Učinkovitost tehnika predobrade podataka u izgradnji modela predviđanja

Višković, Kristina

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni prijediplomski studij računarstva

Završni rad

UČINKOVITOST TEHNIKA PREDOBRADE PODATAKA U IZGRADNJI MODELA PREDVIĐANJA

Rijeka, rujan 2023.

Kristina Višković

0069085751

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni prijediplomski studij računarstva

Završni rad

UČINKOVITOST TEHNIKA PREDOBRADE PODATAKA U IZGRADNJI MODELA PREDVIĐANJA

Mentor: doc. dr. sc. Goran Mauša

Rijeka, rujan 2023.

Kristina Višković
0069085751

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 12. ožujka 2021.

Zavod: **Zavod za računarstvo**
Predmet: **Uvod u objektno orijentirano programiranje**
Grana: **2.09.06 programsko inženjerstvo**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Kristina Višković (0069085751)**
Studij: Preddiplomski sveučilišni studij računarstva

Zadatak: **Učinkovitost tehnika predobrade podataka u izgradnji modela predviđanja / Efficiency of data-preprocessing techniques in building a predictive model**

Opis zadatka:

Proučiti postupke predobrade podataka u dubinskoj analizi podataka, s naglaskom na tehnike čišćenja podataka kojima se uklanjaju instance i značajke. Odabrane tehnike primijeniti nad karakterističnim skupom podataka u izgradnji modela predviđanja te izmjeriti utrošak energije, memorije i vremena za scenarije izvornih i očišćenih podataka. Usporediti dobivene vrijednosti i dovesti ih u relaciju s performansama modela predviđanja.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Kristina Višković

Zadatak uručen pristupniku: 15. ožujka 2021.

Mentor:

Doc. Goran Mauša, dipl. ing.

Predsjednik povjerenstva za završni ispit:

Izv. prof. dr. sc. Kristijan Lenac

POTPISANA IZJAVA O SAMOSTALNOJ IZRADI RADA

Ja, Kristina Višković, izjavljujem da sam završni rad izradila samostalno, uz mentorstvo doc. dr. sc. Gorana Mauše te korištenjem navedene literature.

Rijeka, rujan 2023. _____

Kristina Višković

Zahvaljujem mentoru, doc. dr. sc. Goranu Mauši, na pruženoj pomoći i iskazanom strpljenju i razumijevanju, te svojim bližnjima na iznimnoj podršci.

Sadržaj

1 UVOD **1**

2 PROCES DUBINSKE ANALIZE PODATAKA **3**

2.1 Održiva dubinska analiza podataka 6

3 TEHNIKE PREDOBRADE PODATAKA **10**

3.1 Nedostajuće vrijednosti 10

3.2 Odabir instanci 11

3.3 Odabir značajki 13

4 METODOLOGIJA **18**

4.1 Korištene tehnologije 18

4.2 Skupovi podataka 19

4.3 Primjena tehnika predobrade podataka u izgradnji modela predviđanja 20

5 REZULTATI **25**

6 ZAKLJUČAK **29**

LITERATURA **31**

Popis slika

2.1 *Osnovne faze procesa dubinske analize podataka prema procesnom modelu CRISP-DM [3]* 4
2.2 *Pregled zadataka procesnog modela CRISP-DM te njihovi izlazi [2]* 6
2.3 *Osnovne faze procesa dubinske analize podataka prema procesnom modelu CRISP-DM s dodatkom načela održive analize podataka [4]* 7
3.1 *Ilustrativan primjer relevantnih, redundantnih i beznačajnih značajki [6]* 13
5.1 *Odnos značajki (os x) i rezultata funkcije bodovanja (os y)* 28

Popis tablica

5.1 *Rezultati mjerenja koristeći skup podataka „Dry Bean Dataset" i klasifikator stabla odluke* 26
5.2 *Rezultati mjerenja koristeći skup podataka „Dry Bean Dataset" i klasifikator nasumične šume* 27
5.3 *Rezultati mjerenja koristeći skup podataka „Polish companies bankruptcy" i klasifikator podizanja gradijenata temeljen na histogramu* 28

1 UVOD

U današnje vrijeme velike količine podataka dostupnije su no ikada prije. Transformaciju neobrađenih podataka u korisno znanje moguće je postići korištenjem tehnika dubinske analize podataka. Dubinska analiza podataka (*engl. data mining*) je postupak kojim se otkrivaju obrazaci te ostale korisne informacije iz velikih skupova podataka [1]. Kao glavnu svrhu tehnika dubinske analize podataka moguće je izdvojiti jednu od sljedećih:

- opisivanje skupa podataka ili
- predviđanje izlaza korištenjem algoritama strojnog učenja.

Za dubinsku analizu podataka koristi se i termin otkrivanje znanja iz baza podataka (*engl. Knowledge Discovery in Databases, KDD*).

Jednostavnije prikupljanje i skladištenje podataka nerijetko dovodi i do nagomilavanja nepotrebnih i nekvalitetnih podataka. U trenutku je lako zanemariti utjecaj koji će iskorištavanje naizgled neiscrpnog izvora resursa imati na okoliš. Nažalost, već svjedočimo negativnim posljedicama neodgovornog korištenja resursa, ne samo u računalnom svijetu, već i u svim sferama života. Ovaj završni rad stoga proučava i ispituje neke od načina na koji proces dubinske analize podatka može biti održiviji. Tema rada je učinkovitost tehnika predobrade podataka u izgradnji modela predviđanja. Također, završni rad izrađen je u okviru ERASMUS+ projekta „Promoting Sustainability as a Fundamental Driver in Software development Training and Education” s oznakom 2020-1-PT01-KA203-078646.

Kroz sljedeća poglavlja objašnjen je proces dubinske analize podataka te su navedeni neki od načina koji pojedine faze tog procesa mogu učiniti održivijima. Zatim su proučeni postupci predobrade podataka, a naglasak je na tehnikama čiš-

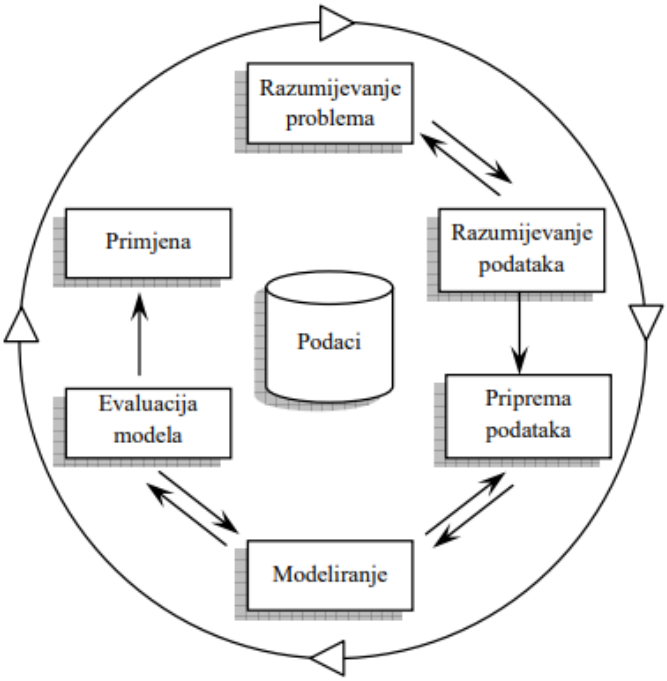
ćenja podataka kojima se uklanjaju instance i značajke. Za kraj su odabrane tehnike čišćenja podataka primijenjene nad karakterističnim skupovima podataka u izgradnji modela predviđanja. Korišteni su algoritmi strojnog učenja stablo odluke i nasumična šuma te algoritam podizanja gradijenata temeljen na histogramu (*engl. Histogram-based Gradient Boosting*). Za scenarije izvornih i očišćenih podataka izmjeren je utrošak energije, memorije i vremena te su dobivene vrijednosti dovedene u relaciju s performansama modela predviđanja.

2 PROCES DUBINSKE ANALIZE PODATAKA

U ovom će radu proces dubinske analize podataka biti opisan procesnim modelom CRISP-DM (CRoss Industry Standard Process for Data Mining), neovisnom o industrijskom sektoru i korištenoj tehnologiji. Spomenuti procesni model koncipiran je kao hijerarhijski procesni model kojeg čine četiri razine apstrakcije, a koje su redom, od općih do specifičnih: faze, generički zadaci, specijalizirani zadaci i instance procesa. Generički zadaci namijenjeni su da budu dovoljno općeniti da pokrivaju sve moguće situacije dubinske analize podataka te su dizajnirani da budu, koliko je moguće, kompletni i stabilni. Ovdje kompletno podrazumijeva da pokrivaju cijeli proces dubinske analize podataka kao i sve moguće primjene dubinske analize podataka, dok stabilno podrazumijeva da model idealno vrijedi i za sve nepredviđene razvoje, kao što su na primjer nove tehnike modeliranja. Na koji način provoditi radnje generičkih zadataka u specifičnim situacijama opisano je specijaliziranim zadacima. Razina instance procesa, koja je ujedno i posljednja, predstavlja zapis radnji, odluka i rezultata procesa dubinske analize podataka čime bilježi stvarne ishode procesa [2].

Procesni model CRISP-DM sastoji se od ukupno šest faza koje su redom: razumijevanje problema, razumijevanje podataka, priprema podataka, modeliranje, evaluacija odnosno vrednovanje modela te primjena. Za lakšu vizualizaciju procesnog modela CRISP-DM, priložena je slika 2.1.

Prema procesnom modelu CRISP-DM, dubinska analiza podataka započinje **fazom poslovnog razumijevanja** koja je usmjerena na razumijevanje ciljeva i zahtjeva projekta iz poslovne perspektive. Ova faza sastoji se od četiri zadataka: određivanja poslovnih ciljeva, procjene situacije, određivanja ciljeva dubinske analize podataka te izrade plana projekta.



Slika 2.1: *Osnovne faze procesa dubinske analize podataka prema procesnom modelu CRISP-DM [3]*

Faza razumijevanja podataka također se sastoji od četiri zadataka. U ovom su slučaju to: prikupljanje početnih podataka, opisivanje podataka, istraživanje podataka i provjera kvalitete podataka.

Iduća je **faza pripreme podataka**. Ova faza započinje tako što se iz skupa svih prikupljenih podataka izdvaja podskup podataka koji će služiti kao osnova za daljnje analize. Odabir podataka vrši se na temelju kriterija kvalitete i tehničkih ograničenja. Zatim slijede čišćenje, konstruiranje i integriranje podataka koji će biti dodatno obrađeni u kasnijem poglavljima. Konačni zadatak faze pripreme podataka, formatiranje podataka, tehničke je prirode te se odnosi na „prilagođavanje zapisa pripremljenih podataka uvjetima koje diktira korištena tehnika modeliranja podataka" [3].

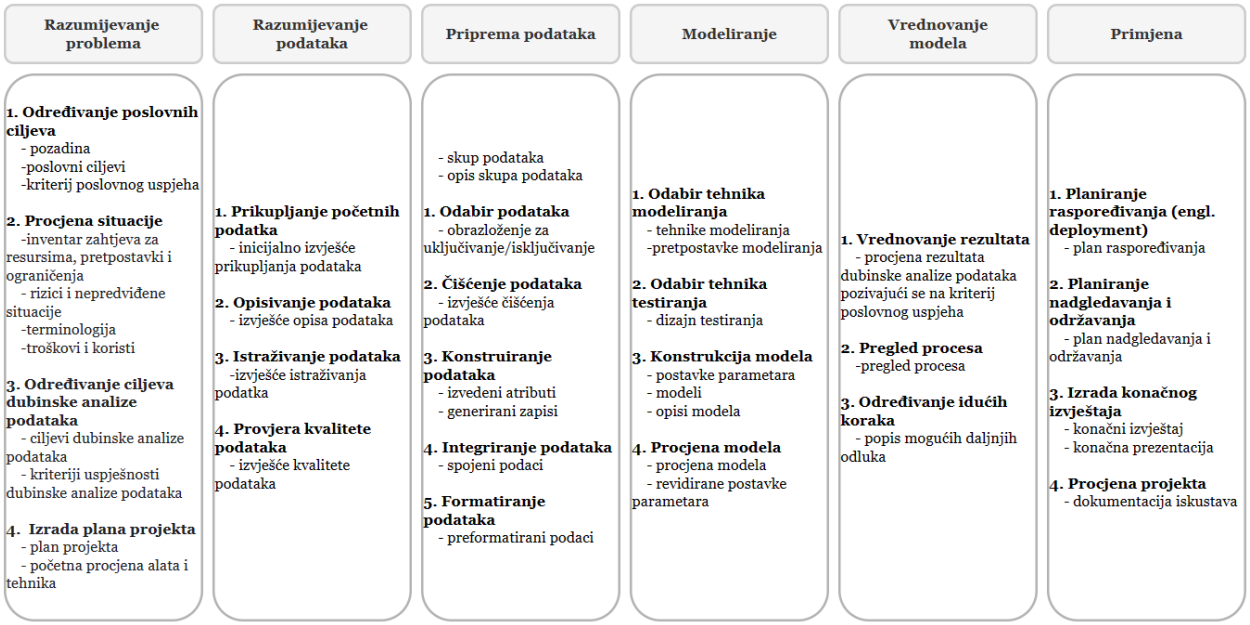
Faza modeliranja, uključuje četiri zadataka: odabir tehnika modeliranja, oda-

bir procedure testiranja, konstrukciju i procjenu modela. U ovoj se fazi odabiru i primjenjuju razne tehnike modeliranja čiji se parametri postavljaju na optimalne vrijednosti [2]. Ključan je dio procesa dubinske analize podataka u kojoj se „korištenjem postupaka strojnog učenja obavlja istinska analiza podataka i pronalaze skrivene pravilnosti koje u njima postoje" [3]. Dodatno, kao što je vidljivo i na slici 2.1, faze pripreme podataka i modeliranja mogu se iterativno ponavljati pošto rezultati faze modeliranja ponekad stvaraju potrebu za daljnjim zahvatima nad podacima.

Zatim slijedi **faza vrednovanja modela**. Za vrijeme vrednovanja rezultata važno je utvrditi zadovoljavaju li konačni modeli kriterije uspješnosti. Između ostalog, ocjenjuju se pouzdanost modela na testnom uzorku. Modeli se ocjenjuju iz aspekta dubinske analize podataka i aspekta domene problema. Vrednovanje modela u odnosu na domenu problema generalno vrši stručnjak iz te domene [3]. Ova faza uključuje i osvrt na cijeli proces kako bi se otkrilo jesu li svi zadaci pravilno izvršeni. Moguće je otkrivanje nedostataka ili mogućnosti poboljšanja modela, te se ovisno o rezultatima osvrta utvrđuje spremnost prelaska u konačnu fazu. Primjerice, postoje li razlozi za dodatnu korekciju modela, dolazi do potrebe odabira novih kombinacija parametara tehnike modeliranja s kojima se potiče nova iteracija konstrukcije modela [3].

Konačna faza dubinske analize podataka, **faza primjene**, podrazumijeva primjenu stečenog znanja u poslovnom okruženju. Tijekom ove faze razvija se i dokumentira plan primjene modela. Uz to, razvija se i plan za praćenje i održavanje s ciljem izbjegavanja problema prilikom operativne faze. Također, stvara se konačni izvještaj te radi pregled čitavog projekta.

Za kraj, slici 2.2 dan je pregled zadataka procesnog modela CRISP-DM te njihovi izlazi.

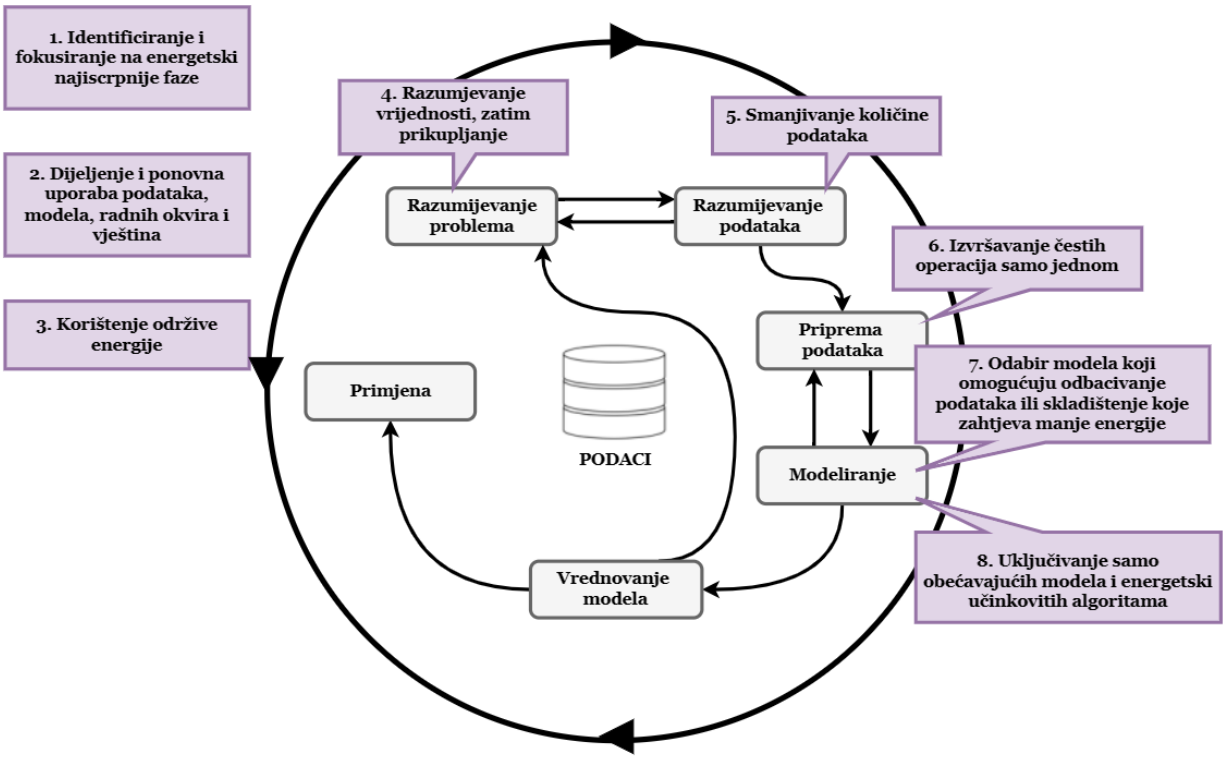


Slika 2.2: *Pregled zadataka procesnog modela CRISP-DM te njihovi izlazi [2]*

2.1 Održiva dubinska analiza podataka

Schneider, Basalla i Seidel u svom radu „Principles of Green Data Mining" [4] predstavljaju načela održive dubinske analize podataka vezana uz ključne faze procesa dubinske analize podatka, a utemeljena na procesnom modelu CRISP-DM te relevantnim metodama dubinske analize podataka i zelenom IT-u. Na slici 2.3 ponovno su prikazane faze dubinske analize podataka prema procesnom modelu CRISP-DM, no ovog puta uz dodatak spomenutih načela održive dubinske analize podataka.

Prvo načelo izneseno u spomenutom radu odnosi se na **identificiranje i fokusiranje na energetski najiscrpnije faze procesa**. Ovo načelo naravno podrazumijeva dobro razumijevanje problema te je čitajući rad moguće uvidjeti da veliku ulogu u smanjenju utroška energije igraju upravo početne faze dubinske analize podataka,



Slika 2.3: *Osnovne faze procesa dubinske analize podataka prema procesnom modelu CRISP-DM s dodatkom načela održive analize podataka [4]*

odnosno faza razumijevanja problema i faza razumijevanja podataka.

Drugo po redu načelo odnosi se na **dijeljenje i ponovnu uporabu već dostupnih podataka, modela, radnih okvira i vještina**. Navodi se i potencijalna energetska učinkovitost uključivanja vanjskih suradnika, to jest outsourcing-a, koje može biti poželjno ukoliko vanjski suradnik pristupa izdvajanju traženih informacija na energetski učinkovitiji način, bilo zbog prethodnih iskustava i znanja ili korištenih tehnologija. [4]. Tu se spominje i tehnika prijenosa učenja (*engl. transfer learning technique*) čija je ideja korištenje znanja iz postojećih modela treniranih za specifične zadatke na drugim zadacima čime je znanje iz jedne domene prenosivo na neku drugu domenu.

Iduće načelo potiče **korištenje održive energije** za vrijeme procesa dubinske ana-

lize podataka te čak predlaže planiranje izvedbe procesa u svrhu maksimiziranja korištenja obnovljive energije.

Fokus četvrtog načela je na vrijednosti samih podataka. Kako svi podaci ne sadrže jednaku vrijednost što samim time čini određene podatke korisnijima, potiče se **određivanje i prikupljanje/skladištenje samo korisnih podataka**. Veća količina podataka ne garantira kvalitetu, a zbog skladištenja i obrade može zahtijevati dodatno korištenje korisnih resursa, uključujući i energiju. Između ostalog navodi se i da se, povećanjem obujma podataka u bazi podataka, povećava i vrijeme upita u bazu podataka (*engl. query times*). Podaci slabije kvalitete nerijetko u kasnijim fazama zahtijevaju i čišćenje čime se dodatno troši energija.

Moglo bi se reći da se sljedeće načelo nastavlja na četvrto time što se bavi temom **smanjivanja količine podataka**. Jedan od načina da se sačuvaju samo najrelevantniji podaci je primjena aktivnog učenja kroz koje se postepeno izdvajaju i dodaju najrelevantniji podaci iz ukupnog skupa podataka. Proces učenja može biti prekinut ukoliko dodavanje novih podataka znatno ne poboljšava model, a neiskorišteni se podaci se u tom slučaju jednostavno odbacuju. Osim aktivnog učenja spomenuta je i tehnika uzorkovanja te se kao najjednostavniji primjer izdvaja nasumično uzorkovanje. Dodatno, preporučuje se i uklanjanje nepotrebnih i „bučnih" atributa (*engl. noisy attributes*) te odabir i ekstrakcija značajki. Tehnike odabira značajki dodatno su razrađene u kasnijim poglavljima. Ponekad se u svrhu smanjenja dimenzionalnosti određeni kompleksniji tipovi podataka transformiraju u manje kompleksnije tipove koji se potencijalno jednostavnije i brže obrađuju. Dodatno, bitno je naglasiti da prikaz podataka može utjecati na energetsku učinkovitost podataka. Za kraj, veliku važnost nosi i točna specifikacija zahtjeva atributa.

Šesto načelo naglašava da bi se **česte operacije trebale izvršavati samo jednom**.

Iduće se načelo odnosi na **odabir modela koji omogućuju odbacivanje podataka ili načina skladištenja s niskom energetskom potrošnjom**. Autori ističu negativnu korelaciju potrošnje energije i pristupačnosti pohranjenim podacima, to jest, napominje se da lakša pristupačnost podacima zahtjeva veću potrošnju energije pri održavanju podataka. Način skladištenja bi stoga idealno trebao odgovarati razini potražnje podataka te se čak s vremenom može razmišljati i o brisanju ili komprimiranju podataka, te alternativno radu na sažetom skupu podataka. Naravno, brisanje podataka nije uvijek moguća opcija pošto neki modeli zahtijevaju cijeli skup podataka (koji uključuje i prijašnje podatke) pri ponovnom treniranju podataka unatoč prilagođavanju novim podacima. Kao održiva opcija ističe se minimiziranje pristupa podacima uz premještanje podataka na energetski prihvatljivije medije. Dodatna je opcija zamjena postojećeg modela modelom koji ne zahtjeva skladištenje starijih podataka.

Osmo, a ujedno i posljednje načelo, predlaže da se **u svrhu odabira modela i algoritama koji će biti korišteni u procesu dubinske analize podataka u obzir uzima i njihova energetska učinkovitost**. U ovom koraku proučavanje literature i dostupnih prijašnjih istraživanja može biti od velike koristi te je naravno održivije od iscrpnog samostalnog eksperimentiranja.

Cilj ovog završnog rada je ispitati učinkovitost tehnika predobrade podataka u izgradnji modela predviđanja te je naglasak na tehnikama čišćenja podataka kojima se uklanjaju instance i značajke. Posljedično tome, u ovome će radu jednim dijelom biti ispitano i peto načelo ekološke dubinske analize podataka, odnosno načelo koje se odnosi na smanjivanje količine podataka.

3 TEHNIKE PREDOBRADE PODATAKA

U ovom je radu naglasak na tehnikama čišćenja podataka kojima se uklanjaju instance i značajke stoga će u sljedećim potpoglavljima te tehnike biti detaljnije obrađene.

3.1 Nedostajuće vrijednosti

Čest slučaj u predobradi podataka predstavljaju nedostajući podaci. Jedan od razloga može biti što svi atributi ne moraju biti primjenjivi za sve slučajeve. Međutim, ponekad vrijednosti mogu nedostajati pošto su jednostavno zaboravljene ili izgubljene (primjerice zbog neispravnosti mjerne opreme) ili zato što nisu inicijalno ni prikupljene [5].

Najjednostavnije rješenje ovog problema je odbacivanje instanci s barem jednom nedostajućom vrijednosti značajke [5]. Nažalost ovo riješenje nije pogodno kada je količina instanci s nedostajućim vrijednostima znatno veća od onih bez nedostajućih vrijednosti pošto se tako uklanja većina podataka.

Alternativno rješenje je nadomještanje nedostajućih vrijednosti. Za nadomještanje nedostajućih vrijednosti moguće je koristiti vrijednost koja se najčešće pojavljuje za zadanu značajku, a u slučaju klasifikacije moguće je sve nedostajuće vrijednosti koje pripadaju istoj klasi zamijeniti najčešćom vrijednosti unutar te klase. U slučaju numeričkih vrijednosti umjesto najčešće vrijednosti koristi se srednja vrijednost dostupnih vrijednosti [5].

Nedostajuće podatke moguće je popuniti i koristeći metode regresije ili klasifikacije razvijanjem regresijskog ili klasifikacijskog modela temeljenom na kompletnim podacima za danu značajku. Vrijednosti se predviđaju koristeći sve ostale

relevatne značajke kao prediktore [5].

Još jedno riješenje je i takozvana *Hot Deck imputacija* gdje se identificira najsličniji slučaj onome s nedostajućom vrijednošću te se vrijedonst tog slučaja koristi da bi se zamijenila vrijednost slučaja koji sadrži nedostajuću vrijednost [5].

3.2 Odabir instanci

Osim nedostajućih vrijednosti, kvalitetu podataka mogu narušavati i takozvane nedopuštene vrijednosti (*engl. illegal values*). Čine ih vrijednosti koje se primjerice nalaze izvan dopuštenog raspona ili skupa očekivanih vrijednosti. Za predobradu takvih vrijednosti koristi se tehnika čišćenja podataka varijablu po varijablu (*engl. variable-by-variable data cleaning*). Ova je tehnika ujedno i primjer odabira instanci koristeći metodu filtera [5].

Metode odabira instanci dijele se na [5]:

- metode filtera koje razmatraju smanjenje podataka no ne uzimaju u obzir aktivnosti, te
- metode omotača (*engl. wrapper method*) koje izričito naglašavaju aspekt strojnog učenja i procjenjuju rezultate koristeći specifične algoritme strojnog učenja za pokretanje odabira instanci.

Kao primjer metode omotača u članku [5] spomenuto je uklanjanje instanci koje sadrže netočno označene podatke prije primjene odabranog algoritma strojnog učenja. To je postignuto korištenjem algoritama strojnog učenja za označavanje instanci kao točno ili netočno označenih, te potom uklanjanjem netočno označenih.

Osim za uklanjanje instanci koje sadrže nepotpune i nedopuštene vrijednosti, tehnike odabira instanci mogu biti korištene i za minimiziranje skupova podataka. Cilj je unatoč minimiziranju skupa podataka zadržati kvalitetu podataka. Kao najpoznatije tehnike korištene u ovu svrhu ističu se [5]:

- nasumično uzorkovanje - podskup podataka se nasumično izabire,
- stratificirano uzorkovanje - manje zastupljene klase učestalije se odabiru kako bi se postigla ravnomjerna raspodjela klasa. Ova tehnika je primjenjiva kada vrijednosti klasa u skupu podataka nisu jednoliko raspodijeljene.

U slučaju kada vrijednosti klasa u skupu podataka nisu jednoliko raspodijeljene također može doći do pretreniranosti (*engl. overfitting*), to jest algoritam strojnog učenja stvori hipotezu koja daje dobre rezultate nad skupom podataka za treniranje, no ne daje jednako dobre rezultate nad neviđenim podacima. Manje zastupljene klase mogu biti zanemarene što rezultira lošijim rezultatima. Za postizanje jednoličnije raspodjele moguće je ukloniti neke od instanci zastupljenije klase. Ova se tehnika naziva downsizing, a suprotna tehnika, odnosno tehnika redundantnog otipkavanja (*engl. oversampling*), uključuje dupliciranje instanci koje sadrže manje zastupljene klase [5].

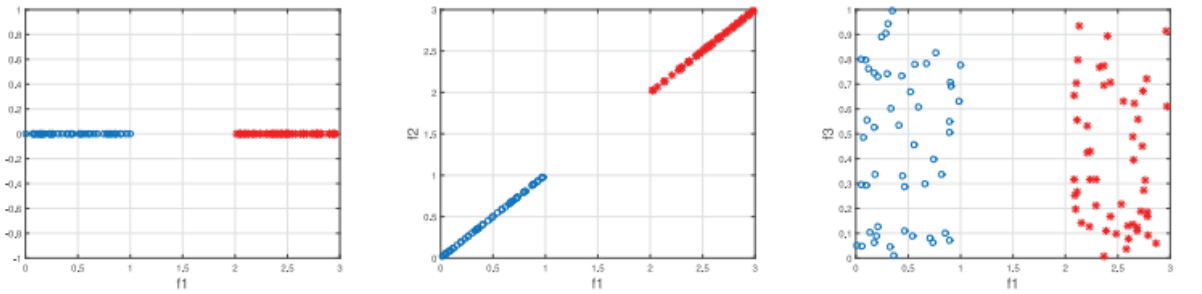
U slučaju kada skup podataka sadrži veći broj dupliciranih instanci koje pretjerano ne utječu na performanse algoritma već su jednostavno redundantne, minimiziranje podataka postiže se njihovim uklanjanjem.

3.3 Odabir značajki

U ovom su potpoglavlju obrađene neke od tehnika odabira značajki za čije je razumijevanje važno razlikovati sljedeće kategorije značajki [5]:

- relevantne - utječu na izlaz te njihovu ulogu ne mogu preuzeti ostale značajke,
- redundantne - značajka može preuzeti ulogu druge značajke,
- beznačajne - ne utječu na izlaz; njihove su vrijednosti nasumično generirane za svaki primjer.

Autori rada „Feature Selection: A Data Perspective" [6] koriste ilustrativan primjer relevantnih, redundantnih i beznačajnih značajki (slika 3.1) kako bi bolje objasnili ovu podjelu. Dvjema različitim bojama, plavom i crvenom, označene su dvije različite klase. 3.1 (a) prikazuje relevantnu značajku (predstavljenu funkcijom f_1); značajka je relevantna pošto se jasno razlikuje dvije klase. U idućem primjeru (b) dodana je i funkcija f_2 koja predstavlja redundantnu značajku pošto su dvije funkcije, f_1 i f_2, u snažnoj korelaciji. Posljednji primjer (c) prikazuje funkcije f_1 i f_3. Funkcija f_3 predstavlja beznačajnu značajku koja ne može jasno razdvojiti dvije klase.



Slika 3.1: *Ilustrativan primjer relevantnih, redundantnih i beznačajnih značajki [6]*

Dimenzionalnost podataka moguće je smanjiti i potpunim uklanjanjem beznačajnih i redundantnih značajki. Posljedično, uklanjanjem takvih značajki moguće je

poboljšavanje točnosti modela pošto se više ne uzimaju u obzir nebitni podaci [5].

Jedna od mogućih podjela odabira značajki je na: nadzirani (*engl. supervised*), nenadzirani (*engl. unsupervised*) i polu-nadzirani (*engl. semi-supervised*) odabir značajki. Nadzirani odabir značajki općenito je dizajniran za probleme klasifikacije i regresije te je cilj ovog pristupa odabrati podskup značajki koje mogu razlikovati uzorke iz različitih klasa ili se približiti ciljevima regresije. Za razliku od nadziranog odabira značajki, nenadzirani odabir značajki općenito je dizajniran za probleme grupiranja (*engl. clustering*) te ne zahtijeva oznake klasa [6]. U ovom su završnom radu korišteni algoritmi klasifikacije stoga je nadzirani odabir značajki detaljnije obrađen.

Dodatno, s obzirom na strategije odabira, metode odabira značajki dijele se na, metode filtera, omotača i ugrađene (*engl. embedded*) metode. U slučaju ugrađenih metoda, odabir značajki ugrađen je u temeljni model [6]. Metode filtera i omotača već su spomenute i obrađene u potpoglavlju „Odabir instanci".

Metode filtera neovisne su o algoritmu koji će kasnije koristiti konačan skup podataka te se sastoje od dva koraka [6]:

1. Važnost značajki se rangira u odnosu na kriterije vrednovanja značajki.
2. Nisko rangirane značajke se filtriraju.

Proces vrednovanja značajki može biti jednovarijantni ili viševarijantni, gdje se kod jednovarijantnog procesa svaka značajka rangira neovisno o drugim značajkama dok se u slučaju multivarijantnog pristupa više značajki rangira na serijski način [6].

Prema [5], kriteriji vrednovanja značajki dijele se u odnosu na:

- udaljenost - za problem koji sadrži dvije klase, značajka ima prednost u odnosu na drugu značajku ukoliko inducira veću razliku između uvjetne vjerojatnosti dviju klasa
- informacije - značajka ima prednost u odnosu na drugu značajku ukoliko se njome dobiva veća količina informacija
- ovisnost - značajka ima prednost u odnosu na drugu značajku ukoliko joj je koeficijent korelacije s određenom klasom veći od koeficijenta korelacije spomenute druge značajke i te iste klase
- konzistentnost - ukoliko primjeri sadrže jednake vrijednosti za podskup značajki, no ne pripadaju istoj klasi, primjeri su u sukobu

Algoritme odabira značajki, vezane uz metode omotača, općenito čine dvije komponente [5]:

- algoritam odabira koji generira predložene podskupove značajki i pokušava pronaći optimalan podskup i
- algoritam vrednovanja koji određuje kvalitetu predloženog podskupa značajki te vraća rezultate algoritmu odabira.

Kako se proces odabira značajki ne bi izvodio iscrpno uključuje se kriterij zaustavljanja; primjerice da dodavanje ili uklanjanje bilo koje značajke ne stvara bolji podskup. Kod pristupa koristeći metode omotača prostor pretraživanja za d značajki (gdje d označuje broj značajki) je 2^d [6] te posljedično, ova metoda može biti nepraktična ukoliko se koristi na velikim skupovima podataka.

Procesom koračajnog unaprijed uključivanja (*engl. forward stepwise selection*)

podskup značajki se iterativno nadograđuje tako što se svaka neiskorištena varijabla redom dodaje modelu te se odabire varijabla koja najbolje pospješuje odabrani model. Ovaj je način odabira značajki brz pri lociranju malih efektivnih podskupova no pošto dodaje varijable jednu po jednu moguće je da ne uspije uključiti sve međuovisne varijable. Suprotno tome, proces koračajnog unazad uključivanja (*engl. backward stepwise selection*) dobro upravlja međuovisnim varijablama no kod ovog procesa rana vrednovanja su relativno skupa. Razlog tome je što algoritam započinje gradnju modela uključujući sve dostupne ulazne varijable te se svakom iteracijom uklanjaju varijable za koje se smatra da će njihovo uklanjanje poboljšati izvođenje modela [5].

Prema [6], tradicionalni algoritmi odabira značajki za konvencionalne podatke grupirani su kao metode koje se temelje na sličnosti, metode koje se temelje na informacijskoj teoriji, metode utemeljene na rijetkom učenju (*engl. sparse-learning-based*), metode temeljene na statistici i druge metode prema korištenim tehnikama. U nastavku će biti obrađene neke od metoda predstavljene u spomenutom radu. Dodatno, u narednim primjerima, značajke će biti označene oznakom f_i.

Kao primjer metode koja se temelji na sličnosti dan je algoritam nadziranog odabira značajki *Fisher Score*. Ovaj algoritam odabire značajke čije su vrijednosti značajki primjera koji pripadaju istoj klasi slične dok su vrijednosti značajki primjera koji ne pripadaju istoj klasi različite [6]. Najbolje se značajke dobivaju odabirom značajki s najvećom ocjenom *fisher score*. U radu je dana i jednadžba kojom se procjenjuje metrika *Fisher Score* svake značajke f_i:

$$fisherscore(f_i) = \frac{\Sigma_{j=1}^{c} n_j(\mu_{ij} - \mu_i)^2}{\Sigma_{j=1}^{c} n_j \sigma_{ij}^2}. \tag{3.1}$$

U izrazu 3.1 oznaka *c* označuje broj postojećih klasa, n_j označuje broj uzoraka u klasi *j*, μ_i srednju vrijednost značajke f_i, μ_{ij} srednju vrijednost značajke f_i za

uzorke iz klase *j* te σ_{ij}^2 vrijednost varijance značajke f_i za uzorke u klasi *j*.

Metode temeljene na statistici većinom su temeljene i na metodama filtera pošto za procjenu značajnosti značajke umjesto algoritama strojnog učenja koriste statističke mjere. Uz to, većina tih algoritama značajke analizira individualno zbog čega se redundantnost značajki zanemaruje. Jedna od ovakvih metoda je *Chi-Square Score* koja koristi test neovisnosti za procjenu je li značajka neovisna o oznaci klase. Jednadžba za *Chi Square Score* za značajku f_i koja poprima *r* različitih vrijednosti glasi [6]:

$$chisquarescore(f_i) = \sum_{j=1}^{r} \sum_{s=1}^{c} \frac{(n_{js} - \mu_{js})^2}{\mu_{js}}. \tag{3.2}$$

Oznaka *c* označuje broj postojećih klasa, oznakom n_{js} označen je broj instanci s *j*-om vrijednošću značajke dane značajke f_i. Vrijednost μ_{js} dobiva se jednadžbom:

$$\mu_{js} = \frac{n_{*s} n_{j*}}{n}, \tag{3.3}$$

gdje n_{j*} označuje broj instanci podataka s *j*-om vrijednošću dane značajke f_i, n_{*s} označuje broj instanci podataka u klasi označenoj oznakom *r*, a *n* označuje broj instanci u podacima. Što je ocjena *Chi-Square* veća, značajka je bitnija.

4 METODOLOGIJA

4.1 Korištene tehnologije

Za ispitivanje učinkovitosti tehnika predobrade podataka u izgradnji modela predviđanja korišten je programski jezik Python. Knjižnice funkcija NumPy, pandas [7] i scikit-learn [8] korištene su za dubinsku analizu podataka. Za mjerenje energetskog utroška procesora i radne memorije te utroška vremena korišten je softverski alat pyRAPL [9] koji koristi Intel *Running Average Power Limit* (RAPL) [10] tehnologiju koja procjenjuje energetsku potrošnju procesora. Sva su mjerenja izvedena na, za vrijeme izrade rada, najnovijoj LTS verziji Ubuntu operacijskog sustava. Uz to, svako je mjerenje ponovljeno 10 puta te je za konačni rezultat uzeta aritmetička sredina rezultata mjerenja.

Kako bi se izmjerila energija koju troši stroj tijekom izvođenja funkcije foo() te u konzoli ispisala zabilježena potrošnja energije svih domena procesora, pokreće se sljedeći kod:

```
import pyRAPL

  pyRAPL.setup()

  @pyRAPL.measure
  def foo():
    # Dodati instrukcije.

  foo()
```

Pošto zabilježena potrošnja energije nije samo potrošnja energije koda koji se izvodi već globalna potrošnja energije svih procesa koji se izvode na stroju tijekom razdoblja mjerenja, preporuča se uklanjanje svih dodatnih programa koji mogu promijeniti potrošnju energije stroja koji hostira eksperimente [9].

S obzirom da pyRAPL ne daje informaciju o utrošku memorije izraženom u bajtovima već samo energetski utrošak radne memorije, koristiti će se i knjižnica funkcija psutil [11] čija funkcija **psutil.virtual_memory()** vraća statistiku o upotrebi memorije sustava izraženu u bajtovima.

4.2 Skupovi podataka

U svrhu ispitivanja učinkovitosti tehnika predobrade podataka u izgradnji modela predviđanja korištena su dva skupa podataka; skupovi podataka *„Dry Bean Dataset”* [12] i *„Polish companies bankruptcy data”* [13]. Skupovi podataka preuzeti su s repozitorija *UCI Machine Learning Repository* [14].

Skup podataka *„Dry Bean Dataset”* sadrži 13 611 instanci te 16 značajki. Svaka instanca predstavlja pojedinačno zrno graha te se temeljem značajki koje opisuju oblik i strukturu predviđa klasa, odnosno jedna od 7 mogućih vrsti graha. U ovoj se bazi ne pojavljuju nedostajuće vrijednosti.

Idući skup podataka, skup podataka *„Polish companies bankruptcy data”* [13] odnosi se na predviđanje bankrota poljskih tvrtki. Na temelju prikupljenih podataka izdvojeno je pet slučajeva klasifikacije, ovisno o razdoblju predviđanja. Za potrebe ovog završnog rada korišten je 3. slučaj, „3rd year”. U ovom slučaju podaci sadrže financijske stope iz 3. godine razdoblja predviđanja i odgovarajuću oznaku klase koja pokazuje status bankrota nakon 3 godine. Skup podataka sadrži 10 503 instanci od kojih 495 predstavljaju poduzeća u stečaju dok ostalih 10 008 poduzeća koja nisu bankrotirala u razdoblju predviđanja. Spomenut skup podataka odabran je za potrebe ovog rada pošto sadrži nedostajuće vrijednosti te se pomoću njega moglo ispitati koliko uklanjanje nedostajućih vrijednosti utječe na točnost modela

i utrošak energije, memorije i vremena.

Priprema podataka zahtijeva i podjelu skupa podataka na ulazni i izlazni skup podataka. Ulazni skup podataka čine neovisne značajke koje se koriste za predviđanje klase, dok izlazni skup sadrži vrijednost atributa klase te ovisi o vrijednostima značajki iz ulaznog skupa. Ulazni skup označen je oznakom X dok je izlazni skup oznakom y.

4.3 Primjena tehnika predobrade podataka u izgradnji modela predviđanja

Nad skupom podataka *„Dry Bean Dataset"* primjenjena je tehnika uklanjanja dupliciranih instanci te tehnika odabira najboljih značajki.

Funkcija **pandas.read_csv()** čita datoteku s vrijednostima odvojenim zarezima (csv) u DataFrame pandas objekt, dvodimenzionalnu strukturu označenih podataka.

```
import pandas as pd


df = pd.read_csv('data/DryBeanDataset/Dry_Bean_Dataset.csv')
```

Nad učitanim skupom podataka za početak je primijenjena tehnika uklanjanja dupliciranih instanci.

```
#inplace: bool, default False
#Odreduje hoce li se DataFrame modificirati umjesto stvaranja
   novog.
df.drop_duplicates(inplace=True)
```

Nakon izvršenja ove linije koda, broj instanci se s inicijalnih 13 611 smanjio na 13 543.

Zatim je skup podataka podijeljen na ulazni skup podataka X i izlazni skup podataka y.

```
X = df.drop(columns=['Class'])
y = df['Class']
```

Prvi klasifikator, odnosno algoritam strojnog učenja za klasifikaciju podataka, na kojemu je ispitana učinkovitost tehnika predobrade podataka je klasifikator stabla odluke. Stabla odluke su neparametarska metoda nadziranog učenja korištena za klasifikaciju i regresiju, a čiji je cilj stvaranje modela koji predviđa vrijednosti ciljne varijable kroz učenje jednostavnih pravila odlučivanja izvedenih iz značajki podataka [15]. Vrijednosti parametara ostavljene su na zadanim vrijednostima.

Kako bi se ocijenio rezultat klasifikacije korištena je unakrsna validacija (*engl. cross-validation*) u 10 preklopa (*engl. folds*). Pošto se radi o klasifikaciji korištena je stratificirana inačica validatora koja omogućuje ravnomjernu raspodjelu klasa po preklopima. Bitno je parametru *shuffle* dodijeliti vrijednost *True* uzimajući u obzir da su podaci u originalnom skupu podataka poredani po klasama.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score,
   StratifiedKFold
from sklearn.metrics import accuracy_score

clf = DecisionTreeClassifier()
skf = StratifiedKFold(10, shuffle = True)
scores = cross_val_score(clf, X, y, cv = skf, scoring="accuracy")
```

Sljedeća tehnika predobrade ispitana u ovom radu je tehnika odabira jednovarijantnih značajki. Funkcionira odabirom najboljih značajki na temelju jednovarijantnih statističkih testova. Scikit-learn izlaže rutine odabira značajki kao objekte koji implementiraju metodu transformacije [16].

```
from sklearn.feature_selection import SelectPercentile
```

```
X_new = SelectPercentile(percentile=75).fit_transform(X, y)
```

Parametar *percentile* određuje koliki će postotak značajki biti zadržan, a kriterij odabira je najviši rezultat funkcije bodovanja. U ovom je slučaju funkcija bodovanja funkcija **sklearn.feature_selection.f_classif** kojom se izračunava F-vrijednost analize varijence (ANOVA) za dani uzorak. To je ujedno i zadana funkcija.

U ovom je radu ispitano koliko na točnost algoritma, utrošak vremena, energije i memorije utječe odabir 75 %, 50 % te konačno 25 % najboljih značajki. Rezultati mjerenja za sve slučajeve predstavljeni su u sljedećem poglavlju, poglavlju „Rezultati".

Sljedeći klasifikator na kojem je ispitana učinkovitost uklanjanja dupliciranih instanci i odabira najboljih značajki je klasifikator nasumične šume. To je meta estimator koji koristi određeni broj klasifikatora stabla odlučivanja na različitim poduzorcima skupa podataka te koristi usrednjavanje za poboljšavanje prediktivne točnosti i kontrolu pretreniranosti [17].

```
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier()
```

Skup podataka *„Polish companies bankruptcy data"* korišten je pošto sadrži nedostajuće vrijednosti te su nad tim skupom podataka primjenjene tehnika uklanjanja instanci i značajki koje sadrže nedostajuće vrijednosti. Učitavanje ovog skupa podataka nešto je drugačije s obzirom da dolazi u .arff formatu.

```
from scipy.io.arff import loadarff
import pandas as pd

#Ucitavanje skupa podataka.
```

```
data = loadarff('data/polish+companies+bankruptcy+data/3year.arff
   ')
df = pd.DataFrame(data[0])


df['class'] = df['class'].str.decode('utf-8')


#Zamjena '?' NaN vrijednoscu.
df = df.replace(r"\s*\?\s*", np.nan, regex=True)
```

Stupac koji sadrži vrijednosti klase bilo je potrebno dekodirati. Dodatno, nedostajuće vrijednosti koje su u ovom skupu podataka predstavljene znakom '?' zamijenjene su **NaN** oznakom koja je u pandas knjižnici zadana oznaka za nedostajuće vrijednosti. Ovaj korak u nastavku pojednostavljuje rad s nedostajućim vrijednostima.

Pošto korišten skup podataka sadrži nedostajuće vrijednosti korišteno je klasifikacijsko stablo za podizanje gradijenata temeljeno na histogramu (*engl. Histogram-based Gradient Boosting Classification Tree*). Razlog tome je što scikit-learn implementacija ovog klasifikatora sadrži izvornu podršku za nedostajuće, odnosno **NaN** vrijednosti. To se postiže tako što se, tijekom treniranja modela, na svakoj točki razdvajanja uči trebaju li uzorci s nedostajućim vrijednostima ići lijevom ili desnom djetetu temeljeno na potencijalnom dobitku. Ukoliko se prilikom predviđanja naiđe na nedostajuću vrijednost koja nije viđena za vrijeme faze treniranja, tada se uzorak s nedostajućom vrijednošću mapiraju na dijete koje sadrži najviše uzoraka [18].

```
clf = HistGradientBoostingClassifier()
```

Za početak je isprobano uklanjanje svih instanci koje sadrže nedostajuće vrijednosti.

```
df = df.dropna(axis=0)
```

Ovime se broj instanci smanjio s inicijalnih 10 503 na 4 885. Time se gubi više od polovica podataka te bi u ovom slučaju bolje riješenje bilo koristiti jednu od tehnika za nadomještanje nedostajućih vrijednosti.

Zatim je ispitano koliko uklanjanje značajki koje sadrže nedostajuće vrijednosti utječe na performanse modela. Za uklanjanje značajki koristi se ista funkcija kao i u prijašnjem primjeru, no parametaru *axis* pridodaje se vrijednost 1.

```
df = df.dropna(axis=1)
```

Broj značajki smanjuje se s inicijalnih 64 na 21.

5 REZULTATI

Tablica 5.1 sadrži rezultate mjerenja koristeći skup podataka *„Dry Bean Dataset”* i klasifikator stabla odluke. Točnost klasifikatora za slučaj bez predobrade podatake je 89,53 % te je vrijeme izvođenja bilo svega t = 6s dok je energetski utrošak procesora bio E = 26,14 J, a radne memorije E = 0,9 J. Utrošak resursa se smanjivanjem količine podataka u većini slučajeva također smanjivao, gdje je za slučaj odabira 50 % najboljih značajki utrošak resursa bio više nego duplo manji od utroška resursa za slučaj s neobrađenim podacima, a točnost algoritma se smanjila za 1,88 %. Najveća razlika je bila za slučaj odabira 25 % najboljih značajki gdje je utrošak vremena bio tri puta manji od utroška vremena za slučaj s neobrađenim podacima, a energetski utrošak procesora se smanjio za 74,14 %. Međutim, u ovom se slučaju i točnost algoritma znatno pogoršala, odnosno za 12,71 %.

Rezultati mjerenja koristeći skup podataka *„Dry Bean Dataset”* i klasifikator nasumične šume prikazani su tablicom 5.2. Kao što je iz rezultata vidljivo uklanjanje duplikata nije uveliko utjecalo na točnost klasifikatora kao ni na uštedu resursa. Štoviše, pri izgradnji modela koristeći podskup podataka koji ne sadrži dupliciranе insatnce utrošak energije bio je veći. Naravno, treba uzeti u obzir da je uklanjanjem dupliciranih instanci uklonjeno samo 0.50% podataka te bi u slučaju gdje je postotak dupliciranih podataka veći, razlika zasigurno bila veća.

Također, moguće je primijetiti i razliku u utrošku resursa u odnosu na prijašnji primjer. Razlika je očekivana s obzirom da je klasifikator nasumične šume složeniji od klasifikatora stabla odluke.

Kao relativno najoptimalnija tehnika istakla se tehnika odabira 50 % najboljih značajki. Korištenjem ove tehnike utrošak vremena smanjio se za 48,86 %, energetski

Tablica 5.1: *Rezultati mjerenja koristeći skup podataka „Dry Bean Dataset" i klasifikator stabla odluke*

RAZINA PREDOBRADE	**TOČNOST KLASIFIKA-TORA (%))**	**KORIŠTENA MEMORIJA (gb)**	**VRIJEME IZVOĐENJA (mm:ss)**	**ENERGETSKI UTROŠAK PROCESORA (J)**	**ENERGETSKI UTROŠAK RADNE MEMORIJE (J)**
BEZ PREDOBRADE	89,53	1,0982	00:06	26,14	0,9
UKLANJANJE DUPLIKATA	89,68	1,1022	00:06	25,66	1,76
ODABIR NAJBOLJIH ZNAČAJKI (75 (%))	87,53	1,1028	00:04	17,96	0,6
ODABIR NAJBOLJIH ZNAČAJKI (50 (%))	87,65	1,1103	00:03	11,92	0,4
ODABIR NAJBOLJIH ZNAČAJKI (25 (%))	76,82	1,1094	00:02	6,76	0,3

utrošak procesora za 43,77 % i energetski utrošak radne memorije za 45,48 %. Uz svu uštedu resursa, točnost klasifikatora smanjila se za 1.9 %.

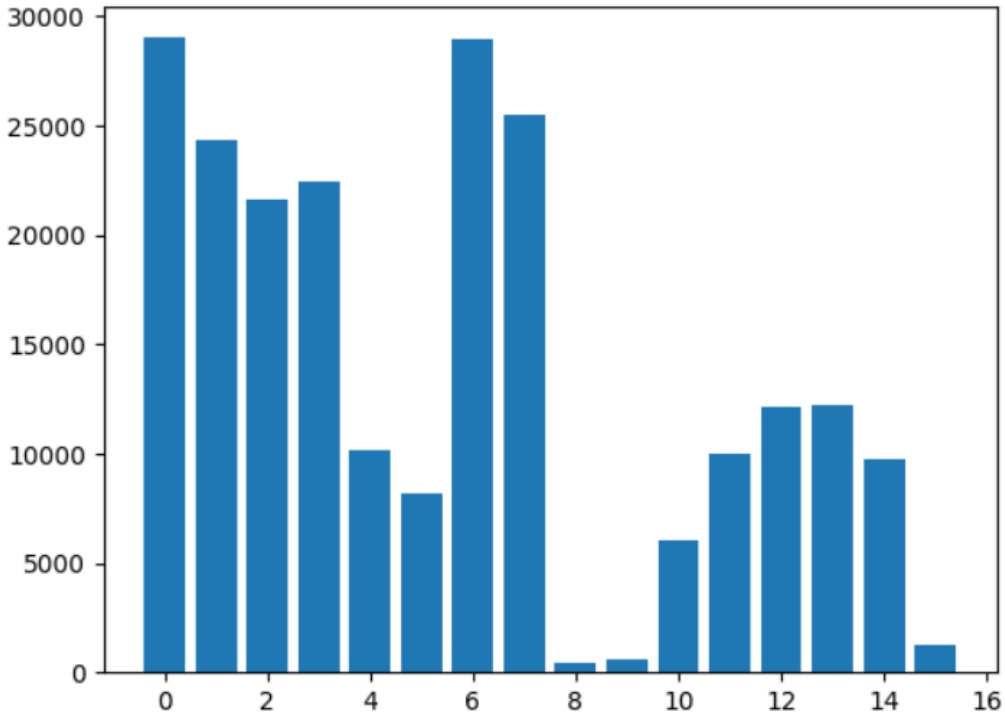
Za slučaj odabira 25 % najboljih značajki točnost klasifikatora je kao i u prijašnjem slučaju znatno pogoršana, točnije 10,9 % u odnosu na slučaj s neobrađenim podacima, a zanimljivo je da je prosječan utrošak memorije, vremena i energije procesora bio malo veći u odnosu na slučaj odabira 50 % najboljih značajki.

Znatan pad u točnosti klasifikatora za zadnji slučaj opravdan je dijagramom na slici 5.1. Dijagram prikazuje odnos značajki (os x) i rezultata funkcije bodovanja značajki (os y). Odabirom 25 % najboljih značajki izostavljene su barem dvije značajke koje su imale veliki utjecaj na performanse klasifikatora.

Tablica 5.2: *Rezultati mjerenja koristeći skup podataka „Dry Bean Dataset" i klasifikator nasumične šume*

RAZINA PREDOBRADE	**TOČNOST KLASIFIKA-TORA (%))**	**KORIŠTENA MEMORIJA (gb)**	**VRIJEME IZVOĐENJA (mm:ss)**	**ENERGETSKI UTROŠAK PROCESORA (J)**	**ENERGETSKI UTROŠAK RADNE MEMORIJE (J)**
BEZ PREDOBRADE	92,75	1,1777	01:28	345,86	13,72
UKLANJANJE DUPLIKATA	92,5	1,0931	01:20	379,15	13,6
ODABIR NAJBOLJIH ZNAČAJKI (75 (%))	90,66	0,9297	01:01	287,12	10,5
ODABIR NAJBOLJIH ZNAČAJKI (50 (%))	90,85	1,1020	00:45	194,46	7,48
ODABIR NAJBOLJIH ZNAČAJKI (25 (%))	81,85	1,0404	00:48	206,00	0,77

Tablica 5.3 prikazuje rezultate mjerenja koristeći skup podataka *„Polish companies bankruptcy"* i klasifikator podizanja gradijenata temeljen na histogramu u slučaju inicijalnih, neobrađenih podataka, te podataka s uklonjenim nedostajućim vrijednostima. Točnost klasifikatora za slučaj neobrađenih podataka iznimno je visoka, 97,03 %. Nakon uklanjanja instanci koje sadrže nedostajuće vrijednosti točnost se poboljšala za 1,05%, a utrošak resursa smanjen je; vrijeme izvođenja s t = 16s na t = 11s, energetski utrošak procesora s E = 106,50 J na E = 73,08 J te energetski utrošak radne memorije s E = 7,87 J na E = 5,65 J. U slučaju uklanjajna značajki koje sadrže nedostajuće vrijednosti utrošak resursa bio je još manji; vrijeme trajanja bilo je t = 7s, energetski utrošak procesora E = 36,82 J te energetski utrošak radne memorije E = 1,72 J. Točnost klasifikatora u ovom slučaju



Slika 5.1: *Odnos značajki (os x) i rezultata funkcije bodovanja (os y)*

bila je 95,02% što je manje od prijašnja dva slučaja, no i dalje jako dobro. Iako ove tehnike daju naizgled odlične rezultate, uklanjanjem nedostajućih vrijednosti izgubila se velika količina podataka te podskupovi obrađenih podataka ne moraju nužno vjerno predstavljati inicijalni skup. Zbog ovoga je nužno dobro razumijevanje problema i podataka čime se još jednom ističe važnosti početnih faza dubinske analize podataka.

Tablica 5.3: *Rezultati mjerenja koristeći skup podataka „Polish companies bankruptcy" i klasifikator podizanja gradijenata temeljen na histogramu*

RAZINA PREDOBRADE	**TOČNOST KLASIFIKA-TORA (%))**	**KORIŠTENA MEMORIJA (gb)**	**VRIJEME IZVOĐENJA (mm:ss)**	**ENERGETSKI UTROŠAK PROCESORA (J)**	**ENERGETSKI UTROŠAK RADNE MEMORIJE (J)**
BEZ PREDOBRADE	97,03	0,9059	00:16	106,50	7,87
UKLANJANJE INSTANCI	98,08	0,9027	00:11	73,08	5,65
UKLANJANJE ZNAČAJKI	95,02	0,9015	00:07	36,82	1,72

6 ZAKLJUČAK

Kako bi se osigurala maksimalna ušteda resursa, procesu dubinske analize podataka uvijek bi trebalo pristupati uzimajući u obzir održivost samog procesa. Odličnu bazu za implementaciju održivosti procesa dubinske analize podataka čine načela predstavljena u radu „Principles of Green Data Mining" [4], a obrađena u poglavlju „Proces dubinske analize podataka" ovog završnog rada.

U ovom su radu nad karakterističnim skupovima podataka primjenjene tehnike uklanjanja instanci i značajki koje sadrže nedostajeće vrijednosti, uklanjanje dupliciranih instanci i tehnika odabira najboljih značajki.

Tehnike uklanjanja instanci i značajki koje sadrže nedostajuće vrijednosti korištene su u kombinaciji s klasifikatorom podizanja gradijenata temeljenom na histogramu te skupom podataka *„Polish companies bankruptcy"*. Nakon uklanjanja instanci koje sadrže nedostajuće vrijednosti utrošak vremena u prosjeku je bio manji za 31,25 %, energetski utrošak procesora za 31,38 % te energetski utrošak radne memorije za 28,21 %. Točnost klasifikatora poboljšala se za 1,05 %, međutim uklanjanjem instanci koje sadrže nedostajuće vrijednosti uklonjeno je ukupno 53,49 % instanci. Uklanjanjem značajki koje sadrže nedostajuće vrijednosti broj značajki smanjio se s inicijalnih 64 na 21. Točnost klasifikatora se smanjila za 2,01 % u odnosu na scenarij izvornih podataka, a smanjio se i utrošak vremena za 43,75 %, energetski utrošak procesora za 65,43 % te energetski utrošak radne memorije za 78,14 %.

Tehnike uklanjanja dupliciranih instanci i odabira najboljih značajki korištene su u kombinaciji s klasifikatorima stabla odluke i nasumične šume. Na korištenom

skupu podataka, „Dry Bean Dataset" skupu podataka, najoptimalniji rezultati dobiveni su odabirom 50 % najboljih značajki. Točnost klasifikatora u tom slučaju smanjila se za 1,88 % u odnosu na scenarij izvornih podataka, međutim vrijeme izvođenja bilo je upola kraće, energetski utrošak smanjio se za 54,40 % te energetski utrošak radne memorije za 55,56 %. Točnost klasifikatora nasumične šume za slučaj odabira 50 % najboljih instanci smanjila se za 1,9 % u odnosu na scenarij izvornih podataka uz smanjenje utroška vremena za 48,83 %, energetskog utroška procesora za 43,77 % te energetskog utroška radne memorije za 45,48 %. Uklanjanje dupliciranih instanci nije uvelike utjecalo na točnost klasifikatora kao ni na uštedu mjerenih resursa što se može pripisati činjenici da za korišten skup podataka uklanjanjem dupliciranih instanci uklonjeno svega 0,5 % instanci.

Kvalitetnom predobradom podataka moguće je smanjiti utoršak energije i vremena bez ugrožavanja performansi modela predviđanja. Štoviše, kvalitetnom se predobradom podataka često rješava i problem pretreniranosti što samim time dodatno poboljšava performanse modela predviđanja. Da bi rezultati predobrade podataka bili što kvalitetniji važno je dobro razumijevanje problema i podataka. Važno je da se tehnikama čišćenja ne narušava reprezentacija određenih vrijednosti u podacima.

Sve u svemu, proces dubinske analize podataka pomaže pri otkrivanju korisnih informacija iz skupova podataka no ukoliko se u obzir ne uzimaju principi održive dubinske analize podataka može biti energetski i vremenski nepotrebno iscrpan proces. Održive navike temelj su budućnosti te je stoga poželjno da se pri implementaciji svakog procesa pridaje posebna pažnja maksimiziranju održivosti procesa.

Literatura

[1] ”What is data mining?”, s Interneta, https://www.ibm.com/topics/data-mining, 30. lipnja 2023.

[2] Wirth, R.; Hipp, J.: ”CRISP-DM: Towards a Standard Process Model for Data Mining”, Njemačka, 2000.

[3] Ujević, F.: ”Postupci i tehnike dubinske analize podataka”, kvalifikacijski rad, Rijeka, 2015.

[4] Schneider, J.; Basalla, M.; Seidel, S.: ”Principles of Green Data Mining”, Proceedings of the 52nd Hawaii International Conference on System Sciences, pp. 2065-2074, 2019.

[5] Kotsiantis, S. B.; Kanellopoulos, D.; Pintelas, P. E.: ”Data Preprocessing for Supervised Leaning”, INTERNATIONAL JOURNAL OF COMPUTER SCIENCE, VOLUME 1 NUMBER 1, 111-117, 2006.

[6] Li, Jundong i dr.: ”Feature Selection: A Data Perspective”, ACM Computing Surveys, Vol. 50, No. 6, Article 94, SAD, 2017.

[7] ”pandas documentation”, s Interneta, https://pandas.pydata.org/docs/index.html, 03. srpnja 2023.

[8] ”scikit-learn”, s Interneta, https://scikit-learn.org/stable/index.html, 03. srpnja 2023.

[9] ”PyRAPL”, s Interneta, https://pypi.org/project/pyRAPL/, 03. srpnja 2023.

[10] Intel Corporation, ”Running Average Power Limit Energy Reporting”, s Interneta, https://www.intel.com/content/www/us/en/developer/articles/technical/

software-security-guidance/advisory-guidance/running-average-power-limit-energy-reporting.html, 10. rujna 2023.

[11] Rodola, G., ”psutil documentation”, s Interneta, https://psutil.readthedocs.io/en/latest/ 10.rujna 2023.

[12] ”Dry Bean Dataset”, UCI Machine Learning Repository, https://doi.org/10.24432/C50S4B, 2020.

[13] Tomczak, S., ”Polish companies bankruptcy data”, UCI Machine Learning Repository, https://doi.org/10.24432/C5F600, 2016.

[14] Markelle Kelly, Rachel Longjohn, Kolby Nottingham, The UCI Machine Learning Repository, https://archive.ics.uci.edu

[15] ”1.10. Decision Trees”, s Interneta, https://scikit-learn.org/stable/modules/tree.html#decision-trees, 03. rujna 2023.

[16] ”1.13. Feature selection”, s Interneta, https://scikit-learn.org/stable/modules/feature_selection.html#feature-selection, 03. rujna 2023.

[17] ”sklearn.ensemble.RandomForestClassifier”, s Interneta, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier, 03. rujna 2023.

[18] ”sklearn.ensemble.HistGradientBoostingClassifier”, s Interneta, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html, 03. rujna 2023.

POPIS OZNAKA I KRATICA

ANOVA - Analiza varijance

CRISP-DM - CRoss Industry Standard Process for Data Mining

IT - Informacijska tehnologija

KDD - Knowledge Discovery in Databases

LTS - Long Term Support

RAPL - Running Average Power Limit

SAŽETAK

Zadatak završnog rada bio je ispitati učinkovitost tehnika predobrade podataka u izgradnji modela predviđanja. Proces dubinske analize podataka objašnjen je CRISP-DM procesnim modelom. Dodatno, obrađena su načela održive dubinske analize podataka vezana uz ključne faze procesa dubinske analize podataka. Proučeni su postupci predobrade podataka u dubinskoj analizi podataka, točnije tehnike čišćenja podataka kojima se uklanjaju instance i značajke. Nad karakterističnim skupovima podataka primjenjene su tehnike uklanjanja instanci i značajki koje sadrže nedostajeće vrijednosti, uklanjanje dupliciranih instanci i tehnika odabira najboljih značajki. Korišteni su algoritmi strojnog učenja stablo odluke i nasumična šuma te algoritam podizanja gradijenata temeljen na histogramu (*engl. Histogram-based Gradient Boosting*). Izmjeren je utrošak energije, memorije i vremena za scenarije izvornih i očišćenih podataka. Zaključeno je da je kvalitetnom predobradom podataka moguće smanjiti utrošak energije i vremena bez ugrožavanja performansi modela predviđanja.

Ključne riječi: dubinska analiza podataka, predobrada podataka, tehnike čišćenja podataka, održivost, održiva dubinska analiza podataka

ABSTRACT

The task of this thesis was to examine the efficiency of data preprocessing techniques in building a prediction model. Data mining is explained using the CRISP-DM process model. The thesis also covers the principles of green data mining related to the key stages of the data mining process. Data preprocessing techniques, specifically data cleaning techniques used to remove instances and features (instance and feature selection techniques), were researched. The techniques of removing instances and features that contain missing values, removing duplicate instances and feature selection were applied to the characteristic data sets. Decision tree, random forest and histogram-based gradient boosting machine learning algorithms were used. The consumption of energy, memory and time was measured for the original and cleaned data scenarios. It was concluded that with high-quality data preprocessing, it is possible to reduce the consumption of energy and time without disrupting the performance of the prediction model.

Key words: data mining, data preprocessing, data cleaning techniques, sustainability, green data mining