Algoritmi strojnog učenja u analizi medicinske dokumentacije

Gabaldo, Fran

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:173654

Rights / Prava: Attribution-NoDerivs 3.0 Unported/Imenovanje-Bez prerada 3.0

Download date / Datum preuzimanja: **2024-12-03**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Fran Gabaldo

Algoritmi strojnog učenja u analizi medicinske dokumentacije

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Fran Gabaldo

Matični broj: 0016153480

Studij: Informacijski i poslovni sustavi

Algoritmi strojnog učenja u analizi medicinske dokumentacije

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Markus Schatten

Varaždin, lipanj 2024.

Fran Gabaldo

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Završni rad bavi se istraživanjem primjene metoda i algoritama strojnog učenja na području medicine, pri tome se osvrćući na teorijski dio i istraživanje o stvarnim primjenama, kao i praktični dio unutar kojega je autor na manjem skupu medicinskih podataka razvio i usporedio rezultate triju različitih modela: stabla odlučivanja (engl. *Decision Tree*), Metode potpornih vektora (engl. *Support Vector Machine*) te K najbližih susjeda (engl. *K Nearest Neighbours*). Radom se na teorijskoj i praktičnoj razini s pomoću programskog jezika Python prikazala mogućnost primjene strojnog učenja na području medicine. Teorijski dio obrađuje teorijske koncepte vezane za strojno učenje te primjenu i probleme primjene strojnog učenja u medicini, dok se praktični dio bavi implementacijom ranije spomenutih algoritama na stvarnom skupu podataka.

Ključne riječi: medicina; strojno učenje; Python; etika; algoritmi; predviđanje; srčani zastoj

Sadržaj

1. Uvod

Strojno učenje kao grana umjetne inteligencije primjenjuje se već duži niz godina u raznim područjima kao što su prepoznavanje fotografija, autonomna vozila, algoritmi preporuka na platformama za pregled video sadržaja, a eksploziju popularnosti doživjelo je krajem 2022. godine pojavom velikog jezičnog modela ChatGPT. Pojava ovog modela uzrokovala je velik interes šire javnosti za područje umjetne inteligencije, specifično strojnog učenja, te njegovu primjenu u raznim područjima života, jedno od kojih je područje medicine.

Svrha korištenja strojnog učenja u medicini jest pomoć medicinskim stručnjacima pri donošenju odluka, otkrivanju bolesti i anomalija u ponašanju organizma pacijenta, odnosno olakšavanje rada medicinskih stručnjaka i ubrzavanje jednostavnih zadataka koje je moguće delegirati strojevima. Ovdje dolazimo do etičkog problema korištenja strojnog učenja u medicini, budući da se radi o strogo povjerljivim podacima pacijenata, kao i o činjenici da greška stroja može biti uzrok krive dijagnoze i potencijalne smrti pacijenta.

1.1. Problemska domena

Cilj ovog rada jest predstaviti teorijski, etički i praktični aspekt primjene strojnog učenja u području medicine. U teorijskom dijelu poseban naglasak bit će stavljen na metode strojnog učenja te mogućnosti njihove primjene u raznim medicinskim problemima, specifično problemima predviđanja ishoda. U etičkom dijelu rada bit će predstavljeni etički problemi korištenja strojnog učenja u ovom području, kao i mišljenje struke o etičkoj komponenti problema. U praktičnom dijelu problem predviđanja mogućnosti srčanog zastoja na temelju medicinskih podataka bit će riješen koristeći 3 metode: KNN (K nearest neighbours, K najbližih susjeda), SVM (Support vector machine, metoda potpornih vektora), te Decision tree classifier (klasifikacijsko stablo odlučivanja), te će rezultati koje je svaka od metoda postigla biti uspoređeni kako bi se došlo do najbolje metode za ovaj tip problema.

1.2. Metode i tehnike rada

Kao osnova programskog rješenja korišten je programski jezik Python, a kod je pisan u formatu Jupyter Notebook bilježnice. Korištene su biblioteke NumPy, Pandas, MatplotLib i ScikitLearn. Kod je vlastita izrada autora uz pomoć internetskih izvora. Za teorijski dio korišteni su internetski izvori, kao što su članci, istraživački radovi i blogovi koji su pronađeni s pomoću pretraživača Google Scholar. Za referenciranje izvora korišten je alat Zotero. Programski kod verzioniran je uz pomoć sustava GitHub, te je dostupan na poveznici.

2. Strojno učenje i medicina

S poboljšanjima i razvojem strojnog učenja unatrag nekoliko godina, došlo je i do povećanja njegove primjene na području medicine. Prema podacima iz 2022. godine, oko 86% zdravstvenih organizacija koristi rješenja strojnog učenja u nekom obliku [1]. Primarno se to odnosi na potpornu ulogu takvih algoritama koji služe kako bi pomogli stručnjacima u boljem i lakšem obavljanju svojih poslova, identificiranju trendova, predviđanju bolesti, ali i na organizacijskom spektru kao pomoć u organizaciji elektroničkih medicinskih podataka[2]. Strojno učenje svoju primjenu pronašlo je i u identifikaciji genetskih sekvenci SARS-CoV2 i stvaranju cjepiva [2].

Neki od najčešćih primjera primjene strojnog učenja u medicini jesu modeli predviđanja bolesti i njihovih ishoda. Na primjer, modelu je moguće kao ulaz dati skup podataka s raznim podacima o pacijentu kao što su krvni tlak, postojanje ovisnosti o cigaretama i povijest bolesti u obitelji, te model na temelju tih podataka predviđa mogućnost da pacijent doživi srčani udar, ili predviđa vjerojatnost smrtnog ishoda srčanog udara. Osim ovog, postoje i drugi poznati klasifikacijski problemi, kao na primjer prepoznavanje i klasifikacija malignih i dobroćudnih tumora s rendgenskih skenova pluća ili mozga pacijenta. Model je moguće istrenirati za automatsku interpretaciju EKG-ova i raznih drugih standardiziranih testova koji se koriste u medicinskoj praksi [3]. Naravno, visoko trenirani medicinski stručnjaci dovoljno su sposobni i sami izvoditi navedene zadatke, ali korištenje modela strojnog učenja uvelike ubrzava i olakšava te poslove, te također pruža drugu perspektivu ako stručnjak nije u potpunosti siguran u vezi interpretacije određenog testa.

Unatoč mnogim poboljšanjima i napretku unutar područja strojnog učenja, koje se dogodilo u sklopu opće informatizacije društva, i u današnje vrijeme aktualni su mnogi problemi primjene strojnog učenja u medicini. Najveći od tih problema jest problem kvalitete podataka. Model koji kreiramo dobar je onoliko koliko su dobri podaci koje mu dajemo, te je samim time za najbolje rezultate potrebno imati velike količine podataka visoke kvalitete [1]. Samim time, pojavljuje se potreba da se u medicinskim ustanovama počne uvoditi upravljanje podacima visoke razine, što sa sobom donosi velike troškove te zahtijeva zapošljavanje stručnjaka, uvođenje novih i modernih sustava te obuku djelatnika. U većim i bogatijim državama to ne bi predstavljalo prevelik problem, no u zemljama nižeg ili srednjeg socioekonomskog statusa, kao što je na primjer Hrvatska, uz dobro poznate probleme sa zastarjelom birokracijom koji su prisutni u Lijepoj Našoj, dolazimo do velike zapreke u ostvarivanju ovog zahtjeva. Zbog činjenice da se radi o vrlo osjetljivim i bitnim podacima, te da greška modela koji se razvija u zdravstvene svrhe potencijalno može dovesti do smrti pacijenta, ovo predstavlja ključan zahtjev za koji ne smije postojati kompromis. Također, ako se ne radi o modelu stabla

odlučivanja, sustav koji predviđa nije u mogućnosti predstaviti razlog određenog predviđanja, što može dovesti do problema u slučaju potencijalne greške, pogotovo ako se u obzir uzmu kompleksne pravne procedure u nekim zemljama [1].

U kompleksnom području kao što je zdravstvo postoje razne vrste svakodnevnih problema i zadataka za koje se mogu koristiti različiti pristupi iz područja strojnog učenja. Primarno se ti pristupi dijele na nadzirano, nenadzirano i potporno učenje, a svaki od pristupa ima različito područje primjene. U nastavku ćemo pobliže pogledati nenadzirano i nadzirano učenje, te vrste problema koje se mogu rješavati uz pomoć tih pristupa.

2.1. Nenadzirano strojno učenje

Prvi oblik koji će biti prikazan u ovom radu jest nenadzirano strojno učenje. Budući da se ovaj rad u praktičnom dijelu fokusira na metode nadziranog strojnog učenja, te da se nenadzirano strojno učenje, manje primjenjuje u medicini, opisat ćemo ga nešto sažetije.

Nenadzirano strojno učenje karakterizira učenje nad skupom podataka bez jasno naznačene ciljne, odnosno zavisne varijable. Stroj uči uzorke u podacima te grupira podatke, bez predviđanja nekakvog ishoda [4]. Korisnost ovakvih modela jest u mogućnosti pronalaženja uzoraka i grupacija unutar podataka koje čovjek možda ne bi zapazio. Također se koriste za redukciju visokodimenzionalnih skupova podataka te ih svode na čitljiviji oblik [5].

2.1.1. Podjela algoritama nenadziranog strojnog učenja

Algoritmi nenadziranog strojnog učenja dijele se na algoritme klasteriranja i informativne transformacije podataka [4].

Algoritam klasteriranja će izračunati mjeru udaljenosti kako bi kvantificirao sličnost ili različitost između različitih subjekata u skupu podataka. Na temelju ove mjere, subjekti će biti klasterizirani (grupirani) ili odvojeni jedni od drugih kako bi se dobili klasteri (skupine) koji imaju najveću sličnost unutar klastera i najveće razlike između klastera [4]. Ovdje postoji potreba za oprezom kada se ovi algoritmi koriste kao potpora za donošenje odluka, budući da ovakvi algoritmi često znaju prenaglasiti sličnosti unutar skupa podataka [1]. Kako bi razumijevanje ovakvog modela bilo jasnije, pogledat ćemo primjer. Imamo fotografiju posude s voćem, algoritam klasteriranja pokušat će prepoznati karakteristike pojedinih voćaka s fotografije te grupirati voće na temelju istog izgleda, odnosno iste vrste. Rezultat je moguće vidjeti na slici ispod. Primjetno kako model nije označio različite predmete sa slike, već ih je samo grupirao na temelju vizualnih značajki.

Slika 1: primjer klasteriranja voća na slici, slika preuzeta sa iStock.com te modificirana od strane autora rada

Informativne transformacije podataka predstavljaju smislene transformacije objekata visokih dimenzija podataka u objekte nižih dimenzija tako da se sačuvaju relevantne informacije [4]. Smanjivanjem broja dimenzija skup podataka se pretvara u mnogo jasniji i čitljiviji oblik, kako za čovjeka tako i za računalo.

2.1.2. Primjena nenadziranog učenja u medicini

Kao što je rečeno, primjena ovakvih modela uglavnom se svodi na prepoznavanje uzoraka i grupiranje podataka. Jedan takav primjer je istraživanje ozljeda kostiju među trkačima. Uz pomoć hijerarhijskog klasteriranja identificirane su grupe sportaša sa sličnom kinematikom zgloba, te su različite grupe koje su kreirane imale su i različite rizike ozljeda [5]. Ovakva spoznaja može pomoći medicinskim stručnjacima da identificiraju načine na koje bi se ozljede sportaša mogle spriječiti, te se time može pružiti bolja skrb na individualnoj razini.

2.2. Nadzirano strojno učenje

Kada se govori o umjetnoj inteligenciji, ili specifičnije o strojnom učenju, najčešći primjeri na koje se nalazi jesu primjeri nadziranog strojnog učenja. **Nadzirano strojno učenje** predstavlja vrstu strojnog učenja u kojoj različiti algoritmi generiraju matematičku funkciju koja preslikava dane ulaze u izlaze [6]. Klasičan i najlakši primjer ovoga bio bi kreiranje jednostavne linearne matematičke funkcije kojoj kao ulaz dajemo veličinu kuće u obliku površine izražene u m^2, a kao izlaz dobivamo predviđenu cijenu kuće. U praksi se ulaz najčešće označava slovom x, a izlaz slovom y [6], tako da jednostavnije možemo reći da je strojno učenje skup funkcija koje pretvaraju x u y.

x još nazivamo i nezavisnom varijablom, jer mijenjanjem drugih varijabli ne utječemo na tu varijablu, dok y nazivamo zavisnom varijablom, jer promjene na drugim varijablama uzrokuju promjenu na toj varijabli. U primjeru površine kuće i cijene, površina kuće je nezavisna varijabla, dok je cijena zavisna varijabla, jer promjenom površine kuće dolazi i do promjene cijene kuće, na primjer povećanjem površine kuće cijena kuće će vjerojatno porasti.

2.2.1. Podjela algoritama nadziranog učenja

Algoritmi nadziranog učenja dijele se na: klasifikacijske i regresijske algoritme, ovisno o tome radi li se o predviđanju kvantitativne ili kategoričke varijable [1]. Ako se vratimo na prethodni primjer o predviđanju cijene kuće, cijena u tom primjeru predstavlja kvantitativnu varijablu, odnosno varijablu koja može imati različite vrijednosti u nekom razumnom rasponu, te se samim time za takav tip problema koriste regresijski algoritmi.

S druge strane, ponovno se vračajući na raniji primjer, u slučaju problema prepoznavanja vrste tumora s rendgenske snimke radi se o problemu klasifikacije, jer se ciljna varijabla dijeli na kategorije, u ovom slučaju dvije: tumor je ili zloćudan ili dobroćudan. Drugi primjer bio bi analiza fotografije automobila u svrhu određivanja marke. U ovom slučaju broj kategorija je veći, ovisno o broju različitih marki automobila koje su prisutne u skupu podataka. Samim time, klasifikacijske algoritme možemo podijeliti na binarne (2 klase) i viševarijantne (3 ili više klasa).

2.2.2. Proces učenja

Proces učenja većine modela strojnog učenja dijeli se na 2 faze: treniranje i testiranje. Prva je faza treniranje, gdje model uči na temelju skupa podataka kojim smo ga „nahranili“. U kontekstu nadziranog učenja modelu koji smo napravili dajemo podatke s oznakama, odnosno podatke koji sadrže skup nezavisnih varijabli te zavisnu varijablu. Model uočava i pamti veze

između određenih vrijednosti nezavisnih varijabli i konačne vrijednosti zavisne varijable, te kako mijenjanje nezavisnih varijabli utječe na zavisnu, te se time izgrađuje model [6].

U fazi testiranja modelu se daje novi skup podataka koji model nikad nije vidio, te su ti podaci bez oznaka, odnosno modelu se daju samo nezavisne varijable. Zadatak modela je na temelju svega naučenog iz faze treniranja koristeći nove podatke s nezavisnim varijablama predvidjeti vrijednost zavisne varijable. Predviđanja modela zatim se uspoređuju sa stvarnim vrijednostima nezavisne varijable, te se na temelju metrika kao što su preciznost ili točnost određuje kvaliteta modela te njegova uspješnost u obavljanju danog zadatka.

2.2.3. Primjena nadziranog učenja u medicini

Velike količine podataka koje su prisutne za većinu pacijenata čine područje medicine plodnim tlom za primjenu nadziranog strojnog učenja u raznim zadacima. Kao što je ranije spominjano, uglavnom se radi o primjerima korištenja ovakvih algoritama za automatizaciju i klasifikaciju kojom se olakšava posao medicinskih stručnjaka [7]. Postoje razni primarno klasifikacijski zadaci u medicini na kojima se može primijeniti nekakav model nadziranog strojnog učenja.

U praksi se to primarno vidi na području klasifikacije slika. Primarno se to očituje u hematologiji, disciplini koja se bavi proučavanjem krvi i krvotvornih organa, bolestima krvnih stanica i njihovim liječenjem [8]. U hematološkoj praksi česta je upotreba modela koji su trenirani na velikim količinama slika te su u mogućnosti raspoznavati tipove i zdravlje krvnih stanica [7]. Primjer je sustav CellaVision DM1200, automatizirani sustav za digitalnu morfologiju stanica [9]. Ovo dakako nije jedini takav uređaj na tržištu, što dodatno pokazuje veliku primjenu ovakvih strojeva u hematološkoj praksi, čime se uvelike ubrzava postupak dijagnosticiranja pacijenta, a uz to se i smanjuje potreba za ljudskim radom.

Bitno je još jednom naglasiti da ovi sustavi služe kao pomoć medicinskom osoblju, ne kao njihova zamjena. Koliko god model bio precizan, on služi samo kao filter. Na hematološkom primjeru model će slike klasificirati kao zdrave ili potencijalno nezdrave, te zatim potencijalno nezdravi primjerci idu na dodatni pregled ljudskog osoblja koje zatim utvrđuje stvarnu dijagnozu. Još jedan primjer bio bi predviđanje mogućnosti srčanog udara. Naglasak je na riječi mogućnosti, jer model samo pretpostavlja da osoba ima predispozicije za srčani udar, te na neki način usmjerava medicinskog stručnjaka na osobe koje bi trebalo dodatno pregledati, staviti pod nadzor ili savjetovati kako bi se potencijalni srčani udar spriječio.

Postoji još mnogo potencijalnih primjera korištenja ovakvih modela, od klasifikacije tumora, predviđanja očekivane životne dobi i drugih. No kako se ne bismo doveli do zabune

da je sve u ovom slučaju med i mlijeko, postoje razni problemi primjene ovakvih modela koje ćemo proučiti u nastavku.

2.3. Etički aspekt

Jedan od najvećih problema koji se suprotstavljaju korištenju strojnog učenja u medicini jest problem etičnosti korištenja takvih algoritama u medicinskoj praksi. Neki od najčešće spominjanih problema su problem ljudske privatnosti te sigurnosti i povjerljivosti podataka [10]. Stvarajući nove modele koji svoju primjenu nalaze u medicini, stručnjaci danas više nego ikad trebaju obraćati pozornost na ove i druge probleme etičnosti. Za vodilju u tom smjeru većina stručnjaka smatra 4 načela bioetike:

1. Autonomnost - načelo koje zahtijeva poštivanje sposobnosti donošenja odluka samostojnih osoba
2. Neškodljivost - načelo koje zahtijeva da se drugima ne nanosi zlo
3. Dobročinstvo - skup načela koja zahtijevaju da spriječimo štetu, omogućimo dobrobit i odmjerimo dobrobit u odnosu na opasnosti i cijenu
4. Pravednost - skup načela koja zahtijevaju ravnomjernu i poštenu raspodjelu dobrobiti, opasnosti i cijene [11]

Rasheed i suradnici [10] u svojem radu podijelili su etičke probleme primjene strojnog učenja na području medicine u 4 kategorije:

1. Problemi vezani uz podatke
2. Privatnost
3. Informirani pristanak
4. Etika njege

U nastavku ćemo ukratko proučiti svaku, a zatim i istražiti mišljenje struke o ovoj temi.

2.3.1. Problemi vezani uz podatke

Kao što je ranije spomenuto, bilo koji model dobar je onoliko koliko su dobri podaci s kojima smo ga „nahranili“. Budući da iza svih tih podataka stoje ljudi, modeli strojnog učenja u suštini imitiraju ljudsko ponašanje i upravo tu dolazi do ključnog problema vezanog uz podatke. Ljudi nisu strojevi, samim time čak su i najveći stručnjaci podložni greškama, te se vrlo lako mogu nesvjesno potkrasti i neke neetične pogreške, nepravedni zaključci, diskriminativni stavovi i slično, što se zatim preslikava na model koji onda u svojem radu replicira ponašanje ljudi.

Čest je problem nebalansiranost podataka koja nastaje zbog neuniformne distribucije primjeraka različitih klasa koje imamo u podacima [10]. Posljedica toga je sklonost modela prema određenim klasama, što može imati katastrofalne posljedice kada su u pitanju predviđanja modela koja mogu uzrokovati smrt pacijenta. Razlog velike raširenosti ovog problema jest činjenica da se većina modela razvija u zapadnim zemljama, te su ti modeli trenirani na nereprezentativnoj populaciji.

Radi boljeg shvaćanja uzet ćemo logičan primjer. AI tim neke kompanije sa sjedištem u New Yorku ima zadatak razviti prediktivni model koji će na temelju značajki predvidjeti mogućnost srčanog udara za pacijenta. Kako bi istrenirali model, istraživači su od jedne od većih bolnica u gradu pribavili skup podataka koji se na prvi pogled činio vrlo reprezentativnim, s dobrom raspodjelom spola, dobi i drugih značajki. Model je nakon treniranja testiran uz nadzor medicinskih stručnjaka te je utvrđena visoka preciznost. Problem se otkriva ako promotrimo da pristup kvalitetnoj zdravstvenoj njezi u SAD-u zbog manjkavosti tamošnjeg zdravstvenog sustava ovisi o socioekonomskom statusu pojedinca [12], što znači da postoje velike šanse da model ne bi bio sposoban provesti kvalitetnu generalizaciju na široj populaciji. Model treniran na pripadnicima srednje i više Američke klase teško će pokazivati dobre rezultate nad populacijom Ugande ili Ujedinjenih Arapskih Emirata.

2.3.2. Privatnost

Medicinski podaci spadaju u kategoriju najosjetljivijih podataka [10], te u skladu s time postoji potreba da se zaštite ti podaci, a samim time i pacijenti. U kontekstu zdravstva, privatnost predstavlja čuvanje pacijentovih podataka od neovlaštenog pristupa [5]. Algoritmi strojnog učenja trebaju velike količine podataka sa što više značajki, stoga dolazi do opasnosti da se za neki model iskoriste vrlo osjetljivi podaci. Ovo je posebno problematično kod pokušaja razvoja modela koji identificiraju postojanje mentalnih problema. Također, zbog prirode procesa razvoja modela podaci koji su namijenjeni isključivo osobnom liječniku postaju dostupni potencijalno desecima drugih ljudi koji rade na razvoju modela, čime se narušava pacijentova privatnost.

2.3.3. Informirani pristanak

Budući da se radi o vrlo povjerljivim i osjetljivim privatnim podacima pacijenata, potrebno je dobiti informirani pristanak za bilo kakvu uporabu tih podataka. To znači da bi pacijenta trebalo detaljno informirati o kojim je podacima riječ, tko će ih koristiti i u koje svrhe. Ovakav tip prakse već postoji u medicinskoj praksi prije bilo kakve medicinske intervencije [10]. Problem u ovom slučaju predstavlja „black-box“ priroda modela strojnog učenja, gdje je za medicinskog stručnjaka i pacijenta teško otkriti zašto je model dao određene rezultate. Kao

odgovor na to Europski GDPR uveo je pravila oko odluka i metoda podatkovnih pristupa. Prema tome pojedinac ima pravo razumjeti zašto je model donio odluku koju je donio [10].

2.3.4. Etika njege

Posljednji aspekt odnosi se na uklanjanje itekako potrebnih međuljudskih odnosa na relaciji medicinski stručnjak-pacijent, što bi se dogodilo eventualnom zamjenom ljudskih radnika u korist umjetne inteligencije. Također, prema Rasheedu i suradnicima [10], postoje negodovanja unutar medicinske struke u vezi smanjivanja sigurnosti zadržavanja posla, koje potencijalno dolazi s većim uvođenjem algoritama strojnog učenja u medicinskoj praksi.

3. Implementacija

Slijedi dio rada posvećen opisu i pregledu implementacije praktičnog dijela rada. Ova sekcije podijeljena je na 6 dijelova. U prvom dijelu predstavljena je problemska domena praktičnog dijela. Zatim su opisane metode i tehnike korištene za pretprocesiranje podataka kako bi se iz algoritama izvukle najbolje performanse. Zatim je zasebno opisana teorija i implementacija svakog od izabranih algoritama. Na kraju su predstavljeni i uspoređeni rezultati praktičnog dijela rada.

3.1. Problemska domena

Sa sustava Kaggle.com preuzet je skup podataka o kliničkim zapisima srčanih zastoja [13], koji je autor preuzeo iz znanstvenog istraživanja Chiccoa i Jurmana [14]. Radi se o skupu podataka u kojem se nalaze klinički podaci o pacijentima koji su doživjeli srčani zastoj, uključujući zavisnu varijablu smrtnog slučaja u određenom periodu nakon medicinske intervencije. Cilj je na temelju značajki predvidjeti postoji li vjerojatnost da bi pacijent mogao umrijeti u narednom periodu nakon intervencije radi srčanog zastoja. Problem je vrlo aktualan, budući da je broj osoba s rizikom od bilo kakve kardiovaskularne bolesti godinama u stalnom porastu [15]. Skup se sastoji od 300 uzoraka (pacijenata), te za svakog postoji 13 značajki. Napomena da su u stvarnom skupu podataka vrijednosti koje su ovdje navedene kao tip podataka boolean zapravo brojevi 0 i 1, te ih je radi lakšeg razumijevanja autor ovdje odlučio svrstati pod tip podataka boolean. Značajke u skupu su [14]:

1. Dob – dob pacijenta, tip podataka float
2. Anaemia (anemija) – postojanje anemije, smanjene količine hemoglobina kod pacijenta, tip podataka boolean
3. Creatinine phosphokinase (Kreatinin fosfokinaza) – razina CPK enzima u krvi, tip podataka integer
4. Diabetes – postojanje dijabetesa kod pacijenta, tip podataka boolean
5. Ejection fraction (istisna frakcija) – postotak krvi koji izlazi iz srca svakim otkucajem, tip podataka integer
6. High blood pressure (visok krvni tlak) – postojanje visokog krvnog tlaka kod pacijenta, tip podataka boolean
7. Platelets (trombociti) – količina trombocita u krvi, tip podataka float
8. Serum creatinine – tip podataka float
9. Serum sodium – tip podataka integer

10. Sex (spol) – spol pacijenta, kategorički tip podataka, vrijednost 0 predstavlja žensko, vrijednost 1 predstavlja muško
11. Smoking (pušenje) – je li pacijent pušač, tip podataka boolean
12. Time (vrijeme) – popratni period nakon srčanog zastoja, tip podataka integer
13. Death event (smrtni događaj) – je li došlo do smrti pacijenta u popratnom razdoblju, tip podataka boolean

Na slikama ispod prikazane su tablice preuzete iz ranije spomenutog rada Chiccoa i Jurmana [14], koje prikazuju opise značajki, mjerne jedinice te raspone u prvoj tablici, te statistički kvantitativni opis kategoričkih značajki u drugoj tablici.

Feature	Explanation	Measurement	Range
Age	Age of the patient	Years	[40, ..., 95]
Anaemia	Decrease of red blood cells or hemoglobin	Boolean	0, 1
High blood pressure	If a patient has hypertension	Boolean	0, 1
Creatinine phosphokinase (CPK)	Level of the CPK enzyme in the blood	mcg/L	[23, ..., 7861]
Diabetes	If the patient has diabetes	Boolean	0, 1
Ejection fraction	Percentage of blood leaving the heart at each contraction	Percentage	[14, ..., 80]
Sex	Woman or man	Binary	0, 1
Platelets	Platelets in the blood	kiloplatelets/mL	[25.01, ..., 850.00]
Serum creatinine	Level of creatinine in the blood	mg/dL	[0.50, ..., 9.40]
Serum sodium	Level of sodium in the blood	mEq/L	[114, ..., 148]
Smoking	If the patient smokes	Boolean	0, 1
Time	Follow-up period	Days	[4,...,285]
(target) death event	If the patient died during the follow-up period	Boolean	0, 1

mcg/L: micrograms per liter. mL: microliter. mEq/L: milliequivalents per litre

Slika 2: tablica opisa, mjernih jedinica te raspona vrijednosti značajki

	Full sample		Dead patients		Survived patients	
Category feature	#	%	#	%	#	%
Anaemia (0: false)	170	56.86	50	52.08	120	59.11
Anaemia (1: true)	129	43.14	46	47.92	3	40.89
High blood pressure (0: false)	194	64.88	57	59.38	137	67.49
High blood pressure (1: true)	105	35.12	39	40.62	66	32.51
Diabetes (0: false)	174	58.19	56	58.33	118	58.13
Diabetes (1: true)	125	41.81	40	41.67	85	41.87
Sex (0: woman)	105	35.12	34	35.42	71	34.98
Sex (1: man)	194	64.88	62	64.58	132	65.02
Smoking (0: false)	203	67.89	66	68.75	137	67.49
Smoking (1: true)	96	32.11	30	31.25	66	32.51

#: number of patients. %: percentage of patients. Full sample: 299 individuals. Dead patients: 96 individuals. Survived patients: 203 individuals.

Slika 3: kvantitativni opis kategoričkih značajki

Cilj praktičnog dijela rada bio je razviti tri različita modela temeljena na 3 različita algoritma, te koristeći metode i postupke karakteristične za strojno učenje izvući najbolje performanse iz svakog od modela te usporedbom performansi doći do zaključaka oko odabira najkvalitetnijeg modela te općenite uporabljivosti modela u svakodnevnom životu.

3.2. Analiza i predprocesiranje podataka

3.2.1. Eksplorativna analiza podataka

Prvi korak praktičnog dijela jest eksplorativna analiza podataka [16]. U tom koraku autor se crtanjem grafova i korištenjem raznih deskriptivnih statističkih metoda upoznaje s podacima i nekim njihovim karakteristikama kako bi bolje razumio podatke te na temelju toga mogao provesti pretprocesiranje podataka, koje je ključno za dobar rad algoritama strojnog učenja.

U ovom slučaju, autor je prvo koristio metode *info* i *describe* [16]. Metoda *info* za rezultat vraća tablicu sa svim stupcima iz skupa podataka te pripadajuće brojeve Non-Null vrijednosti i tipove podataka. U korištenom skupu podataka nalazi se 12 stupaca, odnosno značajki, te svaka značajka broji 299 vrijednosti, što odmah ukazuje na činjenicu da u skupu podataka ne postoje NaN vrijednosti. Što se tiče tipova podataka, 3 su značajke sa skupom podataka float64: age, platelets i serum_creatinine, dok su ostale značajka tipa podataka int64. Metoda *describe* vraća tablicu s deskriptivnom statistikom, koja daje uvid u npr. najveće i najmanje vrijednosti, mean, standardnu devijaciju i kvartile za svaku značajku. Rezultati ovih metoda prikazani su na slikama ispod.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   age                       299 non-null    float64
 1   anaemia                   299 non-null    int64
 2   creatinine_phosphokinase  299 non-null    int64
 3   diabetes                  299 non-null    int64
 4   ejection_fraction         299 non-null    int64
 5   high_blood_pressure       299 non-null    int64
 6   platelets                 299 non-null    float64
 7   serum_creatinine          299 non-null    float64
 8   serum_sodium              299 non-null    int64
 9   sex                       299 non-null    int64
 10  smoking                   299 non-null    int64
 11  time                      299 non-null    int64
 12  DEATH_EVENT               299 non-null    int64
dtypes: float64(3), int64(10)
memory usage: 30.5 KB
```

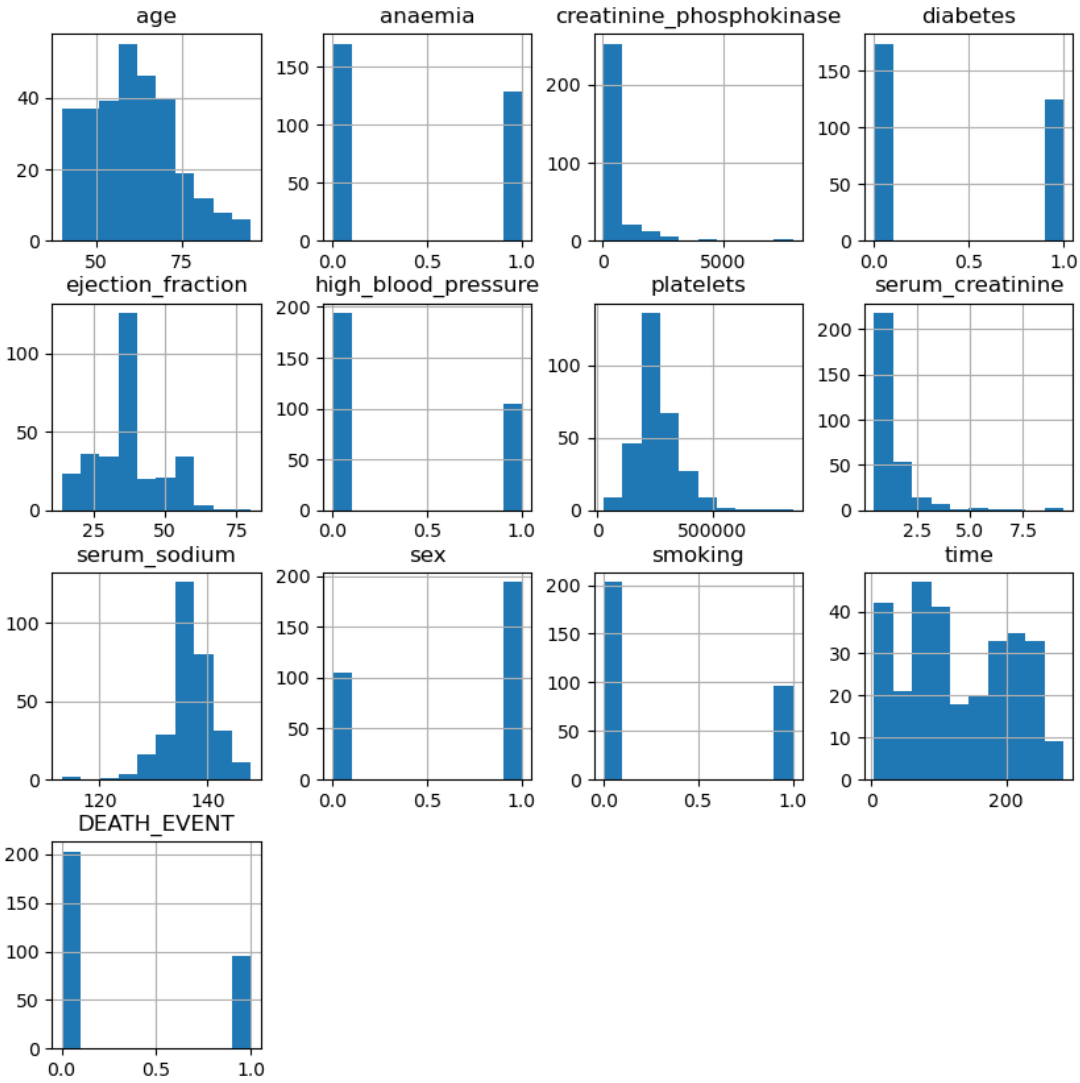
Slika 4: prikaz ispisa metode info (vlastita izrada)

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.00000	299.000000	299.000000	299.00000	299.000000	299.00000
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.351171	263358.029264	1.39388	136.625418	0.648829	0.32107	130.260870	0.32107
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.478136	97804.236869	1.03451	4.412477	0.478136	0.46767	77.614208	0.46767
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	25100.000000	0.50000	113.000000	0.000000	0.00000	4.000000	0.00000
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000000	212500.000000	0.90000	134.000000	0.000000	0.00000	73.000000	0.00000
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000000	262000.000000	1.10000	137.000000	1.000000	0.00000	115.000000	0.00000
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	303500.000000	1.40000	140.000000	1.000000	1.00000	203.000000	1.00000
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.000000	850000.000000	9.40000	148.000000	1.000000	1.00000	285.000000	1.00000

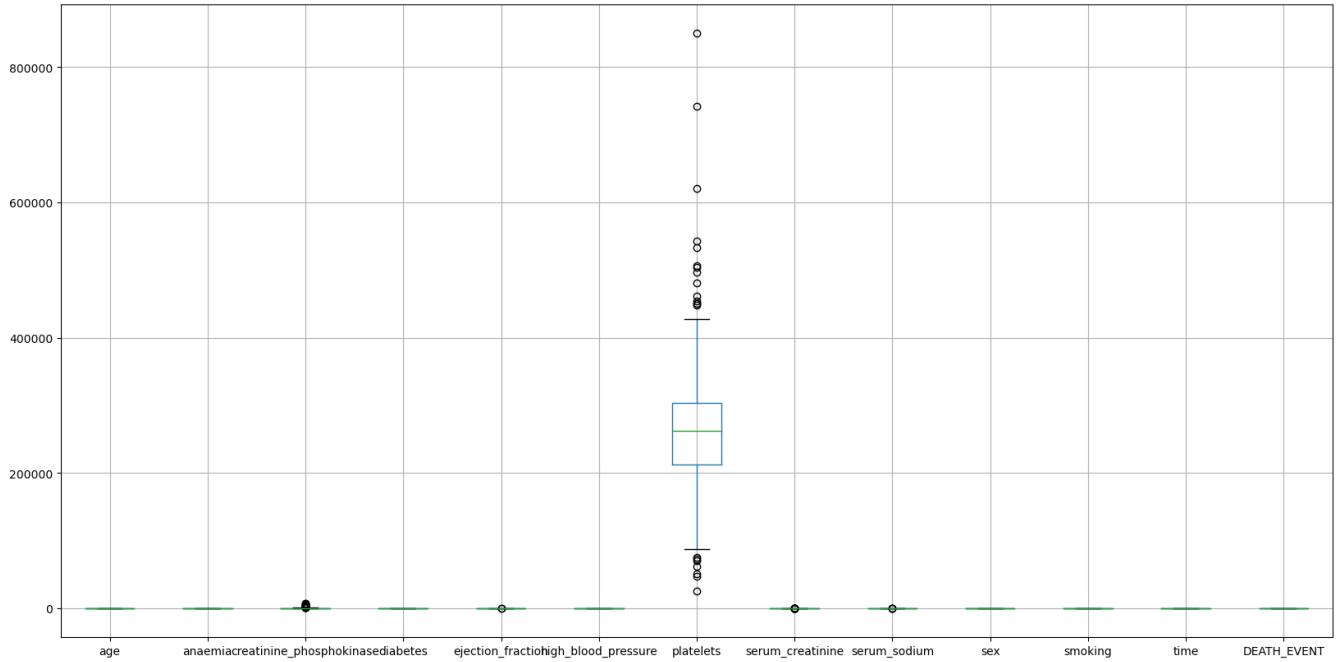
Slika 5: prikaz ispisa metode describe (vlastita izrada)

Idući korak u analizi podataka podrazumijeva crtanje histograma, kako bi se dobio uvid u distribuciju podataka. Histogram je najčešće korišteni graf za prikaz distribucije frekvencije [17]. S histograma je vidljivo da značajke creatinine_phosphokinase i serum_creatinine imaju problem desne iskošenosti, kao što je vidljivo na slici 6 ispod.

Još jedan koristan graf koji se često koristi u praksi jest boxplot, što je graf koji prikazuje distribuciju podataka, uključujući i stršila, tako da se „kutija“ rasprostire od prvog do trećeg kvartila, te brkovi koji predstavljaju minimalnu i maksimalnu vrijednost, a sve izvan toga su eventualna stršila [18]. Na prikazanom boxplot-u (slika 7) ispod vidljivo je da značajka platelets ima velik broj stršila.

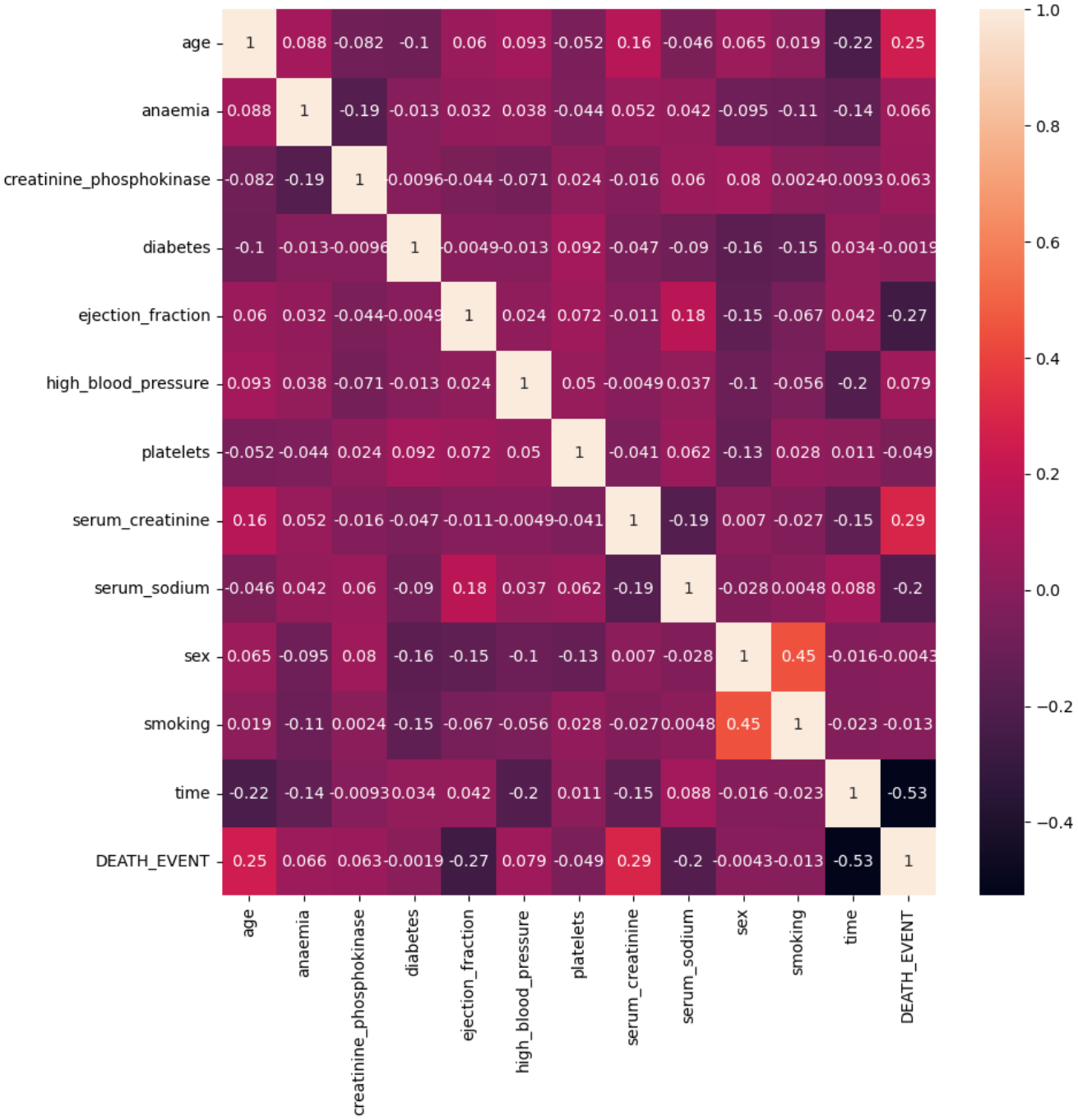


Slika 6: histogrami (vlastita izrada)



Slika 7: boxplot-ovi (vlastita izrada)

Vrlo korisno je i pogledati graf korelacija između pojedinih značajki kako bismo utvrdili koje značajke najviše utječu na ciljnu varijablu.



Slika 8: graf korelacija (vlastita izrada)

Vrijednosti korelacije koje su blizu nuli možemo odmah otpisati, jer to znači da varijabla ne utječe na ciljnu varijablu. Za vrijednosti bliže 1 ili -1 kaže se da imaju pozitivnu korelaciju u slučaju 1, odnosno negativnu korelaciju u slučaju -1. Ako se pogleda korelacija varijabli s ciljnom varijablom DEATH_EVENT, zaključak je da varijabla time ima dosta jaku negativnu

korelaciju, dok varijable age i serum_creatinine imaju relativno slabu pozitivnu korelaciju. Ostale varijable ne pokazuju nikakvu značajnu razinu korelacije s ciljnom varijablom.

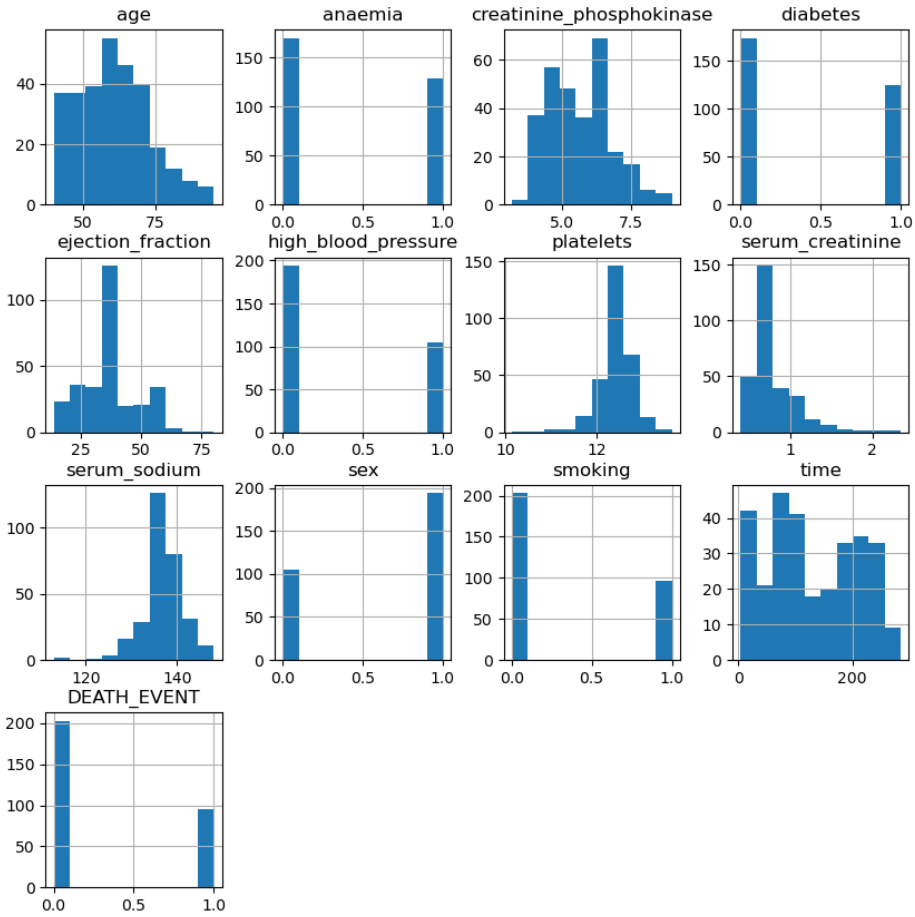
3.2.2. Pretprocesiranje podataka

Kako bi algoritmi mogli kvalitetno učiti i kreirati predviđanja iz podataka, podaci trebaju biti pretprocesirani kako bi bili što čitljiviji za algoritme. Pretprocesiranje u ovom slučaju izvedeno je nad varijablama za koje je u analizi podataka utvrđeno da imaju nekakvu manu, bilo to stršila ili problem iskošenosti. Radi se o varijablama platelets, creatinine_phosphokinase i serum_creatinine. Pretprocesiranje se u ovom slučaju svodi na izračunavanje logaritma za svaku vrijednost koja pripada tim varijablama, te povećanje za 1 na kraju kako bi se izbjegle nule [19]. U kodu to izgleda kako slijedi:

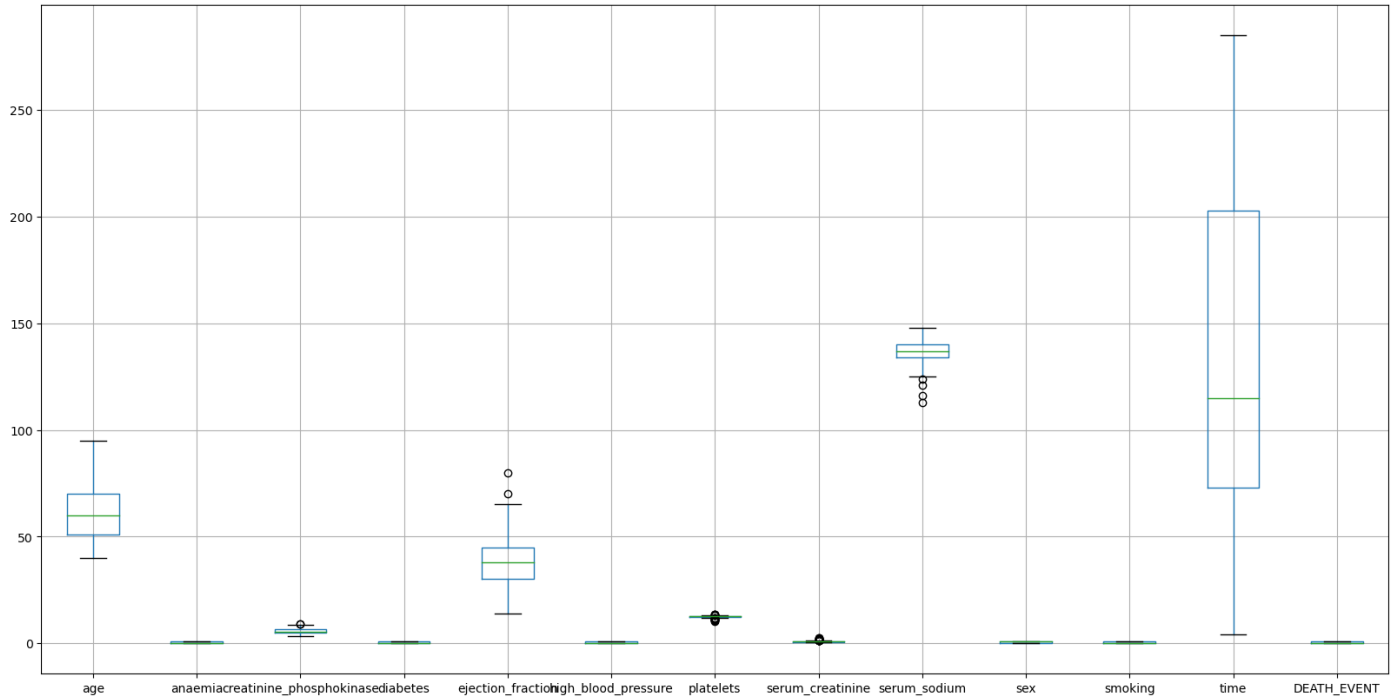
```
df['creatinine_phosphokinase'] = np.log(df['creatinine_phosphokinase'] + 1)
df['serum_creatinine'] = np.log(df['serum_creatinine'] + 1)
df['platelets'] = np.log(df['platelets'] + 1)
```

Programski kod 1: uklanjanje stršila

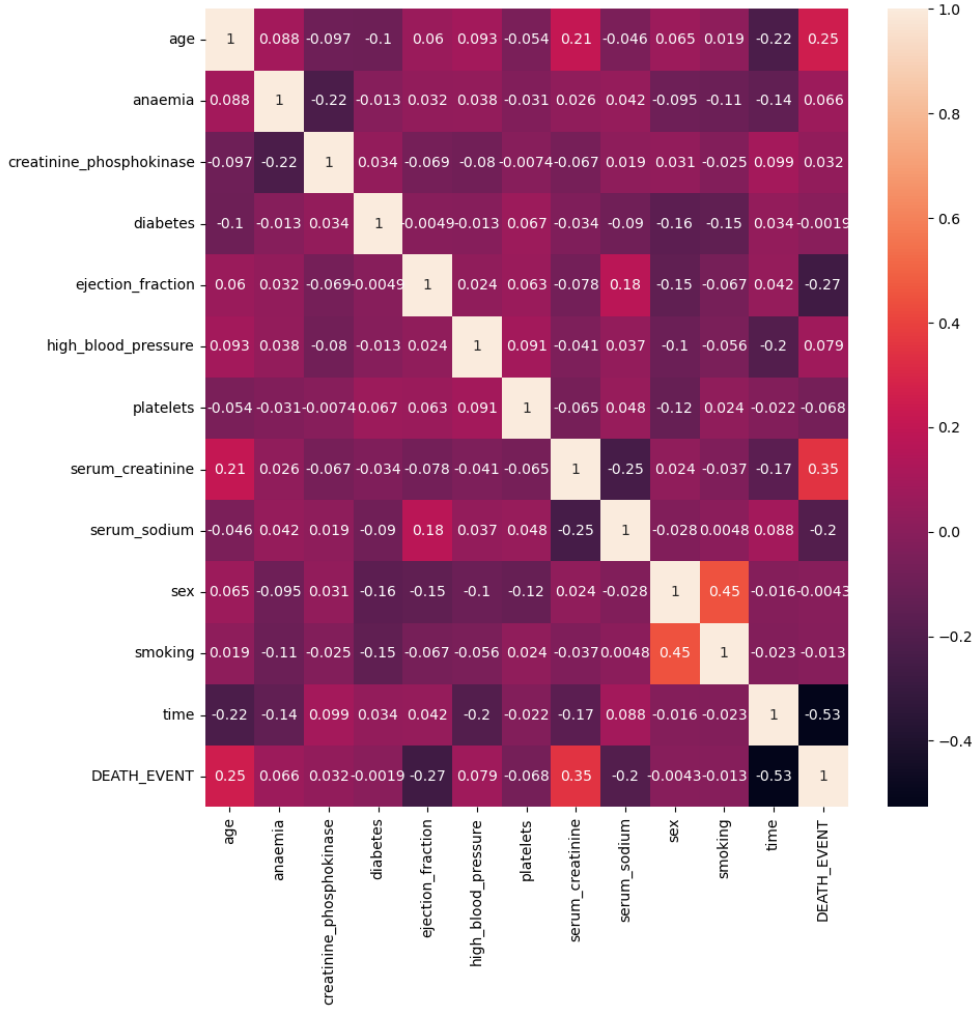
Ovim postupkom riješeni su problemi ovih varijabli, što je moguće vidjeti sa slika grafova spominjanih u poglavlju za analizu podataka crtanih nakon procesiranja.



Slika 9: histogrami nakon predprocesiranja podataka (vlastita izrada)



Slika 10: boxplot-ovi nakon predprocesiranja podataka (vlastita izrada)



Slika 11: dijagram korelacija nakon predprocesiranja podataka (vlastita izrada)

Kao što je vidljivo sa slika 9, 10 i 11, došlo je do izglađivanja distribucije kod varijabli creatinine_phosphokinase i serum_creatinine koje su imale problem iskošenosti, te su kod varijable platelets uklonjena stršila. Također, došlo je i do promjene na grafu korelacija, specifično za varijablu serum_creatinine, koja sad ima korelaciju od 0.35, u odnosu na prethodnih 0.29.

3.3. Podjela skupa podataka

Za treniranje modela potrebno je skup podataka podijeliti na skup za treniranje, na kojem će model učiti veze i ishode između podataka, te skup za testiranje, uz pomoć kojeg će model na do sad neviđenim podacima raditi predviđanja koja se zatim uspoređuju sa stvarnim vrijednostima. Iz toga se na temelju performansi modela određuje njegova kvaliteta.

Skup podataka prvo je podijeljen na x i y skup, gdje x predstavlja skup podataka koji sadrži sve nezavisne varijable, dok se skup y sastoji od ciljne, odnosno zavisne varijable, što je u ovom slučaju DEATH_EVENT.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=101, test_size=0.25, stratify=y)
```

Programski kod 2: podjela skupa podataka na skup za treniranje i testiranje

Podjela podataka na skup za treniranje i testiranje odrađena je uz pomoć *train_test_split* metode iz biblioteke ScikitLearn. Kreiraju se 4 nova skupa: x_train, x_test, y_train, y_test. X_train je skup podataka za treniranje koji se sastoji od nezavisnih varijabli, y_train je skup podataka za treniranje koji sadrži zavisnu varijablu, x_test je skup podataka za testiranje koji se sastoji od nezavisnih varijabli, a y_test skup podataka za testiranje koji sadrži zavisnu varijablu. Argument test_size određuje koliko će se podataka alocirati za treniranje, što je u ovom slučaju 25%. Random_state argument osigurava reproduktivnost postignutih rezultata, a argument stratify osigurava da distribucija proporcije klasa bude jednaka u oba skupa.

224 primjerka alocirana su skupu za treniranje, a ostalih 75 skupu za testiranje. U svakom od skupova 67% primjeraka pripada klasi 0, a 33% klasi 1.

3.4. Stablo odlučivanja

3.4.1. Teorijski opis stabla odlučivanja

Algoritmi stabla odlučivanja temelje se na rekurzivnim particijama podataka prema nekom kriteriju. Stablo odlučivanja sastoji se od korijenskog, unutarnjeg i lisnog čvora (root, internal and leaf node) [20]. Korijenski čvor razlikuje se po tome što nema ulaznih čvorova, odnosno roditelja, dok lisni čvor nema izlaznih čvorova, odnosno djece. Svaki unutarnji čvor dijeli podatke na 2 ili više podskupa na temelju diskretne funkcije vrijednosti ulaznih atributa. Skup podataka se rekurzivno dijeli sve dok se unutar svakog čvora ne nalaze podaci koji potpuno ili dominantno pripadaju nekoj od klasa [20].

Postoje 2 vrste kriterija za podjelu: univarijatni i multivarijatni. Neki od najpoznatijih univarijatnih kriterija jesu: informacijski dobitak (engl. *information gain*), Gini indeks i Chi-square. Multivarijatni kriteriji razlikuju se od univarijatnih po tome što za isti čvor u obzir uzimaju veći broj varijabli, dok univarijatni u obzir uzimaju samo jednu varijablu [20]. Unatoč ovim razlikama, istraživanja su pokazala da izbor kriterija za podjelu nema prevelik utjecaj na konačne performanse algoritama stabla odlučivanja.

3.4.2. Implementacija stabla odlučivanja

```
decision_tree = DecisionTreeClassifier(random_state=10)

decision_tree.fit(x_train, y_train)

y_predict = decision_tree.predict(x_test)
```

Programski kod 3: instanciranje i treniranje modela stabla odlučivanja

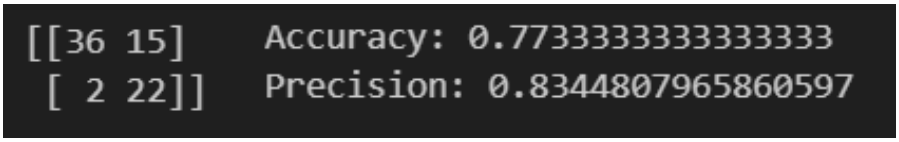
Prvi isječak programskog koda prikazuje kreiranje instance modela Klasifikatora stabla odlučivanja (engl. *Decision Tree Classifier*), što se odvija pozivom klase *DecisionTreeClassifier* s argumentom *random_state* koji omogućava reproduciranje rezultata [21]. Pozivom metode *fit* model se trenira nad skupom podataka za treniranje, podijeljenim na *x_train* i *y_train*. Uz pomoć metode *predict* kreiraju se predviđanja nad *x_test* skupom podataka te se sve sprema unutar varijable *y_predict*.

```
from sklearn.metrics import confusion_matrix

print(confusion_matrix(y_test,y_predict))


from sklearn.metrics import accuracy_score

from sklearn.metrics import precision_score
```

```
print('Accuracy:',accuracy_score(y_test, y_predict))
print('Precision:',precision_score(y_test, y_predict,
average='weighted'))
```

Programski kod 4: uvoženje biblioteka i ispis metrika za procjenu kvalitete modela stabla odlučivanja

Nakon što je model istreniran, potrebno je provjeriti njegovu kvalitetu korištenjem različitih metrika. Za svaki od 3 modela u ovom završnom radu korištene su matrica zabune (engl. *Confusion matrix*), točnost (engl. *Accuracy*) i preciznost (engl. *Precision*). Ove metode uvezene su iz biblioteke ScikitLearn, te su pozvane unutar funkcije print s argumentima *y_test* koji predstavlja stvarne y vrijednosti testnog skupa podataka, te *y_predict* koji predstavlja predviđanja koja je model napravio.



Slika 12: rezultati prvog modela stabla odlučivanja (vlastita izrada)

S lijeve strane slike 12 prikazana je matrica zabune, koja prikazuje da je model ispravno predvidio 36 stvarno pozitivnih vrijednosti (gore lijevo), 15 lažno pozitivnih vrijednosti (gore desno), 2 lažno negativne vrijednosti (dolje lijevo) i 22 stvarno negativne vrijednosti (dolje desno). Točnost modela iznosi 77.33%, a preciznost 83.45%.

```
depth_accuracy = []
for depth in range(1,20):
    new_dt = DecisionTreeClassifier(random_state=10, max_depth=depth)
    new_dt.fit(x_train, y_train)
    new_dt_predict = new_dt.predict(x_test)
    depth_accuracy.append(accuracy_score(y_test, new_dt_predict))

depth_df = pd.DataFrame({'max_depth':range(1,20),
'accuracy':depth_accuracy})

plt.figure(figsize=(15,8))
plt.plot(depth_df['max_depth'], depth_df['accuracy'], marker='o')
```

```
plt.xlabel('Depth of tree')
plt.ylabel('Performance')
```

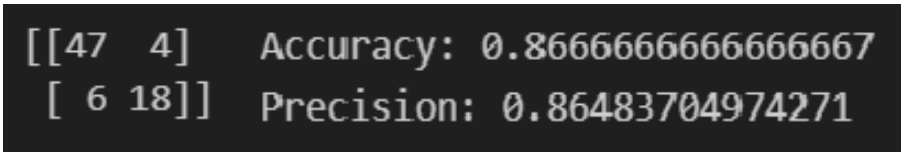
Programski kod 5: pronalaženje najbolje vrijednosti hiperparametra dubine stabla

Kako bi se performanse modela poboljšale, potrebno je pronaći najbolje hiperparametre za model ovisno o problemu. U ovom radu odabrana su 2 hiperparametra za koje je provedeno pronalaženje najbolje vrijednosti: najveća dubina stabla (engl. *Max_depth*) i najveći broj lisnih čvorova (engl. *Max_leaf_nodes*). Primjer pronalaska najbolje vrijednosti hiperparametra najveće dubine prikazan je u programskom kodu 5. Prvo se kreira prazna lista unutar koje će se spremati dobivena točnost modela za svaku vrijednost hiperparametra u rasponu od 1 do 20. Unutar for petlje u rečenom rasponu kreira se novi model s vrijednosti hiperparametra postavljenom na broj prolaza petlje u kojem se trenutno nalazi. Zatim se za taj model provodi postupak treniranja i predviđanja kao u programskom kodu 3. Zatim se od dobivenih rezultata kreira DataFrame, te se na kraju crta graf koji prikazuje vrijednost za svaku treniranu instancu [21]. Isti postupak provodi se za hiperparametar najvećeg broja lisnih čvorova, s jedinom razlikom da je raspon vrijednosti od 2 do 25.

```
dt = DecisionTreeClassifier(random_state=10, max_depth=1,
max_leaf_nodes=2)
```

Programski kod 6: kreiranje novog modela s optimalnim hiperparametrima

Pronađene najbolje vrijednosti hiperparametara su za najveću dubinu 1, te za najveći broj lisnih čvorova 2. S ovim vrijednostima hiperparametara ponovljen je postupak treniranja i predviđanja modela, te su dobiveni rezultati prikazani na slici 13:



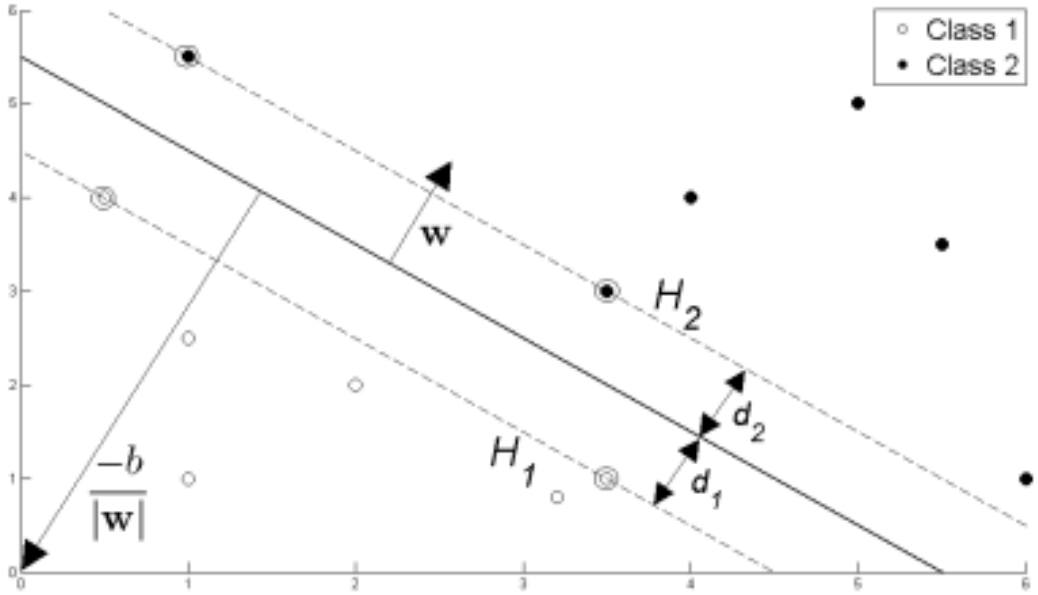
Slika 13: rezultati drugog modela stabla odlučivanja (vlastita izrada)

Kao što je vidljivo sa slike, nakon pronalaska vrijednosti hiperparametara, model je predvidio 47 stvarno pozitivnih, 4 lažno pozitivne, 6 lažno negativnih i 18 stvarno negativnih vrijednosti. Točnost modela iznosi 86.66%, a preciznost 86.48%.

3.5. Metoda potpornih vektora

3.5.1. Teorijski opis metode potpornih vektora

Metoda potpornih vektora (engl. *Support Vector Machine*, kraće SVM) predstavlja algoritam strojnog učenja kojem je cilj razdvojiti pripadnike određenih klasa prema njihovim značajkama. Moguće je to zamisliti kao da se na grafu sa svim primjercima skupa podataka prikazanim kao točkama na tom grafu pokušava nacrtati linija koja je što je više moguće udaljena od točaka različitih klasa koje su međusobno najbliže [22].



Slika 14: vizualni prikaz principa metode potpornih vektora (izvor: Fletcher)

U ovom slučaju linija se naziva hiperravnina, a točke koje predstavljaju podatke potporni vektori. U suštini, Metoda potpornih vektora pronalazi hiperravninu koja najbolje dijeli skup podataka na klase [23]. Algoritam treba pronaći najbolju poziciju i orijentaciju hiperravnine kako bi došlo do što kvalitetnije generalizacije podataka. Matematička formula hiperravnine je sljedeća:

$$w * x + b = 0$$

Gdje je w vektor normale na hiperravninu, a b pristranost (engl. *Bias*). Cilj ovog algoritma je pronaći optimalne vrijednosti za w i b [23].

3.5.2. Implementacija metode potpornih vektora

```
from sklearn.svm import SVC

svc_model = SVC()

svc_model.fit(x_train, y_train)

svc_predict = svc_model.predict(x_test)


print('Accuracy:',accuracy_score(y_test, svc_predict))

print('Precision:',precision_score(y_test, svc_predict,
average='weighted'))

print(confusion_matrix(y_test,svc_predict))
```

Programski kod 7: kreiranje modela SVM

Ponovno iz biblioteke ScikitLearn uvozi se SVC (engl. *Support Vector Classifier*), te se odmah instancira model. Model se zatim metodom *fit* trenira na skupu podataka za treniranje, te se kreira predviđanje na skupu podataka za testiranje bez ciljne varijable, *x_test*. Koriste se iste metrike kao i kod modela stabla odlučivanja, koje se pozivaju na isti način kao i u programskom kodu 4.

```
[[50  1]   Accuracy: 0.8533333333333334
 [10 14]]  Precision: 0.8653333333333334
```

Slika 15: rezultati prvog SVM modela (vlastita izrada)

Model SVM predvidio je 50 stvarno pozitivnih, 1 lažno pozitivnu, 10 lažno negativnih i 14 stvarno negativnih vrijednosti, uz točnost od 85.33% i preciznost od 86.53%.

```
param_grid = {'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001],
'kernel': ['rbf']}


from sklearn.model_selection import GridSearchCV

grid_svc = GridSearchCV(SVC(), param_grid, refit=True, verbose=3)


grid_svc.fit(x_train, y_train)
```

```
grid_predict = grid_svc.predict(x_test)

print(grid_svc.best_params_)
```

Programski kod 8: podešavanje hiperparametara pomoću GridSearchCV te treniranje I predviđanje novog SVM modela

Za ovaj podešavanje hiperparametara u ovom slučaju korišten je *GridSearchCV* pristup. Ovaj pristup funkcionira tako da se unutar varijable najčešćeg naziva *param_grid* kreira rječnik (engl. *Dictionary*) s nazivima hiperparametara i listama vrijednosti za svaki hiperparametar. U ovom slučaju, odabrani su hiperparametri C, odnosno parametar regularizacije, i gamma, odnosno koeficijent korištenog kernela. Vrijednosti za isprobavanje odabrane su po uzoru na vrijednosti s laboratorijskih vježbi na predmetu Uvod u Umjetnu Inteligenciju koji je autor rada slušao u zimskom semestru akademske godine 2024./2024. na Fakultetu organizacije i informatike u Varaždinu.

Nakon uvoženja GridSearchCV-a, isti se instancira uz specifikaciju korištenog algoritma, u ovom slučaju SVC i ranije spomenutog *param_grid*-a. Zatim se za svaku moguću kombinaciju hiperparametara trenira zaseban model, te se kao finalni model koristi onaj s najboljim performansama. S tim modelom kreira se predviđanje, a s *grid_svc.best_params_* moguće je i vidjeti najbolje vrijednosti hiperparametara, u ovom slučaju C=1 i gamma=0.001.

```
[[48  3]  Accuracy: 0.88
 [ 6 18]] Precision: 0.8787301587301587
```

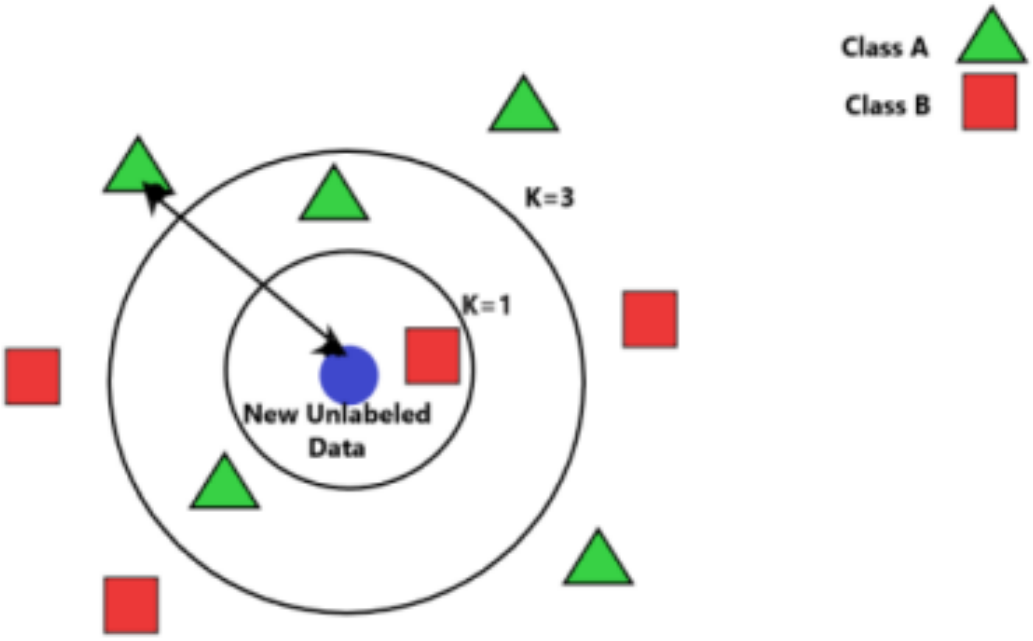
Slika 16: rezultati poboljšanog SVM modela (vlastita izrada)

Novi model SVM predvidio je 48 stvarno pozitivnih, 3 lažno pozitivne, 6 lažno negativnih i 18 stvarno negativnih vrijednosti, uz točnost od 88% i preciznost od 87.87%.

3.6. K najbližih susjeda

3.6.1. Teorijski opis K najbližih susjeda

K najbližih susjeda (KNN) jednostavan je i učinkovit algoritam koji se uglavnom primjenjuje na problemima klasifikacije. Karakterizira ga to što ne radi nikakve predrasude o početnom skupu podataka, zbog čega ga se naziva i neparametarskim algoritmom [24]. Ovaj algoritam provodi klasifikaciju tako da klasificira podatke prema najbližim ili susjednim primjercima unutar neke regije na primjerice nekom grafu, od čega i naziv K najbližih susjeda. Algoritam također nazivamo i lijenim algoritmom, zato što se proces učenja svodi na pamćenje svih primjeraka iz skupa podataka.



Slika 17: prikaz rada KNN algoritma (izvor: Taunk et al.)

Slika 17 prikazuje princip rada KNN algoritma. Za svaki primjerak računaju se udaljenosti do svih drugih primjeraka te se u obzir uzimaju svi primjerci koji se nalaze unutar radijusa K. Klasa novog ulaza određuje se prema klasi kojoj pripada većina pronađenih susjeda [24]. U cijelom procesu vrlo je bitan dobar odabir veličine K, jer kao što je vidljivo sa slike 17, 2 različite K vrijednosti mogu rezultirati s 2 različita predviđanja algoritma. Potencijalni problem ovog pristupa nalazi se u činjenici da ovisno o veličini K algoritam može biti vrlo osjetljiv na anomalije.

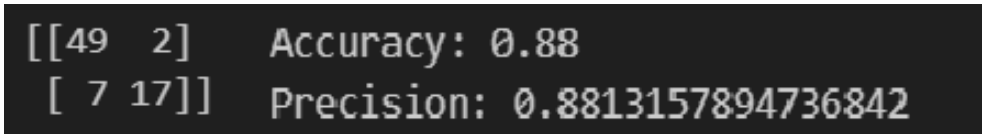
Odabir veličine K može se odviti na razne načine, a dosta je popularna i *brute-force* tehnika, koja podrazumijeva treniranje algoritma više puta s različitim K vrijednostima, te utvrđivanje najboljih rezultata, što je bilo primijenjeno i u ovom radu s upitnim uspjehom.

3.6.2. Implementacija K najbližih susjeda

```
from sklearn.neighbors import KNeighborsClassifier


knn_model = KNeighborsClassifier(n_neighbors=5)

knn_model.fit(x_train, y_train)


knn_predict = knn_model.predict(x_test)


print('Accuracy:',accuracy_score(y_test, knn_predict))

print('Precision:',precision_score(y_test, knn_predict,
average='weighted'))

print(confusion_matrix(y_test,knn_predict))
```

Programski kod 9: kreiranje modela KNN

Iz biblioteke ScikitLearn uvozi se *KNeighborsClassifier*, te se kreira instanca modela s parametrom broja susjeda (*n_neighbors*) postavljenim na 5, što je zadana vrijednost. Model se zatim trenira te se kreiraju predviđanja na isti način kao i u prethodna 2 slučaja, što vrijedi i za ispis metrika za mjerenje kvalitete.



Slika 18: rezultati prvog KNN modela (vlastita izrada)

Model KNN predvidio je 49 stvarno pozitivnih, 2 lažno pozitivne, 7 lažno negativnih i 17 stvarno negativnih vrijednosti, uz točnost od 88% i preciznost od 88.13%.

```
k_accuracy = []


for k in range(1,20):
    new_knn = KNeighborsClassifier(n_neighbors=k)
    new_knn.fit(x_train, y_train)
    new_knn_predict = new_knn.predict(x_test)
    k_accuracy.append(accuracy_score(y_test, new_knn_predict))


k_df = pd.DataFrame({'k':range(1,20), 'accuracy':k_accuracy})


plt.figure(figsize=(15,8))
plt.plot(k_df['k'], k_df['accuracy'], marker='o')
plt.xlabel('K')
plt.ylabel('Performance')
```

Programski kod 10: pronalaženje najboljeg parametra K za KNN

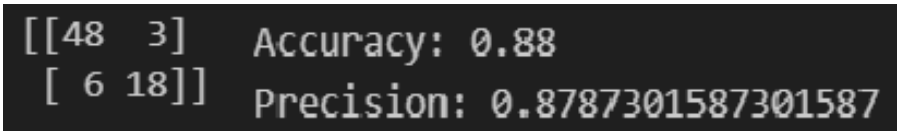
Programski kod 10 prikazuje proces pronalaženja najbolje vrijednosti parametra K, koji je napravljen na isti način kao u programskom kodu 5. Problem se nalazi u tome što je najvišu vrijednost točnosti moguće dobiti za više različitih vrijednosti N, te je stoga proveden i GridSearchCV postupak.

```
k_range = list(range(1,20))


knn_parameters = {
    'n_neighbors': k_range,
    'leaf_size': (10,20,30,40,50),
    'p': (1,2),
```

```
        'weights': ('uniform', 'distance'),

        'metric': ('minkowski', 'chebyshev')}


    grid_knn = GridSearchCV(KNeighborsClassifier(), knn_parameters,
scoring = 'accuracy', refit=True, verbose=3)
```

Programski kod 11: GridSearchCV postupak za KNN model

Za podešavanje hiperparametara uzeti su broj susjeda, veličina lista, p koji određuje metodu distance za minkowski metriku, težine koje određuju funkciju težine za predikciju, te metriku. Odabrane vrijednosti pronađene su na sustavu Kaggle.com [25]. Treniranjem svake kombinacije dobivena je sljedeća lista najboljih parametara: veličina lista: 20, metrika: chebyshev, broj susjeda: 9, p: 1, težine: distance. Dobiveni rezultati prikazani su na slici 19:



Slika 19: rezultati poboljšanog KNN modela (vlastita izrada)

Poboljšani model KNN predvidio je 48 stvarno pozitivnih, 3 lažno pozitivne, 6 lažno negativnih i 18 stvarno negativnih vrijednosti, uz točnost od 88% i preciznost od 87.87%.

3.7. Usporedba rezultata

Najbolje početne performanse u slučaju točnosti i preciznosti pokazao je KNN model s početnom točnošću od 88% i preciznosti od 88.13%. Drugi je bio SVM model s točnosti od 85.33% i preciznosti 86.53%. Uvjerljivo najgore početne performanse pokazao je model stabla odlučivanja s točnosti od 77.33% te preciznosti od 83.45%. Kod matrica zabune priča je malo kompliciranija između KNN i SVM modela, jer je SVM predvidio 50 stvarno pozitivnih i 1 lažno pozitivni slučaj, što je bolje od KNN-ovih 49 i 2, ali KNN je predvidio 7 lažno negativnih i 17 stvarno negativnih vrijednosti, dok je SVM na tom području ostvario 10 i 14. Sveukupno, u ovoj kategoriji KNN ipak pokazuje bolje rezultate.

Najveći porast u performansama pokazao je model stabla odlučivanja kod kojeg je došlo do povećanja točnosti sa 77.33% na 86.67%, te povećanja preciznosti s 83.45% na 86.48%. Poboljšani SVM model donio je povećanje točnosti s 85.33% na 88% i preciznosti s 86.53% na 87.87%. Kod KNN modela točnost je ostala nepromijenjena na 88%, dok je preciznost pala s 88.13% na 87.87%. Po ovim metrikama na kraju je došlo do potpunog izjednačenja performansi kod KNN i SVM modela, koji nakon poboljšanja parametara također imaju i isti rezultat matrice zabune, odnosno 48 stvarno pozitivnih, 3 lažno pozitivna, 6 lažno negativnih i 18 stvarno negativnih predviđenih vrijednosti. Poboljšani model stabla odlučivanja pokazao je vrlo slične rezultate matrice zabune, s 47 stvarno pozitivnih, 4 lažno pozitivna, 6 lažno negativnih i 18 stvarno negativnih predviđanja.

Ovi rezultati pokazuju da su KNN i SVM modeli poprilično izjednačeni na području analize medicinske dokumentacije i predviđanja na temelju iste, dok model stabla odlučivanja pokazuje malo lošije performanse. No, kako bi ovi modeli bili reprezentativni, bilo bi potrebno trenirati ih na puno većem i kvalitetnijem skupu podataka koji bi bio reprezentativan za čitavu populaciju, što nije slučaj s korištenim skupom podataka. Također bi bilo potrebno i upotrijebiti naprednije metode poboljšanja modela, te koristiti više parametara za GridSearchCV postupak, te bi onda ti modeli mogli pokazivati realističnije rezultate.

4. Zaključak

Strojno učenje područje je koje uživa velik interes u današnje vrijeme, bilo iz aspekta javnosti u medijima, ili iz znanstvenog aspekta u vidu količine istraživanja koja se provode. Samim time, ova grana umjetne inteligencije pronašla je svoju primjenu i na vrlo zahtjevnom i kompleksnom području kao što je medicina. Iako proces uvođenja strojnog učenja u medicinu nije tekao glatko, te je taj proces još uvijek aktivan, iz dana u dan pronalazi se sve veći broj inovativnih rješenja koja povezuju ova dva znanstvena područja.

Najčešće primjene odnose se na nadzirano strojno učenje, odnosno granu strojnog učenja koje uči iz podataka za koje već ima poznati odgovor ishoda. Ovakvim modelima daju se velike količine medicinskih podataka iz kojih modeli uče, te zatim kreiraju vlastita predviđanja i tako pružaju podršku medicinskim stručnjacima. Problematika kod ovog pristupa nalazi se u etičkim pitanjima čuvanja vrlo osobnih medicinskih podataka, te količine povjerenja kojeg se može imati u stroj u odnosu na ljudsko biće. Ipak, primjena strojnog učenja u medicini izuzetno je relevantna i prema mišljenju autora taj će se trend nastaviti i u idućim godinama bit će vidljiva sve bolja i naprednija rješenja.

U sklopu rada kreiran je i praktični dio kojim se željela provjeriti kvaliteta najpopularnijih algoritama strojnog učenja: stabla odlučivanja, K najbližih susjeda i metode potpornih vektora, te mogućnost njihove primjene na području medicine. Zaključeno je da ovi algoritmi itekako imaju svoju primjenu za zadatke klasifikacije, no isto tako u ovakvom akademskom okruženju završnog rada na kojem radi jedan student teško je donijeti reprezentativne i uvjerljive zaključke. Za to bi bilo potrebno više istraživanja, potencijalna primjena hibridnih modela, te povećanje skupa podataka na mnogo veći i reprezentativniji skup na temelju kojega bi se mogla provesti kvalitetna i vjerodostojna generalizacija na cijelu populaciju. Područje definitivno ima svojeg potencijala te zahtijeva veliku pažnju u budućnosti. Sve u svemu, na temelju ovog istraživanja teško je izvesti zaključak o mogućnosti primjene ovih algoritama u medicini, ali rezultati istraživanja snažno ukazuju na tu mogućnost.

Na temelju završnog rada može se izvesti zaključak da je medicina jedno od značajnijih područja primjene strojnog učenja, te da se metode strojnog učenja mogu kvalitetno primjenjivati u medicini, i da će se taj trend nastaviti, što je nužno i potrebno kako bi se nadolazećim generacijama omogućila kvalitetna medicinska skrb i visoka kvaliteta života, što na kraju i treba biti cilj svake primjene umjetne inteligencije u današnjem vremenu.

Popis literature

[1] A. Alanazi, “Using machine learning for healthcare challenges and opportunities,” *Inform. Med. Unlocked*, vol. 30, p. 100924, Jan. 2022, doi: 10.1016/j.imu.2022.100924.

[2] H. Habehh and S. Gohel, “Machine Learning in Healthcare,” *Curr. Genomics*, vol. 22, no. 4, pp. 291–300, Dec. 2021, doi: 10.2174/1389202922666210705124359.

[3] R. C. Deo, “Machine Learning in Medicine,” *Circulation*, vol. 132, no. 20, pp. 1920–1930, Nov. 2015, doi: 10.1161/CIRCULATIONAHA.115.001593.

[4] D. Valkenborg, A.-J. Rousseau, M. Geubbelmans, and T. Burzykowski, “Unsupervised learning,” *Am. J. Orthod. Dentofacial Orthop.*, vol. 163, no. 6, pp. 877–882, Jun. 2023, doi: 10.1016/j.ajodo.2023.04.001.

[5] C. M. Eckhardt *et al.*, “Unsupervised machine learning methods and emerging applications in healthcare,” *Knee Surg. Sports Traumatol. Arthrosc.*, vol. 31, pp. 376–381, Nov. 2022.

[6] V. Nasteski, “An overview of the supervised machine learning methods,” *HORIZONS.B*, vol. 4, pp. 51–62, Dec. 2017, doi: 10.20544/HORIZONS.B.04.1.17.P05.

[7] H. H. Rashidi, N. Tran, S. Albahra, and L. T. Dang, “Machine learning in health care and laboratory medicine: General overview of supervised learning and Auto-ML,” *Int. J. Lab. Hematol.*, vol. 43, no. S1, pp. 15–22, Jul. 2021, doi: 10.1111/ijlh.13537.

[8] “Hematologija,” Hrvatski jezični portal. Accessed: May 28, 2024. [Online]. Available: https://hjp.znanje.hr/index.php?show=search_by_id&id=fV1nURE%3D

[9] “Hematologija,” mes.hr. Accessed: May 28, 2024. [Online]. Available: https://meshr.hr/hematologija.html

[10] K. Rasheed, A. Qayyum, M. Ghaly, A. Al-Fuqaha, A. Razi, and J. Qadir, “Explainable, trustworthy, and ethical machine learning for healthcare: A survey,” *Comput. Biol. Med.*, vol. 149, p. 106043, Oct. 2022, doi: 10.1016/j.compbiomed.2022.106043.

[11] T. L. Beauchamp, “Načela u bioetici,” *Druš. Istraživanja*, vol. 5, no. 3–4, 1996, Accessed: May 22, 2024. [Online]. Available: https://hrcak.srce.hr/31831

[12] D. J. McMaughan, O. Oloruntoba, and M. L. Smith, “Socioeconomic Status and Access to Healthcare: Interrelated Drivers for Healthy Aging,” *Front. Public Health*, vol. 8, p. 231, Jun. 2020, doi: 10.3389/fpubh.2020.00231.

[13] N. Pourmoradi, “Heart failure clinical records.” Kaggle. Accessed: Feb. 10, 2024. [Online]. Available: https://www.kaggle.com/datasets/nimapourmoradi/heart-failure-clinical-records

[14] D. Chicco and G. Jurman, “Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone,” *BMC Med. Inform. Decis. Mak.*, vol. 20, no. 1, p. 16, Dec. 2020, doi: 10.1186/s12911-020-1023-5.

[15] B. Dahlöf, “Cardiovascular Disease Risk Factors: Epidemiology and Risk Assessment,” *Am. J. Cardiol.*, vol. 105, no. 1, pp. 3A-9A, Jan. 2010, doi: 10.1016/j.amjcard.2009.10.007.

[16] *Exploratory Data Analysis in Pandas | Python Pandas Tutorials*. Accessed: Mar. 11, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=Liv6eeb1VfE

[17] “What is a histogram?,” ASQ. Accessed: Jun. 25, 2024. [Online]. Available: https://asq.org/quality-resources/histogram

[18] “Box and Whisker Plots,” Newcastle University. Accessed: Jun. 25, 2024. [Online]. Available: https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/data-presentation/box-and-whisker-plots.html

[19] *House Price Prediction in Python - Full Machine Learning Project*.

[20] Anuradha and G. Gupta, “A self explanatory review of decision tree classifiers,” in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, Jaipur, India: IEEE, May 2014, pp. 1–7. doi: 10.1109/ICRAIE.2014.6909245.

[21] *Decision Tree Full Course by Analytics Vidhya*. Accessed: Apr. 12, 2024. [Online Video]. Available: https://www.youtube.com/playlist?list=PLdKd-j64gDcC5TCZEqODMZtAotCfm5Zkh

[22] T. Fletcher, “Support Vector Machines Explained,” *UCL*, Dec. 2008, Accessed: Jun. 27, 2024. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/43282568/SVM_Explained-libre.pdf?1456956021=&response-content-disposition=inline%3B+filename%3DSupport_Vector_Machines_Explained.pdf&Expires=1719496869&Signature=Pk~KZY0jgFGim01EbmRiinxt~F5zRFIwF77CNG9VTkgUQlVCVsV4Sy0A7-CXalWHFOjcto5OPzthQZ83Lzbfdso-zJxSaqHqqQtVNRY8zA~Z5wLK5Se4zpsrgn-kFYpvNHXhvSqa3ZmWPhKuycpLVEbG8Y8NlRbz8NGUgTIJEf3hJQFHe8CH7jwDyUYlC6rh3t5VXjBR0X8kDnzu0Z7GycXmA-6fPkfnclEXq8bkULTo21tEti7H5bLsYbFyYtIClz3wNYOs3rtruqLuXXtMCTaz-eCzUl2Qmvn3cUlKE1xnI2GpSAzS7adH-m-yvLbQ8XHaGK551EfAuZwyCsAR2w__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[23] M. Schatten and N. Vrček, “A Method for Intruder UAV Pilot Localization based on Neural Networks using Intercepting Drone Constellations,” *Cent. Eur. Conf. Inf. Intell. Syst.*, pp. 457–462, 2023.

[24] K. Taunk, S. De, S. Verma, and A. Swetapadma, “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification,” in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India: IEEE, May 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.

[25] M. Kanbay, “KNN Best Parameters GridSearchCV.” Accessed: Apr. 12, 2024. [Online]. Available: https://www.kaggle.com/code/melihkanbay/knn-best-parameters-gridsearchcv

Popis slika

Popis programskih kodova