Predikcija i rangiranje uspjeha studenata na sveučilištu na osnovu rezultata državne mature, uspjeha iz srednje škole i drugih pokazatelja

Martić, Dominik

MEĐIMUSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVO

DOMINIK MARTIĆ

PREDIKCIJA I RANGIRANJE USPJEHA STUDENATA NA SVEUČILIŠTU NA OSNOVI REZULTATA DRŽAVNE MATURE, USPJEHA IZ SREDNJE ŠKOLE I DRUGIH POKAZATELJA

ZAVRŠNI RAD

ČAKOVEC, 2021.

MEĐIMUSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVO

DOMINIK MARTIĆ

PREDIKCIJA I RANGIRANJE USPJEHA STUDENATA NA SVEUČILIŠTU NA OSNOVI REZULTATA DRŽAVNE MATURE, USPJEHA IZ SREDNJE ŠKOLE I DRUGIH POKAZATELJA

PREDICTION AND RANKING OF UNIVERSITY STUDENTS BY SUCCESS BASED ON RESULTS OF STATE SECONDARY SCHOOL LEAVING EXAMINATION, MIDDLE SCHOOL SUCCESS AND OTHER FACTORS

ZAVRŠNI RAD

Mentor:
mr. sc. Željko Knok, v. pred.

ČAKOVEC, 2021.

ZAHVALA

Zahvaljujem svome mentoru mr. sc. Željku Knoku na usmjeravanju i vođenju prilikom pisanja. Svojim mi je savjetima omogućio da temu ovog završnog rada sročim u strukturiranu cjelinu.

Također, zahvaljujem svim djelatnicima Međimurskog veleučilišta u Čakovcu koji su nam prenijeli znanja unutar struke. Posebno želim zahvaliti svojoj obitelji koja me podupirala tijekom studiranja u drugom gradu i omogućila mi sva novostečena poznanstva i znanja.

SAŽETAK

*Tema ovog završnog rada zahtijeva stvaranje seta podataka i filtriranje samo bitnih podataka kako bi se mogla napraviti predikcija budućih ocjena. Jako je bitno da podatci budu povezani kako bi se dobila smislena predikcija podataka. Podatci, koji su se koristili kao set podataka, izgenerirani su putem Python skriptea. Smatra se da je pet smjerova sa 110 studenata po smjeru dovoljno kako bi se napravila predikcija. Uz izgenerirane podatke postojali su zastarjeli podatci o studentima računarstva i menadžmenta. U predviđanje je sveukupno uključen 741 student. Nakon što se stvori set podataka, podatke je potrebno prikazati unutar korisničkog sučelja kako bi se moglo s njima raditi te kako bi se mogli izabrati studenti za koje će se napraviti predikcija ocjena, što je ujedno i tema ovog rada. U tu je svrhu korišten alat Tkinter. Tkinterom se Python veže za Tk GUI (*engl. *graphical user interface) alat i ujedno je standardno sučelje za Pythonov GUI. Predviđanjem se može odrediti ocjena studenata u narednim godinama studiranja na osnovi danih podataka. Postoji mogućnost importiranja podataka iz CSV (*engl. *comma-separated values) formata datoteke u tablicu unutar grafičkog sučelja te, također, izvoz filtrirane ili nefiltrirane liste podataka odabranih studenata u CSV formatu. Podatci, koji se koriste, su bodovi iz srednje škole i bodovi s državne mature (ukupno 1000 tisuću bodova). Kao drugi parametar uzimaju se ocjene s prve godine fakulteta.*

*Znanost o podatcima (*engl. *data science) uz pomoć metoda i algoritama izvlači informacije o danim podatcima. Znanost o podatcima uključuje prikupljanje, obradu, analizu i održavanje podataka.*

Ključne riječi: Python, prediktivna analitika, znanost o podatcima, CSV, Tkinter

Sadržaj

1. OPIS

U ovom se završnom radu, na osnovi usporedbe rezultata ispita državne mature, ocjena iz srednje škole, ocjena iz odabranih predmeta na prvoj godini studija, prediktivnim modelom uz pomoć linearne regresije predviđa uspjeh u nadolazećim godinama studija. Podatci će se analizirati, sortirati i obraditi u digitalnom obliku. Za potrebe rada korišteno je korisničko sučelje u okviru alata otvorenog koda Anaconda te dodatni alati, kao što su programski jezik Python te moduli - biblioteke koje su sastavni dio programskog jezika.

2. UVOD

2.1. Strojno učenje

Strojno je učenje grana umjetne inteligencije koja se bavi oblikovanjem algoritama koji u pravilu poboljšavaju svoju učinkovitost pomoću empirijskih podataka. Zbog velikih mogućnosti strojno učenje jedno je od atraktivnijih područja računarske znanosti. Neke od mogućnosti uključuju raspoznavanje uzoraka, dubinsku analizu podataka, robotiku, računalni vid, bioinformatiku i računalnu lingvistiku.

2.2. Prediktivna analitika

Prediktivna analitika obuhvaća razne statističke tehnike. Strojnim se učenjem, prediktivnim modeliranjem i rudarenjem podataka, uz pomoć podataka iz prošlosti i trenutnih podataka, može napraviti predviđanje budućih/nepoznatih podataka. Prediktivna je analitika nadogradnja za poslovnu analitiku (engl. *business intelligence*). Možemo dobiti uvid u uzorke i odnose među podatcima te procijeniti vjerojatnost pojave određenog događaja u budućnosti.

Svaka poslovna aktivnost, koju ponavljamo puno puta, također je predmet predviđanja. Možemo prikupljati različite podatke o potrošaču, ponudi i samom prodajnom procesu. Na osnovi sakupljenih podataka možemo za nove ponude unaprijed pokušati ocijeniti uspješnost njihova provođenja.

Moguće je izračunati životnu vrijednost svakog potrošača (engl. *life time value*) i dobit koju možemo potencijalno ostvariti. Potrošača ne segmentiramo na osnovi vjerojatnosti odlaska, već ovisno o tržišnom potencijalu i profitabilnosti koju možemo ostvariti. Ti nam rezultati pomažu u definiranju strategije nošenja s odlaskom potrošača.

3. SOFTWARE

3.1. Anaconda

S preko 25 milijuna korisnika *open-source* distribucija Anaconda najlakši je način za korištenje programskih jezika Python i R u znanstveno svrhe kojima je cilj pojednostaviti upotrebu paketa. Distribucija uključuje pakete pogodne za podatkovnu znanost, aplikacije za strojno učenje ili, na primjer, prediktivnu analizu koja je potrebna za ovaj projekt i mnoge ostale. Anaconda je dostupna na Windowsu, macOS-u i Linuxu. Uz pomoć conda-install komande moguće je koristiti preko 7500 tisuća *open-source* Conda, R, Python i puno drugih paketa. Anaconda je svojim radom stekla povjerenje brojnih svjetski poznatih tvrtki diljem svijeta. Neke od njih su: National Grid, Samsung, BMW, Columbia University, HSBC Bank, US Army Corps and Engineers i dr.

SLIKA 1. ANACONDA LOGO [1]

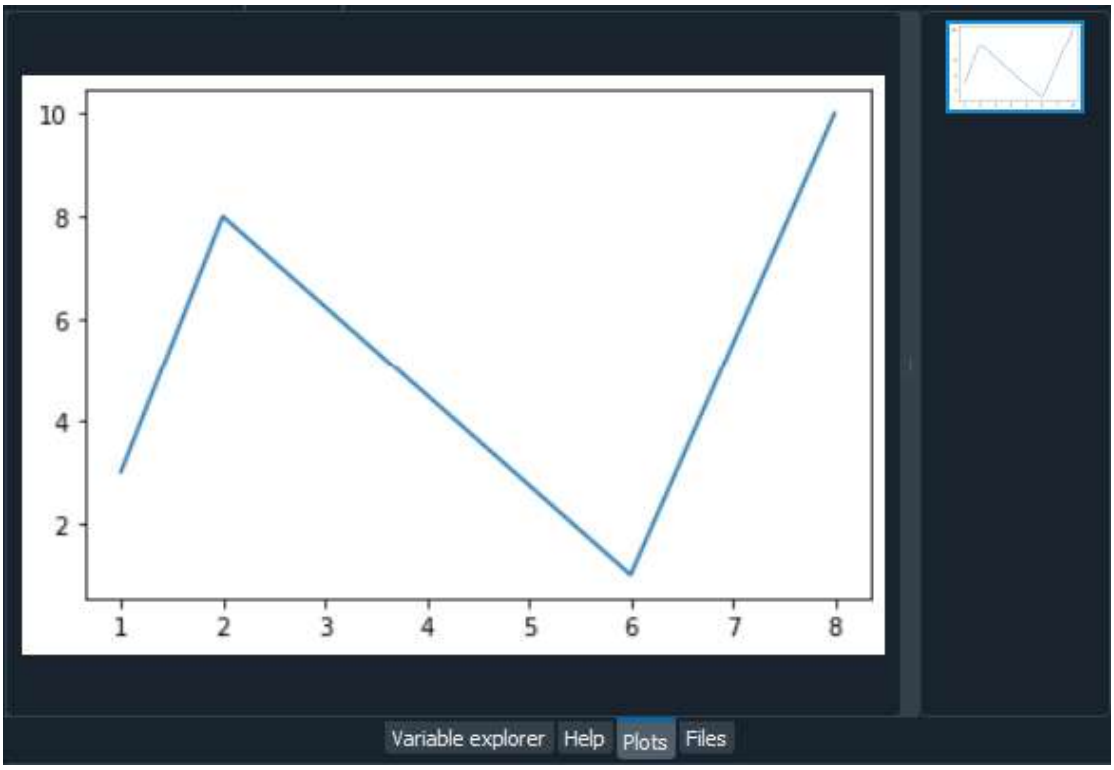
SLIKA 2. SPYDER LOGO [2]

3.2. Spyder

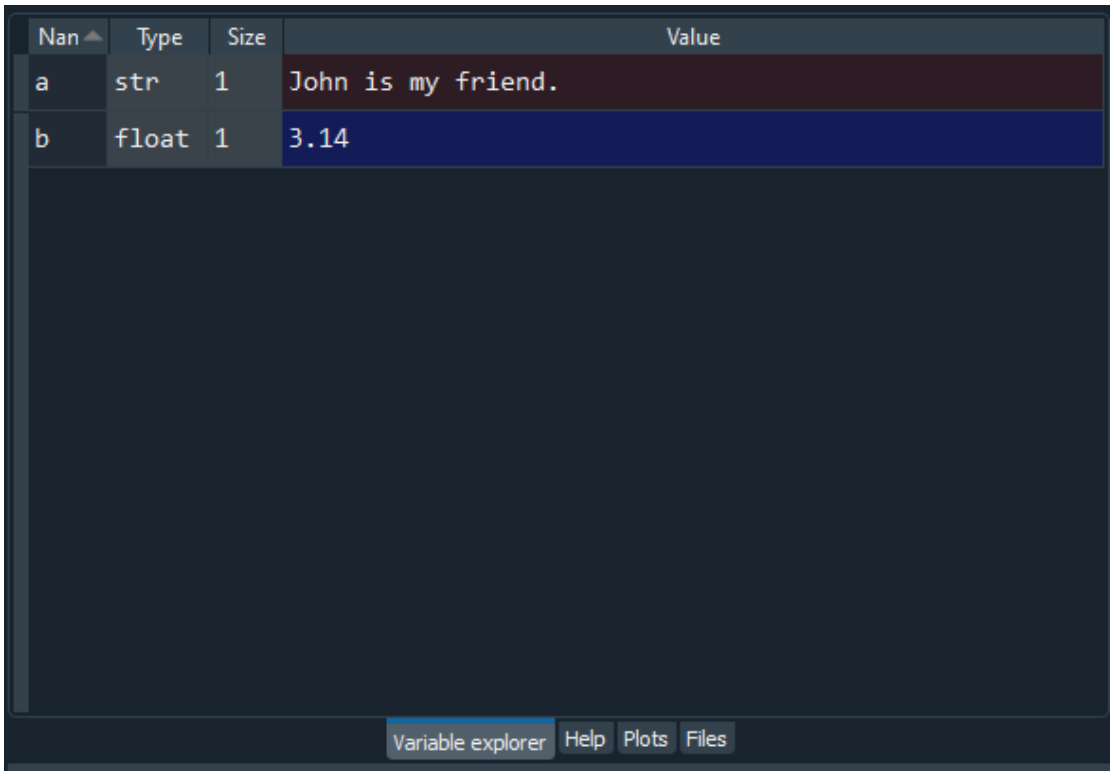
Spyder, okruženje za znanstveni razvoj u pyhtonu, besplatno je integrirano razvojno okruženje (IDE), uključeno s Anacondom. Sadrži kombinaciju napredne funkcije uređivanja, analize, uklanjanja pogrešaka i profiliranja sveobuhvatnog razvojnog alata s istraživanjem podataka, interaktivnim izvršavanjem, dubinskim pregledom i lijepim mogućnostima vizualizacije znanstvenog paketa.

Variable explorer i Plots prozori u Spyderu jako su korisni jer možemo unutar Variable explorera vidjeti sve naše varijable koje smo stvorili pisanjem našeg koda nakon što pokrenemo program i možemo vidjeti tip podataka te podatke zapisane unutar koda. Unutar Plots prozora možemo grafički prikazati različite tablice ili prikaze koji nam mogu pomoći pri vizualizaciji onoga što radimo.

Napravljena su dva jednostavna prikaza ovih mogućnosti unutar Spyder okruženja kako bi se prikazale mogućnosti Plots i Variable explorer prozora. Unutar Variable explorera može se očitati tip podatka, naziv varijable, veličina te sama vrijednost upisana u varijablu.



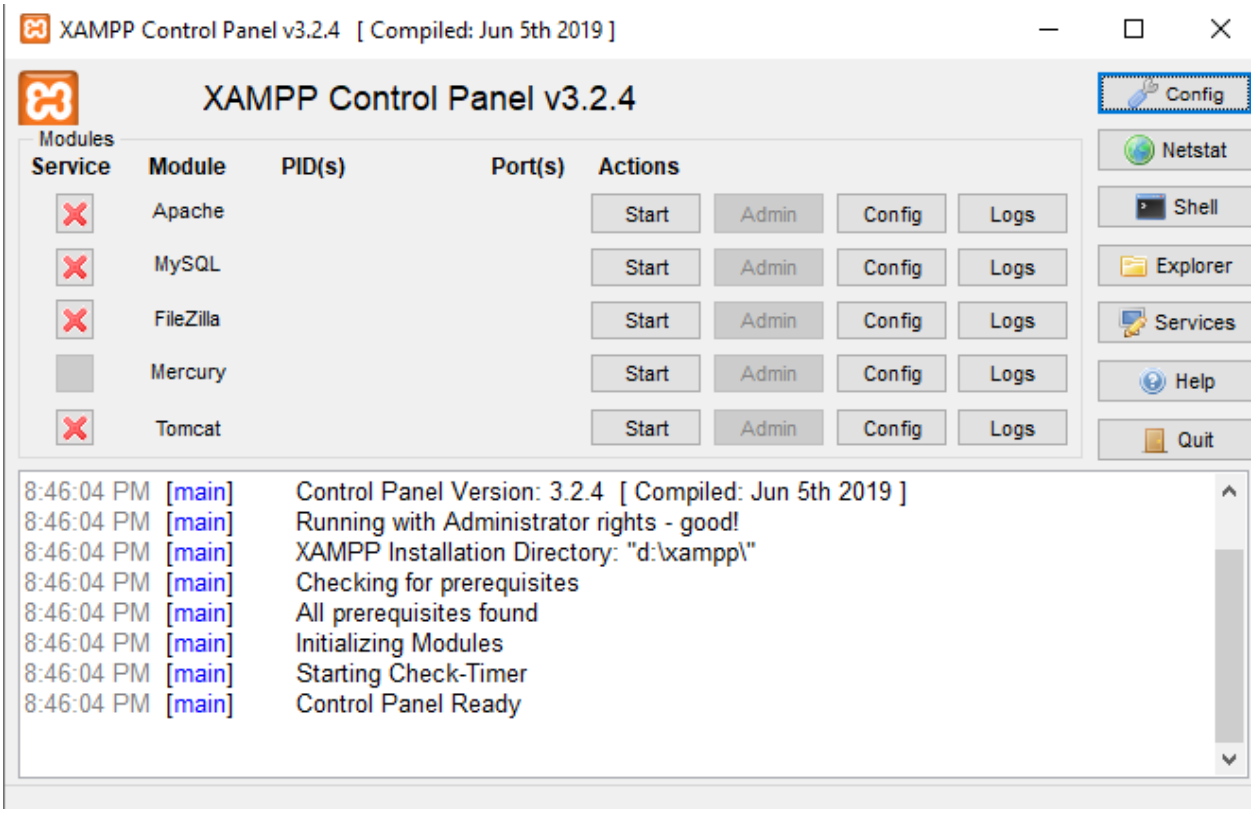
SLIKA 3. PRIKAZ JEDNOSTAVNOG GRAFA UNUTAR PLOTS PROZORA [3]



SLIKA 4. PRIKAZ JEDNOSTAVNIH VARIJABLI S PRIPADAJUĆIM OZNAKAMA UNUTAR VARIABLE EXPLORER PROZORA [4]

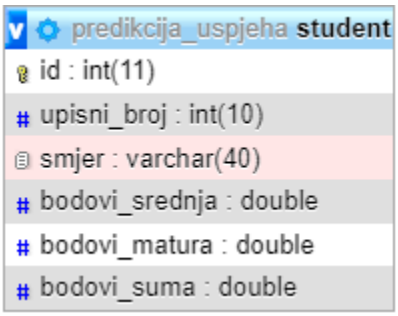
3.3. XAMPP

XAMPP je paket otvorenog koda koji se jako jednostavno instalira i koristi. XAMPP je korišten u sklopu ovog rada za spremanje podataka o studentima bez ocjena kako bi se lakše radilo s podatcima. Korišteni su servisi Apache i MySQL za rad s bazom podataka.



SLIKA 5. APLIKACIJA XAMPP [5]

Baza je vrlo jednostavna te ima jednu tablicu koja se sastoji od šest komponenti.



SLIKA 6. BAZA PODATAKA UNUTAR HTTP://LOCALHOST/PHPMYADMIN/ U DIZAJNER POGLEDU [6]

4. PYTHON

Python je objektno orijentiran programski jezik otvorenog koda (engl. *open-source software*). Sadrži module, klase i jako dinamične tipove podataka. Vrlo je moćan, koncizan i jednostavan za čitanje. Python, također, sadrži „garbage collection" sustav tako da programer ne mora previše brinuti o memoriji.

SLIKA 7. PYTHON LOGO [7]

Python programski jezik dobio je ime po televizijskoj seriji Monty Python's Flying Circus. Glavna je filozofija ovog programskog jezika sažeta u nekoliko rečenica:

- Lijepo je bolje nego ružno.
- Eksplicitno je bolje nego implicitno.
- Jednostavno je bolje nego komplicirano.
- Kompleksno je bolje nego komplicirano.
- Čitljivost se računa.

5. MODULI - BIBLIOTEKE

5.1. Numpy

NumPy je python biblioteka koja se koristi za rad s nizovima. Također, ima funkcije za rad u domeni linearne algebre, furijeove transformacije i matrice. To je projekt otvorenog koda i može se slobodno koristiti. Kratica NumPy označava Numerical Python. U Pythonu imamo liste koje služe za nizove podataka, ali se one sporo obrađuju. NumPy ima mogućnost osiguranja rada s nizovima koji su i do 50 puta brži od tradicionalnih lista Pythona. Objekt arraya u NumPyju naziva se ndarray te nudi puno pomoćnih funkcija koje čine rad s ndarrayem vrlo jednostavnim. Nizovi se vrlo često koriste u znanosti o podatcima, gdje su vrlo važni resursi i brzina.

5.2. Scikit-learn

Scikit-learn je vjerojatno najkorisnija biblioteka za računalno učenje u pythonu. Sklearn biblioteka sadrži niz efikasnih alata za računalno učenje i statističko modeliranje, uključujući klasifikaciju, regresiju, klasterizaciju i smanjenje dimenzionalnosti. Sklearn se koristi za izradu machine-learning modela. Ova nam biblioteka, također, omogućava prikaz postotka točnosti modela i koorelacije podataka, tj. vezu i jačinu te veze.

5.3.Pandas

Pandas je Python paket koji omogućava brze i fleksibilne strukture podataka dizajnirane za rad s relacijskim ili označenim podatcima (engl. *labeled data*) koji su jednostavni i intuitivni. Cilj je pandas paketa biti temeljni paket visoke razine za analizu praktičnih podataka iz svakodnevnice. Njihova je težnja postati najjači i fleksibilan open-source alat za podatkovnu analizu/manipulacijski alat dostupan su svim jezicima. Pandas će se koristiti za učitavanje csv datoteke s podatcima o studentima.

Pandas je koristan za različite tipove podataka:

- podatci zapisani u tablicu s heterogeno tipiziranim stupcima, kao što je SQL tablica ili Excel proračunska tablica
- uređeni i neuređeni podatci (ne nužno fiksne frekvencije) vremenskih serija

- podatci proizvoljne matrice (homogeno upisani ili heterogeni) s oznakama redaka i stupaca
- bilo koji drugi oblik promatračkih/statističkih podataka; podatci ne moraju biti označeni da bi se smjestili u pandas strukturu podataka.

Dvije primarne pandas strukture podataka, Series (1-dimenzionalne) i DataFrame (2-dimenzionalne), obrađuju veliku većinu tipičnih slučajeva upotrebe u financijama, statistici, društvenim znanostima i mnogim područjima inženjerstva. Za R korisnike DataFrame pruža sve ono što R data.frame pruža i još mnogo toga. Pandas je izgrađen na vrhu NumPyja i namijenjen je dobroj integraciji unutar znanstvenog računalnog okruženja s mnogim drugim bibliotekama trećih strana.

Još neke stvari koje pandas može obavljati uključuju:

- jednostavno rukovanje podatcima koji nedostaju (predstavljeni kao NaN) u podatcima s pokretnim zarezom, kao i s podatcima bez pomične točke
- promjenjivost veličine; stupci se mogu umetnuti i izbrisati iz DataFramea i objekata veće dimenzije
- automatsko i eksplicitno poravnanje podataka; objekti se mogu poravnati sa skupom oznaka ili korisnik može jednostavno zanemariti oznake i dopustiti da Series i DataFrame automatski poravnaju podatke za vas u izračunima
- moćnu, fleksibilnu grupu po funkcionalnosti za izvođenje operacija podijeljeno – primijeniti - kombinirati na skupovima podataka, kako za prikupljanje tako i za transformiranje podataka
- olakšano pretvaranje razlupanih, različito indeksiranih podataka u druge Python i NumPy podatkovne strukture u DataFrame objekte
- inteligentno rezanje na temelju oznaka, otmjeno indeksiranje i podskup velikih skupova podataka
- intuitivno spajanje i spajanje skupova podataka
- fleksibilno preoblikovanje i okretanje skupova podataka

- hijerarhijsko označavanje osi (moguće je imati više oznaka)
- robusne IO alate za učitavanje podataka iz ravnih datoteka (CSV i s razgraničenjem), Excel datoteka, baza podataka te spremanje/učitavanje podataka iz ultra brzog HDF5 formata.
- funkcionalnost specifičnu za vremenske serije; generiranje datumskog raspona i pretvorba frekvencije, pomicanje datuma i kašnjenje (engl. *lagging*).

Mnogi su od ovih principa ovdje kako bi riješili nedostatke koji se često događaju korištenjem drugih jezika/okruženja za znanstveno istraživanje. Za znanstvenike koji rade s podatcima obično je posao podijeljen u više faza: mijenjanje i "čišćenje" podataka, njihova analiza/modeliranje, a zatim organiziranje rezultata analize u oblik prikladan za crtanje ili tablični prikaz. Pandas je idealan alat za sve ove zadatke.

5.4. Tkinter

Tkinter paket standardno je Python sučelje za GUI aplikacije. Objekti unutar ovoga paketa koriste se za prikaz tablica, gumba, ćelija za upisivanje koji su korišteni za prikaz podataka unutar istih.

5.5. Matplotlib

Matplotlib paket omogućuje, uz pomoć Python programskog jezika i njegove matematičke ekstenzije NumPy, stvaranje grafičkog prikaza podataka kako bi se bolje razumjeli podatci s kojima se radi ili prikazali rezultati.

5.6. Alati za predikciju

Unutar programskog jezika Python postoji više mogućnosti za predikciju podataka. Moguće je izraditi predikivni model tako da prvo skupimo i pripremimo podatke koje želimo analizirati, a zatim te iste podatke podijelimo na testne podatke i podatke za treniranje. Jedna od biblioteka koja to omogućava je Scikit-learn. Neke od metoda koje se mogu koristiti su: model slučajnih šuma, stablo odluke, strojno učenje i linearna regresija.

Strojno je učenje metoda u kojoj rezultate dobivamo puštanjem podataka putem odabranog algoritma. Nakon što se algoritam istrenira setom podataka, novim podatcima može predvidjeti određeni ishod. Najveće kompanije svijeta, kao što su Google i Amazon, koriste se strojnim učenjem. Jedan je dio strojnog učenja i umjetna inteligencija (engl. *Artificial intelligence*). Umjetna inteligencija omogućuje stroju da uči iz prijašnjih podataka bez programiranja. Cilj je umjetne inteligencije napraviti pametno računalo koje će moći rješavati složene probleme. Neke implementacije strojnog učenja koriste podatke i neuralne mreže na način da se ponašaju kao mozak.

SLIKA 8. STROJNO UČENJE I UMJETNA INTELIGENCIJA[8]

Linearna je regresija najčešći oblik regresije. U linearnoj se regresiji pronalazi pravac (ili složenija linearna kombinacija) koji najbolje odgovara podatcima prema određenim kriterijima. Jedan je od najranije upotrebljavanih kriterija metoda najmanjih kvadrata koju je prvi objavio Carl Friedrich Gauss. Regresijska analiza omogućava procjenu uvjetnog očekivanja ovisne varijable kada neovisne varijable preuzmu određeni skup podataka. Linearna se regresija prvenstveno koristi za predviđanje te je blisko povezana sa strojnim učenjem. U nekim se situacijama regresijska analiza može upotrijebiti za zaključivanje uzročno-posljedičnih odnosa

izmedu neovisnih i ovisnih varijabli. U ovom je radu ujedno korištena linearna regresija kako bi se predvidjele ocjene studenata.

Slučajna je šuma metoda kreiranja mnoštva stabala odlučivanja u vremenu treninga. Može se, također, koristiti za regresiju i klasifikaciju. Slučajne šume većinom nadmašuju stabla odlučivanja, ali im je točnost niža od točnosti stabala s pojačanim nagibom.

SLIKA 9. SLUČAJNA ŠUMA [9]

Stablo odluka jedna je od metoda koje se koriste za klasifikaciju i regresiju. Cilj je stabla odlučivanja izrada modela koji predviđa vrijednost varijable učenjem jednostavnih pravila uzetih iz podataka. Stabla odlučivanja mogu biti vizualizirana pa ih je jednostavno razumjeti.

Elastična mreža (engl. *Elastic net*) model je linearne regresije koji tijekom treniranja uključuje kazne. Jedna je popularna kazna kažnjavanje modela na temelju zbroja vrijednosti kvadrata koeficijenta. To se zove L2 kazna. L2 kazna minimizira veličinu svih koeficijenata, iako sprječava uklanjanje koeficijenata iz modela. Druga je popularna kazna kažnjavanje modela na temelju zbroja vrijednosti apsolutnih koeficijenata. To se zove L1 kazna. L1 kazna minimizira veličinu svih koeficijenata i dopušta da se neki koeficijenti minimiziraju na nulu, što uklanja prediktor iz modela.

Pojačavanje gradijenta (engl. *Gradient boosting*) tehnika je strojnog učenja za regresiju i klasifikaciju koja proizvodi model predviđanja u obliku skupa slabih modela predviđanja, obično stabala odluka. Kada je stablo odluka slabiji učenik, rezultirajući algoritam naziva se stablo s pojačanim gradijentom i ono obično nadmašuje slučajnu šumu.

Algoritam stroja potpornih vektora (engl. *Support Vector Machines (SVM)*) nadzirani je model učenja s pripadajućim algoritmima učenja koji analiziraju podatke za klasifikaciju i regresijsku analizu. S obzirom na skup primjera za treniranje, od kojih je svaki označen kao da pripada jednoj od dvije kategorije, SVM algoritam za treniranje gradi model koji dodjeljuje nove primjere jednoj ili drugoj kategoriji. SVM preslikava primjere treninga na točke u prostoru kako bi maksimizirao širinu između dviju kategorija. Novi se primjeri zatim mapiraju u isti prostor. Smještaju se u određenu kategoriju s obzirom na to na koju stranu rupe između kategorija padnu. Osim izvođenja linearne klasifikacije, SVM-ovi mogu učinkovito izvesti nelinearnu klasifikaciju koristeći kernel trik, implicitno mapirajući svoje ulaze u visokodimenzionalne prostore značajki.

Ekstremno nasumična stabla (engl. *Extra Trees*) algoritam je za strojno učenje cjelina koji kombinira predviđanja iz mnogih stabala odluka. Povezan je sa široko korištenim algoritmom slučajne šume. Često može postići jednako dobre ili bolje performanse od algoritma slučajne šume, iako koristi jednostavniji algoritam za konstruiranje stabala odluka koja se koriste kao članovi cjelina.

6. PROGRAM ZA PREDIKCIJU

Unutar programa Spyder prvo je bilo potrebno kreirati projekt unutar kojeg će biti svi podatci o studentima s njihovim ocjenama te python skripte u kojima je pisan kod.

6.1. Podatci o studentima

Početni podatci, koji su korišteni za rad, zastarjeli su podatci o studentima računarstva i menadžmenta. Unutar datoteka CSV tipa pohranjeni su podatci u sljedećem obliku:

Plasman	Početni plasman	Izbor	Bodovi iz škole	Bodovi s državne mature	Bodovi s dodatnih provjera	Bodovi za dodatna postignuća
1	15	1	382	250.7899933	0	0
2	17	1	374	255.1300049	0	0
3	19	1	436	190.0800018	0	0
4	21	1	385	235.75	0	0
5	22	1	428	188.8200073	0	0
6	28	3	375	230.9600067	0	0
7	29	2	383	222.0299988	0	0
8	34	1	358	239.5299988	0	0
9	36	2	379	216	0	0
10	40	1	438	149.1900024	0	0
11	48	1	387	188.4499969	0	0
12	49	1	374	201.25	0	0
13	51	1	391	182.5399933	0	0
14	52	1	414	159.4199982	0	0
15	54	1	348	220.8099976	0	0
16	55	1	377	190.7899933	0	0
17	57	1	378	189.2299957	0	0
18	58	4	381	186.1399994	0	0
19	59	1	366	200.4299927	0	0

Bodovi ukupno	Pravo prednosti	Ima preduvjete	Pravo upisa	Upisni broj	Datum upisa	Pravo upisa	Upisan na ostale
632,8	Ne	Da	Da	12320		Ne	Ne
629,1	Ne	Da	Da	4741		Ne	Ne
626,1	Ne	Da	Da	25512		Ne	Ne
620,8	Ne	Da	Da	24098		Ne	Ne
616,8	Ne	Da	Da	9326		Ne	Ne
606	Ne	Da	Da	18211		Ne	Ne
605	Ne	Da	Da	15626		Ne	Ne
597,5	Ne	Da	Da	21188		Ne	Ne
595	Ne	Da	Da	11524		Ne	Ne
587,2	Ne	Da	Da	567		Ne	Ne
575,5	Ne	Da	Da	7004		Ne	Ne
575,3	Ne	Da	Da	14243		Ne	Ne
573,5	Ne	Da	Da	27224		Ne	Ne
573,4	Ne	Da	Da	9983		Ne	Ne
568,8	Ne	Da	Da	1162		Ne	Ne
567,8	Ne	Da	Da	14544		Ne	Ne
567,2	Ne	Da	Da	11130		Ne	Ne
567,1	Ne	Da	Da	3927		Ne	Ne
566,4	Ne	Da	Da	18688		Ne	Ne

SLIKA 10. RAČUNARSTVO REDOVNI STUDENTI [10]

Podatke je potrebno filtrirati na način da se uzmu samo oni podatci koji su potrebni za predikciju. Pri generiranju dodatnih podataka o studentima za druge smjerove, kako bi se napravio *dataset* koji se može koristiti za predikciju ocjena u narednim godinama, generirani su upisni broj, bodovi iz škole i bodovi s državne mature kao podatci koji će se koristiti za predikciju. Upisni se broj koristi kao unikatni identifikator svakog studenta na pojedinom smjeru. Kreirano je pet dodatnih datoteka uz zastarjele podatke kako bi se proširio set podataka. Po svakoj CSV datoteci s podatcima o studentima bit će 110 studenata s jedinstvenim upisnim brojem za svoj smjer, s obzirom na to da se pretpostavlja kako svaki fakultet može imati iste

vrijednosti za upisne brojeve studenata, ali ne može imati dva ista upisna broja za dva različita studenta.

Napravljena je python skripta za svaki od pojedinih smjerova, tako što su prvo definirani biblioteke i moduli koji će se koristiti.

```
import csv
import pandas as pd
import random
```

KOD 1. BIBLIOTEKE I MODULI UNUTAR CSV GENERATORA ZA PODATKE O STUDENTIMA [1]

CSV modul implementira klase za čitanje i pisanje tabličnih podataka u CSV formatu. Pandas se koristi kako bi se kreirao DataFrame za lakše manipuliranje podatcima i kreiranje CSV datoteke u kojoj će biti zapisani svi podatci o studentima. Modul *random* implementira generatore pseudoslučajnih brojeva za različite distribucije.

```
kolone=[
"id",
"Upisni broj",
"Smjer",
"Bodovi iz skole",
"Bodovi s drzavne mature",
"Bodovi ukupno"
]
result = pd.DataFrame(columns=kolone)
#Prazne liste
lista=[]
bodSkola=[]
bodMatura=[]
```

KOD 2. KREIRANJE PRAZNIH LISTA ZA UPIS PODATAKA[2]

Unutar svake liste potrebno je napuniti podatke odgovarajućim brojevima. Pretpostavka je da ćemo od ukupnih tisuću bodova, koji se mogu ostvariti od srednje škole i mature za upis na

fakultet, izgubiti barem neki dio. Okvirna je pretpostavka da je maksimalan broj bodova 800, a minimalan 400 (približno stvarnim podatcima).

```
for i in range(110):
    r=random.randint(1,30000)
    if r not in lista:
        lista.append(r)

for i in lista:
    result['Upisni broj'] = i

for i in range(110):
    r=random.randint(220,520)
    bodSkola.append(r)

for i in range(110):
    r=random.randint(180,320)
    bodMatura.append(r)

result['Upisni broj'] = lista
result['Smjer'] = 'Ekonomija'
result['Bodovi iz skole'] = bodSkola
result['Bodovi s drzavne mature'] = bodMatura
```

KOD 3. GENERIRANJE PODATAKA I PUNJENJE PRIPADAJUĆIH LISTI [3]

Svi podatci generirani su uz pomoć modula *random* i njegove ugrađene funkcije *random.randint(a, b)* koja vraća brojeve između a i b. Pri izradi ovoga rada smatralo se da je za upisni broj dovoljno bilo 30 000 kao viši broj kako bi se dobili raznovrsni podatci, a ujedno i ne preveliki brojevi. Također, pri kreiranju podataka unutar for-petlje postoji jedan uvjet. On provjerava postoji li takav broj unutar liste kako ne bi došlo do duplikata pri kreiranju upisnih brojeva.

Za kreiranje bodova iz srednje škole i bodova s državne mature uzeti su brojevi za nijansu veći od brojeva koji su dobiveni unutar zastarjelih podataka kako bi se kreirao što kvalitetniji set podataka.

U konkretnom primjeru generiran je set podataka za fakultet (ekonomski smjer). Unutar pandas DataFramea u svaki odgovarajući stupac učitani su podatci iz kreiranih listi; jedina je iznimka smjer koji je ručno podešen svim redcima na 'Ekonomija'. Takvi setovi podataka kreirani su za podatke o studentima sljedećih smjerova: ekonomije, prometa, informatike, sestrinstva i graditeljstva.

```
for i in result:
            result['Bodovi ukupno'] = result['Bodovi iz skole'] +
            result['Bodovi s drzavne mature']
result = result.sort_values(by=['Bodovi ukupno'], ascending = False)
```

KOD 4. SUMA BODOVA I SILAZNO SORTIIRANJE PREMA SUMI [4]

Za svaki su redak zbrojeni podatci o bodovima iz skole i podatcima s državne mature te naredno sortirani po vrijednosti ukupnih bodova uz pomoć pandas funkcije sort_values() i parametra ascending postavljenog na "False" kako bi se dobila lista sortirana silazno.

```
with open('EKO.csv', 'w', encoding='UTF8', newline='') as f:
     writer = csv.writer(f)
     #rows
     writer.writerows(result)
export_csv = result.to_csv (r'C:\Users\Dominik\Desktop\Predikcija
uspjeha studenata\EKO.csv', header=True, index=False)
```

KOD 5. SUMA BODOVA I SILAZNO SORTIIRANJE PREMA SUMI [5]

Uz pomoć ugrađenih funkcija unutar modula CSV podatci su vrlo lako uneseni u pandas DataFrame i upisani u CSV datoteke. Konačno, podatci upisani u CSV datoteku, uz pomoć ugrađene funkcije .to_csv(), izvoze se u postavljeni folder pod nazivom EKO.csv.

6.2. Podatci o ocjenama studenata

Potrebno je, također, izgenerirati ocjene studenata. Kako bi ova predikcija imala smisla, bilo je potrebno generirati ocjene prema bodovima iz srednje škole i bodovima s državne mature. Za

svaki set podataka o studentima kreirana je CSV datoteka koja sadrži ocjene predmeta s prve godine fakulteta prema postignutim bodovima do srednje škole. Uz svaki redak s ocjenama unutar CSV datoteke kreiran je i stupac gdje se prikazuje upisni broj studenta kako bi se ocjene mogle povezati sa studentom. Kao i za podatke o studentima, potrebno je prvo definirati module i biblioteke koje će se koristiti za izradu generatora CSV datoteke s ocjenama. Kako bi se lakše manipuliralo podatcima, u ovom se dijelu koristio mydb.connector pomoću kojega se spajamo na MySQL bazu podataka koju možemo koristiti zahvaljujući programu XAMPP.

```
import csv

import numpy as np

import pandas as pd

import mysql.connector

mydb = mysql.connector.connect(host='localhost',
user='root',passwd='',
database='predikcija_uspjeha',auth_plugin='mysql_native_password')

cursor = mydb.cursor()
```

KOD 6. SPAJANJE NA BAZU PODATAKA I DEFINIRANJE BIBLIOTEKA I MODULA [6]

Numpy biblioteka koristi se za rad s brojevima kako bi podatcima bili povezani s bodovima iz srednje škole i bodovima s državne mature. Također, implementirani su već spomenuti CSV i pandas moduli/biblioteke.

Spajanje na bazu vrši se prvom naredbom nakon definiranja biblioteka/modula gdje je potrebno navesti “domaćina”, u ovom je slučaju to localhost, korisničko ime i odgovarajuća lozinka koji se koriste za rad s bazom, te samo ime kreirane baze. Također, potrebno je kreirati novi primjerak klase “cursor” koji služi za pokretanje SQL naredbi.

```
bodovi_uk=[]

query = "SELECT id,upisni_broj,bodovi_suma FROM student WHERE
smjer='Ekonomija'"

cursor.execute(query)
```

```
rows = cursor.fetchall()

bodovi_uk = pd.DataFrame(rows,columns=('id','Upisni broj','Bodovi
suma'))

upisni_broj = bodovi_uk['Upisni broj']

best_value = abs(int((len(bodovi_uk)-(1/5*len(bodovi_uk)))-
len(bodovi_uk)))

mid_value = abs(int(len(bodovi_uk)-(3/5*len(bodovi_uk))))+best_value

lower_value = abs(int((len(bodovi_uk))-(best_value+mid_value)))
```

KOD 7. UČITAVANJE PODATAKA IZ BAZE I PODJELA NA TRI DIJELA[7]

Podatci o studentima, koji su upisani u bazu uz pomoć SQL naredbe, čitaju se i upisuju u prethodno kreiranu listu bodovi_uk kao pandas DataFrame. Za rad s podatcima odlučeno je da se studenti podijele na 20 % najboljih, 60 % prosječnih i 20 % najlošijih prema ukupnim bodovima. Iz liste dohvaćenih studenata vrši se podjela kako bi se mogle dodijeliti ocjene prema različitim kriterijima, tako da bi kreirani set podataka bio što kvalitetniji za buduću predikciju.

```
predmeti = ['Upisni broj',
    "Engleski jezik u ekonomiji I",
    "Informatičke tehnologije",
    "Matematika",
    "Osnove ekonomije",
    "Statistika",
    "Engleski jezik u ekonomiji II",
    "Makroekonomija I",
    "Matematika u ekonomiji",
    "Mikroekonomija I",
    "Računovodstvo"
    ]


ocjene = ['2', '3', '4', '5']


df2 = pd.DataFrame(np.random.choice(ocjene,size=(best_value,
11),p=[0.05, 0.15, 0.3, 0.5]), columns=predmeti)
```

```
df3 = pd.DataFrame(np.random.choice(ocjene,size=(mid_value+best_value,
11),p=[0.1, 0.2, 0.3, 0.4]), columns=predmeti)

df3 = df3.tail(-best_value)

df4 =
pd.DataFrame(np.random.choice(ocjene,size=(lower_value+mid_value+best_
value, 11),p=[0.5, 0.35, 0.1, 0.05]), columns=predmeti)

df4 = df4.tail(lower_value)
```

KOD 8. KREIRANJE LISTE PREDMETA I OCJENA I GENERIRANJE OCJENA[8]

Uz pomoć lista ocjena i predmeta te prijašnje izgeneriranih vrijednosti što se tiče broja studenata prema postignućima, generiraju se novi DataFrameovi s različitim vrijednostima p od koje svaka pomnožena sa 100 predstavlja mogućnost odabira jedne od vrijednosti iz liste ocjena iz koje se vrijednosti biraju uz pomoć funkcije random.choice ugrađene u biblioteku numpy.

```
for index, row in df2.iterrows():
    df2.at[index,'Upisni broj'] = upisni_broj[index]
for index, row in df3.iterrows():
    df3.at[index,'Upisni broj'] = upisni_broj[index]
for index, row in df4.iterrows():
    df4.at[index,'Upisni broj'] = upisni_broj[index]
frames = [df2, df3, df4]
result = pd.concat(frames)
```

KOD 9. DODAVANJE ISPRAVNIH UPISNIH BROJEVA PODATCIMA I ZAVRŠAVANJE DATAFRAME-A [9]

DataFrameovima dodajemo odgovarajuće upisne brojeve i na kraju spajamo sva tri DataFramea u jedan zajednički pandas DataFrame naredbom concat().

```
with open('EKO-ocjene1.god.csv', 'w', encoding='UTF8', newline='') as
f:
     writer = csv.writer(f)
     #rows
     writer.writerows(result)
```

```
export_csv = result.to_csv (r'C:\Users\Dominik\Desktop\Predikcija
uspjeha studenata\EKO-ocjene1.god.csv', header=True, index=False)
```

KOD 10. DODAVANJE ISPRAVNIH UPISNIH BROJEVA PODATCIMA I ZAVRŠAVANJE DATAFRAME-A [10]

Rezultat prijašnjih linija koda upisujemo uz pomoć modula CSV u datoteku te je šaljemo na određenu putanju za daljnju uporabu.

6.3. Glavni program

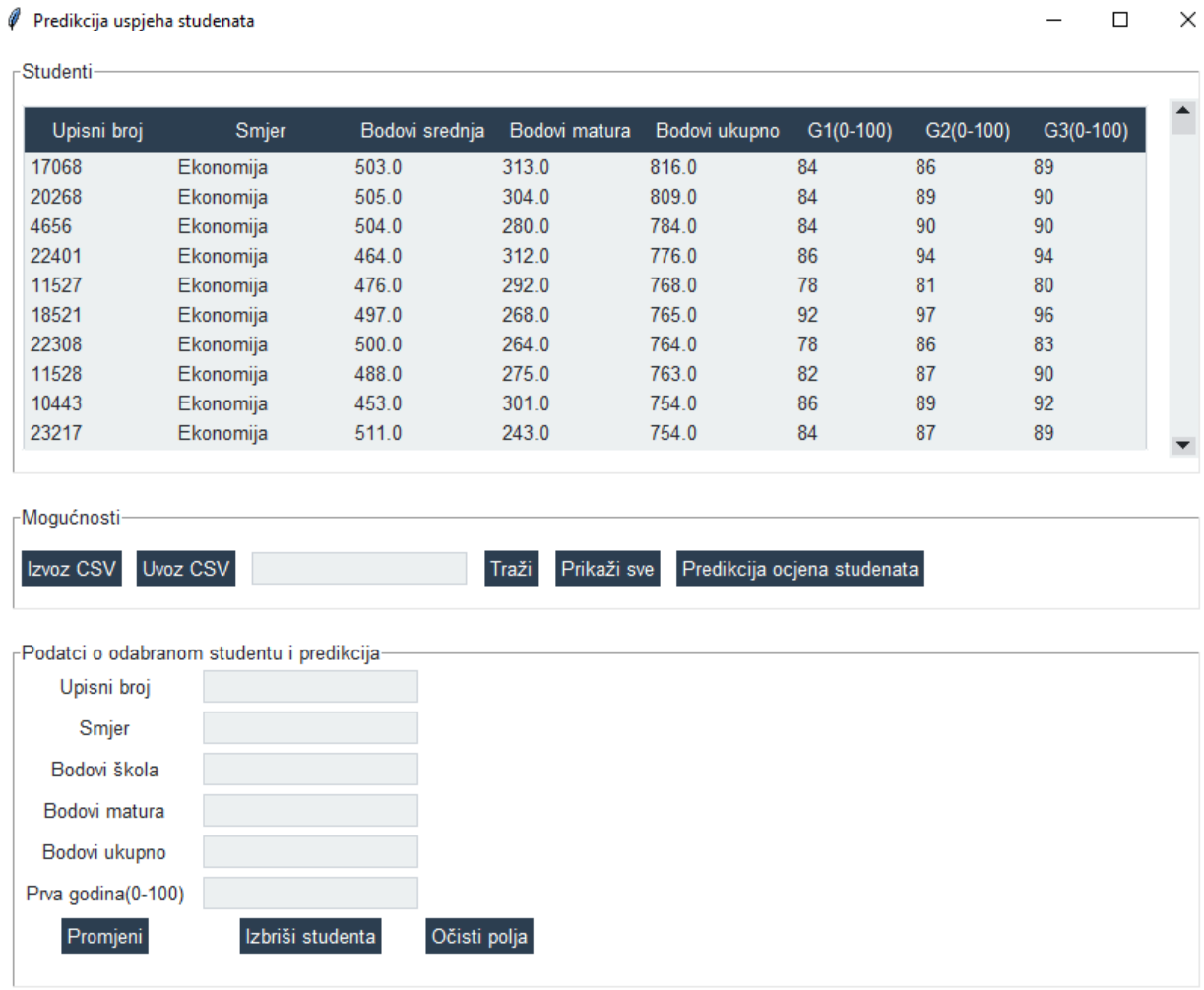
Glavni se prozor programa sastoji od tri okvira. Untar prvog okvira nalazi se tablica u kojoj se prikazuju podatci o studentima. Unutar drugog okvira postoje gumbi s funkcionalnostima uvoza podataka iz CSV datoteke, izvoza podataka unutar tablice prvog okvira u formatu CSV datoteke. Također, postoji tražilica u kojoj se može pretraživati po upisnom broju, smjeru, bodovima iz srednje škole, bodovima s državne mature ili bodovima ukupno. Pritiskom na gumb "Traži" pokreće se funkcija nakon koje se traženi podatci prikazuju unutar tablice u prvome okviru. Uz gumb traži nalazi se gumb "Prikaži sve" uz pomoć kojega se prikazuju svi studenti koji su već upisani u bazi podataka. Na kraju, kada želimo izvršiti predikciju ocjena odabranih studenata, pritiskom na gumb "Predikcija ocjena studenata" vrši se predikcija ocjena studenata u narednim godinama na fakultetu. U trećem se okviru nalaze gumbi funkcionalnosti i polja za upis podataka o studentima ako želimo mijenjati podatke o studentima ili brisati studente iz baze podataka. Podatci se učitavaju unutar okvira za upis tako što unutar prvog okvira stisnemo na jedan od redaka tablice dva puta lijevim klikom miša. Dodatna je opcija unutar trećeg okvira gumb "Očisti polja" uz pomoć kojega se brišu podatci unutar polja za upis podataka o studentima ukoliko je nešto upisano.

Prvo što je trebalo napraviti je napuniti našu bazu podataka podatcima koji su generirani u datotekama CSV formata, klikom na gumb "Uvoz CSV" i odabirom datoteka. Zatim su se odabrane datoteke upisale u tablicu i pritiskom na gumb "Spremi u bazu" podatci su bili spremljeni u bazu. Gumb "Spremi u bazu" koristio se samo za punjenje baze setom podataka. Nakon što je to napravljeno, moguće je generirati podatke o ocjenama naših studenata jer je skripta napravljena tako da uz pomoć podataka iz baze radi predikciju ocjena studenata. Programima i različitim funkcijama manipuliralo se listom podataka unutar varijable "mydata" pa se koristila kao globalna varijabla. Također, za potrebe rada sa SQL naredbama bilo je

potrebno inicijalizirati pokazivač (engl. *cursor*) koji izvršava takve naredbe. Za spajanje na bazu unutar glavnog programa, upisan je sljedeći kod:

```
mydb = mysql.connector.connect(host='localhost',
                               user='root',
                               passwd='',
                               database='predikcija_uspjeha',
                               auth_plugin='mysql_native_password')
cursor = mydb.cursor()
mydata= []
```

KOD 11. SPAJANJE NA BAZU [11]



SLIKA 11. PRIKAZ GLAVNOG PROZORA PROGRAMA [11]

Maksimalan broj bodova od srednje škole i broj bodova s državne mature iznosi 1000. Fakulteti većinom imaju još neke dodatne parametre, kao što su izborni predmeti na maturi koji donose dodatne bodove ili prijemni ispiti. U ovome radu nisu uzeti u obzir ti dodatni parametri, već samo bodovi s mature i bodovi iz srednje škole. Sve ocjene studenata, upisanih u bazu podataka, pohranjene su u CSV datoteci pod nazivom "upisni+ocjene.csv" koja sadrži sve ocjene studenata i njihove upisne brojeve. Kada smo pripremili sve podatke, u drugome se dijelu koda prvo radi predikcija linearnom regresijom za ocjene koje je svaki pojedini student ostvario u drugoj godini studija. Set podataka dijeli se na podatke za treniranje i podatke za testiranje u omjeru 80 : 20.

```
#PRIPREMA PODATAKA
query = "SELECT id, upisni_broj, smjer,
bodovi_srednja,bodovi_matura,bodovi_suma, G1,G2,G3 FROM student LIMIT
741"
cursor.execute(query)
rows = cursor.fetchall()

values = ("id",
          "Upisni broj",
          "Smjer",
          "Bodovi srednja",
          "Bodovi matura",
          "Bodovi ukupno",
          "G1",
          "G2",
          "G3"
          )
df1= pd.DataFrame(rows, columns=values)
df2 = pd.read_csv("upisni+ocjene.csv")
df2 = df2.loc[:, df2.columns != 'Upisni broj']
zbroj = pd.DataFrame((df2.sum(axis=1)/50)*100, columns = ["Finalni
bodovi"])
lista = zbroj.to_numpy().tolist()
```

```
#DRUGA GODINA

data = df1[['Bodovi srednja','Bodovi matura', 'Bodovi ukupno', 'G1',
'G2', 'G3']]
predict = "G2"

data = data.astype('int64')
x = np.array(data.drop([predict], 1))
y = np.array(data[predict])
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
0.2)
regr = LinearRegression()
regr.fit(x_train, y_train)
```

KOD 12. PRIPREMA PODATAKA ZA PREDIKCIJU LINEARNOM REGRESIJOM[12]

Bodovi se pretvaraju u decimalne brojeve od nula do jedan dijeljenjem, kako bi se dobio postotak kojim će se računati finalna ocjena. Finalna se ocjena u ovome radu računa bodovima od 0 do 100 tako što dobiveni decimalni broj pomnožimo s brojem sto. Parametri su postavljeni tako da je svakom broju bodova pridodana ocjena od 1 do 5. Ocjena nedovoljan znači da postoji šansa da student padne godinu.

```
z = 0

length = len(ocjenePred1)

while z < length:
      if ocjenePred1[z] >= 88:
            ocjenePred1[z] = 5
      elif ocjenePred1[z] < 88 and ocjenePred1[z] >= 81:
            ocjenePred1[z] = 4
      elif ocjenePred1[z] < 81 and ocjenePred1[z] >= 67:
            ocjenePred1[z] = 3
      elif ocjenePred1[z] < 67 and ocjenePred1[z] >= 51:
            ocjenePred1[z] = 2
      else:
            ocjenePred1[z] = 1
      z = z+1
```

KOD 13. DODIJELJIVANJE OCJENA STUDENTIMA PREMA OSTVARENIM BODOVIMA ZA PRVU GODINU STUDIJA [13]

Sljedeće što treba napraviti je predvidjeti ocjene za drugu godinu studija prema prethodnim podatcima. Ako se radi predikcija za podatke studenata koji nisu u bazi podataka, dodjeljuje im se broj bodova za drugu godinu prema prijašnjim podatcima koji su u glavni prozor uvezeni u obliku CSV datoteke.

```
upBr = []
smjerovi = []
bodSko = []
bodMat = []
bodUk = []
ocjenePred1 = []
ocjenePred2 = []

br = 0
while br < len(mydata):
 upBr.append(mydata[br][0])
 smjerovi.append(mydata[br][1])
 bodSko.append(mydata[br][2])
 bodMat.append(mydata[br][3])
 bodUk.append(mydata[br][4])
 ocjenePred1.append(float(mydata[br][5]))

 if len(mydata[br])>=7:

      ocjenePred2.append(regr.predict([[mydata[br][2],mydata[br][3],myd
      ata[br][4],mydata[br][5],mydata[br][7]]]))

 else:

      predict = "G2"
      st = pd.DataFrame(mydata, columns=["Upisni broj", "Smjer",
      "Bodovi srednja", "Bodovi matura", "Bodovi ukupno", "G1"])
      st = st.drop('Upisni broj', axis=1)
      st = st.drop('Smjer', axis=1)
      st['Bodovi srednja'] = st['Bodovi srednja'].astype(float)
      st['Bodovi matura'] = st['Bodovi matura'].astype(float)
      st['Bodovi ukupno'] = st['Bodovi ukupno'].astype(float)
```

```
st['G1'] = st['G1'].astype(float)
result = pd.concat([data,st])

                result = result.drop("G3",axis=1)
                regr1 = LinearRegression()
                testdf = result[result['G2'].isnull()==True]
                traindf = result[result['G2'].isnull()==False]
                y = traindf['G2']
                traindf = traindf.drop("G2",axis=1)
                regr1.fit(traindf,y)
                testdf = testdf.drop("G2",axis=1)
                pred = regr1.predict(testdf)
                testdf['G2']= pred
                ocjenePred2.append([pred[br]])

br = br+1
```

KOD 14. PREDIKCIJA OCJENA U DRUGOJ GODINI STUDIJA [14]

Također, potrebno je predvidjeti ocjene za treću godinu studija prema prethodnim podatcima. Ako se radi predikcija za podatke studenata koji nisu u bazi podataka, dodjeljuje im se broj bodova za treću godinu prema prijašnjim podatcima koji su uvezeni u obliku CSV datoteke u glavni prozor i predviđenim brojem za drugu godinu studija na sličan način kao kod predikcije za drugu godinu.

```
#TRECA GODINA
        data = df1[['Bodovi srednja','Bodovi matura', 'Bodovi ukupno',
        'G1', 'G2', 'G3']]
        predict = "G3"
        data = data.astype('float64')
        x = np.array(data.drop([predict], 1))
        y = np.array(data[predict])

        x_train, x_test, y_train, y_test = train_test_split(x, y,
        test_size = 0.2)
        regr = LinearRegression()
        regr.fit(x_train, y_train)

        prec = round((regr.score(x_test, y_test)*100),2)

        ocjenePred3 = []
```

```
        if(len(mydata[0]) >=7):
            br = 0
            while br < len(mydata):

                ocjenePred3.append(regr.predict([[mydata[br][2],mydata
                [br][3],mydata[br][4],mydata[br][5],mydata[br][6]]]))
                br = br+1
        else:
            br = 0
            while br < len(mydata):

                ocjenePred3.append(regr.predict([[float(mydata[br][2])
                ,float(mydata[br][3]),float(mydata[br][4]),float(mydat
                a[br][5]),float(ocjenePred2[br][0])]]))
                br = br+1
```

KOD 15. PREDIKCIJA OCJENA U TREĆOJ GODINI STUDIJA [15]

Kako bi se prikazali što realniji podatci, napravljen je dio koda koji slučajnim odabirom operatora dodaje ili oduzima broj od 1 do 8 (druga i treća godina). Koji će broj od 1 do 8 biti izabran za zbrajanje, odnosno oduzimanje ovisi o koeficijentima unutar koda koji predviđaju vjerojatnost odabira određenog broja.

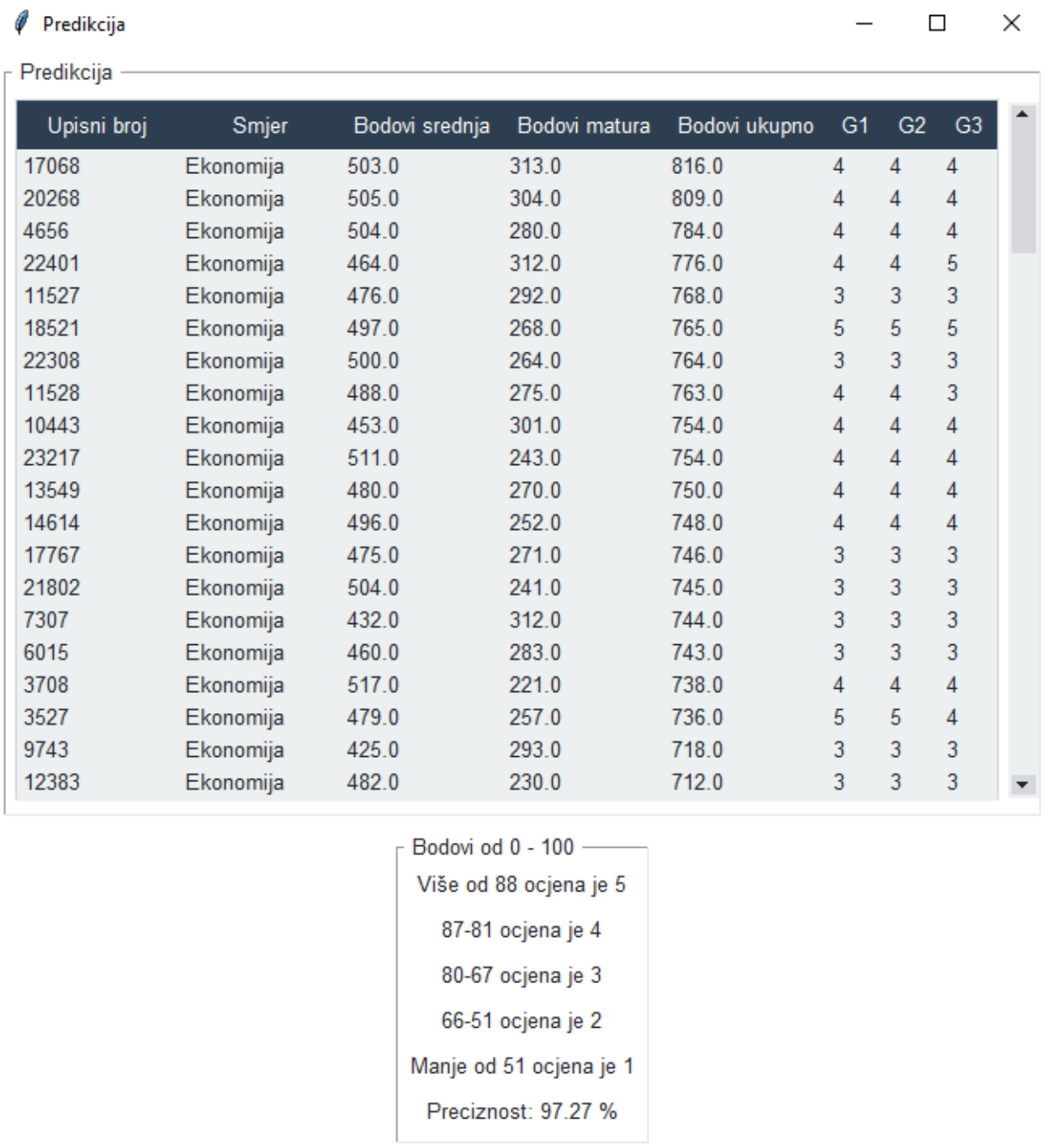
```
#Dodavanje ili oduzimanje vrijednostima
dodatak = [1, 2, 3, 4, 5, 6, 7, 8]

operators = [operator.add, operator.add, operator.sub]
random_operator = random.choice(operators)

df = pd.DataFrame(np.random.choice(dodatak,size=(len(ocjenePred2),
2),p=[0.05, 0.1, 0.15, 0.2, 0.2, 0.15, 0.1, 0.05]),
columns=['G2','G3'])

ocjenePred2 = np.concatenate(ocjenePred2, axis=0)
ocjenePred2 = ocjenePred2.tolist()
ocjenePred3 = np.concatenate(ocjenePred3, axis=0)
ocjenePred3 = ocjenePred3.tolist()

operators = [operator.add, operator.add, operator.sub]
random_operator = random.choice(operators)
ocjenePred2 = pd.DataFrame(random_operator(ocjenePred2, df["G2"]))
ocjenePred3 = pd.DataFrame(random_operator(ocjenePred3, df["G3"]))
ocjenePred2 = ocjenePred2.to_numpy()
ocjenePred3 = ocjenePred3.to_numpy()
ocjenePred2 = [float(i) for i in ocjenePred2]
ocjenePred3 = [float(i) for i in ocjenePred3]
```

KOD 16. DODATAK OCJENAMA [16]

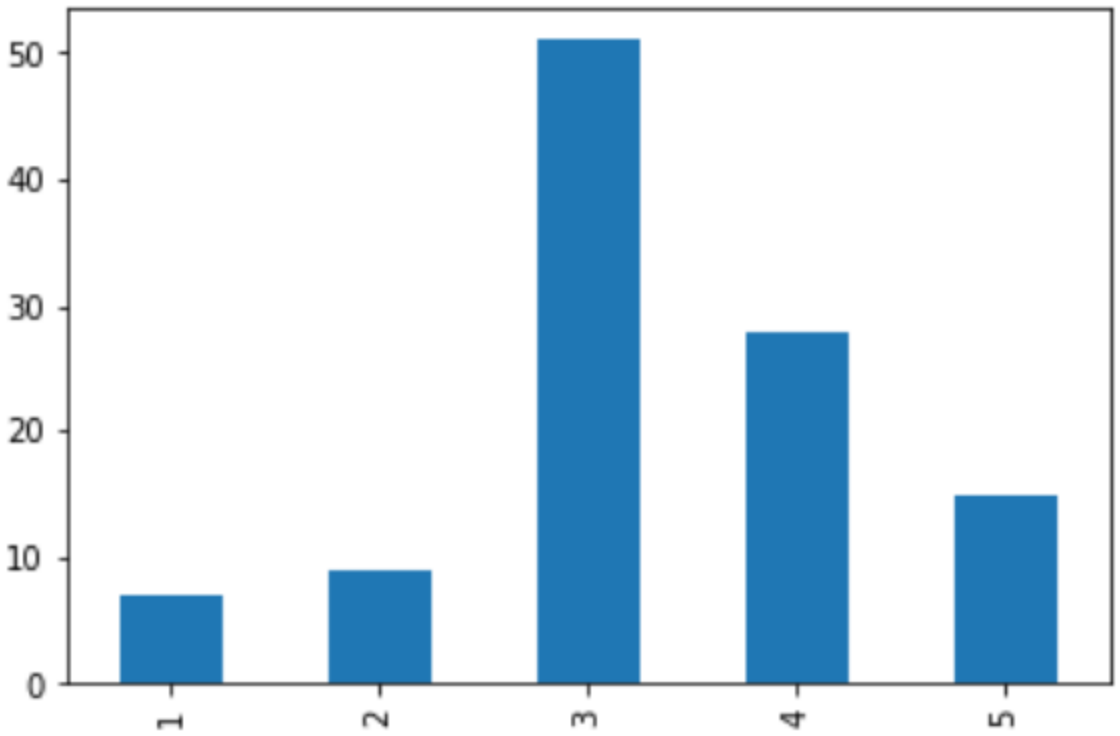
Najvažniji je gumb aplikacije “Predikcija ocjena studenata”. Pritiskom na taj gumb svi studenti, koji su bili prikazani unutar tablice glavnog prozora, bit će upisani u novi prozor gdje će biti napravljena i sama predikcija njihovih ocjena. Pritiskom na ovaj gumb, pokreće se skripta koja stvara novi prozor, radi predikcija za sve godine i daje stupičasti grafikon koji prikazuje koliko je bilo kojih ocjena. Pokazuju se samo studenti koji su bili filtrirani tražilicom, uz svakog od njih nalaze se podatci o njihovim bodovima, upisni broj i smjer kako bi se mogli raspoznati. Također, unutar ovog prozora na dnu je prikazana tablica s bodovima prema kojoj se može očitati koliko je bodova potrebno za pojedinu ocjenu. Ispod prikaza tablice s bodovima prikazuje se preciznost trenutačne predikcije sa zadanim setom podataka u bazi.

```
#ISPIS
ispis = pd.DataFrame(columns = ['Upisni broj','Smjer', 'Bodovi
srednja', 'Bodovi matura', 'Bodovi ukupno','G1','G2','G3'])
ispis['Upisni broj'] = upBr
ispis['Smjer'] = smjerovi
ispis['Bodovi srednja'] = bodSko
ispis['Bodovi matura'] = bodMat
ispis['Bodovi ukupno'] = bodUk
ispis['G1'] = ocjenePred1
ispis['G2'] = ocjenePred2
ispis['G3'] = ocjenePred3
ispis.loc[ispis.G2 > 100, 'G2'] = 100
ispis.loc[ispis.G3 > 100, 'G3'] = 100
predikcijaPrikaz = ispis.to_numpy().tolist()

for row in predikcijaPrikaz:
  trv.insert("", "end", values=row)

#prikaz grafa ocjena prve godine svih studenata u tablici
ocjenePlot = ispis['G1'].value_counts().sort_index()
ocjenePlot.plot.bar()
plt.show()
```

KOD 17. KREIRANJE ISPISA I GRAFA S OCJENAMA PRVE GODINE [17]



SLIKA 12. PREDVIĐENE OCJENE STUDENATA EKONOMIJE [12]



SLIKA 13. PREDVIĐENE OCJENE STUDENATA EKONOMIJE - GRAF [13]

6.4. Testiranje aplikacije

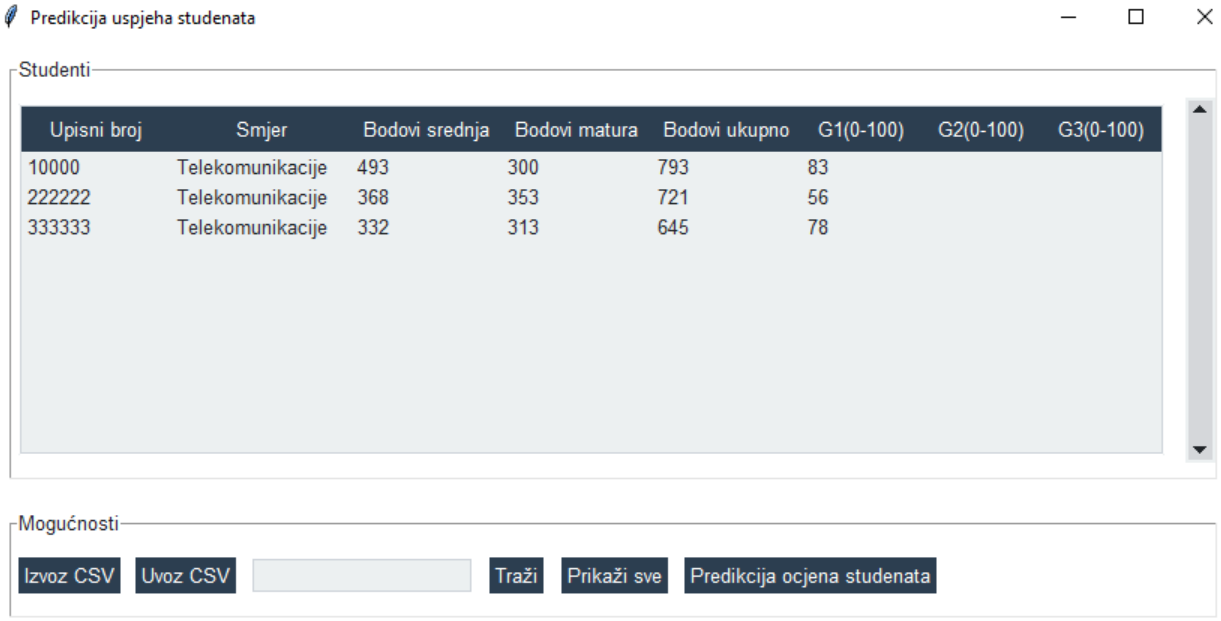
Učitat ćemo podatke o studentima telekomunikacija u tablicu i napraviti predikciju njihovih ocjena u sljedećim godinama. Prvo što trebamo imati je set podataka. Naš set podataka izgleda ovako:

	A	B	C	D	E	F
1	Upisni broj	Smjer	Bodovi srednja	Bodovi matura	Bodovi ukupno	G1
2	10000	Telekomunikacije	493	300	793	83
3	222222	Telekomunikacije	368	353	721	56
4	333333	Telekomunikacije	332	313	645	78

SLIKA 14. FRAGMENT KREIRANE CSV DATOTEKE TEL.CSV [14]

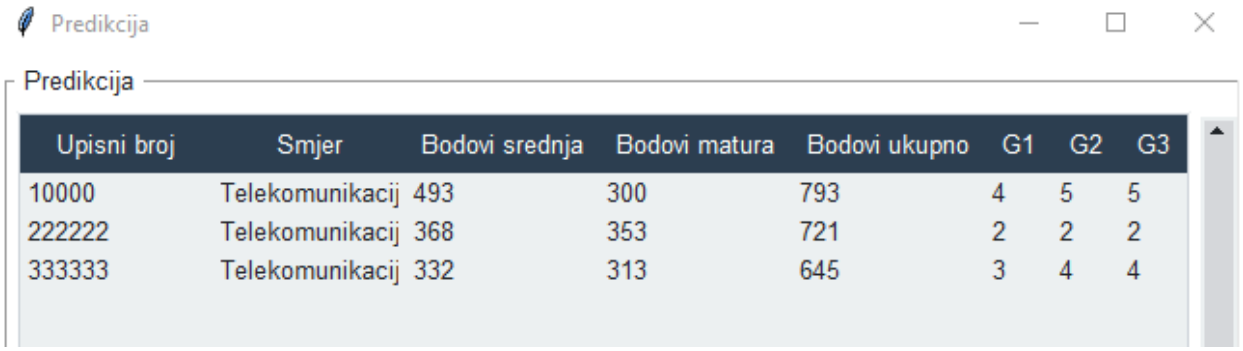
Podatci o studentima mogu se uvesti jednim klikom na gumb “Uvoz CSV” unutar glavnog prozora programa. Nakon odabira datoteke, podatci su učitani u tablicu unutar prvoga okvira sa svim podatcima ispravno postavljenim u stupce. Podatci o studentima koji nedostaju su ocjene pretvorene u broj bodova od 0 do 100 na drugoj i trećoj godini studija. Te podatke nemamo tako da će se predikcija vršiti prema podatcima o njihovim postignućima iz srednje

škole, bodovima s državne mature i ocjenama s prve godine koji se izračunaju tako što se zbroje sve ocjene i podijele s maksimalnim mogućim brojem (ako je 10 predmeta, maksimalni broj je 50). Dobiveni decimalni broj pretvori se u broj s dvije decimale i pomnoži sa 100 te se tako dobije broj koji treba unijeti pod prvu godinu.



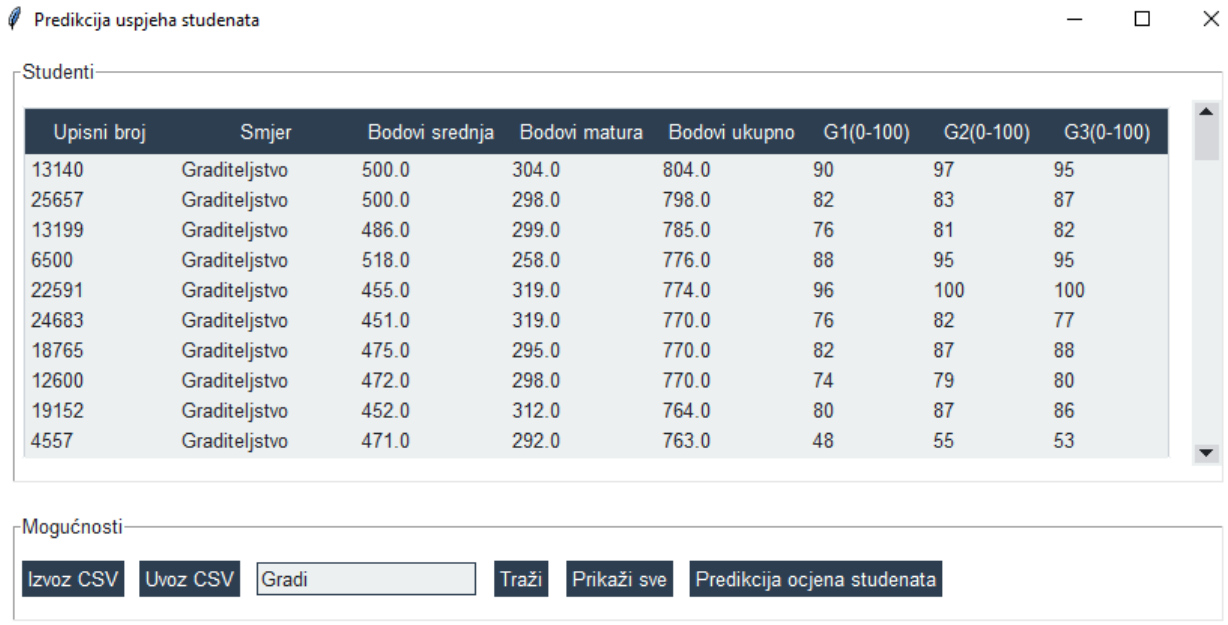
SLIKA 15. UČITANI PODATCI IZ DATOTEKE [15]

Daljnji je korak predviđanje uspjeha studenata u sljedećim godinama studija uz pomoć klika na gumb "Predikcija ocjena studenata". Predikcija se vrši uz pomoć seta podataka koji je unesen u bazu.



SLIKA 16. PREDVIĐENE OCJENE STUDENATA NA DRUGOJ I TREĆOJ GODINI [16]

Kako bi se testiralo pretraživanje, uz pomoć tražilice i pritiskom na gumb “Traži” može se filtrirati prikaz unutar glavnog prozora prema upisnom broju, smjeru, bodovima iz srednje škole, bodovima s državne mature ili bodovima ukupno. Za potrebe testiranja tražilice, pretraživani su samo studenti graditeljstva po smjeru.



SLIKA 17. TRAŽILICA/FILTRIRANJE [17]

7. REZULTATI

Tijekom testiranja aplikacije, srednja vrijednost preciznosti na deset testiranja bila je oko 93 % točna. Najniža je vrijednost iznosila 78 %, a najviša 97 %. Točnost aplikacije ovisi o raspodjeli podataka na testne i podatke za treniranje pa se, ovisno o tome, dobiju različiti podatci.

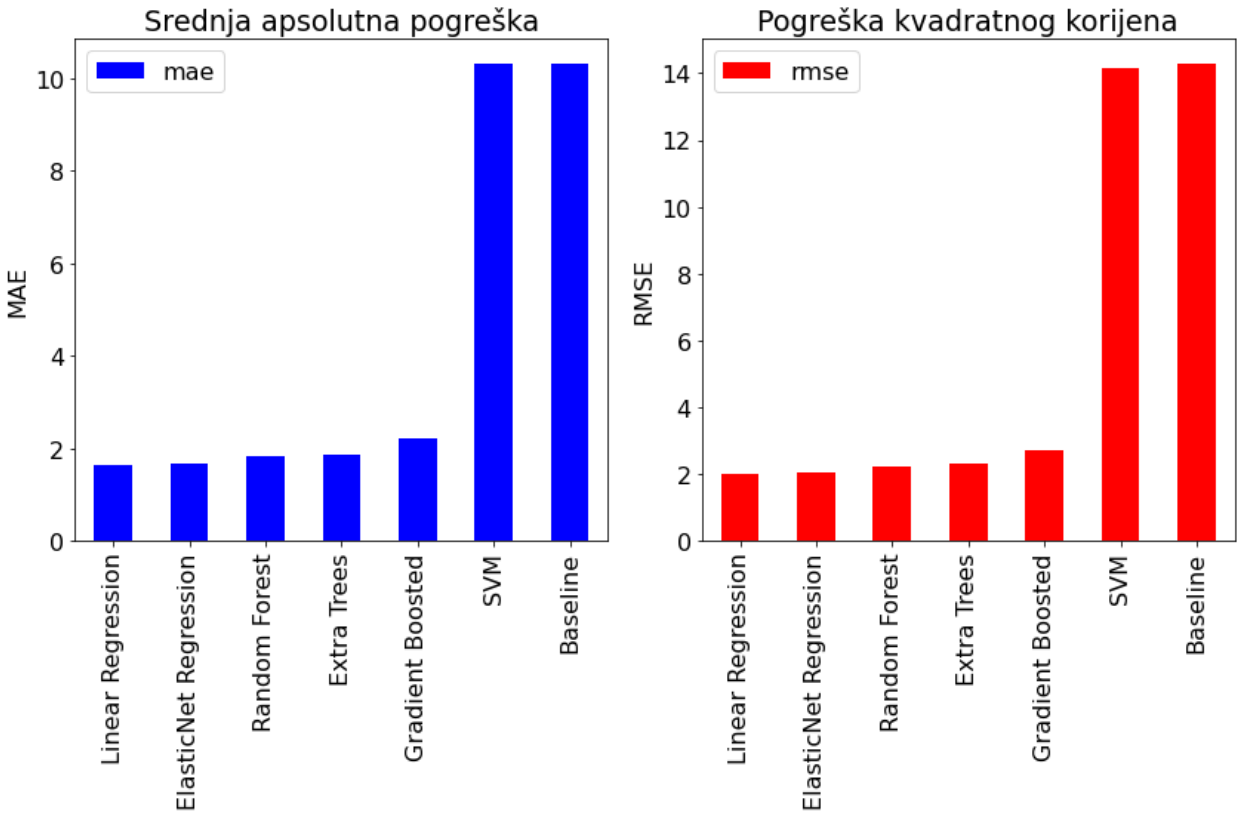
Kako bi se uopće prikazala pogodnost ove teme za strojno učenje, napravljena je analiza putem većine alata za predikciju opisanih u ovome završnom radu u potpoglavlju "Alati za predikciju". Analiza putem modela napravljena je, također, u okruženju Spyder u programskom jeziku Python u kojem je vrlo jednostavno napraviti analizu kada se uključe već gotove biblioteke za svaki pojedini alat. Alati, koji su uzeti u obzir, su: linearna regresija, elastična mreža, slučajna šuma, ekstremno nasumična stabla, pojačavanje gradijenta i SVM.

Osnovna linija predstavlja predviđene srednje vrijednosti (medijane) za vrijednosti iz seta podataka koji služe za treniranje. Ako model strojnog učenja ne može poboljšati ovu jednostavnu osnovu, onda treba isprobati drugačiji pristup. Srednja apsolutna greška (engl. *Mean Absolute Error (MAE)*) mjeri prosječnu veličinu pogrešaka u skupu predviđanja. To je prosjek na testnom uzorku apsolutnih razlika između predviđanja i stvarnog promatranja gdje sve individualne razlike imaju jednaku težinu. Korijenska srednja kvadratna pogreška (engl. *Root mean squared error (RMSE)*) pravilo je kvadratnog bodovanja koje, također, mjeri prosječnu veličinu pogreške. To je kvadratni korijen prosjeka kvadrata razlika između predviđanja i stvarnog promatranja. Budući da su pogreške kvadrirane prije nego što je napravljen prosjek, RMSE daje veću težinu velikim pogreškama. To znači da bi RMSE trebao biti korisniji kada su velike pogreške posebno nepoželjne.

```
                            mae     rmse
Linear Regression       1.64222  2.03267
ElasticNet Regression   1.66861  2.07433
Random Forest           1.82671   2.2302
Extra Trees             1.85678  2.30242
SVM                       10.32  14.1707
Gradient Boosted        2.21288  2.71092
Baseline                10.3356  14.3054
The Linear Regression regressor is 84.11% better than the baseline.
```

SLIKA 19. VRIJEDNOSTI DOBIVENE ANALIZOM ALATA [19]

Analizom su se svi alati pokazali dobrima za predikciju, izuzev SVM alata koji je bio jako sličan pogrješkama kao napravljena osnovna linija. Svi su ostali modeli dobili puno bolje rezultate. Dobivena vrijednost za linearnu regresiju, koja je korištena u ovom završnom radu, bila je 84,11 % bolja od osnovne linije, što govori da je ovaj model pogodan za strojno učenje i izradu prediktivnog modela.



SLIKA 18. GRAFOVI ANALIZE ALATA ZA PREDIKCIJU [18]

8. ZAKLJUČCI

Ovim završnim radom prikazana je upotreba programskog jezika Python za razvoj GUI aplikacije za predikciju ocjena studenata na fakultetima. Korišteno je razvojno okruženje Spyder unutar alata Anaconda. Uz raznovrsne biblioteke i module prikazan je rad s podatcima u Pythonu, kao i razvoj grafičkog sučelja u programskom jeziku Python.

Napravljena predikcija može se poboljšati dodavanjem mnogih parametara. Neki od njih su završen stupanj školovanja majke i oca, slobodno vrijeme, financijsko stanje obitelji, bodovi ostvareni na svakom kolokviju ili ispitu pojedinačno, prijemni ispiti ili određeni predmeti na maturi, udaljenost fakulteta od mjesta stanovanja, pristup internetu te utjecaj svakog pojedinog predmeta s prve godine na ocjene iz predmeta na višim godinama studija.

9. POPIS LITERATURE

1. Prediktivna analitika
   https://www.valicon.net/bs/sva-rjesenja/marketing-analitika-automatizacija/solutions/analitika/prediktivna-analitika/
2. Anaconda Distribution Website
   https://www.anaconda.com/products/individual
3. Spyder okruženje
   https://www.spyder-ide.org/
4. Strojno učenje
   https://www.fer.unizg.hr/predmet/su
5. Scikit-learn
   https://scikit-learn.org/stable/
6. Numpy
   https://numpy.org/doc/stable/user/whatisnumpy.html
7. Pandas
   https://pandas.pydata.org/docs/getting_started/overview.html
8. Tkinter
   https://docs.python.org/3/library/tkinter.html
9. Stabla odlučivanja
   https://scikit-learn.org/stable/modules/tree.html
10. Strojno učenje
    https://towardsdatascience.com/classification-regression-and-prediction-whats-the-difference-5423d9efe4ec
    https://towardsdatascience.com/10-most-popular-machine-learning-software-tools-in-2019-678b80643ceb
    https://www.northeastern.edu/graduate/blog/artificial-intelligence-vs-machine-learning-whats-the-difference/
11. Random
    https://docs.python.org/3/library/random.html
12. CSV
    https://docs.python.org/3/library/csv.html

13. Python
https://hub.docker.com/_/python
14. MySQL connector
https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html
15. Python i R
https://www.newgenapps.com/hr/blogs/r-vs-python-for-data-science-big-data-artificial-intelligence-ml/
16. Znanost o podatcima
https://www.fer.unizg.hr/_download/repository/b5_Znanost_o_podatcima.pdf
17. Srednja apsolutna greška u odnosu na korijensku srednju kvadratnu pogrešku
https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d
18. Extra trees vs Random Forest
https://quantdare.com/what-is-the-difference-between-extra-trees-and-random-forest/
19. SVM
https://scikit-learn.org/stable/modules/svm.html
20. ElasticNet
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html
21. Gradient Boosted
https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/