

Razvoj modela strojnog učenja za predviđanje akademskog uspjeha studenta

Vučić, Petra

Master's thesis / Diplomski rad (sveučilišni)

2025

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:166:402817>

Rights / Prava: [Zaštićeno autorskim pravom.](#) / [In copyright.](#)

Download date / Datum preuzimanja: **2026-09-07**



Repository / Repozitorij:

[Digital Repository of Faculty of Science in Split / Digitalni repozitorij Prirodoslovno-matematičkog fakulteta u Splitu](#)



Dabar

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**RAZVOJ MODELA STROJNOG UČENJA
ZA PREDVIĐANJE AKADEMSKOG
USPJEHA STUDENTA**

Petra Vučić

Split, srpanj 2025.

Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu

Prirodoslovno-matematički fakultet

Odjel za informatiku

Ruđera Boškovića 33, 21000 Split, Hrvatska

RAZVOJ MODELA STROJNOG UČENJA ZA PREDVIĐANJE AKADEMSKOG USPJEHA STUDENTA

Petra Vučić

SAŽETAK

Rad razvija modele strojnog učenja u svrhu predviđanja akademskog uspjeha studenata. Istraživanje se temelji na anonimiziranim podacima iz ISVU-a sustava. Cilj ovog istraživanja je izgradnja modela strojnog učenja koji mogu predvidjeti hoće li student diplomirati ili odustati od studija, kao i modela za predviđanje prosjeka ocjena na temelju broja godina studiranja. Korišteni su algoritmi poput logističke i linearne regresije, stabla odluke i slučajne šume. Modeli su evaluirani pomoću metrika AUC, MAE, MSE i R^2 . Analizom dobivenih rezultata prikazano je da se primjenom odabranih značajki može postići pouzdan uvid u budući akademski uspjeh studenata.

Ključne riječi: strojno učenje, linearna regresija, logistička regresija, stablo odluke, slučajna šuma, razvoj modela, predviđanje uspjeha.

Rad sadrži: 64 stranica, 44 grafičkih prikaza i 28 literaturnih navoda

Mentor: **prof. dr. sc. Ani Grubišić**, redoviti profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Ocjenjivači: **prof. dr. sc. Ani Grubišić**, redoviti profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

prof. dr. sc. Branko Žitko, redoviti profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Ines Šarić-Grgić, dipl. ing., asistent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad prihvaćen: srpanj 2025.

Basic documentation card

Master's thesis

University of Split

Faculty of Science

Department of informatics

Ruđera Boškovića 33, 21000 Split, Croatia

DEVELOPING MACHINE LEARNING MODELS FOR PREDICTING ACADEMIC PERFORMANCE

Petra Vučić

ABSTRACT

The paper develops machine learning models for the purpose of predicting students' academic success. The research is based on anonymized data from the ISVU system. The goal of this research is to build machine learning models that can predict whether a student will graduate or drop out, as well as models for predicting grade point averages based on the number of years of study. Algorithms such as logistic and linear regression, decision trees, and random forests were used. The models were evaluated using the metrics AUC, MAE, MSE, and R^2 . The analysis showed that selected features can provide insight into students' future academic success.

Keywords: machine learning, linear regression, logistic regression, decision tree, random forest, model development, performance prediction.

Thesis consists of: 64 pages, 44 figures and 28 references

Original language: Croatian

Mentor: **Ani Grubišić, Ph.D.** *Full Professor of Faculty of Science, University of Split*

Reviewers: **Ani Grubišić, Ph.D.** *Full Professor of Faculty of Science, University of Split*

Branko Žitko, Ph.D., *Full Professor, Faculty of Science, University of Split*

Ines Šarić-Grgić, MSc., *Assistant, Faculty of Science, University of Split*

Thesis accepted: July 2025.

IZJAVA

o samostalnoj izradi diplomskog rada

Izjavljujem pod punom materijalnom i moralnom odgovornošću da sam ovaj rad izradila samostalno te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu propisano označeni kao citati s navedenim izvorom iz kojeg su preneseni.

U Splitu, 08.07.2025.

Petra Vučić

(student-ica)

Petra Vučić

08.07.2025. 18:38:28

1e30511c88e7f3d4a091f4792b92b8dd41c8a23b

<https://sceduly.com/hr/validate-signature>

ZAHVALA

Ovim putem želim se zahvaliti svojoj mentorici prof. dr. sc. Ani Grubišić na prenesenom znanju i savjetima tijekom školovanja i tijekom izrade diplomskog rada.

Hvala mojim prijateljima i kolegama koji su uljepšali moje studiranje, popravili svaki loš dan smijehom i bili moja potpora i motivacija.

Hvala svim profesorima na prenesenom znanju i životnim savjetima.

Najveće hvala mojim roditeljima, sestri, Petru i Josipu na potpori, savjetima, strpljenju, povjerenju i razumijevanju proteklih pet godina.

Sadržaj

Uvod.....	1
1. Teorijski okvir.....	2
1.1. Strojno učenje - koncept, primjena i metode	3
1.1.1. Nadzirano učenje (engl. <i>Supervised Learning</i>).....	5
1.1.2. Nenadzirano učenje (engl. <i>Unsupervised Learning</i>).....	5
1.1.3. Učenje potkrjepljivanjem (engl. <i>Reinforcement Learning</i>).....	6
1.2. Prediktivna analitika u obrazovanju i primjena strojnog učenja u predviđanju akademskog uspjeha	6
1.2.1. Tehnike prediktivne analitike	8
1.3. Predviđanje završetka studija: logistička regresija kao najučinkovitiji pristup	8
1.4. Pregled literature.....	9
2. Metodologija	11
2.1. Pregled podataka i odabir skupa	12
2.1.1. Ciljana skupina - opći podatci	14
2.1.2. Završetak studija - prosjek, duljina studiranja.....	14
2.2. Instalacija paketa	15
2.3. Distribucija završetka fakulteta	19
2.3.1. Kreiranje kružnog dijagrama (engl. <i>pie charta</i>).....	20
2.4. Analiza povezanosti prosjeka ocjena i duljine studiranja	22
2.4.1. Kreiranje raspršenog dijagrama (engl. <i>scatter plot</i>).....	23
2.5. Predviđanje završetka studija.....	25
2.5.1. Priprema podataka.....	26
2.5.2. Treniranje modela	30
2.5.3. Evaluacija modela	32
2.6. Predviđanje prosjeka ocjena studenta.....	35
2.6.2. Treniranje modela	35

2.6.3.	Evaluacija modela	40
2.6.4.	Linearna regresija	40
2.6.5.	Slučajna šuma	41
2.6.6.	Stablo odluke	42
3.	Rezultati i diskusija.....	44
3.1.	Rezultati predviđanja završetka studija	45
3.1.1.	Model logističke regresije.....	45
3.2.	Rezultati predviđanja prosjeka ocjena studenata	51
3.2.1.	Model linearne regresije	51
3.2.2.	Model slučajne šume.....	54
3.2.3.	Model stabla odluke	56
	Zaključak	59
	Literatura.....	60
	Sažetak	62
	Summary	63
	Skraćenice	64

Uvod

Praćenje i analiza obrazovnog puta studenata važan su segment unapređenja kvalitete visokog obrazovanja. Sveučilišta i visoka učilišta sve se češće susreću sa izazovima poput visokih stopa odustajanja, dužeg trajanja studija i varijabilne akademske uspješnosti među studentima. U tom kontekstu ključno je razviti alate i pristupe koji će omogućiti pravovremenu identifikaciju studenata koji su rizični po pitanju nepovoljnih akademskih ishoda, ali i bolje razumijevanje obrazaca ponašanja onih koji uspješno završavaju studij.

Sve veći broj dostupnih podataka o obrazovanju omogućuju analitičke i prediktivne pristupe zasnovane na metodama strojnog učenja. Ovaj rad zasniva se upravo na primjeni takvih metoda za dvije međusobno povezane predikcije: predviđanje završetka ili odustajanja od studija te predviđanje prosječne ocjene studenata na temelju trajanja njihova studija. Time se ne analizira samo ishod obrazovanja (diplomirao/ispisan), već se pokušava razumjeti i kako napredovanje kroz studij utječe na akademsku uspješnost koja je izražena u obliku prosječne ocjene.

U istraživanju su korišteni anonimizirani podaci iz ISVU sustava koji uključuju informacije o ocjenama, numerički podatak o godinama studiranja, rezultatima državne mature, ocjenama iz srednje škole. U cilju izgradnje prediktivnih modela korištene su različite metode strojnog učenja: logistička regresija za klasifikaciju statusa studenata, linearna regresija i slučajna šuma za predviđanje prosjeka ocjena, te stabla odluke kao dodatna interpretabilna metoda. Svaki od modela evaluiran je pomoću relevantnih metrika kao što su točnost (AUC), srednja apsolutna pogreška (MAE), srednja kvadratna pogreška (MSE) i R-kvadrat (R^2), čime se omogućuje detaljna usporedba njihove učinkovitosti.

Cilj rada je istražiti može li se na temelju ograničenog skupa akademskih podataka predvidjeti tko će diplomirati, tko će odustati, te kakve će biti ocjene aktivnih studenata s obzirom na duljinu studiranja. Osim što se pokušava unaprijediti razumijevanje ključnih obrazovnih čimbenika, rad ima i praktičnu vrijednost jer može poslužiti kao temelj za razvoj sustava ranog upozoravanja i personaliziranih intervencija u visokom obrazovanju.

1. Teorijski okvir

Digitalna svakodnevnica u kojoj živimo rezultira sve većim brojem podataka - o našim navikama, ponašanju, pa i o obrazovanju. Kako podataka ima sve više, razvijaju se i različiti alati koji nam pomažu da iz njih izvučemo smisao. Jedan od najzanimljivijih i najbrže rastućih pristupa svakako je strojno učenje, koje sve češće postaje sastavni dio modernih tehnoloških rješenja.

Strojno učenje (engl. *machine learning*) možemo opisati kao granu umjetne inteligencije koja omogućuje računalnim sustavima da „uče“ iz podataka i donose odluke bez da su za svaku situaciju eksplicitno programirani (Mitchell, 1997). Drugim riječima, umjesto da računalu točno kažemo što treba učiniti ono samo pronalazi obrasce i pravila na temelju dostupnih primjera i podataka. Postoji više načina na koje računala mogu „učiti“ iz podataka, a najčešće se ističu tri osnovna pristupa. Prvi je nadzirano učenje, kod kojeg sustav uči na temelju označenih primjera. Drugi način je nenadzirano učenje koristi se kada podaci nisu označeni, a cilj je pronaći skrivene strukture, povezanosti ili skupine unutar podataka. Treći pristup je učenje potkrepljivanjem, koje se temelji na sustavu nagrađivanja i kažnjavanja. Sustav kroz interakciju s okolinom uči koje su akcije poželjne tako da za ispravne odluke dobiva nagradu, a za pogrešne kaznu.

Pojam prediktivne analitike često se spominje uz strojno učenje jer se odnosi na korištenje povijesnih podataka kako bi se predvidjeli budući rezultati. Ova vrsta analize već se dugo koristi u poslovanju, financijama i zdravstvu, no sve više nalazi svoju primjenu i u obrazovanju. U obrazovnom kontekstu prediktivni modeli mogu pomoći nastavnicima i institucijama da pravovremeno prepoznaju studente koji su u riziku od akademskog neuspjeha i pruže im dodatnu podršku (Koppius, 2011.).

Takvi pristupi postaju dio šireg područja poznatog kao Educational Data Mining (EDM), područja koje se bavi primjenom računalnih metoda za analizu obrazovnih podataka (Inventado, 2014). EDM se temelji na ideji da podaci koje prikupljamo tijekom obrazovnog procesa kao što su ocjene, broj upisa, broj položenih ispita - mogu poslužiti kao osnova za razumijevanje i poboljšanje tog istog procesa.

Za izgradnju prediktivnih modela koriste se različite metode, a to ovisi o prirodi problema i vrsti podataka. Neki od najčešće korištenih algoritama uključuju logističku regresiju, stabla

odluke, neuronske mreže i ansambl metode poput *Slučajne šume*. Svaka od ovih metoda ima svoje prednosti i ograničenja, a njihova učinkovitost uvelike ovisi o kvaliteti i količini podataka, kao i o pažljivoj pripremi ulaznih značajki (Kotsiantis, 2004.).

Cilj ovakvih sustava nije zamijeniti ljudsku procjenu, već je nadopuniti. Ponuditi dodatne uvide koji mogu pomoći nastavnicima, stručnim službama i ostalima koji donose odluke u obrazovanju. Kroz kvalitetnu analizu podataka moguće je ranije prepoznati studente kojima je potrebna podrška, poboljšati način donošenja odluka te bolje usmjeriti resurse i pažnju.

1.1. Strojno učenje - koncept, primjena i metode

Tom Mitchell je definirao strojno učenje kao: „Kažemo da računalni program uči iz iskustva E u kontekstu neke klase zadataka T i mjere uspješnosti izvođenja P , ako se njegova izvedba u obavljanju zadataka iz T , mjereno prema P , poboljšava s prikupljenim iskustvom E “ (Mitchell, 1997, str. 2). Iz ovoga možemo zaključiti da računalni program uči ako ne ponavlja greške i na taj način poboljšava svoju izvedbu i rezultate. Primjer kojeg Mitchell opisuje u svojoj knjizi je problem učenja igranja dame, a može se opisati kroz tri osnovne komponente: zadatak (T), mjera uspješnosti (P) i iskustvo učenja (E). U ovom slučaju zadatak (T) je igranje dame, mjera uspješnosti (P) je postotak pobjeda na svjetskom turniru, dok iskustvo (E) predstavlja partije koje sustav odigra sam protiv sebe. Kako bismo u potpunosti osmislili sustav za učenje potrebno je dodatno definirati tri ključna elementa. Prvo treba jasno odrediti vrstu znanja koju sustav treba usvojiti - funkciju koja ocjenjuje pozicije na ploči. Drugo, mora se odabrati način predstavljanja tog znanja, poput funkcije evaluacije pozicije izražene kroz niz značajki. I naposljetku, potrebno je definirati mehanizam učenja, tj. algoritam koji će na temelju prikupljenog iskustva poboljšavati sustav donošenja odluka u igri (Mitchell, 1997).

Strojno učenje dio je umjetne inteligencije i odnosi se na metode koje omogućuju računalima da sama „uče“ iz podataka, bez da ih se mora unaprijed točno isprogramirati za svaki zadatak. Umjesto da im damo točne upute, dajemo im podatke iz kojih oni sami prepoznaju obrasce i donose zaključke. U mnogim slučajevima količina podataka kojima raspolazemo je tolika da ih čovjek ne može učinkovito analizirati niti prepoznati obrasce i upravo tada u pomoć dolazi strojno učenje.

U posljednjim desetljećima strojno učenje je toliko napredovalo da ga danas koristimo gotovo svaki dan da to i ne primjećujemo. Algoritmi strojnog učenja koriste se posvuda i to od personaliziranih poruka na internet trgovinama, do otključavanja mobilnih uređaja metodom „prepoznavanja lica“, YouTube ili Netflix prijedloga za gledanje koji bi nam se mogli svidjeti, dijagnostike bolesti ili pak aplikacija koje predviđaju gustoću prometa na putu do posla.

Strojno učenje danas spada u jedno od najbrže rastućih područja u znanosti i industriji te se primjenjuje u različitim disciplinama - od medicine i financija do računalne znanosti i obrazovanja. Primjena modela strojnog učenja u obrazovnim istraživanjima omogućuje analizu obrazaca u studentskim podacima i identifikaciju ključnih faktora koji utječu na akademski uspjeh. Strojno učenje podrazumijeva programiranje računala s ciljem optimizacije mjere uspješnosti koristeći primjerne podatke ili prethodno iskustvo. Imamo model definiran do određene razine parametrizacije, a učenje je proces u kojem računalni program optimizira te parametre na temelju podataka za treniranje ili prošlog iskustva (Alpaydin, 2004). Popularnost strojnog učenja proizlazi iz sposobnosti da računalni sustavi uče iz podataka, prepoznaju obrasce i donose odluke bez eksplicitnog programiranja. Za razliku od tradicionalnih metoda analize podataka koje se oslanjaju na stroge matematičke modele i teorijske jednadžbe, strojno učenje omogućuje računalnim sustavima da samostalno otkrivaju povezanosti između podataka i s vremenom poboljšavaju svoje performanse kako im raste količina podataka za učenje (Samuel, 1959., str. 211.).

Postoje tri glavne metode strojnog učenja:

1. Nadzirano učenje (engl. *Supervised Learning*)
2. Nenadzirano učenje (engl. *Unsupervised Learning*)
3. Učenje potkrjepljivanjem (engl. *Reinforcement Learning*)

1.1.1. Nadzirano učenje (engl. *Supervised Learning*)

Kod nadziranog učenja skup podataka koji se koristi za treniranje algoritma sadrži i pripadajuće točne odgovore, takozvane oznake (Geron, 2019). Kod nadziranog učenja algoritmi rade s poznatim skupom ulaznih podataka i poznatim skupom izlaznih podataka. Model se trenira na označenim podacima kako bi naučio prepoznavati obrasce i veze između ulaza i izlaza. Ulazni podaci najčešće su prikazani kao vektori značajki, a cijeli skup podataka može se prikazati matricom. Cilj je napraviti model koji može pravilno predvidjeti izlaze za nove podatke koje nikada prije nije vidio, a sve to na temelju naučenih uzoraka. To se postiže optimizacijom neke funkcije cilja koja mjeri koliko je model uspješan u predviđanju.

Primjeri primjene:

- **Klasifikacijom** model uči kako nekom podatku dodijeliti kategoriju koja mu pripada kao što je prepoznavanje neželjenih poruka u e-pošti, dijagnostika bolesti, predviđanje hoće li student diplomirati ili odustati.
- **Regresija** se koristi kad želimo predvidjeti neku brojčanu vrijednost, npr. predviđanje cijena nekretnina, potrošnje energije ili studentskog uspjeha.

1.1.2. Nenadzirano učenje (engl. *Unsupervised Learning*)

„Kod nenadziranog učenja, podaci za treniranje nemaju oznake. Sustav pokušava učiti bez učitelja“ (Geron, 2019., str. 279.). Nenadzirano učenje koristi se kada imamo samo ulazne podatke i ne znamo očekivane izlaze. Cilj je pronaći skrivene obrasce i odnose u podacima.

Najčešće tehnike uključuju:

- **Grupiranje (engl. *Clustering*)** je pronalaženje skupina u podacima, npr. segmentacija kupaca u marketingu ili u obrazovanju grupiranje se može koristiti za identificiranje skupina učenika ili studenata koji imaju slične obrasce učenja.
- **Asocijativno učenje** je otkrivanje pravila među podacima, npr. „Kupci koji su kupili proizvod A često kupuju i proizvod B.“. U obrazovanju asocijativno učenje pomaže kod otkrivanja povezanosti između različitih obrazovnih resursa i uspjeha učenika, npr. „Studenti koji uče iz materijala A, često polažu ispit B“.

1.1.3. Učenje potkrjepljivanjem (engl. *Reinforcement Learning*)

Učenje potkrjepljivanjem je proces učenja što učiniti tj. kako povezati situacije s akcijama kako bi se maksimizirao brojčani signal nagrade (Geron, 2019., str. 575.). Učenje potkrjepljivanjem temelji se na principu nagrade i kazne gdje model kroz interakciju s okolinom uči optimalne akcije.

Primjeri primjene:

- Autonomna vozila koji uče kako sigurno voziti.
- Digitalna platforma za učenje matematike koristi RL algoritam koji prati koliko brzo učenik rješava zadatke, koliko često griješi, kod kojih tipove zadataka ima poteškoće pri rješavanju.
- Programi za igranje šaha ili pokera koji optimiziraju strategiju igre.

1.2. Prediktivna analitika u obrazovanju i primjena strojnog učenja u predviđanju akademskog uspjeha

U obrazovanju se strojno učenje koristi za analizu podataka o studentima s ciljem predviđanja njihovog akademskog uspjeha, prepoznavanja studenata kojima treba dodatna podrška, te za unaprjeđenje metoda poučavanja.

Prediktivna analitika predstavlja skup tehnika i metoda koje omogućuju predviđanje budućih događaja na temelju analize postojećih podataka. Kako Abbott navodi: „prediktivna analitika je vođena podacima, što znači da algoritmi iz samih podataka izvode ključne karakteristike modela, a ne iz pretpostavki koje postavlja analitičar“ (Abbott, 2014, str. 3.). Korištenjem različitih statističkih i algoritamskih pristupa, prediktivna analitika omogućuje donošenje odluka i otkrivanje obrazaca koji nisu vidljivi klasičnim metodama analize podataka. U visokom obrazovanju prediktivna analitika se sve više koristi za unaprjeđenje studentske podrške, optimizaciju obrazovnih procesa i povećanje stope završavanja studija. S obzirom na količinu podataka o studentima koja je dostupna obrazovnim ustanovama - od srednjoškolskog uspjeha i rezultata prijemnih ispita do akademskih rezultata tijekom studiranja, obrazovne institucije mogu primijeniti modele strojnog učenja kako bi identificirale studente u riziku od odustajanja i pravovremeno poduzele mjere podrške. Prediktivna analitika razlikuje se od drugih oblika analize

podataka po tome što je izrazito vođena podacima. Kod prediktivne analitike algoritmi samostalno otkrivaju obrasce i konstruiraju modele bez potrebe za unaprijed definiranim pretpostavkama analitičara. Koristi povijesne i trenutačne podatke te uz pomoć naprednih tehnika poput statistike, strojnog učenja, rudarenja podataka ne određuje samo koeficijente i parametre modela, već prepoznaje i koje varijable treba uključiti i koji model koristiti (Kumar, 2018). Time se omogućuje automatizacija složenih analitičkih procesa koji bi kada bi se radili ručno, zahtijevali mnogo vremena i pažnje osobito kod velikog broja varijabli.

Prediktivna analitika u obrazovanju ima jako puno praktičnih primjena, a neke od najvažnijih su:

1. Identifikaciju studenata koji imaju povećan rizik od akademskog neuspjeha ili odustajanja (Moyano, 2023).
2. Na temelju predikcija obrazovne institucije mogu pružiti personalizirane savjete studentima (Spector, 2024).
3. Poboljšanje nastavnih programa: Prediktivni modeli mogu pomoći u analizi uspješnosti određenih kolegija i programa (Papamitsiou, 2014).

Prediktivna analitika koristi razne modele i metode kako bi razvila predviđanja. Metode uglavnom možemo podijeliti u nekoliko glavnih skupina:

1. **Prediktivni modeli** - Ovi modeli koriste postojeće podatke kako bi predvidjeli akademski uspjeh pojedinog studenta. Mogu predvidjeti hoće li student završiti studij ili odustati na temelju njegovih srednjoškolskih ocjena i uspjeha u prvim godinama studija.
2. **Deskriptivni modeli** - Analiziraju obrasce među studentima, otkrivaju kako određeni čimbenici utječu na akademski uspjeh i pomažu kod svrstavanja studenata u grupe prema razini rizika od neuspjeha.
3. **Model donošenja odluka** - Koriste se za preporuke mjera koje obrazovne institucije mogu poduzeti kako bi poboljšale studentsku podršku i broj onih koji odustanu od studija.

1.2.1. Tehnike prediktivne analitike

Za razvoj modela prediktivne analitike primjenjuju su različite metode strojnog učenja. Neke od najčešće korištenih su:

1. **Logistička regresija** (engl. *Logistic regression*) - Model koji se koristi za predviđanje binarnih ishoda (u našem istraživanju koristimo logističku regresiju kako bi saznali ishod studiranja, tj. je li student diplomirao ili odustao) (Ujkani, 2022). Ovom metodom možemo bolje razumjeti kako različiti akademski čimbenici utječu na završetak studija.
2. **Stablo odluke** (engl. *Decision tree*) - Algoritam koji na jednostavan i vizualno jasan način prikazuje faktore koji utječu na akademski uspjeh. Stablo odluke postavlja niz uzastopnih pitanja na temelju podataka i „grana“ se kao stablo, te naposljetku vodi do glavne odluke.
3. **Slučajna šuma** (engl. *Random forest*) - Naprednija varijanta stabla odluke koja ne koristi samo jedno stablo nego uz kombinaciju više njih povećava preciznost i pouzdanost rezultata. Na taj način se smanjuje rizik od „prenaučivosti“ modela koji bi lošije predviđao nove situacije.
4. **Stroj vektora odluke** (engl. *Support vector machines*) - SVM je model koji pokušava pronaći optimalnu granicu između studenata koji završavaju i onih koji odustaju na temelju analize značajki (Torres Cruz, 2024.).
5. **K-means grupiranje** (engl. *K-means clustering*) - Nenadzirana metoda koja bez prethodnih oznaka ili kategorija sama pronalazi obrasce u podacima i grupira studente u različite skupine prema na primjer sličnosti akademskog uspjeha (Baradwaj, 2012.).

1.3. Predviđanje završetka studija: logistička regresija kao najučinkovitiji pristup

U današnje vrijeme sve veća količina dostupnih podataka otvara mogućnosti za primjenu strojnog učenja u različitim područjima pa tako i u samom obrazovanju. Jedan od izazova u visokom obrazovanju je kako na vrijeme prepoznati studente koji su podložni odustajanju od studija. Ako se takvi studenti detektiraju na vrijeme, obrazovne institucije

moгу pravovremeno reagirati i poduzeti konkretne mjere koje mogu pomoći da više studenata uspješno završi upisani studij. Koristeći logističku regresiju, jednu od temeljnih metoda u statističkom modeliranju i strojnome učenju analizirani su čimbenici koji utječu na akademski uspjeh studenata. Model logističke regresije pripada generaliziranim linearnim modelima (engl. Generalized Linear Models) koje su popularizirali McCullagh i Nelder u istoimenoj knjizi koja je objavljena 1982. godine. (J.A., 1982.)

Logistička regresija često se koristi kada je zavisna varijabla binarna, odnosno kada želimo klasificirati podatke u dvije kategorije (npr. završio studij ili odustao). Logistička regresija vrsta je regresijske analize u kojoj je ciljna (zavisna) varijabla binarna. Poprima samo dvije moguće vrijednosti, koje su najčešće označene s 0 i 1. Model polazi od pretpostavke da postoji barem jedna neovisna (prediktorska) varijabla pomoću koje se pokušava procijeniti vjerojatnost nastanka određenog ishoda. U praksi se koristi za modeliranje problema u kojima se željeni ishod može izraziti kao kategorijska varijabla na primjer: da ili ne, prisutno ili odsutno, uspješan ili neuspješan ishod.

1.4. Pregled literature

Posljednjih dvadesetak godina raste interes za primjenu strojnog učenja u obrazovanju, posebno kad je riječ o predviđanju uspjeha studenata. Ovakvi pristupi nastoje na vrijeme prepoznati studente kojima prijeti odustajanje od studija ili koji bi mogli imati teškoće u učenju, a sve kako bi im se pružila pravovremena pomoć i podrška. Odabir algoritama, značajki i pristupa obradi podataka ovisi o konkretnom istraživačkom pitanju i kontekstu u kojem se studija provodi.

Jedan od ranijih i često citiranih radova u ovom području je istraživanje Kotsiantisa, Pierrakeasa i Pintelasa (Kotsiantis, 2004.). Koristili su algoritme logističke regresije, stabla odluke i neuronskih mreža za predviđanje uspjeha studenata u obrazovanju na daljinu. Njihovo istraživanje naglašava važnost pažljive selekcije značajki te pokazuje da kombinacijom više modela možemo povećati točnost predikcije. Kasnije, Kotsiantis predlaže i sustav za podršku odlučivanju koji implementira strojno učenje za kontinuirano praćenje studentskog uspjeha, što dodatno potvrđuje potencijal prediktivnih modela u obrazovnim okruženjima (Kotsiantis, 2012).

Romero i Ventura u svom radu daju sveobuhvatan pregled tadašnjih tehnika koje se koriste u području edukacijskog rudarenja podataka (EDM). Riječ je o području koje spaja znanje

iz informatike i obrazovanje, a sve sa ciljem da se iz velikih količina podataka prikupljenih u obrazovnim sustavima izvuku korisne informacije koje mogu pomoći u unapređenju nastavnog procesa i studentskih ishoda. U svom pregledu autori detaljno opisuju primjenu metoda strojnog učenja za zadatke kao što su klasifikacija (hoće li student položiti ispit), regresija (predviđanje prosjeka ocjena) i grupiranje (otkrivanje tipova ponašanja studenata). Ističu da su se u to vrijeme najčešće koristili algoritmi logističke regresije, Bayesove mreže, neuronske mreže i razne ansambl metode poput slučajne šume. Već tada je uočena tendencija primjene jednostavnijih i interpretabilnijih modela u kombinaciji s onima koji daju višu točnost, ali zahtijevaju više računalne snage i tehničkog znanja. Romero i Ventura zaključuju da je primjena strojnog učenja već tada bila čvrsto prisutna u obrazovnoj analitici, osobito u kontekstima gdje su obrazovne institucije imale pristup kvalitetnim i strukturiranim podacima o bivšim studentima. Njihov rad dodatno ističe važnost suradnje između nastavnika, stručnjaka za podatke i informatičkih sustava kako bi se rezultati takvih modela mogli praktično primijeniti za poboljšanje nastavnih procesa i studentske podrške (Ventura, 2010).

Shmueli i Koppius u svom radu naglašavaju da je ključno već na početku jasno odrediti je li cilj modela nešto objasniti ili nešto predvidjeti, jer ta razlika uvelike utječe na izbor metoda i način interpretacije rezultata. U kontekstu predviđanja akademskog uspjeha, cilj nije nužno razumjeti sve uzroke i odnose među podacima, već razviti model koji može pouzdano predvidjeti konkretni ishod, poput završetka studija. Ovakvi modeli osobito su vrijedni u situacijama kada je važno rano uočiti studente u riziku od odustajanja od studija i kako bi se pravovremeno poduzele odgovarajuće mjere podrške (Koppius G. S., 2011).

Baker i Inventado u svom poglavlju o edukacijskom rudarenju podataka detaljno opisuju kako se različiti algoritmi strojnog učenja primjenjuju u obrazovnom kontekstu. Ističu važnost korištenja podataka kao što su prisutnost, aktivnost na e-učenju, ocjene i prolaznost čime se omogućuje bolje razumijevanje studentskog ponašanja i potencijalno ranije otkrivanje problema. Njihov rad pokazuje da je prediktivna analitika korisna ne samo za krajnji ishod studija, nego i za optimizaciju svakodnevne obrazovne prakse (Baker, 2014).

2. Metodologija

Uzimajući u obzir dosadašnja istraživanja, u ovom radu primijenjeni su modeli nadziranog strojnog učenja. Primijenjene su četiri metode nadziranog strojnog učenja: linearna regresija, logistička regresija, stablo odluke i slučajna šuma. Svaka od tih tehnika ima svoje prednosti, a zajedno su korištene kako bi se postigla bolja pouzdanost predikcija i omogućila usporedba različitih pristupa. U ovom istraživanju korišteni su podaci iz ISVU sustava koji uključuju prosjek ocjena iz srednje škole, bodove na državnoj maturi i ukupne bodove za upis na studij. Na temelju prikupljenih podataka trenirani su modeli strojnog učenja kako bi se procijenila vjerojatnost da će studenti završiti studij. Model koji predviđa vjerojatnost završetka fakulteta nadograđuje se na model analize akademskog uspjeha i zajedno čine cjelokupnu sliku o ponašanju i napredovanju studenata tijekom studiranja.

U prvom dijelu ovog istraživanja korištena je logistička regresija. Na temelju dostupnih akademskih podataka navedenom metodom nadziranog učenja pokušalo se predvidjeti hoće li student završiti studij ili odustati. Model je analizirao značajke poput prosjeka ocjena iz srednje škole, broja bodova s državne mature i ukupnog broja bodova za upis na studij.

Osim klasifikacijskog modela, u drugom dijelu istraživanja predviđao se prosjek ocjena studenata tijekom studiranja. U tom dijelu istraživanja korištene su regresijske metode - linearna regresija i model slučajne šume kojima je cilj bio procijeniti prosječnu ocjenu studenta na temelju broja godina studiranja. Na taj način pokušalo se dublje razumjeti postoji li veza između duljine studiranja i akademske uspješnosti.

Kako bi se procijenila točnost modela korištene su metrike: srednja apsolutna pogreška (MAE), srednja kvadratna pogreška (MSE), korijen srednje kvadratne pogreške (RMSE) i koeficijent determinacije (R^2). Rezultati metrika pomogli su da usporedba bude objektivna koliko je koji model dobro „predviđao“ stvarne vrijednosti. Kombinacijom klasifikacijskih i regresijskih pristupa, istraživanje je pružilo širu sliku o obrazovnim ishodima studenata i pružilo bolji uvid u to koji čimbenici najviše utječu na uspjeh studenata. U nastavku rada detaljno su opisani svi koraci obrade podataka, treniranja modela, evaluacije rezultata i analize varijabli koje najviše utječu na akademski uspjeh studenata.

Evaluacija modela izvršena je pomoću:

- Točnosti klasifikacije (engl. accuracy score)
- Matrice konfuzije
- ROC-AUC krivulje koja pokazuje koliko je model dobar u razlikovanju između studenata koji završavaju i onih koji odustaju.

2.1. Pregled podataka i odabir skupa

U ovom istraživanju korišteni su anonimizirani podaci preuzeti iz ISVU-u sustava. Svi osobni podaci su uklonjeni u skladu s GDPR-om kako bi se zaštitila privatnost studenata. Podaci sadrže informacije o studentima i njihovom akademskom putu. Cilj je koristiti ih za predviđanje akademskog uspjeha studenata - hoće li studenti diplomirati ili odustati od studiranja i s obzirom na godine studiranja koji prosjek ocjena je očekivan. Na raspolaganju su bila četiri različita skupa podataka od kojih svaki sadrži različite podatke o studentima:

1. Ciljana skupina - opći podatci (JMBAG, državna matura, srednja škola, opći podatci o studiju)
2. Predmeti - upisano/položeno (informacije o pojedinačnim kolegijima koje su studenti upisali i položili)
3. Studij - dodatni elementi (detalji o specifičnim akademskim aspektima studiranja)
4. Završetak studija - prosjek, duljina studiranja (ključne informacije o akademskom uspjehu studenata, uključujući prosjek ocjena, broj položenih ispita i trajanje studiranja)

Podaci su pohranjeni na Google Drive-u pa je potrebno prvo povezati se sa „Google Drive“ platformom kako bih im mogli pristupiti u okruženju Google Colab. Google Colab je besplatno okruženje u kojem je moguće pisanje i izvršavanje Python koda izravno na web pregledniku, odnosno nije potrebna dodatna instalacija softvera na računalu. Ovim kodom omogućujemo pristup datotekama pohranjenim na Google Drive-u. Kako bih radili s datotekama unutar određenog direktorija, korištena je naredba `%cd /content/drive/MyDrive/Petra diplomski` za promjenu direktorija. Ova naredba postavlja radni direktorij na mapu „Petra diplomski“ gdje se nalaze svi podaci za daljnje analiziranje.

Na taj način datoteke se mogu direktno učitavati bez potrebe za ručnim unosom punih putanja prilikom svakog poziva funkcije za čitanje podataka.

```
[1] #from drive
    from google.colab import drive
    drive.mount('/content/drive')

Mounted at /content/drive

[2] %cd /content/drive/MyDrive/Petra diplomski

/content/drive/MyDrive/Petra diplomski
```

Slika 1 - Naredbe za povezivanje sa Google Drive-om i promjenu direktorija

Za analizu su korištena dva skupa podataka:

1. CILJANA SKUPINA - opći podatci (JMBAG, državna matura, srednja škola, opći podaci o studiju)
2. ZAVRŠETAK STUDIJA - prosjek, duljina studiranja

Ostali skupovi podataka nisu korišteni iz razloga što nisu sadržavali ključne informacije za istraživanje. Također, pokušaj spajanja svih skupova pokazao se problematičnim jer nisu imali zajedničke značajke koje bi omogućile povezivanje podataka na smislen način.

Skupovi podataka korišteni za istraživanje sadržavali su ključne informacije potrebne za predviđanje završetka studija: CILJANA SKUPINA opći podatci (JMBAG, državna matura, srednja škola, opći podaci o studiju) i ZAVRŠETAK STUDIJA prosjek, duljina studiranja. Kombinacijom ovih dvaju skupova podataka pružena je dobra osnova za izgradnju modela logističke regresije, linearne regresije, slučajne šume i stabla odluke. Pomoću ovih model predviđa se hoće li student završiti studij ili odustati i služe za predviđanje akademskog uspjeha tj. prosjeka ocjena studenata.

```
# Učitavanje podataka
file_path_opci = '/content/drive/MyDrive/Petra diplomski/CILJANA SKUPINA - opći podatci (JMBAG, DM, SŠ, studij opći podatci).xlsx'
data_opci = pd.read_excel(file_path_opci)
```

Slika 2 - Prikaz naredbi za učitavanje podataka iz Excel datoteke

```
# Učitavanje data seta
file_path_podaci = '/content/drive/MyDrive/Petra diplomski/završetak studija - prosjek, duljina studiranja.xlsx'
data = pd.read_excel(file_path_podaci)
```

Slika 3 - Prikaz naredbi za učitavanje podataka iz Excel datoteke

2.1.1. Ciljana skupina - opći podatci

Ovaj skup podataka sadrži demografske i akademske podatke o studentima prije nego što su upisali fakultet. Informacije koje se nalaze u ovom skupu su informacije o uspjehu u srednjoj školi i rezultatima na državnoj maturi. To su ključni čimbenici koji mogu utjecati na akademski uspjeh. Popis značajki koje se nalaze u ovom skupu podataka:

Godina upisa, Godina rođenja, Spol (M/Ž), Završena srednja škola (srednje škole koju je student završio), Prosjek ocjena iz srednje škole, Bodovi na državnoj maturi, Bodovi za dodatna postignuća (natjecanja), Ukupni bodovi za upis na studij, Rezultati mature po predmetima (Engleski, Hrvatski, Matematika, Informatika), Status studenta - ključna varijabla koja označava završetak studija (diplomirao, ispisan, aktivan studij), Datum diplomiranja/ispisa (informacija o završetku studija ili ispisu).

Ovaj skup podataka omogućuje analizu veze između srednjoškolskog uspjeha i završetka studija.

2.1.2. Završetak studija - prosjek, duljina studiranja

Skup podataka sadrži informacije o uspjehu studenata tijekom njihovog akademskog puta i omogućuje detaljnu analizu napretka samih studenata. Popis značajki koje se nalaze u ovom skupu podataka: *Datum završetka studija, Akademska godina završetka studija, Akademska godina zadnjeg upisa (posljednja godina u kojoj je student bio upisan), Prosjek ocjena na studiju, Težinski prosjek ocjena (prosjek ponderiran prema ECTS bodovima), Ukupno ECTS bodova, Položeno ispita, Broj predmeta, Broj godina studiranja.*

Na temelju dostupnih informacija ovog skupa podataka analizirano je može li se pomoću značajki vezanih uz akademski uspjeh studenata predvidjeti hoće li netko završiti studij ili će od njega odustati. U nastavku je detaljno opisan postupak pripreme podataka, izgradnje modela i analize dobivenih rezultata.

Naredbom „`data.head()`” imamo uvid u pregled prvih nekoliko redaka skupa podataka i na taj način imamo uvid u pregled strukture podataka. Pomoću ove naredbe prikazani su: nazivi stupaca, prve vrijednosti u skupu podataka, tipovi podataka (brojčani, kategorizirani, tekstualni, datumi). Ovom funkcijom provjeravamo jesu li podaci ispravno učitani, a ujedno imamo uvid u njihov sadržaj što je korisno prije početka detaljne analize i obrade.

✓ [6] data.head()

	Datum završetka studija	JMBAG	Ak. god. završetka studija	Ak. god. zadnjeg upisa na studiju	Prosjek ocjena na studiju	Težinski prosjek ocjena na studiju	Ukupno ECTS bodova	Položeno ispita	Broj predmeta	Student studirao godina
0	2015-07-08	1191207947	2014./2015.	2014./2015.	3.611	3.480	181	36	38	5.77
1	2015-07-10	177042905	2014./2015.	2014./2015.	4.595	4.460	180	37	41	2.77
2	2015-09-25	177042989	2014./2015.	2014./2015.	3.486	3.267	180	37	41	2.98
3	2015-09-25	177042840	2014./2015.	2014./2015.	3.541	3.392	180	37	41	2.98
4	2015-09-24	82040736	2014./2015.	2014./2015.	3.676	3.534	180	37	41	2.98

Slika 4 - Pregled strukture skupa podataka „završetka studija - prosjek, duljina studiranja.xlsx“

Ova tablica prikazuje prvih pet redaka skupa podataka i na taj način dobijemo pregled informacija o studentima koji su završili studij.

Pregled stupaca i njihovog značenja: Datum završetka studija - točan datum kada je student diplomirao, JMBAG - jedinstveni identifikacijski broj studenta (anonimiziran), Ak. god. završetka studija - akademska godina u kojoj je student završio studij, Ak. god. zadnjeg upisa na studiju - posljednja godina kada je student bio upisan, prosjek ocjena na studiju - prosječna ocjena studenta tijekom studiranja, Težinski prosjek ocjena na studiju - težinski prosjek ocjena prema ECTS bodovima, Ukupno ECTS bodova - ukupan broj ECTS bodova koje je student stekao, Položeno ispita - ukupan broj položenih ispita, Broj predmeta - ukupan broj predmeta koje je student imao tijekom studija, Student studirao godina - ukupan broj godina koje je student proveo na studiju prije završetka. Iz pregleda podataka zaključujemo da su vrijednosti numeričke, osim akademskih godina i datuma. Prosjek ocjena varira i to nam može biti korisno za predviđanje uspjeha. Broj godina studiranja nije jednak za sve što ukazuje na različit tijek akademskog uspjeha studenata. Uvidom u podatke lakše je pristupiti daljnjoj analizi kao što su vizualizacija podataka, normalizacija i odabir značajki za modele strojnog učenja.

2.2. Instalacija paketa

Nakon uspješnog učitavanja podataka, sljedeći korak je instalacija i učitavanje potrebnih paketa za izgradnju modela. Budući da standardna Python biblioteka ne sadrži sve alate potrebne za analizu podataka i izgradnju modela strojnog učenja, bilo je potrebno instalirati i učitati dodatne pakete. Ovi paketi omogućuju učinkovitiju obradu podataka, pripremu modela, njihovu evaluaciju te vizualizaciju rezultata.

```
[3] # Prvo pokrenite ove naredbe za instalaciju potrebnih paketa (ako već nisu instalirani)
!pip install pandas matplotlib scikit-learn openpyxl
```

Slika 5 - Naredba za instalaciju potrebnih biblioteka

Nakon instalacije potrebnih biblioteka sljedeći korak bio je učitavanje paketa koji će se koristiti za obradu podataka, treniranje modela i vizualizaciju rezultata. Svaki od učitanih modula ima specifičnu ulogu u analizi i modeliranju.

```
[4] import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
import seaborn as sns
import matplotlib.pyplot as plt
```

Slika 6 - Učitavanje paketa

1. Obrada podataka

Ovi paketi služe za učitavanje i obradu podataka kako bi ih pripremili za rad modela strojnog učenja:

Biblioteka „pandas“ - omogućuje rad s tabličnim podacima i manipuliranjem istih.

Biblioteka „numpy“ - pruža podršku za numeričke operacije i rad s nizovima podataka.

2. Podjela podataka na trening i testni skup

Biblioteka „train_test_split“ iz „sklearn.model_selection“ - Dijeli podatke na trening (engl. *train*) i testni (engl. *test*) skup. Ovo je ključno za procjenu performansi modela.

3. Predobrada podataka

Za uspješno treniranje modela potrebno je pripremiti podatke:

Biblioteka „StandardScaler“ - Standardizira numeričke podatke (npr. normalizacija ocjena).

Biblioteka „OneHotEncoder“ - Pretvara kategorizirane varijable (npr. tip završene srednje škole) u numerički oblik.

Biblioteka „SimpleImputer“ - Popunjava nedostajuće vrijednosti kao srednju vrijednost ili medijan.

Biblioteka „ColumnTransformer“ - Omogućuje paralelnu transformaciju numeričkih i kategoriziranih varijabli.

4. Automatski tijek obrade podataka

Biblioteka „Pipeline“ - Omogućuje stvaranje automatskog procesa za obradu podataka. Tako se osigurava da svi podaci budu obrađeni na isti način, posebice kada se model trenira. Tako osiguravamo konzistentnost i dosljednost podataka kao i samih rezultata.

5. Algoritmi strojnog učenja

Za predviđanje akademskog uspjeha koriste se različiti modeli:

Biblioteka „RandomForestClassifier“ - Algoritam slučajne šume koristi više stabala odluke za klasifikaciju (npr. hoće li student završiti studij ili odustati).

Biblioteka „DecisionTreeRegressor“ - Model stabla odluke za regresijske probleme (npr. predviđanje trajanja studija na temelju ocjena).

Biblioteka „RandomForestRegressor“ - Naprednija verzija stabla odluke koja koristi više stabala kako bi poboljšala točnost regresijskih predikcija.

Biblioteka „LinearRegression“ - Klasična metoda linearne regresije koja se koristi za predviđanje kontinuiranih varijabli.

6. Evaluacija modela

Kako bi se procijenila učinkovitost modela koriste se metričke funkcije:

Biblioteka „classification_report“ - Koristi se za evaluaciju klasifikacijskih modela (točnost, preciznost, F1-mjera).

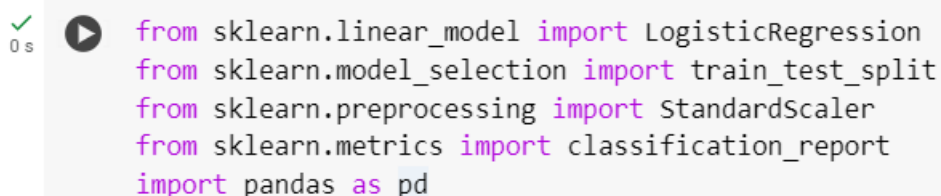
Biblioteke „mean_absolute_error“ (MAE), „mean_squared_error“ (MSE), „r2_score“ - Koriste se za evaluaciju regresijskih modela i pokazuju koliko su predviđanja modela precizna.

7. Vizualizacija podataka i rezultata

Za bolji uvid u podatke i rezultate modela korišteni su paketi za vizualizaciju:

Biblioteka „matplotlib.pyplot“ - Kreira različite vrste grafikona (histogrami, dijagrami).

Biblioteka „seaborn“ - Napredniji alat za vizualizaciju koji olakšava analizu podataka.



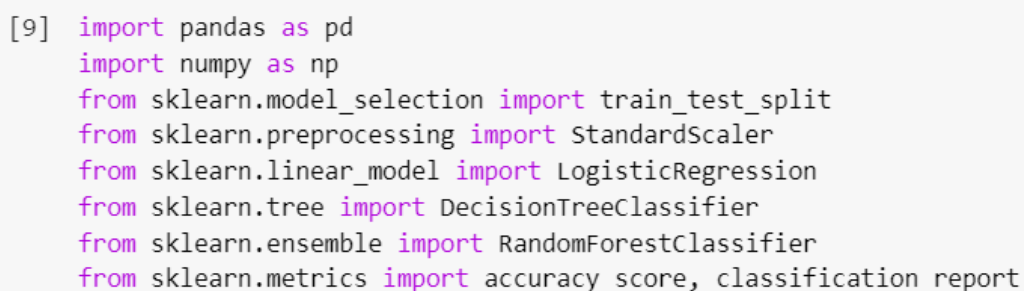
```
0 s  from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    from sklearn.metrics import classification_report
    import pandas as pd
```

Slika 7 - instalacija potrebnih paketa za izgradnju modela logističke regresije

LogisticRegression iz *sklearn.linear_model* paketa koristi se za binarnu ili višeklasnu klasifikaciju podataka, dok se *train_test_split* iz *sklearn.model_selection* paketa koristi za podjelu podataka za trening i test skup.

StandardScaler iz *sklearn.preprocessing* paketa služi kako bi standardizirao podatke tako da imaju srednju vrijednost 0 i standardnu devijaciju 1, a *classification_report* iz *sklearn.metrics* paketa generira sažetak metrika klasifikacije kao što su preciznost, odziv i F1-mjera.

import pandas as pd uobičajeni je paket koji se koristi za rad s tabličnim podacima, najčešće za učitavanje i osnovnu obradu podataka.



```
[9] import pandas as pd
    import numpy as np
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    from sklearn.linear_model import LogisticRegression
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score, classification_report
```

Slika 8 - instalacija paketa potrebnih za modele strojnog učenja za predviđanje akademskog uspjeha studenta

U ovom dijelu korišten je manji skup osnovnih Python paketa koji su potrebni za učitavanje i obradu podataka, izgradnju klasifikacijskih modela i evaluaciju njihove točnosti. Paket *pandas* koristi se za učitavanje i manipulaciju tabličnim podacima, dok paket *numpy* olakšava rad s brojčanim nizovima i vektorima. Korištena je funkcija

`train_test_split` iz `sklearn.model_selection` paketa za podjelu podataka na trening i testni skup. Za standardizaciju podataka koristi se *funkcija* `StandardScaler` koja sve numeričke značajke dovodi na istu skalu. Ova funkcija važna je kod algoritama koji u obzir uzimaju razlike među vrijednostima. Može se dogoditi da ako su neki podaci puno veći od drugih da nepravедno dominiraju pri učenju modela.

Modeli koji su primijenjeni uključuju *LogisticRegression*, algoritam nadziranog učenja koji se koristi za zadatke binarne klasifikacije. *DecisionTreeClassifier* gradi hijerarhijsko stablo odluka koje se temelji na značajkama primjera. *RandomForestClassifier* koristi više stabala odluke kako bi poboljšao točnost i smanjio mogućnost prenaučенosti. Za evaluaciju rezultata korištene su metrike iz paketa `sklearn.metrics` i to funkcija `accuracy_score` koja mjeri točnost klasifikacije i funkcija `classification_report` koja daje detaljan pregled preciznosti, odziva i F1 rezultata modela.

2.3. Distribucija završetka fakulteta

Kako bi se dobio bolji uvid u strukturu podataka i status studenata u skupu podataka, kreiran je kružni dijagram (engl. *pie chart*). Kružnim dijagramom prikazan je broj i postotak studenata koji su diplomirali, ispisali se ili su još uvijek aktivni na studiju.

```
# Priprema podataka za pie chart
status_counts = data_opci['STATUS (diplomirao, ispisan, aktivan studij)'].value_counts().reset_index()
status_counts.columns = ['Status', 'Count']

# Kreiranje pie charta
fig = px.pie(status_counts, names='Status', values='Count',
             color='Status', color_discrete_map={
                 'diplomirao': 'lightblue',
                 'ispisan': 'lightpink',
                 'aktivan studij': 'lightgreen'})

# Dodavanje postotka i broja informacija na pie chart
fig.update_traces(textinfo='label+percent',
                  text=[f'{count}' for count in status_counts['Count']])

# Ažuriranje izgleda
fig.update_layout(title_text="Distribucija završetka fakulteta", title_x=0.5)

fig.show()
```

Slika 9 - Naredbe za prikaz distribucije završetka fakulteta

Prije same vizualizacije bilo je potrebno pripremiti podatke. Iz skupa podataka izdvojen je stupac „STATUS (diplomirao, ispisan, aktivan studij)“, koji sadrži informacije o statusu svakog studenta. Primijenjena je funkcija `value_counts()` kako bi se izračunao broj

studenata u svakoj kategoriji. Rezultat je pretvoren u tablicu (engl. *DataFrame*) sa stupcima:

1. „Status“ – prikazuje status studenta (diplomirao, ispisan, aktivan studij).
2. „Count“ – prikazuje broj studenata u svakoj kategoriji.

Kod:

```
status_counts = data_opci['STATUS (diplomirao, ispisan, aktivan studij)'].value_counts().reset_index()

status_counts.columns = ['Status', 'Count']
```

Objašnjenje koda:

`data_opci['STATUS (diplomirao, ispisan, aktivan studij)'].value_counts()` - broji pojavljivanja svake vrijednosti (diplomirao, ispisan, aktivan studij).

`.reset_index()` - Pretvara rezultate u DataFrame za lakšu daljnju obradu.

`status_counts.columns = ['Status', 'Count']` - Postavlja nazive stupaca: „Status“ - kategorije statusa studenata i „Count“ - broj studenata u svakoj kategoriji.

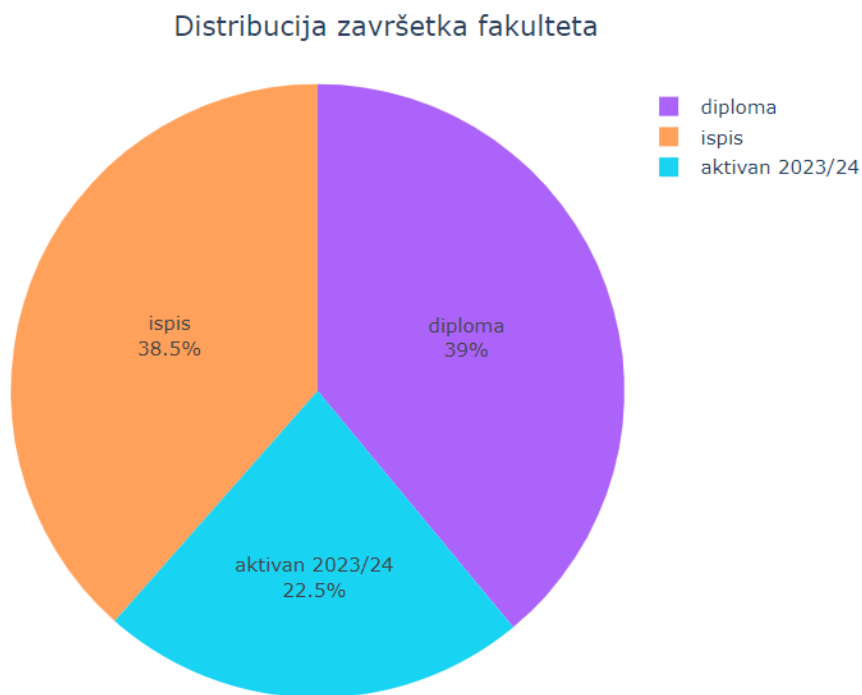
2.3.1. Kreiranje kružnog dijagrama (engl. *pie charta*)

Kružni dijagram omogućuje procjenu distribucije studenata i služi kao oslonac za daljnju analizu akademskog uspjeha. Ovaj dijagram prikazuje jasnu distribuciju studenata prema njihovom akademskom statusu. Grafičkim prikazom vidimo koji status studenata dominira (aktivan, diplomirao, ispisan). Također, mogu se identificirati problemi ili trendovi - ako je veliki broj studenata ispisan to može biti znak da postoje problemi u studijskom programu ili u sustavu studiranja.

Vizualizacija podataka ključna je zbog razumijevanja obrazaca i lakšeg donošenja zaključaka. Korisna je za razvoj prediktivnog modela strojnog učenja koji analizira akademski uspjeh studenata.

Za vizualizaciju je korištena biblioteka *Plotly Express* koja omogućuje dinamičko prikazivanje podataka. Dijagram prikazuje postotak studenata u svakoj kategoriji. Svaka kategorija prikazana je različitom bojom kako bi dijagram bio pregledniji: Ljubičasta (engl. *purple*) označava studente koji su diplomirali, narančasta (engl. *orange*) označava studente

koji su se ispisali sa fakulteta, svjetlo plava (engl. *lightblue*) označava studente koji su još uvijek aktivni odnosno studiraju.



Slika 10 - Distribucija završetka fakulteta prikazana kružnim dijagramom

Kod:

```
fig = px.pie(status_counts, names='Status', values='Count',
             color='Status', color_discrete_map={
                 'diplomirao': 'lightblue',
                 'ispisan': 'lightpink',
                 'aktivan studij': 'lightgreen'})
fig.update_traces(textinfo='label+percent',
                  text=[f'{count}' for count in
                        status_counts['Count']])
fig.update_layout(title_text="Distribucija završetka fakulteta",
                  title_x=0.5)
fig.show()
```

Objašnjenje koda:

`px.pie()` - funkcija iz Plotly Express koja kreira kružni dijagram.

`names='Status'` - definira oznake na dijagramu (kategorije statusa).

`values='Count'` - određuje veličinu svakog segmenta prema broju studenata.

`color_discrete_map={...}` - Pridružuje određene boje svakoj kategoriji radi bolje preglednosti: Ljubičasta (engl. *purple*) označava studente koji su diplomirali, narančasta (engl. *orange*) označava studente koji su se ispisali sa fakulteta, svjetlo plava (engl. *lightblue*) označava studente koji su još uvijek aktivni odnosno studiraju.

`textinfo='label+percent'` - na kružnom dijagramu prikazuje naziv kategorije (Diplomirao, Ispisan, Aktivan studij) i postotak pripadnosti svake kategorije u ukupnom broju studenata.

`text=[f'{count}' for count in status_counts['Count']]` - prikazuje apsolutne vrijednosti (broj studenata) uz postotke.

`title_text="Distribucija završetka fakulteta"` - prikaz naslova dijagrama.

`title_x=0.5` - centriranje naslova iznad dijagrama.

`fig.show()` - Prikazuje dijagram s definiranim izgledom i svim pripadajućim podacima.

2.4. Analiza povezanosti prosjeka ocjena i duljine studiranja

Važan čimbenik u razumijevanju akademskog uspjeha studenata je istražiti kako je prosjek ocjena tijekom studija povezan s duljinom trajanja studiranja. Ovakva vrsta analize pokazuje imaju li studenti s višim prosjekom ocjena tendenciju završiti studij u kraćem vremenskom razdoblju (bez produživanja tj. da sve kolegije položi u zadanim rokovima) ili postoji neka druga povezanost.

Kako bi tu povezanost vizualno prikazali i lakše uočili koristi se raspršeni dijagram (engl. *scatter plot*) pomoću kojeg imamo pregled svakog pojedinog studenta kao točke u prostoru definiranom s dvije varijable:

1. x-osi: prosjek ocjena studenta,
2. y-osi: broj godina koje je student proveo na studiju do završetka studiranja.

```
# Kreiranje scatter plota za analizu duljine studiranja s obzirom na prosjek ocjena
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='Prosjek ocjena na studiju', y='Student studirao godina')
plt.title('Duljina studiranja s obzirom na prosjek ocjena')
plt.xlabel('Prosjek ocjena na studiju')
plt.ylabel('Duljina studiranja (godine)')
plt.grid(True)
plt.show()
```

Slika 11 - Naredbe za prikaz analize povezanosti prosjeka ocjena i duljine studiranja

2.4.1. Kreiranje raspršenog dijagrama (engl. *scatter plot*)

Za analizu povezanosti akademskog uspjeha i duljine trajanja studija, izrađen je raspršeni dijagram (engl. *scatter plot*) koji prikazuje odnos između prosjeka ocjena studenata i duljine trajanja njegovog studiranja. Svaka točka na grafu predstavlja jednog studenta. Na x-osi nalazi se prosjek ocjena pojedinog studenta, a na y-osi koliko godina mu je trebalo da završi fakultet.

Vizualizacija je napravljena korištenjem biblioteka Matplotlib i Seaborn. Standardnih alata u Pythonu koji se koriste za analizu i prikaz podataka. Prvo se funkcijom `plt.figure(figsize=(10, 6))` postavlja veličina prozora dijagrama kako bi bio pregledniji. Seaborn funkcija `sns.scatterplot()` koristi za crtanje samog dijagrama. Na x-osi nalaze se podaci o prosjeku ocjena na studiju, dok se na y-osi prikazuje broj godina koje je student proveo na fakultetu do završetka studija.

Svaka točka na grafu predstavlja jednog studenta, a njen položaj u prostoru definira se pomoću vrijednosti prosjeka i trajanja studija. Primjerice, završavaju li studenti s višim prosjekom studij u kraćem roku. Dodan je i naslov dijagrama i oznake osi - `plt.title()`, `plt.xlabel()`, `plt.ylabel()`. Zbog jednostavnije čitljivosti grafa, u vizualizaciju je dodana opcija `plt.grid(True)`. Pomoću opcije `plt.grid(True)` dodane su vodoravne i okomite linije koje nalikuju na koordinatni sustav u pozadini grafa. Vodoravne i okomite linije pomažu da lakše i preciznije očitamo dijagram tj. koji prosjek i koliko godina studiranja određena točka prikazuje.



Slika 12 - Raspršeni dijagram - prikaz odnosa između prosječne ocjene studenta na studiju i duljine trajanja njegovog studiranja

Kod:

```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='Prosjeck ocjena na studiju', y='Student
studirao godina')
plt.title('Duljina studiranja s obzirom na prosjeck ocjena')
plt.xlabel('Prosjeck ocjena na studiju')
plt.ylabel('Duljina studiranja (godine)')
plt.grid(True)
plt.show()
```

Objašnjenje koda:

`plt.figure(figsize=(10, 6))` - postavlja veličinu grafa (širina 10, visina 6) radi bolje preglednosti.

`sns.scatterplot(data=data, x='Prosjeck ocjena na studiju', y='Student studirao godina')` - ovom linijom crtamo raspršeni dijagram gdje je x-os prosjeck ocjena, a y-os duljina studiranja.

`plt.title('Duljina studiranja s obzirom na prosjeck ocjena')` - naslov iznad grafa

```
plt.xlabel('Prosjek ocjena na studiju') - x-os
```

```
plt.ylabel('Duljina studiranja (godine)') - y-os
```

`plt.grid(True)` - pozadinske linije (mreža horizontalnih i vertikalnih linija) radi lakšeg praćenja točaka

```
plt.show() - prikaz grafa
```

2.5. Predviđanje završetka studija

Naredbom „`print(data_opci.columns)`“ ispisujemo sve nazive stupaca u tablici (engl. *Data Frame*) naziva „`data_opci`“. Korisno je pregledati koje sve značajke (varijable) postoje u skupu podataka. Također, na ovaj način provjeravamo ispravnost naziva stupaca i pregledavamo podatke prije daljnje obrade. Prikazom stupaca i njihovih podataka lakše odlučujemo koje ćemo varijable koristiti u modeliranju, a koje eventualno izbaciti ili dodatno obraditi.

```
✓ [11] print(data_opci.columns)
0 s
Index(['godina upisa', 'JMBAG', 'GODINA ROĐENJA', 'SPOL', 'Završena SŠ',
      'prosjek SŠ', 'Bodovi srednje škole', 'Bodovi na državnoj maturi',
      'Bodovi za dodatna postignuća (natjecanja)',
      'Ukupno bodova za upis na studij', 'Engleski jezik (A) %',
      'Engleski jezik (B) %', 'Hrvatski jezik %', 'Matematika (A) %',
      'Matematika (B) %', 'Informatika %',
      'STATUS (diplomirao, ispisan, aktivan studij)',
      'Datum diplomiranja/ispisa'],
      dtype='object')
```

Slika 13 - prikaz svih naziva stupaca u tablici

Nakon učitavanja podataka, naredbom „`print(data_opci.columns)`“ ispisani su svi nazivi stupaca tablice „`data_opci`“. Iz rezultata je vidljivo kako skup podataka sadrži osnovne identifikacijske podatke studenata kao što su: *JMBAG*, *godina upisa*, *datum diplomiranja/ispisa*, *godina rođenja*, *spol*, *završena SŠ*, kao i podatke povezane s upisom na studij - *prosjek SŠ*, *bodovi s mature*, *ukupni bodovi*. Stupac *STATUS (diplomirao, ispisan, aktivan studij)* predstavlja ciljnu varijablu za klasifikaciju.

```
✓ 0s [19] print(f"Broj redaka u originalnom datasetu: {len(data_opci)}")  
↔ Broj redaka u originalnom datasetu: 400  
  
✓ 0s [20] print("Unikatne vrijednosti STATUS stupca:")  
        print(data_opci["STATUS (diplomirao, ispisan, aktivan studij)"].unique())  
↔ Unikatne vrijednosti STATUS stupca:  
   ['ispis' 'diploma' 'aktivan 2023/24']
```

Slika 14 - prikaz broja redaka i unikatnih vrijednosti u tablici

U ovom dijelu analize najprije se ispisuje broj redaka (instanci) izvornog skupa podataka „*data_opci*“. Skup podataka sadrži podatke za 400 studenata. Zatim se analiziraju sve unikatne vrijednosti u stupcu „*STATUS (diplomirao, ispisan, aktivan studij)*“. Stupac sadrži tri različite kategorije: 'ispis', 'diploma' i 'aktivan 2023/24', a svaki od njih označava je li student odustao od studija, je li ga je završio ili je trenutno aktivan (upisan u akademsku godinu 2023./2024.). Pomoću ovih značajki možemo definirati ciljnu klasu za klasifikacijski model. Primjerice, binarnom klasifikacijom možemo predvidjeti je li student završio ili odustao od studija, dok će podaci o aktivnim studentima biti korišteni za testiranje modela i predviđanje rezultata.

2.5.1. Priprema podataka

Za daljnju obradu podataka potrebno je mapirati tekstualne vrijednosti statusa studenata u numeričke oznake. Vrijednosti „*Diplomirao*“, „*Ispisan*“ i „*Aktivan studij*“ zamijenjene su brojevima 1, 0 i 2. Mapiranje podataka iz tekstualnog formata u numerički format nužan je jer neki algoritmi ne mogu izravno raditi s tekstualnim podacima već zahtijevaju numerički format. Nakon mapiranja podataka, distribucijom su prikazane nove numeričke vrijednosti kako bi se dobio uvid u broj studenata po kategorijama:

1. Koliko ih je diplomiralo
2. Koliko odustalo
3. Koliko ih je još aktivno.

```

[21] # Mapiranje statusa
data_opci["status"] = data_opci["STATUS (diplomirao, ispisan, aktivan studij)"].map({
    "Diplomirao": 1,
    "Ispisan": 0,
    "Aktivan studij": 2
})

# Provjera distribucije statusa
print("Distribucija statusa:")
print(data_opci["status"].value_counts(dropna=False))

```

Distribucija statusa:

status	count
NaN	400

Name: count, dtype: int64

Slika 15 - prikaz koda mapiranja studenata i distribucije statusa

```

[22] data_opci["STATUS (diplomirao, ispisan, aktivan studij)"] = data_opci["STATUS (diplomirao, ispisan, aktivan studij)"].str.lower().str.strip()

[23] data_opci["status"] = data_opci["STATUS (diplomirao, ispisan, aktivan studij)"].map({
    "diploma": 1, # Završio studij
    "ispis": 0    # Odustao
})

# Svi statusi koji sadrže riječ 'aktivan' tretiraju se kao studenti koji su još na studiju
data_opci.loc[data_opci["STATUS (diplomirao, ispisan, aktivan studij)"].str.contains("aktivan", na=False), "status"] = 2

[24] print("Distribucija statusa nakon popravka:")
print(data_opci["status"].value_counts())

```

Distribucija statusa nakon popravka:

status	count
1.0	156
0.0	154
2.0	90

Name: count, dtype: int64

Slika 16 - prikaz koda dodatne obrade stupaca kako bi se osigurala konzistentnost pri mapiranju vrijednosti

Prikazan je kod pomoću kojega je izvršena dodatna obrada stupca sa statusom studenata kako bi se osigurala konzistentnost i točnost podataka pri mapiranju vrijednosti. Prvo su sve vrijednosti u stupcu „STATUS (diplomirao, ispisan, aktivan studij)“ pretvorene u mala slova pomoću naredbe „str.lower()“ i uklonjeni su suvišni razmaci s početka i kraja teksta pomoću naredbe „str.strip()“. Na taj način se izbjegavaju pogreške prilikom uspoređivanja tekstualnih podataka. Različiti zapisi poput „Diploma“, „diploma“ ili „DIPLOMA“ više neće biti tretirani kao različite vrijednosti. Uklanjanjem nepotrebnih razmaka i standardizacijom teksta poboljšava se preglednost podataka, smanjuje se zauzimanje memorije, te se smanjuje mogućnost dupliranja kategorija. Prethodno objašnjeno uređivanje teksta rezultira učinkovitijom obradom i boljom pripremom skupa podataka za strojno učenje.

Zatim je provedeno mapiranje statusa: vrijednost „diploma“ označena je brojem 1 (student je završio studij), vrijednost „ispis“ je označen brojem 0 (student je odustao), dok su sve vrijednosti koje u sebi sadrže riječ „aktivan“ označene s brojem 2 (student je još uvijek na studiju). Ovakav pristup pretvara sve tri kategorije statusa u numerički kodirane kategorije.

Na kraju je ispisana distribucija novih vrijednosti iz stupca *status*, iz koje je vidljivo kako je od ukupno 400 studenata: 156 diplomiralo, 154 odustalo, a 90 ih još uvijek studira.

```
[25] data_opci = data_opci[data_opci["status"] != 2] # Brišemo aktivne studente

[26] print("Broj NaN vrijednosti po značajci:")
      print(data_opci[["prosjeck SŠ", "Bodovi na državnoj maturi", "Ukupno bodova za upis na studij"]].isna().sum())
```

Broj NaN vrijednosti po značajci:

prosjeck SŠ	0
Bodovi na državnoj maturi	0
Ukupno bodova za upis na studij	0
dtype:	int64

Slika 17 - Prikaz koda uklanjanja redaka koji nisu potrebni

Kod prikazuje na koji način se uklanjaju redci u kojima je status studenta označen kao „*aktivan*“, odnosno studenti koji još studiraju. Tako se model strojnog učenja trenira isključivo na podacima studenata koji su već završili studij ili su se ispisali, jer samo za njih je poznat ishod (završetak studija ili ispis sa studija).

Nakon filtriranja provjerava se ima li praznih (NaN) vrijednosti u trima značajkama koje će se koristiti za predviđanje modela, a to su: *Prosjeck SŠ*, *Bodovi na državnoj maturi*, *Ukupno bodova za upis na studij*.

Rezultat prikazuje kako nema praznih vrijednosti ni u jednoj od ovih značajki, što znači da se podaci mogu koristiti bez dodatne obrade (npr. imputacije) prije treniranja modela.

```
[32] # Učitavanje podataka
      file_path_opci = '/content/drive/MyDrive/Petra diplomski/CILJANA SKUPINA - opći podatci (JMBAG, DM, SŠ, studij opći podatci).xlsx'
      data_opci = pd.read_excel(file_path_opci)

print(data_opci.columns)
```

Index(['godina upisa', 'JMBAG', 'GODINA ROĐENJA', 'SPOL', 'Završena SŠ',
'prosjeck SŠ', 'Bodovi srednje škole', 'Bodovi na državnoj maturi',
'Bodovi za dodatna postignuća (natjecanja)',
'Ukupno bodova za upis na studij', 'Engleski jezik (A) %',
'Engleski jezik (B) %', 'Hrvatski jezik %', 'Matematika (A) %',
'Matematika (B) %', 'Informatika %',
'STATUS (diplomirao, ispisan, aktivan studij)',
'Datum diplomiranja/ispisa'],
 dtype='object')

Slika 18 - podaci korišteni za logističku regresiju

Nakon učitavanja i pregleda podataka vidljivo je da originalni skup podataka sadrži informacije o 400 studenata, uključujući njihove osnovne karakteristike poput godine upisa, školskog uspjeha, bodova s državne mature i ukupnog broja bodova ostvarenih pri

upisu. Ključni stupac za ovu analizu je „STATUS (diplomirao, ispisan, aktivan studij)“, koji prikazuje trenutni ishod studiranja svakog pojedinog studenta.

Za izgradnju prediktivnog modela odabrane su tri značajke koje se smatraju najrelevantnijima za izgradnju modela logističke regresije: *prosjeck srednje škole*, *bodovi na državnoj maturi* te *ukupno bodova za upis na studij*. Kako bi se obradili podaci bez izmjene originalnog skupa, izrađena je kopija podataka korištenjem naredbe „data_opci.copy()“. Na taj način se osigurala sigurnost izvornih podataka u daljnjoj analizi.

```
✓ [35] print("Unikatne vrijednosti STATUS stupca:")
0s      print(data_opci["STATUS (diplomirao, ispisan, aktivan studij)"].unique())

Unikatne vrijednosti STATUS stupca:
['ispis' 'diploma' 'aktivan 2023/24']

✓ [36] #Odabir značajki i ciljne varijable
0s      features = ['prosjeck SŠ', 'Bodovi na državnoj maturi', 'Ukupno bodova za upis na studij']
      data_opci = data_opci.copy()
```

Slika 19 - pregled unikatnih vrijednosti i odabir značajki i ciljne varijable

Kod za mapiranje studenata:

```
#Mapiranje statusa: 0 = ispisan, 1 = diplomirao, 2 = aktivan studij

data_opci["status"] = data_opci["STATUS (diplomirao, ispisan, aktivan studij)"].map({
    "diplomirao": 1,
    "ispisan": 0
})

data_opci.loc[data_opci["STATUS (diplomirao, ispisan, aktivan studij)"].str.contains("aktivan", na=False), "status"] = 2
```

Ovaj dio koda pretvara tekstualne vrijednosti iz stupca o statusu studenata u numeričke oznake koje se koriste u daljnjem modeliranju. „Diplomirao“ se mapira kao 1, „ispisan“ kao 0, dok se svi statusi koji sadrže riječ „aktivan“ označavaju s 2. Na taj način omogućeno je jednostavnije grupiranje i analiza studenata prema ishodu studiranja.

2.5.2. Treniranje modela

```
✓ [27] # Odabir značajki za predviđanje  
0s features = [  
    "prosjeck SŠ", "Bodovi na državnoj maturi", "Ukupno bodova za upis na studij"  
]  
X = data_opci[features]  
y = data_opci["status"]
```

Slika 20 - prikaz koda odabira značajki za predviđanje

U ovom dijelu koda skup podataka se priprema za treniranje modela strojnog učenja. Prvotno se definiraju značajke (engl. *features*) koje se koriste za predviđanje statusa studenta, a to su: prosjek SŠ (prosječna ocjena iz srednje škole), bodovi na državnoj maturi (ukupni bodovi ostvareni na maturi), ukupno bodova za upis na studij (zbroy svih bodova temeljem kojih je student ostvario pravo upisa). Ove značajke čine ulaznu matricu X, dok se ciljna varijabla y postavlja kao stupac „status“, koji označava ishod studija (0 - ispisan, 1 - diplomirao).

```
✓ [28] # Podjela podataka na trening i testni skup  
0s X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

Slika 21 - prikaz koda podjele podataka na trening i test skup

Ovom linijom koda dijelimo podatke na skup za trening i testni skup. To je standardan korak u pripremi modela strojnog učenja.

- X i y: skup značajki i ciljne varijable koje se dijele,
- test_size=0.2: 20% podataka se koristi za testiranje, a 80% za treniranje modela,
- random_state=42: osigurava ponovljivost rezultata (fiksira slučajni odabir),
- stratify=y: osigurava da razmjerni udjeli klasa (ispisan/diplomirao) u testnom i trening skupu ostanu isti kao u originalnom skupu podataka.

Na taj način model trenira odnosno uči na većini podataka, dok se manji dio ostavlja po strani kako bi se testiralo koliko je model pouzdan u predviđanju na podacima sa kojima se do sada nije susreo.

Kod za odvajanje skupa za treniranje:

```
#Odvajanje skupa za treniranje (samo oni koji nisu aktivni)  
train_df = data_opci[data_opci["status"] != 2].dropna(subset=features +  
    ['status'])  
X_train = train_df[features]
```

```
y_train = train_df["status"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)
```

U ovom dijelom iz skupa podataka izdvajaju se samo studenti koji su ili diplomirali ili odustali, dok se aktivni studenti isključuju jer za njih još nije poznat ishod studiranja. Nakon toga podaci se dijele na trening i testni skup kako bi omjer klasa ostao uravnotežen u obje skupine. Nakon podjele podataka značajke se standardiziraju pomoću „*StandardScaler*“ kako bi se svi podaci doveli na istu skalu, a zatim se trenira model logističke regresije nad pripremljenim skupom podataka.

```
✓ [29] # Standardizacija značajki
0s scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Trening modela
models = {
    "Logistic Regression": LogisticRegression(),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42)
}
```

Slika 22 - prikaz koda standardizacije značajki i treninga modela

Kod predikcije za studente koji su još aktivni:

```
#Predikcija za studente koji su još aktivni
active_df = data_opci[data_opci["status"] == 2].dropna(subset=features)
X_active = scaler.transform(active_df[features])
predicted_probs = model.predict_proba(X_active)
```

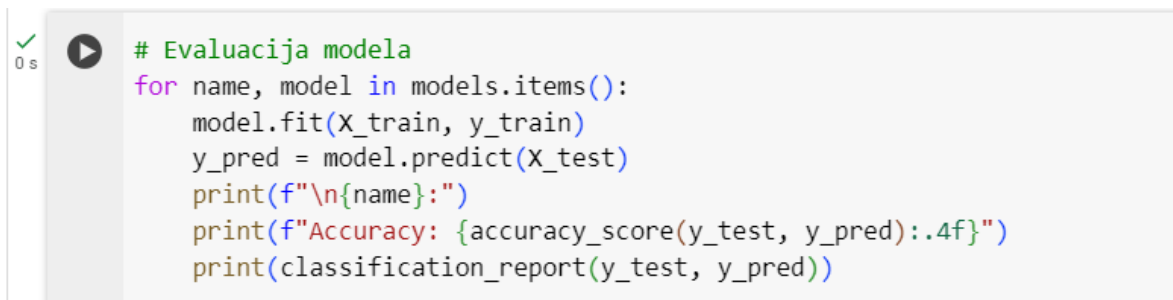
Ovaj kod izdvaja podatke za studente koji su još uvijek aktivni pri čemu se prethodno odabrane značajke standardiziraju istim „*scalerom*“ koji je korišten za treniranje modela. Zatim se pomoću metode „*predict_proba*“ izračunavaju vjerojatnosti hoće li pojedini student diplomirati ili odustati od studija.

U ovom dijelu prikazana su dva glavna koraka kod razvijanja modela strojnog učenja - standardizacija značajki i definiranje modela strojnog učenja. Ovdje koristimo *StandardScaler* pomoću kojega se standardiziraju značajke. Standardizirane značajke predstavljaju podatke koji su prilagođeni na način da imaju usporedivu skalu: srednju vrijednost (engl. *mean*) 0 i standardnu devijaciju 1. To je važno zato što mnogi drugi modeli (npr. logistička regresija) daje bolje rezultate kad su svi ulazni podaci na istoj skali. Time sprječavamo da dominantne značajke (npr. one s velikim brojčanim rasponom) imaju

veći utjecaj u odnosu na one sa manjim vrijednostima. Pomoću tri različita klasifikacijska modela strojnog učenja razvijati ćemo model predviđanja statusa studenta (diplomirao ili ispisan):

1. **Logistička regresija (engl. *Logistic Regression*)** - jednostavan i interpretabilan model koji procjenjuje vjerojatnost pripadnosti nekoj klasi.
2. **Stablo odluke (engl. *Decision Tree*)** - model koji gradi hijerarhiju odluka na temelju značajki.
3. **Slučajna šuma (engl. *Random Forest*)** - ansambl model temeljen na više stabala odluke i tako povećava točnost i robusnost, te smanjuje prenaučенost modela.

Svi modeli su spremni za treniranje, a cilj je usporediti njihovu preciznost i odabrati najpouzdaniji pristup za predviđanje akademskog ishoda.



```
0 s  # Evaluacija modela
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(f"\n{name}:")
    print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
    print(classification_report(y_test, y_pred))
```

Slika 23 - prikaz koda evaluacije modela

2.5.3. Evaluacija modela

U ovom dijelu rada provodi se evaluacija triju različitih modela klasifikacije, a to su logistička regresija, stablo odluke i slučajna šuma s ciljem predviđanja završetka studija (diplomirao ili ispisan). Značajke pomoću kojih se razvijaju modeli strojnog učenja su srednjoškolski prosjek i bodovi s državne mature. Svaki model se trenira na istom skupu podataka, a zatim testira na izdvojenom skupu podataka kako bi se provjerila njegova učinkovitost. Nakon treniranja modela predviđanja modela se uspoređuju sa stvarnim vrijednostima. Za svaki model izračunava se osnovna metrika točnosti (engl. *accuracy*), kao i detaljni klasifikacijski izvještaj koji uključuje preciznost, odziv i F1-mjeru za obje klase (diplomirao i ispisan). Analizom rezultata imamo uvid u to koliko je svaki model bio uspješan u razlikovanju između studenata koji su završili studij i onih koji su odustali, a najbolje rezultate pokazivala je logistička regresija koja je u nastavku detaljno opisana i korištena za predviđanje završetka studija.

Nakon što je model logističke regresije istreniran na podacima studenata koji su diplomirali ili odustali primijenjen je na skupinu studenata koji su još uvijek aktivni na studiju. Cilj ovog koraka bio je procijeniti vjerojatnost završetka studija za svakog aktivnog studenta na temelju njegovih početnih značajki tj. prosjeka ocjena iz srednje škole, rezultata na državnoj maturi i ukupnih bodova ostvarenih prilikom upisa na studij. Za svakog aktivnog studenta model je izračunao vjerojatnost diplomiranja, numeričku vrijednost između 0 i 1. Što je ta vrijednost bliže 1 to je veća vjerojatnost da će student završiti studij dok niže vrijednosti ukazuju na veći rizik od mogućeg odustajanja. Dodatno je uvedena i konačna klasifikacija (predikcija) gdje su svi studenti s vjerojatnošću većom ili jednakom 0.5 klasificirani kao oni koji će diplomirati (1), a ostali kao oni koji neće (0). Ovakvim pristupom prepoznavamo studente koji imaju manju vjerojatnost završetka studija, pa u vidu toga obrazovne institucije mogu intervenirati. Prikazana tablica sadrži JMBAG svakog studenta, njegove ulazne podatke (poput prosjeka i bodova), izračunatu vjerojatnost diplomiranja i konačnu klasifikaciju. Ako se želi prikazati više redaka tablice u okruženju poput „Google Colaba“, moguće je koristiti naredbu `pd.set_option("display.max_rows", None)` kako bi se uklonio ograničeni prikaz i omogućila potpuna analiza svih aktivnih studenata.

Konfuzijska matrica je koristan alat za procjenu klasifikacijskih modela. Pruža detaljan pregled u to kako model klasificira podatke, a ujedno identificira područja gdje model griješi. U kontekstu binarne klasifikacije, poput predviđanja hoće li student diplomirati ili odustati od studija matrica prikazuje četiri osnovne kategorije: istinito pozitivne (TP), istinito negativne (TN), lažno pozitivne (FP) i lažno negativne (FN) predikcije. Ako model točno predvidi da će student diplomirati to je istinito pozitivno (TP), ako pogrešno predvidi da će diplomirati, a student zapravo odustane to je lažno pozitivno (FP). Zahvaljujući ovakvoj strukturi (konfuzijskoj matrici) moguće je izračunati metrike poput preciznosti, odziva i F1-mjere. Ove metrike daju jasniju sliku o ponašanju modela. Posebice kad je jedna skupina broјčano veća od druge. U takvim situacijama točnost obično navede na krivi trag pa su preciznost, odziv i F1-mjera odlični pokazatelji koliko dobro model prepoznaje one koji diplomiraju i one koji odustaju.

ROC-AUC (engl. *Receiver Operating Characteristic - Area Under Curve*) analiza važan je korak u procjeni klasifikacijskog modela. Ova analiza pruža vizualni i kvantitativni uvid u to koliko je model dobar u razlikovanju dviju skupina - diplomiranih studenata i onih koji su se ispisali sa studija. U prikazanom kodu za svaki model izračunate su vjerojatnosti da

student pripada klasi „diplomirao“. Te vjerojatnosti su zatim iskorištene za izradu ROC krivulje.

```
from sklearn.metrics import roc_curve, auc
plt.figure(figsize=(7, 5))
for name, model in models.items():
    y_pred_prob = model.predict_proba(X_test)[: , 1]
    fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
    auc_score = auc(fpr, tpr)

    plt.plot(fpr, tpr, label=f"{name} (AUC = {auc_score:.2f})")

plt.plot([0, 1], [0, 1], linestyle="--", color="gray")
plt.xlabel("Lažno pozitivna stopa (FPR)")
plt.ylabel("Istinито pozitivna stopa (TPR)")
plt.title("ROC-AUC krivulja")
plt.legend()
plt.show()
```

2.6. Predviđanje prosjeka ocjena studenta

Nakon učitavanja podataka, naredbom „`print(data.columns)`“ ispisani su svi nazivi stupaca tablice „`data`“. Iz rezultata je vidljivo kako skup podataka sadrži osnovne podatke o studentima poput: *Datum završetka studija*, *JMBAG*, *Ak. god. završetka studija*, *Ak. god. zadnjeg upisa na studiju*, *Prosjek ocjena na studiju*, *Težinski prosjek ocjena na studiju*, *Ukupno ECTS bodova*, *Položeno ispita*, *Broj predmeta*, *Student studirao godina*. Stupci *Student studirao godina* i *Prosjek ocjena na studiju* predstavljaju ciljne varijable za klasifikaciju.

```
✓ 21 s [6] # Učitavanje data seta
      file_path_podaci = '/content/drive/MyDrive/Petra diplomski/završetak studija - prosjek, duljina studiranja.xlsx'
      data = pd.read_excel(file_path_podaci)

✓ 0 s [7] print(data.columns)

Index(['Datum završetka studija', 'JMBAG', 'Ak. god. završetka studija',
      'Ak. god. zadnjeg upisa na studiju', 'Prosjek ocjena na studiju',
      'Težinski prosjek ocjena na studiju', 'Ukupno ECTS bodova',
      'Položeno ispita', 'Broj predmeta', 'Student studirao godina'],
      dtype='object')
```

Slika 24 - prikaz podataka koji su korišteni za modele strojnog učenja za predviđanje prosjeka ocjena

2.6.1. Priprema podataka

U ovom izdvajaju se samo dvije ključne značajke iz izvornog skupa podataka: „*Student studirao godina*“ i „*Prosjek ocjena na studiju*“. Time se zadržavaju isključivo informacije relevantne za daljnju analizu, primjerice za istraživanje odnosa između trajanja studiranja i postignutog prosjeka. Ovakav pristup pojednostavljuje obradu podataka jer uklanja sve ostale varijable koje u ovom kontekstu nisu potrebne. Osim što čini analizu preglednijom, omogućuje i bržu i učinkovitiju izgradnju modela, posebno kada se radi o zadacima poput predikcije ili statističkog modeliranja.

```
# Odaberite relevantne kolone za analizu
data = data[['Student studirao godina', 'Prosjek ocjena na studiju']]
```

Slika 25 - prikaz pripreme podataka

2.6.2. Treniranje modela

U ovom radu primijenjena su tri različita regresijska modela s ciljem predviđanja prosjeka ocjena studenata na temelju broja godina provedenih na studiju: linearna regresija, slučajna šuma (engl. *Random Forest Regressor*) i stablo odluke (engl. *Decision Tree Regressor*).

Svi modeli trenirani su na istom skupu podataka imena „data“ koji je prethodno podijeljen na trening i test skup. 80% podataka koristi se za učenje modela, a preostalih 20% za testiranje.

Skup podataka odvaja se na ulazne značajke i ciljnu varijablu, što je osnovni korak u pripremi podataka za izgradnju modela strojnog učenja.

- X predstavlja ulazne podatke, odnosno skup značajki. U ovom slučaju stupac „*Student studirao godina*“ koja se koristi kao ulazna varijabla.
- y predstavlja ciljnu varijablu koju želimo predvidjeti, a to je „*Prosjek ocjena na studiju*“.

Na ovaj način jasno definiramo što će model koristiti kao ulaz, a što kao izlaz. Time pripremamo podatke za kasniju podjelu na trening i test skup, a ujedno i za treniranje regresijskog modela.

```
# Podijelite podatke na trening i test skupove
X = data[['Student studirao godina']]
y = data[['Prosjek ocjena na studiju']]
```

Slika 26 - priprema podataka za kasniju podjelu na trening i test skup

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Slika 27 - podjela podataka na trening i test skup

Ova linija koda dijeli podatke na trening i test skup, a to je ključni korak u pripremi modela strojnog učenja. Cilj je omogućiti modelu da uči iz jednog dijela podataka (trening skup), a zatim se njegova točnost provjerava na drugom dijelu (testni skup) koji nije bio uključen u fazu učenja.

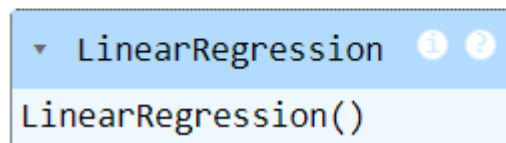
- X_{train} i y_{train} : predstavljaju ulazne značajke i ciljne vrijednosti koje se koriste za treniranje modela.
- X_{test} i y_{test} : predstavljaju podatke koji će poslužiti za evaluaciju modela, tj. provjeru njegove sposobnosti generalizacije na nove podatke.
- $test_size = 0.2$: označava da će 20% ukupnih podataka biti odvojeno za testiranje, dok će preostalih 80% biti korišteno za treniranje.

- `random_state = 42`: postavlja fiksno početno stanje generatora slučajnih brojeva kako bi podjela podataka bila uvijek ista pri svakom pokretanju, što omogućuje ponovljivost rezultata.

Ova podjela omogućuje pouzdano testiranje modela i smanjuje rizik od prenaučivosti modela, jer model ne „pamti“ testne podatke tijekom učenja.

Linearna regresija jedan je od najjednostavnijih i najčešće korištenih modela u analizi podataka. Polazi od pretpostavke da između ulazne varijable (broja godina studiranja) i ciljne vrijednosti (prosječna ocjena) postoji linearna veza. Prilikom treniranja model pronalazi optimalnu linearnu funkciju koja najbolje opisuje odnos između tih varijabli. Zbog svoje jednostavne strukture i interpretabilnosti, linearna regresija se često koristi kao početna točka prilikom usporedbe s kompleksnijim algoritmima strojnog učenja.

```
# Trenirajte linearni regresijski model
model = LinearRegression()
model.fit(X_train, y_train)
```



Slika 28 - prikaz koda treniranja linearnog regresijskog modela

Ovom linijom koda trenira se linearni regresijski modela, što znači da model na temelju dostupnih podataka uči odnose između ulazne varijable (godine studiranja) i ciljne vrijednosti (prosjek ocjena).

Prvo se kreira instanca modela pomoću linije `LinearRegression()`, a kreiranjem instance definirano je da će se za model koristiti linearna regresija.

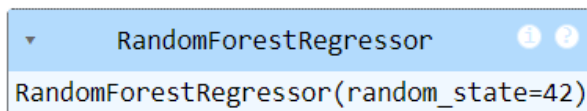
Model se zatim trenira korištenjem metode `.fit(X_train, y_train)`, pri čemu model „uči“ iz trening skupa:

- **X_train** predstavlja ulazne vrijednosti (godine studiranja),
- **y_train** su ciljne vrijednosti (prosječna ocjena).

Tijekom treniranja model pronalazi optimalne koeficijente (nagib i presjek pravca) kako bi najbolje objasnio odnos između ulazne značajke i izlazne varijable. Nakon treniranja model je spreman za predikciju na novim nepoznatim podacima.

Model slučajne šume (engl. *Random Forest Regressor*) koristi princip ansambla tj. kombinira rezultate većeg broja stabala odlučivanja (u ovom slučaju 100) kako bi dobio stabilnije i pouzdanije rezultate. Svako stablo u šumi trenira se na različitom slučajnom podskupu podataka i značajki, a konačna predikcija računa se kao prosjek rezultata svih stabala. Ova metoda često postiže bolje rezultate od pojedinačnih modela jer smanjuje varijancu i problem prenaučivosti modela.

```
# Trenirajte Random Forest Regressor model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

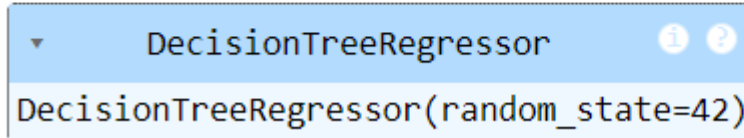


Slika 29 - prikaz koda treniranja modela slučajne šume

Model temeljen na kombinaciji više stabala odluke koristi se za rješavanje regresijskog problema. Kreira se pomoću linije koda *RandomForestRegressor(n_estimators=100, random_state=42)*, gdje parametar *n_estimators=100* označava da će model izgraditi 100 stabala, a *random_state=42* osigurava da rezultati budu konzistentni i ponovljivi pri svakom izvođenju. Model slučajne šume funkcionira tako da svako stablo u šumi uči na različitim slučajnim podskupovima podataka, a konačna predikcija dobiva se prosjekom rezultata svih stabala. Ova metoda smanjuje prenaučivost koja se javlja kod pojedinačnih stabala i poboljšava točnost modela jer koristi princip kolektivnog odlučivanja. Treniranje modela izvršava se pozivom metode *.fit(X_train, y_train)*, kojom model na temelju podataka uči povezanost između ulazne značajke i ciljne varijable. Model slučajne šume pokazao se vrlo uspješnim u regresijskim zadacima, posebno kada podaci ne prate jednostavnu linearnu strukturu ili sadrže složenije obrasce i varijacije.

Stablo odluke je model koji koristi hijerarhijsku strukturu za donošenje predikcija, razdvajajući podatke prema vrijednostima ulazne značajke kroz niz uvjetnih pravila. Za razliku od linearne regresije ne pretpostavlja linearan odnos između varijabli i to ga čini boljim za složenije obrasce. Međutim, takvi modeli mogu biti skloni prenaučivosti modela.

```
# Trenirajte Decision Tree Regressor model
model = DecisionTreeRegressor(random_state=42)
model.fit(X_train, y_train)
```



```
DecisionTreeRegressor(random_state=42)
```

Slika 30 - prikaz koda treniranja modela stabla odluke

U ovom koraku koristi se model stabla odluke (engl. *Decision Tree Regressor*) za predviđanje prosjeka ocjena studenata na temelju broja godina studiranja. Instancom modela *DecisionTreeRegressor(random_state=42)* postavlja se parametar *random_state* kako bi se osigurala ponovljivost rezultata. Svaki put kada se kod pokrene model će generirati istu strukturu stabla. Nakon definiranja modela poziva se metoda *.fit(X_train, y_train)* kojom model uči iz skupa podataka za trening. Stablo se grana prema uvjetima koji se temelje na vrijednostima ulazne značajke iz skupa podataka *X_train*, stvarajući pravila pomoću kojih pokušava što preciznije predvidjeti ciljnu varijablu (*y_train*). Za razliku od linearne regresije, stablo odluke ne pretpostavlja linearnost i može modelirati složenije odnose između varijabli. Model je fleksibilniji, ali potencijalno osjetljiviji na prenaučenosť ako se ne regulira dubina stabla ili broj listova.

```
# Predvidite rezultate na test skupu
y_pred = model.predict(X_test)
```

Slika 31 - prikaz koda za predviđanje rezultata na testnom skupu

Nakon što je model treniran na trening skupu, sljedeći korak je predikcija rezultata na testnom skupu. To se ostvaruje pomoću metode *predict(X_test)*, koja koristi prethodno naučene obrasce kako bi na temelju ulaznih podataka iz testnog skupa (*X_test*) generirala procijenjene vrijednosti ciljne varijable. Dobivene vrijednosti spremaju se u varijablu *y_pred*, koja sadrži predviđene prosjeke ocjena studenata. Ova faza je važna za evaluaciju modela jer se predikcije uspoređuju sa stvarnim vrijednostima (*y_test*) kako bi se izmjerila točnost i pouzdanost modela. Na taj način možemo objektivno procijeniti koliko dobro model generalizira na podatke koje nije vidio tijekom treniranja.

2.6.3. Evaluacija modela

Evaluacija regresijskih modela u ovom radu provodi se usporedbom predviđenih vrijednosti sa stvarnim vrijednostima na testnom skupu koristeći standardne metrike: MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error) i R^2 (koeficijent determinacije). Na taj način kvantitativno se procjenjuje točnost svakog modela i na taj način dobijemo objektivnu usporedbu njihove učinkovitosti i odabir modela koji najbolje generalizira na nove podatke.

2.6.4. Linearna regresija

Linearna regresija metoda je prediktivne analize koja se koristi za modeliranje odnosa između jedne zavisne i jedne ili više nezavisnih varijabli. Pretpostavlja se da je taj odnos linearan tj. da se može prikazati pravcem. Linearna regresija predstavlja osnovu mnogih drugih metoda predviđanja i analize podataka. Iako jako jednostavna, korisna je za razumijevanje osnovnog odnosa između dviju varijabli i često se koristi kao polazna točka za složenije modele.

U ovom slučaju linearna regresija se koristi za procjenu prosječne ocjene studenta na temelju broja godina koje je proveo na studiju. Iako se ovim pristupom može uočiti određeni trend (npr. studenti s duljim studiranjem mogu imati niže prosjeke), ovakav model je najprikladniji kada postoji linearan odnos između varijabli. Rezultati dobiveni ovom analizom mogu poslužiti kao preliminarni uvid u akademsku učinkovitost studenata u kontekstu trajanja studiranja.

Kod:

```
# Odaberite relevantne kolone za analizu
data = data[['Student studirao godina', 'Prosjek ocjena na studiju']]

# Podijelite podatke na trening i test skupove
X = data[['Student studirao godina']]
y = data['Prosjek ocjena na studiju']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Trenirajte linearni regresijski model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
# Predvidite rezultate na test skupu
y_pred = model.predict(X_test)
```

Model linearne regresije pokušava predvidjeti prosječnu ocjenu studenta na temelju broja godina provedenih na studiju. Prvo se iz izvornih podataka izdvajaju samo dvije relevantne varijable: "*Student studirao godina*" i "*Prosjek ocjena na studiju*". Zatim se ovi podaci dijele na trening i test skup, pri čemu se 80% koristi za treniranje modela, a 20% za testiranje. Korišten je model linearne regresije koji se trenira na skupu za učenje kako bi naučio odnos između broja godina studiranja i prosjeka ocjena. Na kraju model predviđa prosječne ocjene za podatke iz testnog skupa. Ovaj dio koda služi za vizualni prikaz rezultata linearnog regresijskog modela.

2.6.5. Slučajna šuma

Slučajna šuma (engl. *Random Forest*) je ansambl metoda strojnog učenja koja koristi više odlučnih stabala za regresiju ili klasifikaciju. Random Forest modeli stvaraju šumu odlučnih stabala od kojih svako stablo donosi svoju odluku, a konačna odluka (u slučaju klasifikacije) ili predviđanje (u slučaju regresije) temelji se na prosjeku ili većinskom glasanju rezultata svih stabala.

Algoritam slučajnih šuma (engl. *Random Forest*) temelji se na unaprijeđenoj varijanti metode bagging (engl. *Bootstrap Aggregating*) koja pomaže kod poboljšavanja stabilnosti i preciznosti u modelima. Ovaj algoritam je razvio Breiman u svom poznatom radu *Random Forests* (Breiman, 2001.), gdje je kombinirao različite postojeće koncepte iz statistike koji uključuju elemente slučajnosti. Osim bagginga, važan doprinos predstavlja i tehnika nasumičnog odabira skupa značajki (atributa) iz cijelog skupa prilikom izgradnje svakog stabla što je prethodno predložio Dietterich (Dietterich, 2000.). Obje metode uvode kontrolirani šum u podatke i model čime se poboljšava robusnost i preciznost algoritma prilikom predviđanja na novim, nepoznatim podacima.

Kod:

```
# Odaberite relevantne kolone za analizu
data = data[['Student studirao godina', 'Prosjek ocjena na studiju']]
# Podijelite podatke na trening i test skupove
X = data[['Student studirao godina']]
y = data['Prosjek ocjena na studiju']
```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Trenirajte Random Forest Regressor model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
# Predvidite rezultate na test skupu
y_pred = model.predict(X_test)

```

Metoda Random Forest Regressor za predviđanje prosjeka ocjena studenata na temelju broja godina studiranja funkcionira na način da se prvo odabiru dvije relevantne varijable: "*Student studirao godina*" kao nezavisna varijabla (X) i "*Prosjek ocjena na studiju*" kao zavisna varijabla (y). Podaci se zatim dijele u trening i test skup pomoću funkcije *train_test_split*, pri čemu se 20% podataka koristi za testiranje, a ostalo za treniranje.

Zatim se koristi *RandomForestRegressor* model koji gradi više stabala odluke i kombinira njihove rezultate kako bi poboljšao točnost predikcije i smanjio prenaučenosť. Model se trenira na trening podacima pomoću naredbe *.fit()*, a zatim se koristi naredba *.predict()* za izračun predviđenih vrijednosti na test skupu.

2.6.6. Stablo odluke

Stablo odluke (engl. *Decision Tree*) je algoritam strojnog učenja koji se koristi za klasifikaciju i regresiju. Modelira odluke i njihove moguće posljedice pomoću strukture stabla. Svaki unutarnji čvor postavlja pitanje na temelju određene značajke, grane prikazuju moguće odgovore na to pitanje, a listovi na kraju stabla predstavljaju konačne odluke ili predviđene vrijednosti.

Kod:

```

# Odaberite relevantne kolone za analizu
data = data[['Student studirao godina', 'Prosjek ocjena na studiju']]
# Podijelite podatke na trening i test skupove
X = data[['Student studirao godina']]
y = data['Prosjek ocjena na studiju']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Trenirajte Decision Tree Regressor model
model = DecisionTreeRegressor(random_state=42)
model.fit(X_train, y_train)
# Predvidite rezultate na test skupu
y_pred = model.predict(X_test)

```

Ovaj kod prikazuje kako se koristi regresijsko stablo odluke (engl. *Decision Tree Regressor*) za predviđanje prosjeka ocjena studenata na temelju broja godina studiranja.

Prvo se iz skupa podataka odabiru dvije relevantne kolone: 'Student studirao godina' i 'Prosjek ocjena na studiju'. Nakon toga se podaci dijele u ulazne značajke (X) i ciljnu varijablu (y) gdje X sadrži samo broj godina studiranja, a y predstavlja prosjek ocjena. Slijedi podjela podataka na trening skup i testni skup, pri čemu se 20% podataka koristi za testiranje, a 80% za treniranje modela. Argument *random_state = 42* osigurava ponovljivost rezultata.

Zatim se instancira *DecisionTreeRegressor* model koji pokušava naučiti pravila na temelju podataka tako da dijeli vrijednosti na granama stabla. Nakon što se model nauči na trening skupu, koristi se za predikciju rezultata na testnom skupu pomoću naredbe *model.predict(X_test)*. Ovim se procjenjuju predviđene vrijednosti koje će se kasnije usporediti s pravim ocjenama radi evaluacije točnosti modela.

3. Rezultati i diskusija

Rezultati su pokazali da logistička i linearna regresija postižu najbolje rezultate u vidu točnosti predviđanja. Modeli stabla odluke i slučajne šume pružaju bolji uvid u to koji čimbenici najviše utječu na uspjeh studenata. U nastavku su detaljno prikazani koraci pripreme i obrade podataka, izgradnje modela i evaluacije.

Vizualni prikaz grafa (Slika 12 - Raspršeni dijagram - prikaz odnosa između prosječne ocjene studenta na studiju i duljine trajanja njegovog studiranja) pokazuje blagu negativnu povezanost između promatranih varijabli. Studenti s višim prosjekom ocjena uglavnom završavaju studij u kraćem vremenskom periodu. Iako podaci pokazuju određeni stupanj raspršenosti što znači da postoje individualne razlike, opći uzorak sugerira da uspješniji studenti (prema ocjenama) u prosjeku kraće studiraju. S druge strane, kod studenata s nižim prosjekom češće je duže trajanje pohađanja studija. Takvi rezultati upućuju na ponavljanje godina, sporije polaganje ispita ili druge poteškoće tijekom studija. Važno je napomenuti da se unatoč općem trendu javljaju i iznimke gdje pojedini studenti s visokim prosjekom završavaju studij u produljenom roku. To može biti posljedica vanjskih okolnosti, a i da je model gledao podatke na isti način za diplomski i prijediplomski studij.

Na temelju dobivenih rezultata ROC-AUC analize možemo zaključiti kako je logistička regresija najučinkovitiji model u ovom istraživanju s AUC vrijednošću od 0.73. Ova vrijednost ukazuje na to da model pouzdano razlikuje studente koji će diplomirati od onih koji će odustati, što je znatno bolje od slučajnog pogađanja. Iako nije savršen, ovakav rezultat pokazuje potencijal modela kojeg se može dodatno unaprijediti. Njegova preciznost dodatno bi se mogla poboljšati uključivanjem dodatnih značajki ili boljim balansiranjem podataka. S druge strane, model temeljen na stablu odluke pokazao je vrlo loš rezultat gdje je AUC vrijednost iznosila 0.45. Takav rezultat znači da model predviđa ishode lošije nego što bi to učinio slučajnim odabirom. Takav rezultat može ukazivati na prenaučenos modela, nedovoljnu količinu podataka ili nepovoljnu strukturu značajki. Model slučajne šume iako poznat kao pouzdan algoritam u mnogim klasifikacijskim zadacima, u ovom slučaju nije dao korisne rezultate. AUC vrijednost od 0.50 pokazuje da predikcije modela nisu bolje od nasumičnog pogađanja. To može značiti da model nije

uspio prepoznati relevantne obrasce u podacima ili da odabrane značajke nisu pružile dovoljno korisnih informacija za ovu vrstu klasifikacije.

3.1. Rezultati predviđanja završetka studija

U ovom istraživanju primijenjena je logistička regresija kao jedan od najčešće korištenih modela za binarnu klasifikaciju. Model je treniran kako bi predvidio hoće li student završiti studij ili odustati (Ujkani, Minkovska, & Stoyanova, 2022).

3.1.1. Model logističke regresije

Iz rezultata je vidljivo da studenti s višim ukupnim brojem bodova i višim srednjoškolskim prosjekom uglavnom imaju i veću vjerojatnost završetka studija. Student koji je pri upisu ostvario 1769 bodova ima više od 75% šanse da će diplomirati, dok studenti s manje od 500 bodova često imaju vjerojatnost nižu od 40%. Model to interpretira kao povećan rizik od odustajanja. Ovi rezultati mogu pomoći obrazovnim institucijama da na vrijeme uoče studente kojima bi mogla trebati dodatna podrška. Pomoću ovog modela predviđanja i dobivenih rezultata institucije mogu intervenirati kako bi povećali šanse za uspješno završavanje studija.

Za svaki model izračunava se osnovna metrika točnosti (engl. *accuracy*), kao i detaljni klasifikacijski izvještaj koji uključuje preciznost, odziv i F1-mjeru za obje klase (diplomirao i ispisan). Analizom rezultata imamo uvid u to koliko je svaki model bio uspješan u razlikovanju između studenata koji su završili studij i onih koji su odustali.

```

Logistic Regression:
Accuracy: 0.6613
      precision    recall  f1-score   support

    0.0         0.66     0.68     0.67         31
    1.0         0.67     0.65     0.66         31

   accuracy          0.66         62
  macro avg          0.66     0.66     0.66         62
 weighted avg          0.66     0.66     0.66         62

Decision Tree:
Accuracy: 0.4355
      precision    recall  f1-score   support

    0.0         0.44     0.48     0.46         31
    1.0         0.43     0.39     0.41         31

   accuracy          0.44         62
  macro avg          0.43     0.44     0.43         62
 weighted avg          0.43     0.44     0.43         62

Random Forest:
Accuracy: 0.5645
      precision    recall  f1-score   support

    0.0         0.56     0.58     0.57         31
    1.0         0.57     0.55     0.56         31

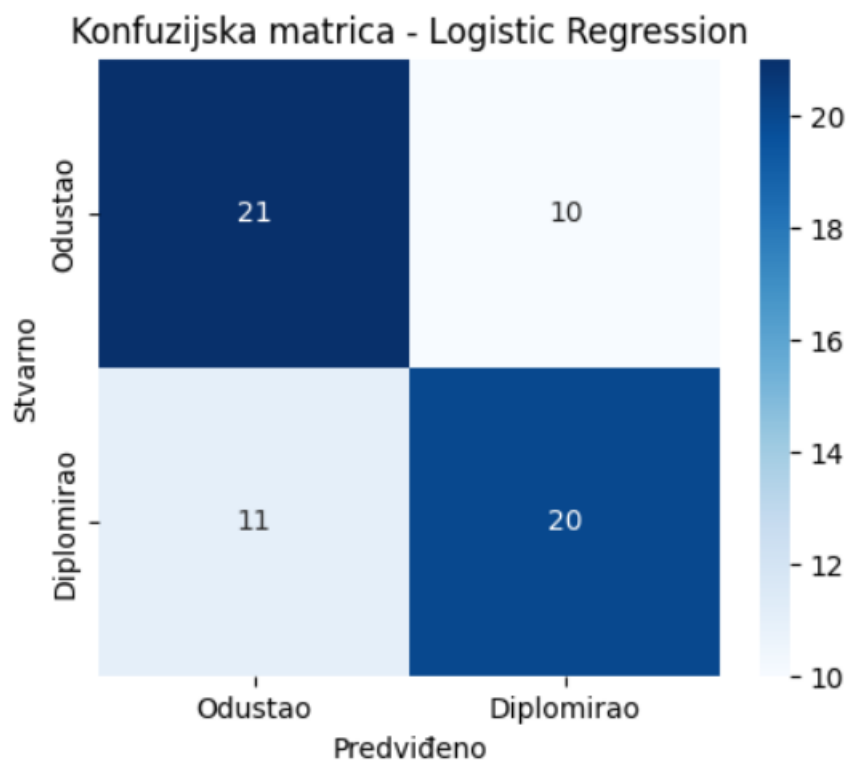
   accuracy          0.56         62
  macro avg          0.56     0.56     0.56         62
 weighted avg          0.56     0.56     0.56         62

```

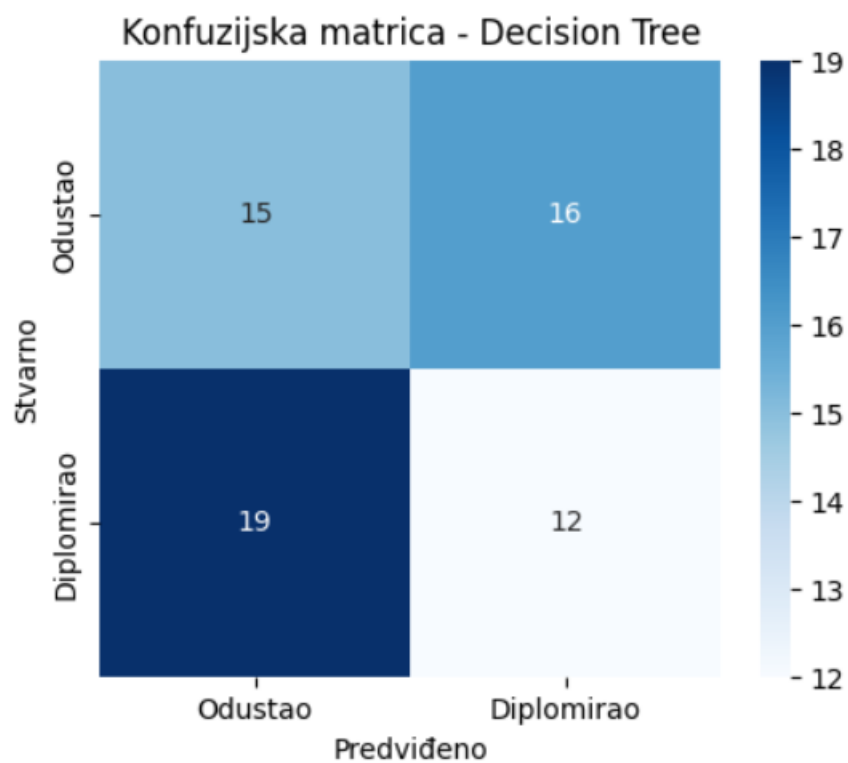
Slika 32 - rezultati evaluacije modela

Rezultati evaluacije pokazuju koliko je model klasifikacije bio uspješan u predviđanju statusa studenata na temelju ulaznih značajki. Najuspješniji rezultate evaluacije daje model logističke regresije, s točnošću od 66.13%. Model je jednako dobro prepoznao studente koji su diplomirali i one koji su se ispicali. Preciznost i odziv za obje klase (0 = ispiscani, 1 = diplomirali) bili su uravnoteženi za obje skupine. F1-mjera dodatno potvrđuje da model jednako dobro prepoznaje obje skupine studenata. Stablo odluke ostvarilo je znatno lošiju točnost (43.55%) pa možemo uvidjeti kako model nije pronašao dovoljno pouzdane obrasce u podacima. Zabrinjavajuće je što su vrijednosti odziva i F1-mjere niske, što znači da model često pogrešno predviđa status studenata. Slučajna šuma kao napredniji ansambl model postiže bolji rezultat od običnog stabla odluke i točnost iznosi 56.45% . To je bolji rezultat od stabla odluke, ali i dalje lošiji rezultat u usporedbi sa logističkom regresijom.

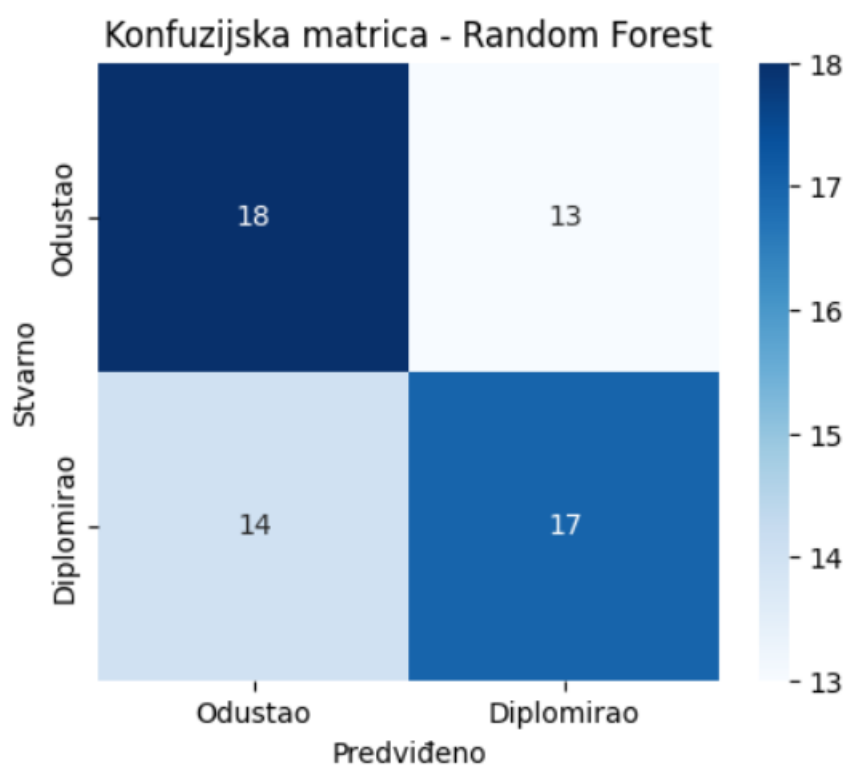
Iako su složeniji modeli tehnički robusniji, u ovom slučaju nisu doveli do boljih rezultata. Razlog tome vjerojatno je ograničena količina podataka i broja značajki. Logistička regresija pokazala se kao najstabilnije i najpouzdanije rješenje za ovaj klasifikacijski problem.



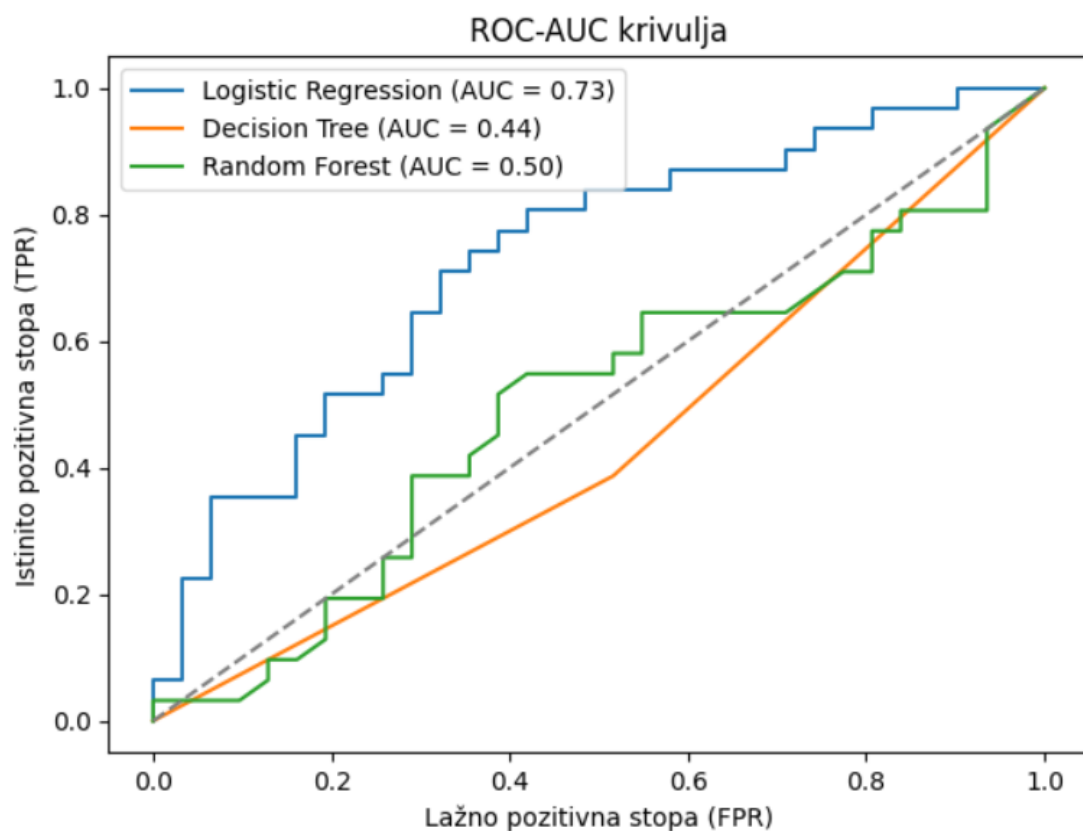
Slika 33 - rezultati prikazani konfuzijskom matricom - Logistička regresija



Slika 34 - rezultati prikazani konfuzijskom matricom - Stablo odluke



Slika 35 - rezultati prikazani konfuzijskom matricom - Slučajna šuma



Slika 36 - ROC-AUC krivulja

Krivulje prikazuju odnos između istinito pozitivne stope (TPR) i lažno pozitivne stope (FPR) pri različitim pragovima odlučivanja. Površina ispod krivulje (AUC) prikazuje sažeto koliko je model uspješan u razlikovanju klasa. Što je AUC vrijednost bliža 1 to je model bolji u razlikovanju klasa. Na grafu je prikazana i dijagonalna linija koja predstavlja nasumičnu klasifikaciju. Ako se ROC krivulja modela nalazi iznad te linije, to znači da model ima stvarnu prediktivnu vrijednost i bolje razlikuje klase nego da samo nagađa.

```
✓ [45] #Prikaz rezultata
0 s pd.set_option("display.max_columns", None)
print(active_df[["JMBAG"] + features + ["vjerojatnost_diplomiranja", "predikcija"]])
```

	JMBAG	prosjeak SŠ	Bodovi na državnoj maturi \	
215	0177057322	3.82	358.13	
219	0177057294	3.58	348.45	
244	0177059145	3.92	215.31	
245	0177059311	4.38	211.56	
246	0177054502	4.55	351.47	
..	...	...	...	
395	0177065032	4.31	392.09	
396	0177065048	3.87	278.34	
397	0177065053	3.73	507.91	
398	0177066136	4.35	498.75	
399	0177065069	2.91	324.16	

	Ukupno bodova za upis na studij	vjerojatnost_diplomiranja	predikcija
215	625.5	0.589765	1.0
219	599.1	0.505771	1.0
244	489.7	0.365819	0.0
245	518.2	0.485269	0.0
246	670.0	0.757221	1.0
..	...	...	...
395	693.8	0.761134	1.0
396	549.2	0.461629	0.0
397	1769.0	0.757044	1.0
398	803.3	0.877619	1.0
399	527.9	0.287788	0.0

Slika 37 - prikaz rezultata logističke regresije

Nakon što je model logističke regresije istreniran na podacima studenata koji su diplomirali ili odustali primijenjen je na skupinu studenata koji su još uvijek aktivni na studiju. Cilj ovog koraka bio je procijeniti vjerojatnost završetka studija za svakog aktivnog studenta na temelju njegovih početnih značajki tj. prosjeka ocjena iz srednje škole, rezultata na državnoj maturi i ukupnih bodova ostvarenih prilikom upisa na studij. Za svakog aktivnog studenta model je izračunao vjerojatnost diplomiranja, numeričku vrijednost između 0 i 1. Što je ta vrijednost bliže 1 to je veća vjerojatnost da će student završiti studij dok niže vrijednosti ukazuju na veći rizik od mogućeg odustajanja. Dodatno je uvedena i konačna klasifikacija (predikcija) gdje su svi studenti s vjerojatnošću većom ili jednakom 0.5 klasificirani kao oni koji će diplomirati (1), a ostali kao oni koji neće (0). Ovakvim pristupom prepoznavamo studente koji imaju manju vjerojatnost završetka studija, pa u vidu toga obrazovne institucije mogu intervenirati. Prikazana tablica sadrži JMBAG svakog studenta, njegove ulazne podatke (poput prosjeka i bodova), izračunatu vjerojatnost diplomiranja i konačnu klasifikaciju. Ako se želi prikazati više redaka tablice u okruženju poput „Google Colaba“, moguće je koristiti naredbu „`pd.set_option("display.max_rows", None)`“ kako bi se uklonio ograničeni prikaz i omogućila potpuna analiza svih aktivnih studenata.

3.2. Rezultati predviđanja prosjeka ocjena studenata

3.2.1. Model linearne regresije

Dobiveni rezultati prikazuju usporedbu stvarnih i predviđenih prosjeka ocjena studenata u odnosu na broj godina studiranja, pri čemu su korišteni podaci iz testnog skupa modela linearne regresije. Svaki redak sadrži: broj godina koje je student proveo na studiju, stvarni prosjek ocjena koji je student ostvario te prosjek koji je predvidio model.

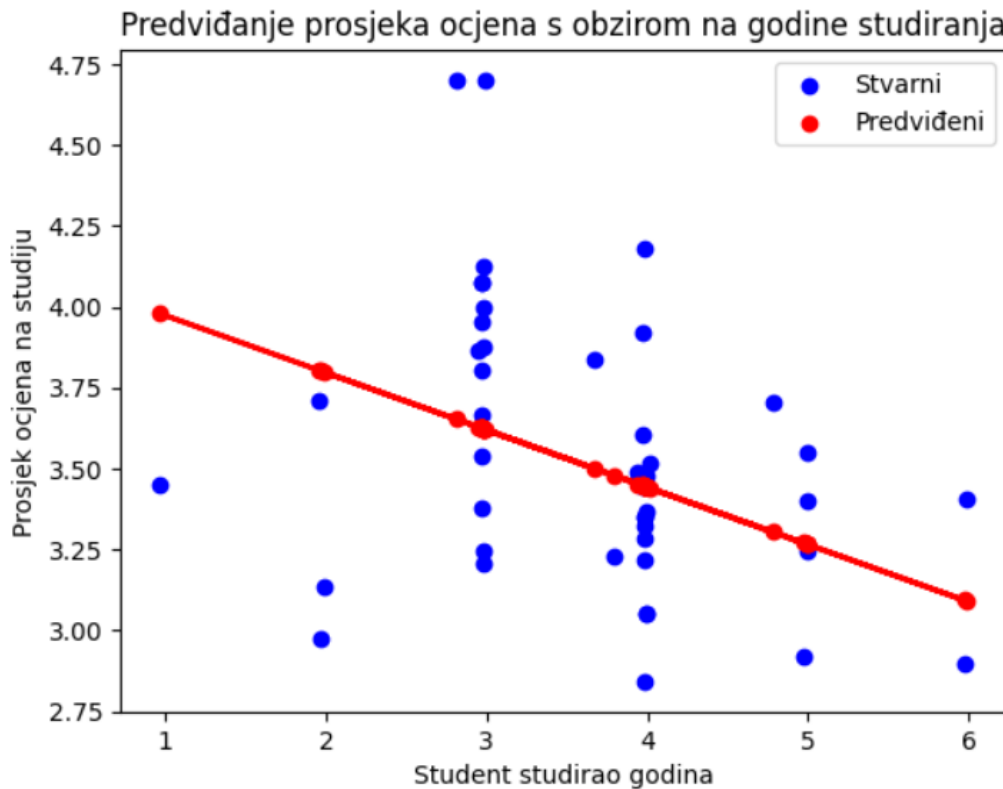
Analizom tablice uočavamo da su predviđene ocjene uglavnom smještene oko srednje vrijednosti (između 3.2 - 3.6). To je uobičajeno kod jednostavnih linearnih modela kada ne postoji jaka linearna povezanost između varijabli. U slučajevima kada student ima visok stvarni prosjek (npr. 4.7), model često daje nižu procjenu od stvarne vrijednosti. S druge strane kada je stvarni prosjek niži model ga nekada procijeni višim nego što zapravo jest. Ovo pokazuje da model nije savršeno „naučio“ varijacije u podacima i da bi za bolje rezultate trebalo uključiti dodatne značajke i podatke (npr. srednjoškolski uspjeh, bodove na maturi itd.).

Općenito, model uspijeva donekle predvidjeti opći smjer relacija među podacima. Studenti koji studiraju kraće često imaju više prosjeke, dok oni koji studiraju duže obično imaju niže ocjene. Ta veza nije toliko jaka, a to uočavamo po tome što su razlike u predikcijama između studenata s različitim trajanjem studija vrlo male. Linearna regresija u ovom slučaju može poslužiti kao gruba orijentacija, ali za preciznije predviđanje prosjeka ocjena trebali bismo koristiti složenije modele i u obzir uzeti više informacija.

U ovom slučaju, budući da je koeficijent negativan, pravac ima silazni nagib. To znači da krivulja počinje u najvišoj točki pri ishodištu (kada je broj godina studiranja manji), a zatim postupno opada kako broj godina studiranja raste. Drugim riječima, model predviđa da studenti koji studiraju kraće u prosjeku imaju više ocjene, dok se s povećanjem broja godina studiranja prosječna ocjena smanjuje. Ovaj trend može ukazivati na to da dulje studiranje može biti povezano s određenim poteškoćama koje utječu i na akademsku uspješnost, odnosno možda dolazi do zasićenja ako su studenti upisali i diplomski studij, a možda im je bilo teže savladati gradivo pa su rješavali kolegije u konačnici sa lošijim ocjenama.

Pomoću biblioteke *matplotlib*, na *scatter* dijagramu prikazuju se stvarne vrijednosti prosjeka ocjena (*y_test*) u plavoj boji, kao i predviđene vrijednosti (*y_pred*) u crvenoj boji,

koje je model izračunao na temelju broja godina studiranja (X_{test}). Crvena linija koja se povlači kroz predviđene točke predstavlja regresijsku liniju - najbolju pravocrtnu aproksimaciju odnosa između godina studiranja i prosjeka ocjena. Osim vizualne usporedbe stvarnih i predviđenih podataka, ova slika daje dobar uvid u to koliko dobro model predviđa stvarne ocjene te postoji li linearna povezanost između promatranih varijabli.



Slika 38 - prikaz rezultata modela linearna regresije

Koeficijenti modela

```

✓ [54] # Prikaz koeficijenata modela
0 s coefficients = model.coef_
    intercept = model.intercept_

    print(f'Koeficijent: {coefficients[0]}')
    print(f'Presjek: {intercept}')

```

⇒ Koeficijent: -0.17698581585724182
Presjek: 4.149835770733942

Slika 39 - prikaz koeficijenata modela linearne regresije

Nakon što je model linearne regresije treniran na podacima, ispisani su njegovi ključni parametri: koeficijent i presjek. Koeficijent predstavlja nagib regresijske linije te pokazuje kolika je prosječna promjena u prosjeku ocjena na studiju za svaku dodatnu godinu studiranja.

U ovom slučaju koeficijent iznosi otprilike -0.18, što upućuje na negativnu povezanost - dulje trajanje studija povezano je s nešto nižim prosjekom ocjena. Presjek (engl. *intercept*), koji iznosi 4.15, predstavlja predviđeni prosjek ocjena kada bi broj godina studiranja bio nula. Iako takva situacija nije stvarna, presjek je važan dio jednadžbe koja definira regresijsku liniju i omogućuje izračun predviđenih vrijednosti za bilo koju duljinu studiranja. Na temelju ovih parametara model formira pravac koji najbolje opisuje vezu između duljine studiranja i prosjeka ocjena.

Izračunavanje metrika

```
✓ 0 s [55] # Izračunavanje metrika
      mae = mean_absolute_error(y_test, y_pred)
      mse = mean_squared_error(y_test, y_pred)
      rmse = np.sqrt(mse)
      r2 = r2_score(y_test, y_pred)

      print(f'Mean Absolute Error (MAE): {mae}')
      print(f'Mean Squared Error (MSE): {mse}')
      print(f'Root Mean Squared Error (RMSE): {rmse}')
      print(f'R-squared (R²): {r2}')
```

➞ Mean Absolute Error (MAE): 0.31447792852992446
Mean Squared Error (MSE): 0.16451779018780652
Root Mean Squared Error (RMSE): 0.40560792668266055
R-squared (R²): 0.07916701399860793

Slika 40 - prikaz metrika modela linearne regresije

Izračunate su četiri ključne metrike koje služe za evaluaciju kvalitete linearnog regresijskog modela.

- MAE (engl. *Mean Absolute Error*) pokazuje prosječnu apsolutnu razliku između stvarnih i predviđenih vrijednosti. U ovom slučaju, prosječna pogreška iznosi oko 0.31 što znači da model u prosjeku prikazuje krive podatke za nešto više od tri desetinke ocjene.

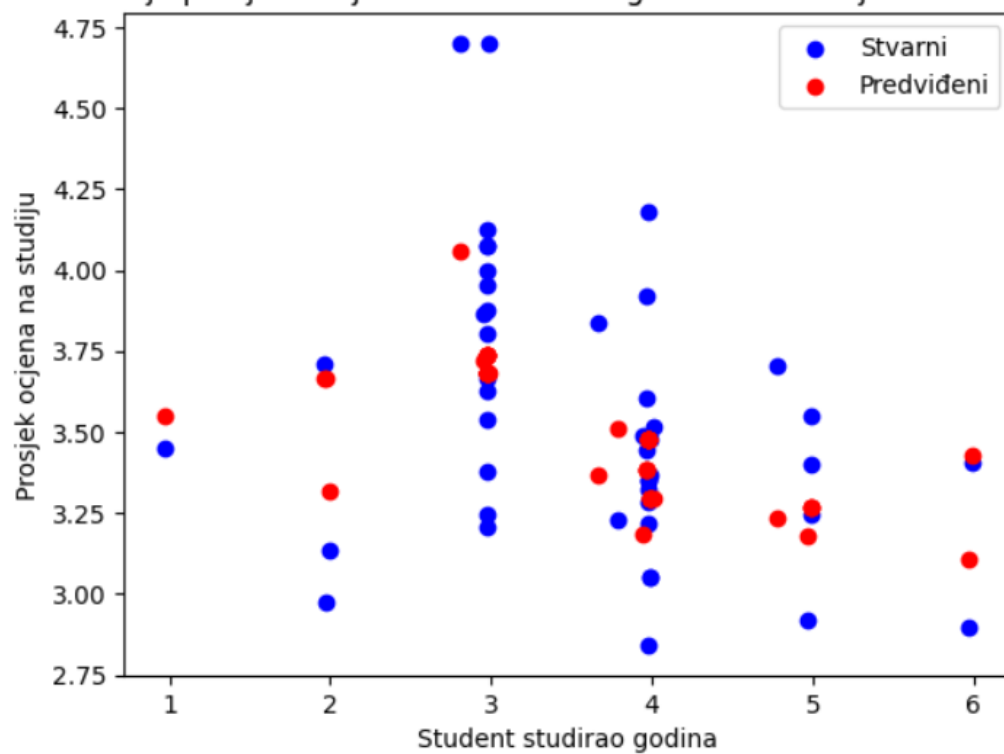
- MSE (engl. *Mean Squared Error*) iznosi 0.16 i predstavlja prosječnu kvadratnu pogrešku modela. MSE dodatno naglašava odstupanja većih razlika jer kvadrira razliku između stvarnih i predviđenih vrijednosti.
- RMSE (engl. *Root Mean Squared Error*), kao korijen MSE-a, vraća pogrešku u izvornim jedinicama ciljne varijable i iznosi oko 0.41. Ova vrijednost je korisna jer je interpretabilna tj. model u prosjeku odstupa za 0.41 u predviđanju prosjeka ocjena.
- R-squared (R^2) vrijednost od 0.079 znači da model objašnjava samo oko 7.9% varijance ciljne varijable. To je prilično nizak rezultat, što znači da broj godina studiranja sam po sebi ne daje dovoljno informacija da bi se na temelju njega precizno predvidio prosjek ocjena.

Zajedno ove metrike pokazuju da model ima ograničenu sposobnost predviđanja i da bi za bolje rezultate bilo potrebno uključiti dodatne značajke koje više utječu na prosjek ocjena.

3.2.2. Model slučajne šume

Na temelju prikazanih rezultata može se vidjeti usporedba stvarnog i predviđenog prosjeka ocjena studenata koristeći metodu Random Forest Regressor, uzimajući u obzir broj godina koje su proveli na studiju kao jedinu ulaznu varijablu. Model za svaki testni primjer generira predviđeni prosjek ocjena koji se zatim uspoređuje sa stvarnim prosjekom iz podataka. Student koji je studirao 2.97 godina ima stvarni prosjek 4.077, dok model predviđa 3.68. Ovo pokazuje razmjerno dobro predviđanje koje je blizu stvarnim vrijednostima, ali i odstupanja u slučajevima ekstremnih vrijednosti (npr. kod studenata s vrlo visokim prosjecima).

Predviđanje prosjeka ocjena s obzirom na godine studiranja - Random Forest



Slika 41 - prikaz rezultata modela slučajne šume

Izračunavanje metrika

```
✓ 0 s ▶ # Izračunavanje metrika
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f'Mean Absolute Error (MAE): {mae}')
print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R-squared (R2): {r2}')
```

➡ Mean Absolute Error (MAE): 0.2840105377806881
Mean Squared Error (MSE): 0.12673194685230948
Root Mean Squared Error (RMSE): 0.3559943073313244
R-squared (R²): 0.29066056072985613

Slika 42 - prikaz metrika modela slučajne šume

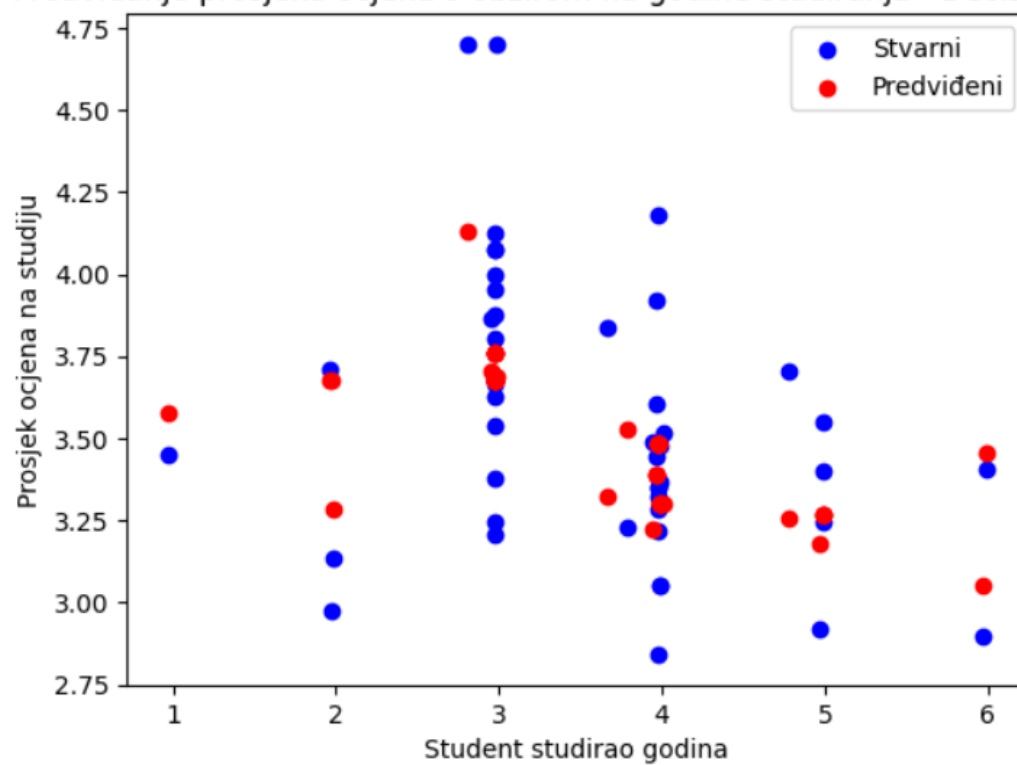
Rezultati evaluacije modela pokazuju sljedeće: Mean Absolute Error (MAE) iznosi 0.28, što znači da model u prosjeku griješi za otprilike 0.28 ocjena prilikom predviđanja prosječnog uspjeha studenata. Mean Squared Error (MSE) ima vrijednost 0.13. Root Mean Squared Error (RMSE) iznosi 0.36, što pokazuje da model u prosjeku prilično dobro predviđa rezultate, ali ipak osjetljivije reagira na veća odstupanja. Koeficijent determinacije (R^2) iznosi 0.29, što znači da model objašnjava oko 29 % varijacije u prosjeku ocjena na temelju broja godina studiranja. Iako to nije visok postotak, rezultat ipak ukazuje na postojanje određene povezanosti između duljine studija i akademskog uspjeha.

3.2.3. Model stabla odluke

Model predviđa prosjek ocjena studenata na temelju broja godina koje su proveli na studiju. Iz prikazanih rezultata vidljivo je da se predviđeni prosjeci najčešće kreću između ocjena 3.2 i 3.7, dok stvarni prosjeci imaju veću varijaciju od 2.9 do 4.7.

Student s 2.81 godina studiranja ima stvarni prosjek 4.700, dok model predviđa 4.132. Kod većine vrijednosti oko 2.97 ili 3.98 godina model često daje istu ili vrlo sličnu predikciju, što pokazuje da model nije jako osjetljiv na sitne razlike u broju godina studiranja. To može značiti da se model oslanja na obrasce iz podataka koje je naučio, ali nije dovoljno fleksibilan da uhvati specifične razlike među pojedincima.

Predviđanje prosjeka ocjena s obzirom na godine studiranja - Decision Tree



Slika 43 - prikaz rezultata modela stabla odluke

Izračunavanje metrika

```
✓ 0 s # Izračunavanje metrika
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f'Mean Absolute Error (MAE): {mae}')
print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R-squared (R²): {r2}')
```

➞ Mean Absolute Error (MAE): 0.2810940912391894
Mean Squared Error (MSE): 0.124863617946791
Root Mean Squared Error (RMSE): 0.3533604646063153
R-squared (R²): 0.3011179032636778

Slika 44 - prikaz metrika modela stabla odluke

Dobiveni rezultati evaluacije modela koji predviđa prosjek ocjena studenata na temelju broja godina studiranja pokazuju umjerenu razinu točnosti. Srednja apsolutna pogreška (MAE) iznosi približno 0.28 što znači da model u prosjeku griješi za manje od jedne trećine ocjene. Root Mean Squared Error (RMSE) koji dodatno naglašava veće pogreške, iznosi oko 0.35. To ukazuje na relativno mala odstupanja između stvarnih i predviđenih vrijednosti. Vrijednost R-kvadrata (R^2) iznosi 0.30, što znači da model uspijeva objasniti oko 30% varijance u podacima.

Zaključak

U ovom radu provedena su dva različita modela prediktivne analize s ciljem dubljeg razumijevanja studentskog uspjeha. Prvi dio analize bio je usmjeren na predviđanje vjerojatnosti završetka ili odustajanja od fakulteta koristeći klasifikacijske modele. Najuspješniji model bio je logistička regresija, koja se pokazala najpouzdanijom u razlikovanju između studenata koji su diplomirali i onih koji su odustali. Metrika točnosti je bila zadovoljavajuća, dok je metrika AUC vrijednosti iznosila oko 0.73. Ova analiza pokazala je da je moguće već na temelju ranih ulaznih podataka (poput bodova iz srednje škole i sa državne mature) unaprijed identificirati studente koji su u riziku od odustajanja. Time se otvara mogućnost da im se na vrijeme ponudi podrška i poduzmu konkretne mjere kako bi završili studij.

Drugi dio rada je temelji se na predviđanju prosjeka ocjena studenata putem regresijskih modela, konkretno linearnom regresijom, slučajnom šumom i stablima odluke. Iako su sva tri modela pokazala određenu razinu točnosti, model slučajne šume postigao je nešto bolje metrike s R^2 vrijednosti od oko 0.30, što znači da model objašnjava oko 30% varijacija u ocjenama na temelju trajanja studiranja. Ovi rezultati ukazuju da sama duljina studiranja nije dovoljan prediktor za ocjene, već bi predikcija bila znatno učinkovitija uz uključivanje dodatnih značajki.

Analizom je utvrđeno da se modeli strojnog učenja mogu učinkovito koristiti za rano prepoznavanje rizika odustajanja od studija, dok je za preciznije predviđanje akademskih postignuća poželjno koristiti više značajki i podataka. Takvi pristupi mogu pomoći obrazovnim institucijama u pravovremenoj intervenciji i podršci studentima.

Literatura

- [1] STALLINGS, W. *Data and Computer Networks*. London: John Wiley, 2006.
- [2] ABBOTT, D. *Applied predictive analytics: Principles and techniques for the professional data analyst*. John Wiley & Sons, 2014.
- [3] KELLEHER, D. J., NAMEE, M. B., D'ARCY, A. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. Cambridge, Massachusetts London, England: The MIT Press, 2015.
- [4] KUMAR, V., & GARG, M. L. *Predictive analytics: a review of trends and techniques*. *International Journal of Computer Applications*, 182(1), 31-37., 2018.
- [5] AGRESTI, A. *Categorical Data Analysis (Second Edition)*. New York: John Wiley, 2007.
- [6] BREIMAN, L., *Random Forests*. *Machine Learning*, Volume 45, Issue 1, pp. 5–32, 2001.
- [7] DIETTERICH, T.G., *An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization*. *Machine Learning*, Volume 40, Issue 2, pp. 139–157, 2000.
- [8] T.M. MITCHELL, *Machine learning*, McGraw-Hill Science, 1997.
- [9] PRABA R., DARSHAN G., ROSHANRAJ K.T., SURYA PRAKASH B., *Study On Machine Learning Algorithms*, 2021.
- [10] ALPAYDIN, E. *Introduction to Machine Learning*. Cambridge: The MIT Press, 2004.
- [11] GERON, A. *Hands-on machine learning with with Sickit-Learn and TensorFlow*. Sebastopol: O'Reilly Media, 2017.
- [12] MCCULLAGH P., & NELDER J.A., *Generalized Linear Models*, London, New York, Chapman and Hall, 1982.
- [13] HUSSEIN ALMUALLIM, SHIGEO KANEDA, YASUHIRO AKIBA, *Development and applications of decision trees*, 2002.
- [14] A. L. SAMUEL, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.
- [15] UJKANI, B., MINKOVSKA, D., & STOYANOVA, L. Y., *Application of Logistic Regression Technique for Predicting Student Dropout*. In 2022 International Scientific Conference Electronics – ET2022 (pp. 1–4)., 2022.
- [16] TORRES-CRUZ, F. *Prediction of University Dropouts Through Random Forest-Based Models*. *Journal of Applied and Physical Education Research*, 2024.
- [17] VILLAR, A., & ANDRADE, C. R. V. *Supervised Machine Learning Algorithms for Predicting Student Dropout and Academic Success: A Comparative Study*. *Discover Artificial Intelligence*, 4(2), 2024.

- [18] BARADWAJ, B. K., & PAL, S. *Mining educational data to analyze students' performance*. International Journal of Computer Applications, 61(13), 1–5, 2012.
- [19] MURAINA, I. O., AIYEGBUSI, E. A., & ABAM, S. O. *Decision tree algorithm use in predicting students' academic performance in advanced programming course*. International Journal of Higher Education Pedagogies, 3(4), 13–23, 2022.
- [20] G. SHMUELI AND O. R. KOPPIUS, “*Predictive analytics in information systems research*,” MIS Quarterly, vol. 35, no. 3, pp. 553–572, 2011.
- [21] R. S. J. BAKER AND P. S. INVENTADO, “*Educational Data Mining and Learning Analytics*,” in Learning Analytics: From Research to Practice, J. A. Larusson and B. White, Eds., New York, NY, USA: Springer, pp. 61–75, 2014
- [22] S. B. KOTSIANTIS, C. PIERRAKEAS, AND P. PINTELAS, “*Predicting students' performance in distance learning using machine learning techniques*,” Applied Artificial Intelligence, vol. 18, no. 5, pp. 411–426, 2004
- [23] KOTSIANTIS, S. B., *Use of machine learning techniques for educational purposes: A decision support system for forecasting students' grades*. Artificial Intelligence Review, 37(4), 331–344, 2012
- [24] C. ROMERO AND S. VENTURA, “*Educational data mining: A review of the state of the art*,” IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 40, no. 6, pp. 601–618, 2010.
- [25] G. SHMUELI AND O. R. KOPPIUS, “*Predictive analytics in information systems research*,” MIS Quarterly, vol. 35, no. 3, pp. 553–572, 2011.
- [26] MOYANO, L., MOYA, M., & HERRERA, F., „*Student dropout prediction in higher education: A data-driven systematic review*“. Frontiers in Education, 8, Article 1244686, 2023
- [27] SPECTOR, J. M., IFENTHALER, D., SAMPSON, D. G., & YANG, L., „*Personalized learning paths: Leveraging data analytics for tailored education*“, ResearchGate [Preprint], 2024
- [28] PAPAMITSIOU, Z., & ECONOMIDES, A. A., „*Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence*“, Educational Technology & Society, 17(4), 49–64, 2014

Sažetak

RAZVOJ MODELA STROJNOG UČENJA ZA PREDVIĐANJE AKADEMSKOG USPJEHA STUDENTA

Rad razvija modele strojnog učenja u svrhu predviđanja akademskog uspjeha studenata. Istraživanje se temelji na anonimiziranim podacima iz ISVU-a sustava. Cilj ovog istraživanja je izgradnja modela strojnog učenja koji mogu predvidjeti hoće li student diplomirati ili odustati od studija, kao i modela za predviđanje prosjeka ocjena na temelju broja godina studiranja. Korišteni su algoritmi poput logističke i linearne regresije, stabla odluke i slučajne šume. Modeli su evaluirani pomoću metrika AUC, MAE, MSE i R^2 . Analizom dobivenih rezultata prikazano je da se primjenom odabranih značajki može postići pouzdan uvid u budući akademski uspjeh studenata.

Summary

DEVELOPING MACHINE LEARNING MODELS FOR PREDICTING ACADEMIC PERFORMANCE

The paper develops machine learning models for the purpose of predicting students' academic success. The research is based on anonymized data from the ISVU system. The goal of this research is to build machine learning models that can predict whether a student will graduate or drop out, as well as models for predicting grade point averages based on the number of years of study. Algorithms such as logistic and linear regression, decision trees, and random forests were used. The models were evaluated using the metrics AUC, MAE, MSE, and R^2 . The analysis showed that selected features can provide insight into students' future academic success.

Skraćenice

AUC	accuracy	točnost
MAE	Mean Absolute Error	srednja apsolutna pogreška
MSE	Mean Squared Error	srednja kvadratna pogreška
R ²	R-squared	R-kvadrat
RMSE	Root Mean Squared Error	korijen srednje kvadratne pogreške
SVM	Support vector machines	Stroj vektora odluke
TP	True Positives	istinite pozitivne
TN	True Negatives	istinite negativne
FP	False Positives	lažno pozitivne
FN	False Negatives	lažno negativne
ROC-AUC Receiver Operating Characteristic - Area Under Curve ROC-AUC krivulja		
TPR	True Positive Rate	istinito pozitivne stope
FPR	False Positive Rate	lažno pozitivne stope
GLM	Generalized Linear Models	generalizirani linearni modeli
EDM	Educational Data Mining	rudarenje obrazovnih podataka
RL	Reinforcement Learning	algoritam učenja potkrjepljivanjem