz iz datog neurona

Ulaz	Izlaz
A	A
B	
B	
B	
B	
A	

Iz tabele 5.5.1. se vidi da samo ulaz čija je vrednost A može da da tačan izlaz čija je vrednost, takođe, A. Kada je ulaz u neuron impuls čija je vrednost B, kao izlaz se ne može dobiti impuls čija će vrednost biti B. Iz tog razloga ulaz čija je vrednost A treba da ima koeficijent težine koji je jednak jedinici, dok kod ulaza čija je vrednost B koeficijent težine treba da bude nula. Problem koji se može javiti je pogrešan odabir koeficijenata težine.

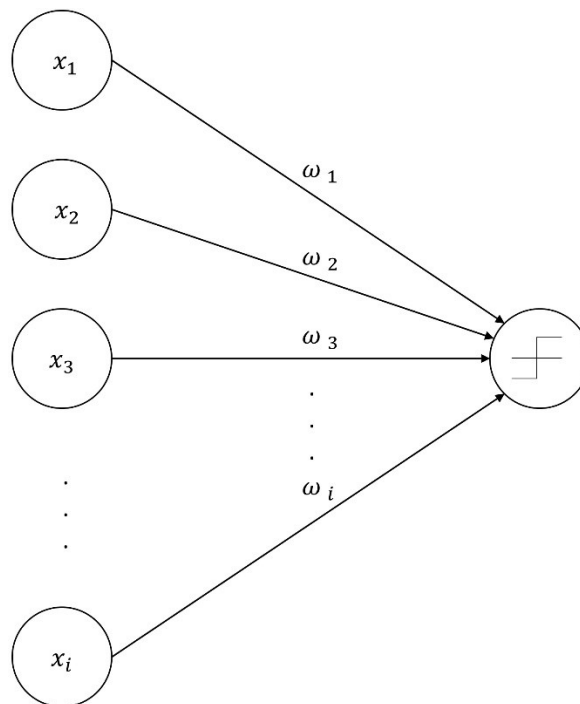
Za rešavanje problema kvalitetnog određivanja koeficijenata težine ulaznih veličina u neuron koristi se Vidrov-Hofovo pravilo koje je poznato kao i Delta pravilo učenja ili metoda najmanjeg kvadrata vrednosti greške – LMS (eng. Least Mean Square). Kod Delta

učenja se ne koristi aktivaciona funkcija praga (kao kod perceptronske metode učenja), već se za računanje aktivacione funkcije koristi linearna suma proizvoda koeficijenata težine i vrednosti njihovog ulaza [42]:

$$y_{dobijeno} = \sum_j(\omega_{ij} * x_{ip}) \quad (9)$$

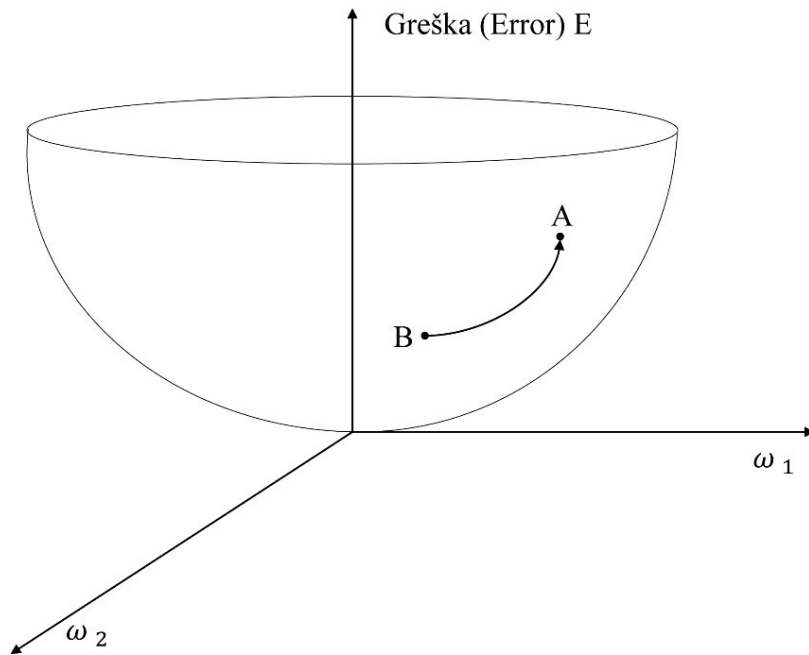
U formuli (9) indeks p stoji za *pattern (obrazac)*, odnosno p je obuka svakog od parova neurona koji učestvuju u razmeni signala određenih koeficijenata težine.

Ova funkcija se naziva Linearna Aktivaciona Funkcija.



Slika 5.5.2. Ulaz kod Delta učenja

Osnovna ideja LMS algoritma je iterativno ažuriranje (eng. update) sa svakim novim parom koji se trenira. Ovim treningom se smanjuje ukupna greška u procesu treniranja. Algoritam ovog pravila je algoritam nagledanog učenja koji ima kontinualnu funkciju aktivacije. Na slici 5.5.3. je prikazana funkcija greške (E) u zavisnosti od razlike između željene vrednosti (tačka A) i dobijene vrednosti (tačka B).



Slika 5.5.3. Funkcija greške u zavisnosti od razlike između željene i stvarne vrednosti izlaza

Ukoliko su željena i dobijena vrednost izlaza jednake, tada razlika između željene i dobijene vrednosti izlaza nula, tj. $\Delta\omega_i = 0$ i ne dolazi do promene koeficijenta težine. U suprotnom je $\Delta\omega_i \neq 0$, pa dolazi do prilagođavanja vrednosti koeficijenta težine u zavisnosti od vrednosti te razlike i nova vrednost koeficijenta težine je:

$$\omega(novo) = \omega(staro) + \Delta\omega \quad (10)$$

Iz ove jednačine $\Delta\omega$ je:

$$\Delta\omega = \omega(novo) - \omega(staro) \quad (11)$$

Za svaki par neurona koji je podvrgnut obuci, tj. treniranju, javlja se greška čija se ukupna vrednost za dati trenirani par neurona, može iskazati formulom:

$$E_p = \frac{1}{2} * \sum_n (y_{\text{željeno}} - y_{\text{dobijeno}})^2 \quad (12)$$

U formuli (12) $y_{\text{željeno}}$ predstavlja željenu vrednost izlaza (realni broj), dok je y_{dobijeno} dobijena vrednost izlaza, odnosno to je vrednost funkcije aktivacije. U razmatranom slučaju funkcija aktivacije će biti funkcija identita, tj. važiće $f(x) = x$.

Da bi se razlika između željene i dobijene vrednosti smanjila potrebno je izvršiti aproksimaciju greške učenja metodom smanjenja nagiba koeficijenata težine u odnosu na funkciju greške.

$$\frac{\partial E}{\partial \omega} = \frac{\partial (y_{\text{željeno}} - y_{\text{dobijeno}})^2}{\partial \omega} \quad (13)$$

$$\frac{\partial E}{\partial \omega} = -2(y_{\text{željeno}} - y_{\text{dobijeno}}) * \frac{\partial y_{\text{dobijeno}}}{\partial \omega} \quad (14)$$

Radi pojednostavljenja formule (14), linearnu aktivacionu funkciju možemo napisati u obliku:

$$y_{\text{dobijeno}} = \sum_j (\omega_{ij} * x_{ip}) = w * x \quad (15)$$

Tada će parcijalni izvod greške u učenju po koeficijentu težine biti:

$$\frac{\partial E}{\partial \omega} = -2(y_{\text{željeno}} - y_{\text{dobijeno}}) * \frac{\partial (w * x)}{\partial \omega} \quad (16)$$

$$\frac{\partial E}{\partial \omega} = -2(y_{\text{željeno}} - y_{\text{dobijeno}}) * x \quad (17)$$

Formula (17) se može zapisati i u sledećem obliku:

$$\frac{\partial E}{\partial \omega} = 2(y_{\text{dobijeno}} - y_{\text{željeno}}) * x \quad (18)$$

Razlika između željene i dobijene vrednosti $\Delta \omega$ koeficijenata težine će biti:

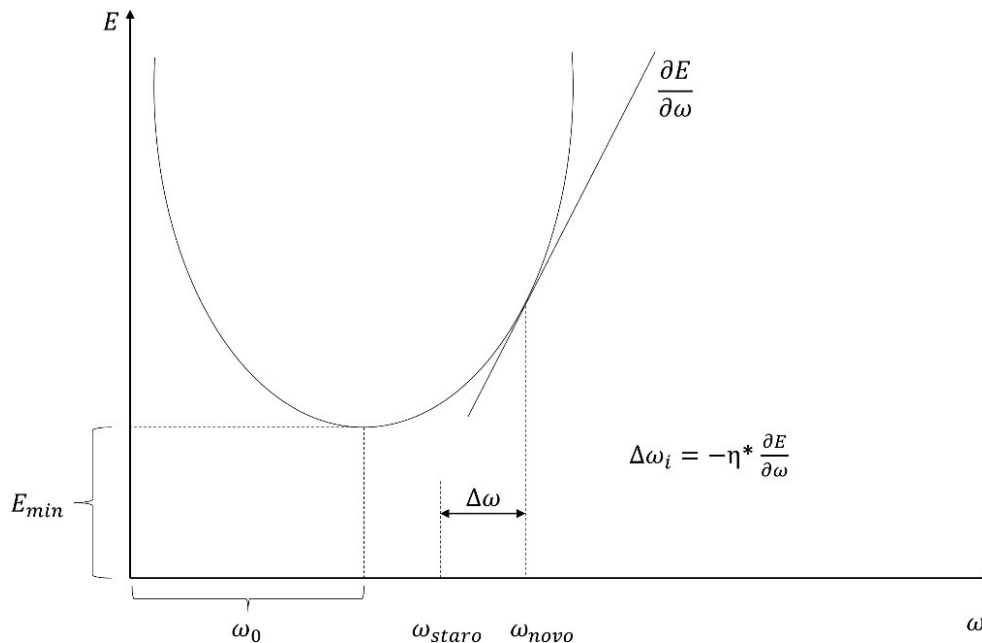
$$\Delta \omega_i = -\eta * \frac{\partial E}{\partial \omega} \quad (19)$$

Ukoliko koeficijente težine iskažemo kao vektore, formula (19) može biti zapisana u obliku:

$$\Delta \vec{w} = \eta \nabla E[\vec{w}] \quad (20)$$

U formulama (19) i (20) parametar η je korak treniranja.

Sledi slika 5.5.4. koja prikazuje postupak aproksimacije funkcije greške.

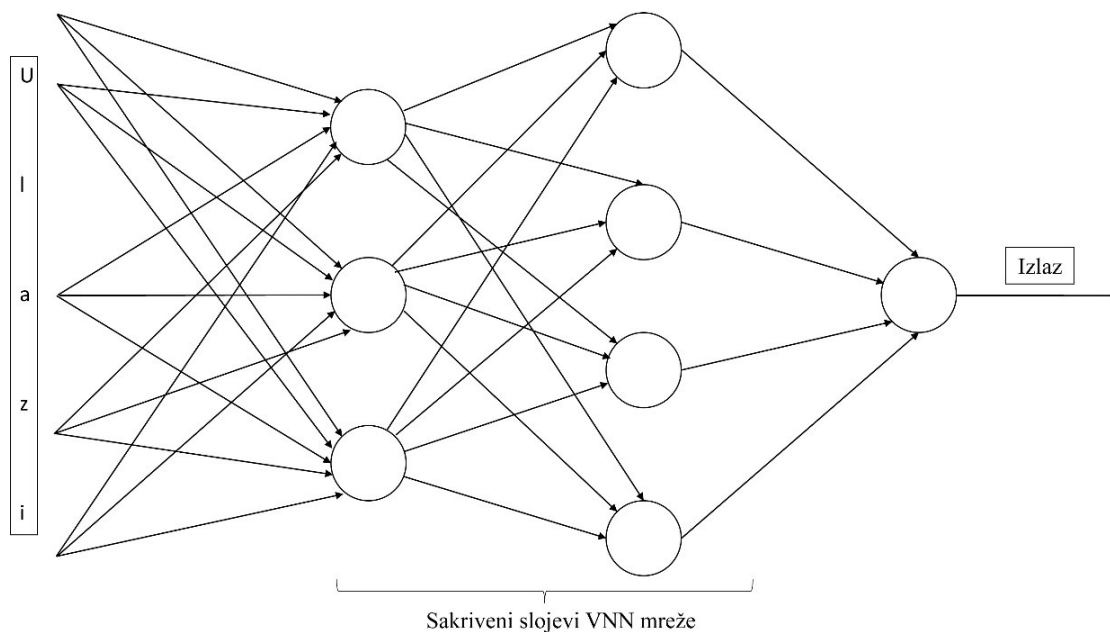


Slika 5.5.4. Delta učenje

Razlike između Delta učenja i perceptronskog učenja su:

- Vrednost greške kod Delta učenja može imati bilo koju vrednost. Kod perceptronskog učenja greška može imati vrednosti: 1, 0, -1.
- Kod Delta učenja može se naći izvod za bilo koju vrednost funkcije aktivnosti. Kod perceptronskog učenja funkcija aktivacije će biti diferencijabilna samo kada je vrednost te funkcije 0 ili 1. U svakom drugom slučaju funkcija nije diferencijabilna.

Kod višeslojnih veštačkih neuronskih mreža javlja se sloj neurona koji ne primaju direktno stimulanse iz okoline, već su njihovi ulazi u stvari izlazi prethodnog sloja neurona. Izlaz ovog sloja neurona predstavlja ulaz u sledeći sloj neurona. Odavde sledi da ovi ulazi nisu u direktnom kontaktu sa okolinom koja šalje originalne stimulanse prvom sloju neurona. Višeslojna neuronska mreža može imati jedan ili više slojeva što zavisi od niza faktora, a prvenstveno od stepena složenosti problema koji se rešava.



Slika 5.5.5. Višeslojna veštačka neuronska mreža

Određivanje broja skrivenih slojeva veštačke neuronske mreže je proces koji je kontinualan, tj. ne može se unapred precizno utvrditi broj skrivenih slojeva. Razlog tome leži u činjenici da je, najčešće, okruženje u kome se nalazi veštačka neuronska mreža dinamično. Dinamičnost okruženja dovodi do čestih promena stimulansa i broja stimulansa koji su ulaz u mrežu. Da bi ovaj problem bio prevaziđen razvijeni su različiti heuristički principi za kreiranje i određivanje broja skrivenih slojeva mreže.

Neuronske mreže se na osnovu smera prostiranja podataka u neuronskoj mreži mogu podeliti na:

- neuronske mreže sa prostiranjem podataka unapred – fidforward mreže (eng. Feedforward)
- neuronske mreže sa prostiranjem podataka unazad (eng. Backpropagation).

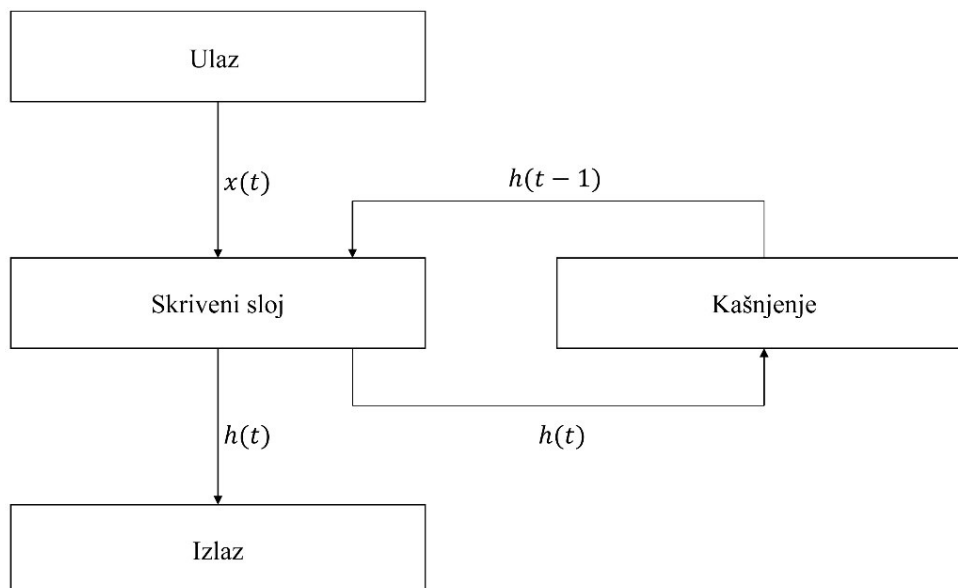
Kod Fidforward veštačke neuronske mreže neuroni samo prosleđuju ulazni signal koji su obradili ka sledećem sloju neurona, pri čemu nema povratne sprege. Izlaz neurona iz ove vrste mreže se izračunava na osnovu koeficijenata težine ulaznih signala.

Problem kod primene petlji kod ovih mreža ležu u paradoksu da bi ulaz u sigma funkciju jednog sloja neurona zavisio od izlaza tog istog sloja neurona. Ove mreže su poznate i pod imenom nerekurventne neuronkse mreže.

Neuronske mreže sa prostiranjem podataka unazad (eng. Backpropagation) ili rekurentne mreže, prilikom rešavanja složenog problema pokazuju svoji puni potencijal. U tom smislu je potrebno da neuronska mreža ima odgovarajuću arhitekturu koja će omogućiti obradu ulaznih signala, brzo i precizno izračunavanje funkcije aktivnosti i sopstveno učenje na osnovu prethodnog iskustva. Dodatni faktor koji treba uzeti u obzir kada neuronske mreže uče iz sopstvenog iskustva je činjenica da to učenje treba da bude sa prostiranjem unazad. Učenje sa prostiranjem unazad predstavlja korekciju stvarnih vrednosti izlaza iz neurona na osnovu unapred zadatih izlaznih vrednosti posmatranih neurona. Unapred zadate izlazne vrednosti neurona su definisane u algoritmu propagacije unazad (eng. Backpropagation Algorithm). Na osnovu korekcije izlaznih vrednosti neurona dolazi do smanjenja razlike između stvarnih i unapred zadatih vrednosti izlaza, odnosno neuronska mreža uči na osnovu te korekcije kako da izlazni rezultat bude približan unapred definisanom rezultatu, tj. neuronska mreža uči iz sopstvenog prethodnog iskustva [40].

5.5.1 Rekurentni modeli neuronskih mreža

Rekurentni modeli veštačkih neuronskih mreža su modeli neuronskih mreža u kojima je moguće postojanje povratne petlje. Ovaj model neuronskih mreža je osmislio i kreirao Džon Hopfield (John Hopfield) 1983. godine [43]. Kod rekurentnih neuronskih mreža dolazi do vraćanja izlaznog signala iz jednog sloja u prethodni sloj, odnosno dolazi do pamćenja prethodnog stanja (videti sliku 5.5.1.1.). Primena ovakvih mreža je manja, jer broj algoritma koji su za njih razvijeni nije veliki sa jedne, a sa druge strane potrebna je velika složenost takvih algoritama što zahteva i odgovarajuće alate za njihovo kreiranje. To sve poskupljuje proces stvaranja rekurentnih modela veštačkih neuronskih mreža. Princip rada ovakvih mreža više odgovara principu rada ljudskog mozga, nego što je to slučaj sa klasičnim feedforward neuronskim mrežama [44].



Slika 5.5.1.1. Princip rada rekurentne neuronske mreže

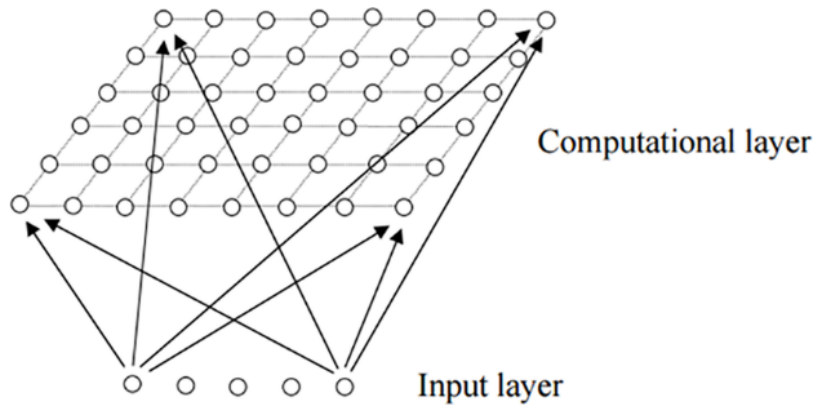
5.5.2 Podela neuronskih mreža u zavisnosti od metoda učenja

Pod učenjem neuronske mreže se podrazumeva treniranje neuronske mreže na promene koje se mogu javiti na ulazu. Postoje dve vrste učenja neuronske mreže:

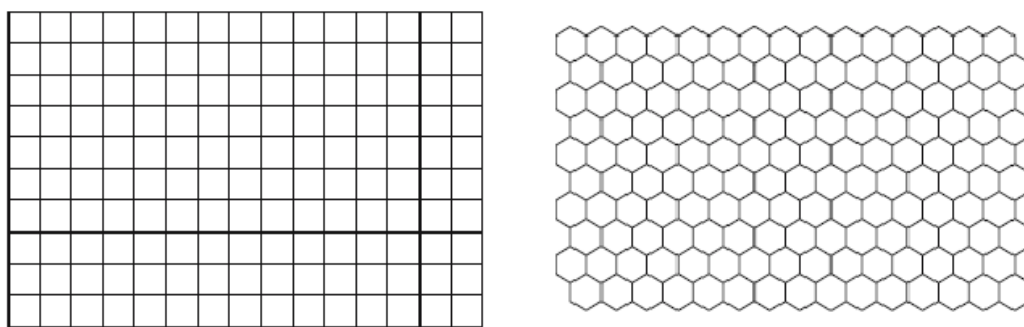
- nadgledano učenje
- nenadgledano učenje

Kod nadgledanog učenja se teži predviđanju izlaza na osnovu primera koji su se koristili u toku treniranja mreže [45]. Ovaj tip učenja se najčešće koristi kod nerekurentnih foward mreža.

Neuronska mreža kod nenadgledanog učenja vidi samo ulazne podatke dok izlazne podatke ne vidi. Neuronska mreža mora da odabere osobine ulaznih signala na osnovu kojih će izvršiti organizovanje tih ulaznih podataka. Na osnovu navedene karakteristike ove neuronske mreže se nazivaju samoorganizujuće mape ili SOM (eng. Self-Organizing Maps), tj. Kohonen mape [46]. Samoorganizujuća mapa se zasniva na tehnici grupisanja neurona u dvodimenzionalnu mrežu oblika pravouganika ili heksagona. Neuroni se kod ovih mreža nazivaju čvorovi ili referentni vektori.



Slika 5.5.2.1. Kohonenova ili samoorganizujuća mapa (izvor: [47])



Slika 5.5.2.2. Izgled Kohonenove dvodimenzionalne mreže ([88])

Kod Kohonen mapa može doći do:

- saradnje između neurona
- do njihovog međusobnog „takmičenja”
- ili do oba.

Do grupisanja neurona dolazi kroz višestruke iteracije. Tada će neuroni u mreži početi postepeno spajanje oko područja sa velikom gustinom podataka. Kretanjem neurona ka delovima mreže sa većom gustinom podataka dolazi do savijanja i uvijanja mreže, odnosno dolazi do prilagođavanja neurona vrednostima unešenih podataka. Slede koraci kroz koje prolaze neuroni u toku procesa nenadgledanog učenja kod Kohonenove neuronske mreže:

- Nulti korak: Nasumično pozicioniranje neurona mreže u prostoru podataka.
- Prvi korak: Bira se jedna tačka podataka. Izbor te tačke može biti ili nasumičan ili sistematičan (kruženjem kroz skup podataka).

- Drugi korak: Pronalazi se neuron koji je najbliži izabranoj tački podataka. Ovaj neuron naziva se jedinica najboljeg podudaranja (eng. Best Matching Unit).
- Treći korak: Neuron se dovodi u najbliži, skoro podudarajući položaj sa jedinicom najboljeg podudaranja. Rastojanje između položaja neurona i jedinice najboljeg podudaranja se u svakoj iteraciji smanjuje.
- Četvrti korak: Dolazi do kretanje ostalih neurona koji se pomeraju ka jedinici najboljeg podudaranja. Broj neurona koji se pomeraju ka jedinici najboljeg podudaranja je u obrnutoj proporciji sa rastojanjem neurona od jedinice najboljeg podudaranja (veća udaljenost sledi da je manji broj neurona koji će se pokrenuti).
- Peti korak: Kada su svi neuroni koji su mogli došli u blizinu jedinice najboljeg podudaranja, potrebno je ažurirati rastojanje, tj. potrebno je naći novi neuron koji u predašnjoj iteraciji nije pomevan. Nakon toga sledi ponavljanje koraka od prvog do četvrtog koraka.

Ukoliko brzina pomeranja neurona i rastojanje između neurona i jedinice najboljeg podudaranja imaju previsoke vrednosti, neuroni će se stalno kretati, tj. biće teško odrediti koji je neuron najbliži jedinici najboljeg podudaranja. Sa druge strane, ako su te vrednosti niske, može se desiti da proces pomeranja neurona ne bude nikada priveden kraju. Pored toga potrebno je pažljivo odrediti koliko će neurona činiti mrežu.

Nedostaci Kohonenove neuronske mreže su:

- Radi samo sa diskretnim varijabilama.
- Skup podataka sa više varijabila zahteva više vremena za izračunavanje udaljenosti i identifikaciju jedinice najboljeg podudaranja.
- Ponekad se prevelike grupe podataka mogu podeliti i predstaviti sa dve odvojene grupe neurona, pa je potrebna naknadna provera konzistentnosti podataka.

5.5.3 Modularne neuronske mreže

Do razvoja modularnih neuronskih mreža dolazi sredinom osamdesetih godina XX veka. Motiv za razvoj modularnih mreža leži u istraživanjima koja su vršena u oblastima psihologije i biologije. Ova istraživanja su dala detaljniji uvid u princip rada ljudskog

mozga. Nakon ovih istraživanja ljudski mozak se više ne tretira kao monolitna neuronska mreža, već kao mreža koja se sastoji više međusobno povezanih mreža [48]. Modularnost omogućava ovim mrežama da koriste prethodno znanje na mnogo efikasniji način što dovodi do lakše integracije i ažuriranje znanja prilikom učenja mreže. Zahvaljujući slojevitoj strukturi modularne mreže moguće je složen problem rastaviti na manje celine, a te manje celine će rešavati odgovarajući delovi modularne mreže. Ovo omogućava modularnim mrežama da budu proširive, odnosno one nisu sputane složenošću jedne monolitne celine [49]. Pored toga ove mreže mogu dati adekvatan odgovor na promene ulaznih signala u toku vremena.

Kod kreiranja modularnih neuronskih mreža koriste se dva principa:

- podeli i vladaj (eng. divide and-conquer)
- promovisanje različitosti (eng diversity-promotion)

Princip podeli i vladaj primenjuje metodologiju koja omogućava rešavanje složenog problema podelom tog problema u manje probleme koji su relativno jednostavni za rešavanje. Pojedinačna rešenja tih problema se kombinuju u konačno, jedinstveno rešenje.

Sa druge strane princip promovisanja različitosti se zasniva na postojanju različitih načina na koji je moguće pristupiti obradi datog problema. Pristupi u rešavanju datog problema dovode do postojanja dva modela modularnih mreža i to:

- modeli sa jakim vezom između komponenti mreže
- modeli sa slabom vezom između komponenti mreže

Kod modela sa jakim vezom između komponenti mreže sve komponente mreže su zajedno trenirane i međusobno povezane. Uzima se interakcija između komponenti u toku jedne faze učenja. Na osnovu te interakcije se ažuriraju preostali parametri mreže.

Kod modela sa slabom vezom između komponenti mreže postoji različita strategija u rešavanju problema u različitim delovima mreže. Te strategije mogu biti, ali i ne moraju, međusobno povezane određenim relacijama.

5.5.4 Neuronske mreže sa osnovnim radijalnim funkcijama

Osnovna ideja linearne odvojivosti je da se proveri da li postoji mogućnost podele tačaka u n -dimenzionalnom prostoru korišćenjem $n - 1$ dimenzija. Kod odabira dva broja postoje dve mogućnosti:

- Odabrati dva ista broja
- Odabrati dva različita broja

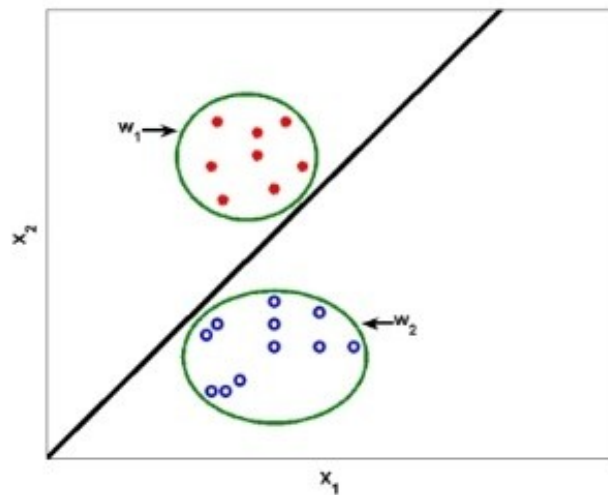
U drugom slučaju će postojati broj koji može da se nađe između dva odabrana broja. Taj broj odvoja izabrana dva broja, pa se kaže da su ta dva broja linearno odvojiva (videti sliku 5.5.4.1).



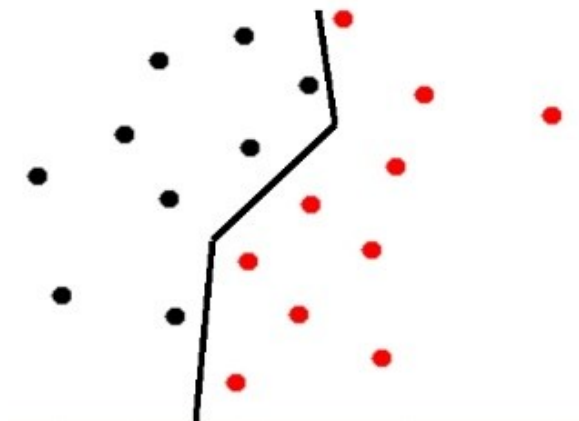
Slika 5.5.4.1. Linerna skalabilnost

U slučaju odabira brojeva, može se desiti da budu odabrana dva ista broje, što će predstavljati problem. Tada nije moguće izvršiti linearnu odvojivost. Ovaj problem se može javiti kod neuronskih mreža i to u vidu odabira istog koeficijenta težine. Do sada su razmatrane linearne neuronske mreže gde je bilo moguće uočiti razliki između ulaznih veličina, pri čemu su te ulazne veličine određene koeficijentima težine.

U slučaju dvodimenzionalnog prostora moguće je, takođe, uočiti linearnu i nelinearnu odvojivost, što je prikazano na slikama koje slede. U slučaju slike 5.5.4.2. moguće je izvršiti linearnu podelu tačaka po određenom obrascu, tj. moguće je napraviti dve klase podataka na osnovu veličine koeficijenata težine tih ulaznih podataka. Na slici 5.5.4.3. nije moguće izvršiti linearnu odvojivost po određenom obrascu i kreirati dve klase podataka.



Slika 5.5.4.2. Linearna skalabilnost dvodimenzionalnog prostora ([89])



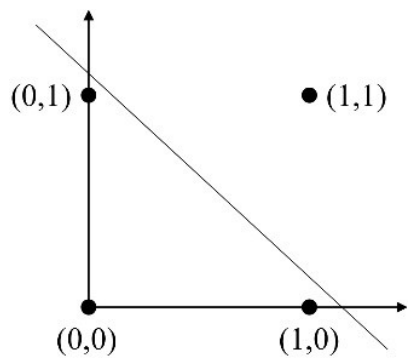
Slika 5.5.4.3. Nelinearna skalabilnost dvodimenzionalnog prostora

Primer linearne podele ulaznih podataka su: I (AND), ILI (OR) i NI (NAND) logička kola. Slede tabele i slike linearne odvojivosti za I (AND), ILI (OR) i NI (NAND) logička kola.

Tabela 5.5.4.1. Tablica istinitosti za I (AND) logičko kolo

p	q	AND
⊤	⊤	⊤
⊤	⊥	⊥
⊥	⊤	⊥
⊥	⊥	⊥

p	q	AND
1	1	1
1	0	0
0	1	0
0	0	0

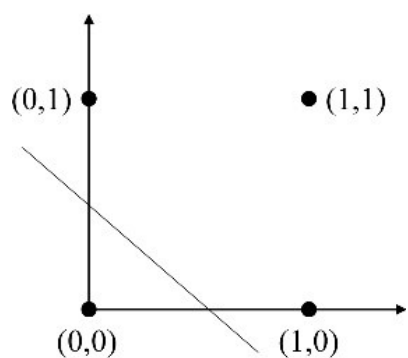


Slika 5.5.4.4. Raspored ulaznih parova I (AND) logičkog kola

Tabela 5.5.4.2. Tablica istinitosti za ILI (OR) logičko kolo

p	q	OR
T	T	T
T	⊥	T
⊥	T	T
⊥	⊥	⊥

p	q	OR
1	1	1
1	0	1
0	1	1
0	0	0

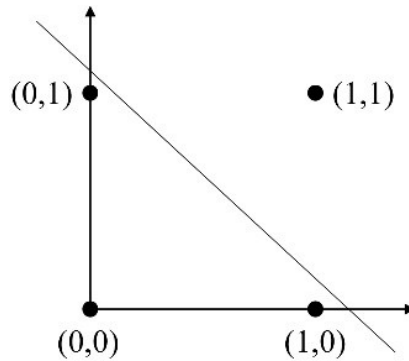


Slika 5.5.4.5. Raspored ulaznih parova ILI (OR) logičkog kola

Tabela 5.5.3.3. Tablica istinitosti za NI (NAND) logičko kolo

p	q	$NAND$
T	T	⊥
T	⊥	T
⊥	T	T
⊥	⊥	T

p	q	$NAND$
1	1	0
1	0	1
0	1	1
0	0	1



Slika 5.5.4.6. Raspored ulaznih parova NI (NAND) logičkog kola

Sledi kod (sa komentarima) napisan u Pajtonu za neuronsku mrežu koja oponaša rad I (AND) logičkog kola:

"""Program koji oponasa rad AND (I) logickog kola, odnosno program ce uraditi linearnu odvojivost. Bice kreirana dva skupa podataka koji ce biti predstavljeni u dvodimenzionalnom prostoru.

Uvozi se numpy paket (Numerical Python) kao osnovni paket koji sadrzi alate i tehnike koje omogucavaju primenu racunara u naucne svrhe, odnosno primenu racunara za resavanje slozenih problema i kreiranje objekata multidimenzionalnih nizova (arrays).

NumPy array je centralna struktura podataka u NumPy paketu.

Obican niz predstavlja mrežu (redovi i kolone) argumenata koji imaju vrednosti istog tipa. Ukoliko se posmatra sama struktura niza vidi se da niz nije nista drugo do pokazivac (bolje je koristiti eng. rec pointer).

Niz se sastoji od:

- *pointera podataka ili data pointer koji ukazuje na memorijsku adresu prvog bajta u nizu*
- *pointera tipa podataka ili data type pointer koji opisuje vrstu/tip elementa koji se nalazi u okviru niza*
- *oblik ili shape koji ukazuje na oblik niza*
- *napred ka ili dugackog koraka ili strides koji predstavlja broj bajtova koji treba biti preskocen u memoriji da bi se pristupilo sledecem elementu niza.*

Multidimenzioni nizovi su nizovi kod kojih svaka celija niza predstavlja jedan niz."""

Komanda import uvozi numpy paket pod (as) pseudonimom/skracenicim np.

import numpy as np


```

#Kreira se klasa JednoslojnaNeuronskaMreza
class JednoslojnaNeuronskaMreza:
# Klasa je apstraktan pojam koji postoji u logickom svetu, dok u fizickom svetu ne postoji.
# Zato je potrebno kreirati objekat date klase koji ce predstavljati instancu, tj. biti fizicki
# reprezent klase.
# Da bi klasa bila uspesno kreirana potrebno je koristiti ugradjenu funkciju __init__()
# Funkcija koja je definisana u okviru klase se naziva metoda.
# Metoda __init__() ce inicijalizovati objekat date klase, tj. ova funkcija je konstruktor u
# Pajtonu i kreirace objekat u memoriji.
# Preostali clanovi u zagradi __init__() metode su argumenti.
# init() kreira objekat, a parametar self omogucava perzistentnost argumenata u objektu.
# Ako se parametar self ne koristi onda ce vrednosti argumenata koje su definisane init()
# metodom
# biti odbacene kada se izadje iz opsega te metode.
# Koriscenjem self parametra varijabile postaju clanovi kreiranog objekta.
# Parametri koji ce postojati u okviru kreiranog objekta su: ulazna_vrednost i koef_tezine
# Parametar ulazna_vrednost moze biti jedna od kombinacija logickih iskaza (vise u
# komentaru koji ce biti dat kasnije u kodu).
# Parametar koef_tezine je parametar cija je predifinisana vrednost None
def __init__(self, ulazna_vrednost, koef_tezine=None):
# Metoda np.ones() upisuje sve jedinice u argument ulazna_vrednost.
# Kreira se if-else petlja
# Da bi I kolo dalo tacan iskaz potrebno je da na ulazu budu dve jedinice, odnosno dve
# tacne
# tvrdnje.
# Kada se funkcija pozove svaki put dolazi do kreiranja objekta.
# Da bi bilo naglaseno da objekat nema argumente koristi se parametar None koji u
# Pajtonu
# predstavlja ekvivalent nuli, tj. argument ima vrednost null
# Ukoliko je koef. tezine = null, onda ce u polje koje je nazvano koeficijent biti upisana
# vrednost argumenta
# ulazna_vrednost * 0.5.

```

```

# Koristi se modul np.ones koji ce u koloni ulazna_vrednost postaviti sve 1.
# Modul np.ones ima odredjene pointere (vec navedeno sta su pointeri). Jedan od njih je
# shape koji definise oblik #
# ulazne matrice. U ovom slucaju je povucena prva kolona matrice
# koja je definisana sa init() metodom, tj. povucena je kolona ulazna_vrednost
# U suprotnom ukoliko argument koef_tezine ima neku vrednost tada ce u polje koeficijent
# biti upisana vrednost samog argumenta.
# Mnozi se sa 0.5, jer verovatnoca da izlaz ima vrednost 0 ili 1 iznosi 0.5
if koef_tezine is None:
self.koeficijent = np.ones(ulazna_vrednost) * 0.5
else:
self.koeficijent = koef_tezine
# @staticmethod predstavlja dekorater u Pajtonu. Dekorater uzima funkciju, dodaje joj
# neke
# funkcionalnosti i vraća je.
# Ovim dekoraterom je funkciji staticmethod pridružena nova funkcija, tj. osobina.
# Ugradjena staticmethod funkcija vraca staticku metodu za datu funkciju.
# Ova metoda ne moze da menja stanje objekta niti stanje klase, odnosno ne moze da
# prima
# cls parametar (parameter class metode) niti self parametar (ne menja stanje kreiranog
# objekta). Kao prvi argument se prosledjuje x koji je dodeljen funkciji
# hevisajdova_funkcija.
# Kreira se if uslov koji kada x ima vrednost vecu od 0.5 vraca 1, a u suprotnom funkcija
# hevisajdova_funkcija vraca 0.
@staticmethod
def hevisajdova_funkcija(x):
if x > 0.5:
return 1
return 0
# Funkcija call poziva objekat/instancu kao funkciju
# Kreirano je novo polje koef_tezine_ulaza cija ce vrednost biti proizvod vrednosti polja
# koeficijent i parametra ulaz

```

```

# koji je definisan metodom call()
# Novo polje suma_koef_tezine_ulaza predstavlja sumu polja koef_tezine_ulaza
# return ubacuje suma_koef_tezine_ulaza kao parametar funkcije hevisajdova_funkcija.
# Ovaj rezultat se proseldjuje klasi JednoslojnaNeuronskaMreza
def __call__(self, ulaz):
    koef_tezine_ulaza = self.koeficijent * ulaz
    suma_koef_tezine_ulaza = koef_tezine_ulaza.sum()
    return JednoslojnaNeuronskaMreza.hevisajdova_funkcija(suma_koef_tezine_ulaza)
# Definise se novi parametar p tako sto mu se dodeljuje vrednost klase
# JednoslojnaNeuronskaMreza pri cemu se kreira
# multidimenzionalni objekat, tj. niz kome matrica koja ce imati dve kolone koje imaju
# vrednost 0.5 i 0.5
# Kreira se for petlja koja za ulazne vrednosti argumenta x kome se dodeljuju vrednosti
0 i
# 1
# Kreira se varijabila y koja ce predstavljati niz koji predstavlja resenje logicke operacije
I
# Stampa se matrica (x,y)
p = JednoslojnaNeuronskaMreza(2, np.array([0.5, 0.5]))
for x in [np.array([0, 0]), np.array([0, 1]),
np.array([1, 0]), np.array([1, 1])]:
    y = p(np.array(x))
    print(x, y)

```

```

C:\Users\Ivica\AppData\Local\Programs\Python\Python37>python JednoslojnaNeuronskaMreza.py
[[0 0] 0
 [0 1] 0
 [1 0] 0
 [1 1] 1
C:\Users\Ivica\AppData\Local\Programs\Python\Python37>

```

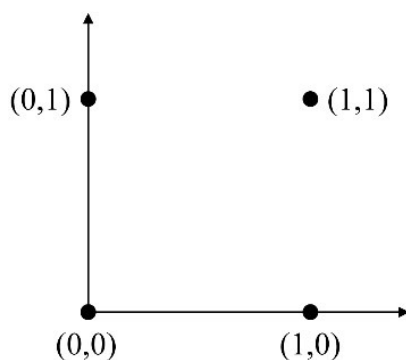
Slika 5.5.4.7. Rezultat rada koda pokrenut preko CMD shell-a Windows OS-a
 Sa gornje se vidi da je jedino tačno rešenje u slučaju kada oba ulaza imaju vrednost 1.

Primer nelinearne podele ulaznih podatka je Ekskluzivno ILI kolo, tj. XILI (XOR) logičko kolo. Sledi tablica istinitosti za XILI logičko kolo.

Tabela 5.5.4.4. Tablica istinitosti za XILI (XOR) logičko kolo

p	q	XOR
\top	\top	\perp
\top	\perp	\top
\perp	\top	\top
\perp	\perp	\perp

p	q	XOR
1	1	0
1	0	1
0	1	1
0	0	0



Slika 5.5.4.8. Raspored ulaznih parova XILI (XOR) kola bez mogućnosti linearne podele

Kao što se vidi sa gornje slike kod XILI (XOR) kola nije moguće napraviti linearnu podelu ulaznih parova na dve klase:

- klasu koja će predstavljati skup ulaznih veličina (uređeni parovi: (1,0) i (0,1)) čiji će izlaz biti 1,
- klasu koja će predstavljati skup ulaznih veličina (uređeni parovi: (1,1) i (0,0)) čiji će izlaz biti 0.

Dati problem podele ulaznih podataka se može rešiti korišćenjem neuronskih mreža sa osnovnim radijalnim funkcijama (eng. Radial Basis Function Neural Networks). Kod primene radijalnih funkcija se teži grupisanju ulaznih podataka u radijalni oblik, tj. kružni oblik sa laganim prelazima između datog skupa i okoline koja ga okružuje. Udaljenost podatka od periferije ukazuje na to da li je podatak bliži centru radijalnog oblika ili periferiji tog istog radijalnog oblika. Neuronske mreže sa radijalnom funkcijom su dvoslojne fidforvard mreže kod kojih skriveni slojevi implementiraju skup radijalnih osnovnih funkcija. Primena radijalnih funkcija u skrivenom sloju neuronske mreže se

zasniva na Koverovoj (Cover, 1965) teoremi linearne odvojivosti ulaznih podataka koja glasi: „Ulazni podaci koji se nalaze u višedimenzionalnom prostoru, imaju veću verovatnoću da budu linearno odvojivi, nego da se ti isti podaci nalaze u nižedimenzionalnom prostoru.“ [50]. Ako se skup ulaznih podataka predstavi kao skup vektora u d -dimenzionalnom Euklidovom prostoru, postojaće homogena linearna funkcija praga (eng. threshold) koja će podeliti Euklidov prostor u dva skupa: X^+ i X^- [49], odnosno homogena linearna funkcija praga deli Euklidov prostor [50]:

$$f_{\omega}(x) = \begin{cases} 1, & \omega * x > 0 \\ 0, & \omega * x = 0 \\ -1, & \omega * x < 0 \end{cases} \quad (21)$$

Iz formule (21) se vidi da Euklidov prostor, u oznaci E^d [50]:

- predstavlja proizvod vektora koeficijenta težine za dati ulaz i vektora ulaznih signala
- podeljen na dva skupa:
 - skup vektora kod kojih je $f_{\omega}(x) = 1$
 - skup vektora kod kojih je $f_{\omega}(x) = -1$

Ova dva skupa razdvaja hiperravan (kada je $f_{\omega}(x) = 0$). Hiperravan predstavlja podprostor Euklidovog prostora koji ima d dimenzija. Dobijena hiperravan ima $(d - 1)$ dimenzija i predstavlja ortogonalnu ravan u odnosu na vektor koji predstavlja koeficijent težine [50].

Da bi se, po Koveru [50], postigla potpuna binarna razdvojenost ili dihotomija (starogrčki: dikhotomia) potrebno je da postoji skalar t , tako da važi:

$$\text{za } \forall x \in X^+ \text{ važi } \omega * x > t \quad (22)$$

$$\text{za } \forall x \in X^- \text{ važi } \omega * x < t \quad (23)$$

Na osnovu ove teoreme je kod mreža sa radijalnom funkcijom potrebno izvršiti preciznu interpolaciju skupa ulaznih podataka (sadrži N podataka u sebi) u podatke koji će se nalaziti u prostoru koji ima više dimenzija nego početni prostor iz koga potiču originalni ulazni podaci.

Neka je X skup ulaznih veličina u neuronsku mrežu i neka sadrži do x_N ulaznih veličina, tj. neka važi:

$$X = \{x_1, x_2, \dots, x_N\} \quad (24)$$

Ova dva skupa će biti podeljena familijom površina, pri čemu svaka od površina deli ulazni prostor na dva regiona podataka. Podaci su grupisani u regione na osnovu karakteristika koje poseduju. Svaki od regiona obuhvata podatke čije su karakteristike skroz suprotne drugom regionu. Za svaki ulazni podatak za koji važi da za $\forall x \in X$ se definiše novi vektor koji predstavlja radijalanu funkciju za dati ulazni parametar, tj. definiše se vektor $\varphi_i(x)$, pri čemu $i = \{1, 2, \dots, M\}$. Odavde sledi da novi prostor ima M dimenzija. Novi skup vektora može biti predstavljen sledećom formulom:

$$\varphi(x) = [\varphi_1(x), \varphi_2(x), \dots, \varphi_M(x)]^T \quad (25)$$

Funkcija $\varphi(x)$ se naziva skrivena funkcija. Nakon uvođenja funkcije $\varphi_i(x)$ mogu se definisati nove zavisnosti:

$$\omega_T \varphi(x) > 0 \text{ za } \forall x \in X^+ \quad (26)$$

$$\omega_T \varphi(x) < 0 \text{ za } \forall x \in X^- \quad (27)$$

Koverova formula [49] po kojoj je slučajna dihotomija φ -odvojiva glasi:

$$P(N, M) = \left(\frac{1}{2}\right)^{N-1} * \sum_{m=0}^{M-1} \binom{N-1}{m} \quad (28)$$

Neke od funkcija koje se mogu koristiti kao osnovne funkcije su: Gausova funkcija, multikvadratna funkcija, ...

Trening ove vrste neuronske mreže se izvodi u dva koraka:

- u prvom koraku se određuju koeficijenti težine od ulaza do skrivenog sloja
- u drugom koraku se određuju koeficijenti težine od skrivenog sloja ka izlazu.

Trening i učenje ovakvih mreža je znatno brže u odnosu na obične višeslojne neuronske mreže, ali sa druge strane preciznost koja ova mreža ima prilikom obrade podataka i davanja izlaza je znatno niža u odnosu na obične višeslojne neuronske mreže.

5.5.5 Primena evolutivnih algoritama u neuronskim mrežama

Evolutivni algoritmi se koriste za simulaciju okruženja i ponašanja entiteta u populaciji. Da bi evolutivni algoritmi bili primenjeni potrebno je kreirati okruženje u

kome je moguća evolucija potencijalnog rešenja. Dva osnovna tipa evolutivnih algoritama su:

- genetski algoritmi [38]
- evolutivne strategije [51],[52]

Sličnost između veštačkih neuronskih mreža i evolutivnih algoritama leži u činjenici da se oba pristupa veštačkoj inteligenciji zasnivaju na principima koji su proistekli iz bioloških istraživanja.

5.5.5.1 Genetski algoritmi

Genetski algoritmi su algoritmi pretrage koji se zasnivaju na:

- prirodnoj selekciji koja se javlja u biološkom svetu
- genetici biološkog sveta.

Genetski algoritmi su heuristički modeli učenja zasnovani na principima prirodne evolucije i selektivnog razvoja [53].

Kod genetskih algoritama je potrebno unapred definisati skup mogućih rešenja. Karakteristike genetskih algoritama omogućavaju im da izvrše brzo poboljšanje prosečnih sposobnosti populacije i da brzo prepoznaju regione visokih performansi u populaciji u okruženju koje je složeno za pretragu. Genetski algoritmi sadrže skup probnih rešenja za dati problem. Skup probnih rešenja se može nazvati populacijom. Početna populacija nastaje slučajnim izborom iz skupa mogućih rešenja, odnosno domena mogućih rešenja. Izbor domena mogućih rešenja predstavlja najteži deo kreiranja i primene genetskih algoritama, jer se neadekvatnim izborom domena mogućih rešenja može ugroziti efikasnost samog genetskog algoritma. Pojedinačno rešenje pripada određenoj populaciji na osnovu niza bitova koji ga čine (u biološkom smislu taj niz bitova je niz hromozoma koji čine datu DNK). Da bi populacija evoluirala postojeća rešenja se moraju stalno preispitivati, odnosno potrebno je stalno prolaziti kroz sistem i primenjivati rešenje koje će dati optimalne rezultate. Rešenja koja su prevaziđena se eliminišu iz domena mogućih rešenja. Na taj način nova rešenja zamenjuju stara rešenja, odnosno dolazi do procesa evolucije u populaciji i podizanja kvaliteta iste. Funkcija genetskog algoritma koja vrši selekciju i proces eliminacije prevaziđenih pojedinačnih rešenja iz

skupa, odnosno domena mogućih rešenja, se naziva funkcija cilja ili funkcija ocene kvaliteta predloženog rešenja. Proces evolucije pojedinca u populaciji (misli se na rešenja, tj. na računarske programe) se može odvijati na dva načina:

- kreiranjem potpuno novih osobina pojedinca na osnovu okruženja u kome se nalazi i adaptaciji prema istom
- nasleđivanjem osobina od roditelja – novi program nastaje kombinovanjem dva programa – roditelja. Programi se pišu u formi drva, a uklanjanjem određenih grana iz jednog drva i postavljanjem tih grana u drugo drvo, nastaje novi program koji je, takođe, u formi drva.

5.5.5.2 Evolucione strategije

Evolucione strategije su evolucionarni algoritmi koji kao i genetski algoritmi svoju osnovu nalaze u biološkoj evoluciji. One se zasnivaju na primeni mutacije, rekombinacijama i selekciji u populacijama domena mogućih rešenja [54]. Sa algoritamske tačke gledišta, evolutivne strategije predstavljaju metode optimizacije koje se primenjuju u postupku uzorkovanja novih rešenja. Ovim metodama se postiže da je proces uzorkovanja stohastičan [54]. Kod dizajniranja evolutivnih strategija je potrebno da su sledeći principi zadovoljeni:

- nepristrasnost u odabiru rešenja
- prilagodiva kontrola parametara

Princip rada evolutivnih strategija se ogleda u tome da dolazi do ponavljanja malih stohastičkih varijacija nakon čega sledi selekcija: u svakoj generaciji, odnosno prilikom svake iteracije se dobijaju novi potomci, tj. kandidati koji predstavljaju nova rešenja. Ukoliko su ta rešenja bolja od prethodnih rešenja iz kojih su nastala, tj. ukoliko su bolja od svojih roditelja, tada se ona odabiraju da postanu roditelji za sledeću generaciju rešenja [55].

Da bi primena evolucionih strategija bila efikasna potrebno je sledeće [56]:

- Izvršiti odgovarajući odabir okruženja. Odabir okruženja je unapred određen iako postoji stohastička komponenta u odabiru verovatnoće.

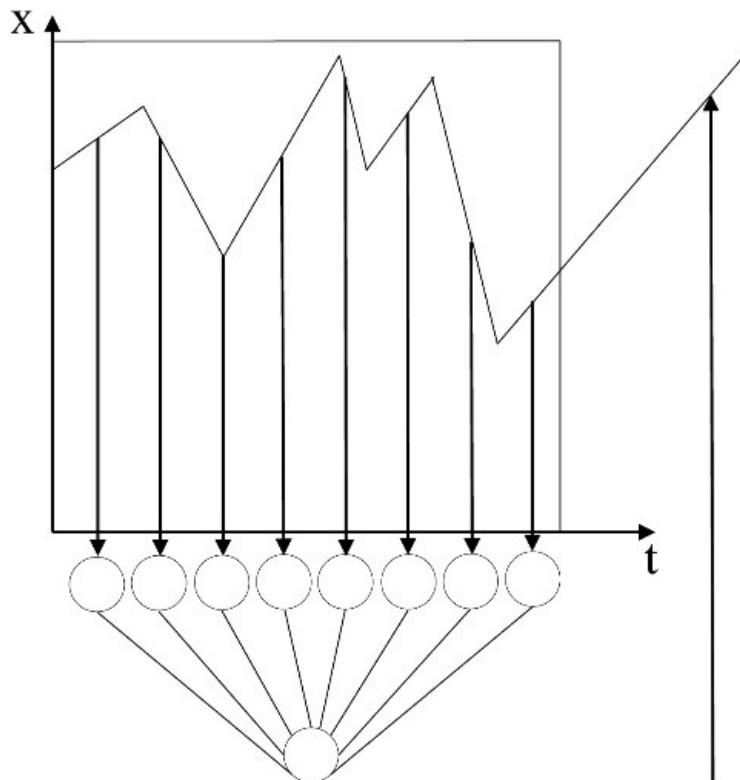
- Parenje i rekombinovanje. U toku procesa parenja se iz datog skupa rešenja biraju rešenja koja će dati nove potomke, tj. nova rešenja. Rekombinacijom se bira jedan potomak od svih novokreiranih potomaka, odnosno bira se najoptimalnije rešenje.
- Mutacija i kontrola parametara. Mutacija dovodi do malih, slučajnih i nepristrasnih izmena pojedinca. Prosečna veličina ovih promena zavisi od kontrolnih parametara, tj. endogenih parametara strategije. Ovi parametri definišu pojam „male izmene pojedinca“. Parametri kontrole ne nalaze uvek svoju osnovu u biološkoj evoluciji, ali su oni neophodni i predstavljaju najvažniju osobinu evolucionih strategija.
- Postojanje nepristrasnosti. Nepristrasnost je opšti princip dizajna strategija evolucije. Varijacije koje su nastale kao rezultat mutacije ili rekombinacija, uvode novu, nepristrasnu „informaciju“. Selekcija ovu „informaciju“ usmerava ka rešenjima sa većim stepenom prilagodljivosti datom problemu. Kod neutralne selekcije je poželjno da svi operatori varijacije budu nepristrasni. Na taj način će biti dobijeno maksimalno iskorišćenje novokreiranog rešenja.

5.6 Neuronske mreže i vremenske serije

Jedna od bitnijih primena neuronskih mreža se ogleda u procesu predviđanja vremenskih serija. Vremenske serije predstavljaju niz diskretnih vrednosti posmatrane veličine čije se promene prate u određenom periodu vremena [57]. Primeri vremenskih serija su:

- meteorološki podaci koji se mere i prate na dnevnom nivou.
- vrednosti cena akcija koje se prate svake sekunde na dnevnom, nedeljnom, mesečnom ili godišnjem nivou.

Pored diskretnih veličina elementi vremenske serije mogu biti i vrednosti neke kontinualne veličine koji se očitavaju u određenim vremenskm razmacima Δt [57]. Na slici 5.6.1. je prikazana neka kontinualna veličina x koja je podeljena na konačan (diksretan) broj delova u vremenu t , odnosno na ovaj način je definisana neprekidna funkcija $x = f(t)$. Diskretni uzorci kontinualne veličine mogu biti ulazni podaci neuronske mreže koja će, upravo, na osnovu tih ulaznih uzoraka moći da predvidi dalje ponašanje kontinualne funkcije.



Slika 5.6.1. Vremenske serije kontinualne veličine

Za predviđanje vremenske serije potrebna je neuronska mreža koja mapira prethodne vrednosti serije u budući razvoj vremenske serije (slika 5.6.1.). Da bi predviđanje neuronske mreže bilo precizno potreban je veći broj uzoraka analiziranog dela vremenske serije. Ovim postupkom se pokušava generalizacija procesa predviđanja budućih vremenskih serija kontinualnih veličina. Da bi proces predviđanja vremenskih serija mogao da započne potrebno je prvo razmotriti sledeća pitanja:

- Da li postoje dokazi da će vrednosti budućih vremenskih serija zavisiti od vrednosti analizirane vremenske serije? Ukoliko je odgovor na ovo pitanje Ne, onda proces predviđanja ne može biti definisan, tj. nema smisla kreirati neuronske mreže za predviđanje, jer buduće vrednosti vremenske serije ne zavise od vrednosti prethodno analizirane vremenske serije. U slučaju da je odgovor Da, onda se radi o sistemu koji je u potpunosti određen unapred (deterministički sistem), tako da će buduće stanje takvog sistema u potpunosti zavisiti od trenutnog stanja u kome se sistem nalazi. Da bi neuronske mreže mogle da predviđaju vremenske serije za probleme iz realnog sveta potrebno je uzeti u obzir sve faktore koji mogu uticati na dati sistem, što je veoma teško. Odavde sledi da je poznavanje

trenutnog stanja sistema i svih veličina koje mogu biti u interakciji sa posmatranim sistemom poželjno, ali ne i nophodno, a u većini slučajeva je i nemoguće. Izvlačenje zaključaka za neki mali skup podataka na osnovu ponašanja celog sistema kome taj skup podataka pripada može dovesti do davanja pogrešnih procena (primer je već spomenuta vremenska prognoza). Na osnovu vremenske prognoze za neki veći region (ili čak celu državu) u kome se nalazi neki mali podsistem (npr. voćnjak), pored nekih opštih faktora (temperature, vlažnost vazduha, atmosferskog pritiska, ...) uključuju i faktore koji su karakteristični za posmatrani podsistem, a koji mogu uticati na vremensku prognozu za dato podneblje (npr. u okolini voćnjaka se nalazi fabrika za preradu voća koja u rečni tok ispušta određene supstance koje će izazivati povećane atmosferskih padavina). Na ovom primeru je pokazano da je vrlo teško uzeti u obzir sve faktore koji mogu uticati na dati sistem. Zato se teži kompromisu koji će omogućiti postojanje određene greške u granicama tolerancije.

- Da li postoji dovoljan i potreban broj uzoraka vremenskih serija na osnovu kojih se mogu kreirati obrasci za treniranje neuronskih mreža? Po Nikvist-Šenonovoj (Nyquist-Shannon) [58] teoremi o uzorkovanju sledi da što je broj uzoraka koji će biti analiziran veći, to će predviđanje budućih vremenskih serija biti preciznije.
- Kako odabrati veličinu vremenskog razmaka, Δt , uzoraka u slučaju neprekidnih, tj. kontinualnih funkcija?

Najznačajniji pristupi za predviđanje vremenskih serija su:

- Predviđanje jedan korak unapred.

U ovom koraku sistem kao ulaz prima n prethodno posmatranih i analiziranih uzoraka. Na osnovu obrade primljenih uzoraka neuronska mreža predviđa ponašanje sistema jedan korak unapred, odnosno predviđa sledeće stanje u koje će sistem ući. Cilj ovog koraka je realizacija funkcije:

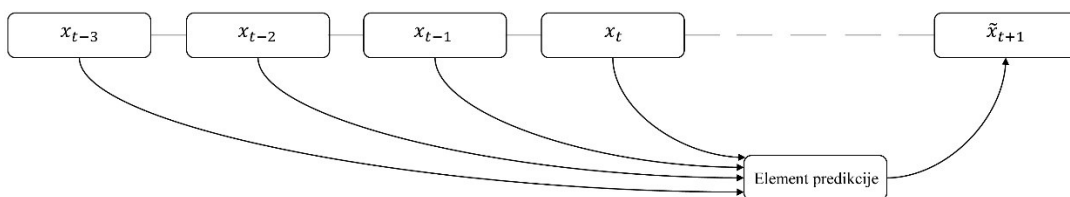
$$f(x_{t-n+1}, \dots, x_{t-1}, x_t) = \tilde{x}_{t+1} \quad (26)$$

U datoj jednačini simbol tilda \sim se koristi da bi se napravila razlika između veličina koja se predviđaju (\tilde{x}_t) i stvarnih budućih veličina koje će biti dostignute. Ukoliko se koristi neuronska mreža sa jednim slojem, tada je moguće primeniti linearni pristup rešenju postavljenog problema korišćenjem sistema koji se sastoji

od m jednačina sa n nepoznatih, pri čemu je radi što boljeg predviđanja poželjno da je $m > n$ i da je zbir kvadrata srednjih grešaka već postojećeg predviđanja minimalan. Prethodni uslov se naziva procedura pomeranja proseka, a ceo postupak se naziva ARMA (AutoRegressive Moving Average) metoda [57]. Sistem jednačina koji treba da se reši ima sledeći oblik:

$$\begin{aligned}
 x_t &= a_0 x_{t-1} + \dots + a_i x_{t-1-(n-1)} \\
 x_{t-1} &= a_0 x_{t-2} + \dots + a_i x_{t-2-(n-1)} \\
 &\cdot \\
 &\cdot \\
 &\cdot \\
 x_{t-n} &= a_0 x_{t-n} + \dots + a_i x_{t-n-(n-1)}
 \end{aligned}
 \tag{27}$$

Na slici 5.6.2 je prikazan princip rada neuronske mreže kao elementa koji služi za predviđanje budućih vremenskih serija.



Slika 5.6.2. Predviđanje jedan korak unapred

Problemi koji se zadaju neuronskoj mreži su problemi iz realnog života, pa upotreba jednoslojne neuronske mreže u cilju rešavanja celog problema nema smisla, jer su takvi problemi previše složeni za ovu vrstu neuronske mreže.

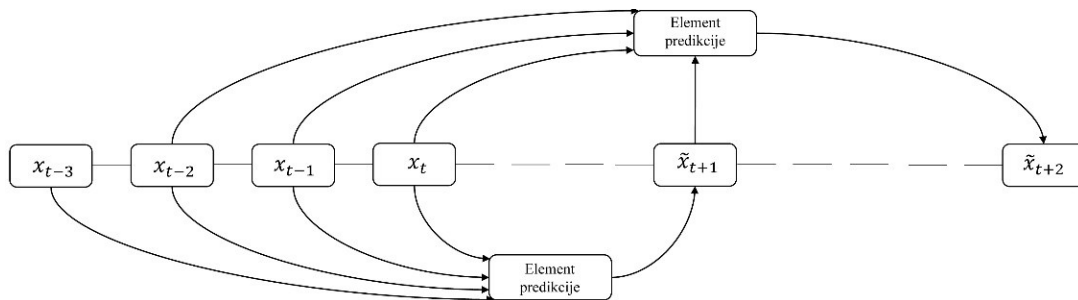
Primena višeslojnih neuronskih mreža i nelinearnih funkcija aktivacije daje mogućnost za kreiranje opšteg nelinearnog aproksimatora.

- Predviđanje dva koraka unapred

Postupak dva koraka unapred koji može biti primenjen kod budućih vremenskih serija ima dva različita pristupa i to:

- Rekurzivno predviđanje sa dva koraka unapred (slika 5.6.3.)

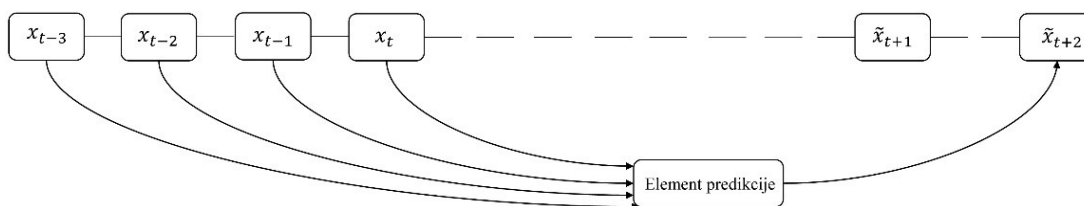
Vrednost određena pomoću predviđanja ove vrste neuronske mreže je neprecizna, tako da može doći do akumulacije grešaka tokom procesa rekurzije.



Slika 5.6.3. Rekurzivno predviđanje sa dva koraka unapred

- Direktno predviđanje dva koraka unapred

U ovom slučaju neuronska mreža se direktno trenira da predviđa vremenske serije dva koraka unapred. Ovaj način predviđanja vremenskih serija je isti kao i predviđanje jedan korak unapred. Razlika je u treningu neuronske mreže. Kod ove vrste neuronske mreže drugi vremenski korak je direktno predviđen dok je prvi vremenski korak izostavljen (slika 5.6.4.).



Slika 5.6.4. Direktno predviđanje dva koraka unapred

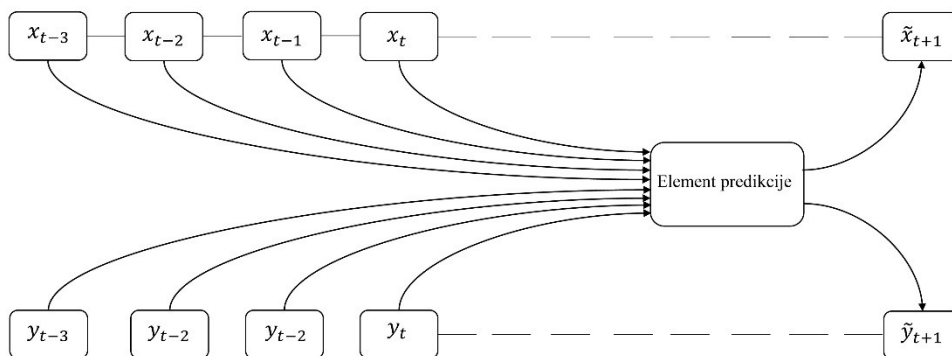
Da bi posmatrana neuronska mreža za predviđanje mogle da se koriste i u predviđanju periodičnih vremenskih serija potrebno je izvršiti dodatnu optimizaciju te mreže, odnosno treba da se izvrši prostorno-vremenska optimizacija neuronske mreže [59]. Dodatne optimizacije koje je moguće izvršiti su:

- Promena vremenskih parametara

Moguće je kombinovati različite vremenske intervale (Δt) prethodnih uzorkovanja. Da bi se, npr. rešio problem gužve u saobraćaju svakog petka od 17:00 časova do 20:00 časova u Beogradu, mogu se posmatrati rezultati koji su zabeleženi prethodnih nedelja i meseci za taj dan. Kombinujući različite vremenske intervale (Δt) uzorkovanja vremenskih serija koje su se već odigrale, moguće je predvideti ponašanje novih vremenskih serija, tj. predvideti koje će ulice biti najviše opterećene saobraćajem. Neuronske mreže za ovaj tip

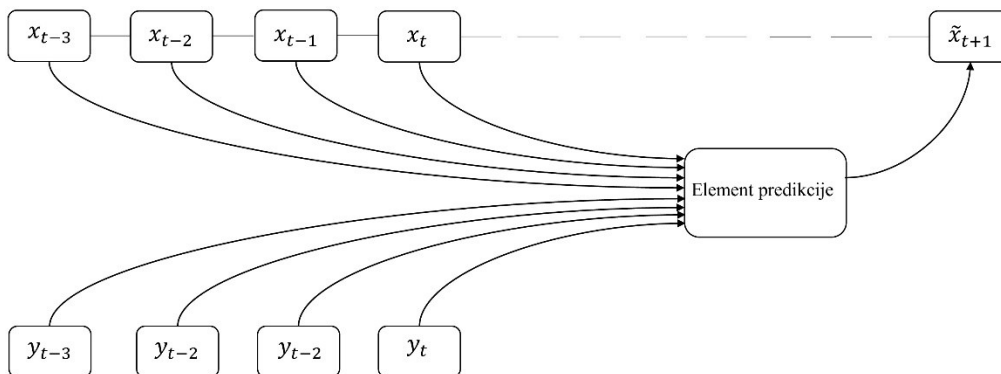
predviđanja u sebe uključuju dodatne faktore koji se mogu pojaviti u posmatranom vremenskom periodu (npr. asfaltira se neka ulica, koja je u prethodnom periodu bila u funkciji). Ovo je razlog što se zahteva veći broj kvalitetnih i pravovremenih podataka. Može postojati veliki broj podataka u vremenskoj seriji, ali možda ti podaci u sebi ne sadrže odgovarajuće informacije za posmatrani problem.

- Predviđanja iz različitih izvora vremenskih serija ili heterogena predviđanja se često koriste radi uprošćavanja početnih jednačina. Postoji dva tipa ove vrste predviđanja i to:
 - Heterogeno predviđanja kod koga se predviđaju vrednosti dve vremenske serije na osnovu dve međusobno povezane i prethodno analizirane vremenske serije. U ovom slučaju će se koristiti dva paralelna tzv. jedan korak unapred predviđanja (slika 5.6.5.).



Slika 5.6.5. Heterogeno predviđanje sa dva izlaza

- Predviđanja vrednosti jedne vremenske serije na osnovu dve međusobno povezane i prethodno analizirane vremenske serije (slika 5.6.6.).



Slika 5.6.6. Heterogeno predviđanje sa jednim izlazom

5.6.1 Učenje neuronskih mreža na osnovu postojećih vremenskih serija

Da bi problemi realnog sveta bili pretočeni u logički svet neuronskih mreža, signali iz realnog sveta moraju da budu konvertovani u binarna stanja logičke nule i logičke jedinice.

Neuronske mreže mogu da uče o ulaznim podacima i predviđaju buduće vrednosti na osnovu dva modela:

- modela crne kutije (eng. Black box model) [60]

Kod ovog modela učenja struktura neuronske mreže je nepoznata dok se u praksi prati njeno ponašanje. Sistem koji poseduje neuronsku mrežu prima signale iz svoje okoline i obrađuje ih. Na osnovu obrađenih signala on može da predvidi akciju koju će preduzeti ukoliko dođe do promene okruženja. Unutrašnja struktura neuronske mreže koja se koristi ostaje nepoznata korisniku sistema. Korisnik može da prati samo ponašanje sistema, ali ne i način na koj je uređaj obradio podatke i doneo odluke o svom ponašanju u datom okruženju.

- Model bele kutije (eng. White box model) [61]

Struktura sistema koji se analizira je unapred poznata i na osnovu te strukture je moguće izvršiti predviđanje budućih vremenskih serija.

5.7 Podela neuronskih mreža u zavisnosti od načina njihove implementacije

Po načinu implementacije veštačke neuronske mreže mogu biti sa:

- softverskom implementacijom
- hardverskom implementacijom

Veštačke neuronske mreže su najčešće implementirane u softver. Nedostatak ovakvih mreža je što aplikacije koje rade u realnom vremenu (eng. real-time) imaju duže vreme odziva u odnosu na hardverski implementirane mreže [62].

Neuronske mreže sa hardverskom implementacijom mogu biti sa digitalnim i analognim kolima.

Neuronske mreže sa digitalnim kolima nastaju korišćenjem CMOS (Complementary Metal-Oxide Semiconductor) [63] digitalnih kola [63]. Prednost ovih mreža je u jednostavnom dizajnu i lakoj izgradnji. Težine ulaznih signala se mogu implementirati

pomoću ćelija memorije digitalnog kola. Broj bitova koji može uskladištiti ulazne signale različitih koeficijenata težine zavisi od samog algoritma koji se implementira u neuronsku mrežu. Skladištenje ulaznih signala je naročito kritično u fazi učenja. Zbirno stanje neurona se implementira pomoću multipleksera.

Funkcija aktivacije je komplikovanija za implementaciju, jer je ona po definiciji nelinearna, što dovodi do sporijeg računanja izlaza. Sigmoidna funkcija je komplikovanija od funkcije praga prolaznosti, pa je za njeno računanje potrebno više vremena. Prednost hardverske implementacije je u činjenici da postojeće CMOS kolo u sistemu može biti korišćeno i za kreiranje veštačkih neuronskih mreža [65].

Neuronske mreže sa analognim kolima su teže za kreiranje u odnosu na neuronske mreže sa analognim kolima. Sa druge strane ove mreže mogu imati veću snagu i veću efikasnost u obradi primljenih signala. Kao i kod digitalnih neuronskih mreža i ovim mrežama se ulazi sa različitim koeficijentima težine čuvaju u ćelijama digitalne memorije. Ipak to će dovesti do kašnjenja u obradi signala, jer je potrebno izvršiti transformaciju analognog u digitalni signal pomoću ADC/DAC elemenata [66]. Eventualno rešenje može biti korišćenje analognih memorijskih medijuma kao što su to otpornici i kondenzatori.

Što se tiče softverske implementacije neuronskih mreža ona se najčešće koristi u slučaju tehnika rudarenja (eng. Deep Mining) velike količine podataka i davanja prognoza na osnovu analizirane količine podataka. Softverska implementacija predstavlja virtuelizaciju neurona i njihovo međusobno povezivanje u mrežu. Softver za implementaciju veštačkih neuronskih mreža je najčešće pisan:

- C-jezikom (proceduralno orijentisan jezik)
- C++ jezikom koji podržava Objektno-Orijentisano Programiranje (npr.glavna funkcija nije deo objekta)
- C#/Java jezikom (Objektno-Orijentisani jezici)
- Pajton (Python) je u upotrebi sve češće jer:
 - je kod u Pajtonu blizu pseudo-koda [57]
 - se odlikuje jednostavnom sintaksom,
 - programi pisani u Pajtonu su efikasni i dobro optimizovani
 - ima pravila koja omogućavaju eleganciju u pisanju koda (PEP8 [68])

- su programi pisani u Pajtonu relativno kratki u odnosu na druge programske jezike. Relativno kratak kod je omogućen korišćenjem tzv, pajplajnova (eng pipelines).
- Kao i programski jezik Java, Pajton radi na različitim platformama. Pajton može biti smatran i proceduralnim jezikom (kao što je to C jezik), ali i Objektno-orijentisanim jezikom (kao Java ili C#).

Algoritmi softvera koji se koristi za kreiranje veštačkih neuronskih mreža su relativno jednostavni za implementaciju, ali veliki deo vremena i ostalih resursa se troši na defisnisanju specifičnih zadataka, kao što su to obrada podataka i korisnički interfejsi.

Pisanje opšteg softvera za kreiranje veštačkih mreža nije moguće, jer napisani softver je primenjiv za određeni skup problema, što dovodi do pitanja komercijalizacije ovakvog tipa mreža. Pored toga korisnik softverski napisane neuronske mreže nema direktan pristup izvornom kodu što može ograničiti upotrebu te neuronske mreže kada naiđu složeniji problemi i kada je potrebno izvršiti modifikaciju izvornog koda.

Iz gore navedenih razloga se softverski način kreiranja veštačke neuronske mreže koristi u fazi testiranja neuronske mreže, dok se hardverski način kreiranja veštačke neuronske mreže koristi za konkretnu implementaciju kreirane mreže. Da bi kreirana veštačka neuronska mreža bila uspešno implementirana najbolje je koristiti oba pristupa pri njenoj implementaciji.

5.7.1 Implementacija neuronskih mreža korišćenjem koda napisanog u pajtonu

U ovom delu rada će biti predstavljeni originalni kodovi koji omogućavaju određivanje koeficijenata težine, odabir podele uzoraka i kreiranje određenih predviđanja. Ovaj deo je uvod za eksperimentalni deo koji će usledit u sledećem poglavlju.

Bitno je odrediti kvalitetne koeficijente težina. U pajtonu je najbolje koristiti numpy i metodu uniform koja pripada paketu numpy.random. Sledi kod sa histogramom koji pokazuje odabir slučajnih vrednosti koeficijenata težine iz uzorka od 2000 nasumično odabranih uzoraka. Slede kod i slike.

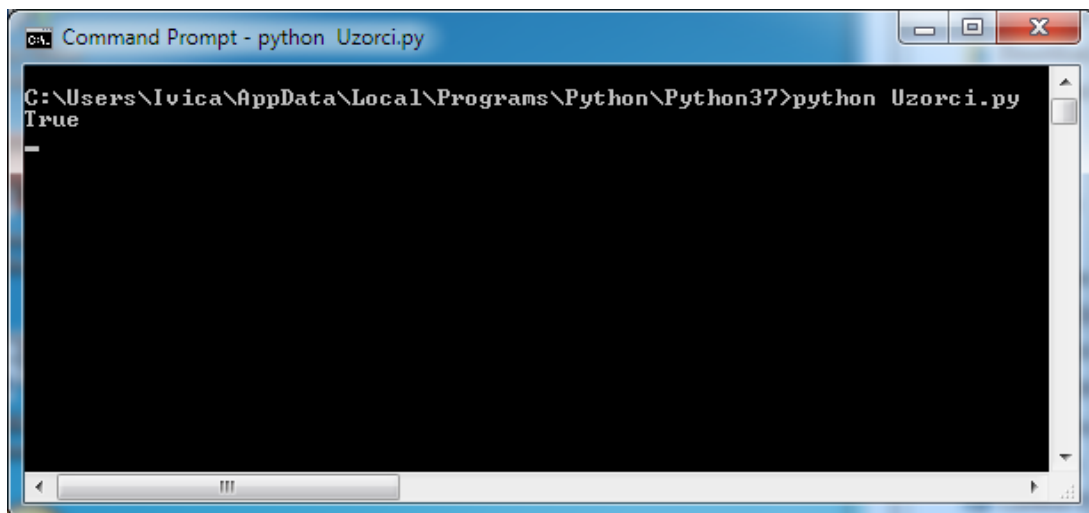
```
import numpy as np
```

```
# importuje se paket za crtanje grafa. matplotlib.pyplot je skup funkcija koje rade kao
```

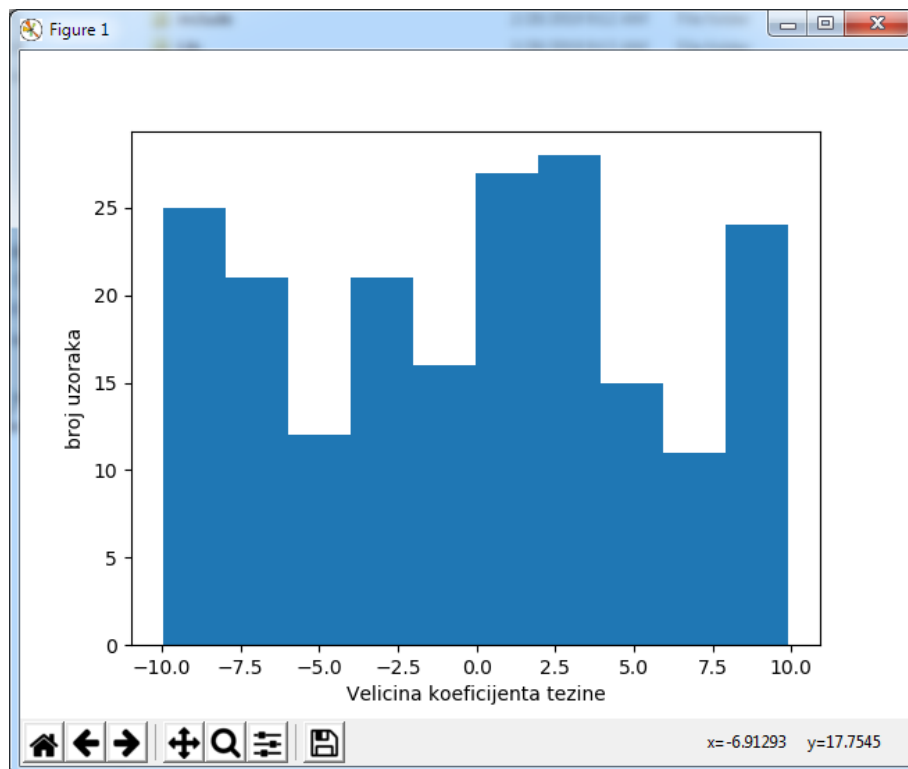
```

# MATLAB
import matplotlib.pyplot as plt
# dodeljuju se nazivi osama koordinatnog sistema
plt.xlabel('Velicina koeficijenta tezine')
plt.ylabel('broj uzoraka')
# definise se varijabila broj_uzoraka kojoj se dodeljuje vrednost 200
broj_uzoraka = 200
najnizi_koef_tezine = -10
najvisi_koef_tezine = 10
# Varijabili s se dodeljuju vrednosti uzoraka koji ce biti ravnomerno (funkcija uniform)
# rasporedjeni u intervalu -10 do 10, pri cemu je to interval zatvoren sa obe strane
s = np.random.uniform(najnizi_koef_tezine, najvisi_koef_tezine, broj_uzoraka)
# stampanje vrednosti u celom intervalu
print(np.all(s > -10) and np.all(s < 10))
# Kreiranje histograma i predstavljanje plota/grafa
plt.hist(s)
plt.show()

```



Slika 5.7.1.1. Pokretanje programa preko CMD shell-a



Slika 5.7.1.2 Određivanje koeficijenata težine nasumičnim odabirom iz skupa uzoraka

Sledi program koji određuje granicu prilikom donošenja odluke kod izraza regresije, a zatim i program koji će trenirati mrežu da izvrši ravnomernu podelu podataka.

Mali skupovi podataka kod obučavanja/treniranja velikih veštačkih neuronskih mreža mogu da izazovu probleme. Problem se ogleda u tome da model umesto da uči mapiranje od ulaza ka izlazu, može da prihvati primere koje je koristio u procesu treniranja kao stvarne ulaze i za njih da daje izlaze. To znači da će model uvek davati odlične rezultate kada je u pitanju ispitivanje skupa podataka koji pripadaju skupu za treniranje, ali će zato taj isti model davati loše rezultate kada naiđu novi podaci koji se ne nalaze u datom skupu podataka. Mali skup podataka daje mali broj mogućnosti za opis strukture izlaznog prostora sa jedne, a sa druge strane mali skup podataka ima mali broj relacija koje može da kreira sa izlazom. Ovi nedostaci kod malog skupa podataka mogu dovesti do problema povećanja ukupnog odstupanja dobijenog rezultata u odnosu na željeni rezultat, tj. dolazi do povećanja ukupne greške. Što je skup podataka veći to će opis problema biti precizniji i model će moći bolje da uči. Predočeni problemi kod malog skupa podataka u nekim slučajevima ne mogu biti sprečeni prikupljanjem većeg broja podataka.

Jedno od rešenja ovog problema je dodavanje šuma (eng. noise). Šum, u modelima za predviđanje, predstavlja nezavisne nasumične podatke koji mogu uticati na podatke koji ulaze u komunikacioni kanal, tj. šum može uticati na koeficijente težine kod neuronskih mreža. Za razliku od šuma, signal u modelima za predviđanje, predstavlja istiniti obrazac koji je poželjno naučiti iz skupa podataka [71]. Šum može da bude, pored ulaza, pridodat i koeficijentima težine i funkcijama aktivacije. Šum se dodaje koeficijentima težina kada su u pitanju rekurentne neuronske mreže [72]. Dodavanje šuma mreži značiće da će mreža biti manje sposobna da memoriše uzorke obuke/treninga, jer će se ti uzorci stalno menjati što dovodi do smanjenja koeficijentata težine ulaznih podataka, ali i do povećanja robusnosti mreže, tj. povećanja sposobnosti mreže da se bolje nosi sa greškama ulaza [73]. Uzorci koji će predstavljati šum se izvlače iz domena koji se nalazi u blizini poznatih uzoraka. Šum pridodat uzorcima i uzorci ulaza zajedno čine ulazni prostor. Svaki put kada se u procesu treniranja mreže, mreži predstavi novi uzorak za treniranje tom uzorku će biti pridodat i različit šum što dovodi do toga da ulazni uzorak za trening ima različitu vrednost svaki put kada je predstavljen modelu neuronske mreže. Ovim dodavanjem šuma se povećava skup ulaznih podataka u neuronsku mrežu. Najčešći tip šuma koji se dodaje ulazima je tzv. Gausov (nazvan i beli) šum. Gausov šum je statistički šum kod koga je funkcija gustine verovatnoće (Probability Density Function - PDF) jednaka funkciji normalne raspodele, tj. vrednosti Gausovog šuma podležu zakonu Gausove normalne distribucije [74]. Za kreiranje Gausovog šuma može se koristiti generator pseudoslučajnih brojeva.

Šum može biti dodat bilo kojoj vrsti neuronske mreže u toku procesa njenog treniranja. Da bi efekat dodavanja šuma imao smisla, potrebno je pre dodavanja šuma da svi ulazni podaci, tj. varijabile budu predefinisane na taj način da sve vrednosti ulaznih podataka budu u okviru istog opsega. Proces predefinisavanja ulaznih podataka je moguće uraditi putem standardizacije ulaznih podataka. Ponekad je, slučaju dodavanja nasumičnih vrednosti šuma, potrebno da se izvrši ponovna rekalkibracija ulaznih podataka (iako su ti ulazni podaci prethodno rekalkibrisani).

U procesu dodavanja šuma ulaznim podacima, ne može se unapred znati kolika je potrebna količina šuma koju treba dodati. U tu svrhu je potrebno uraditi niz eksperimenata sa različitim količinama dodatog šuma da bi mogao da bude izabran adekvatan model treniranja neuronske mreže. Da bi eksperimenti bili uspešni potrebno je raditi sa manjim

skupovima ulaznih podataka različitih opsega. Šum se dodaje samo u procesu treniranja neuronske mreže.

Sledi kod (sa komentarima) za grafičko predstavljanje skupa od 250 ulaznih uzoraka.

```
import matplotlib.pyplot as plt
import numpy as np
# scikit-learn ili sklearn je paket alata koji omogućava data mining i analizu podataka
import sklearn
import sklearn.datasets
import sklearn.linear_model
import matplotlib as mpl
# np.random.seed() omogućava da slučajni brojevi (funkcija random iz paketa np) budu
# predvidivi (metoda seed).
# Pseudo-slučajni (pseudo-random) brojevi su slučajno odabrani brojevi iz skupa
određenih
# brojeva, pri čemu
# je početna vrednost tog skupa slučajnih brojeva poznata i taj skup slučajnih brojeva ce
# uvek biti isti bez obzira
# što se koristi generator pseudo-slučajnih brojeva. Termin pseudo se koristi, jer
algoritam
# koji se koristi za kreiranje skupa slučajnih brojeva može ponoviti sekvencu niza
slučajnih
# brojeva, pa ta sekvencija onda neće predstavljati slučajni odabir brojeva i samim tim
ceo
# postupak odabira slučajnih brojeva nije u potpunosti slučajni.
# Proces generisanja pseudo-slučajnih brojeva se zasniva na principu množenja početne
# vrednosti nekim velikim brojem i zatim traženjem modula dobijenog broja. Sledeća
# vrednost skupa se dobija tako što se dobijena vrednost množi velikim brojem i traži se
# moduo tog broja,... Početna vrednost skupa pseudo-slučajnih brojeva se naziva seed
# (seme). Ukoliko je seed isti tada će se pri svakoj novoj upotrebi generatora pseudo-
# slučajnih brojeva dobiti isti skup slučajnih brojeva.
# Upravo je to razlog upotrebe seed metode.
# seed() metoda može da primi intenzere ili nizove intenzera različitih dužina.
```

```
# Kada bi u seed metodi stajao neki broj tada bi pri svakom pokretanju programa bili  
# generisani isti brojevi, jer taj pocetni broj predstavlja  
# "seme" od koga pocinje racunanje slucajnih promenljivih. Ovo se koristi u toku procesa  
# testiranja odredjenih rezultata i potvrđivanja rezultata koje oni daju.  
# Posto u seed metodi nema broja, vec stoji None to ce nakon svakog puta generator  
pseudo-  
# slucajnih brojeva da generise razlicite brojeve.  
# numpy ce u slucaju seed(None) da generise slucajne brojeve iz dev/urandom (za Linux  
# operativne sisteme) ili  
# iz odgovarajuceg foldera za Windows operativne sisteme. U suprotnom slucajne  
vrednosti  
# mogu biti generisane iz # ugradjenog casovnika.  
np.random.seed(None)  
# Koristi se metoda make_moons koja vrsi binarnu raspodelu, tj. to je alat za  
vizuelizaciju.  
# Skup podataka ce se sastojati od 250 uzoraka koji ce biti rasporedjeni po x i y osi.  
# Bice dati primeri razlicitih vrednosti parametra noise. Sto parametar noise ima vecu  
# vrednost to ce uzorci biti rastrkaniji po grafu. U suprotnom uzorci ce biti potpuno  
odvojeni.  
# Kod ce biti isti samo ce u drugom slucaju noise biti 10, a u trecem slucaju noise ce imati  
# vrednost 0.000001  
# X je ime varijabile. To nije ime argumenta. Ova varijabila je dvodimenzionalni niz koji  
se  
# sastoji od tri kolone:  
# prva kolona ili x-varijabila grafa, druga kolona ili y-varijabila grafa i treca kolona  
koja  
# odredjuje velicinu  
# tacaka na grafu. Ulazne velicine ce biti x i y vednosti, dok ce izlazne vrednosti biti  
clanovi  
# izlazne klase (0 ili 1).  
X, y = sklearn.datasets.make_moons(250, noise=0.25)
```

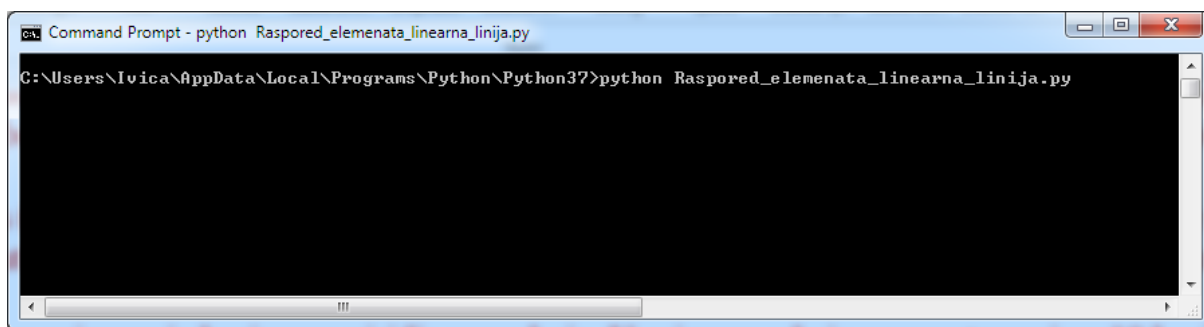
```

# Funkcija scatter koja je pozvana iz paketa pyplot kreira sketer grafik. Pozicija
pojedinačne
# tacke zavisi od njene dve dimenzije: horizontalne i vertikalne vrednosti. U ovom slucaju
#[:,0] je prva koordinata dvodimenzionalnog niza, dok je X[:,1] druga koordinata
# dvodimenzionalnog niza.
# 0 i 1 u ovim koordinatama oznacaju da se radi o binarnoj raspodeli.
# funkcija scatter moze da renderuje svaku tacku raylicito (oblik, boja, dimenzija), dok kod
# plot funkcije su sve tacke
# medjusobno klonovi. Upravo zato je kod velikog skupa podataka korisnije plot funkcije
# efikasnije. Atribut marker odredjuje izgled tacaka na grafu (kruzic, kvadrat, ...). Ukolik
ose
# ovaj parametar izostavi
# podrazumevana vrednost ovog atributa ce biti kruzic. Atribut s (size) odredjuje velicinu
# tacaka na grafu.
# Atributom c se dodeljuje boja mapi, cmap se postavlja da ima jet semu boja. U ranijim
# verzijama je ova sema boja bila podrazumevana. Ako se nestavi cmap onda ce
# podrazumevana sema boja, od verzije matplotlib 2.0.0
# biti viridis (lat. zelen). Opseg boja Viridis prikazuje jasnije uzorke (jet opseg moze da
# prikaze kontraste i gde ih nema).
# Atribut cmap (od colormap) je interfejs izmedju obradjenih i prikazanih podataka i
njihove
# iterpretacije u mozgu coveka.
# U ovom primeru je koriscena viridis sema boja. U primeru gde je parametar noise=10
# koriscenja je jet sema boja.
# Atribut edgecolors je boja ivice objekta.
# Binarna raspodela u opsegu date seme boja.
plt.scatter(X[:,0], X[:,1], marker='o', c=y, s=30, cmap='viridis', edgecolors='#000000')
# Metoda colorbar() pokazuje sa strane opseg boja.
plt.colorbar();
plt.show()

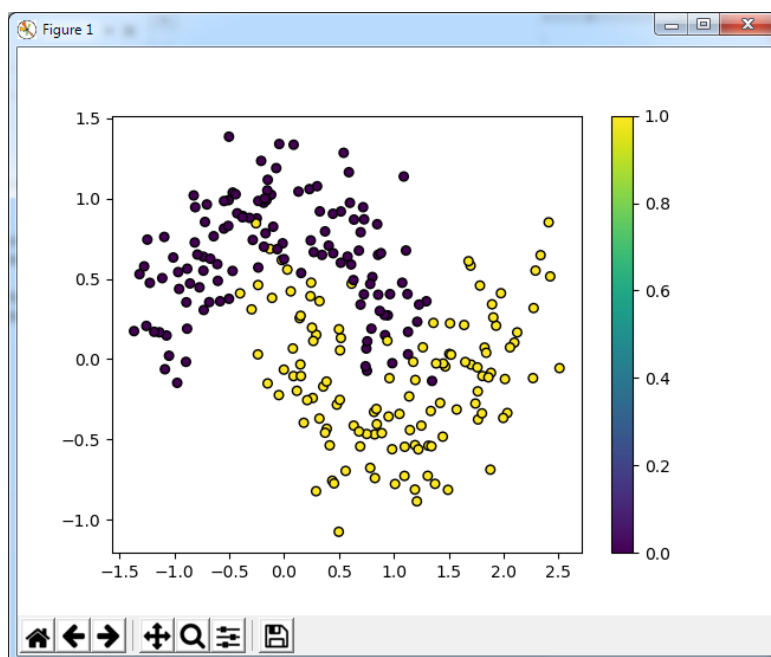
```

Na sledećim slikama se vidi šta se dešava sa rasporedom ulaznih podataka kada je parametar *noise=10* i kada je parametar *noise = 0.000001*. U prvom slučaju je vrlo teško

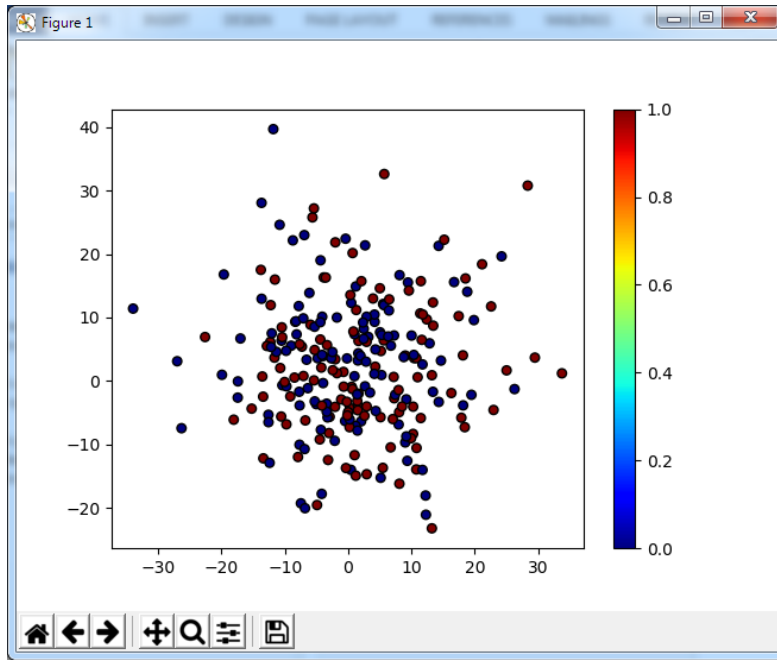
uraditi linearno razgraničenje uzoraka. Pored toga, u prvom slučaju, je korišćena vrednost atributa `cmap='jet'` koja (kao što je u komentaru koda navedeno) dovodi do zamućenja grafa, jer atribut `jet` može da prikaže kontraste tamo i gde ih nema. U drugom slučaju dobijaju se glatke krive što nije cilj, jer je onda mreža nepogodna za treniranje pošto će model uvek davati odlične rezultate kada je u pitanju ispitivanje skupa podataka koji pripadaju skupu za treniranje



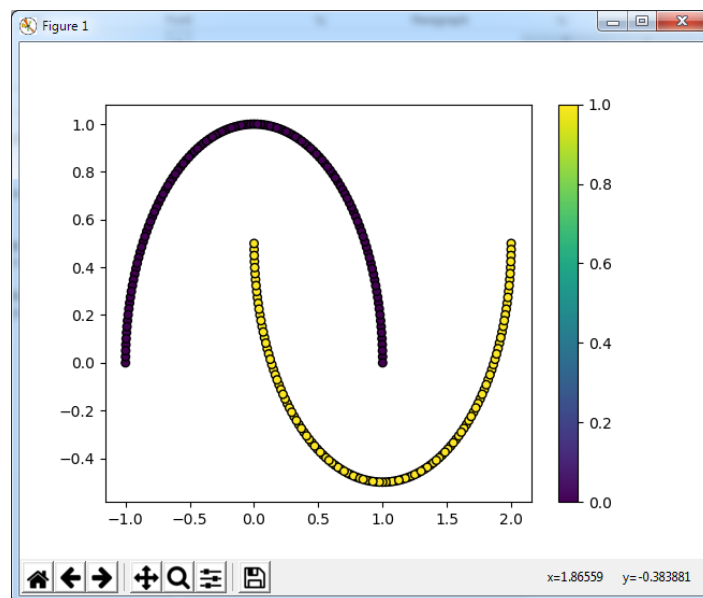
Slika 5.7.1.3 Pokretanje programa preko CMD shell-a



Slika 5.7.1.4. Skup uzoraka kada je noise parametar jednak 0.25



Slika 5.7.1.5. Skup uzoraka kada je noise parametar jednak 10 (*cmap='jet'*)



Slika 5.7.1.6 Skup uzoraka kada je noise parametar jednak 0.000001

Sa prethodnih slika se vidi da pokrenuti kod kreira dve klase podataka.

U sledećem koraku je potrebno izvršiti kreiranje granice koja će određivati kojoj klasi izabrani subjekat, tj. uzorak pripada. U kodu koj sledi biće urađena logistička regresija. Logistička regresija je tzv. idi-ka (eng. go-to) metoda za rešavanje problema binarne

klasifikacije [75]. Srž logističke regresije je logistička funkcija tj. sigmoidna funkcija čija definicija i karakteristike su date ranije u okviru ovog rada.

```
import matplotlib.pyplot as plt
import numpy as np
import sklearn
import sklearn.datasets
import sklearn.linear_model
import matplotlib as mpl
# Ovaj deo koda je objasnen u kodu prethodnog primera.
np.random.seed(None)
X, y = sklearn.datasets.make_moons(250, noise=0.25)
# Kreiranje klase za predviđanje (dato ime klasi je klasifikator, jer su ove klase za
# predviđanje nazivaju klasifikatori).
# Objekat koji predstavlja instancu klase klasifikator ce se popuniti vrednostima koje su
# dobijene primenom metode
# najmanjih kvadrata (metoda linear_model u okviru sklearn paketa) koja se koristi kod
# linearne agresije.
# Metoda linear_model se poziva iz paketa sklearn.
# Metoda linear_model ce biti primenjena nad klasifikatorom LogisticRegressionCV().
# Pored ovog klasifikatora u paketu sklearn postoji i klasifikator LogisticRegression().
# Razlika izmedju ova dva klasifikatora je u tome sto:
# instanca klasifikatora LogisticRegression() samo trenira logicku regresiju nad datim
# podacima.
# Instanca klasifikatora LogisticRegressionCV() deli dati skup podataka u razlicite
# treniraj/proveravaj kombinacije i to pre samog procesa treniranja. To dovodi do toga
da
# model ima visoku efikasnot kod primene nad svim kreiranim skupovima podataka. Ovaj
# postupak se naziva K-Fold Cross Validation.
# Pozivanjem metode fit dati model se trenira.
klasifikator = sklearn.linear_model.LogisticRegressionCV()
klasifikator.fit(X, y)
# Kreira se funkcija pod imenom "granica" koja ce omoguciti iscrtavanje podele izmedju
```

```

# uzoraka na grafiku. Funkciji # granica je dodeljen jedan parametar pod imenom
# podela.
# Da bi linija podele bila uspesno kreirana potrebno je
# definisati dve vrednosti u odnosu na koje ce, kasnije, biti rasporedjeni ostali uzorci.
# Parametar d predstavlja debljinu # granicne linije.
def granica(podela):
    x_min, x_max = X[:, 0].min() - .9, X[:, 0].max() + .9
    y_min, y_max = X[:, 1].min() - .9, X[:, 1].max() + .9
    d = 0.01
# Poziva se metoda meshgrid() iz paketa numpy. Ova metoda kreira mrezu tacaka na
# grafiku.
# Metod arange ima parametre (start, stop, step), tj.(min, max i d). Primena ove metode
# omogucava rasporedjivanje tacaka u mrezi.
    xx, yy = np.meshgrid(np.arange(x_min, x_max, d), np.arange(y_min, y_max, d))
# Predvidjanje funkcije za celu mrezu
# np.c prevodi isecene objekte u seriju vrednosti koje ce biti povezane duz druge
# ose.Iseceni
# objekat jeste niz.
# metoda ravel() ce prevesti mulidimenzionalni objekat u jednodimenzionalni niz. Vazi za
# obe koordinate.
# Metoda reshape() daje novi oblik nizu bez da menja vrednosti samog niza, odnosno
# omogucava da matrica bude preoblikovana u vektor. Ovo je razlog kreiranja objekta
# Novi_objekat.
    Novi_objekat = podela(np.c_[xx.ravel(), yy.ravel()])
    Novi_objekat = Novi_objekat.reshape(xx.shape)
# Primena lamda funkcije na klasifikator.
# Lambda je anonimna funkcija u Pajtonu koja moze imati bilo koji broj argumenata, ali
# samo jedan izraz.
# Lamda funkcija se moze koristiti bilo gde u kodu kada je potrebno. Kod se skrakuje.
Umesto
# lambda funkcije je moguće definisati novu funkciju, ali onda mora da se definisu i
# parametri i sta ona vraca kao rezultat.

```

```

# Funkcija granicca uzima ceo izraz u zagradi kao argument i vraca
# Metodu predict primenjujemo na klasifikator.
plt.contourf(xx, yy, Novi_objekat, cmap='viridis')
plt.scatter(X[:,0], X[:,1], marker='o', c=y, s=30, cmap='viridis',
edgecolors='#000000')
# Crtanje granice odlucivanja
granica(lambda x: klasifikator.predict(x))
plt.title("Logistička regresija")
plt.xlabel('x-osa')
plt.ylabel('y-osa')
plt.colorbar()
plt.show()

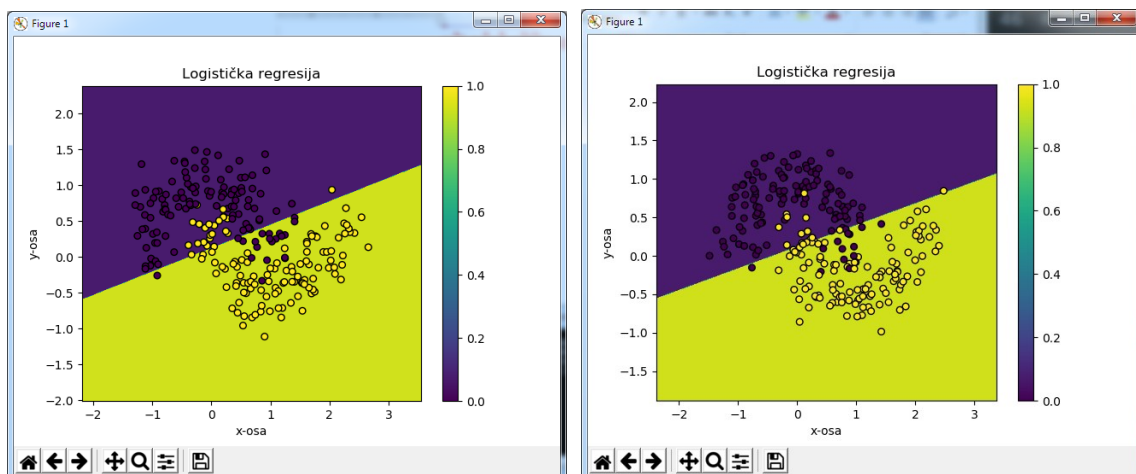
```

```

C:\Users\Ivica\AppData\Local\Programs\Python\Python37>python logistica_regresija.py
C:\Users\Ivica\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\model_selection.py:100: FutureWarning: The default value will change from 3 to 5 in version 0.22.
warnings.warn(CU_WARNING, FutureWarning)

```

Slika 5.7.1.7. Pokretanje koda iz CMD sela (shell)

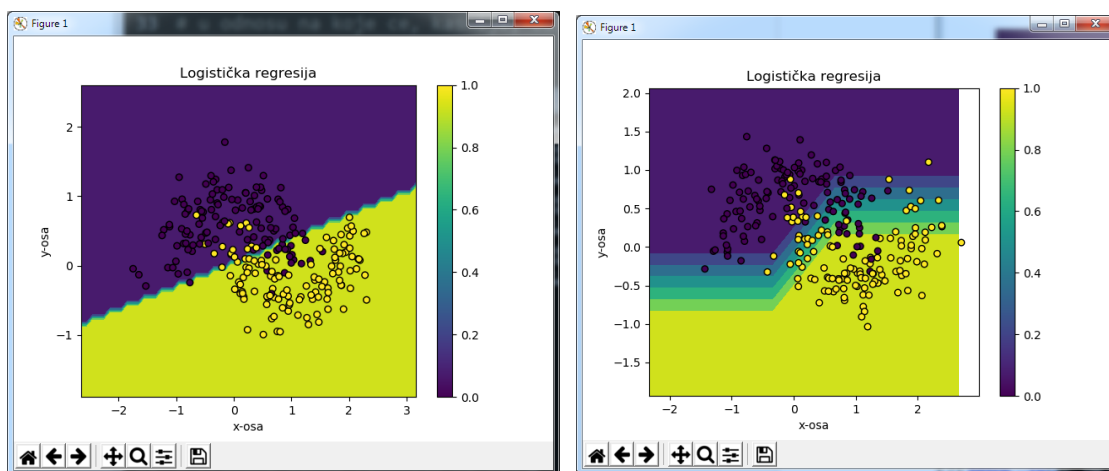


Slika 5.7.1.8. Logistička regresija

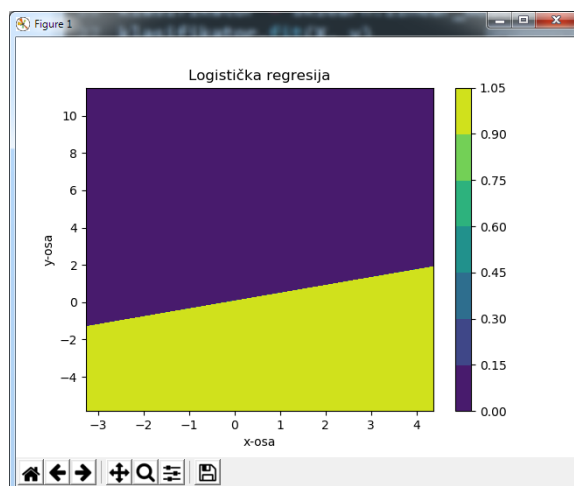
Na gornjoj slici je demonstrirana upotreba generatora pseudo-slučajnih brojeva. Kako je seed() metodi pridodat None parametar, to će svaki put prilikom pokretanja koda doći

do stvaranja nove raspodele uzoraka. To otežava određivanje granice između uzoraka. Potrebno je izvršiti nekoliko eksperimenata da bi granica bilo određena na kvalitetan način.

Slede slike koje pokazuju kako odabir parametara može da utiče na postavljanje granice i debljinu granice. Određivanje parametara koji određuju položaj granice i debljinu linije razdvajanja direktno utiče na klasifikovanje uzoraka. Ukoliko je klasifikovanje uzoraka urađeno kako treba onda će i proces treniranja mreže, koji sledi nakon ove podele, biti uspešan.



Slika 5.7.1.9. Parametar d ima vrednost 0.1 i parametar d ima vrednost 1



Slika 5.7.1.10 Promena položaja linije granice

Na prethodnoj slici dolazi do pomeranja granice podeljenosti uzoraka usled promena vrednosti min i max u kodu, tj. umesto linija koda:

$$x_min, x_max = X[:, 0].min() - 5, X[:, 0].max() + 1$$

$y_{min}, y_{max} = X[:, 1].min() - 5, X[:, 1].max() + 1$

stavljene su linije koda

$x_{min}, x_{max} = X[:, 0].min() - 2, X[:, 0].max() + 2$

$y_{min}, y_{max} = X[:, 1].min() - 5, X[:, 1].max() + 10$

Pored toga radi bolje prezentacije linije granice izostavljena je linija koda:

```
plt.scatter(X[:,0], X[:,1], marker='o', c=y, s=30, cmap='viridis', edgecolors='#000000')
```

Izostavljanjem ove linije koda izbačeni su uzorci sa grafa.

Problem kod ovakve klasifikacije je u tome što nije moguće uraditi podelu koja nije linearna, odnosno potrebno je uraditi dosta eksperimenata da bi se dobila zadovoljavajuća linearna podela (prikazano na prethodnim slikama). Što je više puta ponavljan proces deljenja u cilju traženja optimalnog rešenja za granicu podele, to će ukupni proces duže trajati što dovodi do povećanja troškova. Potrebno je izvršiti podelu koja neće biti samo linearne prirode već će imati oblik polumeseca (moon-shape).

U sledećem kodu će biti prikazan način treninga neuronske mreže. Da bi trenirali neuronsku mrežu potrebno je da postoji bar jedan skriveni sloj. Neuronska mreža koja će biti demonstrirana kodom koji sledi će imati tri sloja: ulazni, jedan skriveni sloj i izlazni sloj, tj. mreža koja će biti trenirana imaće najprostiju formu. Sledi struktura slojeva po broju čvorova, tj. nodova koje će posedovati:

- ulazni sloj: Ovaj sloj će imati dva čvora, jer se razmatra podela dve vrste ulaznih podataka, tj. niz će biti dvodimenzionalan. Svaka od ulaznih vrednosti će imati dve koordinate: x-koordinatu i y-koordinatu.
- skriveni sloj: Radi dobijanja bolje podeljenosti ulaznih podataka i što kvalitetnijeg učenja mreže potrebno je koristiti veći broj čvorova. Primena više čvorova u skrivenom sloju dovodi do primene funkcije koštanja na datu neuronsku mrežu tj. upotrebom većeg broja čvorova u skrivenom sloju će biti žrtvovane vrednosti određenih parametara neuronske mreže. Jedan od parametara koji će pretrpeti promene, usled povećanja broja čvorova u skrivenom sloju, je povećanje složenosti proračunavanja izlazne vrednosti. Broj čvorova u skrivenom sloju zavisi od datog problema i načinu pristupa datom problemu, tj. zavisi od tačke gledišta na dati problem onoga ko kreira taj skriveni sloj. Za isti problem se može koristiti

skriveni sloj sa različitim brojem čvorova. Vrednosti izlaznog sloja će zavisiti od broja čvorova u skrivenom sloju.

- izlazni sloj: Ovaj sloj se sastoji od dva čvora, jer je potrebno izvršiti podelu dve vrste ulaznih podataka, pa će, bez dodatnih uslova, i sam izlaz imati dve vrste izlaznih podataka. U slučaju ovako proste neuronske mreže je moguće koristiti i jedan čvor u izlazu koji će predviđati da li će na izlazu biti nula ili jedinica. Prednost korišćenja izlaza sa dva čvora leži u činjenici da je moguće, relativno jednostavno, takav izlaz nadograditi sa još nekim čvorom. U tom slučaju će izlazni podaci biti klasifikovani u više u dve klase. Broj klasa će zavisiti od broja čvorova u izlazu.

Pored odabira broja čvorova (neurona) potrebno je odabrati i odgovarajuću funkciju aktivacije za skriveni sloj. Da bi bila izvršena nelinearna podela potrebno je koristiti aktivacionu funkciju koja će omogućiti takav vid podele ulaznih podataka. Funkcija koje se mogu koristiti za nelinearnu podelu su:

- sigmoidna funkcija: opisana u prethodnom delu rada
- tanh funkcija: daje tangentu na posmatranu hiperboličku funkciju. Hiperboličke funkcije su funkcije koje se odnose na eksponencijalnu funkciju i koje važe u ne-Euklidskom prostoru [76]. Euklidska ravan u ne-Euklidskom prostoru se naziva Orisfera. Važi i da u Euklidovom prostoru postoji ne-Euklidska ravan koja se naziva pseudosfera. Oznake hiperboličkih funkcija su slične oznakama trigonometrijskih funkcija (koje važe u Euklidovom prostoru): $\sinh x$, $\cosh x$, $\tanh x$, $\coth x$, $\operatorname{sech} x$, $\operatorname{cosech} x$. Formula za $\tanh x$ [77] je:

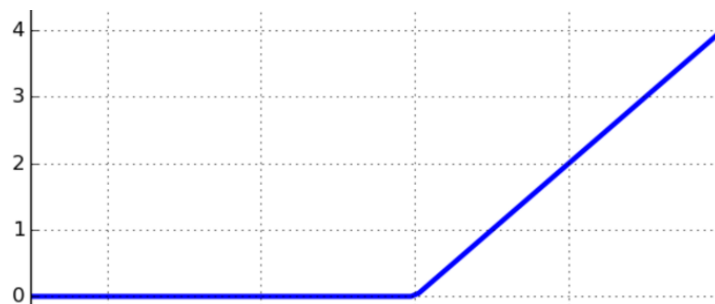
$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (28)$$

Vrednost izlaza koju može dati $\tanh x$ funkcije se kreće u opsegu od -1 do 1 . Sigmoidna funkcija, kako je već u radu navedeno, može dati vrednost izlaza koji će se kretati u opsegu od 0 do 1 . Funkcija $\tanh x$ je sličnog oblika kao i sigmoidna funkcija. Prednost proširenog opsega $\tanh x$ funkcije omogućava da su neuroni kod ove funkcije orijentisani ka nuli [78].

- ReLU (Rectified Linear Unit) funkcija: je zadata formulom:

$$f(z) = \begin{cases} 0 & , \quad x < 0 \\ x & , \quad x \geq 0 \end{cases} \quad (29)$$

Funkcija je nula za negativne vrednosti, a linearno raste za pozitivne vrednosti [79]. ReLU funkcija ne aktivira sve neurone u isto vreme, jer kada je ulazna vrednost negativna, ReLU funkcija ima vrednost nula, pa ne dolazi do aktivacije neurona. Ovo čini mrežu optimalnom i efikasnom, jer nisu u isto vreme svi neuroni aktivirani. Kao posledica toga proces računanja izlaza iz neurona koji se nalaze u skrivenom sloju se pojednostavljuje. Osnovna prednost ReLU funkcije je da sa relativno malim brojem parametara može biti predstavljen veliki broj funkcija [80]. Sledi slika ReLU funkcije.



Slika 5.7.1.11 ReLU funkcija (Izvor: [79])

Aktivaciona funkcija izlaznog sloja koja će biti korišćena u datom primeru je softmax funkcija. Softmax funkcija će dobijene podatke transformisati u verovatnoće čiji će raspon biti [0,1]. Softmax funkcija se koristi u modelima sa više klasa, a kao rezultat primene softmax funkcije se dobijaju verovatnoće za svaku od klasa u modelu [77]. Najčešći oblik formule koja opisuje softmax funkciju je [81]:

$$\sigma(z) = \frac{e^{\lambda z_i}}{\sum_{j=1}^n e^{\lambda z_j}} \quad (30)$$

gde su: λz_i skup svih elemenata koji pripadaju datoj klasi i λz_j rezultati koje je izračunala neuronska mreža za svaku od klasa u datom skupu klasa koji se sastoji od jedne (1) do n klasa.

Za dvodimenzionlani ulaz (varijabila x) predviđanje u oznaci \hat{y} će biti izračunato po formulama:

$$z_1 = x_1 * \omega_1 + b_1 \quad (31)$$

$$a_1 = \tanh(z_1) \quad (32)$$

$$z_2 = a * \omega_2 + b_2 \quad (33)$$

$$\hat{y} = \text{softmax}(z_2) \quad (34)$$

Prvi korak u dobijanju vrednosti izlaza iz izaznog sloja neuronske mreže je primena softmax funkcije na izračunavanje verovatnoća vrednosti koje se mogu javiti na izlazu. Drugi korak je primena funkcije koštanja *cross-entropy* na dobijene rezultate softmax funkcije.

Da bi greška bila minimalna potrebno je koristiti funkciju koštanja/gubitka. Najčešća funkcija koštanja koja se koristi kod *softmax* funkcije je *cross-entropy* (unakrsna-entropija) funkcija. Kod primene funkcije koštanja unakrsna-entropija u kreiranju neuronskih mreža teži se tome da primenjena funkcija koštanja zadrži maksimalnu verodostojnost u odnosu na originalnu funkciju, jer na taj način neće doći do usporavanja procesa učenja [82]. Upravo ta verodostojnost omogućava da izvod funkcije unakrsna-entropija može biti korišćen kao funkcija koštanja [82]. Formula funkcije unakrsna-entropija za pojedinačan neuron je:

$$CE = -\frac{1}{n} \sum_{i=1}^C y_i \ln(a_i) \quad (35)$$

gde su: n – broj neurona, C – broj klasa u koje treba klasifikovati probleme, a_i – je funkcija aktivacije za koju važi $a_i = \sigma(z_i)$, y_i – željena vrednost.

Gornja formula može biti zapisana i u obliku:

$$CE = -\frac{1}{n} \sum_{i=1}^C [y \ln a + (1 + y) * \ln(1 - a)] \quad (36)$$

Iz gornje formule se može izvući zaključak da je funkcije unakrsna-entropija nenegativna funkcija, tj. $C > 0$. Razlog leži činjenicama da su vrednosti oba prirodna logaritima negativne i da se ispred sume nalazi znak minus. Vrednosti za koje se traži prirodni logaritama leže u opsegu (0,1), a rezultat prirodnog logaritima za taj opseg vrednosti je, po definiciji logaritima, negativan. Što je izlazna vrednost neurona bliža željenoj vrednosti, to će vrednost funkcije unakrsna-entropija težiti nuli. Ovo važi pod pretpostavkom da su izlazne vrednosti ili 0 ili 1 (što je slučaj kod klasifikacije podataka). Pozitivna vrednost funkcije koštanja i njena težnja nuli kada se dobijene vrednosti izlaza te funkcije približavaju željenim vrednostima, svrstava funkciju unakrsna-entropija u funkciju koštanja.

Parametri koeficijenti težine i bajas su parametri koje neuronska mreža treba da proračuna i odredi da budu takvi da greška koja se javlja pri treniranju neuronske mreže bude minimalna. Zbog traženja minimalne greške koja će zavisiti od koeficijenata težine

i bajasa, potrebno je pratiti kako promene datih parametara utiču na veličinu greške. To je razlog zašto je potrebno naći parcijalne izvode greške po ovim parametrima, tj.

potrebno je naći $\frac{\partial CE}{\partial \omega_1}, \frac{\partial CE}{\partial \omega_2}, \frac{\partial CE}{\partial b_1}, \frac{\partial CE}{\partial b_2}$.

$$\frac{\partial CE}{\partial \omega_j} = -\frac{1}{n} \sum_x \left(\left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) x_j \sigma'(z) \right) \quad (37)$$

Uprošćavanjem gornje jednačine dobija se:

$$\frac{\partial CE}{\partial \omega_j} = \frac{1}{n} \sum_x \left(\frac{(\sigma(z)-y)}{\sigma(z)(1-\sigma(z))} x_j \sigma'(z) \right) \quad (38)$$

Daljim uprošćavanjem dobijaju se jednačine promene funkcije unakrsne-entropije po koeficijentima težine i bajasu:

$$\frac{\partial CE}{\partial \omega_j} = \frac{1}{n} \sum_x \left(x_j (\sigma(z) - y) \right) \quad (39)$$

$$\frac{\partial CE}{\partial b} = \frac{1}{n} \sum_x (\sigma(z) - y) \quad (40)$$

Sledi kod.

```
import matplotlib.pyplot as plt
import numpy as np
import sklearn
import sklearn.datasets
import sklearn.linear_model
import matplotlib as mpl
np.random.seed(None)
X, y = sklearn.datasets.make_moons(250, noise=0.25)
def granica(podela):
    x_min, x_max = X[:, 0].min() - .9, X[:, 0].max() + .9
    y_min, y_max = X[:, 1].min() - .9, X[:, 1].max() + .9
    d = 0.01
    xx, yy = np.meshgrid(np.arange(x_min, x_max, d), np.arange(y_min, y_max, d))
    Novi_objekat = podela(np.c_[xx.ravel(), yy.ravel()])
    Novi_objekat = Novi_objekat.reshape(xx.shape)
```

```

plt.contourf(xx, yy, Novi_objekat, cmap='viridis')
plt.scatter(X[:,0], X[:,1], marker='o', c=y, s=30, cmap='viridis',
edgecolors='#000000')
# Odredjivanje velicine skupa za trening i dimenzija ulaznog i izlaznog sloja.
vel_skupa_za_trening = len(X)
vel_ulaznog_sloja = 2
vel_izlaznog_sloja = 2
# Zadavanje parametra koji predstavlja koeficijant silaznog nagiba
nagib = 0.01
# Da bi se izbegao tzv. proces overfitting kada dolazi do pojave greske koristi se postupak
# regularizacije. Overfitting je vezan za modele koji previse dobro treniraju mrezu.
# Overfitting se javlja kada je model mreze previse dobro naucio da prepoznaje ulazne
# podatke i sum koji se koriste u procesu treniranja, jer to moze dovesti do toga da novi
# nepoznati podaci koji stizu nece biti razvrstani na kvalitetan nacin.
# Suprotno overfitting procesu postoji proces underfitting koji se javlja kod mreza
# jednostavne strukture kada model, iz razloga jednoatavnosti, ne proucava mrezu na
# odgovarajuci nacin, tj. dolazi do "podcenjivanja" mreze usled njene jednostavnosti.
# Ova dva procesa su procesi koji su glavni "krivci" za pojavljivanje velikih odstupnja
# dobijenih vrednosti od zeljenih vrednosti. Regularizacija je postupak kada se mreza
# vestacki i prisilno uproscava, a koristi se kod overfitting-a.
koef_regularizacije = 0.01
# Potrebno je definisati funkciju koja dati sumu svih gubitaka koji se javlja nad
# posmatranim skupom podataka. Definisu se koef. tezine i bajas.
def ukupan_gubitak(model):
    w1, b1, w2, b2 = model['w1'], model['b1'], model['w2'], model['b2']
    # Koriste se formule definisane u radu, a koje predstavljaju model fidforward mreze.
    # Funkcija dot() prihvata dva niza koji mogu biti jednodimenzionalni ili
    dvodimenzionalni.
    # Proizvod u slucaju je 1D nizova je vektor, dok kod 2D nizova je u pitanju mnozenje
    # matrica.
    z1 = X.dot(w1) + b1
    a1 = np.tanh(z1)

```

```

z2 = a1.dot(w2) + b2
# Poziva se eksponencijalna funkcija koja pripada numpy paketu
# Funkcija sum() koja je deo numpy paketa sabira sve elemente niza
# Parametar axis oznacava po kojoj osi ce vrednosti niza biti sabirane. Kako je u kodu
# stavljeno da je axis=1 to znaci da ce po prvoj osi biti sabirane vrednosti. Kada je
axis=1
# to znaci da ce sabiranje biti izvršeno po kolonama matrice, dok kada je axis=0
sabiranje
# ce biti izvršeno po redovima matrice.
# Parametar keepdims kome je dodeljena vrednost True znaci da ce izlazna vrednost
imati
# isti broj dimenzija kao i ulazna vrednost. Funkcija sum() uzima multidimenzionalni
# objekat i sabira vrednosti tog objekta. To sabiranje moze dati skalarni rezultat
ukoliko
# se saboraju sve vrednosti tog objekta, tj. sve dimenzije multidimenzionog objekta se
# sumiraju u jednu dimenziju, tj. dolazi do kolapsa multidimenzionog objekta.
ocekivani_rezultat = np.exp(z2)
uzorci = ocekivani_rezultat / np.sum(ocekivani_rezultat, axis=1, keepdims=True)
# Sledi izracunavanje gubitka primenom log() funkcije numpy paketa. Ova funkcija
# u Pajtonu predstavlja prirodan logaritam koji je inverzna funkcija eksponencijalnoj
# funkciji.
log_vrednosti = -np.log(uzorci[range(vel_skupa_za_trening), y])
gubitak = np.sum(log_vrednosti)
# Dodavanje regularizacije. Ovo je opcioni dodatak. Bice prikazana razlika kada se
# upotrebljava koef. regularizacije, a kada ne.
gubitak += koef_regularizacije/2 * (np.sum(np.square(w1)) +
np.sum(np.square(w2)))
return 1./vel_skupa_za_trening * gubitak
# Funkcija koja sledi, a koja je nazvana predvidanje, omogucava predvidjanje izlaza (0
ili 1).
def predvidjanje(model, x):
    w1, b1, w2, b2 = model['w1'], model['b1'], model['w2'], model['b2']

```

Sledi ponovna primena fidforward modela. Komentar je isti kao i kod linija koda 58-68.

Funkcija argmax() vraca najveću vrednost dobijenog niza uzorci.

z1 = x.dot(w1) + b1

a1 = np.tanh(z1)

z2 = a1.dot(w2) + b2

ocekivani_rezultat = np.exp(z2)

uzorci = ocekivani_rezultat / np.sum(ocekivani_rezultat, axis=1, keepdims=True)

return np.argmax(uzorci, axis=1)

Definise se funkcija model_za_ucenje koja ce definisati potrebne parametre za ucenje mreze. U okviru date funkcije ce biti definisani sledeci parametri:

bn - broj neurona u skrivenom sloju

bp - broj prolaza kroz skup podataka za trening u procesu ucenja kada je u pitanju

model gubitaka sa nagibom nadole.

stampaj_gubitke - Ako je vrednost ovog parametra True tada ce biti stampan gubitak # nakon svakih hiljadu iteracija.

def model_za_ucenje(bn, bp=50000, stampaj_gubitke=False):

Pocetni parametri se postavljau na slucajne vrednosti.

np.random.seed(None)

w1 = np.random.randn(vel_ulaznog_sloja, bn) / np.sqrt(vel_ulaznog_sloja)

b1 = np.zeros((1, bn))

w2 = np.random.randn(bn, vel_izlaznog_sloja) / np.sqrt(bn)

b2 = np.zeros((1, vel_izlaznog_sloja))

Vracanje parametra model na kraju treninga.

model = {}

Ucenje sa nagibom nadole

for i in range(0, bp):

Fidvorad model

z1 = X.dot(w1) + b1

a1 = np.tanh(z1)

z2 = a1.dot(w2) + b2

ocekivani_rezultat = np.exp(z2)

```

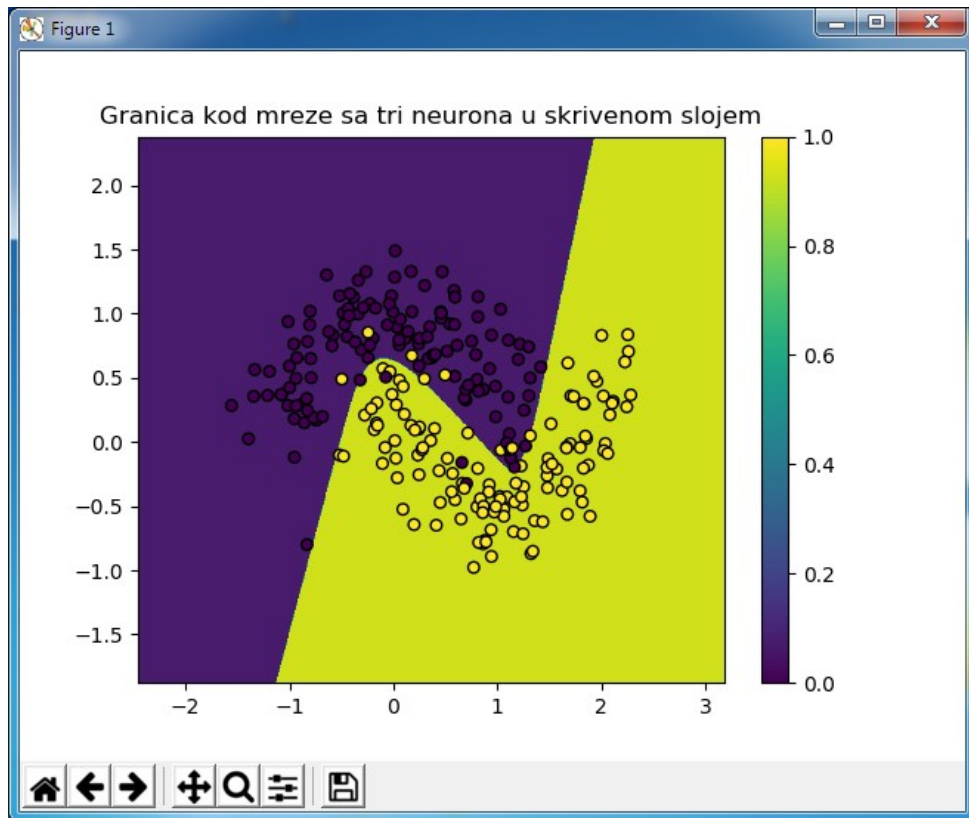
uzorci = ocekivani_rezultat / np.sum(ocekivani_rezultat, axis=1, keepdims=True)
# Propagacija sa sirenjem unazad (backpropagation model)
D3 = uzorci
D3[range(vel_skupa_za_trening), y] -= 1
izvodw2 = (a1.T).dot(D3)
izvodb2 = np.sum(D3, axis=0, keepdims=True)
D2 = D3.dot(w2.T) * (1 - np.power(a1, 2))
izvodw1 = np.dot(X.T, D2)
izvodb1 = np.sum(D2, axis=0)
# Dodavanje koef regularizacije za koef, tezine w1 i w2.
izvodw2 += koef_regularizacije * w2
izvodw1 += koef_regularizacije * w1
# Obnavljanje parametra nagiba nadole.
w1 += -nagib * izvodw1
b1 += -nagib * izvodb1
w2 += -nagib * izvodw2
b2 += -nagib * izvodb2
# Dodeljivanje novih parametara modelu
model = { 'w1': w1, 'b1': b1, 'w2': w2, 'b2': b2}
# Stampanje gubitaka koje je opciono, jer zahteva prolazak kroz ceo skup podataka
# i trosenje resursa memorije.
if stampaj_gubitke and i % 500 == 0:
    print("Gubitak posle iteracije je %i: %f" %(i, ukupan_gubitak(model)))
return model
# Kreiranje modela sa mrezom koja ima tri neurona u skrivenom sloju.
model = model_za_ucenje(3, stampaj_gubitke=True)
# Crtanje granice podele
granica(lambda x: predvidjanje(model, x))
plt.title("Granica kod mreze sa tri neurona u skrivenom slojem")
plt.colorbar()
plt.show()

```

```
Command Prompt - python_Ceo_trening_tri_neurona_skriveni_sloj.py
C:\Users\Nico\AppData\Local\Programs\Python\Python37>python_Ceo_trening_tri_neurona_skriveni_sloj.py
Gubitak po le iteracije je 0: 0.404531
Gubitak po le iteracije je 500: 0.143546
Gubitak po le iteracije je 1000: 0.141532
Gubitak po le iteracije je 1500: 0.141072
Gubitak po le iteracije je 2000: 0.140818
Gubitak po le iteracije je 2500: 0.140647
Gubitak po le iteracije je 3000: 0.140523
Gubitak po le iteracije je 3500: 0.140428
Gubitak po le iteracije je 4000: 0.140351
Gubitak po le iteracije je 4500: 0.140287
Gubitak po le iteracije je 5000: 0.140231
Gubitak po le iteracije je 5500: 0.140183
Gubitak po le iteracije je 6000: 0.140146
Gubitak po le iteracije je 6500: 0.140101
Gubitak po le iteracije je 7000: 0.140065
Gubitak po le iteracije je 7500: 0.140032
Gubitak po le iteracije je 8000: 0.140002
Gubitak po le iteracije je 8500: 0.139973
Gubitak po le iteracije je 9000: 0.139946
Gubitak po le iteracije je 9500: 0.139921
Gubitak po le iteracije je 10000: 0.139877
Gubitak po le iteracije je 10500: 0.139874
Gubitak po le iteracije je 11000: 0.139852
Gubitak po le iteracije je 11500: 0.139831
Gubitak po le iteracije je 12000: 0.139812
Gubitak po le iteracije je 12500: 0.139793
Gubitak po le iteracije je 13000: 0.139775
Gubitak po le iteracije je 13500: 0.139757
Gubitak po le iteracije je 14000: 0.139741
Gubitak po le iteracije je 14500: 0.139725
Gubitak po le iteracije je 15000: 0.139709
Gubitak po le iteracije je 15500: 0.139694
Gubitak po le iteracije je 16000: 0.139680
Gubitak po le iteracije je 16500: 0.139667
Gubitak po le iteracije je 17000: 0.139653
Gubitak po le iteracije je 17500: 0.139641
Gubitak po le iteracije je 18000: 0.139629
Gubitak po le iteracije je 18500: 0.139617
Gubitak po le iteracije je 19000: 0.139606
Gubitak po le iteracije je 19500: 0.139595
Gubitak po le iteracije je 20000: 0.139585
Gubitak po le iteracije je 20500: 0.139575
Gubitak po le iteracije je 21000: 0.139565
Gubitak po le iteracije je 21500: 0.139556
Gubitak po le iteracije je 22000: 0.139547
Gubitak po le iteracije je 22500: 0.139538
Gubitak po le iteracije je 23000: 0.139530
Gubitak po le iteracije je 23500: 0.139522
Gubitak po le iteracije je 24000: 0.139514
Gubitak po le iteracije je 24500: 0.139507

Command Prompt - python_Ceo_trening_tri_neurona_skriveni_sloj.py
Gubitak po le iteracije je 24500: 0.139507
Gubitak po le iteracije je 25000: 0.139500
Gubitak po le iteracije je 25500: 0.139493
Gubitak po le iteracije je 26000: 0.139486
Gubitak po le iteracije je 26500: 0.139480
Gubitak po le iteracije je 27000: 0.139474
Gubitak po le iteracije je 27500: 0.139468
Gubitak po le iteracije je 28000: 0.139462
Gubitak po le iteracije je 28500: 0.139457
Gubitak po le iteracije je 29000: 0.139451
Gubitak po le iteracije je 29500: 0.139446
Gubitak po le iteracije je 30000: 0.139441
Gubitak po le iteracije je 30500: 0.139437
Gubitak po le iteracije je 31000: 0.139432
Gubitak po le iteracije je 31500: 0.139428
Gubitak po le iteracije je 32000: 0.139424
Gubitak po le iteracije je 32500: 0.139420
Gubitak po le iteracije je 33000: 0.139416
Gubitak po le iteracije je 33500: 0.139412
Gubitak po le iteracije je 34000: 0.139408
Gubitak po le iteracije je 34500: 0.139405
Gubitak po le iteracije je 35000: 0.139401
Gubitak po le iteracije je 35500: 0.139398
Gubitak po le iteracije je 36000: 0.139395
Gubitak po le iteracije je 36500: 0.139392
Gubitak po le iteracije je 37000: 0.139389
Gubitak po le iteracije je 37500: 0.139386
Gubitak po le iteracije je 38000: 0.139383
Gubitak po le iteracije je 38500: 0.139381
Gubitak po le iteracije je 39000: 0.139378
Gubitak po le iteracije je 39500: 0.139376
Gubitak po le iteracije je 40000: 0.139373
Gubitak po le iteracije je 40500: 0.139371
Gubitak po le iteracije je 41000: 0.139369
Gubitak po le iteracije je 41500: 0.139367
Gubitak po le iteracije je 42000: 0.139365
Gubitak po le iteracije je 42500: 0.139363
Gubitak po le iteracije je 43000: 0.139361
Gubitak po le iteracije je 43500: 0.139359
Gubitak po le iteracije je 44000: 0.139357
Gubitak po le iteracije je 44500: 0.139356
Gubitak po le iteracije je 45000: 0.139354
Gubitak po le iteracije je 45500: 0.139352
Gubitak po le iteracije je 46000: 0.139351
Gubitak po le iteracije je 46500: 0.139349
Gubitak po le iteracije je 47000: 0.139348
Gubitak po le iteracije je 47500: 0.139346
Gubitak po le iteracije je 48000: 0.139345
Gubitak po le iteracije je 48500: 0.139344
Gubitak po le iteracije je 49000: 0.139342
Gubitak po le iteracije je 49500: 0.139341
```

Slika 5.7.1.12 Učenje neuronske mreže sa tri neurona u skrivenom sloju-kod pokrenut preko CMD



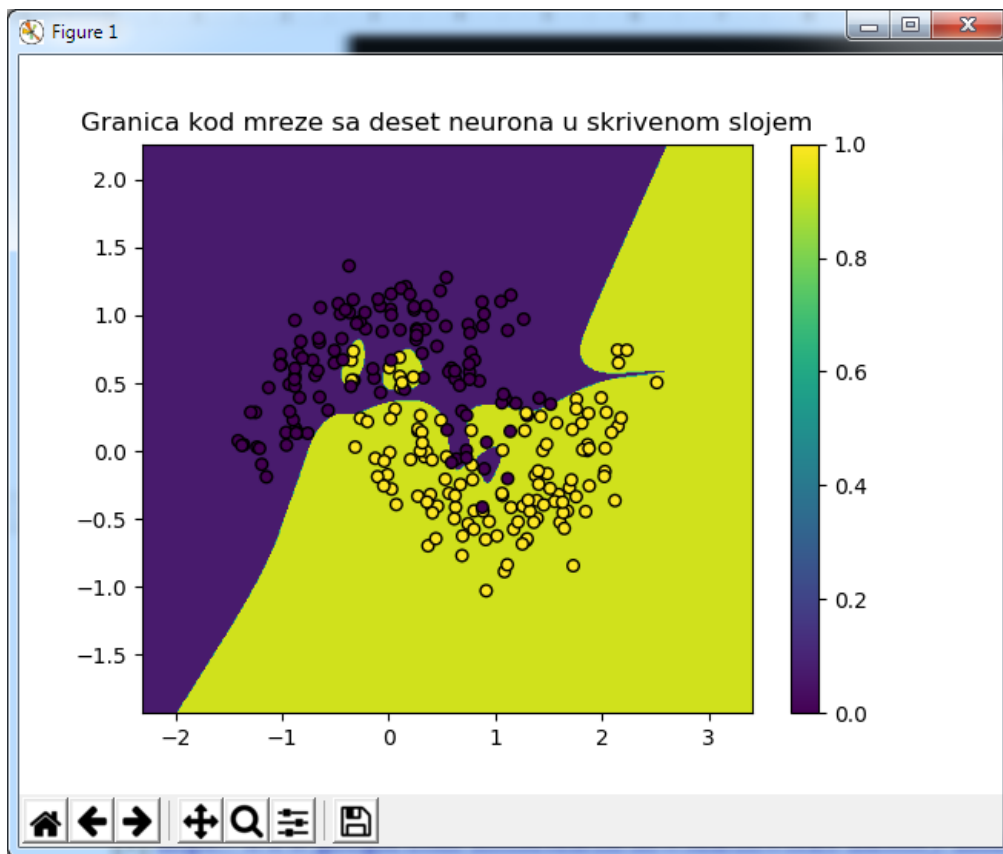
Slika 5.7.1.13 Dijagram učenja neuronske mreže sa tri neurona u skrivenom sloju

U skrivenom sloju može biti stavljen različit broj neurona. Slede slike kada je u skrivenom sloju deset neurona i kada je u skrivenom sloju sto neurona. Nakon slika će biti data kratka diskusija o dobijenim rezultatima.


```
Command Prompt - python Ceo_trening_deset_neurona_skriveni_sloj.py
C:\Users\Ivica\AppData\Local\Programs\Python\Python37>python Ceo_trening_deset_neurona_skriveni_sloj.py
Gubitak posle iteracije je 0: 0.503467
Gubitak posle iteracije je 500: 0.154638
Gubitak posle iteracije je 1000: 0.148930
Gubitak posle iteracije je 1500: 0.147940
Gubitak posle iteracije je 2000: 0.143073
Gubitak posle iteracije je 2500: 0.140314
Gubitak posle iteracije je 3000: 0.132270
Gubitak posle iteracije je 3500: 0.134087
Gubitak posle iteracije je 4000: 0.132736
Gubitak posle iteracije je 4500: 0.129210
Gubitak posle iteracije je 5000: 0.126319
Gubitak posle iteracije je 5500: 0.125691
Gubitak posle iteracije je 6000: 0.123153
Gubitak posle iteracije je 6500: 0.120992
Gubitak posle iteracije je 7000: 0.119989
Gubitak posle iteracije je 7500: 0.117418
Gubitak posle iteracije je 8000: 0.115866
Gubitak posle iteracije je 8500: 0.115377
Gubitak posle iteracije je 9000: 0.115162
Gubitak posle iteracije je 9500: 0.113770
Gubitak posle iteracije je 10000: 0.111324
Gubitak posle iteracije je 10500: 0.109693
Gubitak posle iteracije je 11000: 0.108399
Gubitak posle iteracije je 11500: 0.107279
Gubitak posle iteracije je 12000: 0.106310
Gubitak posle iteracije je 12500: 0.105506
Gubitak posle iteracije je 13000: 0.104901
Gubitak posle iteracije je 13500: 0.104450
Gubitak posle iteracije je 14000: 0.104030
Gubitak posle iteracije je 14500: 0.103611
Gubitak posle iteracije je 15000: 0.103170
Gubitak posle iteracije je 15500: 0.102690
Gubitak posle iteracije je 16000: 0.102114
Gubitak posle iteracije je 16500: 0.101368
Gubitak posle iteracije je 17000: 0.100635
Gubitak posle iteracije je 17500: 0.099940
Gubitak posle iteracije je 18000: 0.099238
Gubitak posle iteracije je 18500: 0.0987320
Gubitak posle iteracije je 19000: 0.0986661
Gubitak posle iteracije je 19500: 0.098145
Gubitak posle iteracije je 20000: 0.098717
Gubitak posle iteracije je 20500: 0.0985359
Gubitak posle iteracije je 21000: 0.09851
Gubitak posle iteracije je 21500: 0.094914
Gubitak posle iteracije je 22000: 0.094606
Gubitak posle iteracije je 22500: 0.094431
Gubitak posle iteracije je 23000: 0.094282
Gubitak posle iteracije je 23500: 0.094154
Gubitak posle iteracije je 24000: 0.094045
Gubitak posle iteracije je 24500: 0.093951
Gubitak posle iteracije je 25000: 0.093870
Gubitak posle iteracije je 25500: 0.093800
Gubitak posle iteracije je 26000: 0.093740

Gubitak posle iteracije je 26000: 0.093740
Gubitak posle iteracije je 26500: 0.093688
Gubitak posle iteracije je 27000: 0.093643
Gubitak posle iteracije je 27500: 0.093605
Gubitak posle iteracije je 28000: 0.093572
Gubitak posle iteracije je 28500: 0.093544
Gubitak posle iteracije je 29000: 0.093519
Gubitak posle iteracije je 29500: 0.093499
Gubitak posle iteracije je 30000: 0.093481
Gubitak posle iteracije je 30500: 0.093466
Gubitak posle iteracije je 31000: 0.093454
Gubitak posle iteracije je 31500: 0.093443
Gubitak posle iteracije je 32000: 0.093434
Gubitak posle iteracije je 32500: 0.093427
Gubitak posle iteracije je 33000: 0.093421
Gubitak posle iteracije je 33500: 0.093416
Gubitak posle iteracije je 34000: 0.093413
Gubitak posle iteracije je 34500: 0.093410
Gubitak posle iteracije je 35000: 0.093407
Gubitak posle iteracije je 35500: 0.093405
Gubitak posle iteracije je 36000: 0.093404
Gubitak posle iteracije je 36500: 0.093404
Gubitak posle iteracije je 37000: 0.093403
Gubitak posle iteracije je 37500: 0.093403
Gubitak posle iteracije je 38000: 0.093403
Gubitak posle iteracije je 38500: 0.093404
Gubitak posle iteracije je 39000: 0.093405
Gubitak posle iteracije je 39500: 0.093405
Gubitak posle iteracije je 40000: 0.093406
Gubitak posle iteracije je 40500: 0.093407
Gubitak posle iteracije je 41000: 0.093409
Gubitak posle iteracije je 41500: 0.093410
Gubitak posle iteracije je 42000: 0.093411
Gubitak posle iteracije je 42500: 0.093412
Gubitak posle iteracije je 43000: 0.093414
Gubitak posle iteracije je 43500: 0.093415
Gubitak posle iteracije je 44000: 0.093417
Gubitak posle iteracije je 44500: 0.093418
Gubitak posle iteracije je 45000: 0.093419
Gubitak posle iteracije je 45500: 0.093421
Gubitak posle iteracije je 46000: 0.093422
Gubitak posle iteracije je 46500: 0.093424
Gubitak posle iteracije je 47000: 0.093425
Gubitak posle iteracije je 47500: 0.093426
Gubitak posle iteracije je 48000: 0.093428
Gubitak posle iteracije je 48500: 0.093429
Gubitak posle iteracije je 49000: 0.093430
Gubitak posle iteracije je 49500: 0.093431
```

Slika 5.7.1.14. Učenje neuronske mreže sa deset neurona u skrivenom sloju-kod pokrenut preko CMD šela

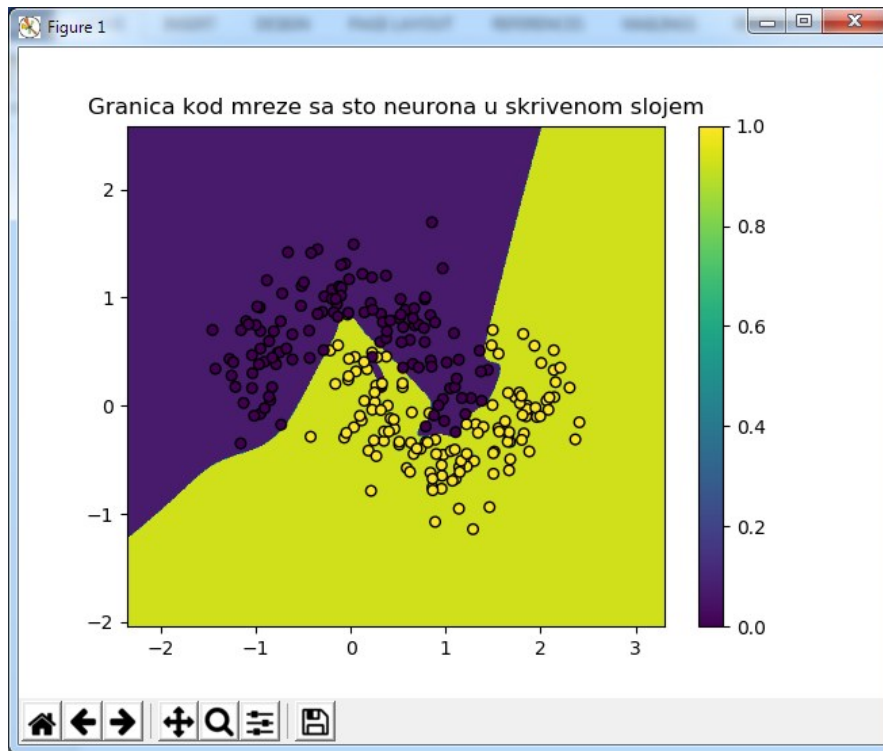


Slika 5.7.1.15. Dijagram klasifikacije uzoraka neuronske mreže sa deset neurona u skrivenom sloju-kod pokrenut preko CMD šela

```
Command Prompt - python Ceo_trening_sto_neurona_skriveni_sloj.py
C:\Users\Ivica\AppData\Local\Programs\Python\Python37>python Ceo_trening_sto_neurona_skriveni_sloj.py
Gubitak posle iteracije je 0: 4.514365
Gubitak posle iteracije je 500: 0.130333
Gubitak posle iteracije je 1000: 0.090841
Gubitak posle iteracije je 1500: 0.087415
Gubitak posle iteracije je 2000: 0.144035
Gubitak posle iteracije je 2500: 0.073389
Gubitak posle iteracije je 3000: 0.064784
Gubitak posle iteracije je 3500: 0.064239
Gubitak posle iteracije je 4000: 0.061436
Gubitak posle iteracije je 4500: 0.059564
Gubitak posle iteracije je 5000: 0.058312
Gubitak posle iteracije je 5500: 0.057451
Gubitak posle iteracije je 6000: 0.056821
Gubitak posle iteracije je 6500: 0.056324
Gubitak posle iteracije je 7000: 0.055893
Gubitak posle iteracije je 7500: 0.055479
Gubitak posle iteracije je 8000: 0.055047
Gubitak posle iteracije je 8500: 0.054571
Gubitak posle iteracije je 9000: 0.054126
Gubitak posle iteracije je 9500: 0.053678
Gubitak posle iteracije je 10000: 0.053270
Gubitak posle iteracije je 10500: 0.052919
Gubitak posle iteracije je 11000: 0.052630
Gubitak posle iteracije je 11500: 0.052402
Gubitak posle iteracije je 12000: 0.052220
Gubitak posle iteracije je 12500: 0.052061
Gubitak posle iteracije je 13000: 0.051906
Gubitak posle iteracije je 13500: 0.051745
Gubitak posle iteracije je 14000: 0.051584
Gubitak posle iteracije je 14500: 0.051438
Gubitak posle iteracije je 15000: 0.051315
Gubitak posle iteracije je 15500: 0.051207
Gubitak posle iteracije je 16000: 0.051097
Gubitak posle iteracije je 16500: 0.050970
Gubitak posle iteracije je 17000: 0.050834
Gubitak posle iteracije je 17500: 0.050692
Gubitak posle iteracije je 18000: 0.050546
Gubitak posle iteracije je 18500: 0.050395
Gubitak posle iteracije je 19000: 0.050235
Gubitak posle iteracije je 19500: 0.050064
Gubitak posle iteracije je 20000: 0.049878
Gubitak posle iteracije je 20500: 0.049686
Gubitak posle iteracije je 21000: 0.049493
Gubitak posle iteracije je 21500: 0.049310
Gubitak posle iteracije je 22000: 0.049147
Gubitak posle iteracije je 22500: 0.048981
Gubitak posle iteracije je 23000: 0.048852
Gubitak posle iteracije je 23500: 0.048877
Gubitak posle iteracije je 24000: 0.048810
Gubitak posle iteracije je 24500: 0.048758
Gubitak posle iteracije je 25000: 0.048717
Gubitak posle iteracije je 25500: 0.048683
Gubitak posle iteracije je 26000: 0.048656

Command Prompt - python Ceo_trening_sto_neurona_skriveni_sloj.py
Gubitak posle iteracije je 26000: 0.048656
Gubitak posle iteracije je 26500: 0.048634
Gubitak posle iteracije je 27000: 0.048612
Gubitak posle iteracije je 27500: 0.048602
Gubitak posle iteracije je 28000: 0.048590
Gubitak posle iteracije je 28500: 0.048580
Gubitak posle iteracije je 29000: 0.048573
Gubitak posle iteracije je 29500: 0.048566
Gubitak posle iteracije je 30000: 0.048561
Gubitak posle iteracije je 30500: 0.048556
Gubitak posle iteracije je 31000: 0.048552
Gubitak posle iteracije je 31500: 0.048549
Gubitak posle iteracije je 32000: 0.048545
Gubitak posle iteracije je 32500: 0.048542
Gubitak posle iteracije je 33000: 0.048539
Gubitak posle iteracije je 33500: 0.048536
Gubitak posle iteracije je 34000: 0.048534
Gubitak posle iteracije je 34500: 0.048531
Gubitak posle iteracije je 35000: 0.048528
Gubitak posle iteracije je 35500: 0.048526
Gubitak posle iteracije je 36000: 0.048523
Gubitak posle iteracije je 36500: 0.048521
Gubitak posle iteracije je 37000: 0.048520
Gubitak posle iteracije je 37500: 0.048518
Gubitak posle iteracije je 38000: 0.048517
Gubitak posle iteracije je 38500: 0.048516
Gubitak posle iteracije je 39000: 0.048515
Gubitak posle iteracije je 39500: 0.048515
Gubitak posle iteracije je 40000: 0.048514
Gubitak posle iteracije je 40500: 0.048514
Gubitak posle iteracije je 41000: 0.048513
Gubitak posle iteracije je 41500: 0.048513
Gubitak posle iteracije je 42000: 0.048513
Gubitak posle iteracije je 42500: 0.048512
Gubitak posle iteracije je 43000: 0.048512
Gubitak posle iteracije je 43500: 0.048511
Gubitak posle iteracije je 44000: 0.048511
Gubitak posle iteracije je 44500: 0.048510
Gubitak posle iteracije je 45000: 0.048509
Gubitak posle iteracije je 45500: 0.048508
Gubitak posle iteracije je 46000: 0.048507
Gubitak posle iteracije je 46500: 0.048506
Gubitak posle iteracije je 47000: 0.048505
Gubitak posle iteracije je 47500: 0.048504
Gubitak posle iteracije je 48000: 0.048502
Gubitak posle iteracije je 48500: 0.048501
Gubitak posle iteracije je 49000: 0.048499
Gubitak posle iteracije je 49500: 0.048498
```

Slika 5.7.1.16. Učenje neuronske mreže sa sto neurona u skrivenom sloju-kod pokrenut preko CMD šela

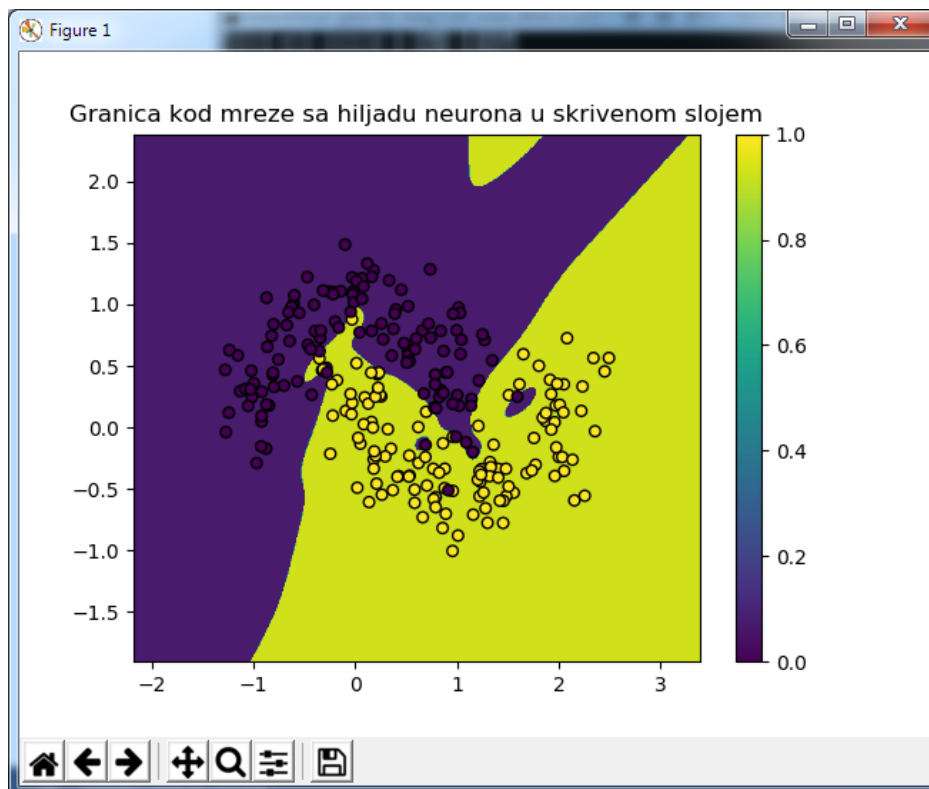


Slika 5.7.1.17. Dijagram klasifikacije uzoraka neuronske mreže sa sto neurona u skrivenom sloju-kod pokrenut preko CMD šela

```
Command Prompt - python Ceo_trening_hiljadu_neurona_skriveni_sloj.py
C:\Users\Ivica\AppData\Local\Programs\Python\Python37\python Ceo_trening_hiljadu_neurona_skriveni_sloj.py
Gubitak posle iteracije je 0: 26.67238
Gubitak posle iteracije je 500: 0.229669
Gubitak posle iteracije je 1000: 0.137326
Gubitak posle iteracije je 1500: 0.126101
Gubitak posle iteracije je 2000: 0.120730
Gubitak posle iteracije je 2500: 0.116381
Gubitak posle iteracije je 3000: 0.111837
Gubitak posle iteracije je 3500: 0.108837
Gubitak posle iteracije je 4000: 0.106798
Gubitak posle iteracije je 4500: 0.105021
Gubitak posle iteracije je 5000: 0.103565
Gubitak posle iteracije je 5500: 0.102238
Gubitak posle iteracije je 6000: 0.100825
Gubitak posle iteracije je 6500: 0.099346
Gubitak posle iteracije je 7000: 0.097837
Gubitak posle iteracije je 7500: 0.096396
Gubitak posle iteracije je 8000: 0.095136
Gubitak posle iteracije je 8500: 0.094073
Gubitak posle iteracije je 9000: 0.093154
Gubitak posle iteracije je 9500: 0.092348
Gubitak posle iteracije je 10000: 0.091638
Gubitak posle iteracije je 10500: 0.090957
Gubitak posle iteracije je 11000: 0.090251
Gubitak posle iteracije je 11500: 0.089638
Gubitak posle iteracije je 12000: 0.089328
Gubitak posle iteracije je 12500: 0.089299
Gubitak posle iteracije je 13000: 0.089456
Gubitak posle iteracije je 13500: 0.089573
Gubitak posle iteracije je 14000: 0.089958
Gubitak posle iteracije je 14500: 0.084229
Gubitak posle iteracije je 15000: 0.083506
Gubitak posle iteracije je 15500: 0.082809
Gubitak posle iteracije je 16000: 0.082169
Gubitak posle iteracije je 16500: 0.081623
Gubitak posle iteracije je 17000: 0.081186
Gubitak posle iteracije je 17500: 0.080803
Gubitak posle iteracije je 18000: 0.080398
Gubitak posle iteracije je 18500: 0.079946
Gubitak posle iteracije je 19000: 0.079472
Gubitak posle iteracije je 19500: 0.078996
Gubitak posle iteracije je 20000: 0.078372
Gubitak posle iteracije je 20500: 0.077224
Gubitak posle iteracije je 21000: 0.0774496
Gubitak posle iteracije je 21500: 0.076085
Gubitak posle iteracije je 22000: 0.073727
Gubitak posle iteracije je 22500: 0.072534
Gubitak posle iteracije je 23000: 0.071598
Gubitak posle iteracije je 23500: 0.070840
Gubitak posle iteracije je 24000: 0.070216
Gubitak posle iteracije je 24500: 0.069699
Gubitak posle iteracije je 25000: 0.069269
Gubitak posle iteracije je 25500: 0.068910

Gubitak posle iteracije je 26000: 0.068608
Gubitak posle iteracije je 26500: 0.068354
Gubitak posle iteracije je 27000: 0.068138
Gubitak posle iteracije je 27500: 0.067953
Gubitak posle iteracije je 28000: 0.067773
Gubitak posle iteracije je 28500: 0.067654
Gubitak posle iteracije je 29000: 0.067532
Gubitak posle iteracije je 29500: 0.067424
Gubitak posle iteracije je 30000: 0.067328
Gubitak posle iteracije je 30500: 0.067242
Gubitak posle iteracije je 31000: 0.067165
Gubitak posle iteracije je 31500: 0.067095
Gubitak posle iteracije je 32000: 0.067033
Gubitak posle iteracije je 32500: 0.066977
Gubitak posle iteracije je 33000: 0.066926
Gubitak posle iteracije je 33500: 0.066881
Gubitak posle iteracije je 34000: 0.066842
Gubitak posle iteracije je 34500: 0.066806
Gubitak posle iteracije je 35000: 0.066774
Gubitak posle iteracije je 35500: 0.066746
Gubitak posle iteracije je 36000: 0.066721
Gubitak posle iteracije je 36500: 0.066697
Gubitak posle iteracije je 37000: 0.066676
Gubitak posle iteracije je 37500: 0.066656
Gubitak posle iteracije je 38000: 0.066638
Gubitak posle iteracije je 38500: 0.066621
Gubitak posle iteracije je 39000: 0.066604
Gubitak posle iteracije je 39500: 0.066589
Gubitak posle iteracije je 40000: 0.066574
Gubitak posle iteracije je 40500: 0.066560
Gubitak posle iteracije je 41000: 0.066547
Gubitak posle iteracije je 41500: 0.066534
Gubitak posle iteracije je 42000: 0.066521
Gubitak posle iteracije je 42500: 0.066509
Gubitak posle iteracije je 43000: 0.066497
Gubitak posle iteracije je 43500: 0.066484
Gubitak posle iteracije je 44000: 0.066472
Gubitak posle iteracije je 44500: 0.066459
Gubitak posle iteracije je 45000: 0.066446
Gubitak posle iteracije je 45500: 0.066433
Gubitak posle iteracije je 46000: 0.066418
Gubitak posle iteracije je 46500: 0.066404
Gubitak posle iteracije je 47000: 0.066388
Gubitak posle iteracije je 47500: 0.066372
Gubitak posle iteracije je 48000: 0.066355
Gubitak posle iteracije je 48500: 0.066338
Gubitak posle iteracije je 49000: 0.066320
Gubitak posle iteracije je 49500: 0.066303
```

Slika 5.7.1.18. Učenje neuronske mreže sa hiljadu neurona u skrivenom sloju-kod pokrenut preko CMD šela



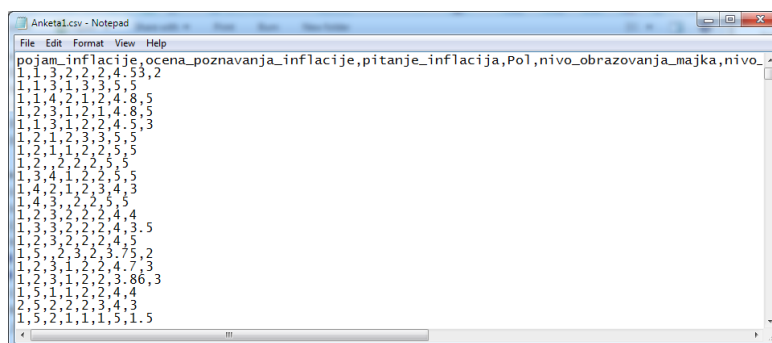
Slika 5.7.1.19. Dijagram klasifikacije uzoraka neuronske mreže sa hiljadu neurona u skrivenom sloju-kod pokrenut preko CMD šela

Proces iteracije traje duže što je veći broj neurona u skrivenom sloju. Najduže vreme iteracije je, u gore razmatrnim primerima, u slučaju kada skriveni sloj ima hiljadu neurona. Vreme je u tom slučaju iznosilo 25[min] i 30[s]. Najkraće vreme odigravanja procesa je bilo u slučaju neuronske mreže sa tri neurona u skrivenom sloju i iznosilo je svega 22 [s]. Komparativnom analizom između pokrenutih kodova vidi se da u slučaju neuronske mreže sa hiljadu neurona u prvom koraku koraku iteracije najveća greška. Kako se iteracija odigrava u istom broju koraka za sve urađene primere, greška iteracije se smanjuje i najmanja greška u poslednjem koraku se javlja kod neuronske mreže sa hiljadu neurona.

6 Eksperimentalni deo rada

6.1 Uvod

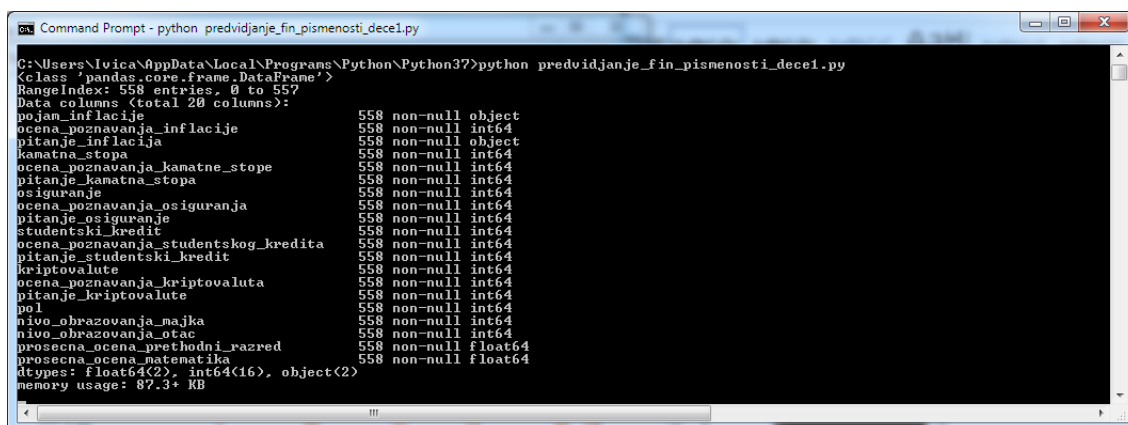
U ovom delu rada će biti data analiza prikupljenih podatka pomoću originalnog koda kreiranog u okviru ove doktorske disertacije. U analizi će biti razmatrane sumirane karakteristike sprovedenog istraživanja. Uzorak ima 558 ispitanika. Anketa koja je rađena je odobrena od strane Ministarstva za nauku, prosvetu i tehnološki razvoj Republike Srbije. U dodatku rada je pozivno pismo Ministarstva za nauku, prosvetu i tehnološki razvoj srednjim školama za učestvovanje u anketi. Anketa se nalazi u dodatku rada. Korišćenje metoda opisne statistike i vizuelizacije će omogućiti prikazivanje podataka u obliku obrazaca i trendova. Cilj eksperimenta je dokazati da veštačke neuronske mreže, tj. veštački inteligentni agenti mogu kvalitetno i u kratkom vremenskom roku odrede složene međuzavisnosti u skupu velike količine podataka. Primenom kreiranih rešenja iz ovog dela rada biće pokazano da upotreba savremenih alata može dovesti do donošenja kvalitetne strategijske odluke, pri čemu se ne gubi vreme na seketovanje i obradu prikupljenih podataka, jer razvijeni kod to sam radi. Kod će biti dat u prilogu rada. Fajl u kome se nalaze prikupljeni podaci je u .csv formatu (comma separated value) i naslovljen je kao Anketa.csv. Format .csv je pogodan, jer omogućava upis tabelarnih podataka kao običan tekst (eng. plaintext), pri čemu su ti podaci odvojeni zarezom (eng. comma separated). U ovom formatu svaki red je predstavljen novom linijom, a svaka kolona je odvojena zarezom. Zahvaljući tome što su podaci upisani kao običan tekst, uvoz (eng. import) ovih podataka u bilo koju bazu ili bilo koji spreadsheet (eng. spreadsheet), bez obzira koji je operativni sistem u pitanju i bez obzira koja je aplikacija u pitanju, je olakšan. Sledi skrin-šot dela rezultata ankete iz Anketa.csv fajla koji je otvoren pomoću notepad tekst editora.



Slika 6.1.1. Anketa1.csv otvoren pomoću notepad tekst editora

6.2 Pokretanje koda

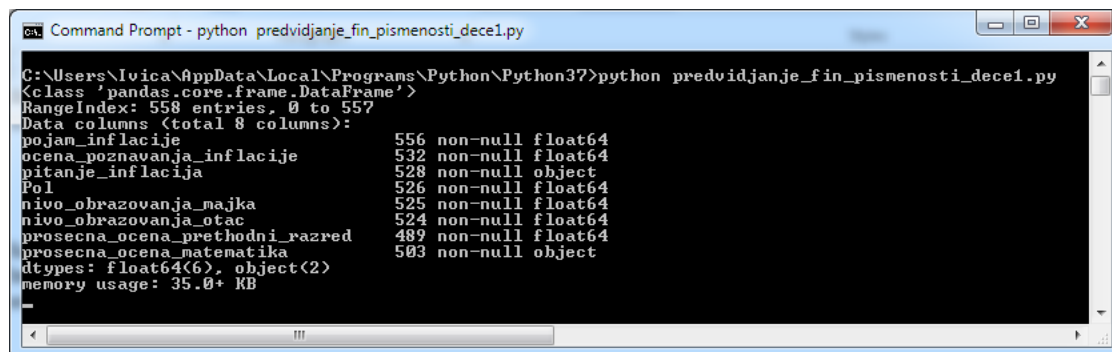
Kod koji je napisan je nazvan `predvidjanje_fin_pismenosti_dece1.py`. Ovaj kod će uraditi potpunu analizu ankete u zavisnosti od zadatih parametara i ujedno će dati određena predviđanja. Kod je pisan u programskom jeziku pajton. Kod je pokrenut preko CMD šela. Sledi slika pokrenutog koda.



```
Command Prompt - python predvidjanje_fin_pismenosti_dece1.py
C:\Users\Ivica\AppData\Local\Programs\Python\Python37>python predvidjanje_fin_pismenosti_dece1.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 558 entries, 0 to 557
Data columns (total 20 columns):
pojam_inflacije          558 non-null object
ocena_poznavanja_inflacije  558 non-null int64
pitanje_inflacija        558 non-null object
kamatna_stopa            558 non-null int64
ocena_poznavanja_kamatne_stope  558 non-null int64
pitanje_kamatna_stopa    558 non-null int64
osiguranje               558 non-null int64
ocena_poznavanja_osiguranja  558 non-null int64
pitanje_osiguranje       558 non-null int64
studentski_kredit        558 non-null int64
ocena_poznavanja_studentskog_kredita  558 non-null int64
pitanje_studentski_kredit  558 non-null int64
kriptovalute            558 non-null int64
ocena_poznavanja_kriptovaluta  558 non-null int64
pitanje_kriptovalute     558 non-null int64
pol                       558 non-null int64
nivo_obrazovanja_majka    558 non-null int64
nivo_obrazovanja_otac     558 non-null int64
prosecna_ocena_prethodni_razred  558 non-null float64
prosecna_ocena_matematika  558 non-null float64
dtypes: float64(2), int64(16), object(2)
memory usage: 87.3+ KB
```

Slika 6.2.1. Pokrenut kod `predvidjanje_fin_pismenosti_dece1.py` gde su sve ćelije popunjene

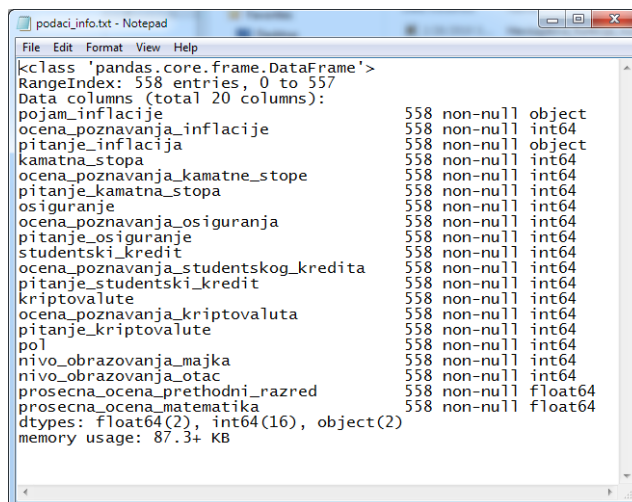
Sa slike 6.1.1. se vidi da je broj ispitanika 558, a da su podaci indeksirani od 0 do 557 (ukupno 558 ispitanika). Ukoliko se u `Anketa.csv` fajlu umesto 0 (pitanje nema odgovor) i broj ponuđenih odgovora +1 (pitanje sa više odgovora) ostave prazne ćelije, pokretanjem koda će se dobiti broj ispitanika koji je dao odgovor na određeno pitanje (not-null). Prazne ćelije se ne računaju u odgovor i odbacuju se. Odatle razlika između ukupnog broja ispitanika (558) i prikazanih brojeva. U notepad tekst editoru prazna mesta su označena sa: , , .



```
Command Prompt - python predvidjanje_fin_pismenosti_dece1.py
C:\Users\Ivica\AppData\Local\Programs\Python\Python37>python predvidjanje_fin_pismenosti_dece1.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 558 entries, 0 to 557
Data columns (total 8 columns):
pojam_inflacije          556 non-null float64
ocena_poznavanja_inflacije  532 non-null float64
pitanje_inflacija        528 non-null object
Pol                       526 non-null float64
nivo_obrazovanja_majka    525 non-null float64
nivo_obrazovanja_otac     524 non-null float64
prosecna_ocena_prethodni_razred  489 non-null float64
prosecna_ocena_matematika  503 non-null object
dtypes: float64(6), object(2)
memory usage: 35.0+ KB
```

Slika 6.2.2. Pokrenut kod `predvidjanje_fin_pismenosti_dece1.py` gde nisu sve ćelije popunjene

Deo koda omogućava da funkcija info() bude prevedena u .txt format čiji skrin-šot sledi. Više o samoj funkciji info() je dato u komentaru koji se nalazi u kodu.



```
podaci_info.txt - Notepad
File Edit Format View Help
kclass 'pandas.core.frame.DataFrame'>
RangeIndex: 558 entries, 0 to 557
Data columns (total 20 columns):
pojam_inflacije          558 non-null object
ocena_poznavanja_inflacije  558 non-null int64
pitanje_inflacija         558 non-null object
amatna_stopa             558 non-null int64
ocena_poznavanja_kamatne_stope  558 non-null int64
pitanje_kamatna_stopa    558 non-null int64
osiguranje               558 non-null int64
ocena_poznavanja_osiguranja  558 non-null int64
pitanje_osiguranje       558 non-null int64
studentski_kredit        558 non-null int64
ocena_poznavanja_studentskog_kredita  558 non-null int64
pitanje_studentski_kredit  558 non-null int64
kriptovalute            558 non-null int64
ocena_poznavanja_kriptovaluta  558 non-null int64
pitanje_kriptovalute     558 non-null int64
pol                      558 non-null int64
nivo_obrazovanja_majka    558 non-null int64
nivo_obrazovanja_otac     558 non-null int64
prosecna_ocena_prethodni_razred  558 non-null float64
prosecna_ocena_matematika  558 non-null float64
dtypes: float64(2), int64(16), object(2)
memory usage: 87.3+ KB
```

Slika 6.2.3. Rezultat pozivanja info() funkcije i redirektovanje rezultata te funkcije u fajl podaci_info.txt

6.3 Anketa

Anketa je podeljena u šest segmenata i to:

- Inflacija
- Kamatna stopa
- Osiguranje
- Studentski kredit
- Kriptovalute
- Opšti anonimni podaci o ispitaniku

U radu će biti predstavljen kod koji analizira svaki od segmenata ankete ponaosob, a zatim će svi segmenti biti razmatrani kao jedna celina.

6.4 Analiza opštih anonimnih podataka o ispitanicima

Ovaj segment je poslednji u anketi, ali će u radu biti prvo biti analizirani podaci o ispitanicima. U kasnijem delu rada će ovi podaci biti povezani sa podacima pojedinih segmenata, a zatim će biti povezani sa podacima cele ankete. U zavisnosti od odabira vrste podataka iz skupa opštih podataka, kod koji je napisan i koji koristi određene pakete

čiji se alati zasnivaju na radu neuronskih mreža, biće moguće dati određena predviđanja. Radi pojednostavljenja predstavljanja rezultata ispitivanja opštih podataka uvedeno je numeričko vrednovanje odgovora.

Opšti podaci koji su traženi od ispitanika su:

- Pol (0 – nije dat odgovor, 1 – ženski, 2 – muški, 3 – dato više odgovora),
- Nivo obrazovanja majke (0 – nije dat odgovor, 1 – osnovno, 2 – srednje, 3 – visoko, 4 – dato više odgovora),
- Nivo obrazovanja oca (0 – nije dat odgovor, 1 – osnovno, 2 – srednje, 3 – visoko, 4 – dato više odgovora),
- Prosečna ocena na kraju prethodne školske godine (podatak je definisiran kao float)
- Prosečna ocena iz matematike na kraju prethodne školske godine (podatak je definisiran kao int)

Sledi analiza ovog segmenta ankete koju je izvršio napisani kod.

Tabela 6.4.1. Udeo ispitanika u zavisnosti od pola ispitanika.

Pol	0	1	2	3	Σ
	32	250	271	5	558
[%]	5,73	44,80	48,57	0,90	100,00

Iz tabele koja se odnosi na pol vidi se da je odnos ženskih i muških ispitanika skoro podjednak. Na dato pitanje nije se izjasnilo 5,73 [%] ispitanika, dok je na isto pitanje 0,90 [%] ispitanika dalo više odgovora.

Tabela 6.4.2. Nivo obrazovanja majke ispitanika

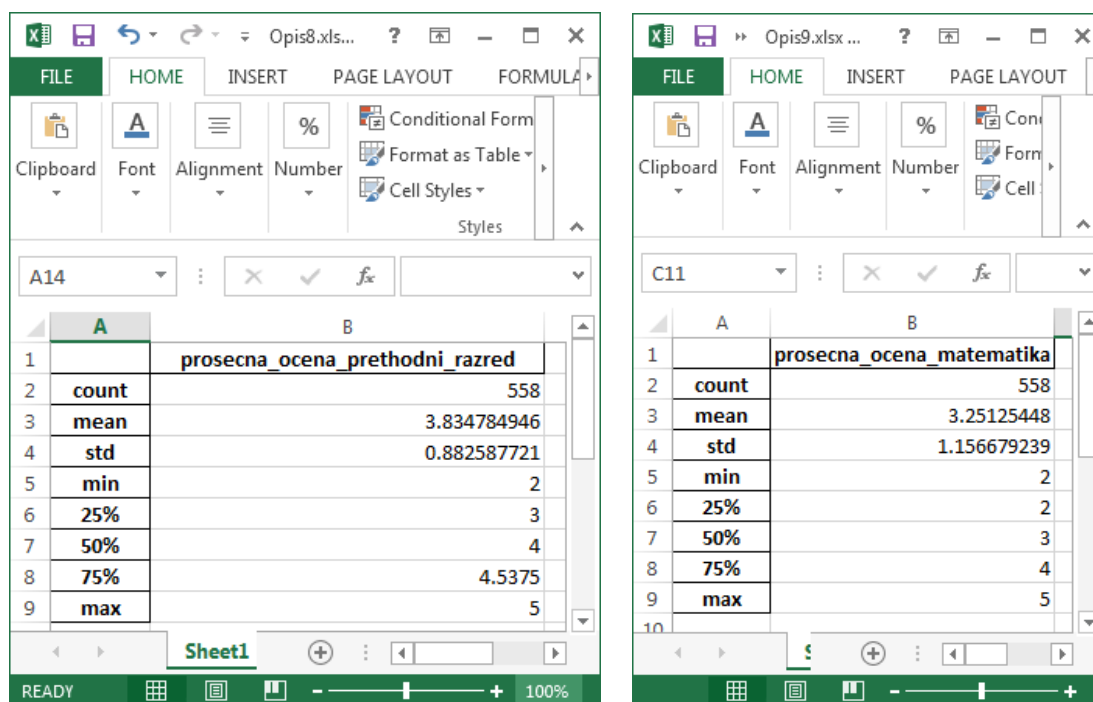
Nivo obrazovanja majke	0	1	2	3	Σ
	33	37	336	152	558
[%]	5,91	6,63	60,22	27,24	100,00

Tabela 6.4.3. Nivo obrazovanja oca ispitanika

Nivo obrazovanja otac	0	1	2	3	Σ
	34	42	347	135	558
[%]	6,09	7,53	62,19	24,19	100,00

Iz tabela koje prikazuju nivo obrazovanja majke i nivo obrazovanja oca se može zaključiti da ima veoma malih odstupanja. U najvećem broju slučajeva su i majka i otac srednje stručne spreme (60,22 [%] i 62,19 [%]) i ukupan prosek koji ovaj nivo obrazovanja roditelja zauzima je 61,21 [%]. Na drugom mestu je visoki nivo obrazovanja čiji je prosek 25,72 [%]. Na trećem mestu je osnovno obrazovanje roditelja čiji je prosek 7,08 [%]. Od ukupnog broja ispitanika njih 6 [%] nije dalo odgovor na postavljeno pitanje. Na pitanja o nivou obrazovanja majke i nivou obrazovanja oca nema zaokruženo više odgovora.

Sledi skrin-šotovi spred-šitova koji prikazuju statističke parametre koji se odnose na prosečnu ocenu na kraju prethodnog razreda i prosečnu ocenu iz matematike na kraju prethodnog razreda. Spred-šitovi su dobijeni korišćenje funkcije describe() u kodu. Više o samoj funkciji u kodu.



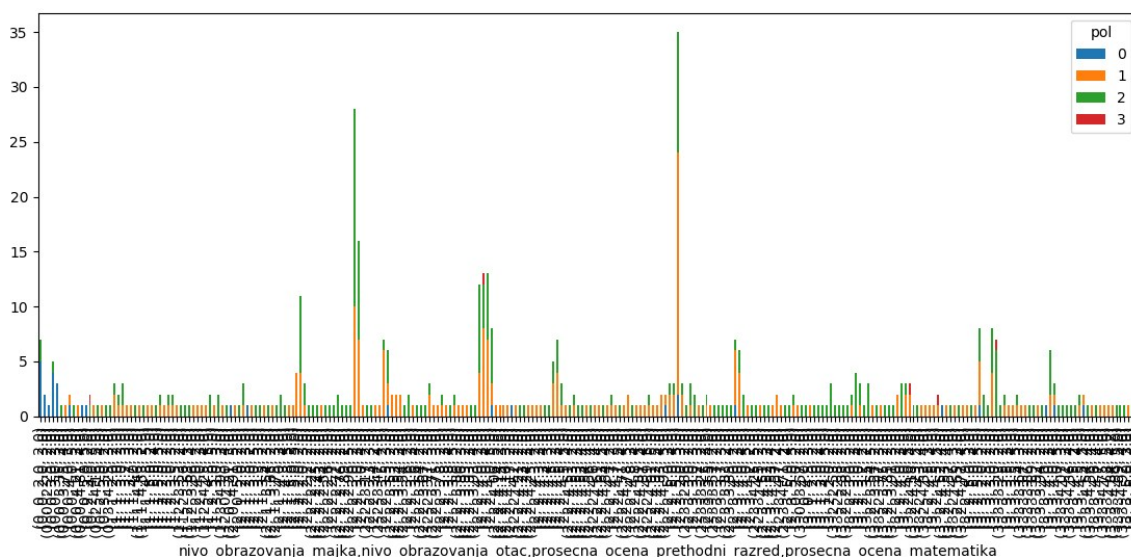
Slika 6.4.1. Preosečna ocena na kraju prethodnog razreda i prosečna ocena iz matematike na kraju prethodnog razreda

Iz navedenih tabela se može uočiti da prosečna ocena iz matematike iznosi 3,25 i da ona odstupa od ukupne prosečne ocene koja iznosi 3,83. Opisnim vrednovanjem prosečnih ocena se može reći da je prosečna ocena iz matematike *dobar*, dok je ukupna prosečna ocena *vrlo dobar*. Iako ovo odstupanje ne deluje preveliko u kasnijem delu radu će biti uočene određene zavisnosti.

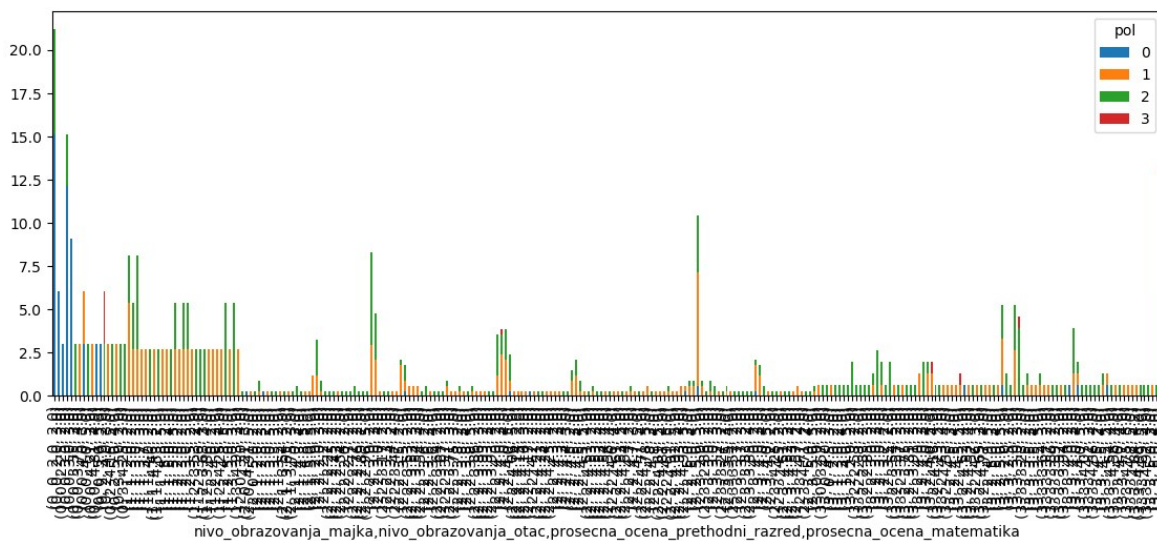
Kod koji je napisan i implementiran omogućava povezivanje ovih podataka u jednu celinu. Zamenom mesta kolona u kodu moguće je dobiti različite zavisnosti koje će biti u funkciji poslednje imenovane kolone u redu koda koji se odnosi na grupisanje i eksportovanje opštih podataka u odgovarajući spreadsheet, npr. ukoliko se želi zavisnost opštih podataka u odnosu na pol, linija koda će glasiti:

```
podaci.groupby(['nivo_ obrazovanja_majka','nivo_ obrazovanja_otac','prosecna_ocena_
prethodni_razred','prosecna_ocena_matematika','pol']).size().unstack().fillna(0).to_ exc
el('Anketa7_ rezultati.xlsx')
```

Slede grafici koji predstavljaju međusobnu zavisnost opštih podataka. U ovom slučaju će biti dat i tabelaran prikaz u okviru samog eksperimentalnog dela doktorske disertacije u cilju što boljeg pojašnjenja dobijenih rezultata. Tabela koja će biti predstavljena je eksportovana iz .xlsx formata u .docx format.



Slika 6.4.2. Međusobna zavisnost opštih podataka od pola ispitanika



Slika 6.4.3. Međusobna zavisnost opštih podataka od pola ispitanika u procentima

Sa gornjih slika se vide zavisnosti opštih podataka od pola ispitanika. Nedostatak grafičkog predstavljanja složenih zavisnosti je u činjenici da veličine na apcisi nisu u potpunosti vidljive (zbog previše zavisnosti). Vizuelno je moguće utvrditi određene odnose među veličinama, ali je iz ovog razloga bolje koristiti tabelarni prikaz veličina. Sledi slika tabelarno dobijenih rezultata (slika se nastavlja na nekoliko strana). Iz tabele se vidi da je najviše ispitanika čija su oba roditelja sa srednjom školom, peticom kao prosečnom ocenom na kraju prethodnog razreda i peticom kao prosečnom ocenom iz matematike na kraju prethodnog razreda. Ukupan broj takvih ispitanika je 35, a po strukturi pola: dvoje je neopredeljeno, dvadesetdvoje ženskog pola i jedanaestoro muškog pola. U odnosu na ukupan broj ispitanika to je 6,27 [%].

Tabela 6.4.4. Međusobna zavisnost opštih podataka od pola ispitanika

nivo_ obrazovanja_majka	nivo_ obrazovanja_otac	prosečna_ oce na_ prethodni_ razred	prosečna_ oc ena_ matema tika	0	1	2	3	Ukupn o_ redo vi
0	0	2	2	5	0	2	0	7
			3	2	0	0	0	2
		2.25	2	1	0	0	0	1
			3	2	4	0	1	0
		3	2	3	0	0	0	3
			3	3	0	0	1	0
		3.35	2	0	0	1	0	1
			3	4	0	1	0	0
3.75	2	3	1	1	0	0	2	
	3	5	0	0	1	0	1	
4	2	3	0	1	0	0	1	
	3	4.21	3	0	1	0	0	1

		4.81	4	1	0	0	0	1		
		5	5	1	0	0	0	1		
		2	3	2	0	0	1	1	2	
			4	3	0	1	0	0	1	
			4.16	5	0	0	1	0	1	
	3	5	4	0	1	0	0	1		
		3	2	0	0	1	0	1		
			4.25	3	0	0	1	0	1	
	1	1	2	2	0	2	1	0	3	
3			2	0	1	1	0	2		
			3	0	1	2	0	3		
3.2			3	0	1	0	0	1		
3.7			3	0	1	0	0	1		
4			2	0	0	1	0	1		
4.46			3	0	1	0	0	1		
4.53			2	0	0	1	0	1		
4.8			5	0	1	0	0	1		
5			2	0	1	0	0	1		
			4	0	0	1	0	1		
			5	0	1	1	0	2		
2			2	2	2	0	1	0	0	1
		3		2	0	1	1	0	2	
		3	3	0	1	1	0	2		
			3.5	2	0	1	0	0	1	
		3.53	2	0	0	1	0	1		
		3.6	4	0	0	1	0	1		
		3.63	3	0	0	1	0	1		
		3.86	4	0	1	0	0	1		
		4	2	0	1	0	0	1		
			3	0	1	0	0	1		
		4.25	4	0	1	0	0	1		
		4.8	5	0	0	2	0	2		
		5	5	0	1	0	0	1		
3			3	3	0	0	2	0	2	
			4.93	3	0	1	0	0	1	
2		0	3	4	0	0	1	0	1	
			4.21	3	1	0	0	0	1	
			4.91	5	0	1	0	0	1	
			5	2	0	0	1	0	1	
		1		2	2	0	1	2	0	3
				2.8	3	1	0	0	0	1
	3			2	0	1	0	0	1	
				5	0	0	1	0	1	

		3.3	3	0	0	1	0	1
		3.5	2	0	1	0	0	1
		3.54	3	0	0	1	0	1
		3.65	2	0	1	0	0	1
		3.75	3	0	0	1	0	1
		4	3	0	0	2	0	2
		4.3	4	0	0	1	0	1
		4.8	5	0	1	0	0	1
		5	3	0	1	0	0	1
			5	0	4	0	0	4
	2	2	2	0	4	7	0	11
			3	0	1	2	0	3
		2.2	2	0	0	1	0	1
		2.22	2	0	0	1	0	1
		2.25	2	0	0	1	0	1
		2.43	2	0	1	0	0	1
		2.54	2	0	0	1	0	1
		2.6	2	0	0	1	0	1
		2.7	2	0	0	1	0	1
		2.72	2	0	0	2	0	2
		2.76	2	0	0	1	0	1
		2.89	2	0	0	1	0	1
		2.96	5	0	0	1	0	1
		3	2	0	10	18	0	28
			3	0	7	9	0	16
			4	0	1	0	0	1
		3.17	2	0	0	1	0	1
		3.3	3	0	0	1	0	1
		3.4	2	0	1	0	0	1
		3.47	4	0	1	0	0	1
		3.5	2	0	6	1	0	7
			3	1	2	3	0	6
		3.53	2	0	2	0	0	2
			3	0	2	0	0	2
		3.54	3	0	2	0	0	2
			4	0	0	1	0	1
		3.6	2	0	0	2	0	2
			3	0	1	0	0	1
			4	0	0	1	0	1
		3.64	2	0	0	1	0	1
		3.67	2	0	0	1	0	1
		3.7	3	0	2	1	0	3
	3.71	3	0	1	0	0	1	

		3.75	3	0	1	0	0	1
		3.8	2	0	1	1	0	2
			3	0	1	0	0	1
			4	0	0	1	0	1
			5	0	1	1	0	2
		3.86	2	0	1	0	0	1
			3	0	1	0	0	1
		3.93	2	0	1	0	0	1
			3	0	0	1	0	1
			4	0	0	1	0	1
		4	2	0	4	8	0	12
			3	0	8	4	1	13
			4	0	7	6	0	13
			5	1	2	5	0	8
		4.06	5	0	1	0	0	1
		4.14	2	0	1	0	0	1
		4.15	3	0	1	0	0	1
		4.16	3	0	1	0	0	1
		4.17	2	1	0	0	0	1
		4.2	2	0	1	0	0	1
		4.21	2	0	0	1	0	1
		4.23	2	0	0	1	0	1
		4.24	3	0	1	0	0	1
		4.25	4	0	1	0	0	1
		4.3	2	0	1	0	0	1
			3	0	1	0	0	1
			4	0	0	1	0	1
		4.4	3	0	0	1	0	1
		4.5	2	0	3	2	0	5
			3	0	4	3	0	7
			4	0	1	2	0	3
			5	0	1	0	0	1
		4.51	4	0	0	1	0	1
		4.53	2	0	1	1	0	2
			3	0	0	1	0	1
			4	0	0	1	0	1
		4.54	4	0	1	0	0	1
		4.56	4	0	0	1	0	1
		4.6	4	0	1	0	0	1
		4.61	4	0	1	0	0	1
		4.63	5	0	0	1	0	1
		4.64	4	0	1	0	0	1
		4.7	2	0	1	1	0	2

			3	0	1	0	0	1	
			5	0	0	1	0	1	
		4.71	4	0	1	0	0	1	
		4.78	5	0	2	0	0	2	
		4.8	3	0	0	1	0	1	
			4	0	1	0	0	1	
		4.81	5	0	1	0	0	1	
		4.87	4	0	1	0	0	1	
		4.9	5	0	1	1	0	2	
		4.91	5	0	1	0	0	1	
		4.92	5	0	1	0	0	1	
		4.93	4	0	2	0	0	2	
			5	1	1	0	0	2	
		5	3	0	2	1	0	3	
			4	0	2	1	0	3	
			5	2	22	11	0	35	
	3	2	3	0	2	1	0	3	
			2.86	2	0	0	1	0	1
			3	2	0	1	2	0	3
				3	0	0	2	0	2
			3.2	3	0	1	0	0	1
			3.27	2	0	0	1	0	1
			3.5	3	0	1	1	0	2
				4	0	1	0	0	1
			3.57	2	0	0	1	0	1
			3.7	2	0	0	1	0	1
				3	0	0	1	0	1
			3.83	2	0	0	1	0	1
			3.85	4	0	0	1	0	1
			4	2	1	5	1	0	7
				3	0	4	2	0	6
				4	0	0	2	0	2
				5	0	1	0	0	1
			4.2	2	0	0	1	0	1
			4.25	5	0	0	1	0	1
			4.5	2	0	1	0	0	1
			4.51	3	0	0	1	0	1
			4.52	3	0	0	1	0	1
			4.53	4	0	1	0	0	1
			4.57	2	0	2	0	0	2
			4.7	4	0	1	0	0	1
			4.71	2	0	0	1	0	1
		5	4	0	0	1	0	1	

			5	0	1	1	0	2
3	0	2	2	0	1	0	0	1
		3.57	2	0	0	1	0	1
		4	2	0	0	1	0	1
		4.5	3	0	1	0	0	1
	1	2	2	0	0	1	0	1
		3	2	0	0	1	0	1
			4	0	0	1	0	1
	4	5	0	0	1	0	1	
	2	2	2	0	0	3	0	3
		2.53	2	0	0	1	0	1
		2.8	2	0	0	1	0	1
		2.83	2	0	0	1	0	1
		2.85	3	0	0	1	0	1
		3	2	0	1	1	0	2
			3	0	0	4	0	4
			4	0	1	2	0	3
		3.2	2	0	0	1	0	1
		3.5	2	0	0	3	0	3
		3.54	5	0	1	0	0	1
		3.57	2	0	0	1	0	1
		3.7	3	0	1	0	0	1
		3.75	2	0	0	1	0	1
		3.85	2	0	0	1	0	1
		3.91	3	0	0	1	0	1
		4	2	0	2	0	0	2
			3	0	0	3	0	3
			4	0	2	1	0	3
			5	0	2	0	1	3
		4.18	4	0	0	1	0	1
		4.23	5	0	0	1	0	1
		4.3	4	0	1	0	0	1
		4.33	4	0	1	0	0	1
		4.5	3	0	1	0	0	1
			4	0	1	0	0	1
			5	0	1	0	1	2
		4.52	4	1	0	0	0	1
		4.53	4	0	1	0	0	1
		4.57	4	0	0	1	0	1
		4.6	3	0	1	0	0	1
		4.65	3	0	0	1	0	1
	4.77	4	0	1	0	0	1	
4.9	5	0	1	0	0	1		

		5	3	0	0	1	0	1	
			4	0	1	0	0	1	
			5	1	4	3	0	8	
3		2	2	0	0	2	0	2	
			2.5	2	0	0	1	0	1
			3	2	0	4	4	0	8
			3	3	0	0	6	1	7
				3.23	2	0	0	1	0
			3.5	2	0	1	1	0	2
				3	0	1	0	0	1
				4	0	1	0	0	1
				5	0	1	1	0	2
			3.64	3	0	1	0	0	1
			3.7	2	0	1	0	0	1
			3.86	2	0	0	1	0	1
			3.9	2	0	0	1	0	1
			3.92	3	0	1	0	0	1
			3.94	2	0	0	1	0	1
			4	2	1	0	0	0	1
				3	0	2	4	0	6
				4	1	1	1	0	3
				5	0	0	1	0	1
			4.07	3	0	0	1	0	1
			4.2	3	0	0	1	0	1
			4.36	4	0	0	1	0	1
			4.5	2	0	0	1	0	1
				4	0	1	1	0	2
				5	1	1	0	0	2
			4.54	4	0	0	1	0	1
			4.56	4	0	1	0	0	1
			4.6	4	0	0	1	0	1
			4.7	4	0	1	0	0	1
			4.75	5	0	1	0	0	1
			4.8	5	0	1	0	0	1
			4.84	4	0	1	0	0	1
			4.86	5	0	0	1	0	1
4.9	5	0	0	1	0	1			
4.92	4	0	0	1	0	1			
5	3	0	1	0	0	1			
	4	0	0	1	0	1			
	5	0	9	11	0	20			
Ukupno kolone			32	25	27	5	558		
				0	1				

							55	
							8	

6.5 Analiza prvog segmenta pitanja

U prvom segmentu ankete se razmatra pojam inflacije. Ovaj segment se sastoji od tri pitanja sa ponuđenim odgovorima. Ukoliko ispitanik na ponuđeno pitanje nije dao odgovor, takav odgovor na pitanje je vrednovan nulom. Sa druge strane ukoliko je ispitanik na ponuđeno pitanje zaokružio više odgovora, takav odgovor je vrednovan sa: broj ponuđenih odgovora +1.

Slede pitanja koja se nalaze u segmentu i način njihovog vrednovanja u Anketa.csv fajlu.

Pitanje 1, Segment 1 ankete

1. *Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam INFLACIJA?*

Odgovori: DA NE

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, DA, NE, 3

0 – nije dat odgovor

DA

NE

3 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na prvo pitanje je u Anketa.csv fajlu nazvana pojam_inflacije.

Tabela 6.5.1. Udeo odgovora na Pitanje 1, Segment 1

<i>Pitanje 1, Segment 1</i>	0	DA	NE	3	Σ
	2	508	48	0	558
Udeo u procentima	0,36	91,04	8,60	0,00	100,00

Pitanje: „*Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam INFLACIJA?*“ predstavlja ujedno i kontrolno pitanje celokupne ankete. Ispitanici su učenici četvrtih (završnih) godina srednjih škola različitih profila. Ispitanici su punoletna ili uskoro punoletna lica koja su u stalnom dodiru sa novim podacima i znanjima, tj. nalaze se u veoma dinamičnom okruženju. Pojam inflacija je pojam koji se, u većoj ili manjoj meri, spominje u različitim predmetima tokom školovanja u srednjoj školi.

Primeri mogu biti sledeći predmeti: matematika (npr. procentni račun), geografija, istorija, ekonomija, ... Procenti odgovora, koji su dati u gornjoj tabeli, u potpunosti potvrđuju malopre navedene činjenice, jer je 91,04 [%] ispitanika odgovorilo potvrdno na dato pitanje. Mali procenat odgovora NE, kao i ne davanja odgovora na postavljeno pitanje, upućuje na to da određeni broj ispitanika, možda, nije čuo za navedeni pojam sa jedne strane (manje verovatno), odnosno da taj procenat ispitanika nije najozbiljnije shvatio učešće u anketi (veća verovatnoća).

Kako su podaci koji su uneti u kolonu pitanje_inflacija tipa string, to će se kao rezultat pokretanja funkcije describe() koja je deo pandas paketa, dobiti frekvenciju odgovora koja se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spređšit Opis.xlsx. Iz tabele se vidi postoji ukupno 558 ispitanika, 3 jedinstvena odgovora (0, DA, NE) i da je 508 ispitanika odgovorilo da je upoznato sa pojmom inflacije.

Da bi mogli da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je string prevesti u podatke tipa int. To je urađeno tako što je u Anketa.csv fajlu promenjena vrednost, pa je: 0=0, DA=1, NE=2, 3=3. Podaci su eksportovani u spređšit Opis1.xlsx. Iz tabele Opis1.xlsx se vidi da je aritmetička sredina približno 1,08 , a to znači da je skoro cela populacija ispitanika (91,04 [%]) potvrdno odgovorila na pitanje o tome da li je upoznato sa pojmom inflacije. Ovaj statistički prikaz pokazuje još jednom da je anketa uspešno sprovedena.

	A	B	C	D
1		pojam_inflacije		
2	count	558		
3	unique	3		
4	top	DA		
5	freq	508		
6				

	A	B	C
1		pojam_inflacije	
2	count	558	
3	mean	1.082437276	
4	std	0.288025175	
5	min	0	
6	25%	1	
7	50%	1	
8	75%	1	
9	max	2	
10			

Slika 6.5.1. Statistički podaci za kolonu pojam_inflacije kada je u pitanju tip podataka string i tip podataka int.

Pitanje 2, Segment 1 ankete

2. Na osnovu Vašeg mišljenja, na skali od 1 do 5. ocenite u kojoj meri ste upoznati sa pojmom INFLACIJA?

Odgovori: 5-odlično, 4-vrlo dobro, 3-dobro, 2-dovoljno, 1-nisam siguran/na

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, 1, 2, 3, 4, 5, 6

0 – nije dat odgovor

1 – nisam siguran/na

2 – dovoljno

3 – dobro

4 – vrlo dobro

5 – odlično

6 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na drugo pitanje je u Anketa.csv fajlu nazvana ocena_poznavanja_inflacije.

Tabela 6.5.2. Udeo odgovora na Pitanje 2, Segment 1

Pitanje 2, Segment 1	0	1	2	3	4	5	6	Σ
	23	79	81	161	116	95	3	558
Udeo u %	4,12	14,16	14,52	28,85	20,79	17,03	0,54	100,01

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni ocena_poznavanja_inflacije. Podaci koji su dobijeni su: ukupan broj ispitanika, aritmetička sredina, standardna devijacija, minimalna vrednost analiziranog skupa, maksimalna vrednost analiziranog skupa, percentili (25., 50. (medijana) i 75. percentil). Podaci su eksportovani u spread-šit Opis2.xlsx. Na osnovu slike koja sledi može se zaključiti da je prosečna ocena poznavanja inflacije (uključujući i pitanja na koja nije dat odgovor, kao i pitanja na koja je dato više odgovora) 3, tj. opisna ocena je *dobro*. Ova ocena je zadovoljavajuća.

	A	B	C	D
1		ocena_poznavanja_inflacije		
2	count	558		
3	mean	3.012544803		
4	std	1.422385905		
5	min	0		
6	25%	2		
7	50%	3		
8	75%	4		
9	max	6		
10				

Slika 6.5.2. Statistički podaci za kolonu ocena_poznavanja_inflacije

Pitanje 3, Segment 1 ankete

3. *Pretpostavimo da se tokom narednih 10 godina cene onoga što kupujete dupliraju. Ako se Vaši prihodi takođe dupliraju, da li ćete moći da kupite više, manje ili približno isto u odnosu na ono što možete da kupite danas?*

Odgovori: a-više, b-manje, C-približno isto, d-ne znam

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, a, b, c, d, 5

0 – nije dat odgovor

a – više

b – manje

c – približno isto

d – ne znam

5 – dato više odgovora

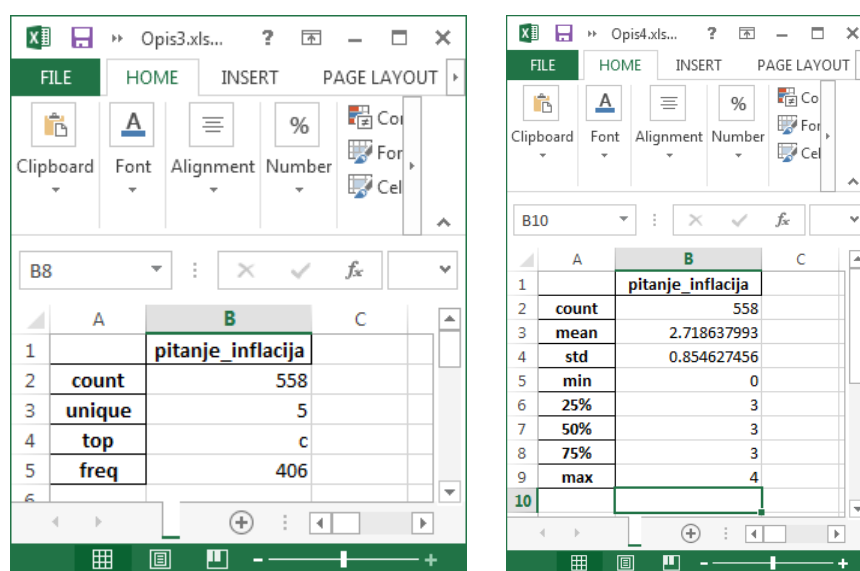
Kolona u koju su smešteni rezultati odgovora na treće pitanje je u Anketa.csv fajlu nazvana pitanje_inflacija.

Tabela 6.5.3. Udeo odgovora na Pitanje 3, Segment 1

<i>Pitanje 3, Segment</i>	0	a	b	c	d	5	Σ
<i>1</i>	28	25	61	406	38	0	558
Udeo u procentima	5,02	4,3	10,93	72,94	6,81	0	100

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni pitanje_inflacija. Kako su podaci koji su uneti u kolonu pitanje_inflacija tipa string, to će kao rezultat pokretanja funkcije describe() kao dela pandas paketa, dati frekvenciju odgovora koji se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spred-šit Opis3.xlsx.

Da bi mogla da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je string prevesti u podatke tipa int. To je urađeno tako što je u Anketa.csv fajlu promenjena vrednost, pa je: 0=0, a=1, b=2, c=3, d=4, 5=5. Podaci su eksportovani u spred-šit Opis4.xlsx. Sa slike koja sledi se može uočiti da je najčešći odgovor koji je dobijen c (3) i da je aritmetička sredina približno 2,72, tj. može se smatrati da je odgovor *c* – *približno isto* odgovor najčešći odgovor na dato pitanje.



Slika 6.5.3. Statistički podaci za kolonu pitanje_inflacija kada je u pitanju tip podataka string i int.

6.5.1 Povezivanje pitanja prvog segmenta u jednu celinu

Na sledećim slikama je data međusobna povezanost prva tri pitanja. Ovo je moguće uraditi zahvaljujući paketu pandas čiji alati omogućavaju složenu analizu podataka. U okviru komentara samog koda, koji će biti kasnije prikazan, se nalaze dodatna objašnjenja za pandas paket. Grafici koji slede su interaktivni i u svakom trenutku se može dobiti odgovarajuća zavisnost između veličina. Kod sadrži komande koje omogućavaju predstavljanje zavisnosti u tabelarnom prikazu, tj. u .xlsx formatu, pa ispod svakog grafika sledi i taberni prikaz rezultata.



Slika 6.5.1.1 Međusobna zavisnost prva tri pitanja u anketi od datih odgovora i broja ispitanika

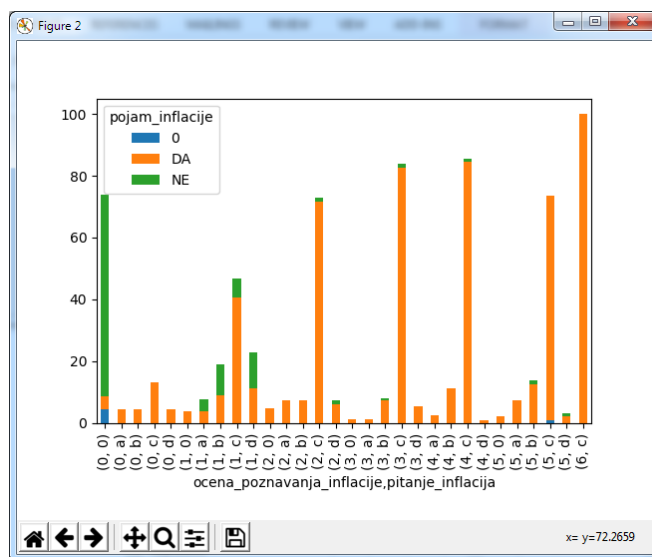
	A	B	C	D	E	F	G	H	I	J
1	pojam_inflacije	ocena_poznavanja_inflacije	0	a	b	c	d	Ukupno_redovi		Ukupno_pojam_inflacije
2	0	0	1	0	0	0	0	1		
3		5	0	0	0	1	0	1		2
4	DA	0	1	1	1	3	1	7		
5		1	3	3	7	32	9	54		
6		2	4	6	6	58	5	79		508
7		3	2	2	12	133	9	158		
8		4	0	3	13	98	1	115		
9		5	2	7	12	69	2	92		
10		6	0	0	0	3	0	3		
11	NE	0	15	0	0	0	0	15		
12		1	0	3	8	5	9	25		
13		2	0	0	0	1	1	2		
14		3	0	0	1	2	0	3		48
15		4	0	0	0	1	0	1		
16		5	0	0	1	0	1	2		
17	Ukupno_kolone		28	25	61	406	38	558		558
18					558					

Slika 6.5.1.2. Tabela prikaz dobijenih rezultata koji su eksportovani u Anкета1_rezultati.xlsx fajl

Analizom međusobno dobijenih rezultata može se ponovo potvrditi da je anketa uspešna, jer se iz prethodne tabele može uočiti da je samo jedan od ispitanika koji je odgovorio da nije upoznat sa pojmom inflacije zaokružio na pitanje o inflaciji odgovor *c*, iako je po anketi, bilo predviđeno da ko ne zaokruži prvo pitanje pređe na drugu grupu pitanja (segment 2). Drugi ispitanik koji ništa nije zaokružio na prvom pitanju se pridržavao uputstva i prešao je na drugi segment pitanja. Iz tabele se može zaključiti da od ispitanika koji su odgovorili potvrdno na pitanje o pojmu inflacije je o oceni poznavanja inflacije dalo ocenu tri, a ujedno su ti ispitanici na pitanje iz oblasti inflacije dali najviše odgovora pod *c*, tj. dali su najviše odgovora *približno isto*. Udeo ispitanika koji su zaokružili pitanja prvog segmenta ankete po sledećem redosledu: DA, 3, *c* je 133, tj. u procentima je to 23,84 [%] od ukupnog broja ispitanika. Ukoliko se pogledaju ostale zavisnosti prikazane u gornjoj tabeli može se zaključiti da je samovrednovanje ispitanika dosta objektivno, jer od broja ispitanika koji su dali odgovor na pitanje o oceni poznavanja inflacije 3,4 ili 5, pogrešan odgovor je dalo (tačan odgovor je pod *c*) 16,71 [%] ili 61 ispitanik od 365 ispitanika koji su ocenili sopstveno poznavanje inflacije sa ocenom 3,4 ili 5. Od 48 ispitanika koje je na pitanje o poznavanju pojma inflacija (pitanje 1, segment 1) dalo odgovor NE, je njih devetoro dalo tačan odgovor na pitanje iz oblasti inflacije, što

je 18,75 [%] od odgovora NE. Ukupan udeo ispitanika koji je dao tačan odgovor na pitanje iz oblasti inflacije je 309 od ukupnog broja ispitanika, tj. 55,38 [%] je dao tačan odgovor na postavljeno pitanje, što se može smatrati prelaznom ocenom. Ukoliko se u obzir uzme činjenica da je na pitanje o pojmu_inflacije odgovor dalo 556 ispitanika (videti: slika 6.2.2) od 558 ispitanika, može se zaključiti da je ukupan procenat ispitanika koji su dali tačan odgovor na pitanje iz oblasti inflacije 55,58 [%] što je razlika od 0,2 [%].

Treba naglasiti da u procentalnoj zavisnosti koja je data na slikama koje slede, procenti se odnose na redove, a ne na kolone, odnosno svaki red se tretira kao posebna jedinica, tj. svaki red tabele se ocenjuje sa sto procenata. Ovo je naglašeno da ne bi došlo do pogrešnog tumačenja rezultata, jer sa slika koje slede se može, pogrešno, zaključiti da o pojmu inflacije najviše znaju ispitanici koji su na pitanje o oceni poznavanja inflacije dali više odgovora (vrednovano sa 6) i koji su, ujedno, na pitanje iz oblasti inflacije dali odgovor c. Odgovor koji je dobijen se odnosi isključivo na dati red, a u datom redu je samo jedan ispitanik zaokružio dva odgovora na pitanje o oceni poznavanja inflacije i na smo pitanje iz oblasti inflacije je zaokružio odgovor pod c. Kako u je u datom redu samo jedan ispitanik, on predstavlja sto procenata za dati red.



Slika 6.5.1.3. Međusobna zavisnost prva tri pitanja u anketi od datih odgovora i broja ispitanika

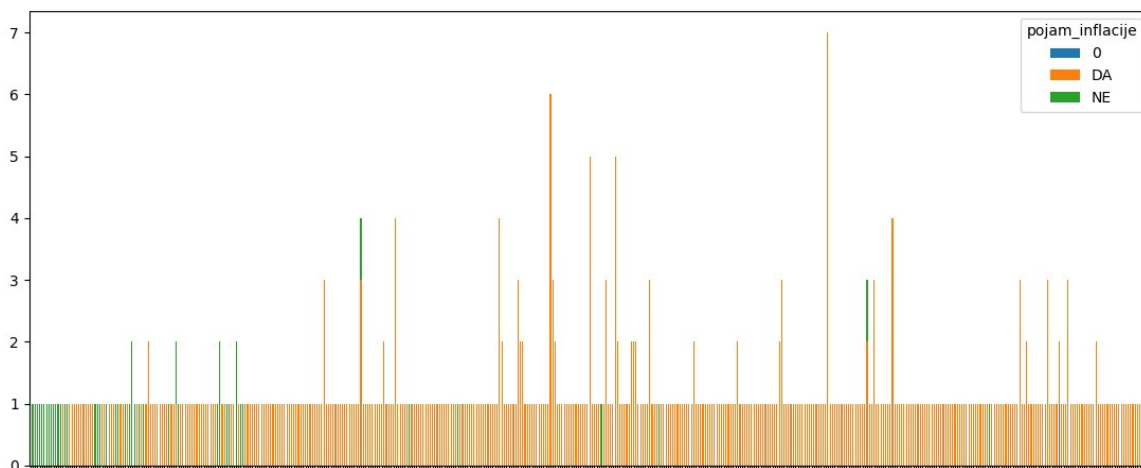
	A	B	C	D	E	F	G	H
	pojam_inflacije	ocena_poznavanja_inflacije	0	a	b	c	d	Ukupno_redovi
1			100					100
2	0	0						100
3		5				100		100
4	DA	0	14.28571	14.28571	14.28571	42.85714	14.28571	100
5		1	5.555556	5.555556	12.96296	59.25926	16.66667	100
6		2	5.063291	7.594937	7.594937	73.41772	6.329114	100
7		3	1.265823	1.265823	7.594937	84.17722	5.696203	100
8		4	2.608696	11.30435	85.21739	0.869565		100
9		5	2.173913	7.608696	13.04348	75	2.173913	100
10		6				100		100
11	NE	0	100					100
12		1		12	32	20	36	100
13		2				50	50	100
14		3			33.33333	66.66667		100
15		4				100		100
16		5			50		50	100

Slika 6.5.1.4. Tabelarni prikaz dobijenih rezultata u procentima koji su eksportovani u Anketa2_rezultati.xlsx fajl

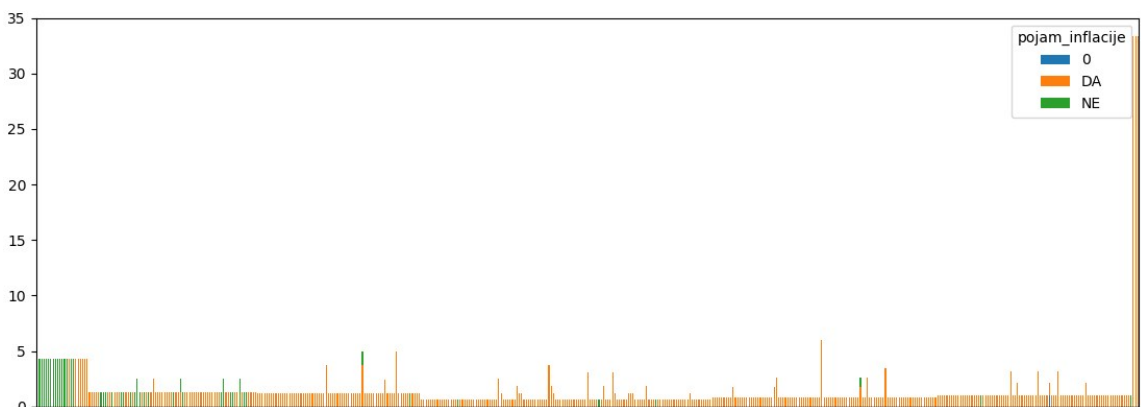
6.5.2 Povezivanje pitanja prvog segmenta i pitanja opšteg tipa u jednu celinu

Sledi statistički pregled prvog segmenta pitanja koji će sada u sebe uključiti pol ispitanika, nivo obrazovanja majke i oca, prosečnu ocenu na kraju prethodno završenog razreda i prosečnu ocenu iz matematike na kraju prethodno završenog razreda. Uvođenjem dodatnih parametara situacija se usložnjava i za utvrđivanje zavisnosti između navedenih veličina je potrebno dosta vremena. Da bi proces rešavanja ovako složenog problema bio olakšan i znatno ubrzan (svega nekoliko sekundi) koristi se kod koji je napisan u programskom jeziku pajton. U kodu je korišćen paket pandas koji omogućava brzo rešavanje problema korišćenjem alata za pretragu ogromnih skupova podataka i određivanje međusobnih zavisnosti takvog skupa podataka. U osnovi većine tih alata leži primena neuronskih mreža.

Kao osnovni parametar koji se koristi tokom analize prvog segmenta anketa u zavisnosti od pola ispitanika, nivoa obrazovanja majke i oca, prosečne ocene na kraju prethodno završenog razreda i prosečne ocene iz matematike na kraju prethodno završenog razreda, se koristi prvo pitanje prvog segmenta koje je ujedno označeno i kao kontrolno pitanje, a koje se odnosi na poznavanje pojma inflacije. Sledi grafički prikaz zavisnosti.



Slika 6.5.2.1. Pitanja prvog segmenta povezana sa opštim pitanjima



Slika 6.5.2.2. Pitanja prvog segmenta povezana sa opštim pitanjima (rezultat je dat u procentima)

Sa prethodne dve slike se vidi da je ovakav prikaz podataka dosta složen. Pitanje o pojmu inflacije je korišćeno kao osnovna veličina u odnosu na koju su računate zavisnosti. Iz tabelarnog prikaza se može zaključiti da kolona *prosecna_ocena_prethodni_razred* predstavlja opterećenje za ramatranje rezultate. Razlog tome je u činjenici da je opseg ocena od 2 do 5 (nije bilo nedovoljnih) kada je u pitanju tip podataka float veliki, tj. nema puno podudaranja prosečnih ocena. Iz navedene tabele se može zaključiti da je najveći stepen podudarnja kada je prosečna ocena na kraju godine bila pet. Sa druge strane kolona *prosecna_ocena_matematika* je tipa int, tj. može imati celobrojne vrednosti između 2 i 5 (nije bilo nedovoljnih). Zato je u kodu kopirana linija koda koja daje sveobuhvatnu zavisnost opštih podataka i prvog segmenta pitanja, a zatim je iz te linije koda izbačena kolona *prosecna_ocena_prethodni_razred*. Iz ovih

rezultata se može zaključiti da najveći broj ispitanika koji odgovorio da poznaje pojam inflacije dolazi iz skupa koji u sebe obuhvata sledeće karakteristike:

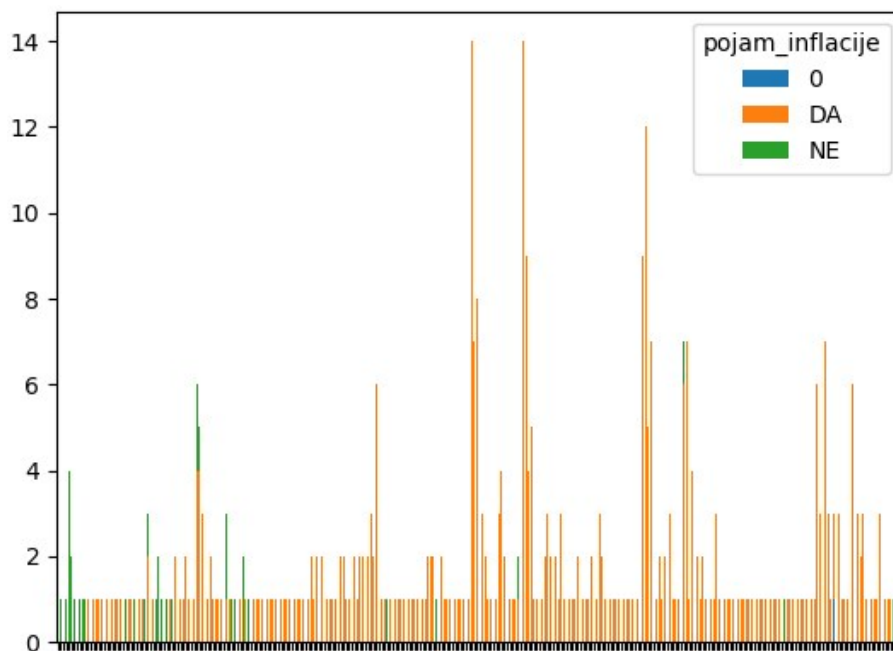
- 14 ispitanika ženskog pola je dalo ocenu poznavanja inflacije 3, odgovor na pitanje iz inflacije c (tačan odgovor), nivo obrazovanja majke – srednjoškolski, nivo obrazovanja oca – srednjoškolski, prosečna ocena matematike iz prethodnog razreda 2.
- 14 ispitanika muškog pola je dalo ocenu poznavanja inflacije 3, odgovor na pitanje iz inflacije c (tačan odgovor), nivo obrazovanja majke – srednjoškolski, nivo obrazovanja oca – srednjoškolski, prosečna ocena matematike iz prethodnog razreda 2.

Sa druge strane broj ispitanika (njih devetnaestoro) koji su potvrdno odgovorili o poznavanju pojma inflacije, ocenili svoje poznavanje pojma inflacije ocenom pet i tačno odgovorili na pitanje iz inflacije je izvučen iz spredšita Anketa12_rezultati i dat je u tabeli koja sledi. Iz petog rede date tabele se vidi da u ovom skupu ispitanika njih sedmoro ima iste karakteristike, a to su da su ženskog pola, da im je nivo obrazovanja majke srednjoškolski i da im je nivo obrazovanja oca srednjoškolski.

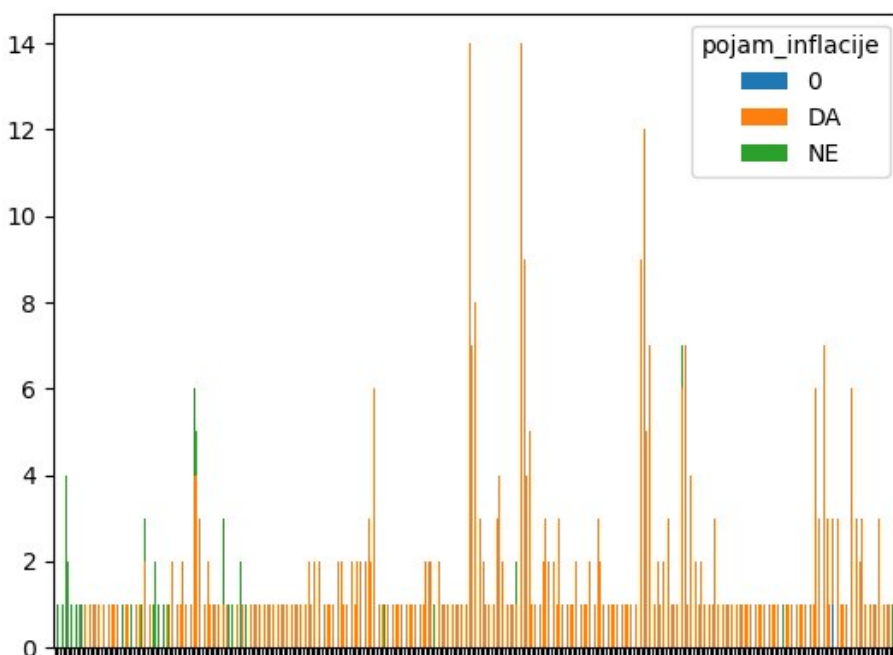
Tabela 6.5.2.1. Opšti podaci za ispitanike sa prosečnom ocenom pet iz matematike

Broj ispitanika	Pol	Nivo obrazovanja majke	Nivo obrazovanja Oca
1	0	2	2
1	0	3	2
1	1	1	2
7	1	2	2
1	1	2	3
2	1	3	2
1	1	3	3
1	2	1	1
2	2	2	2
1	2	3	2
1	2	3	3

Slede grafici zavisnosti ispitavainih parametara.



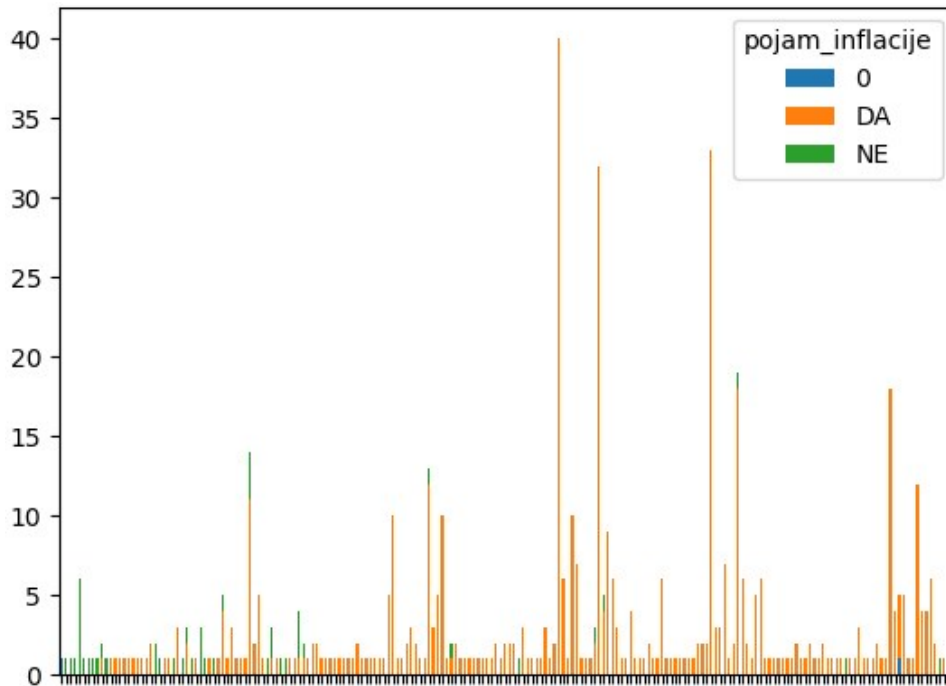
Slika 6.5.2.3. Pitanja prvog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)



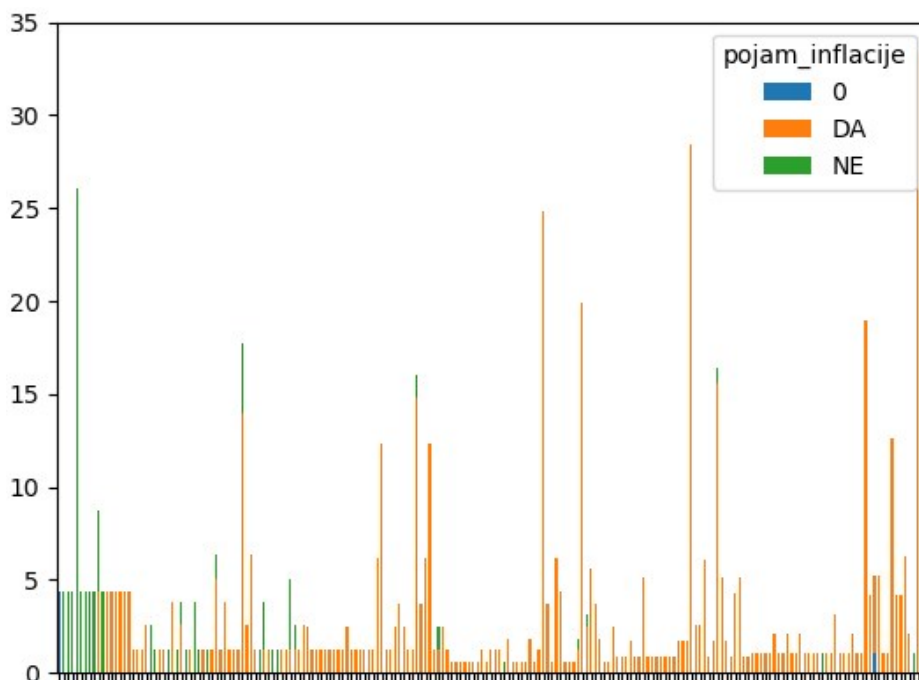
Slika 6.5.2.4. Pitanja prvog segmenta povezana sa opštim pitanjima procentulano (bez pitanja za ukupnu prosečnu ocenu)

Za kraj ovog dela ispitivanja ankete sledi zavisnot parametara prvog segmenta pitanja od opštih parametara, pri čemu će iz te zavisnoti biti uklonjene dve kolone i to:

prosecna_ocena_prethodni_razred i prosecna_ocena_matematika. Cilj je da se rezultati ukupne. Anketa je data u spređšitu Anketa13_rezultati. Iz date tabele se može uočiti da je najveći broj ispitanika (njih četrdesetoro) potvrdno odgovorilo na pitanje o poznavanju pojma inflacije, svoje poznavanje je ocenilo ocenom tri i dalo je tačan odgovor na pitanje o inflaciji. U pitanju je skup koji čine ispitanici ženskog pola kod koga su nivo obrazovanja majke i oca podjednaki, tj. srednjoškolski.



Slika 6.5.2.5. Pitanja prvog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)



Slika 6.5.2.6. Pitanja prvog segmenta povezana sa opštim pitanjima_procentulano (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Iz svih dosad analiziranih podataka koji se tiču prvog segmenta pitanja i opštih podataka, može se izvući zaključak da je najviše ispitanika koji su odgovorili pozitivno na pitanje o poznavanju pojma inflacije svoje znanje inflacije ocenilo ocenom tri, a ujedno i dalo tačan odgovor na postavljeno pitanje o inflaciji. Ocena samovrednovanja poznavanja infalcije je trojka (dobar) i može se zaključiti da je ona u jakoj vezi sa prosečnom srednjom ocenom iz matematike ispitanika koja iznosi 3.25 (dobar). U odnosu na ukupnu prosečnu ocenu 3,86 (vrlo dobar) vidi se veće odstupanje od ocene samovrednovanja i ocene koja se dobija na osnovu odgovora na pitanje o inflaciji, što može dovesti do zaključka da predmeti koji nisu direktno u vezi sa pojmom inflacije podižu vrednost prosečne ocene.

6.6 Analiza drugog segmenta pitanja

U drugom segmentu pitanja se razmatra pojam kamatne stope. Ovaj segment se sastoji od tri pitanja sa ponuđenim odgovorima. Ukoliko ispitanik na ponuđeno pitanje nije dao odgovor, takav odgovor na pitanje je vrednovan nulom. Sa druge strane ukoliko je ispitanik na ponuđeno pitanje zaokružio više odgovora, takav odgovor je vrednovan sa: broj ponuđenih odgovora +1.

Slede pitanja koja se nalaze u segmentu i način njihovog vrednovanja u Anketa.csv fajlu.

Pitanje 1, Segment 2 ankete

1. Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam KAMATNA STOPA?

Odgovori: DA NE

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, DA, NE, 3

0 – nije dat odgovor

DA

NE

3 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na prvo pitanje je u Anketa.csv fajlu nazvana kamatna_stopa.

Tabela 6.6.1. Udeo odgovora na Pitanje 1, Segment 2

<i>Pitanje 1, Segment 2</i>	0	DA	NE	3	Σ
	3	513	42	0	558
Udeo u procentima	0,54	91,94	7,53	0,00	100,01

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni kamatna_stopa. Kako su podaci koji su uneti u kolonu kamatna_stopa tipa string, to će kao rezultat pokretanja funkcije describe() kao dela pandas paketa, dati frekvenciju odgovora koji se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spred-šit Opis10.xlsx. Iz tabele se vidi postoji ukupno 558 ispitanika, 3 jedinstvena odgovora (0, DA, NE) i da je 513 ispitanika odgovorilo da je upoznato sa pojmom .

	A	B	C
1		kamatna_stopa	
2	count	558	
3	unique	3	
4	top	DA	
5	freq	513	
6			
7			
8			
9			

Slika 6.6.1. Statistički podaci za kolonu kamatna_stopa kada je u pitanju tip podataka string i int.

Da bi mogla da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je string prevesti u podatke tipa int (slika 6.6.1.). To je urađeno tako što je u Anketa.csv fajlu promenjena vrednost, pa je: 0=0, DA=1, NE=2, 3=3. Podaci su eksportovani u spređ-šit Opis11.xlsx. Iz tabele Opis11.xlsx se vidi da je aritmetička sredina približno 1,07 , a to znači da je skoro cela populacija ispitanika (91,94 [%]) potvrdno odgovorila na pitanje o tome da li je upoznato sa pojmom kamatne stope. Može se zaključiti da je skoro isti procenat ispitanika odgovorio potvrdno na prvo pitanja prva dva segmenta ankete.

Pitanje 2, Segment 2 ankete

2. Na osnovu Vašeg mišljenja, na skali od 1 do 5. ocenite u kojoj meri ste upoznati sa pojmom KAMATNA STOPA?

Odgovori: 5-odlično, 4-vrlo dobro, 3-dobro, 2-dovoljno, 1-nisam siguran/na

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, 1, 2, 3, 4, 5, 6

0 – nije dat odgovor

1 – nisam siguran/na

2 – dovoljno

3 – dobro

4 – vrlo dobro

5 – odlično

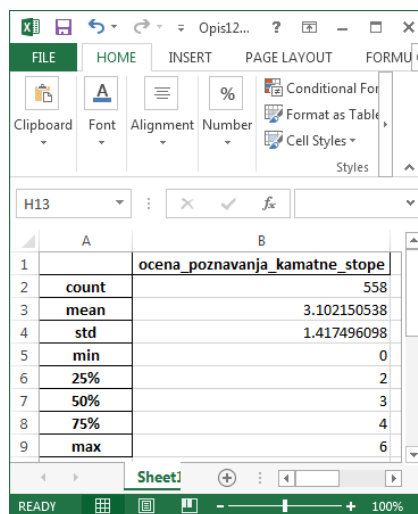
6 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na drugo pitanje je u Anketa.csv fajlu nazvana *ocena_poznavanja_kamatne_stope*.

Tabela 6.6.2. Udeo odgovora na Pitanje 2, Segment 2

<i>Pitanje 2,</i>	0	1	2	3	4	5	6	Σ
<i>Segment 2</i>	27	54	99	141	132	102	3	558
Udeo u %	4,84	9,68	17,74	25,27	23,66	18,28	0,54	100,01

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni *ocena_poznavanja_kamatne_stope*. Podaci koji su dobijeni su: ukupan broj ispitanika, aritmetička sredina, standardna devijacija, minimalna vrednost analiziranog skupa, maksimalna vrednost analiziranog skupa, percentili (25., 50. (medijana) i 75. percentil). Podaci su eksportovani u spred-šit Opis12.xlsx. Na osnovu slike koja sledi može se zaključiti da je prosečna ocena poznavanja kamatne stope (uključujući i pitanja na koja nije dat odgovor, kao i pitanja na koja je dato više odgovora) 3.1, tj. opisna ocena je *dobro*. Ova ocena je zadovoljavajuća. Vide se da u odnosu na drugo pitanje drugog segmenta imamo nešto veću prosečnu ocenu, što je povezano sa češćim pominjanjem pojma kamatna stopa u svakodnevnom životu u odnosu na pojam inflacije.



The screenshot shows an Excel spreadsheet with the following data:

	A	B
1		ocena_poznavanja_kamatne_stope
2	count	558
3	mean	3.102150538
4	std	1.417496098
5	min	0
6	25%	2
7	50%	3
8	75%	4
9	max	6

Slika 6.6.2. Statistički podaci za kolonu *ocena_poznavanja_kamatne_stope*

Pitanje 3, Segment 2 ankete

3. *Pretpostavimo da imate 100 EUR na Vašem štednom računu i da Vam banka pripisuje kamatu od 10 % godišnje. Koliko novca ćete imati na računu posle 5 godina pod uslovom da ne podižete ništa u međuvremenu?*

Odgovori: a-više od 150 EUR, b-manje od 150 EUR, C-Tačno 150 EUR, d-ne znam

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, a, b, c, d, 5

0 – nije dat odgovor

a – više od 150 EUR

b – manje od 150 EUR

c – tačno 150 EUR

d – ne znam

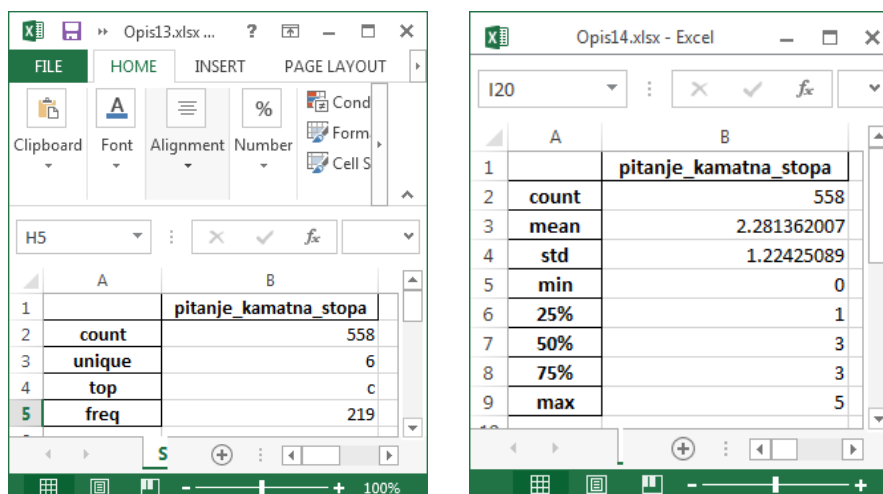
5 – dato više odgovora

Kolona u koju su smešteni rezultati odgovora na treće pitanje je u Anketa.csv fajlu nazvana pitanje_kamatnka_stopa.

Tabela 6.6.3. Udeo odgovora na Pitanje 3, Segment 1

<i>Pitanje</i> 3,	0	a	b	c	d	5	Σ
<i>Segment 2</i>	44	143	68	219	83	1	558
Udeo u %	7,89	25,63	12,19	39,25	14,87	0,18	100,01

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni pitanje_kamatna_stopa. Kako su podaci koji su uneti u kolonu pitanje_kamatna_stopa string, to će kao rezultat pokretanja funkcije describe() kao dela pandas paketa, dati frekvenciju odgovora koji se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spread-šit Opis13.xlsx.

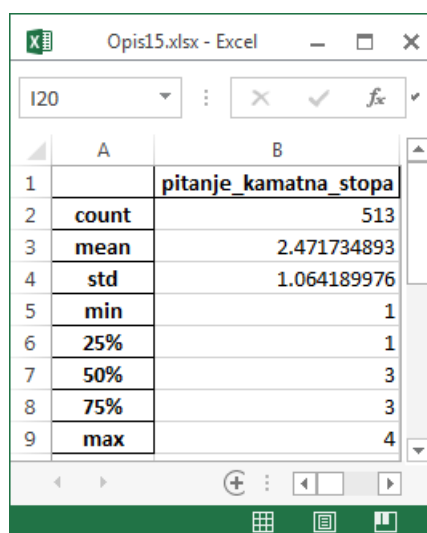


	A	B
1		pitanje_kamatna_stopa
2	count	558
3	unique	6
4	top	c
5	freq	219

	A	B
1		pitanje_kamatna_stopa
2	count	558
3	mean	2.281362007
4	std	1.22425089
5	min	0
6	25%	1
7	50%	3
8	75%	3
9	max	5

Slika 6.6.3. Statistički podaci za kolonu pitanje_kamatna_stopa kada je u pitanju tip podataka string i int.

Da bi mogla da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je string prevesti u podatke tipa int. To je urađeno tako što je u Anketa.csv fajlu promenjena vrednost, pa je: 0=0, a=1, b=2, c=3, d=4, 5=5. Podaci su eksportovani u spređ-šit Opis14.xlsx. Sa slike koja sledi se može uočiti da je najčešći odgovor koji je dobijen c (3) i da je aritmetička sredina približno 2,28. U ovaj prosek ulaze i opcije odgovora 0 (nije dat odgovor) i opcija 5 (dato više odgovora). Ukoliko bi se izbacile te dve opcije iz kolone pitanje_kamatna_stopa dobila bi se ocena 2.47. Primena describe() funkcije na ovako kreiranu kolonu u fajlu Anketa.csv je data na sedećoj slici.



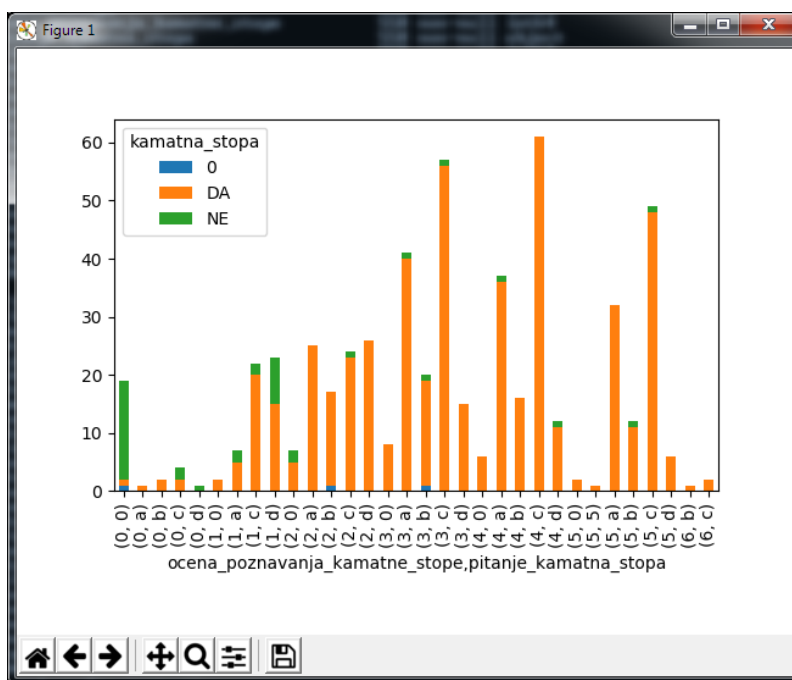
	A	B
1		pitanje_kamatna_stopa
2	count	513
3	mean	2.471734893
4	std	1.064189976
5	min	1
6	25%	1
7	50%	3
8	75%	3
9	max	4

Slika 6.6.4. Statistički podaci za kolonu pitanje_kamatna_stopa

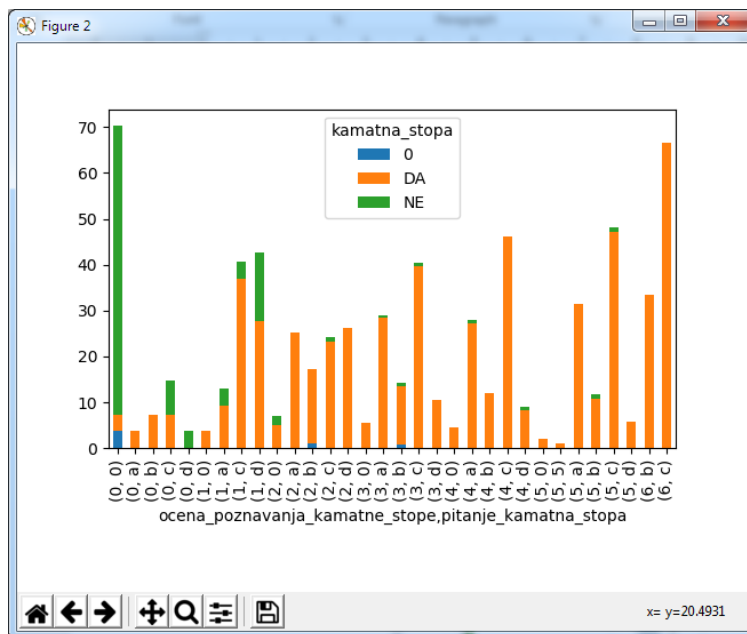
Dobijene ocene na oba načina se razlikuju za 0,19. Ova razlika ne utiče bitnije na sliku da su ispitanici davali dosta pogrešnih odgovora na postavljeno pitanje. U kasnijem delu rada će biti povezan drugi segment pitanja ankete sa opštim podacima i biće pokušano da se dokaže veza između prosečne ocene iz matematike i kolone pitanja_kamatna_stopa. Odgovor na ovo pitanje zahteva poznavanje prostog kamatnog računa i ukazuje na usku povezanost sa matematikom. Na taj način odgovori na ovo pitanje predstavljaju jedan od osnovnih parametara ove ankete i finansijske pismenosti dece.

6.6.1 Povezivanje pitanja drugog segmenta u jednu celinu

Na sledećim slikama je data međusobna povezanost prva tri pitanja drugog segmenta ankete.



Slika 6.6.1.1. Grafik odnosa pitanja drugog segmenta ankete



Slika 6.6.1.2. Grafik odnosa pitanja drugog segmenta ankete u procentima

	A	B	C	D	E	F	G	H	I
1	kamatna_stopa	ocena_poznavanja_kamatne_stope	0	5	a	b	c	d	Ukupno_redovi
2	0	0	1	0	0	0	0	0	1
3		2	0	0	0	1	0	0	1
4		3	0	0	0	1	0	0	1
5	DA	0	1	0	1	2	2	0	6
6		1	2	0	5	0	20	15	42
7		2	5	0	25	16	23	26	95
8		3	8	0	40	18	56	15	137
9		4	6	0	36	16	61	11	130
10		5	2	1	32	11	48	6	100
11	NE	6	0	0	0	1	2	0	3
12		0	17	0	0	0	2	1	20
13		1	0	0	2	0	2	8	12
14		2	2	0	0	0	1	0	3
15		3	0	0	1	1	1	0	3
16	4	0	0	1	0	0	1	2	
17	5	0	0	0	1	1	0	2	
18	Ukupno_kolone		44	1	143	68	219	83	558
19								558	

Slika 6.6.3. Podaci eksportovani u spread-sheet

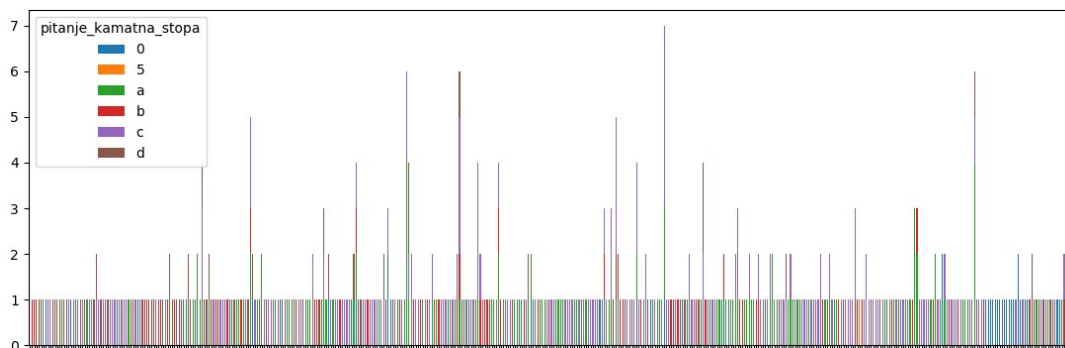
Iz tabele se može zaključiti da od ispitanika koji su odgovorili potvrdno na pitanje o pojmu kamatne stope je o oceni kamatne stope dao ocenu četiri, a ujedno su ti ispitanici na zadatak iz prostog kamatnog računa dali najviše odgovora pod c, tj. dali su najviše

odgovora *tačno 150 EUR*. Udeo ispitanika koji su zaokružili pitanja drugog segmenta ankete po sledećem redosledu: DA, 4, c je 61. tj, u procentima je to 10,93 [%] od ukupnog broja ispitanika. Vrlo blizu ovog rezultata se nalazi rezultat u redosledu: DA, 3, c i on iznosi 10,04 [%]. Ukoliko se pogledaju ostale zavisnosti prikazane u gornjoj tabeli može se zaključiti da je samovrednovanje ispitanika, za razliku od prvog segmenta pitanja ankete, na granici objektivnosti, jer od broja ispitanika koji su dali odgovor na pitanje o oceni poznavanja kamatne stope 3,4 ili 5, pogrešan odgovor je dalo (tačan odgovor je pod c) 55,04 [%] ili 202 ispitanik od 367 ispitanika. Iz datog spred-šita se može zaključiti da je matematika jedan od predmeta koji zadaje probleme đacima, tj. đaci nisu u dovoljnoj meri spremni da samostalno otvaraju bankovne račune ili uzimaju kredit. Ova činjenica predstavlja upozorenje, jer su ispitanici skoro svršeni srednjoškolci koji će imati priliku da rade i zarađuju novac, ali nisu dovoljno upoznati sa načinima na koji način treba postupati sa datim novcem i na koji način zastupati svoje interese pred finansijskim institucijama.

6.6.2 Povezivanje pitanja drugog segmenta i pitanja opšteg tipa u jednu celinu

Sledi statistički pregled drugog segmenta pitanja koji će sada u sebe uključiti pol ispitanika, nivo obrazovanja majke i oca, prosečnu ocenu na kraju prethodno završenog razreda i prosečnu ocenu iz matematike na kraju prethodno završenog razreda.

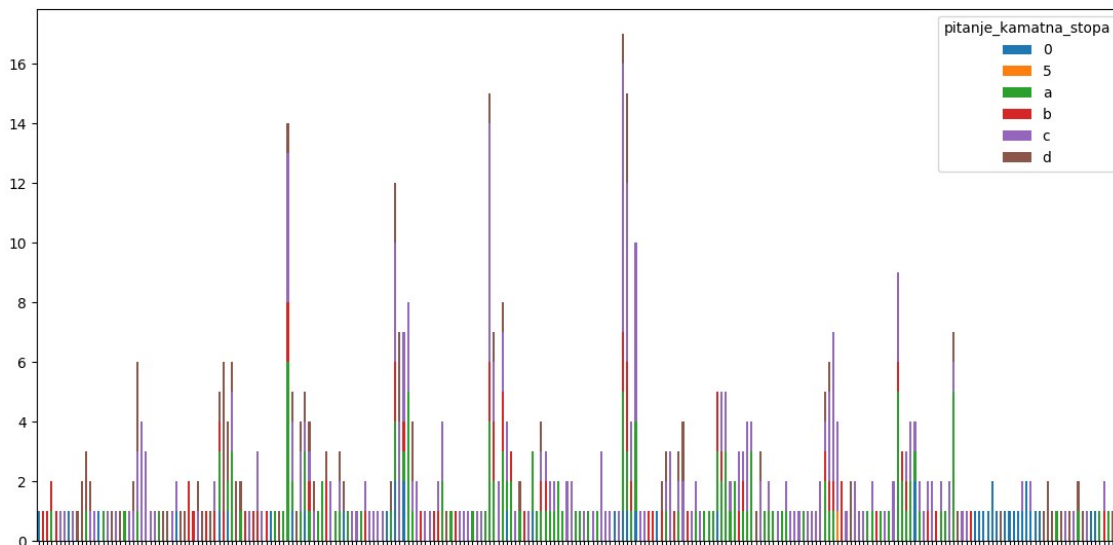
Kao osnovni parametar koji se koristi tokom analize drugog segmenta anketa u zavisnosti od pola ispitanika, nivoa obrazovanja majke i oca, prosečne ocenu na kraju prethodno završenog razreda i prosečne ocenu iz matematike na kraju prethodno završenog razreda, se koristi zadatak iz prostog kamatnog računa. Sledi grafički prikaz zavisnosti.



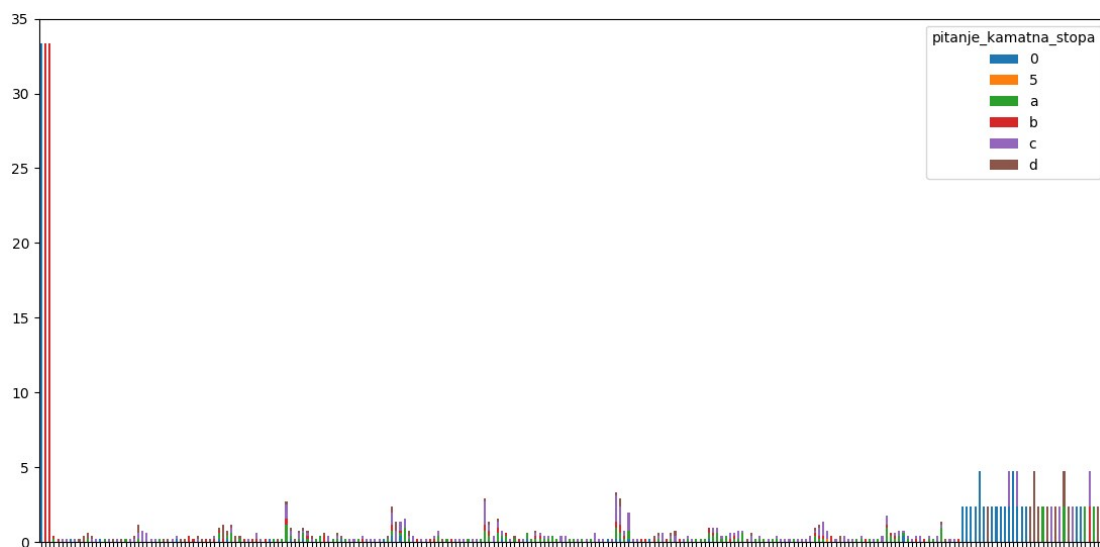
Slika 6.6.2.1. Pitanja drugog segmenta povezana sa opštim pitanjima



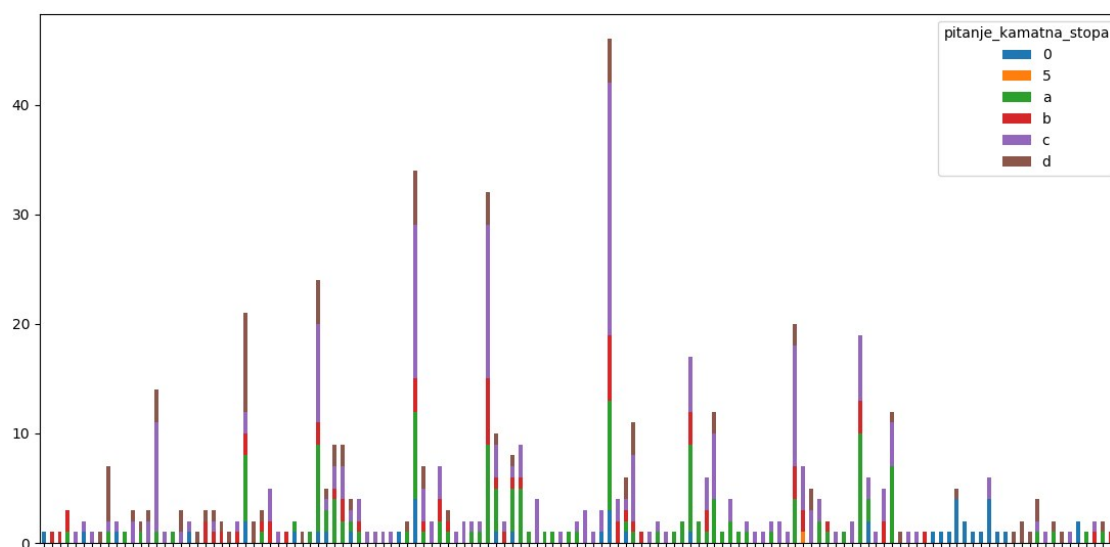
Slika 6.6.2.2. Pitanja drugog segmenta povezana sa opštim pitanjima_procentulano



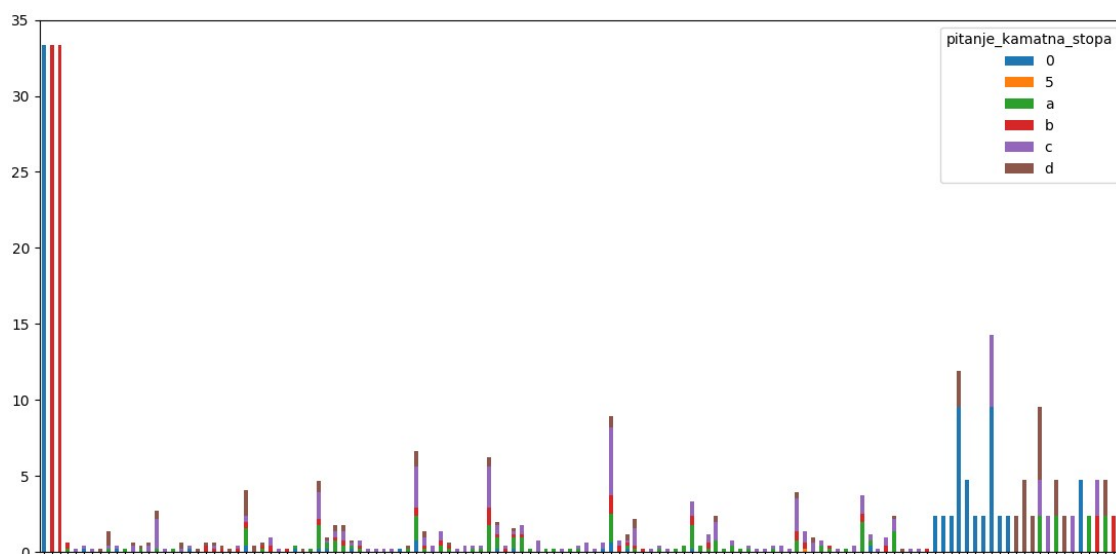
Slika 6.6.2.3. Pitanja drugog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)



Slika 6.6.2.4. Pitanja drugog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu)



Slika 6.6.2.5. Pitanja drugog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)



Slika 6.6.2.6. Pitanja drugog segmenta povezana sa opštim pitanjima `_procentualno` (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Iz datog tabelarnog prikaza se može zaključiti da kolona `prosecna_ocena_prethodni_razred` predstavlja i u ovom slučaju opterećenje za rmatranje rezultata. Razlog tome je u činjenici da je opseg ocena od 2 do 5 (nije bilo nedovoljnih) kada je u pitanju tip podataka float veliki, tj. nema puno podudaranja prosečnih ocena. Iz navedene tabele se može zaključiti da je najveći stepen podudarnja kada je prosečna ocena na kraju godine bila pet (četiri ispitanika je dalo odgovor na pitanje o pojmu kamatne stope DA, ocenilo je svoje poznavanje pojma kamatne stope peticom, dalo je tačan odgovor zadatak iz prostog kamatnog računa, majka i otac su istog niova obrazovanja – srednjoškolskog i prosečna ocena iz matematike je petica). Sa druge strane kolona `prosecna_ocena_matematika` je tipa int. tj. može imati celobrojne vrednosti između 2 i 5 (nije bilo nedovoljnih). Zato je u kodu kopirana linija koda koja daje sveobuhvatnu zavisnost opštih podataka i prvog segmenta pitanja, a zatim je iz te linije koda izbačena kolona `prosecna_ocena_prethodni_razred`. Iz tabelarnih rezultata se može zaključiti da najveći broj ispitanika koji odgovorio da poznaje kamatne stope dolazi iz skupa koji u sebe obuhvata sledeće karakteristike:

- 9 ispitanika muškog pola je odgovorilo potvrdno pitanje o poznavanju pojma kamatne stope i svoje poznavanje kamatne stope je ocenilo ocenom 4. Ovi ispitanici su odgovori tačno na zadatak iz prostog kamatnog računa , Nivo

obrazovanja majke i oca je isti, tj. srednjoškolski, a prosečna ocena matematike iz prethodnog razreda je 2.

- 8 ispitanika ženskog pola je odgovorilo potvrdno pitanje o poznavanju pojma kamatne stope i svoje poznavanje kamatne stope je ocenilo ocenom 3. Ovi ispitanici su odgovori tačno na zadatak iz prostog kamatnog računa. Nivo obrazovanja majke i oca je isti, tj. srednjoškolski, a prosečna ocena matematike iz prethodnog razreda je 2.

Ukoliko se izbaci i kolona koja se odnosi na prosečnu ocenu iz matematike može se uočiti da je 23 ispitanika odgovorilo tačno na zadatak iz prostog kamatnog računa. Karakteristike takvog skupa su: ocena poznavanja pojma kamatna stopa je 4, pol muški, nivo obrazovanja majke srednjoškolski i nivo obrazovanja oca srednjoškolski. Ujedno ovaj skup ispitanika predstavlja skup sa najvećim brojem elemenata u datom razmatranju (odnosi se na sve odgovore u koloni pol)

Što se tiče ispitanika ženskog pola, njih 14 je odgovorilo tačno na zadatak iz prostog kamatnog računa. Karakteristike takvog skupa su: ocena poznavanja pojma kamatna stopa 3, nivo obrazovanja majke srednjoškolski i nivo obrazovanja oca srednjoškolski. Ujedno ovaj skup ispitanika predstavlja skup sa najvećim brojem elemenata u datom razmatranju (odnosi se na odgovore u koloni pol = „ženski“). Pored toga ovaj skup je drugi najbrojniji skup u datom razmatranju (odnosi se na sve odgovore u koloni pol).

6.7 Analiza trećeg segmenta pitanja

U trećem segmentu pitanja se razmatra pojam osiguranja. Ovaj segment se sastoji od tri pitanja sa ponuđenim odgovorima. Ukoliko ispitanik na ponuđeno pitanje nije dao odgovor, takav odgovor na pitanje je vrednovan nulom. Sa druge strane ukoliko je ispitanik na ponuđeno pitanje zaokružio više odgovora, takav odgovor je vrednovan sa: broj ponuđenih odgovora +1.

Slede pitanja koja se nalaze u segmentu i način njihovog vrednovanja u Anketa.csv fajlu.

Pitanje 1, Segment 3 ankete

- 1. Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam OSIGURANJE?*

Odgovori: DA NE

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, DA, NE, 3

0 – nije dat odgovor

DA

NE

3 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na prvo pitanje je u Anketa.csv fajlu nazvana osiguranje.

Tabela 6.7.1. Udeo odgovora na Pitanje 1, Segment 3

<i>Pitanje 1, Segment 3</i>	0	DA	NE	3	Σ
	5	534	19	0	558
Udeo u procentima	0,90	95,70	3,41	0,00	100,01

Mali procenat odgovora NE, kao i ne davanja odgovora na postavljeno pitanje, upućuje na to da određeni broj ispitanika, možda, nije čuo za navedeni pojam sa jedne strane (manje verovatno), odnosno da taj procenat ispitanika nije najozbiljnije shvatio učešće u anketi (veća verovatnoća).

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni osiguranje. Podaci su eksportovani u spređ-šit Opis16.xlsx. Iz tabele se vidi postoji ukupno 558 ispitanika, 3 jedinstvena odgovora (0, DA, NE) i da je 534 ispitanika odgovorilo da je upoznato sa pojmom .

	A	B
1		osiguranje
2	count	558
3	mean	1.025089606
4	std	0.206051821
5	min	0
6	25%	1
7	50%	1
8	75%	1
9	max	2

	A	B
1		osiguranje
2	count	558
3	unique	3
4	top	DA
5	freq	534
6		

Slika 6.7.1. Statistički podaci za kolonu osiguranje kada je u pitanju tip podataka string i int.

Da bi mogla da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je string prevesti u podatke tipa int. To je urađeno tako što je u Anketa.csv fajlu promenjena vrednost, pa je: 0=0, DA=1, NE=2, 3=3. Podaci su eksportovani u spređ-šit Opis17.xlsx. Iz sred-šita Opis17.xlsx se vidi da je aritmetička sredina približno 1,03 , a to znači da je skoro cela populacija ispitanika (95,70[%]) potvrdno odgovorila na pitanje o tome da li je upoznato sa pojmom osiguranja. Može se zaključiti da je u odnosu na prva dva segmenta u trećem segmentu potvrdno na prvo pitanje odgovorilo skoro 4 [%] ispitanika više.

Pitanje 2, Segment 3 ankete

2. Na osnovu Vašeg mišljenja, na skali od 1 do 5, ocenite u kojoj meri ste upoznati sa pojmom OSIGURANJE?

Odgovori: 5-odlično, 4-vrlo dobro, 3-dobro, 2-dovoljno, 1-nisam siguran/na

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, 1, 2, 3, 4, 5, 6

0 – nije dat odgovor

1 – nisam siguran/na

2 – dovoljno

3 – dobro

4 – vrlo dobro

5 – odlično

6 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na drugo pitanje je u Anketa.csv fajlu nazvana ocena_poznavanja_osiguranja.

Tabela 6.7.2. Udeo odgovora na Pitanje 2, Segment 3

<i>Pitanje 2,</i>	0	1	2	3	4	5	6	Σ
<i>Segment 3</i>	13	14	30	96	175	227	3	558
Udeo u %	2,33	2,51	5,38	17,20	31,36	40,68	0,54	100,00

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni `ocena_poznavanja_osiguranja`. Podaci koji su dobijeni su: ukupan broj ispitanika, aritmetička sredina, standardna devijacija, minimalna vrednost analiziranog skupa, maksimalna vrednost analiziranog skupa, percentili (25., 50. (medijana) i 75. percentil). Podaci su eksportovani u spred-šit `Opis18.xlsx`. Na osnovu slike koja sledi može se zaključiti da je prosečna ocena poznavanja pojma opsiguranje (uključujući i pitanja na koja nije dat odgovor, kao i pitanja na koja je dato više odgovora) $3,97 \approx 4,00$, tj. opisna ocena je *vrlo dobro*. Ova ocena je viša u odnosu na odgovore na pitanja koja se bave ocenjivanjem poznavanja određenog pojma u prva dva segmenta.

The screenshot shows an Excel spreadsheet with the following data:

	A	B
1		ocena_poznavanja_osiguranja
2	count	558
3	mean	3.96953405
4	std	1.189784525
5	min	0
6	25%	3
7	50%	4
8	75%	5
9	max	6

Slika 6.7.2. Statistički podaci za kolonu `ocena_poznavanja_osiguranja`

Pitanje 3, Segment 3 ankete

4. *Sklapanjem polise osiguranja sa osiguravajućom kompanijom:*

Odgovori: a-Vi preuzimate rizik od osiguravajuće kompanije, b – Oiguravajuća kompanija preuzima rizik od Vas, c-Niko ne preuzima rizik, d-ne znam

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, a, b, c, d, 5

0 – nije dat odgovor

a – Vi preuzimate rizik od osiguravajuće kompanije

b – Osiguravajuća kompanija preuzima rizik od Vas

c – Niko ne preuzima rizik

d – ne znam

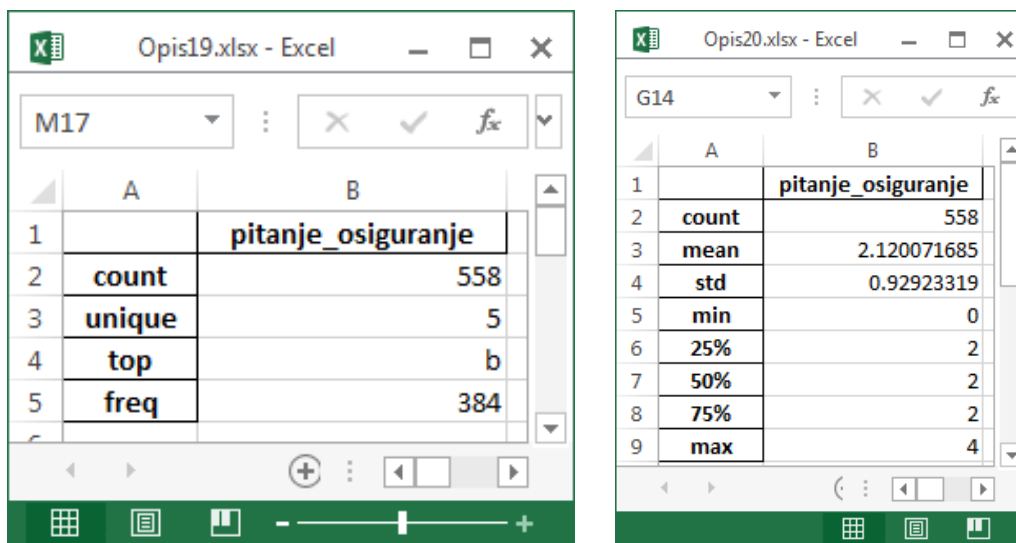
5 – dato više odgovora

Kolona u koju su smešteni rezultati odgovora na treće pitanje je u Anketa.csv fajlu nazvana pitanje_osiguranje.

Tabela 6.7.3. Udeo odgovora na Pitanje 3, Segment 3

<i>Pitanje 3, Segment 3</i>	0	a	b	c	d	5	Σ
	42	22	384	47	63	0	558
Udeo u procentima	7,53	3,94	68,82	8,42	11,29	0	100

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni pitanje_osiguranje. Kako su podaci koji su uneti u kolonu pitanje_osiguranje tipa string, to će kao rezultat pokretanja funkcije describe() kao dela pandas paketa, dati frekvenciju odgovora koji se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spred-šit Opis19.xlsx. Sa slike koja sledi, vidi se da je broj ponuđenih odgovora 5, a da je odgovor pod b zaokružilo 384 ispitanika



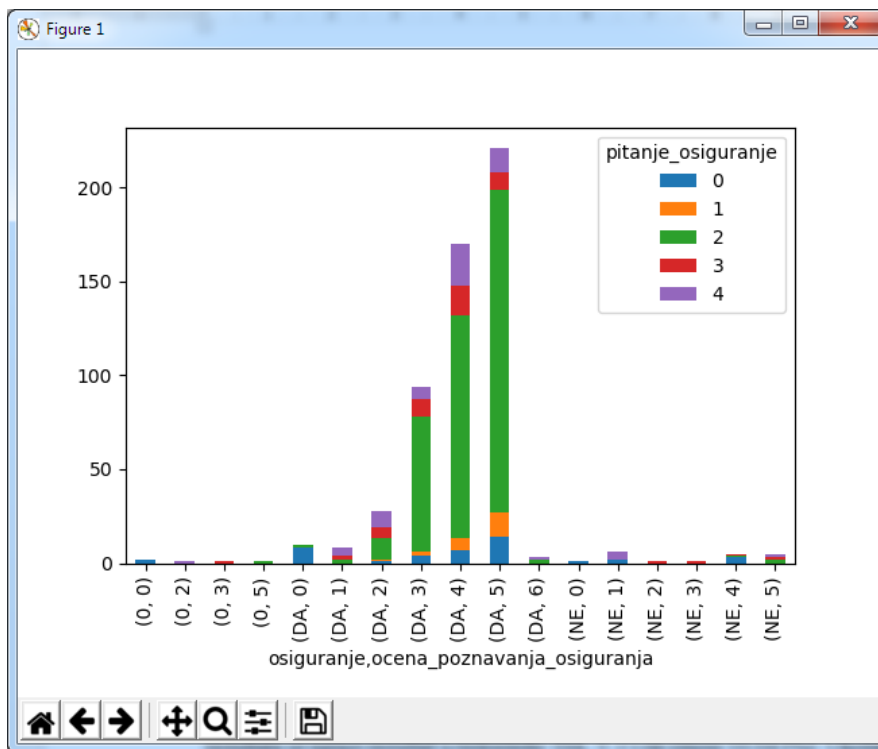
Slika 6.7.3. Statistički podaci za kolonu pitanje_osiguranje kada je u pitanju tip podataka string i int.

Da bi mogla da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je string prevesti u podatke tipa int. To je urađeno tako što je u Anketa.csv fajlu promenjena vrednost, pa je: 0=0, a=1, b=2, c=3, d=4, 5=5. Podaci su eksportovani u spred-šit Opis20.xlsx. Sa slike koja sledi se može uočiti da je

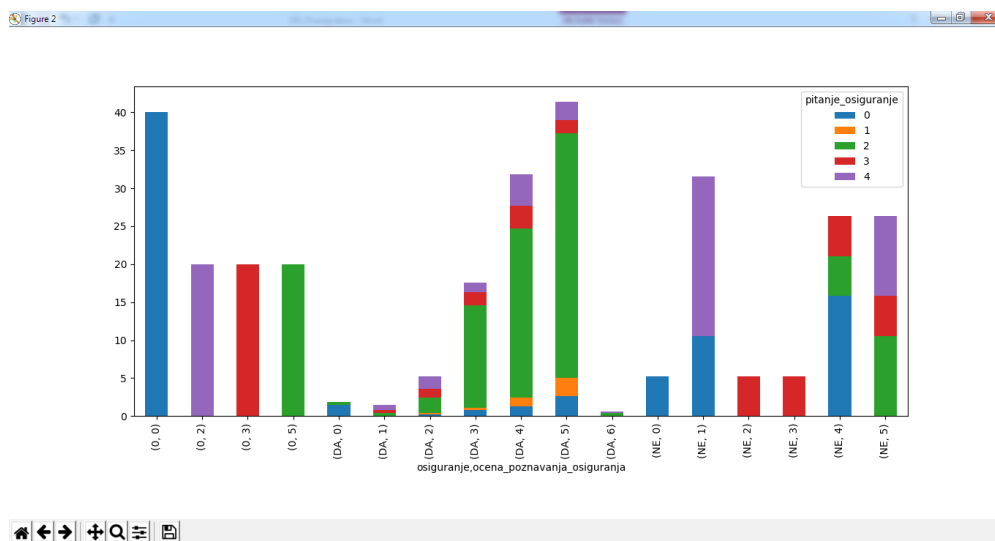
aritmetička sredina približno 2,12, a da je najčešći odgovor na pitanje iz osiguranja odgovor *b – Osiguravajuća kompanija preuzima rizik od Vas* odgovor najčešći odgovor na dato pitanje.

6.7.1 Povezivanje pitanja trećeg segmenta u jednu celinu

Slede slike međusobnih zavisnosti pitanja koji treći segment ankete



Slika 6.7.1.1. Grafik odnosa pitanja trećeg segmenta ankete



Slika 6.7.1.2. Grafik odnosa pitanja trećeg segmenta ankete_procentualno

	A	B	C	D	E	F	G	H
1	osiguranje	ocena_poznavanja_osiguranja	0	1	2	3	4	Ukupno_redovi
2	0	0	2	0	0	0	0	2
3		2	0	0	0	0	1	1
4		3	0	0	0	1	0	1
5		5	0	0	1	0	0	1
6		0	8	0	2	0	0	10
7	DA	1	0	0	2	2	4	8
8		2	1	1	11	6	9	28
9		3	4	2	72	9	7	94
10		4	7	6	119	16	22	170
11		5	14	13	172	9	13	221
12		6	0	0	2	0	1	3
13	NE	0	1	0	0	0	0	1
14		1	2	0	0	0	4	6
15		2	0	0	0	1	0	1
16		3	0	0	0	1	0	1
17		4	3	0	1	1	0	5
18	5	0	0	2	1	2	5	
19	Ukupno_kolone		42	22	384	47	63	558
20							558	

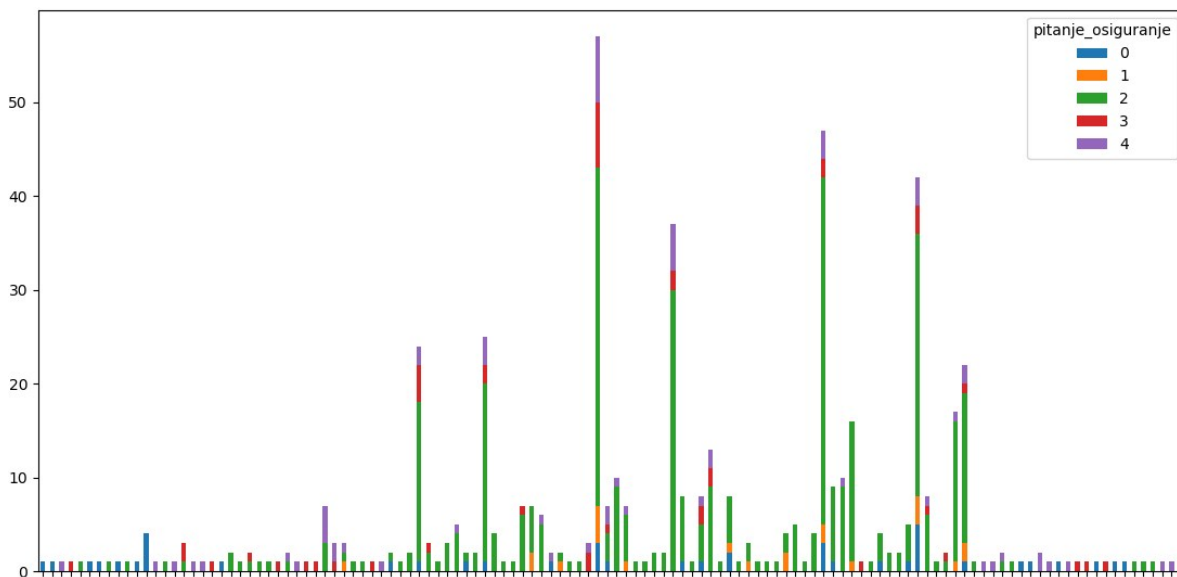
Slika 6.7.1.3. Podaci eksportovani u spread-sheet

Sa gornje slike se može zaključiti da je najveći broj ispitanika (njih 172 ili 30,82 [%] od ukupnog broja ispitanika) odgovorilo da poznaje pojam osiguranja dajući sebi ocenu pet. Ovaj broj ispitanika je ujedno i tačno odgovorio na postavljeno pitanje (odgovor b). Vrlo blizu ovog rezultata se nalazi rezultat u redosledu: DA, 3, b i on iznosi 21,33 [%] ili 119 ispitanika. Ukoliko se pogledaju ostale zavisnosti prikazane u gornjoj tabeli može se zaključiti da je samovrednovanje ispitanika veoma objektivno, jer od broja ispitanika koji su dali odgovor na pitanje o oceni poznavanja osiguranja 3,4 ili 5 tačan odgovor je dalo 74,85 [%] ili od 485 koji su dali odgovor na pitanje o oceni poznavanja osiguranja 3,4 ili 5 tačan odgovor je dalo 363. Iz ovoga se može zaključiti da je pitanje iz oblasti osiguranja, za razliku od trećeg pitanja iz oblasti inflacije i zadatka iz oblasti kamatne stope, bliže ispitanicima, tj. ovo pitanje je opšteg tipa.

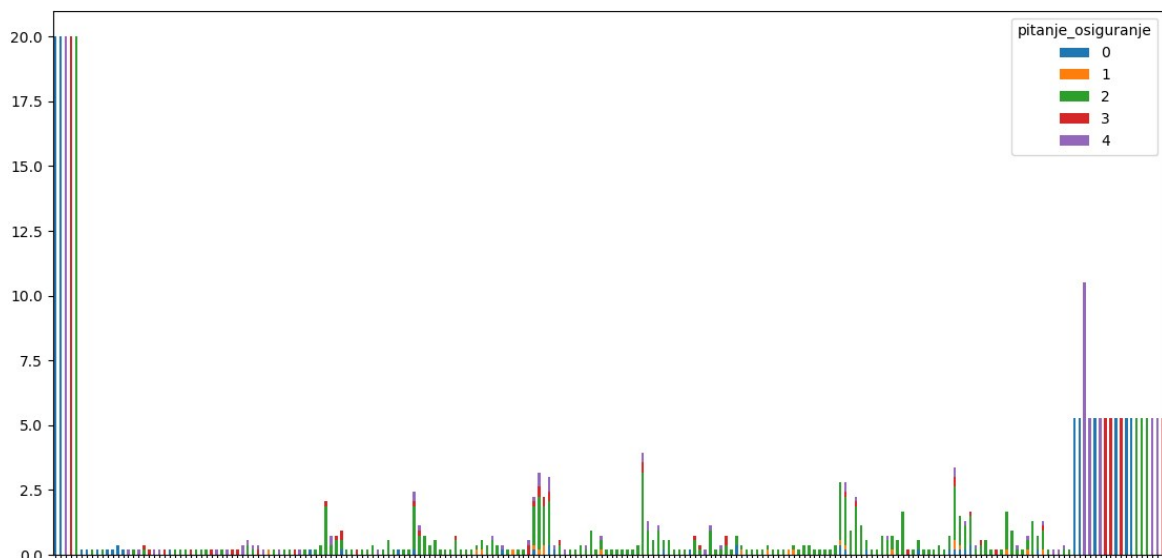
6.7.2 Povezivanje pitanja trećeg segmenta i pitanja opšteg tipa u jednu celinu

Sledi statistički pregled trećeg segmenta pitanja koji će sada u sebe uključiti pol ispitanika, nivo obrazovanja majke i oca, prosečnu ocenu na kraju prethodno završenog razreda i prosečnu ocenu iz matematike na kraju prethodno završenog razreda.

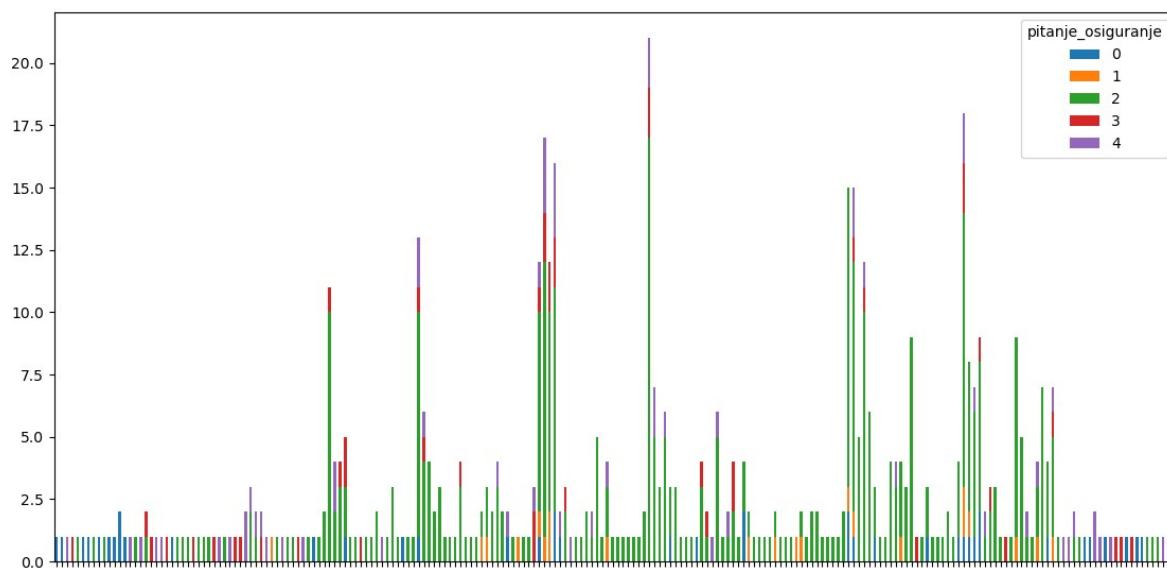
Slede slike uzajamne zavisnosti segmenta 3 ankete i opštih podataka. Razmatranja će biti data na osnovu tabela iz spred-šitova koje je kod kreirao.



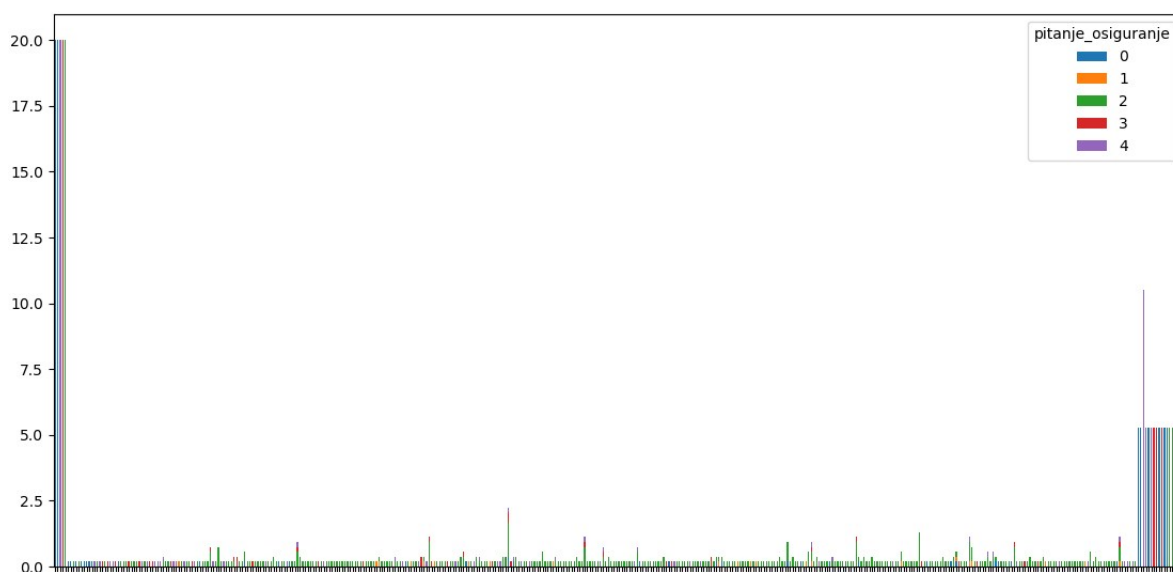
Slika 6.7.2.1. Pitanja trećeg segmenta povezana sa opštim pitanjima



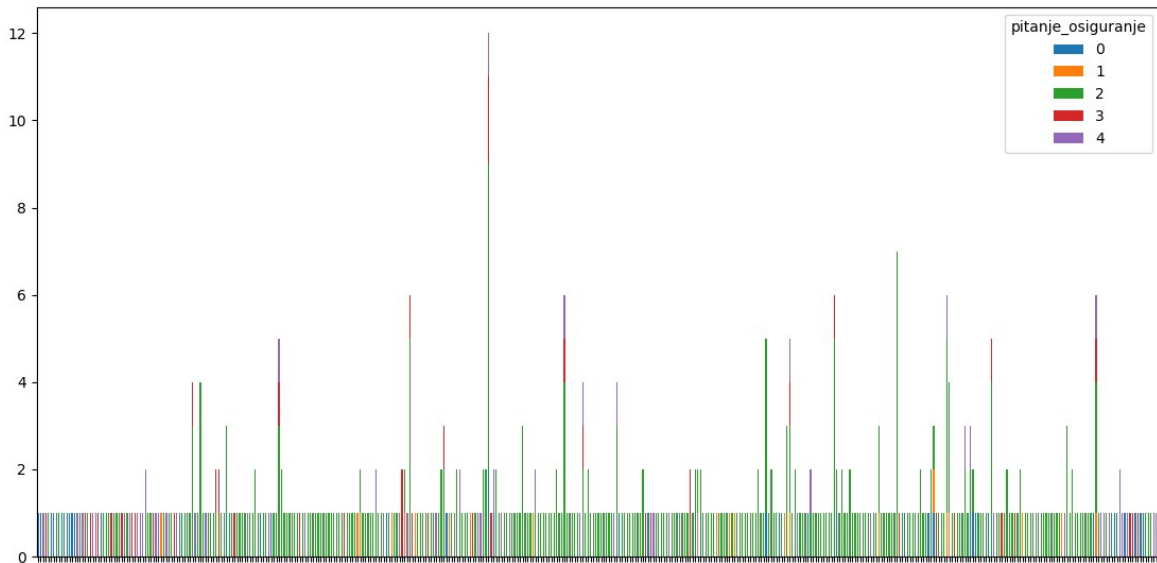
Slika 6.7.2.2. Pitanja trećeg segmenta povezana sa opštim pitanjima _procentualno



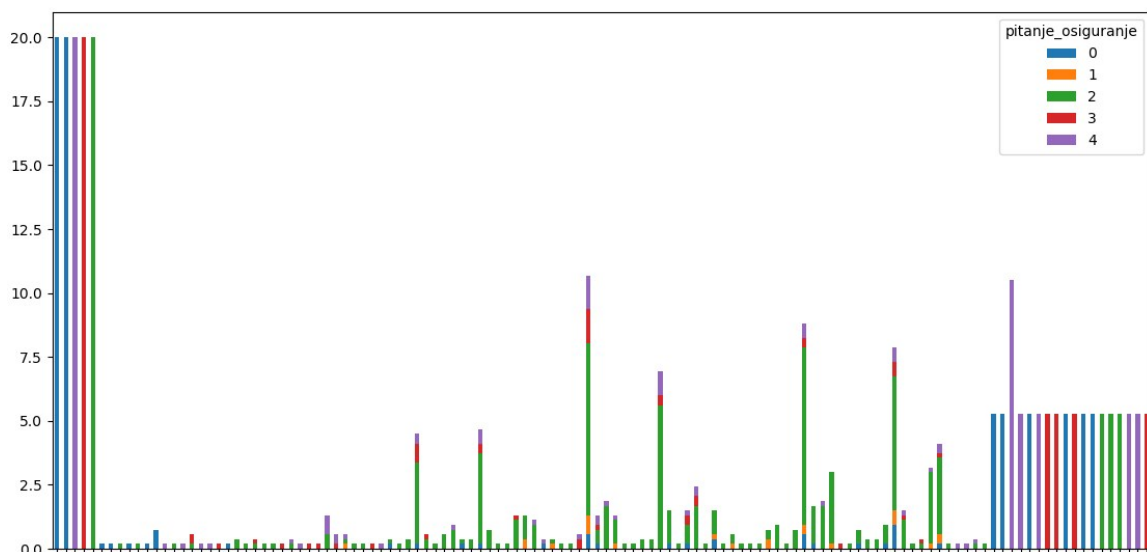
Slika 6.7.2.3. Pitanja trećeg segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)



Slika 6.7.2.4. Pitanja trećeg segmenta povezana sa opštim pitanjima_percentualno (bez pitanja za ukupnu prosečnu ocenu)



Slika 6.7.2.5. Pitanja trećeg segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)



Slika 6.7.2.6. Pitanja trećeg segmenta povezana sa opštim pitanjima procentualno (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Iz tabelarnog prikaza se može zaključiti da kolona *prosecna_ocena_prethodni_razred* predstavlja i u ovom slučaju opterećenje za ramatanje rezultata. Razlog tome je u činjenici da je opseg ocena od 2 do 5 (nije bilo nedovoljnih) kada je u pitanju tip podataka float veliki, tj. nema puno podudaranja prosečnih ocena. Iz navedene tabele se može zaključiti da je najveći stepen podudarnja kada je ukupna prosečna ocena na kraju godine bila pet. Sedmero ispitanika ženskog pola je dalo odgovor na pitanje o pojmu osiguranja

DA, ocenilo je svoje poznavanje pojma osiguranja četvorkom, dalo je tačan odgovor na pitanje o osiguranju, majka i otac su istog nivoa obrazovanja – srednjoškolskog. Pored toga, takođe je sedmero ispitanika ženskog pola dalo odgovor na pitanje o pojmu osiguranja DA, ocenilo je svoje poznavanje pojma osiguranja četvorkom, dalo je tačan odgovor na pitanje o osiguranju, majka i otac su istog nivoa obrazovanja – visokog. Što se tiče muških ispitanika njih četvoro je dalo odgovor na pitanje o pojmu osiguranja DA, ocenilo je svoje poznavanje pojma osiguranja četvorkom, dalo je tačan odgovor na pitanje o osiguranju, majka i otac su istog nivoa obrazovanja – srednjoškolskog .

Sa druge strane kolona *prosecna_ocena_matematika* je tipa int. tj. može imati celobrojne vrednosti između 2 i 5 (nije bilo nedovoljnih). Zato je u kodu kopirana linija koda koja daje sveobuhvatnu zavisnost opštih podataka i prvog segmenta pitanja, a zatim je iz te linije koda izbačena kolona *prosecna_ocena_prethodni_razred*. Iz dobijenih tabelarnih rezultata se može zaključiti da najveći broj ispitanika koji odgovorio da poznaje pojam osiguranja dolazi iz skupa koji u sebe obuhvata sledeće karakteristike:

- 17 ispitanika muškog pola je odgovorilo potvrdno pitanje o poznavanju pojma osiguranje i svoje poznavanje pojma osiguranja je ocenilo ocenom 4. Ovi ispitanici su odgovori tačno na postavljeno pitanje iz oblasti osiguranja. Nivo obrazovanja majke i oca je isti, tj. srednjoškolski, a prosečna ocena matematike iz prethodnog razreda je 2.
- 12 ispitanika ženskog pola je odgovorilo potvrdno pitanje o poznavanju pojma osiguranje i svoje poznavanje pojma osiguranje je ocenilo ocenom 5. Ovi ispitanici su odgovori tačno na zadatak iz prostog kamatnog računa. Nivo obrazovanja majke i oca je isti, tj. srednjoškolski, a prosečna ocena matematike iz prethodnog razreda je 2.

Ukoliko se izbaci i kolona koja se odnosi na prosečnu ocenu iz matematike (spred-šit Anketa26_rezultati) može se uočiti da je 37 ispitanika odgovorilo tačno pitanje iz oblasti osiguranja. Karakteristike takvog skupa su: ocena poznavanja pojma osiguranja 5, pol ženski, nivo obrazovanja majke srednjoškolski i nivo obrazovanja oca srednjoškolski. Ujedno ovaj skup ispitanika predstavlja skup sa najvećim brojem elemenata u datom razmatranju (odnosi se na sve odgovore u koloni pol) Što se tiče ispitanika muškog pola, njih 30 je odgovorilo tačno na pitanje iz oblasti osiguranja. Karakteristike takvog skupa su: ocena poznavanja pojma osiguranja 4, nivo obrazovanja majke srednjoškolski i nivo

obrazovanja oca srednjoškolski. Ujedno ovaj skup ispitanika predstavlja skup sa najvećim brojem elemenata u datom razmatranju (odnosi se na odgovore u koloni pol = „muški“). Pored toga ovaj skup je treći najbrojniji skup u datom razmatranju (odnosi se na sve odgovore u koloni pol).

6.8 Analiza četvrtog segmenta pitanja

U četvrtom segmentu pitanja se razmatra pojam studentskog kredita. Ovaj segment se sastoji od tri pitanja sa ponuđenim odgovorima. Ukoliko ispitanik na ponuđeno pitanje nije dao odgovor, takav odgovor na pitanje je vrednovan nulom. Sa druge strane ukoliko je ispitanik na ponuđeno pitanje zaokružio više odgovora, takav odgovor je vrednovan sa: broj ponuđenih odgovora +1.

Slede pitanja koja se nalaze u segmentu i način njihovog vrednovanja u Anketa.csv fajlu.

Pitanje 1, Segment 4 ankete

1. *Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam STUDENTSKI KREDIT?*

Odgovori: DA NE

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, DA, NE, 3

0 – nije dat odgovor

DA

NE

3 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na prvo pitanje je u Anketa.csv fajlu nazvana studentski_kredit.

Tabela 6.8.1. Udeo odgovora na Pitanje 1, Segment 4

<i>Pitanje 1, Segment 4</i>	0	DA	NE	3	Σ
	14	369	174	1	558
Udeo u %	2,51	66,13	31,18	0,18	100

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni `studentski_kredit`. Kako su podaci koji su uneti u kolonu `kamatna_stopa` tipa `string`, to će kao rezultat pokretanja funkcije `describe()` kao dela `pandas` paketa, dati frekvenciju odgovora koji se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spred-šit `Opis21.xlsx`. Iz tabele se vidi postoji ukupno 558 ispitanika, 4 jedinstvena odgovora (0, DA, NE, 3) i da je 369 ispitanika odgovorilo da je upoznato sa pojmom .

	A	B
1		studentski_kredit
2	count	558
3	unique	4
4	top	DA
5	freq	369

	A	B
1		studentski_kredit
2	count	558
3	mean	1.290322581
4	std	0.51016198
5	min	0
6	25%	1
7	50%	1
8	75%	2
9	max	3

Slika 6.8.1 Statistički podaci za kolonu `studentski_kredit` kada je u pitanju tip podataka `string` i `int`.

Da bi mogla da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je `string` prevesti u podatke tipa `int`. To je urađeno tako što je u `Anketa.csv` fajlu promenjena vrednost, pa je: 0=0, DA=1, NE=2, 3=3. Podaci su eksportovani u spred-šit `Opis22.xlsx`. Iz sred-šita `Opis22.xlsx` se vidi da je aritmetička sredina približno 1,3. Razlog leži u činjici da je 31,18 [%] ispitanika odgovorilo negativno na pitanje. Ovaj podatak je se može opravdati činjenicom da određeni deo populacije koja je ispitivana neće upisivati studije, pa se nije ni raspitalo o uslovima studiranja. Na osnovu podataka koje je obradio Republički zavod za statistiku Republike Srbije [83] za visoko obrazovanje u školskoj 2017/2018 godini i podataka koje je isti zavod obradio za srednjoškolsko obrazovanje za kraj školske 2016/2017 godine [84], može se utvrditi

odnos broja svršenih srednjoškolaca četvrtog stepena koji su upisali fakultet u odnosu na ukupan broj svršenih srednjoškolaca četvrtog stepena. Taj odnos je računat po formuli:

Broj upisanih srednjoškolaca na fakultet

$$= \frac{\text{Ukupan broj svršenih srednjoškolaca} * 0,893}{\text{Broj upisanih srednjoškolaca na fakultet}} * 100[\%]$$

$$\text{Broj upisanih srednjoškolaca na fakultet} = \frac{58948 * 0,893}{56312} * 100[\%]$$

$$\text{Broj upisanih srednjoškolaca na fakultet} = 93,48 [\%]$$

U prvoj formuli koja se bavi brojem srednjoškolaca upisanih na fakultet, broj 0,893 [84] predstavlja procenat srednjoškolaca koji su završili četvorogodišnje srednjoškolsko obrazovanje i koji imaju pravo na upis na fakultet. Kako Republički zavod za statistiku republike Srbije nije izdao publikaciju za kraj školske 2017/2018 godine za srednje obrazovanje, kao ni publikaciju za početak školske 2018/2019 godine za visoko obrazovanje, procenat koji je dobijen u gornjoj formuli će biti prihvaćen i razmatran u okviru doktorske disertacije.

Podatak da je na prvo pitanje četvrtog segmenta 31,18 [%] ispitanika odgovorilo da nije upoznato sa pojmom studentski kredit, a da je 93,48 [%] srednjoškolaca upisalo fakultet, navodi na zaključak da ova tema nije dovoljno obrađena ni u školama ni u medijumima.

Pitanje 2, Segment 4 ankete

2. *Na osnovu Vašeg mišljenja, na skali od 1 do 5, ocenite u kojoj meri ste upoznati sa pojmom STUDENTSKI KREDIT?*

Odgovori: 5-odlično, 4-vrlo dobro, 3-dobro, 2-dovoljno, 1-nisam siguran/na

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, 1, 2, 3, 4, 5, 6

0 – nije dat odgovor

1 – nisam siguran/na

2 – dovoljno

3 – dobro

4 – vrlo dobro

5 – odlično

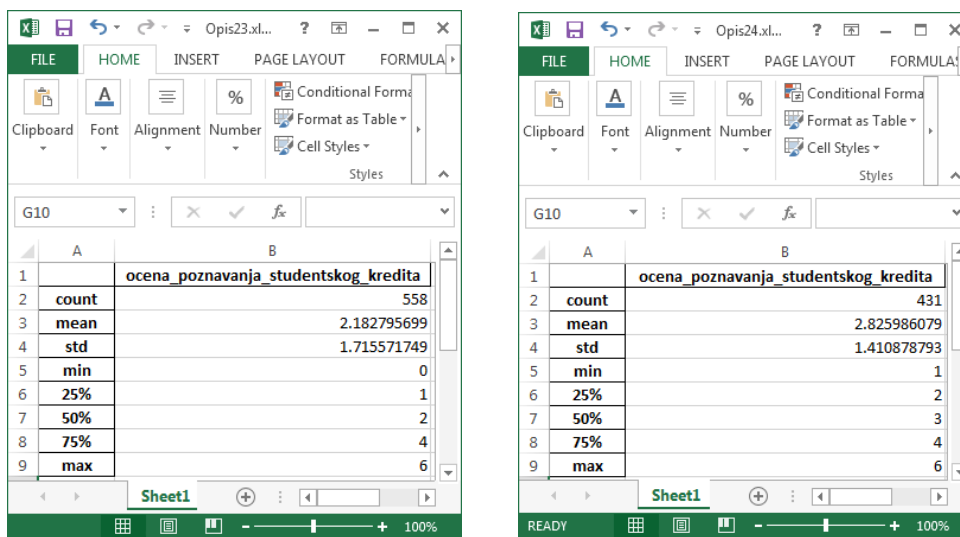
6 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na drugo pitanje je u Anketa.csv fajlu nazvana *ocena_poznavanja_studentskog_kredita*.

Tabela 6.8.2. Udeo odgovora na Pitanje 2, Segment 4

Pitanje 2,	0	1	2	3	4	5	6	Σ
Segment 4	127	98	98	93	68	71	3	558
Udeo u %	22,76	17,56	17,56	16,67	12,19	12,72	0,54	100,00

Podaci su eksportovani u spred-šit Opis23.xlsx. Na osnovu slike koja sledi može se zaključiti da je prosečna ocena poznavanja pojma studentskog kredita (uključujući i pitanja na koja nije dat odgovor, kao i pitanja na koja je dato više odgovora) $2,18 \approx 2,2$, tj. opisna ocena je *dovoljan*. Ova ocena u skladu sa prvim pitanjem ovog segmenta, jer veliki broj ispitanika nije čuo za pojam studentski kredi. Ukoliko bi se izbacile opcije iz ankete ko nije odgovorio na pitanje iz kolone *ocena_poznavanja_studentskog_kredita* dobila bi se ocena 2.82, tj. 431 ispitanik je dao odgovor na postavljeno pitanje. Podaci su eksportovani u spred-šit Opis24.xlsx. Ocena 2,82 je blizu ocene 3 (*dobro*), ali i ovako dobijena ocena ukazuje na nedovoljnu informisanost učenika.



Slika 6.8.2 Statistički podaci za kolonu *ocena_poznavanja_studentskog_kredita*

Pitanje 3, Segment 4 ankete

3. Ukoliko pretpostavimo da ste od banke uzeli studentski kredit, to podrazumeva da koristite:

Odgovori: a-Bespovratna finansijska sredstva tokom čitavog perioda studiranja, b – Bespovratna finansijska sredstva tokom prve godine studiranja, c-Pozajmljena finasijska sredstva sa ugovornim periodom otplate u više mesečnih rata, d-ne znam

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, a, b, c, d, 5

0 – nije dat odgovor

a – Bespovratna finansijska sredstva tokom čitavog perioda studiranja

b – Bespovratna finansijska sredstva tokom prve godine studiranja

c – Pozajmljena finasijska sredstva sa ugovornim periodom otplate u više mesečnih rata

d – ne znam

5 – dato više odgovora

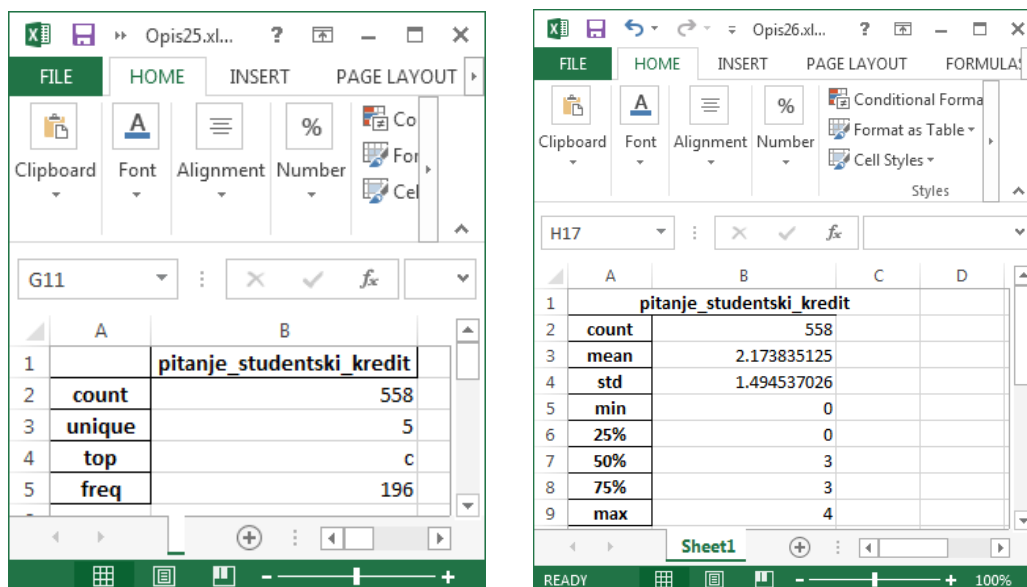
Kolona u koju su smešteni rezultati odgovora na treće pitanje je u Anketa1.csv fajlu nazvana pitanje_studentski_kredit.

Tabela 6.8.3. Udeo odgovora na Pitanje 3, Segment 4

<i>Pitanje 3,</i>	0	a	b	c	d	5	Σ
<i>Segment 4</i>	141	45	62	196	114	0	558
Udeo u %	25,27	8,06	11,11	35,13	20,43	0	100

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni pitanje_studentski_kredit. Kako su podaci koji su uneti u kolonu pitanje_inflacija tipa string, to će kao rezultat pokretanja funkcije describe() kao dela pandas paketa, dati frekvenciju odgovora koji se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spred-šit Opis25.xlsx. Sa slike koja sledi, vidi se da je broj ponuđenih odgovora 5, a da je odgovor pod c zaokružilo 196 ispitanika.

Da bi mogla da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je string prevesti u podatke tipa int. To je urađeno tako što je u Anketa.csv fajlu promenjena vrednost, pa je: 0=0, a=1, b=2, c=3, d=4, 5=5. Podaci su eksportovani u spred-šit Opis26.xlsx. Sa slike koja sledi se može uočiti da je najčešći odgovor koji je dobijen c (3) i da je aritmetička sredina približno 2,18.



Slika 6.8.3. Statistički podaci za kolonu pitanje_studentski_kredit kada je u pitanju tip podataka string i int.

Konkurs za studentske kredite može raspisati ministarstvo prosvete, nauke i tehnološkog razvoja. Za školsku 2017/2018 godinu [85] se vidi da u članu 5 ovog konkursa [85] postoji mogućnost potpunog oslobađanja od vraćanja studentskog kredita. Jedan od uslova je da je prosečna ocena na kraju studiranja veća od 8,50. Na sajtu Unicredit banke [86], takođe, stoji konkurs za studentske kredite za školsku 2018/2019 godinu koje je ministarstvo raspisalo. Prema ovom izvoru [86] uslovi za oslobađanje vraćanja studentskog kredita su isti kao i školske 2017/2018 godine [85].

Sa druge strane postoje i komercijalni studentski krediti [87] koji se ne oslobađaju vraćanja. Prema izvoru [83] za 2017/2018 školsku godinu, svega 110 studenata od ukupno diplomiranih 45406 studenata je uzelo komercijalni studentski kredit tokom studiranja, odnosno 0,24 [%]. Interesantno je da je od tih 110 diplomiranih studenata njih 83 završilo fakultete i akademije umetnosti [83], odnosno 75,45 [%]. Udeo muškaraca u tih 110 diplomiranih studenata je 42 studenta [83] ili 38,18 [%], a udeo studentkinja je 68 ili 61,82 [%].

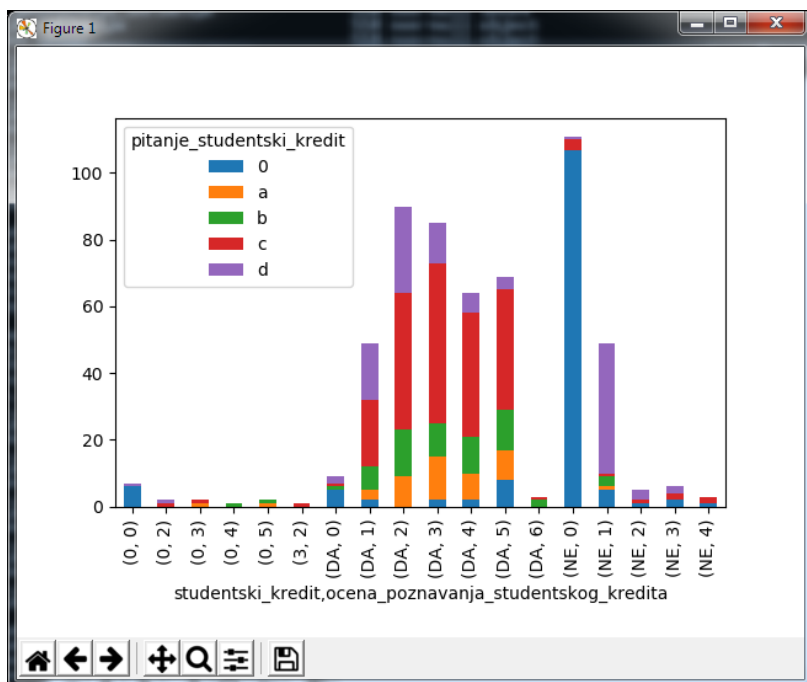
Na osnovu publikacije [83] se može zaključiti da je 56,35 [%] diplomiranih studenata tokom svog studiranja izdržavano lice, a da se 23,73 [%] diplomiranih studenata tokom svog studiranja izdržava samostalno.

Data analiza upućuje na odgovor zašto je mali broj studenata odgovorio pozitivno na pitanje o načinu vraćanja komercijalnog studentskog kredita.

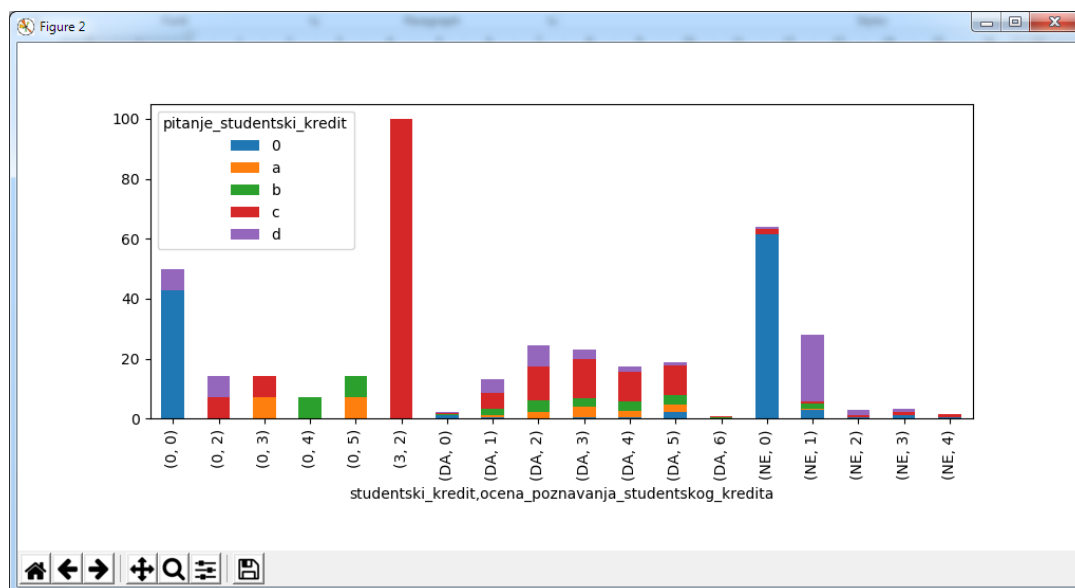
6.8.1 Povezivanje pitanja četvrtog segmenta u jednu celinu

Sledi statistički pregled četvrtog segmenta pitanja koji će sada u sebe uključiti pol ispitanika, nivo obrazovanja majke i oca, prosečnu ocenu na kraju prethodno završenog razreda i prosečnu ocenu iz matematike na kraju prethodno završenog razreda.

Slede slike uzajamne zavisnosti segmenta 4 ankete i opštih podataka. Razmatranja će biti data na osnovu tabela iz spred-šitova koje je kod kreirao.



Slika 6.8.1.1. Grafik odnosa pitanja četvrtog segmenta ankete



Slika 6.8.1.2. Grafik odnosa pitanja četvrtog segmenta ankete

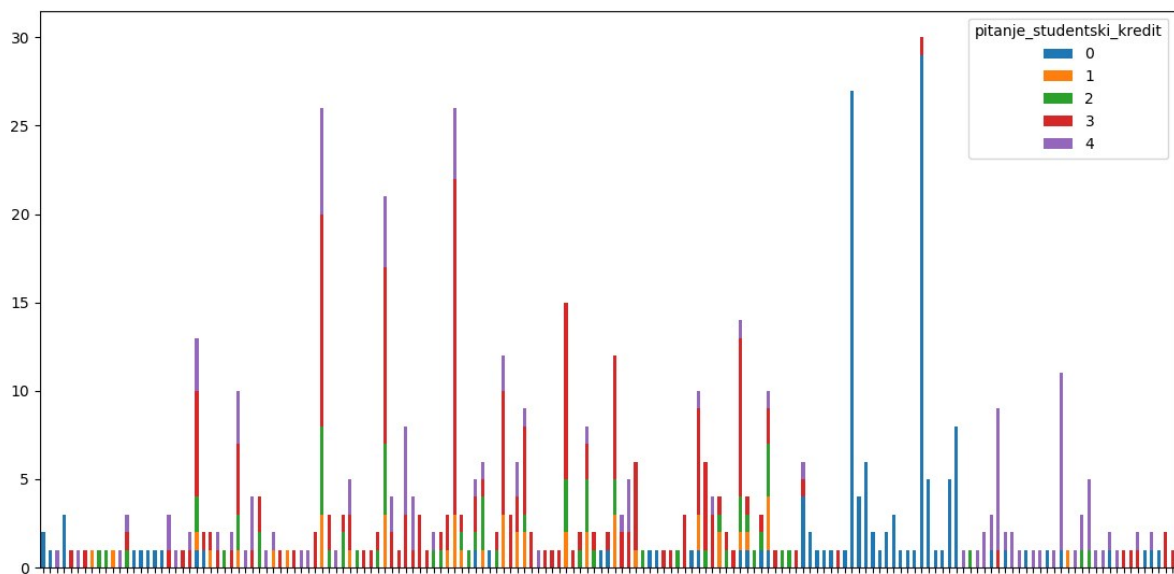
	A	B	C	D	E	F	G	H
1	studentski_kredit	ocena_poznavanja_studentskog_kredita	0	a	b	c	d	Ukupno_redovi
2	0	0	6	0	0	0	1	7
3		2	0	0	0	1	1	2
4		3	0	1	0	1	0	2
5		4	0	0	1	0	0	1
6		5	0	1	1	0	0	2
7		2	0	0	0	1	0	1
8	DA	0	5	0	1	1	2	9
9		1	2	3	7	20	17	49
10		2	0	9	14	41	26	90
11		3	2	13	10	48	12	85
12		4	2	8	11	37	6	64
13		5	8	9	12	36	4	69
14	NE	6	0	0	2	1	0	3
15		0	107	0	0	3	1	111
16		1	5	1	3	1	39	49
17		2	1	0	0	1	3	5
18		3	2	0	0	2	2	6
19	4	1	0	0	2	0	3	
20	Ukupno_kolone		141	45	62	196	114	558
21							558	

Slika 6.8.1.3. Podaci eksportovani u spred-šit

Iz spred-šita Anкета28_rezultati.xlsx se može zaključiti da je najveći broj ispitanika (njih 48 ili 8,6 [%] od ukupnog broja ispitanika) odgovorilo da poznaje pojam studentski kredit dajući sebi ocenu tri. Ovaj broj ispitanika je ujedno i tačno odgovorio na postavljeno pitanje (odgovor c). Vrlo blizu ovog rezultata se nalazi rezultat u redosledu: DA, 2, c i on iznosi 7,35 [%] ili 41 ispitanika. Ukoliko se pogledaju ostale zavisnosti prikazane u gornjoj tabeli može se zaključiti da je samovrednovanje ispitanika na granici objektivnosti, jer od broja ispitanika koji su dali odgovor na pitanje o oceni poznavanja osiguranja 3,4 ili 5 tačan odgovor je dalo 55,50 [%] ili od 218 koji su dali odgovor na pitanje o oceni poznavanja osiguranja 3,4 ili 5 tačan odgovor je dao 121 ispitanik.

6.8.2 Povezivanje pitanja četvrtog segmenta i pitanja opšteg tipa u jednu celinu

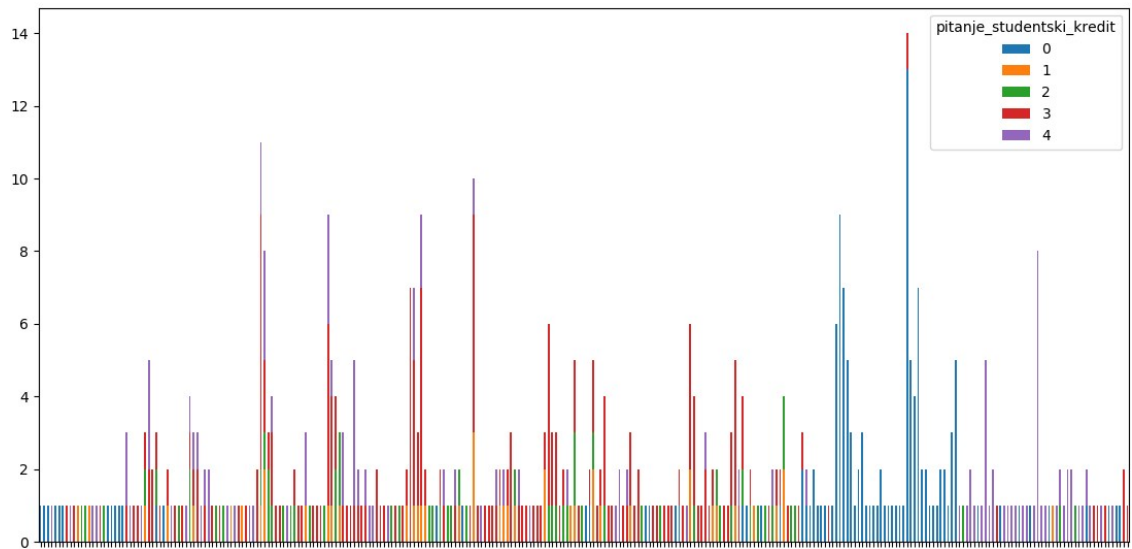
Sledi statistički pregled četvrtog segmenta pitanja koji će sada u sebe uključiti pol ispitanika, nivo obrazovanja majke i oca, prosečnu ocenu na kraju prethodno završenog razreda i prosečnu ocenu iz matematike na kraju prethodno završenog razreda. Slede slike uzajamne zavisnosti segmenta 4 ankete i opštih podataka. Razmatranja će biti data na osnovu tabela iz spred-šitova koje je kod kreirao.



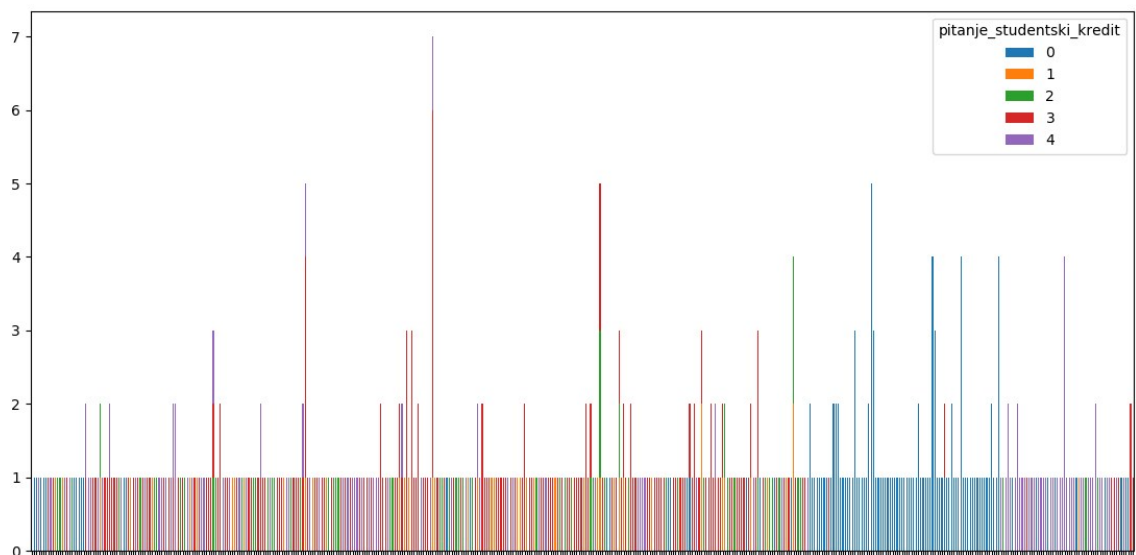
Slika 6.8.2.1. Pitanja četvrtog segmenta povezana sa opštim pitanjima



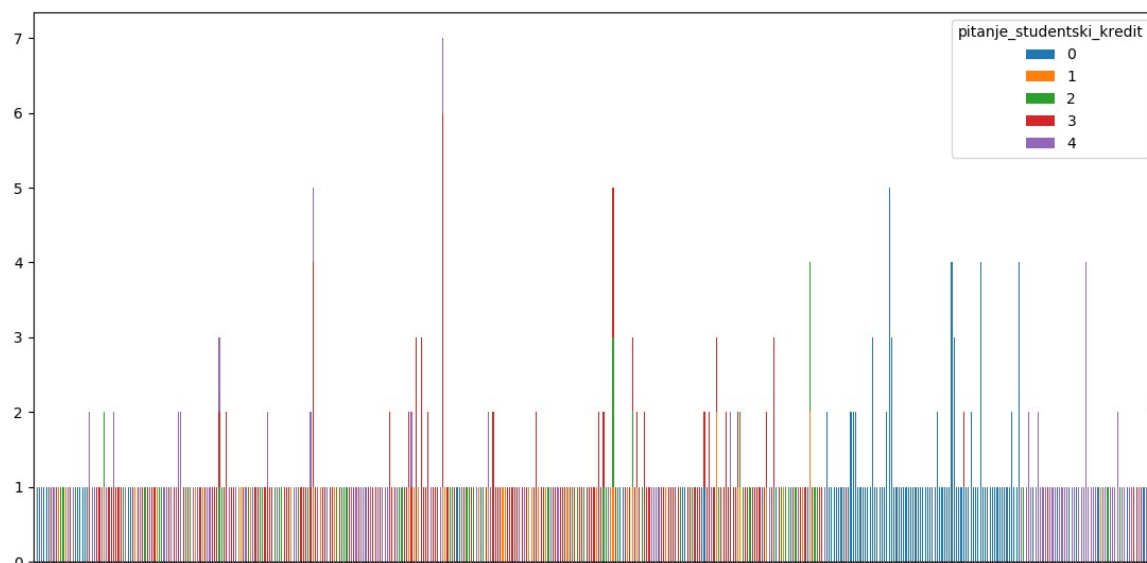
Slika 6.8.2.2. Pitanja četvrtog segmenta povezana sa opštim pitanjima_procentualno



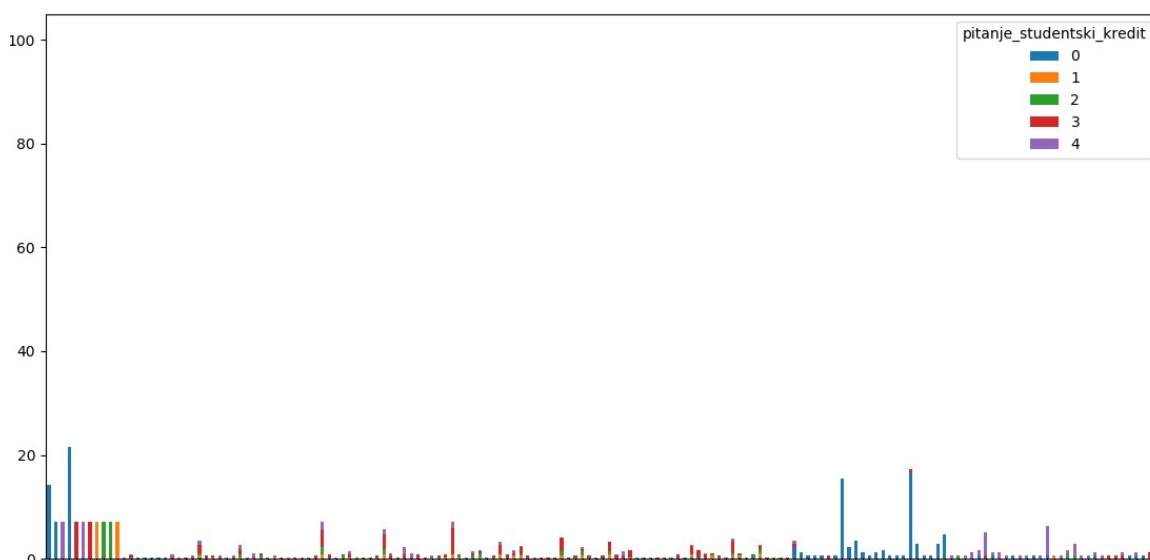
Slika 6.8.2.3. Pitanja četvrtog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)



Slika 6.8.2.4. Pitanja četvrtog segmenta povezana sa opštim pitanjima procentualno (bez pitanja za ukupnu prosečnu ocenu)



Slika 6.8.2.5. Pitanja četvrtog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)



Slika 6.8.2.6. Pitanja četvrtog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Tabelarni prikaz se nalazi u dodatku rada kao fajl *Anketa30_rezultati*. Iz datog tabelarnog prikaza se može zaključiti da kolona *prosecna_ocena_prethodni_razred* predstavlja i u ovom slučaju opterećenje za ramatranje rezultata. Razlog tome je u činjenici da je opseg ocena od 2 do 5 (nije bilo nedovoljnih) kada je u pitanju tip podataka float veliki, tj. nema puno podudaranja prosečnih ocena. Iz navedene tabele se može zaključiti da je najveći stepen podudarnja kada je ukupna prosečna ocena na kraju godine

bila pet. Sedmero ispitanika ženskog pola je dalo odgovor na pitanje o pojmu studentskog kredita DA, ocenilo je svoje poznavanje pojma studentskog kredita ocenom 3, dalo je tačan odgovor na pitanje o osigiranju njih pet, majka i otac su istog nivoa obrazovanja – srednjoškolskog.

Sa druge strane kolona *prosecna_ocena_matematika* je tipa int. tj. može imati celobrojne vrednosti između 2 i 5 (nije bilo nedovoljnih). Zato je u kodu kopirana linija koda koja daje sveobuhvatnu zavisnost opštih podataka i prvog segmenta pitanja, a zatim je iz te linije koda izbačena kolona *prosecna_ocena_prethodni_razred*. Iz dobijenih tabelarnih rezultata se može zaključiti da najveći broj ispitanika koji odgovorio da poznaje pojam osiguranja dolazi iz skupa koji u sebe obuhvata sledeće karakteristike:

- 6 ispitanika muškog pola je odgovorilo potvrdno pitanje o poznavanju pojma osiguranje i svoje poznavanje pojma osiguranja je ocenilo ocenom 2. Ovi ispitanici su odgovori tačno na postavljeno pitanje iz oblasti osiguranja. Nivo obrazovanja majke i oca je isti, tj. srednjoškolski, a prosečna ocena matematike iz prethodnog razreda je 2.
- 7 ispitanika ženskog pola je odgovorilo potvrdno pitanje o poznavanju pojma osiguranje i svoje poznavanje pojma osiguranje je ocenilo ocenom 5. Ovi ispitanici su odgovori tačno na zadatak iz prostog kamatnog računa. Nivo obrazovanja majke i oca je isti, tj. srednjoškolski, a prosečna ocena matematike iz prethodnog razreda je 2.

Ukoliko se izbaci i kolona koja se odnosi na prosečnu ocenu iz matematike (spred-šit Anketa34_rezultati) može se uočiti da je 19 ispitanika odgovorilo tačno pitanje iz studentskog kredita. Karakteristike takvog skupa su: ocena poznavanja pojma studentskog kredita 3, pol ženski, nivo obrazovanja majke srednjoškolski i nivo obrazovanja oca srednjoškolski. Ujedno ovaj skup ispitanika predstavlja skup sa najvećim brojem elemenata u datom razmatranju (odnosi se na sve odgovore u koloni pol).

Što se tiče ispitanika muškog pola, njih 10 je odgovorilo tačno na pitanje iz oblasti studentskog kredita. Karakteristike takvog skupa su: ocena poznavanja pojma studentskog kredita 2, nivo obrazovanja majke srednjoškolski i nivo obrazovanja oca srednjoškolski. Ujedno ovaj skup ispitanika predstavlja skup sa najvećim brojem elemenata u datom razmatranju (odnosi se na odgovore u koloni pol = „muški“).

6.9 Analiza petog segmenta pitanja

U petom segmentu pitanja se razmatra pojam kriptovaluta. Ovo je pojam o kome se poslednjih godina može često čuti ili pročitati. Većina podataka koja je dostupna o kriptovalutama na najpopularnijim sajtovima u Srbiji (sajtovi dnevnopolitičkih novina) je razmatrana nestručno. Akcenat je u većini slučajeva stavljan na to ko je kako potrošio novac od prodaje kriptovaluta (prvenstveno se misli na BitKoin, eng. BitCoin), ne ulazeći u samo pojašnjenje o tome šta je, na čemu se zasniva i koje su pogodnosti kriptovaluta. Ovaj segment se sastoji od tri pitanja sa ponuđenim odgovorima. Ukoliko ispitanik na ponuđeno pitanje nije dao odgovor, takav odgovor na pitanje je vrednovan nulom. Sa druge strane ukoliko je ispitanik na ponuđeno pitanje zaokružio više odgovora, takav odgovor je vrednovan sa: broj ponuđenih odgovora +1.

Slede pitanja koja se nalaze u segmentu i način njihovog vrednovanja u Anketa.csv fajlu.

Pitanje 1, Segment 5 ankete

1. *Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam KRIPTOVALUTA (eng. Cryptocurrency)?*

Odgovori: DA NE

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, DA, NE, 3

0 – nije dat odgovor

DA

NE

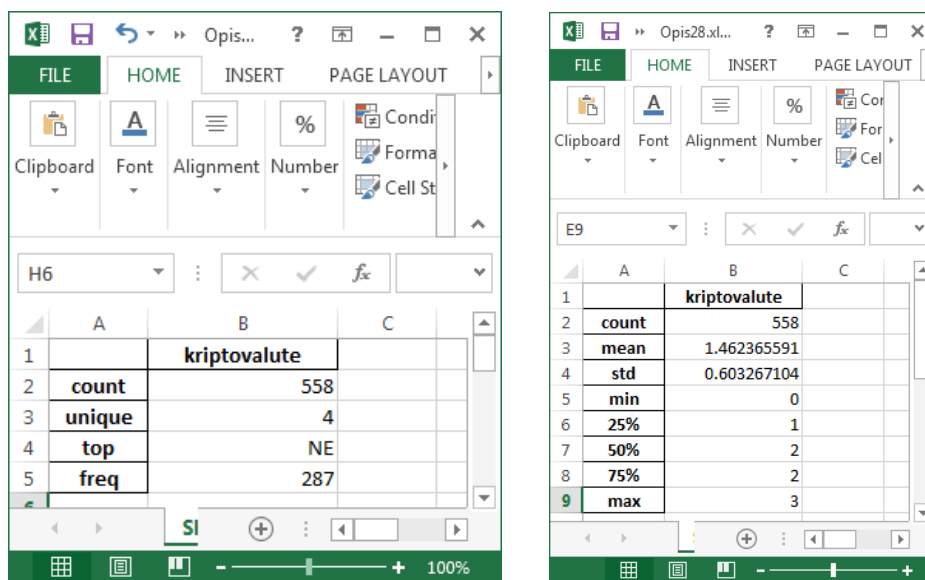
3 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na prvo pitanje je u Anketa.csv fajlu nazvana kriptovalute.

Tabela 6.9.1. Udeo odgovora na Pitanje 1, Segment 5

<i>Pitanje 1, Segment 5</i>	0	DA	NE	3	Σ
	31	239	287	1	558
Udeo u procentima	5,56	42,83	51,43	0,18	100

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni kriptovalute. Kako su podaci koji su uneti u kolonu kamatna_stopa tipa string, to će kao rezultat pokretanja funkcije describe() kao dela pandas paketa, dati frekvenciju odgovora koji se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spređ-šit Opis27.xlsx. Iz tabele se vidi postoji ukupno 558 ispitanika, 4 jedinstvena odgovora (0, DA, NE, 3) i da je 287 ispitanika odgovorilo da je upoznato sa pojmom . Ako se pogledaju sva prva pitanja svih pet segmenata ankete (ne uzimajući u obzir opšte podatke ispitanika) može se zaključiti da je ovo jedino prvo pitanje iz svih segmenata koji ima više negativnih nego pozitivnih odgovora.



Slika 6.9.1. Statistički podaci za kolonu pitanje_studentski_kredit kada je u pitanju tip podataka string i int.

Pitanje 2, Segment 5 ankete

2. Na osnovu Vašeg mišljenja, na skali od 1 do 5, ocenite u kojoj meri ste upoznati sa pojmom **KRIPTOVALUTA**?

Odgovori: 5-odlično, 4-vrlo dobro, 3-dobro, 2-dovoljno, 1-nisam siguran/na

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, 1, 2, 3, 4, 5, 6

0 – nije dat odgovor

1 – nisam siguran/na

2 – dovoljno

3 – dobro

4 – vrlo dobro

5 – odlično

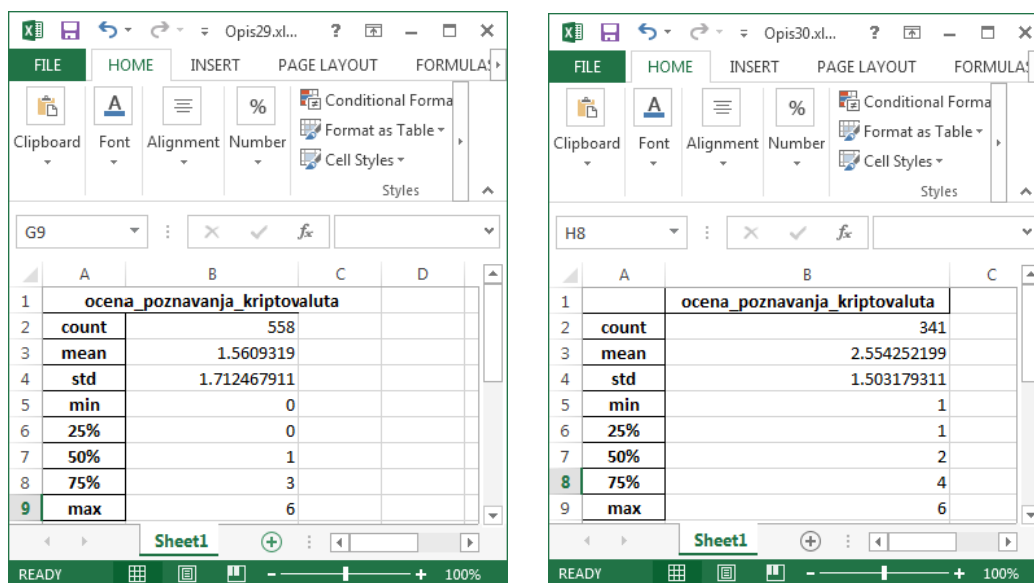
6 – dato je više odgovora

Kolona u koju su smešteni rezultati odgovora na drugo pitanje je u Anketa.csv fajlu nazvana ocena_poznavanja_kriptoaluta.

Tabela 6.9.2. Udeo odgovora na Pitanje 2, Segment 5

Pitanje 2,	0	1	2	3	4	5	6	Σ
Segment 5	217	130	48	59	56	44	4	558
Udeo u %	38,89	23,3	8,6	10,57	10,04	7,89	0,72	100,01

Podaci su eksportovani u spređ-šit Opis29.xlsx. Na osnovu slike koja sledi može se zaključiti da je prosečna ocena poznavanja pojma studentskog kredita (uključujući i pitanja na koja nije dat odgovor, kao i pitanja na koja je dato više odgovora) $1,56 \approx 1,6$, tj. opisna ocena je *nedovoljan*. Ova ocena u skladu sa prvim pitanjem ovog segmenta, jer veliki broj ispitanika nije čuo za pojam kriptoaluta. Ukoliko bi se izbacile opcije iz ankete ko nije odgovorio na pitanje iz kolone ocena_poznavanja_kriptoaluta dobila bi se ocena 2.55, tj. 341 ispitanik je dao odgovor na postavljeno pitanje. Podaci su eksportovani u spređ-šit Opis30.xlsx. Ocena 2,55 ukazuje na nedovoljnu informisanost učenika o pojmu kriptoaluta.



Slika 6.9.2 Statistički podaci za kolonu ocena_poznavanja_kriptoaluta

Pitanje 3, Segment 5 ankete

4. *Ukoliko pretpostavimo da imate određeni iznos sopstvenih finansijskih sredstava koje želite da investirate, kako biste opisali mogućnost investiranja u kriptovalute?*

Odgovori: a-Niska stopa zarade, ali i investicija niskog rizika, b-Veliki potencijal zarade, ali i investicija visokog rizika, c- Veliki potencijal zarade, sa umerenom stopom rizika, d- ne znam

Ćelije u Anketa.csv fajlu su vrednovane sa: 0, a, b, c, d, 5

0 – nije dat odgovor

a – Niska stopa zarade, ali i investicija niskog rizika

b – Veliki potencijal zarade, ali i investicija visokog rizika

c – Veliki potencijal zarade, sa umerenom stopom rizika

d – ne znam

5 – dato više odgovora

Kolona u koju su smešteni rezultati odgovora na treće pitanje je u Anketa.csv fajlu nazvana pitanje_kriptovalute.

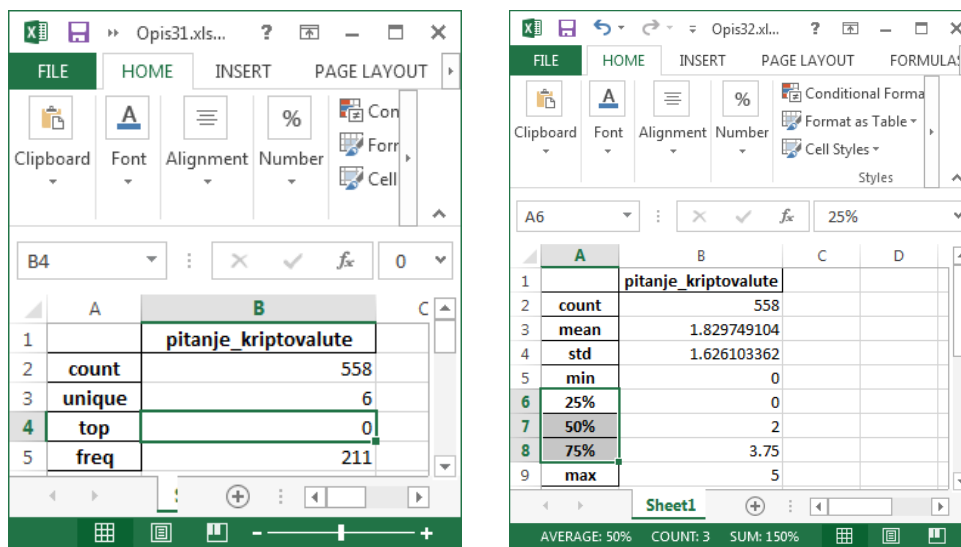
Tabela 6.9.3. Udeo odgovara na Pitanje 3, Segment 5

<i>Pitanje 3,</i>	0	a	b	c	d	5	Σ
<i>Segment 5</i>	211	17	127	63	139	1	558
Udeo u %	37,81	3,05	22,76	11,29	24,91	0,18	100

Sledi tabela koja je dobijena implementacijom razvijenog koda, a koja daje osnovne statističke podatke o koloni pitanje_kriptovalute. Kako su podaci koji su uneti u kolonu pitanje_kriptovalute tipa string, to će kao rezultat pokretanja funkcije describe() kao dela pandas paketa, dati frekvenciju odgovora koji se najčešće ponavlja. Pored toga biće dati i: ukupan broj ispitanika i ukupan broj jedinstvenih karaktera koji se javljaju u ispitivanoj koloni. Podaci su eksportovani u spred-šit Opis31.xlsx. Sa slike koja sledi, vidi se da je broj ponuđenih odgovora 6, a da najveći broj ispitanika nije zaokružio ništa.

Da bi mogla da se odredi aritmetička sredina, standardna devijacija i percentili, potrebno je tip podataka koji je string prevesti u podatke tipa int. To je urađeno tako što je u Anketa.csv fajlu promenjena vrednost, pa je: 0=0, a=1, b=2, c=3, d=4, 5=5. Podaci su eksportovani u spred-šit Opis32.xlsx. Sa slike koja sledi se može uočiti da je aritmetička sredina približno 1,83, što je najniža ocena za sva pitanja na trećem mestu u svim

segmentima. Ova aritmetička sredina opravdava odgovore prvog i drugog pitanja segmenta broj pet.

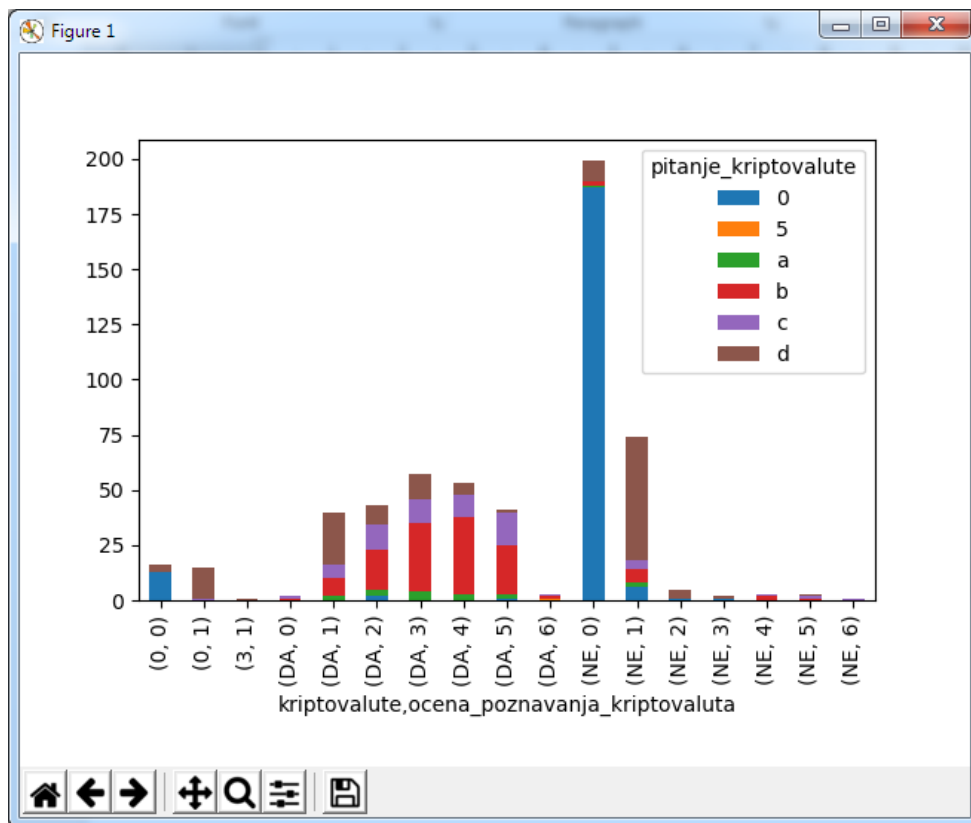


Slika 6.9.3. Statistički podaci za kolonu pitanje_kriptoalute kada je u pitanju tip podataka string i int.

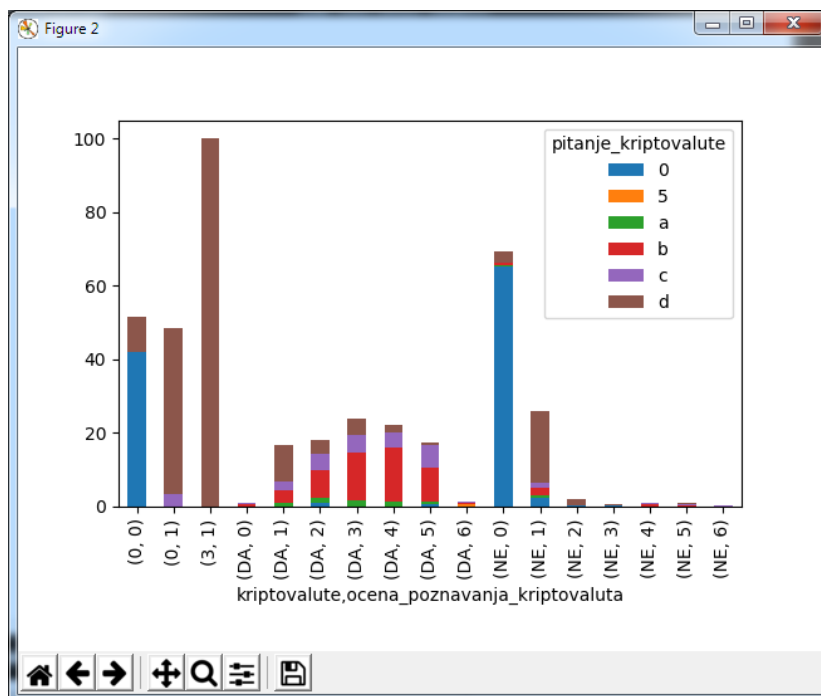
6.9.1 Povezivanje pitanja petog segmenta u jednu celinu

Sledi statistički pregled petog segmenta pitanja koji će sada u sebe uključiti pol ispitanika, nivo obrazovanja majke i oca, prosečnu ocenu na kraju prethodno završenog razreda i prosečnu ocenu iz matematike na kraju prethodno završenog razreda.

Na sledećim slikama je data međusobna povezanost prva tri pitanja. Grafici koji slede su interaktivni i u svakom trenutku se može dobiti odgovarajuća zavisnost između veličina. Kod sadrži komande koje omogućavaju predstavljanje zavisnosti u tabelarnom prikazu, tj. u .xlsx formatu, pa ispod svakog grafika sledi i taberni prikaz rezultata.



Slika 6.9.1.1. Grafik odnosa pitanja petog segmenta ankete



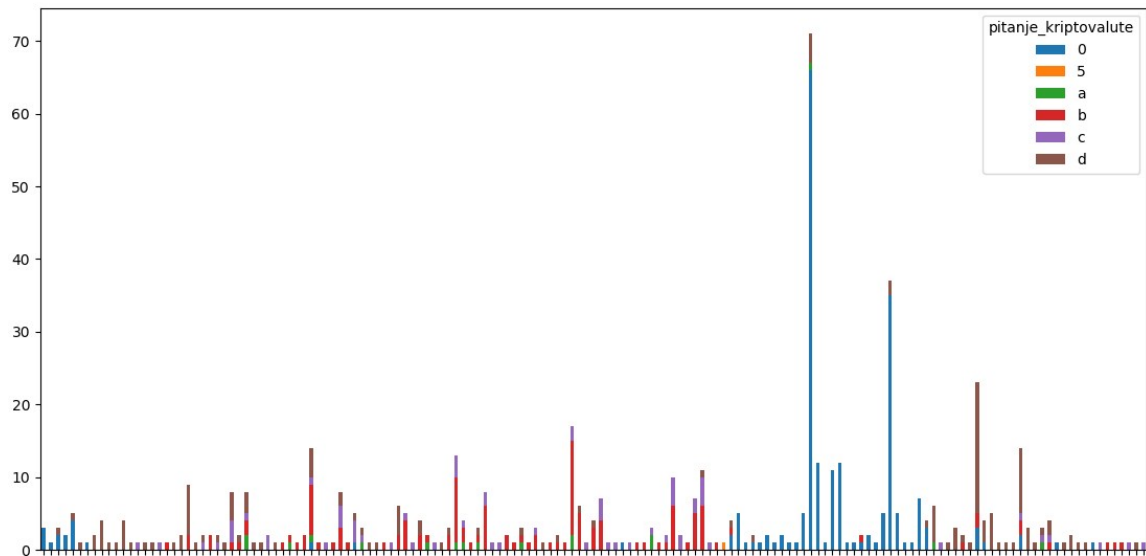
Slika 6.9.1.2. Grafik odnosa pitanja petog segmenta ankete_procentualno

The screenshot shows an Excel spreadsheet with the following data:

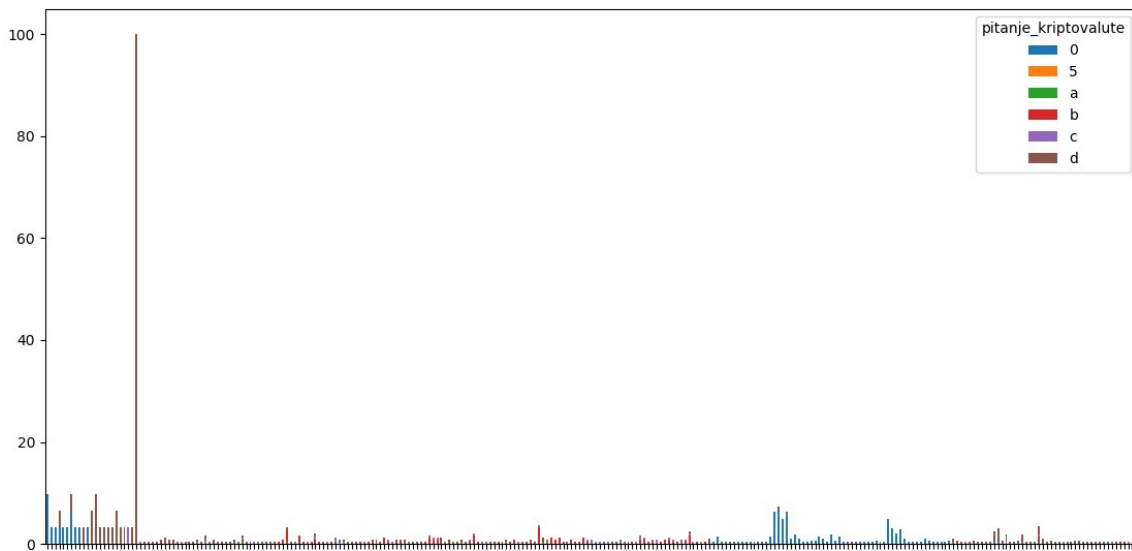
	A	B	C	D	E	F	G	H	I
1	kriptovalute	ocena_poznavanja_kriptoaluta	0	5	a	b	c	d	Ukupno_redovi
2	0	0	13	0	0	0	0	3	16
3	1	1	0	0	0	0	1	14	15
4	3	1	0	0	0	0	0	1	1
5	DA	0	0	0	0	1	1	0	2
6		1	0	0	2	8	6	24	40
7		2	2	0	3	18	11	9	43
8		3	0	0	4	31	11	11	57
9		4	0	0	3	35	10	5	53
10		5	1	0	2	22	15	1	41
11		6	0	1	0	1	1	0	3
12	NE	0	187	0	1	2	0	9	199
13		1	6	0	2	6	4	56	74
14		2	1	0	0	0	0	4	5
15		3	1	0	0	0	0	1	2
16		4	0	0	0	2	1	0	3
17		5	0	0	0	1	1	1	3
18		6	0	0	0	0	1	0	1
19	Ukupno_kolone		211	1	17	127	63	139	558
20								558	

Slika 6.9.1.3. Podaci eksportovani u sred-šit

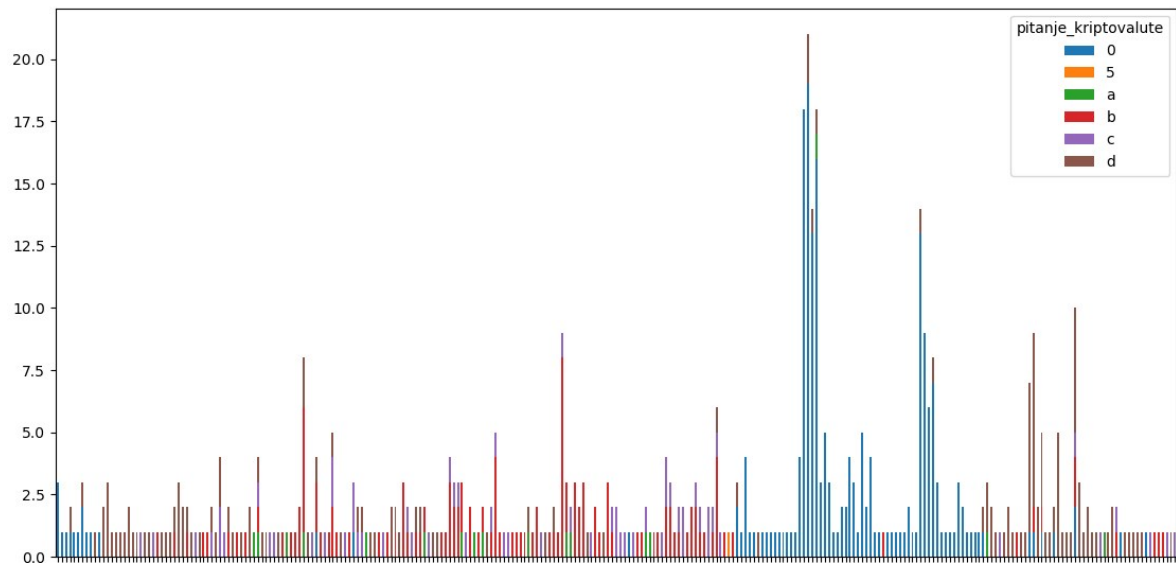
Udeo ispitanika koji su zaokružili pitanja prvog segmenta ankete po sledećem redosledu: DA, 5, c je 15. tj, u procentima je to 2,69 [%] od ukupnog broja ispitanika. Ukoliko se pogledaju ostale zavisnosti prikazane u gornjoj tabeli može se zaključiti da je samovrednovanje ispitanika neobjektivno, jer od broja ispitanika koji su dali odgovor na pitanje o oceni poznavanja kriptoaluta 3,4 ili 5, pogrešan odgovor je dalo (tačan odgovor je pod c) 62,83 [%] ili 61 ispitanik od 97 ispitanika koji su ocenili sopstveno poznavanje kriptoaluta ocenom 3,4 ili 5



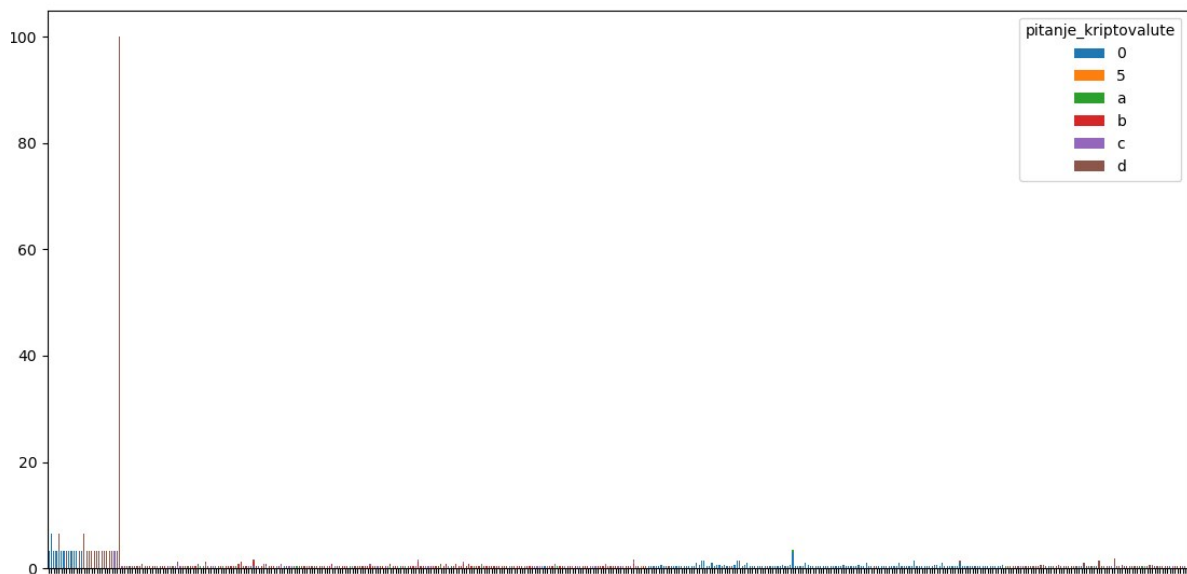
Slika 6.9.2.4. Pitanja petog segmenta povezana sa opštim pitanjima



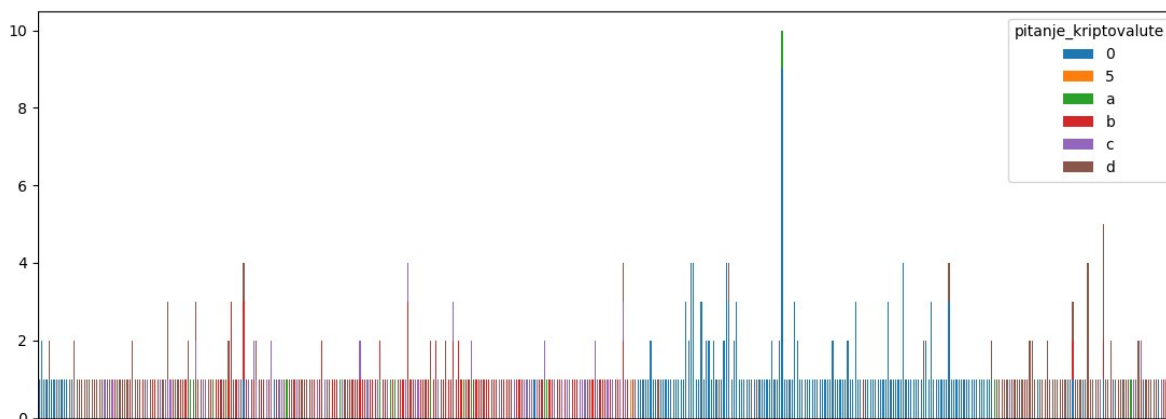
Slika 6.9.2.5. Pitanja petog segmenta povezana sa opštim pitanjima_procentualno



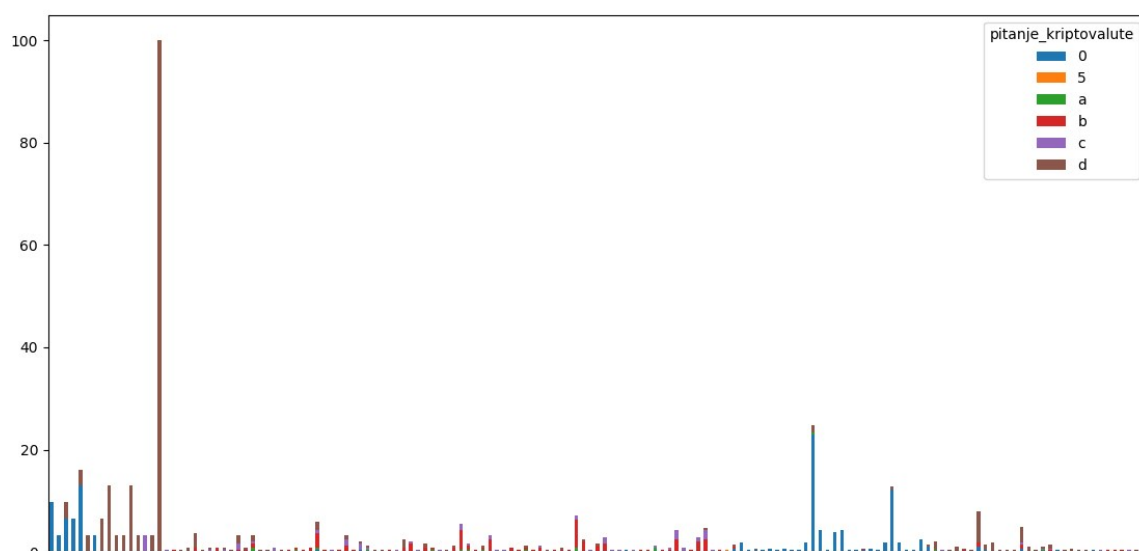
Slika 6.9.2.6. Pitanja petog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu)



Slika 6.9.2.7. Pitanja petog segmenta povezana sa opštim pitanjima_procentualno (bez pitanja za ukupnu prosečnu ocenu)



Slika 6.9.2.8. Pitanja petog segmenta povezana sa opštim pitanjima (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)



Slika 6.9.2.9. Pitanja petog segmenta povezana sa opštim pitanjima _procentualno (bez pitanja za ukupnu prosečnu ocenu i pitanja za prosečnu ocenu iz matematike)

Iz tabelarnog prikaza se može zaključiti da kolona *prosecna_ocena_prethodni_razred* predstavlja i u ovom slučaju opterećenje za razmatranje rezultata. Razlog tome je u činjenici da je opseg ocena od 2 do 5 (nije bilo nedovoljnih) kada je u pitanju tip podataka float veliki, tj. nema puno podudaranja prosečnih ocena. Iz navedene tabele se može zaključiti da je najveći stepen podudarnja kada je ukupna prosečna ocena na kraju godine bila pet. Najveći skup ispitanika čini desetero ispitanika ženskog pola nije dalo odgovor na pitanje o pojmu kriptovaluta, nije ništa zaokružilo na pitanje o samovrednovanju pojma kriptovalute, nije ništa zaokružilo na pitanje iz kriptovaluta, majka i otac su istog nivoa obrazovanja – srednjoškolskog.

Sa druge strane kolona *prosecna_ocena_matematika* je tipa int. tj. može imati celobrojne vrednosti između 2 i 5 (nije bilo nedovoljnih). Zato je u kodu kopirana linija koda koja daje sveobuhvatnu zavisnost opštih podataka i prvog segmenta pitanja, a zatim je iz te linije koda izbačena kolona *prosecna_ocena_prethodni_razred*. Iz dobijenih tabelarnih rezultata se može zaključiti da najveći broj ispitanika koji odgovorio da poznaje pojam osiguranja dolazi iz skupa koji u sebe obuhvata sledeće karakteristike:

- 21 ispitanika ženskog pola nije odgovorilo na pitanje o poznavanju pojma kriptovalute, nije dalo ocenu o poznavanju kriptovaluta, . Ovi ispitanici nisu odgovorili na postavljeno pitanje iz oblasti kriptovaluta: devetnaestoro njih nije odgovorilo na pitanje iz kriptovaluta, dok je dvoje ispitanika zaokružilo ne znam. Nivo obrazovanja majke i oca je isti, tj. srednjoškolski, a prosečna ocena matematike iz prethodnog razreda je 3. Postoje još tri skupa sa većim brojem elemenata kada je u pitanju ženski pol: 2 skupa po 18 elemenata i 1 skup od 14 elemeanta. Razlika između ova četiri skupa je samo u odnosu ko nije odgovorio na pitanje i ko je zaokružio odgovor ne znam.
- 14 ispitanika muškog pola nije odgovorilo na pitanje o poznavanju pojma kriptovaluta i svoje poznavanje pojma kriptovaluta nije ocenilo. Samo je jedan ispitanik odgovorio i to pogrešno na pitanje iz oblasti kriptovaluta dok ostali ispitanici nisu davali odgovor. Nivo obrazovanja majke i oca je isti, tj. srednjoškolski, a prosečna ocena matematike iz prethodnog razreda je 2.

Ukoliko se izbaci i kolona koja se odnosi na prosečnu ocenu iz matematike (spred-šit Anketa42_rezultati) može se uočiti da najveći skup ispitanika istih karakteristika ima 71 element. Karakteristike ovakvog skupa su: ženski pol, NE, nije dalo ocenu o poznavanju kriptovaluta, nije odgovorilo na pitanje o kriptovalutama, majka i otac su istog nivoa obrazovanja – srednjoškolskog.

Što se tiče ispitanika muškog pola, najveći skup ima 37 elemenata sledećih karakteristika: nije čulo za pojam kriptovalute, nije dalo ocenu poznavanja pojma kriptovalute, njih 35 nije dalo odgovor na pitanje iz oblasti kriptovaluta, dok su dva ispitanika zaokružila odgovor ne znam, nivo obrazovanja majke i oca je isti – srednjoškolski.

6.10 Grupisanje pitanja iz segmenata u klastere

Da bi podaci vizuelno bili pregledniji moguće ih je grupisati u klastere. U kodu koji je razvijen u okviru doktorske disertacije koristi se sledeća linija koda za grupisanje:

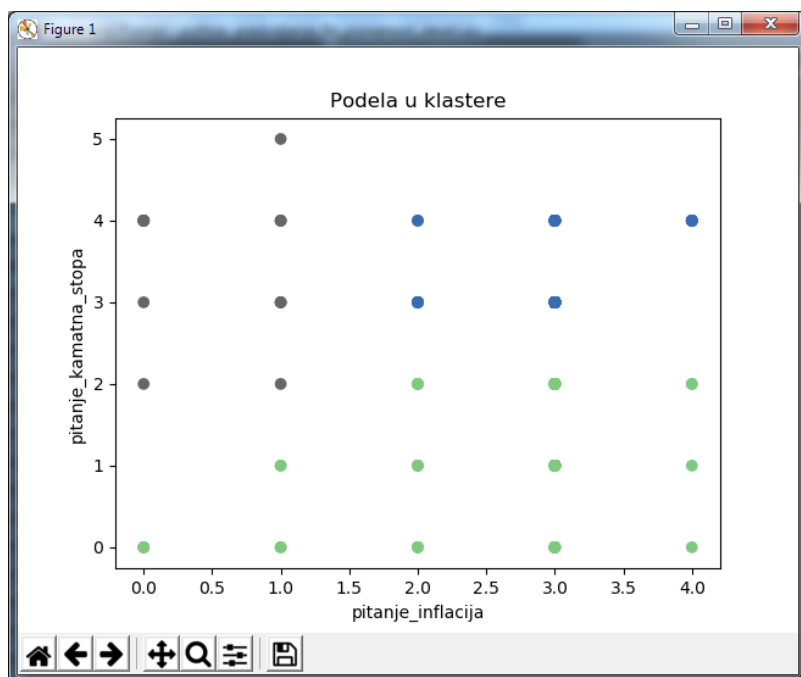
```
from sklearn.cluster import Kmeans
```

Paket `sklearn.cluster` omogućava uvoz algoritama za grupisanje skupova podataka. `Kmeans` je jedan od najkorišćenijih algoritama za grupisanje podataka u mašinskom učenju. Grozdovi (eng. clusters) ili grupe podataka koje se formiraju primenom ovog algoritma se formiraju na osnovu sličnosti između podataka koje je uočio ovaj algoritam. `Kmeans` algoritam traži unapred određeni broj klastera unutar neobebeženog višedimenzionalnog skupa podataka. `Kmeans` algoritam kreira klastere u dva koraka:

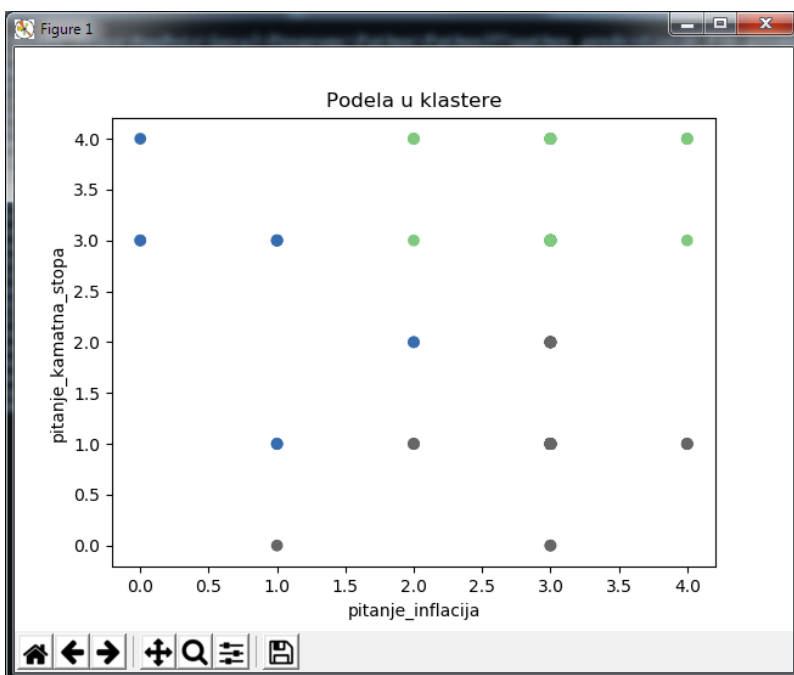
- prvo se kreira centar klastera koji predstavlja aritmetičku sredinu svih tačaka koje pripadaju klasteru
- u drugom koraku se vrši razvrstavanje tačaka, tako da je bilo koja tačka bliža centru svog klastera nego centru nekog drugog klastera.

U radu će biti dat primer grupisanja podataka kolona `pitanje_inflacija` i `pitanje_kamatna_stopa` u zavisnosti od pola ispitanika koji je odabran da bude ženski, tj. 2. Broj klastera koji će biti postavljen je tri.

Na analogan način mogu biti grupisane ostale veličine u zavisnosti od bilo kog drugog parametra. Nedostatak je što je moguće prikazati zavisnosti dve veličine od treće veličine, tj. ne može biti urađena vizuelizacija podataka kao na graficima u prethodnim poglavljima. Sa druge strane dobijeni grafik je pregledniji. Moguće je ispitati zavisnost bilo koje dve veličine od npr. prosečne ocene iz matematike čija je vrednost postavljena da bude pet.



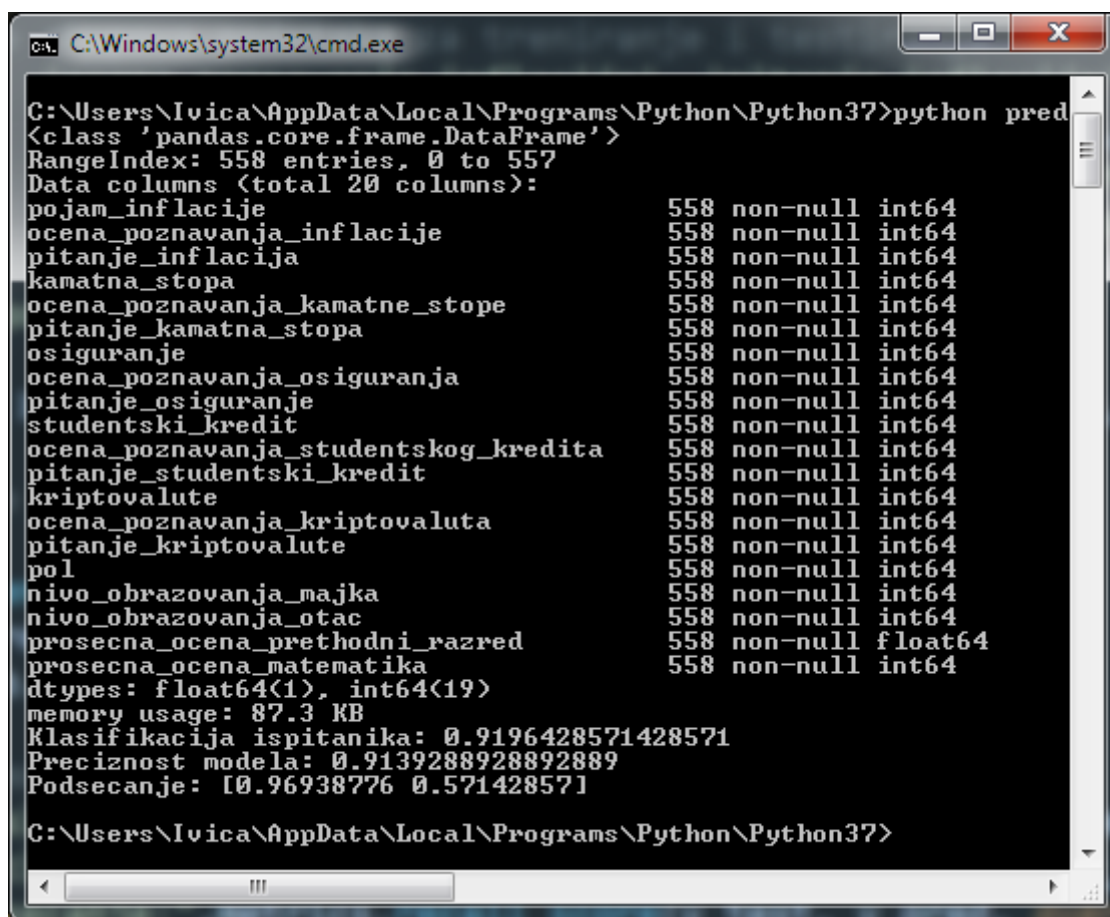
Slika 6.10.1 Grupisanje odgovora iz kolona pitanje_inflacija i pitanje_kamatna_stopa od ženskog pola ispitanika



Slika 6.10.2 Grupisanje odgovora iz kolona pitanje_inflacija i pitanje_kamatna_stopa od prosečne ocene iz matematike čija je vrednost pet

6.11 Predviđanje

Da bi kod razvijen u doktorskoj disertaciji mogao da da određena predviđanja potrebno je da ispitivane kolone imaju u sebi numeričke vrednosti. U poglavlju *Neuronske mreže* doktorske disertacije, je naveden i kroz primer demonstriran, način treniranja neuronske mreže. U predviđanju koje će neuronska mreža kreirana u okviru ovog rada izvršiti koristiće se skup podataka za treniranje i skup podataka za testiranje neuronske mreže. Odnos u kodu koji će biti postavljen je 80:20 u korist skupa podataka za treniranje neuronske mreže. U primeru čija slika sledi uzete su zavisnosti prvih pet segmenata ankete od izabranog parametra koji pripada segmentu opšti podaci, tj. kao parameter je uzeta kolona 'pol'.



```
C:\Windows\system32\cmd.exe

C:\Users\Ivica\AppData\Local\Programs\Python\Python37>python pred
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 558 entries, 0 to 557
Data columns (total 20 columns):
pojam_inflacije          558 non-null int64
ocena_poznavanja_inflacije  558 non-null int64
pitanje_inflacija        558 non-null int64
kamatna_stopa            558 non-null int64
ocena_poznavanja_kamatne_stope  558 non-null int64
pitanje_kamatna_stopa    558 non-null int64
osiguranje               558 non-null int64
ocena_poznavanja_osiguranja  558 non-null int64
pitanje_osiguranje       558 non-null int64
studentski_kredit        558 non-null int64
ocena_poznavanja_studentskog_kredita  558 non-null int64
pitanje_studentski_kredit  558 non-null int64
kriptovalute             558 non-null int64
ocena_poznavanja_kriptovaluta  558 non-null int64
pitanje_kriptovalute     558 non-null int64
pol                       558 non-null int64
nivo_obrazovanja_majka    558 non-null int64
nivo_obrazovanja_otac     558 non-null int64
prosecna_ocena_prethodni_razred  558 non-null float64
prosecna_ocena_matematika  558 non-null int64
dtypes: float64(1), int64(19)
memory usage: 87.3 KB
Klasifikacija ispitanika: 0.9196428571428571
Preciznost modela: 0.9139288928892889
Podsecanje: [0.96938776 0.57142857]

C:\Users\Ivica\AppData\Local\Programs\Python\Python37>
```

Slika 6.11.1. Predviđanje koda

Sa gornje slike se vidi da je procenat tačne klasifikacije podataka 91,96 [%]. Preciznost modela je 91,39 [%]. Ovaj procenat znači da će model biti u pravu u 91,39 [%] slučajeva da se radi o ispitaniku ženskog pola. Podsecanje predstavlja parameter koji

ukazuje da dati model u 96,94 [%] vremena može da otkrije ispitanika koji je kategorizovan i smešten u kolonu pol u skupu podataka koji se koristi za testiranje.

Rezultati su zadovoljavajuću uzimajući u obzir činjenicu da ima veliki broj različitih promenljivih koje treba dovesti u međusobnu zavisnost.

Moguće je koristiti ovaj model za predviđanje ostalih zavisnosti između različitih kolona posmatrane ankete.

7. Zaključna razmatranja

U doktorskoj disertaciji su pokazane prednosti korišćenja savremenih tehnologija u procesu donošenja odluke. Akcenat u radu je stavljen na kreiranje originalnog koda koji može izvršiti analizu velikog skupa podataka i dati određena predviđanja. Kroz različite originalne primere je prikazan način klasifikacije podataka i njihova analiza.

Današnji sistemi su sve složeniji i klasičnim pristupom rešavanju problema se vreme odlučivanja znatno produžava, jer je okruženje sistema podložno brzim promenama. Upravo te stalne promene dovode do otežavanja procesa donošenja kvalitetnih odluka.

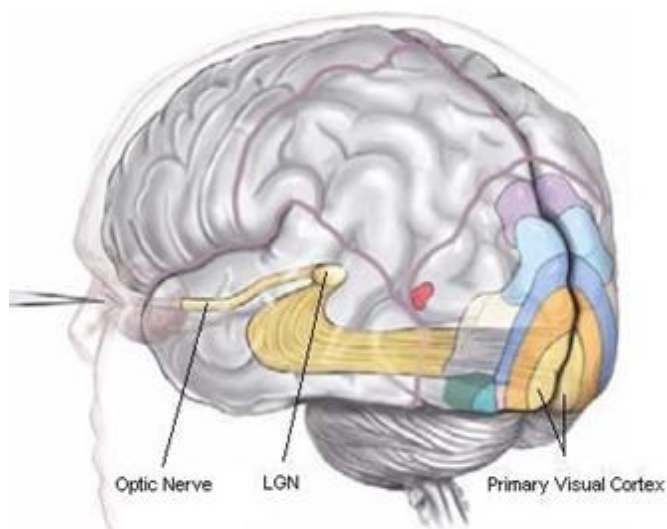
Oblast IKT-a preuzima vodeću ulogu u savremenom poslovanju. Sledeći stepen razvoja u poslovnom odlučivanju je primena veštačke inteligencije, tj. veštačkih inteligentnih agenata koji, kako je pokazano u disertaciji, mogu brzo i kvalitetno da obave proces obrade i analize podataka. Veštački inteligentni agenti mogu da uče na osnovu sopstvenog iskustva koristeći algoritme koji su implementirani u njih. Proces samoučenja dovodi do evolutivnog razvoja agenata i adaptiranja na promene u okruženju, što je i pokazano u doktorskoj disertaciji. Ovi agenti mogu znatno olakšati proces donošenja odluke.

U okviru doktorske disertacije je razvijen kod koji može da obradi veliku količinu podataka i odredi njihove složene međuzavisnosti. Fokus u toku ispitivanja je stavljen na određivanje finansijske pismenosti završnih godina srednjoškolaca. Pitanje je značajno i kroz rad su data razmatranja i zaključci koji omogućavaju sagledavanje ovog problema kod nas. Jedan od ciljeva disertacije je pokazivanje uspešnosti primene IKT-a u donošenju odluka. Obrada rezultata klasičnim pristupom i davanje ovolikog broja međuzavisnosti bi zahtevala dosta vremena i velike finansijske izdatke. U radu je pokazano da je moguće izvršiti analizu podataka na mnogo jednostavniji i znatno brži način razvijanjem koda i implementacijom istog u dinamično okruženje. Ovakav pristup omogućava kreiranje softverskih alata koji će izvršiti pretragu velikih količina podataka, kao što je to ljudska DNK. Kreiranjem bioinformatičkih alata omogućava se analiza DNK u potrazi za retkim bolestima.

Zaključna razmatranja doktorske disertacije je najbolje završiti primerom upotrebe veštačkih inteligentnih agenata na prepoznavanju rukopisa. Jedan od delova koji oduzima

dosta vremena kod obrada anketa je unos podataka u odgovarajući format fajla koji će obraditi veštački inteligentni agenti. Zato je cilj da i ovaj proces bude automatizovan.

Primarni vizuelni (vidni) korteks, koji se često naziva i V1, se nalazi u potiljačnom režnju (lat. Lobus occipitalis) u obe moždane hemisfere ljudskog mozga i on čini mali deo vidljive površine korteksa u potiljačnom režnju, ali se zato proteže u dubinu. Primarni vizuelni korteks je neophodna struktura koja omogućava svesnu obradu vizuelnih podražaja. Ukoliko je primarni vizuelni korteks oštećen dolazi do poremećaja vizuelne percepcije koji se kreće u opsegu od gubitka specifičnih aspekata vida (npr. percepcija dubine) do potpunog gubitka svesnosti o primljenim vizuelnim stimulansima [96].



Slika 7.1. Primarni vizuelni korteks ([91])

Pored primarnog vizuelnog korteksa ljudski vid obuhvata i vizuelne kortekse u oznakama V2, V3, V4 i V5. Ljudski mozak upotrebom svih navedenih vizuelnih korteksa u potpunosti obrađuje primljene vizuelne stimulanse, a kao rezultat te obrade nastaje slika. Kvalitet dobijene slike, između ostalog, zavisi i od kvaliteta primljenih vizuelnih stimulansa. Usavršavanje procesa obrade primljenih nadražaja je posledica evolucije [97].

Na datoj slici su prikazane arapske cifre koje su mutne i koje mogu dovesti do zabune u njihovom prepoznavanju. Primarni vizuelni korteks ljudskog mozga, koji sadrži preko stočetrestdeset miliona neurona koji su međusobno povezani milijardama veza, će ove cifre prepoznati.



Slika 7.2. Arapske cifre pisane rukom

Skenirani brojevi sa gornje slike mogu biti unešeni u računar. Bez obzira da li je gornja slika skenirana kao celina ili je skenirana cifra po cifra, pa onda svaki od skenova unešen u računar, računar će dati/e sken/ove videti kao sliku, a ne kao tekst. Određeni računarski programi, kao što je to npr. Adobe Acrobat Reader, imaju ugrađen OCR (Optical Character Recognition) koji omogućava da svaka slika bude prevedena u format teksta. Tako dobijeni tekst može biti obrađen pomoću nekog od tekst editora koji su instalirani na datom računaru (notepad, VI editor, Microsoft Word,...). Program za prepoznavanje karaktera neće moći da prepozna karaktere koje nema u svojoj bazi na raspolaganju. Dati program može da analizira određenu reč i da eventualni nejasni karakter prepozna na osnovu skupa reči koje ima pohranjene u svojoj bazi. Ukoliko program nema datu reč u svoj bazi on neće moći da prepozna nejasne karaktere. To je slučaj kod jezika koje ne podržava dati program.

Napisati algoritam koji će moći da prepozna karakter u datoj reči je vrlo kompleksan posao koji zahteva veliku količinu podataka i to samo za pojedinačan ljudski jezik. Ovaj program nema intuiciju da prepozna koji broj ili koja reč, odnosno koji je karakter u pitanju.

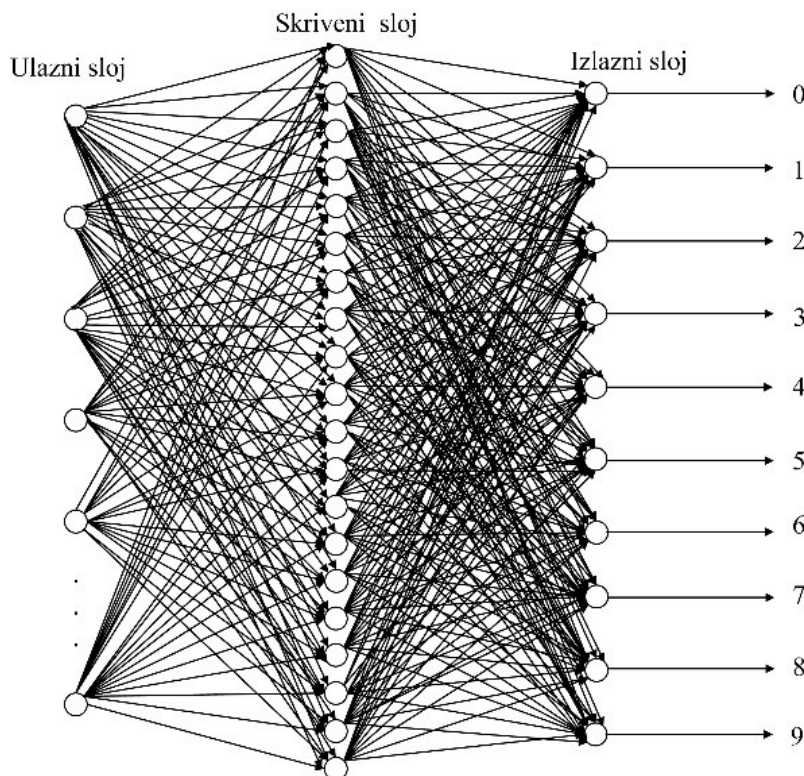
Da bi taj problem bio uspešno rešen potrebno je koristiti drugačiji pristup, tj. treba koristiti neuronske mreže. Slično klasičnom pristupu i kod neuronskih mreža je potreban veliki broj primera rukom pisanih karaktera, ali ne i konačan broj, već dovoljno veliki broj (do nekoliko miliona primera) iz koga će neuronske mreže moći da uče. Neuronska mreža će te primere koristiti da bi mogla da izvuče određena saznanja za prepoznavanje rukom napisanih karaktera i na osnovu tog stečenog znanja da predvidi o kom se karkateru radi.

Kod neuronskih mreža su vrlo bitna koncepti veštačkih neurona (perceptron i sigmoidni neuron) i stohastiškog pristupa problemu.

Da bi veštačka neuronska mreža prepoznala ručno napisanu i skeniranu cifru, npr. četiri, potrebno je koristiti 784 neurona, jer je gustina skenirane slike $28 \times 28 = 784$ [px]. Radi uprošćavanja može se smatrati da je boja napisane cifre četiri sive nijanse koja varira u opsegu od 0.0 (bela) do 1.0 (crna). U ovom slučaju je potrebno koristiti neuronsku mrežu koja ima skriveni sloj. Broj neurona u skrivenom sloju može da varira i eksperimentalno može biti određen adekvatan broj neurona u skrivenom sloju. U primeru koji će biti obrađen u skrivenom sloju će biti 20 neurona. Kod izlaznog sloja moguće je koristiti dve varijante i to:

- izlaz ima deset neurona (za cifre od 0 do 9)
- izlaz ima četiri neurona.

Kako neuroni rade/operišu sa dva binarna stanja (logička nula i logička jedinica) koristi se binarni sistem, pa je broj mogućih kombinacija koje se mogu dobiti pomoću četiri neurona: $2^4 = 16$, što je više nego dovoljno za deset cifara. Pisanje cifre u sebe uključuje niz pokreta rukom, tako da se svaka rukom napisana cifra, u zavisnosti od toga ko je i kako pisao, može sastojati od različitog broja detalja. To je razlog primene izlaza koji će se sastojati od deset neurona, iako je efikasnije koristiti samo četiri izlazna neurona. Sa druge strane korišćenjem četiri neurona može doći do povećanja i akumulacije greške pri prepoznavanju cifara. Primenom izlaza sa deset neurona se dobija optimalno rešenje u pogledu odnosa efikasnost – pojava greške. U ovom primeru se uočava heuristički pristup rešavanju problema, jer je moguće kreirati neuronsku mrežu koja se sastoji od većeg broja skrivenih slojeva i od različitog broja neurona, što dovodi do različitog pristupa u rešavanju zadatka/problema. Na slici 7.3. je dat izgled mreže, pri čemu je slika uprošćena, jer se početni sloj sastoji od 784 neurona što na datoj slici nije prikazano.



Slika 7.3.3. Izgled neuronske mreže (izvor: autor)

Podaci koji su potrebni pri treniranju neuronskih mreža za prepoznavanje slika se nalaze u bazi podataka koja se naziva MNIST (Modified National Institute of Standards and Technology) [92]. Ova baza podataka sadrži preko 60000 skenova cifara napisanih rukom koji služe za treniranje [92] i preko 10000 skenova cifara napisanih rukom koje služe za testiranje [92]. Skenovi cifara iz MNIST baze podataka su klasifikovani u prema vrednosti cifre koja je napisana. Baza MNIST je razvijena iz baze koja se naziva NIST (National Institute of Standards and Technology) [93].

Iz navedenog se može uočiti velika složenost problema koju nije moguće rešiti bez upotrebe savremenih informacionih tehnologija i savremenih pristupa u rešavanju problema.

Literatura

- [1] <https://www.google.com/insidesearch/howsearchworks/thestory/index.html> [Pristup: 25/11/2016]
- [2] <http://gs.statcounter.com/search-engine-market-share> [Pristup: 10/03/2019]
- [3] Agrawal, R., et al. (2016) „Overlap in the Web Search Results of Google and Bing” *The Journal of Web Science*, Vol. 2, No. 2, Pp 17-30
- [4] Sui, D. et al. (2015), *The Deep Web and the Darknet: A Look Inside the Internet's Massive Black Box*, Wilson Center, Washington DC, Dostupno na: https://www.wilsoncenter.org/sites/default/files/stip_dark_web.pdf, Pristup [18/04/2018]
- [5] Darwin, C. (1859), *On the origin of species*, D. Appleton & Company, New York
- [6] Brian, A. (2009), *The Nature of Technology: What It Is and How It Evolves*, The Free Press (Simon & Schuster), New York
- [7] McCarthy, J., et al. (1956) „A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence”, Konferencija održana na Dartmout koledžu. Dostupno na: <http://www.formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>, Pristup: [20.03.2017]
- [8] Collins Dictionaries (2016), *Essential English dictionary – Essential edition*, HarperCollins Publishers, London
- [9] Turing, A. (1950), *Computing machinery and intelligence*, *Mind*, Vol. 49, Pp 433-460
- [10] McCulloch, W., and Pitts, W. (1943) „A logical calculus of the ideas immanent in nervous activity“ *Bulletin of the mathematical biophysics*, Vol 5, Pp 115-133
- [11] <https://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/one.html> Pristup: [17/03/2018]
- [12] Shannon, C.E. (1950) „Programming a Computer for Playing Chess”, *Philosophical Magazine*, Vol. 41, No. 314, Pp 56-275
- [13] McCarthy, J. (1958) „Programs with Common Sense”, *Proceedings of the Teddington Conference of the Mechanization of Thought processes*, Her Majesty's Stationery Office, London, Pp 75-91
- [14] Rosenblatt, F. (1962), *Principles of neurodynamics; perceptrons and the theory of brain mechanisms*, Spartan Books, Washington DC
- [15] Newell, A. and Simon, H. (1972), *Human Problem Solving*, Prentice Hall, Englewood Cliffs, New Jersey.

- [16] Zadeh, L. (1965) „Fuzzy sets” *Information and control*, Vol. 8, Pp 338-353
- [17] Feigenbaum, E., and Buchanan, B. (1993) „DENDRAL and Meta-DENDRAL Roots of knowledge Systems and Expert System Applications”, *Artificial Intelligence*, Vol.59, Pp 233-240
- [18] Shortliffe, E. (1976), *Computer-Based Medical Consultations: MYCIN*, Elsevier Inc.
- [19] <https://pdfs.semanticscholar.org/c922/d577c0d8074a43808bb233d26eadb6bb811e.pdf>
Pristup [21/05/2018]
- [20] https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading3.pdf [06/04/2018]
- [21] Waterman, D. (1986), *A Guide to Expert Systems*, Addison-Wesley, Massachusetts
- [22] Durkin, J. (1994), *Expert Systems: Design and Development*, Prentice Hall, Englewood Cliffs, New Jersey
- [23] Grossberg, S. (1980) „How does a brain build a cognitive code?., *Psychological Review*, Vol.87, No. 1 Pp 1–51
- [24] Hopfield, J. (1982) „Neural networks and physical systems with emergent collective computational abilities., *Proceeding of the National Academy of Sciences of the United States of America*, Vol. 79, Pp 2254-2258
- [25] McClelland, J., and Rumerhart, D. (1986), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Volume 1: Foundations*, A Bradford Book, The MIT Press, Cambridge, Massachusetts
- [26] Bryson, A., and Ho, Y.C. (1969), *Applied Optimal Control*, Blaisdell, New York
- [27] Cun, Y., L. (1988) „A Theoretical Framework for Back-Propagation., *Proceedings of the 1988 Connectionist Models Summer School*, CMU, Pittsburg, PA, Morgan Kaufmann
- [28] Broomhead, D.S., and Lowe, D. (1988) „Multivariable Function Interpolation and Adaptive Networks“ *Complex Systems*, Volume 2, Issue 3, Pp 321-355
- [29] Holland, J. (1975), *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan
- [30] Koza, J. (1992), *Genetic Programming: On the Programming of the Computers by Means of Natural Selection*, A Bradford Book, The MIT Press Cambridge, Massachusetts

- [31] Azevedo, F., A. et al. (2009) „Equal Numbers of Neuronal and Nonneuronal Cells Make the Human Brain an Isometrically Scaled-Up Primate Brain“ *The Journal of comparative neurology*, Vol. 513, Issue 5, Pp 532-541
- [32] Squire, L. et al. (2008), *Fundamental Neuroscience*, Elsevier Inc.
- [33] Rubin, D.,I. and Daube, J.R. (2016), *Clinical Neurophysiology*, Oxford University Press, New York
- [34] <http://nobaproject.com/modules/neurons> [Pristup: 15/3/2017]
- [35] Neumann, v., J. (1945), *First Draft of a Report on EDVAC*, Contract No. W-670-ORD-4926 Between the United States Army Ordnance Department and the University of Pennsylvania, University of Pennsylvania
- [36] Zhang, H. et al. (2017) „Theta and alpha oscillations are traveling waves in the human neocortex“ *Neuron*, Vol. 98, Issue 6, Pp 1269-1281
- [37] Kopell, N. J., et al. (2014) „Beyond the connectome: the dynamo“ *Neuron*, Vol. 83, Pp 1319-1328
- [38] Idris, Z. et al. (2014) „Principles, Anatomical Origin and Applications of Brainwaves: A Review, Our Experience and Hypothesis Related to Microgravity and the Question on Soul“ *Biomedical Science and Engineering*, Vol. 7, Pp 435-445
- [39] Hebb, D. O. (1949), *The Organization of Behavior*, John Wiley & Sons, Inc., New York
- [40] Haykin, S. (2009), *Neural Networks and Learning Machines*, Pearson Education, Inc., Upper Saddle River, New Jersey
- [41] Widrow, B., Hoff., B.E. (1960) „Adaptive Switching Circuits“, *Technical Report No. 1553-1*, Stanford Electronics Laboratories, Stanford University, Stanford California
- [42] McClelland, J.L. (2015), *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*, Dostupno na: <https://web.stanford.edu/group/pdplab/pdphandbook/handbook.pdf> Pristup [20/07/18]
- [43] Hopfield, J. (1982) „Neural networks and physical systems with emergent collective computational abilities“ *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 79, Pp 2554-2558
- [44] Rojas, R. (1996), *Neural Networks*, Springer

- [45] Magoulas, G., and Vrahatis, M.N. (2005) „Adaptive Algorithms for Neural Network Supervised Learning: a Deterministic Optimization Approach“ *International Journal of Bifurcation and Chaos*, Vol. 16, No. 7, Pp 1929–1950
- [46] Kohonen, T. (1990) „The self-organizing Map“ *Proceedings of the IEEE*, Vol. 78, Issue 9, Pp 1464-1480
- [47] Pian C., et al. (2016) „LncRNApred: Classification of Long Non-Coding RNAs and Protein-Coding Transcripts by the Ensemble Algorithm with a New Hybrid Feature“ *Plos One*, Vol. 11, Issue 5, Dostupno na: <http://europepmc.org/articles/PMC4882039> Pristup [25/08/2018]
- [48] Eluyode1, O.E., and Akomolafe, D.T. (2013) „Comparative study of biological and artificial neural networks” *European Journal of Applied Engineering and Scientific Research*, Vol. 2, Issue 1, Pp 36-46
- [49] Auda, G., and Kamel, M. (1999) „Modular neural networks: A survey” *International Journal of Neural Systems*, Vol. 9, No. 2, Pp. 129-151
- [50] Cover, T., M. (1965) „Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition” *IEEE Transactions on Electronic Computers*, Vol EC-14, Issue 3, Pp 326-334
- [51] Back, T., et al. (1991) „A Survey of Evolution Strategies” *Proceedings of the Fourth International Conference on Genetic Algorithms*, Pp. 2-10
- [52] Beyer, H.G., and Schwefel, H. P. (2002) „Evolution strategies” *Natural Computing*, Vol. 1, Pp. 3-52
- [53] Brabazon, A. (2015), *Natural Computing Algorithms*, Springer
- [54] Gajbhiye1, A. V., et al. (2017) „An Overview of Natural Computing Paradigms“ *IOSR Journal of Computer Engineering*, Vol. 19, Issue 2, Ver. IV, Pp. 42-44
- [55] Eiben, A. E., and Smith, J.E. (2015), *Introduction to Evolutionary Computing, second edition*, Springer
- [56] Back, T., et al. (2000), *Evolutionary Computation I Basic Algorithms and Operators*, Institute of Physics Publishing, Bristol and Philadelphia
- [57] Montgomery, D.C., et al. (2015), *Time Series Analysis and Forecasting, second edition*, John Wiley & Sons, Inc., Hoboken, New Jersey
- [58] Shannon, C. (1949) „Communication in the Presence of Noise” *Proceedings of the IRE*, Vol. 37, Issue 1, Pp. 10-21

- [59] Ziat, A., et al. (2017) „Spatio-Temporal Neural Networks for Space-Time Series Forecasting and Relations Discovery” *2017 IEEE International Conference on Data Mining (ICDM)*, IEEE Computer Society, Pp. 705-714
- [60] Oh, S.J., et al. (2018) „Towards Reverse-Engineering Black-Box Neural Networks” *ICLR 2018 Conference*, Vancouver, Dostupno na: <https://openreview.net/pdf?id=BydjJte0-> Pristup [21/09/18]
- [61] Pei, K., et al. (2017) „DeepXplore: Automated Whitebox Testing of Deep Learning Systems” *Proceedings of the 26th Symposium on Operating Systems Principles*, Dostupno na: <https://arxiv.org/pdf/1705.06640.pdf> Pristup [22/09/18]
- [62] Mukhopadhyay, I., and Chakraborty, M. (2014) „Hardware Realization of Artificial Neural Network Based Intrusion Detection & Prevention System“ *Journal of Information Security*, Vol. 2, Pp 154-165
- [63] Backer, R.J. (2010), *CMOS Circuit Design, Layout, and Simulation, third edition*, John Wiley & Sons, Inc., Hoboken, New Jersey
- [64] Merolla, P.A., et al. (2014) „A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, Vol. 345, no. 6197, Pp 668–673
- [65] Aparin, V., and Levin, J.A. (2014) „Methods and systems for cmos implementation of neuron synapse” US Patent No 8694452, Dostupno na: <https://patents.justia.com/patent/8694452> Pristup [26/04/2018]
- [66] Pickering, J. (2015) „Analogue to Digital and Digital to Analogue Converters (ADCs and DACs): A Review Update“ *Proceedings of the CAS-CERN Accelerator School: Power Converters*, Pp 363-377
- [67] Poole, D.L., and Mackworth, A.K. (2017), *Python code for Artificial Intelligence: Foundations of Computational Agents*, Cambridge University Press, Dostupno na: <https://artint.info/2e/html/ArtInt2e.html>, Pristup [08/08/2018]
- [68] Dostupno na: <https://pep8.org/> Pristup [20/10/2018]
- [69] Dostupno na: <http://yann.lecun.com/exdb/mnist/> Pristup [10/08/2018]
- [70] NIST Special Database 19 Handprinted Forms and Characters Database, second edition (2016) Dostupno na: <https://www.nist.gov/srd/nist-special-database-19> Pristup [10/08/2018]

- [71] Silver, N. (2012), *The Signal and the Noise: Why So Many Predictions Fail--but Some Don't*, The Penguin Press, New York, USA
- [72] Goodfellow, I., et al. (2016), *Deep Learning (Adaptive Computation and Machine Learning series)*, The MIT Press
- [73] Fawzi, A., et al. (2017) „The Robustness of Deep Networks - A geometrical perspective” *IEEE Signal Processing Magazine*, Vol. 34, Pp 50-62
- [74] Barbu, T. (2013) „Variational Image Denoising Approach with Diffusion Porous Media Flow” *Abstract and Applied Analysis*, Vol. 2013, Pp
- [75] Aggarwal, C., C. (2018), *Neural Networks and Deep learning*, Springer
- [76] Popov, A. (2014), *Lobachevsky Geometry and Modern Nonlinear Problems*, Springer International Publishing Switzerland
- [77] Nwankpa, C., E., et al. (2018) „Activation Functions: Comparison of Trends in Practice and Research for Deep Learning” *arXiv*, Dostupno na: <https://arxiv.org/pdf/1811.03378.pdf> Pristup [10/12/2018]
- [78] Buduma, N. (2017), *Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms*, O'Reilly, USA
- [79] Pal, S., and Gulli, A. (2017), *Deep Learning with Keras : Implement various deep-learning algorithms in Keras and see how deep-learning can be used in games*, O'Reilly, USA
- [80] Arora, R., et al. (2018) „Understanding Deep Neural Networks with Rectified Linear Units” *ICLR 2018 Conference*, Vankuver, Dostupno na: https://openreview.net/pdf?id=B1J_rgWRW Pristup [07/12/18]
- [81] Gao, B., and Pavel, N. (2018) „On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning” *arXiv*, Dostupno na: <https://arxiv.org/pdf/1704.00805v4.pdf> Pristup [10/12/2018]
- [82] Ketkar, N. (2017), *Deep Learning with Python: A Hands-on Introduction*, Apress Inc., New York, USA
- [83] <http://publikacije.stat.gov.rs/G2018/Pdf/G20186004.pdf> Pristup: 05/03/2019
- [84] <http://publikacije.stat.gov.rs/G2018/Pdf/G20181160.pdf> Pristup: 06/03/2019
- [85] <http://www.mpn.gov.rs/wp-content/uploads/2015/08/Studentski-krediti.pdf> Pristup: [10/03/2019]

- [86]<https://studenti.unicreditbank.rs/korisne-informacije/najcesce-postavljana-pitanja.13.html> Pristup: [10/03/2019]
- [87]<https://www.kombank.com/sr/stanovnistvo/kredit-gradjanima/studenti-kredit>
Pristup: [11/03/2019]
- [88] <https://www.semanticscholar.org/paper/Using-Self-Organizing-Map-and-Heuristics-to-Small-D%C3%BCzg%C3%BCn-Yavuzoglu/23a033d177e0764d4ff189ab0d0092b1636b013e>
- [89]http://cs.joensuu.fi/pages/oili/PR/?a=Some_Material&b=Linear_And_Nonlinear_Classifiers
- [90]http://cs.joensuu.fi/pages/oili/PR/?a=Some_Material&b=Linear_And_Nonlinear_Classifiers)
- [91]https://www.nottingham.ac.uk/biochemcourses/NTM3/Thomas_Kinniburgh/The_Brain/Primary_visual_cortex.html Pristup: [07/03/2019]
- [92] Dostupno na: <http://yann.lecun.com/exdb/mnist/> Pristup [10/08/2018]
- [93] NIST Special Database 19 Handprinted Forms and Characters Database, second edition (2016) Dostupno na: <https://www.nist.gov/srd/nist-special-database-19> Pristup [10/08/2018]
- [94]<https://pdfs.semanticscholar.org/f8d4/f06a66ba65783deeb575bbd21377fc85aa7c.pdf> Pristup [08/08/2018]
- [95] http://static.usenix.org/event/usenix99/invited_talks/mashey.pdf Pristup [06/08/2017]
- [96]<http://www.nmr.mgh.harvard.edu/nouchinelab/pdfs/Tootell&alPNASV11998.pdf>
Pristup [21/06/2018]
- [97] http://www.esi-frankfurt.de/fileadmin/user_upload/Articles/Bergmann2014.pdf
Pristup [22/06/2018]

Prilog A

U ovom prilogu se nalazi razvijeni kod.

```
# pandas je biblioteka u pajtonu koja u sebi sadrzi veliki broj alata koji omogucavaju  
# slozenu analizu podataka. pandas je biblioteka otvorenog koda.  
# pandas je koriscrn u kodu radi kreiranja dataframe-ova. Dataframe u pandasu  
predstavlja  
# dvodimenzionalne imenovane strukture podataka sa potencijano razlicitim tipovima  
# podataka u kolonama. Dataframe se sastoji od samog podatka, indeksa i kolona.  
import pandas as pd  
import openpyxl  
import matplotlib.pyplot as plt  
# Seaborn je biblioteka u pajtonu koja omogucava vizuelizaciju podataka.  
# Seaborn biblioteka je zasnovana na matplotlib-u.  
# Ova biblioteka omogucava crtanje statističkih grafika.  
import seaborn as sb  
from sklearn.cluster import KMeans  
# sklearn.cluster omogucava uvoz algoritama za grupisanje skupova podataka.  
# KMeans je jedan od najkoriscenijih algortama za grupisanje podataka u masinskom  
ucenju.  
# Grozdovi (eng. clusters) ili grupe podataka koje se formiraju primenom ovog algoritma  
# se formiraju na osnovu slicnosti izmedju podataka koje je uocio ovaj algoritam.  
# Kmeans algoritam traži unapred određeni broj klastera unutar neobeležnog  
# višedimenzionalnog skupa podataka. Kmeans algoritam kreira klasteru u dva koraka:  
# prvo se kreira centar klastera koji predstavlja aritmeticku sredinu svih tacaka koje  
# pripadaju klasteru, a u drugom koraku se vrši razvrstavanje tacaka, tako da je bilo koja  
# tacka bliza centru svog klastera nego centru nekog drugog klastera.  
from sklearn.model_selection import train_test_split  
from sklearn.ensemble import GradientBoostingClassifier  
from sklearn import metrics  
import io  
# Ugradjena (u pandas) funkcija read_csv cita csv vrednosti iz fajla Anketa.csv i unosi ih  
# u dataframe koji je nazvan podaci
```

```

podaci=pd.read_csv('Anketa.csv')
# Funkcija head() vraca n (po difoltu 5) gornjih redova dataframe-a.
# Sintaksa glasi: ime_dataframe-a.head(n="int koji predstavlja broj redova koji zelimo
da
# vratimo. Ukoliko ne stoji nista u zagradi onda se vraca prvih pet redova. Ako stoji n=n
# onda vraca sve redove dataframa. Napomena: prvi red dataframa je indeksiran
nulom." ).
podaci.head().to_excel('Anketa_head.xlsx')
# Funkcija tail() radi slicno kao i funkcija head(), ali vraca po difoltu poslednjih pet
# redova dataframe-a.
podaci.tail().to_excel('Anketa_tail.xlsx')
# Funkcija info() stampa informacije o DataFrame-u. U slucaju Anketa.csv fajla funkcija
# info() ce dati nazive kolona, broj elemenata u pojedinicnim kolonama ,tipove podataka
u
# kolonama, ukupan broj razlicitih tipova podataka, kao i memoriju koju zauzima
dataframe.
# Ukupan broj uzoraka, tj. ispitanika je 558, dok je ukupan broj atributa/kolona .
# Pored toga ova funkcija ce pokazati da li postoje polja koja imaju vrednost null, tj,
# da li postoje prazna polja, koliko ih je i u kojim kolonama se nalaze.
# Ukupan broj atributa/kolona koji se nalazi u Anketa.csv fajlu je 25. Vise o samim
# atributima u radu.
podaci.info()
# Prosledjivanje izlaza funkcije info() koja je primenjena nad dataframe-om podaci u
podaci_info.txt fajl
# U ovom slucaju ce i u cmd-u i u podaci_info.txt fajlu biti podaci o podaci o dataframe-
u.
# Funkcija dataframe ne vraca dataframe, pa funkcija .to_excel ne moze biti direktno
primenjena na info() funkciju.
prevodjenje = io.StringIO()
podaci.info(buf=prevodjenje)
prevod = prevodjenje.getvalue()
with open("podaci_info.txt", "w", encoding="utf-8") as info:

```



```

    info.write(prevod)
    """
# Grupisanje podataka pomocu funkcije groupby().
opsti_podaci =
podaci.groupby(['nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pol']).size().unstack().plot(kind='bar',stacked=True)
opsti_podaci_procenti =
podaci.groupby(['nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pol']).size().groupby(level=0).apply(lambda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
inflacija_pitanja =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pojam_inflacije']).size().unstack().plot(kind='bar',stacked=True)
# Grupisanje podataka pomocu funkcije groupby(). Zavisnost ce biti data u procentima.
inflacija_pitanja_procenti =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pojam_inflacije']).size().groupby(level=0).apply(lambda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
inflacija_pitanja_pol =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pojam_inflacije']).size().unstack().plot(kind='bar',stacked=True)
inflacija_pitanja_pol_procenti =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','pojam_inflacije']).size().groupby(level=0).apply(lambda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
# Grafik zavisnosti prvog segmenta pitanja i opstih podataka ispitanika
prvi_segment_opsti_podaci =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pojam_inflacije']).size().unstack().plot(kind='bar',stacked=True)

```

```

prvi_segment_opsti_podaci_procenti =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocen
a_matematika','pojam_inflacije']).size().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)

prvi_segment_opsti_podaci_bez_ukupne_prosecne_ocene =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pojam_inflacije']).siz
e().unstack().plot(kind='bar',stacked=True)

prvi_segment_opsti_podaci_procenti_bez_ukupne_prosecne_ocene =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pojam_inflacije']).siz
e().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)

prvi_segment_opsti_podaci_bez_prosecnih_ocena =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','pojam_inflacije']).size().unstack().plot(kind='bar',st
acked=True)

prvi_segment_opsti_podaci_procenti_bez_prosecnih_ocena =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','pojam_inflacije']).size().groupby(level=0).apply(la
mbda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)

"""

# kamatna_stopa_pitanja =
podaci.groupby(['ocena_poznavanja_kamatne_stope','pitanje_kamatna_stopa','kamatna
_stopa']).size().unstack().plot(kind='bar',stacked=True)

# kamatna_stopa_pitanja_procenti =
podaci.groupby(['ocena_poznavanja_kamatne_stope','pitanje_kamatna_stopa','kamatna
_stopa']).size().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)

podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pitanje_kamatna
_stopa']).size().unstack().fillna(0).to_excel('Anketa15_rezultati.xlsx')

```

```

podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pitanje_kamatna
_stopa']).size().groupby(level=[0,1]).apply(lambda x: 100 * x /
x.sum()).unstack().to_excel('Anketa16_rezultati.xlsx')
# Ukoliko postoje prazna polja u .csv fajlu moze se koristiti funkcija fillna(0) kojoj se
dodaje argument 0
# i to znaci da ce u .xlsx fajl ta funkcija da postavlja nulu na mesta gde nema odgovora
ili je zaokruzeno vise odgovora.
podaci.groupby(['pojam_inflacije','ocena_poznavanja_inflacije','pitanje_inflacija']).siz
e().unstack().fillna(0).to_excel('Anketa1_rezultati.xlsx')
# Namerno nije stavljena funkcija fillna(0) da bi se uocila razlika izmedju ova dva sita
(Anketa1_rezultati.xlsx i Anketa2_rezultati.xlsx)
podaci.groupby(['pojam_inflacije','ocena_poznavanja_inflacije','pitanje_inflacija']).siz
e().groupby(level=[0,1]).apply(lambda x: 100 * x /
x.sum()).unstack().to_excel('Anketa2_rezultati.xlsx')
# Ubacen pol
podaci.groupby(['nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_
prethodni_razred','prosecna_ocena_matematika','pol']).size().unstack().fillna(0).to_exc
el('Anketa7_rezultati.xlsx')
podaci.groupby(['pol','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','pr
osecna_ocena_matematika','nivo_obrazovanja_majka']).size().unstack().fillna(0).to_exc
el('Anketa9_rezultati.xlsx')
podaci.groupby(['nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_
prethodni_razred','prosecna_ocena_matematika','pol']).size().groupby(level=[0,1]).app
ly(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa8_rezultati.xlsx')
podaci.groupby(['pojam_inflacije','ocena_poznavanja_inflacije','pitanje_inflacija','pol']
).size().groupby(level=[0,1]).apply(lambda x: 100 * x /
x.sum()).unstack().to_excel('Anketa6_rezultati.xlsx')
podaci.groupby(['pol','nivo_obrazovanja_majka','prosecna_ocena_prethodni_razred','p
rosecna_ocena_matematika','nivo_obrazovanja_otac']).size().unstack().fillna(0).to_exc
el('Anketa10_rezultati.xlsx')
# Opsti podaci u tabelu

```

```

podaci.groupby(['pojam_inflacije','ocena_poznavanja_inflacije','pitanje_inflacija']).size().unstack().fillna(0).to_excel('Anketa1_rezultati.xlsx')
# ocena_poz_inflacije = podaci.groupby('ocena_poznavanja_inflacije')
# ocena_poz_inflacije.mean().to_excel('Srednja_vrednost.xlsx')
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pojam_inflacije']).size().unstack().fillna(0).to_excel('Anketa11_rezultati.xlsx')
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pojam_inflacije']).size().unstack().fillna(0).to_excel('Anketa12_rezultati.xlsx')
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','pojam_inflacije']).size().unstack().fillna(0).to_excel('Anketa13_rezultati.xlsx')
# Kamatna stopa
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pitanje_kamatna_stopa']).size().unstack().fillna(0).to_excel('Anketa17_rezultati.xlsx')
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_kamatna_stopa']).size().unstack().fillna(0).to_excel('Anketa18_rezultati.xlsx')
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_kamatna_stopa']).size().unstack().fillna(0).to_excel('Anketa19_rezultati.xlsx')
"""
drugi_segment_opsti_podaci =
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pitanje_kamatna_stopa']).size().unstack().plot(kind='bar',stacked=True)

```

```

drugi_segment_opsti_podaci_procenti =
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obraz
ovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna
ocena_matematika','pitanje_kamatna_stopa']).size().groupby(level=0).apply(lambda x:
100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
drugi_segment_opsti_podaci_bez_ukupne_prosecne_ocene =
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obraz
ovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_kamatn
a_stopa']).size().unstack().plot(kind='bar',stacked=True)
drugi_segment_opsti_podaci_procenti_bez_ukupne_prosecne_ocene =
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obraz
ovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_kamatn
a_stopa']).size().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)
drugi_segment_opsti_podaci_bez_prosecnih_ocena =
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obraz
ovanja_majka','nivo_obrazovanja_otac','pitanje_kamatna_stopa']).size().unstack().plot(
kind='bar',stacked=True)
drugi_segment_opsti_podaci_procenti_bez_prosecnih_ocena =
podaci.groupby(['kamatna_stopa','ocena_poznavanja_kamatne_stope','pol','nivo_obraz
ovanja_majka','nivo_obrazovanja_otac','pitanje_kamatna_stopa']).size().groupby(level
=0).apply(lambda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
"""
# osiguranje_pitanja =
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pitanje_osiguranje']).siz
e().unstack().plot(kind='bar',stacked=True)
# osiguranje_pitanja_procenti =
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pitanje_osiguranje']).siz
e().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)
"""

```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pitanje_osiguranje']).size().unstack().fillna(0).to_excel('Anketa20_rezultati.xlsx')
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pitanje_osiguranje']).size().groupby(level=[0,1]).apply(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa21_rezultati.xlsx')
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pitanje_osiguranje']).size().unstack().fillna(0).to_excel('Anketa22_rezultati.xlsx')
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pitanje_osiguranje']).size().groupby(level=[0,1]).apply(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa23_rezultati.xlsx')
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_osiguranje']).size().unstack().fillna(0).to_excel('Anketa24_rezultati.xlsx')
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_osiguranje']).size().groupby(level=[0,1]).apply(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa25_rezultati.xlsx')
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_osiguranje']).size().unstack().fillna(0).to_excel('Anketa26_rezultati.xlsx')
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_osiguranje']).size().groupby(level=[0,1]).apply(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa27_rezultati.xlsx')
```

Osiguranje

```
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pitanje_studentski_kredit']).size().unstack().fillna(0).to_excel('Anketa28_rezultati.xlsx')
```

```
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pitanje_studentski_kredit']).size().groupby(level=[0,1]).apply(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa29_rezultati.xlsx')
```

```
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo_
_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','pro
secna_ocena_matematika','pitanje_studentski_kredit']).size().unstack().fillna(0).to_exce
l('Anketa30_rezultati.xlsx')
```

```
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo_
_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','pro
secna_ocena_matematika','pitanje_studentski_kredit']).size().groupby(level=[0,1]).appl
y(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa31_rezultati.xlsx')
```

```
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo_
_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_s
tudentski_kredit']).size().unstack().fillna(0).to_excel('Anketa32_rezultati.xlsx')
```

```
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo_
_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_s
tudentski_kredit']).size().groupby(level=[0,1]).apply(lambda x: 100 * x /
x.sum()).unstack().to_excel('Anketa33_rezultati.xlsx')
```

```
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo_
_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_studentski_kredit']).size().unsta
ck().fillna(0).to_excel('Anketa34_rezultati.xlsx')
```

```
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo_
_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_studentski_kredit']).size().group
by(level=[0,1]).apply(lambda x: 100 * x /
x.sum()).unstack().to_excel('Anketa35_rezultati.xlsx')
```

"""

"""

```
treci_segment_opsti_podaci =
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_
majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_
matematika','pitanje_osiguranje']).size().unstack().plot(kind='bar',stacked=True)
```

```
treci_segment_opsti_podaci_procenti =
```

```
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_
majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_
```

```

matematika','pitanje_osiguranje']).size().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)

treći_segment_opšti_podaci_bez_ukupne_prosecne_ocene =
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_
majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_osiguranje']).siz
e().unstack().plot(kind='bar',stacked=True)

treći_segment_opšti_podaci_procenti_bez_ukupne_prosecne_ocene =
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_
majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_osiguranje']).siz
e().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)

treći_segment_opšti_podaci_bez_prosecnih_ocena =
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_
majka','nivo_obrazovanja_otac','pitanje_osiguranje']).size().unstack().plot(kind='bar',st
acked=True)

treći_segment_opšti_podaci_procenti_bez_prosecnih_ocena =
podaci.groupby(['osiguranje','ocena_poznavanja_osiguranja','pol','nivo_obrazovanja_
majka','nivo_obrazovanja_otac','pitanje_osiguranje']).size().groupby(level=0).apply(la
mbda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
"""

# studentski_kredit =
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pitanje_st
udentski_kredit']).size().unstack().plot(kind='bar',stacked=True)

# studentski_kredit_procenti =
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pitanje_st
udentski_kredit']).size().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)
"""

četvrti_segment_opšti_podaci =
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo
_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','pro

```



```

secna_ocena_matematika','pitanje_studentski_kredit']).size().unstack().plot(kind='bar',s
tacked=True)

cetvrti_segment_opsti_podaci_procenti =
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo
_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','pro
secna_ocena_matematika','pitanje_studentski_kredit']).size().groupby(level=0).apply(la
mbda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)

cetvrti_segment_opsti_podaci_bez_ukupne_prosecne_ocene =
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo
_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_s
tudentski_kredit']).size().unstack().plot(kind='bar',stacked=True)

cetvrti_segment_opsti_podaci_procenti_bez_ukupne_prosecne_ocene =
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo
_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_s
tudentski_kredit']).size().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)

cetvrtii_segment_opsti_podaci_bez_prosecnih_ocena =
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo
_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_studentski_kredit']).size().unsta
ck().plot(kind='bar',stacked=True)

cetvrti_segment_opsti_podaci_procenti_bez_prosecnih_ocena =
podaci.groupby(['studentski_kredit','ocena_poznavanja_studentskog_kredita','pol','nivo
_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_studentski_kredit']).size().group
by(level=0).apply(lambda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
"""
"""

# kriptovalute
kriptovalute =
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptovaluta','pitanje_kriptovalute'])
.size().unstack().plot(kind='bar',stacked=True)

kriptovalute_procenti =
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptovaluta','pitanje_kriptovalute'])

```

```

.size()).groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptoaluta','pitanje_kriptoalute'])
.size().unstack().fillna(0).to_excel('Anketa36_rezultati.xlsx')
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptoaluta','pitanje_kriptoalute'])
.size().groupby(level=[0,1]).apply(lambda x: 100 * x /
x.sum()).unstack().to_excel('Anketa37_rezultati.xlsx')
#peti_segment_opsti_podaci =
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptoaluta','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocen
a_matematika','pitanje_kriptoalute']).size().unstack().plot(kind='bar',stacked=True)
#peti_segment_opsti_podaci_procenti =
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptoaluta','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocen
a_matematika','pitanje_kriptoalute']).size().groupby(level=0).apply(lambda x: 100 * x
/x.sum()).unstack().plot(kind='bar',stacked=True)
peti_segment_opsti_podaci_bez_ukupne_prosecne_ocene =
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptoaluta','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_kriptoalute']
).size().unstack().plot(kind='bar',stacked=True)
peti_segment_opsti_podaci_procenti_bez_ukupne_prosecne_ocene =
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptoaluta','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_kriptoalute']
).size().groupby(level=0).apply(lambda x: 100 * x /
x.sum()).unstack().plot(kind='bar',stacked=True)
peti_segment_opsti_podaci_bez_prosecnih_ocena =
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptoaluta','pol','nivo_obrazovanj
a_majka','nivo_obrazovanja_otac','pitanje_kriptoalute']).size().unstack().plot(kind='ba
r',stacked=True)
peti_segment_opsti_podaci_procenti_bez_prosecnih_ocena =
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptoaluta','pol','nivo_obrazovanj

```

```

a_majka','nivo_obrazovanja_otac','pitanje_kriptovalute']).size().groupby(level=0).apply(lambda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptovaluta','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pitanje_kriptovalute']).size().unstack().fillna(0).to_excel('Anketa38_rezultati.xlsx')
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptovaluta','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_prethodni_razred','prosecna_ocena_matematika','pitanje_kriptovalute']).size().groupby(level=[0,1]).apply(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa39_rezultati.xlsx')
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptovaluta','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_kriptovalute']).size().unstack().fillna(0).to_excel('Anketa40_rezultati.xlsx')
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptovaluta','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','prosecna_ocena_matematika','pitanje_kriptovalute']).size().groupby(level=[0,1]).apply(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa41_rezultati.xlsx')
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptovaluta','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_kriptovalute']).size().unstack().fillna(0).to_excel('Anketa42_rezultati.xlsx')
podaci.groupby(['kriptovalute','ocena_poznavanja_kriptovaluta','pol','nivo_obrazovanja_majka','nivo_obrazovanja_otac','pitanje_kriptovalute']).size().groupby(level=[0,1]).apply(lambda x: 100 * x / x.sum()).unstack().to_excel('Anketa43_rezultati.xlsx')
# Zavisnost pitanja sa polom
#inflacija_pitanja_pol =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol'])['pojam_inflacije'].size().unstack().plot(kind='bar',stacked=True)
#inflacija_pitanja_pol_procenti =
podaci.groupby(['ocena_poznavanja_inflacije','pitanje_inflacija','pol']).size().groupby(level=0).apply(lambda x: 100 * x / x.sum()).unstack().plot(kind='bar',stacked=True)
podaci['pojam_inflacije'].describe().to_excel('Opis1.xlsx')
podaci['ocena_poznavanja_inflacije'].describe().to_excel('Opis2.xlsx')

```

```

podaci['pitanje_inflacija'].describe().to_excel('Opis4.xlsx')
podaci['pol'].describe().to_excel('Opis5.xlsx')
# podaci['nivo_obrazovanja_majka'].describe().to_excel('Opis6.xlsx')
# podaci['nivo_obrazovanja_otac'].describe().to_excel('Opis7.xlsx')
podaci['prosecna_ocena_prethodni_razred'].describe().to_excel('Opis8.xlsx')
podaci['prosecna_ocena_matematika'].describe().to_excel('Opis9.xlsx')
podaci['kamatna_stopa'].describe().to_excel('Opis11.xlsx')
podaci['ocena_poznavanja_kamatne_stope'].describe().to_excel('Opis12.xlsx')
podaci['pitanje_kamatna_stopa'].describe().to_excel('Opis15.xlsx')
podaci['osiguranje'].describe().to_excel('Opis17.xlsx')
podaci['ocena_poznavanja_osiguranja'].describe().to_excel('Opis18.xlsx')
podaci['pitanje_osiguranje'].describe().to_excel('Opis20.xlsx')
podaci['studentski_kredit'].describe().to_excel('Opis22.xlsx')
podaci['ocena_poznavanja_studentskog_kredita'].describe().to_excel('Opis24.xlsx')
podaci['pitanje_studentski_kredit'].describe().to_excel('Opis26.xlsx')
podaci['kriptoalute'].describe().to_excel('Opis28.xlsx')
podaci['ocena_poznavanja_kriptoaluta'].describe().to_excel('Opis30.xlsx')
podaci['pitanje_kriptoalute'].describe().to_excel('Opis32.xlsx')
#podaci['pojam_inflacije','ocena_poznavanja_inflacije','pitanje_inflacija','pol'].describe().to_excel('Opis5.xlsx')
# inflacija=podaci.groupby('pojam_inflacije').count()
"""
predvidjanje_mat_3          =          podaci[['pitanje_inflacija',
'pitanje_kamatna_stopa']][(podaci['prosecna_ocena_matematika'] == 5)]
#predvidjanje_mat_4          =          podaci[['pitanje_inflacija',
'pitanje_kamatna_stopa']][(podaci['prosecna_ocena_matematika'] == 4)]
#predvidjanje_mat_5          =          podaci[['pitanje_inflacija',
'pitanje_kamatna_stopa']][(podaci['prosecna_ocena_matematika'] == 5)]
#predvidjanje_pol_z          =          podaci[['pitanje_inflacija',
'pitanje_kamatna_stopa','pitanje_osiguranje','pitanje_studentski_kredit','pitanje_kriptov
alute']][(podaci['pol'] == 1)]

```

```

#predvidjanje_pol_m = podaci[['pitanje_inflacija',
'pitanje_kamatna_stopa','pitanje_osiguranje','pitanje_studentski_kredit','pitanje_kriptov
alute']][(podaci['pol'] == 2)]
p3 = KMeans(n_clusters = 3, random_state = 0).fit(predvidjanje_mat_3)
#p4 = KMeans(n_clusters = 3, random_state = 0).fit(predvidjanje_mat_4)
#p5 = KMeans(n_clusters = 3, random_state = 0).fit(predvidjanje_mat_5)
#pz = KMeans(n_clusters = 3, random_state = 0).fit(predvidjanje_pol_z)
#pm = KMeans(n_clusters = 3, random_state = 0).fit(predvidjanje_pol_m)
predvidjanje_mat_3['label1'] = p3.labels_
#predvidjanje_mat_4['label2'] = p4.labels_
#predvidjanje_mat_5['label3'] = p5.labels_
#predvidjanje_pol_z['label4'] = pz.labels_
#predvidjanje_pol_m['label5'] = pm.labels_
plt.scatter(predvidjanje_mat_3['pitanje_inflacija'],
predvidjanje_mat_3['pitanje_kamatna_stopa'],
c=predvidjanje_mat_3['label1'],cmap='Accent')
#plt.scatter(predvidjanje_mat_4['pitanje_inflacija'],
predvidjanje_mat_4['pitanje_kamatna_stopa'],
c=predvidjanje_mat_4['label2'],cmap='Accent')
#plt.scatter(predvidjanje_mat_5['pitanje_inflacija'],
predvidjanje_mat_5['pitanje_kamatna_stopa'],
c=predvidjanje_mat_5['label3'],cmap='Accent')
#plt.scatter(predvidjanje_pol_z['pitanje_inflacija'],
predvidjanje_pol_z['pitanje_kamatna_stopa'],
c=predvidjanje_pol_z['label4'],cmap='Accent')
#plt.scatter(predvidjanje_pol_m['pitanje_inflacija'],
predvidjanje_pol_m['pitanje_kamatna_stopa'],
c=predvidjanje_pol_m['label5'],cmap='Accent')
plt.xlabel('pitanje_inflacija')
plt.ylabel('pitanje_kamatna_stopa')
plt.title('Podela u klastere')
# plt.bar(inflacija_da.index.values, inflacija_da['ocena_poznavanja_inflacije'])

```

```

# plt.xlabel('Ocena poznavanja inflacije')
# plt.ylabel('Pojam inflacije')
plt.show()
#
podaci.groupby(['pojam_inflacije', 'ocena_poznavanja_inflacije', 'pitanje_inflacija', 'pol', 'nivo_ obrazovanja_majka', 'nivo_ obrazovanja_otac']).size().unstack().to_excel('Anketa1_rezultati.xlsx')
# podaci.to_excel("")
#
podaci.groupby(['pojam_inflacije', 'ocena_poznavanja_inflacije', 'pitanje_inflacija']).to_excel('Anketa1_rezultati.xlsx')
#
podaci.groupby(['ocena_poznavanja_inflacije', 'pitanje_inflacija']).size().unstack().to_excel('Anketa5_rezultati.xlsx')
#
podaci.groupby(['ocena_poznavanja_inflacije', 'pitanje_inflacija', 'pol', 'nivo_ obrazovanja_majka'])['pojam_inflacije'].size().unstack().to_excel('Anketa2_rezultati.xlsx')
#
podaci.groupby(['ocena_poznavanja_inflacije', 'pitanje_inflacija', 'pol', 'nivo_ obrazovanja_majka', 'nivo_ obrazovanja_otac'])['pojam_inflacije'].size().unstack().to_excel('Anketa3_rezultati.xlsx')
#
podaci.groupby(['pojam_inflacije', 'ocena_poznavanja_inflacije', 'pitanje_inflacija']).sum(axis = 0, skipna = True).to_excel('Anketa4_rezultati.xlsx')
#
podaci.loc['Total'] = podaci.groupby(['pojam_inflacije', 'ocena_poznavanja_inflacije', 'pitanje_inflacija']).sum(axis=1).to_excel('Anketa1_rezultati.xlsx')
# .size().unstack()
# Kreiranje skupova podataka za treniranje i testiranje
X = podaci[['pojam_inflacije', 'pitanje_inflacija', 'ocena_poznavanja_inflacije', 'kamatna_stopa', 'ocena_poznavanja_kamatne_stope', 'pitanje_kamatna_stopa', 'osiguranje', 'pitanje_osiguranje', 'pitanje_osiguranje']]

```

```

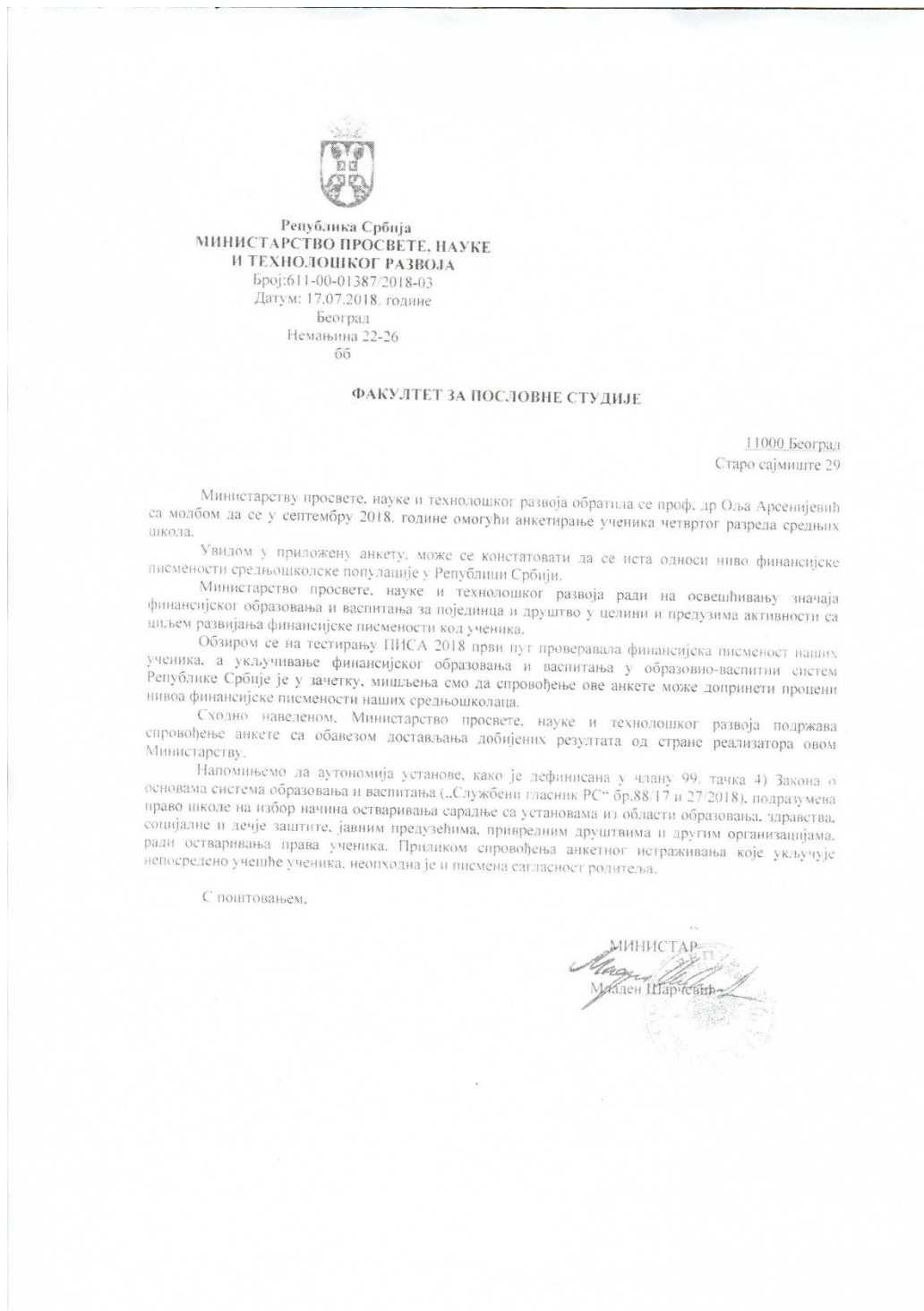
        'kriptovalute','ocena_poznavanja_kriptovaluta','pitanje_kriptovalute']]
    #'studentski_kredit','ocena_poznavanja_studentskog_kredita','pitanje_studentski
    _kredit'
y=podaci['pol']
X_trening, X_test, y_trening, y_test = train_test_split(X, y, test_size=0.2,
random_state=30)

# Kreiranje modela za predviđanje
gb = GradientBoostingClassifier()
gb.fit(X_trening, y_trening)
y_pred = gb.predict(X_test)
# Procena modela
print("Klasifikacija ispitanika:",metrics.accuracy_score(y_test, y_pred))
print("Preciznost          modela:",metrics.precision_score(y_test,          y_pred,
average='weighted'))
print("Podsecanje:",metrics.recall_score(y_test, y_pred, average=None))

```



Prilog B

U ovom prilogu se nalazi pozivno pismo Ministarstva za prosvetu, nauku i tehnološki razvoj.



Prilog C

U ovom prilogu se nalazi pozivno pismo Fakulteta poslovne studije i pravo za učestvovanje u anketi i primer anketnog lista.

 **FAKULTET ZA POSLOVNE STUDIJE I PRAVO,**
UNIVERZITETA „UNION – NIKOLA TESLA“ U BEOGRADU 

Broj: 531/18
Datum: 21.09.2018. godine

Poštovani,


Fakultet za poslovne studije i pravo (FPSP) i Fakultet za informacione tehnologije i inženjerstvo (FITI), u sastavu Univerziteta Union – Nikola Tesla iz Beograda, zajednički realizuju projekat pod nazivom „Finansijska pismenost kod srednjoškolske populacije u Srbiji, Crnoj Gori i Sloveniji“. Cilj prve faze projekta je da se utvrdi nivo finansijske pismenosti srednjoškolske populacije u tri pomenute države, a zatim će se u narednoj fazi organizovati specijalizovani programi za podizanje nivoa finansijske pismenosti srednjoškolaca.

Obraćamo Vam se sa molbom da u okviru vaše škole omogućite sprovođenje ankete o proceni nivoa finansijske pismenosti među učenicima četvrtog razreda (anketni list Vam dostavljamo u prilogu). Rezultati koji budu dobijeni biće objavljeni u specijalnoj međunarodnoj monografskoj publikaciji pod nazivom „Komparativna analiza aktuelnog stanja finansijske pismenosti kod srednjoškolske populacije u Srbiji, Crnoj Gori i Sloveniji“, čije se publikovanje očekuje u prvoj polovini naredne godine. Takođe, rezultati ankete služiće kao polazna osnova za strukturiranje sadržaja specijalizovanih edukativnih programa i tematskih edukativnih kampova, koji će biti organizovani tokom sledećeg leta.

Imajući u vidu značaj projekta za širu društvenu zajednicu, kao i postavljene strateške prioritete u sistemu obrazovanja u Republici Srbiji, dobijena je podrška Ministarstva prosvete, nauke i tehnološkog razvoja za njegovo sprovođenje (u prilogu Vam dostavljamo pismo podrške Ministarstva).

U nadi da ćemo ovaj projekat realizovati na obostrano zadovoljstvo,

S poštovanjem,
prof. dr Zoran Ćirović
rukovodilac Projekta



Fakultet za poslovne studije i pravo, Univerziteta „Union – Nikola Tesla“ u Beogradu
11070 Novi Beograd
www.fpsp.edu.rs Tel: +381(0)31 31 246, +381(0)21 32 372 PIR: 107072407



**ANKETNI LIST O PROCENI NIVOA FINANSIJSKE PISMENOSTI
SREDNJOŠKOLSKE POPULACIJE
U REPUBLICI SRBIJI, REPUBLICI SLOVENIJI I CRNOJ GORI
(posmatrano za period kalendarske 2018-2019. godine)**

Napomena o poverljivosti podataka:

Proces anketiranja je anonimam. Ispitanicima se garantuje anonimnost i diskreciono pravo na sve podatke iznete tokom procesa anketiranja osim u slučaju zbirnog prezentovanja rezultata i zaključaka dobijenih obradom podataka iz anketnih listova u cilju uspešne realizacije naučno-istraživačkog projekta.

I blok pitanja

Molimo Vas da odgovorite na sledeća pitanja zaokruživanjem isključivo jednog od ponuđenih odgovora

01. Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam **INFLACIJA**? DA NE

Ukoliko ste na pitanje broj 01 odgovorili sa (NE), pređite na pitanje broj 02.

01a. Na osnovu Vašeg mišljenja, na skali od 1 do 5, ocenite u kojoj meri ste upoznati sa pojmom **INFLACIJA**? (Ocena 5- odlično upoznat/a; 4- vrlo dobro; 3-dobro; 2- dovoljno; 1-nisam siguran/na)

5, 4, 3, 2, 1

01b. Pretpostavimo de se tokom narednih 10 godina cene onoga što kupujete dupliraju. Ako se Vaši prihodi takode dupliraju, da li ćete tada moći da kupite više, manje ili približno isto u odnosu na ono što možete da kupite danas?

a) Više b) Manje c) Približno isto d) Ne znam

02. Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam **KAMATNA STOPA**? DA NE

Ukoliko ste na pitanje broj 02 odgovorili sa (NE), pređite na pitanje broj 03.

02a. Na osnovu Vašeg mišljenja, na skali od 1 do 5, ocenite u kojoj meri ste upoznati sa pojmom **KAMATNA STOPA**? (Ocena 5- odlično upoznat/a; 4- vrlo dobro; 3-dobro; 2- dovoljno; 1-nisam siguran/na)

5, 4, 3, 2, 1

02b. Pretpostavimo da imate 100 EUR na Vašem štednom računu i da Vam banka pripisuje kamatu od 10% godišnje. Koliko novca ćete imati na računu posle 5 godina pod uslovom da ne podižete ništa u međuvremenu?

a) Više od 150 EUR b) Manje od 150 EUR c) Tačno 150 EUR d) Ne znam

03. Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam **OSIGURANJE**? DA NE

Ukoliko ste na pitanje broj 03 odgovorili sa (NE), pređite na pitanje broj 04.

03a. Na osnovu Vašeg mišljenja, na skali od 1 do 5, ocenite u kojoj meri ste upoznati sa pojmom OSIGURANJE?		(Ocena 5- odlično upoznat/a; 4- vrlo dobro; 3- dobro; 2- dovoljno; 1-nisam siguran/na)		
		5, 4, 3, 2, 1		
03b. Sklapanjem polise osiguranja sa osiguravajućom kompanijom:				
a) Vi preuzimate rizik od osiguravajuće kompanije	<input checked="" type="radio"/> b) Osiguravajuća kompanija preuzima rizik od Vas	c) Niko ne preuzima rizik	d) Ne znam	
4. Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam STUDENTSKI KREDIT?		<input checked="" type="radio"/> DA	NE	
Ukoliko ste na pitanje broj 04 odgovorili sa (NE), predite napitanje broj 05.				
04a. Na osnovu Vašeg mišljenja, na skali od 1 do 5, ocenite u kojoj meri ste upoznati sa pojmom STUDENTSKI KREDIT?		(Ocena 5- odlično upoznat/a; 4- vrlo dobro; 3- dobro; 2- dovoljno; 1-nisam siguran/na)		
		5, 4, 3, 2, 1		
04b. Ukoliko pretpostavimo da ste od banke uzeli studentski kredit, to podrazumeva da koristite:				
<input checked="" type="radio"/> a) Bespovratna finansijska sredstva tokom čitavog perioda studiranja	b) Bespovratna finansijska sredstva tokom prve godine studiranja	c) Pozajmljena finansijska sredstva sa ugovornim periodom otplate u više mesečnih rata	d) Ne znam	
5. Da li ste do ovog trenutka iz bilo kog izvora informisanja čuli za pojam KRIPTOVALUTA (eng. Cryptocurrency)?		<input checked="" type="radio"/> DA	NE	Bez odgovora
Ukoliko ste na pitanje broj 05 odgovorili sa (NE), predite na II blok pitanja (Osnovne informacije o ispitaniku).				
5a. Na osnovu Vašeg mišljenja, na skali od 1 do 5, ocenite u kojoj meri ste upoznati sa pojmom KRIPTOVALUTA?		(Ocena 5- odlično upoznat/a; 4- vrlo dobro; 3- dobro; 2- dovoljno; 1-nisam siguran/na)		
		5, 4, 3, 2, 1		
5b. Ukoliko pretpostavimo da imate određeni iznos sopstvenih finansijskih sredstava koje želite da investirate, kako biste opisali mogućnost investiranja u kriptovalute?				
a) Niska stopa zarade, ali i investicija niskog rizika	b) Veliki potencijal zarade, ali i investicija visokog rizika	c) Veliki potencijal zarade, sa umerenom stopom rizika	<input checked="" type="radio"/> d) Ne znam	

II blok pitanja – Osnovne informacije o ispitaniku			
II/1	Pol ispitanika:	<input checked="" type="radio"/> a) ženski <input type="radio"/> b) muški	
II/2	Nivo obrazovanja roditelja:	a) majka: - visoko <input checked="" type="radio"/> - srednje - osnovno	b) otac: - visoko <input checked="" type="radio"/> - srednje - osnovno
II/3	Ukupna procečna ocena u prethodno završenom razredu srednjoškolske ustanove:	a) Upisati ukupnu prosečnu ocenu: 3	Procečna ocena iz predmeta matematika u prethodno završenom razredu srednjoškolske ustanove: b) Upisati ukupnu prosečnu ocenu iz matematike: 2

Hvala Vam na saradnji!

Biografija autora

Ivica Stanković (26.02.1977, Bor) je stekao zvanje master drugog stepena na katedri za Informacione tehnologije i sisteme (smer: Bezbednost informacija, prosečna ocena 9,67, tema: „*Simulacija i dizajn eksperimenta odbrane umreženog sistema sa primenom kriptografije*“) Fakulteta Informacionih tehnologija Metropolitan Univerziteta u Beogradu. Pored ovog mastera, ima i master drugog stepena na katedri Metalurgije Tehničkog fakulteta u Boru Univerziteta u Beogradu (prosečna ocena 8,05). Radio je na poslovima:

2012 – : Asistent u nastavi:

Fakultet za poslovne studije i pravo, Univerzitet „Union – Nikola Tesla“ u Beogradu
Fakultet za informacione tehnologije i inženjerstvo, Univerzitet „Union – Nikola Tesla“ u Beogradu

2012 – 2015: Asistent i glavni predavač:

Asistent i glavni predavač u nastavi u udruženom fakultetskom programu Fakulteta za poslovne studije i pravo Univerziteta „Union – Nikola Tesla“ u Beogradu i Linkoln Univerziteta (Oakland, CA, USA, ogranak u Beogradu). Celokupna nastava se odvijala na engleskom jeziku.

2010 – 2012: Sistem inženjer:

IT Systems, Brooklyn, NY, USA lokalna tehnička podrška u Srbiji.

Imao je sledeća zaduženja i odgovornosti:

- Održavanje Veb servera
- Pisanje šel skripti
- Korišćenje i razvoj kastomiziranih alata za proveru, praćenje i održavanje bezbednosti mrežnog saobraćaja u sajber prostoru
- Priprema tehničke dokumentacije
- Prevod tehničke dokumentacije sa engleskog na srpski i obrnuto
- Priprema i predstavljanje odgovarajućih poslovnih prezentacija za klijente

2007-2010: Inženjer metalurgije:

Metali 1992 D.O.O. Beograd, Srbija.

Imao je sledeća zaduženja i odgovornosti:

- Vodeći inženjer u metalurškim procesima
- Razvoj savremenih alata iz oblasti metalurgije
- Zaštita životne sredine, kontrola hazardnih situacija i koordinacija sa ministarstvom za zaštitu životne sredine
- Učešće u kreiranju poslovne politike preduzeća
- Učešće u pregovorima sa klijentima

2003-2006: saradniku u laboratoriji za hemijsko-tehničku kontrolu

Institut za rudarstvo i metalurgiju, Bor, Srbija

Imao je sledeća zaduženja i odgovornosti:

- Kreiranje eksperimenata za fizičku analizu jedinjenja bakra
- Fizička analiza jedinjenja bakra
- Testiranje jedinjenja bakra na čistoću

2000-2003: Držanje kurseva iz programskih jezika: C, C++ i Java

Školske 2012/13 godine upisao je doktorske studije na Fakultetu za informacione tehnologije i inženjerstvo, studijski program Inženjerski menadžment. Tokom doktorskih studija položio je 5 ispita sa prosečnom ocenom 9,60 (devet i 60/100) i odbranio Završni ispit u formi seminarsko g rada sa ocenom 10 (deset) pod nazivom “Baze znanja i savremene tehnike pretraživanja velikih skupova podataka.“.