Komparativna analiza metoda za klasifikaciju podataka

Vulin, Mate

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Mate Vulin

KOMPARATIVNA ANALIZA METODA ZA KLASIFIKACIJU PODATAKA

DIPLOMSKI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Mate Vulin

Matični broj: 44944/16–R

Studij: Baze podataka i baze znanja

KOMPARATIVNA ANALIZA METODA ZA KLASIFIKACIJU PODATAKA

DIPLOMSKI RAD

Mentorica:

Dr. sc. Jasminka Dobša

Varaždin, veljača 2024.

Mate Vulin

Izjava o izvornosti

Izjavljujem da je moj 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

Rad obrađuje klasifikaciju podataka kao jedan od zadataka rudarenja nad podacima. Objašnjene su različite metode klasifikacije podataka kao što su: linearna klasifikacija (perceptron, logistička regresija, metoda potpornih vektora sa linearnom jezgrom), naivni Bayesov klasifikator, stablo odlučivanja, k najbližih susjeda, slučajna šuma (*engl. random forest*) i umjetne neuronske mreže. Uz kratak teorijski pregled pojma rudarenja nad podacima te klasifikacija, rad se usredotočuje na spomenute metode i na komparativnu analizu metoda za klasifikaciju podataka. U radu su prikazane prednosti i nedostaci pojedinih metoda za klasifikaciju podataka. Preko raznih primjera u radu su dani primjeri korištenja klasifikacije podataka u stvarnom svijetu te se upućuje na samu korisnost klasifikacije podataka. Praktični primjeri opisa i vizualizacije skupova podataka je napravljen u python programskom jeziku, a klasifikacije podataka je implementiran u r programskom jeziku. Prema raznim kriterijima (točnost, preciznost, odaziv i f-1 rezultat) komparativne analize biti će dana procjena klasifikacijskih metoda nad praktičnim primjerima.

Ključne riječi: Komparativna analiza, perceptron, linearna klasifikacija, naivni Bayesov klasifikator, stablo odlučivanja, k najbližih susjeda, rudarenje nad podacima, slučajna šuma, umjetne neurnoske mreže

Sadržaj

1. Uvod

Rad obrađuje klasifikaciju podataka kao jedan od zadataka rudarenja nad podacima. U radu su prikazane različite metode klasifikacije podataka kao što su: metoda potpornih vektora, logistička regresija, perceptron, umjetne neuronske mreže, stablo odlučivanja, slučajne šume, naivna Bayesova metoda te alogoritam k-najbližih susjeda. Procjena samih metoda je napravljena pomoću različitih parametara u komparativnoj analizi, a oni su: točnost, preciznost, odaziv i f-1 rezultat. Motivacija prilikom odabira teme je bila u tome što me zanimalo područje rudarenja nad podacima, a posebice njezin zadatak klasifikacije podataka. Zanimalo me otkrivanje rada klasifikacije podataka te koje su njene primjene u stvarnom svijetu.

Nakon uvodnog dijela o motivaciji i značaju teme bit će dana struktura rada. U početnom dijelu rada nalaze se metode i tehnike rada koje su korištene prilikom razrade teme. Također su dani i svi programski alati i aplikacije koje su korištene. U radu je korišten Overleafov LateX tekstualni uređivač te kaggleove bilježnice. U Overleafovom tekstualnom uređivaču citiranje IEEE meotdom je zaista jednostavno te je zbog toga i izabran. Pomoću kaggle-a su izabrana 2 skupa podataka, a to su skup podataka o predviđanju i analizi srčanog udara i skup podataka o predviđanju zatajenja srca. U kaggleovim bilježnicama je izvršena vizalizacija te dodatni opis podataka u python programskom jeziku te klasifikacija podataka i komparativna analiza u r programskom jeziku. U daljnjem dijelu rada razrađena je tema diplomskog rada. Prvo su objašnjeni ključni pojmovi poput: klasa podataka i rudarenje nad podatcima. U radu je klasificiranje podataka predstavljeno kao jedno od mnogih zadataka rudarenja nad podacima. Nakon kratkog uvoda u temu rad će se usmjeriti na metode klasifikacije podataka, a one su: metoda potpornih vektora, logistička regresija, perceptron, umjetne neuronske mreže, stablo odlučivanja, slučajne šume, naivni Bayesov metoda te alogoritam k-najbližih susjeda. Osim samog teorijskog objašnjenja klasifikacijskih metoda i njezinih principa rada navest će se prednosti i nedostaci pojedinih metoda. Dani su primjeri iz stvarnog svijeta u kojem se primjenjuje klasifikacija podataka te to upućuje na samu koristnost metoda. Nakon teorijskog objašnjenja klasifikacijskih metoda prelazi se na objašnjenje osnovnih pojmova skupova podataka te na opise i vizualizaciju skupa podataka o predviđanju i analizi srčanog udara i skupa podataka o predviđanju zatajenja srca. Kao što je već to i spomenuto to je napravljeno u kaggleovim bilježnicama u python programskom jeziku. Nakon opisa skupova podataka, klasifikacijske metode će zatim biti primjenjene te će biti provedena komparativna analiza pomoću parametara: točnosti, preciznosti, odaziva i f-1 rezultata koji će izvršiti procjenu klasifikacijskih metoda. Na kraju samoga rada je dan zaključak kao osvrt na samu temu.

2. Metode i tehnike rada

Istraživanje informacija je provedeno internetskim putem u kojem su pronađene mnoge knjige, članci te video materijali koji su vezani za temu diplomskoga rada. Diplomski rad je napisan preko online LateX uređivača Overleaf.

Overleaf je jako poznati tekstualni uređivač kojeg sam već koristio prilikom izrade završnog rada. Rad u njemu je doista jednostavan te sam ga odabrao zbog jednostavnosti strukturiranja teksta i citiranja prilikom rada. Za cititranje rada odabrao sam IEEE stil citiranja. Slika samog loga Overleafovog uređivaća se može vidjeti na slici ispod.

Slika 1: Overleaf logo([1])

Zadataci klasifikacije i vizualizacije podataka te komparativna analiza klasifikacijskih metoda odrađena je pomoću kaggle bilježnica. Klasifikacija i komparativna analiza je napravljena pomoću r programskog koda dok su opisi i vizualizacija skupova podataka napravljeni pomoću python programskog koda.

Kaggle bilježnice omogućuju istraživanje i pokretanje koda koji je vezan za strojno učenje. Rad u njima je omogućen u cloud servisu. One zapravo pripadaju Jupyterovim bilježnicama u kojima se programski kod može opisivati u obliku teksta te se može i implementirati. Opis i implementacija koda se dodaje u sekvencijalne čelije. Takve bilježnice podržavaju r i python programski kod. [2]

Kaggleov logo se može vidjeti na slici ispod

Slika 2: Kaggle logo([3])

3. Klasifikacija podataka

U ovome poglavlju je obrađeni teorijski dio diplomskoga rada. Započinje jednostavnim objašnjenjem pojma klase te šireg pojma rudarenja podataka. Jedna od zadataka rudarenja podataka je klasifikacija podataka koja predstavlja glavnu temu diplomskoga rada.

3.1. Definicija klase podataka

Klasa je skup atributa i metoda. Njoj se može pristupiti i koristi se za stvaranje objekta te klase. Dakle, svi objekti koji pripadaju određenoj klasi imaju zajedničke atribute i metode. [4]

Prema hrvatskoj enciklopediji klasa je "Skupina predmeta, pojava, pojmova, bića i dr. koji imaju jedno ili više zajedničkih svojstava; razred, kategorija".[5]

3.2. Rudarenje podataka *(engl. Data Mining)*

3.2.1. Definicija rudarenja podataka

To je analiza promatranih skupova podataka. Ona omogućuje pronalazak odnosa ili veza podataka u kojem se radi sažimanje podataka. Iz takvih sažetaka podaci su korisni i razumljivi vlasniku podataka. Modeli ili uzorci se dobivaju pomoću treniranja upravo iz tih sažetaka i odnosa.[6, str 6]

Obično se rudarenje podataka se bavi podacima koji su već prikupljeni u neku svrhu. Dakle, ono ne sadržava nikakve strategije prikupljanje podataka.[6, str 6]

Bitno je navesti kako efikasnost rudarenja podataka u velikoj mjeri ovisi o kvaliteti podataka. Pojavljuje se opasnosti koja rezultira lošim rezultatima zbog toga što rudarenje podataka uključuje sekundarnu analizu velikih skupova podataka. [6, str 32]

Rudarenje podataka je proces izvlačenja korisnih informacija iz velikih skupova podataka. Može se primjeniti u: relacijskim bazama podataka, skladištima podataka, objektno orijentiranim bazama podataka itd.[7]

Sam proces rudarenja podataka uključuje izvlačenje znanja korištenjem alata za analizu podataka. Alati za analizu podataka mogu koristiti: statističke modele, tehnike strojnog učenja i matematičke algoritme. Stoga rudarenje podataka obuhvaća zadatke za analizu i predviđanje.[8]

3.2.2. Zadaci rudarenja podataka

Prema Handu, Mannili i Smythu vrlo je korisno kategorizirati rudarenje podataka u tipove zadataka koji odgovaraju različitim ciljevima. U takve zadatke spadaju eksploratorna analiza podataka *(engl. Exploratory Data Analysis (EDA))*, deskriptivno modeliranje *(engl. Descriptive Modeling)*, prediktivno modeliranje: klasifikacija i regresija: *(engl. Predictive Modeling: Classification and Regression)*, otkrivanje obrazaca i pravila *(engl. Discovering Patterns and Rules)* i dohvaćanje prema sadržaju *(engl. Retrieval by Content)* [6, str 12-15]

- **Eksploratorna analiza podataka**: njen cilj je istraživanje podataka bez unaprijed definiranih ciljeva ili jasnih ideja o onome što se traži. Ove tehnike su obično interaktivne i koriste vizualne alate prilikom kojih se korisite razne grafičke metode za prikaz podataka. No, problem nastaje kada postoji velik broj varijabli ili postoji visoka dimenzionalnost podataka. Tada postaje izazovno vizualizirati podatke te se koristi tehnika projekcije. Pomoću tehnika projekcije omogućava se stvaranje niskodimenzionalnih projekcija podataka koje daju sažete i korisne informacije. Ponekad, kako bi se vizualizirali podatke u nižoj dimenziji, moraju se žrtvovati neki detalji ili informacije. Dobiti će se niža rezolucija prikaza podataka koji će biti pregledniji, ali uz mogućnost nedostajanja važnih detalja.[6, str 12-13]

- **Deskriptivno modeliranje**: njegov cilj je opis svih podataka ili procesa koji generira takve podatke. Takvi opisi se mogu kreirati pomoću: procjene gustoće *(engl. density estimation)*, klasterske analize i segmentacije *(engl. cluster analysis and segmentation)*[6, str 13]:

- procjena gustoće: model koji opisuje ukupnu distribuciju vjerojatnosti podataka. Pomaže pri razumjevanju kako su podaci raspoređeni u prostoru [6, str 13]

- klasterska analiza i segmentacija : ovaj pristup se koristi za grupiranje sličnih podataka zajedno, odnosno podjela p-dimenzionalnog prostora na grupe. Korištenjem segmentacijske analize cilj je grupirati slične zapise zajedno odnosno želimo razdvojiti zapise u homogene skupine kako bi se lako razumjeli. U klaster analizi, žele se otkriti prirodne skupine ili grupe u podacima. [6, str 13]

- modeliranje ovisnosti : ovdje se istražuje kako se varijable međusobno odnose. Pomaže pri razumijevanju kako jedna varijabla utječe na drugu ili kako se mijenjaju zajedno. [6, str 13]

- **Prediktivno modeliranje:** se ostvaruje putem klasifikacije i regresije. U kojima se podstavlja cilj izgradnje modela koji omogućava predviđanje vrijednost jedne varijable na temelju poznatih vrijednosti drugih varijabli. Ovisno o vrsti problema, varijabla koja se želi predvidjeti može biti različita: kategorička u slučaju klasifikacije ili kvantitativna u slučaju regresije. Temeljna razlika između problema predviđanja i deskripcije je u tome što kod

predviđanja postoji jedna jedinstvena varijabla , dok kod deskriptivnih problema nijedna varijabla nije centralna za modeliranje. [6, str 13-14]

- **Otkrivanje obrazaca i pravila** : uzorci se mogu otkriti na razne načine: otkrivanjem prevara *(engl. fraudulent behavior)*, otkrivanjem neobičnih pojava, otkrivanjem čestih kombinacija itd. [6, str 14-15]
- **Dohvaćanje prema sadržaju** : ovo se često koristi na tekstualnim i fotografskim skupovima podataka. Uzorak sa skupom ključnih riječi omogućuje korisniku pronalazak važnih relativnih dokumenta unutar velikog skupa mogućih relativnih dokumenata na primjeru dohvaćanja teksta. Oglednim slikama, skicama ili opisom slika korisniku se omogućuje pronalazak relevantnih slika iz velikog skupa slikovnih podataka na primjeru dohvaćanja slike. [6, str 15]

3.2.3. Komponente algoritama za rudarenje podataka

Prema Handu, Mannili i Smythu postoje četiri osnovne komponente algoritama za rudarenje podataka:

- 1. **struktura modela ili uzorka *(engl. Model or Pattern Structure)***: radi na određivanju temeljne strukture ili funkcionalnog oblika koji se traži iz podataka
- 2. **funkcija bodovanja *(engl. Score Function)***: radi na procjeni kvalitete uklopljenog modela (engl. fitted model)
- 3. **metoda optimizacije i pretraživanja *(engl. Optimization and Search Method)***: radi na optimizaciji funkcije bodovanja i na pretraživanja preko različitih modela i struktura uzoraka
- 4. **strategija upravljanja podacima *(engl. Data Management Strategy)***: radi na učinkovitom rukovanju pristupom podacima tijekom pretraživanje i optimizacija.

[6, str 15]

Struktura modela ili uzorka

Struktura modela daje globalni sažetak skupa podataka te može dati izjavu o bilo kojoj točki u promatranom podatkovnom prostoru dok strukture uzoraka mogu davati izjave samo o ograničenom području prostora obuhvaćenog varijablama. [6, str 11-12]

Funkcija bodovanja

radi na kvantificiranju odnosno daje ocjenu koliko dobro struktura parametara ili struktura modela odgovara danom skupu podataka u kojima se često koriste funkcije najmanjih kvadrata i točnost klasifikacije. Najčešće funkcije bodovanja su: vjerojatnost, zbroj kvadratne pogreške i

stopa pogreške. Funkcija bodovanja kvadratne pogreške definirana je kao "

$$\sum_{i=1}^{n}(y(i) - \hat{y}(i))^2$$

gdje se predviđa n ciljanih vrijednosti $y(i), 1 = i = n$, a naša predviđanja za svaku su označena kao $\hat{y}(i)$." [6, str 15-16]

Prema Handu, Mannili i Smythu postoje postoje 3 vrste funkcija bodovanja:

1. funkcija bodovanja za modele i funkcija bodovanja za uzorke
2. funkcija bodovanja za prediktivne strukture i funkcija bodovanja za deskriptivne strukture
3. funkcija bodovanja za modele fiksne složenosti i funkcija bodovanja za modele različite složenosti.

[6, str. 129]

Metode optimizacije i pretraživanja
Pronalazak strukture modela ili strukture uzorka i odabir odgovarajućih vrijednosti parametara u kojim se dobivaju najmanje (ili najveće) vrijednost bodovne funkcije je glavni cilj optimizacije i pretraživanja. [6, str 16]

Strategija upravljanja podacima
Glavni cilj strategije upravljanja podacima je da se svim pojedinačnim podatkovnim točkama može pristupiti brzo i učinkovito prilikom pristupanja u radnoj memoriji. [6, str 16-17]

3.2.4. Sličnost tradicionalne statistike i rudarenja podataka

Glavna razlika između tradicionalnih statističkih aplikacija i rudarenja podacima je veličina skupa podataka nad kojim one rade. Tradicionalne statistika smatra velikim skupom podataka nekoliko stotina ili tisuća točaka podataka, a rudarenja podataka nekoliko milijuna ili milijardi podatkovnih točaka. [6, str 17-19]

3.3. Definicija klasifikacije podataka

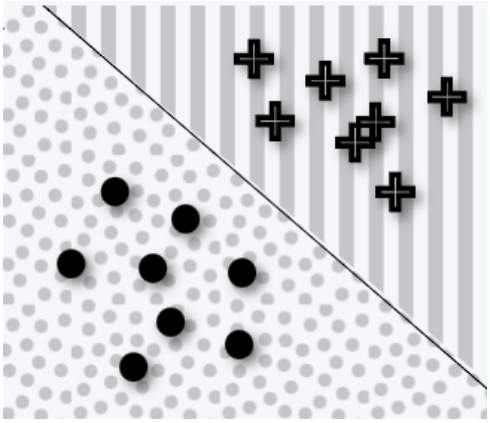
Klasifikacija podataka je tehnika rudarenja podataka koja se temelji na strojnom učenju. Ona omogućava razvrstavanje svakog podatka u skup unaprijed definiranih klasa ili grupa. Ova tehnika koristi mnoge matematičke koncepte. Neki od algoritama koji se koriste za klasifikaciju podataka su: logistička regresija, naivni Bayesov algoritam, algoritam k-najbližih susjeda itd.[7]

Prema Handu, Mannili i Smythu "U klasifikaciji se želi postići mapiranje vektora x na kategoričku varijablu Y. Varijabla koju treba predvidjeti nazivamo varijablom klase". Varijabla klase C koja može poprimati vrijednosti $c_1, ..., c_m$, a promatrane varijable x se najčešće nazivaju ulaznim varijablama. Najčešće se koristi raspon od 0 do 1 za funkciju gubitka u kojoj ispravno predviđanje uzrokuje gubitak 0, a netočno predviđanje klase izaziva gubitak od 1. [6, str 197]

3.4. Aspekti klasifikacije podataka

3.4.1. Diskriminirajuća klasifikacija i granice odlučivanja

S gledišta diskriminativnog okvira klasifikacijski model kao ulaz prima mjerenja u vektoru x i proizvodi kao izlaz klasu sa oznakom iz skupa $C\{c_1, ..., c_m\}$. Ovdje je riječ zapravo o preslikavanju koje proizvodi konstantu površinu. U određenim regijama površina zauzima vrijednost klase c_1, a unija svih takvih regija gdje je predviđena klasa c_1 poznata je kao odluka regija odluke za klasu c_1. Kod realnih problemskih situacija klasifikacija klase nije savršeno odvojiva u X prostoru odnosno postoji mogućnost preklapanja klase. Zbog preklapanja klasa dovodi se do drugog načina gledanja problema klasifikacije podataka u kojem se ne fokusira na površine odlučivanja nego se traži funkcija koje maksimizira odvojenosti između klasa. Takve funkcije nazivaju se diskriminativne funkcije. [6, str 197-198] Na slici 2 je prikazana najednostavnija linerna diskrimativna funkcija između klasa križića i kružića.



Slika 3: Linerna diskriminativna funkcija([9])

3.4.2. Klasifikatori

Prema Handu, Mannili i Smythu postoje tri temeljna pristupa konstruiranja klasifikatora:

- Diskriminativni pristup *(engl. discriminative approach)*: ovdje se pokušava izravno preslikavanje s ulaza x na jedan od m oznaka razreda $c_1, ..., c_m$. Ovakve pristup koriste perceptron i općenitije metodu potpornih vektora.

- Regresijski pristup *(engl. regression approach)*: pokušava izračunati vjerojatnosti posteriorne klase $p(c_k|x)$. Ovakav pristup koristi logistička regresija.
- Klasno-uvjetni pristup *(engl. class-conditional approach)*: koristi Bayesov teorem te ovakav pristup koriste Bayesovi klasifikatori.

[6, str 200-201]

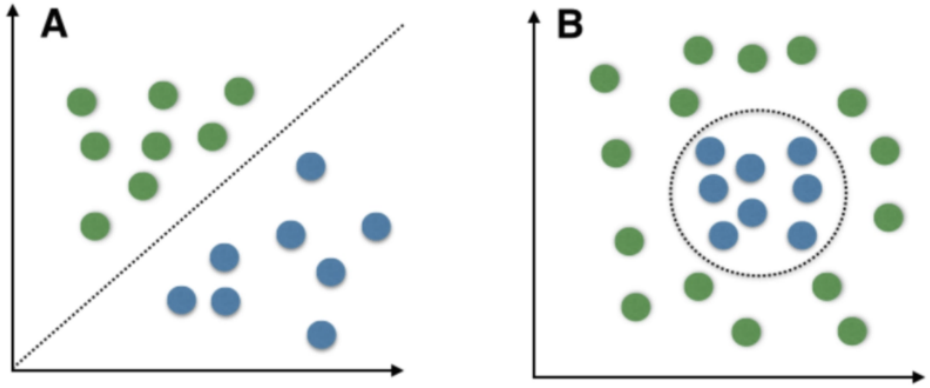
3.5. Metode klasifikacije podataka

3.5.1. Linearna klasifikacija

Koristi za rješavanje problema klasifikacije te pripada nadziranom algoritam strojnog učenja. Najjednostavniji klasifikatori su linearni klasifikatori i oni se sastoje od linearnih funkcija za klasificiranje opažanja. [10]

Učinkovitost ovih modela leži u mogućnostima modela koji pronalaze matematičke kombinacije značajki koje grupiraju podatkovne točke kada pripadaju istoj klasi odnosno razdvaja ih kada pripadaju različitoj klasi. Na takav način prikazuju se jasne granice za klasifikaciju. [11]

Na doljnoj slici s lijeve strane je ilustracija koja djeli zelene i plave podatkovne točke, a što je primjer linerne klasifikacije plave i zelene klase. Desno je primjer nelinerne klasifikacije gdje se ne može primjeniti linerna klasifikacija.



Slika 4: Linearna i nelinearna klasifikacija([12])

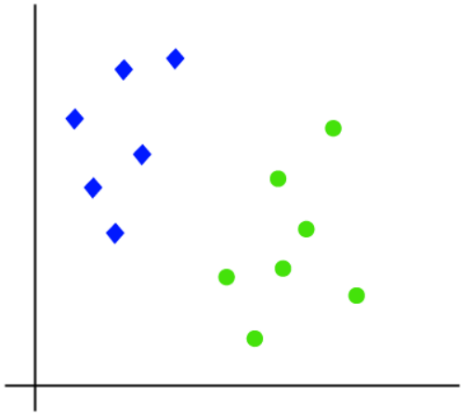
3.5.2. Metoda potpornih vektora *(engl. Support Vector Machine)*

Prema Karatzoglou, Meyeru i Horniku "metoda potpornih vektora (kratica SVM) primjenjuje jednostavnu linearnu metodu na podatke, ali u visokodimenzionalnom prostoru te djeluje nelinearno s obzirom na ulazne podatke."[13]

Prema Karatzoglou, Meyeru i Horniku metoda potpornih vektora ima funkcije jezgre koje vraćaju unutarnji umnožak između dvije točke u prikladnom podatkovnom prostoru, čime se definira pojam sličnosti. Funkcije jezgre mogu biti: linearne, Gaussova radijalna bazna funkcija (RBF) *(engl. Gaussian Radial Basis Function (RBF))*, polinomna, hiperbolička tangenta itd... [13]

Uz pomoću različitih funkcija jezgre mogu se transformirati podatci u linerne klasifikacije. Koristi li se linerna funkcijska jezgra tada je ova metoda jedna od metoda koja se bavi isključivo problemom linearne klasifikacije podataka. SVM najbolje radi kada je skup podataka mali i složen te stvara marginu korištenjem točaka koje su najbliže granici odlučivanja.[14]

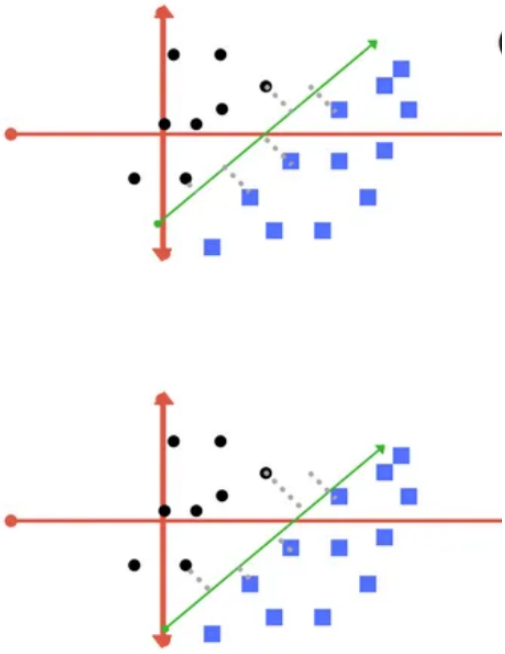
Na slici ispod vidi se primjer podatkovnih točaka od dvije različite klase.



Slika 5: Linearna klasifikacija([14])

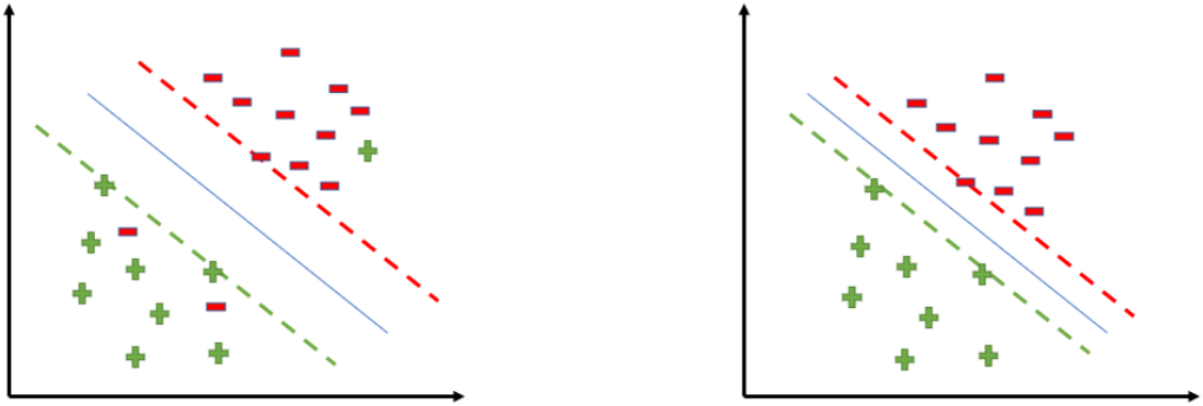
U slučaju potrebe klasificiranja podatkovnih točki na dvije različite klase, moguće je postojanje mnogo granica odlučivanja, zbog čega se javlja pitanje kako pronaći najbolju granicu odlučivanja.[14]

Prema Patelu, granica odlučivanja koja odvaja dvije klase bi trebala ostaviti što više prostora između klasa. Ispravno postavljena granica klasifikacije bi trebala biti podjednako udaljena od obje klase. Na donjoj ilustraciji slike može se vidjeti da je granica odlučivanja previše blizu postavljena plavim podatkovnim točkama te takvu klasifikaciju možemo smatrati pogrešnom na zadanom primjeru.[15]



Slika 6: Granica klasifikacije([15])

Prostor između granice odlučivanja i podatkovne točke klase se može nazvati marginom. Na slici 6 je prikaz ilustracije u kojoj su margine prikazane isprekidanom linijom. Vidljiva je jako velika margina na lijevoj slici dok je desnoj slici jako mala margina. [14]



Slika 7: Greške prilikom linerne klasifikacije([14])

Prednosti i nedostaci metode potpornih vektora:

Prema Yadavu napravljeni su prednosti i nedostaci u metodi potpornih vektora

Prednosti:

- učinkovit je u visokodimenzionalnom skupu podataka
- metoda je učinkovita kada je broj podataka veći od broja podataka koji se koriste prilikom treniranja
- metoda najbolje radi kada su klase odvojive
- na hiperavninu utječu samo vektori potpore tako da extremne vrijednosti imaju mali utjecaj
- metoda je prikladana za binarnu klasifikaciju sa ekstremnim podacima.

[16]

Nedostatci:

- za veći skup podataka potrebn je duži vremenski period obrade podataka.
- ne radi dobro u slučaju preklapanja klasa.
- odabir odgovarajuće funkcije jezgre može biti nezgodan.

[16]

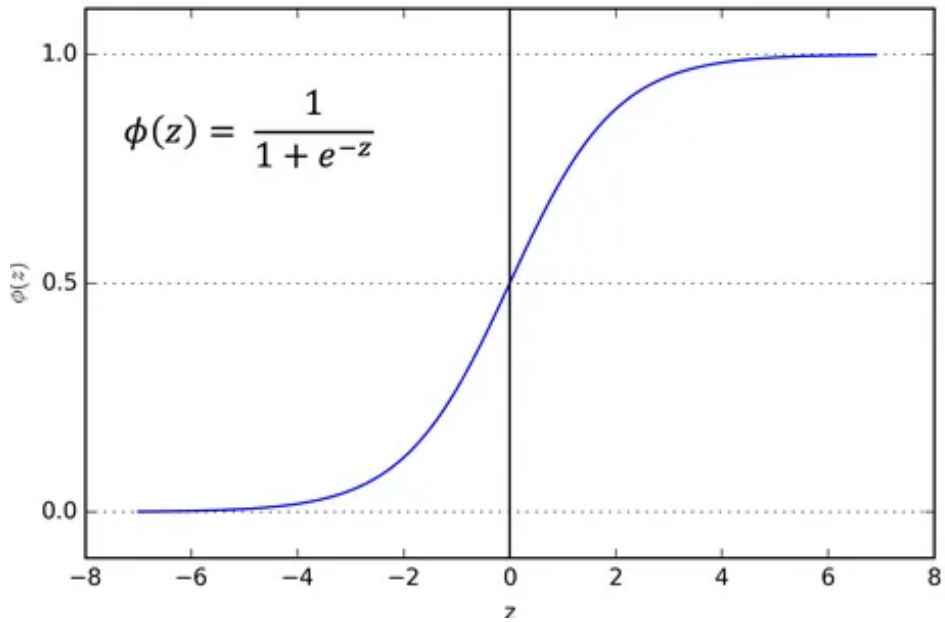
3.5.3. Logistička regresija *(engl. Logistic Regression)*

Ona se dobiva pomoću sigmoidne funkcije koja radi transformacije nad logističkom funkcijom u svrhu ograničavanja vrijednosti y unutar raspona 0-1. Vjerojatnost dane klasifikacije se može predstaviti s okomitom osi, a vrijednost nezavisne varijable x se može predstaviti s vertikalnom osi. Formula logističke regresije je:

$$F_x = \frac{1}{1 + e^{-\beta_o + \beta_1 x}}$$

[17]

Slika ispod prikazuje sigmoidnu funkciju.



Slika 8: Sigmoidna funkcija([18])

Podaci modela logističke regresije ovisni su o težinama preko kojih se vrši klasifikacija samih podataka . Svaki ulazni podatak preslikan je na vrijednost između 0 i 1 u kojim se takva vrijednost smatra kao vjerojatnost da ulazni podatak pripada nekoj klasi. Pomoću algoritama učenja podešavaju se težine kako bi se ispravno izvršila klasifikacija podataka tijekom treniranja. Kada su težine ispravno podstavljenje može se primjeniti logistička funkcija kako bi svaki primjer dobio vjerojatnost da pripada nekoj klasi. [17]

Logistička regresija nije sklona prekomjernom prilagođavanju *(engl. overfitting)* zbog linearnih odnosno nekompleksnih granica odlučivanja. [17]

Vrste logističke regresije

Prema Kanade-u napravljena je podjela logističke regresije na 3 vrste, a to su binarna logistička regresija, multinomna logistička regresija i ordinalna logistička regresija.

1. Binarna logistička regresija

ona predviđa odnos između neovisnih i binarnih zavisnih varijabli.

2. Multinomna logistička regresija

ona predviđa kategoričnu zavisna varijabla koja ima dva ili više diskretnih ishoda.

3. Ordinalna logistička regresija

ona se primjenjuje kada je zavisna varijabla u uređenom stanju, tj. ordinalna. Zavisna varijabla (y) specificira poredak s dvije ili više kategorija ili razina.

[19]

Ključne prednosti logističke regresije

- lakša implementacije metode strojnog učenja
- prikladna za linearno odvojive skupove podataka
- pruža vrijedne uvide.

[19]

Primjeri korištenja logističke regresije Obično se koristi za probleme predviđanja i klasifikacije. Ovo su neki od primjera: otkrivanje prijevare, predviđanje bolesti i predviđanje odljeva korisnika. [20]

3.5.4. Perceptron

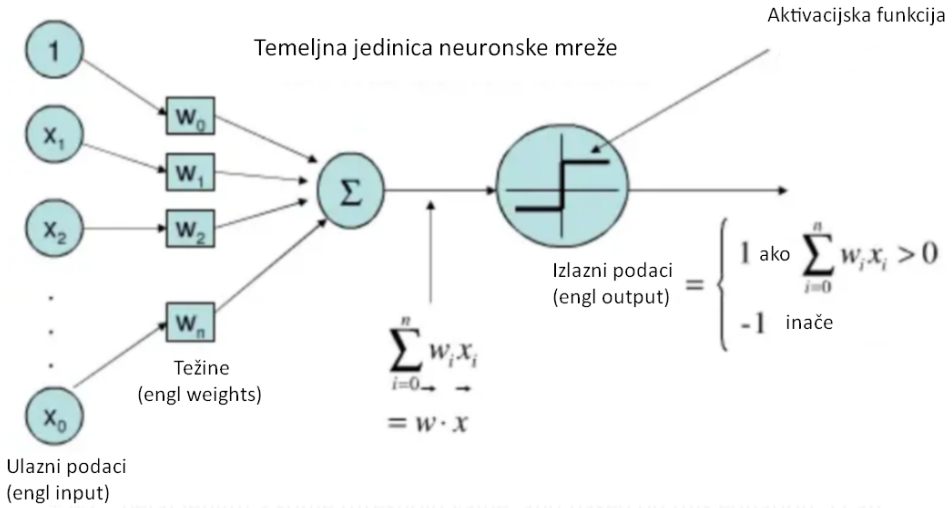
Algoritam perceptrona koristi se za binarnu klasifikaciju te spada u grupu algoritama umjetnih neuronskih mreža. On je jednoslojna neuronska mreža s unaprijednim prijenosom čija je sposobnost ograničena na zadatke binarne i linearne klasifikacije.[21]

Perceptron je linearni odnosno binarni klasifikator koji se koristi u nadziranom učenju. [22]

Algoritam perceptrona pokušava opanašati mehanizam biološkog prjenosa signala neurona.[23]

Način funkcioniranja perceptrona

Svaki ulaz predstavljen je s $x_1, x_2, x_3 \ldots x_n$. Ako zbroj ulaznih signala prijeđe određenu vrijednost (prag), neuron će se aktivirati te će poslati poruku sljedećem neuronu. Predpostavlja se da postoji n ulaza koji su označeni kao $x_1, x_2, x_3 \ldots x_n$. Kada se ulazi pomnože s težinama $w_1, w_2, w_3 \ldots w_n$ redom te se dodaje član (w_0), što rezultira $x_0 w_0 + x_1 w_1 + x_2 w_2 + \ldots + x_n w_n > 0$, neuron se aktivira. Težine $w_1, w_2 \ldots w_n$ i izraz pristranosti w_0 uči na temelju prošlih iskustava, gdje je x_0 uvijek jednak 1. Za binarnu klasifikaciju ako su rezultati ovih izračuna veći od 0 klasificira se kao reprezentator klase +1, a ako su manji ili jednaki 0 on se klasificira kao reprezentator klase -1. Na taj način se vrši binarna klasifikacija ulaznih podataka. Vizalni prikaz opisanoga može se vidjeti na slici broj 8.[23]



Slika 9: Funkcioniranje perceptrona([23], slobodan prijevod)

Prednosti perceptrona:

- najjednostavniji je algoritam neuralnih mreža za problem linearne klasifikacije.

[23]

Nedostaci perceptrona:

- ovaj algoritam će se zaustaviti samo kada su podaci linearno odvojivi
- algoritam perceptron može samo odrediti pripada li rezultat klasi A ili klasi B, ali ne može pružiti vjerojatnosti pripadnosti klasi A ili klasi B.

[23]

3.5.5. Umjetne neuronske mreže *(engl. Artificial Neural Networks(ANN))*

"Umjetna neuronska mreža je uređena trojka (N, V, w) koju čine 2 skupa N, V i funkcija w. N skup je skup neurona, a V skup $\{(i, j)|i, j \in N\}$ je skup veza između neurona i i neurona j. Funkcija w definira težinu (snagu) veze. $W(i, j)$ je težina veze između neurona i i neurona j koju skraćeno možemo zapisati kao $w_{i,j}$. Težina ili snaga veze se implementira u kvadratnu matricu W ili u težinski vektor W gdje redak predstavlja gdje započinje veza, a stupac završetak veze odnosno ciljani neuron."[24, str. 34][25]

Umjetne neuronska mreža je vrsta umjetne inteligencije koja je uvelike ispirirana načinom rada ljudskog mozga. Ona radi stvaranjem veze između elemenata koji se nazivaju umjetnim neuronima, a njihova organizacija i težine veze kojim su međusobno povezani određuju rezultat. Mogu učinkovito obaviti predviđanje događaja kada imaju veliki broj primjerima pomoću kojih izvršava predikciju na sami događaj. Većina neuronskih mreža potpuno su povezane. [26]

Umjetne neuronske mreže uobičajno su pogodne za rad iz velikih količina podataka u kojima pronalaze složene uzorke. Za optimizaciju performansi koriste algoritam propagacije pogrešaka unatrag za treniranje i gradijentni spust . Pripadaju algoritmima strojnog učenja te imaju široku primjenu u stvarnom svijetu. Umjetne neuronske mreže se sastoje od međusobno povezanih slojeva neurona koji obrađuju informacije i daju predviđanja. Svaki neuron u mreži prima ulaz od drugih neurona te primjenjuje nelinearnu transformaciju na ulazne podatke i prosljeđuje izlaz drugim neuronima u mreži. [27]

Učenje u ANN-u

Prema Zakari, AL-Shebany i Sarhanu postoje tri glavne vrste učenja: nadzirano učenje *(engl. Supervised learning)*, nenadzirano učenje *(engl. Unsupervised learning)* i ojačano učenje *(engl. Reinforcement learning)*.[28]

Nadzirano učenje

To je tehnika strojnog učenja koja postavlja parametre umjetne neuronske mreže pomoću podataka treniranja. Tijekom učenja umjetne neuronske mreže ona postavljanje vrijednosti parametra za valjanu ulaznu vrijednost nakon što su dobivene izlazne vrijednosti. Podaci za treniranje sastoje se od parova ulaznih i željenih izlaznih vrijednosti. Nadzirano učenje može izvršavati zadatke klasifikacije podataka. [28]

Nenadzirano učenje

To je tehnika strojnog učenje koja postavlja parametre umjetne neuronske mreže pomoću danih podataka i troškovne funkcije koje treba minimizirati. Funkcija troška je najčešće određena vrstom zadatka. Nenadzirano učenje se koristi prilikom problema procjene. Ovdje se nastoji se utvrditi kako su podaci organizirani. Razlika između nadziranog učenja i nenadziranog učenja je upravo u tome što prima samo neoznačene podatke. [28]

Ojačano učenje

To je tehnika strojnog učenja koja postavlja parametre umjetne neuronske mreže prema podacima koje se dobivaju iz interakcije s okolinom. Ojačano učenje planira radnje umjetne neuronske mreže koje bi ona trebala poduzeti u okolini kako bi se nagrada postala maksimalna. [28]

Rad umjetne neuronske mreže

Prema Pandeyu rad umjetne neuronske mreže sastoji se od šest faza: priprema podataka, arhitektura modela, propagacija unaprijed *(engl. Forward Propagation)*, funkcija gubitaka i propagacija pogrešaka unatrag, optimizacija i evaluacija modela. [27]

Faza 1: Priprema podataka

To uključuje: prikupljanje i čišćenje podataka, međusobno odvajanje u skupove za treniranje i testiranje te normalizacija ili standardizacija podataka kako bi se osigurao određeni raspon podataka. [27]

Faza 2: Arhitektura modela

To uključuje odabir broja slojeva i neurona u svakom sloju. Ovdje se također bira aktivacijska funkcija za svaki neuron. [27]

Faza 3: Propagacija unaprijed

Ovdje se isprobava model umjetne neuronske mreže preko ulaznih podataka, propuštajući ih kroz mrežu i dopuštajući svakom neuronu da reagira na dio ulaznog podatka preko kojeg se dobiva rezultat. [26]

Ovdje se ulazini podaci unose u mrežu, a izlaz se računa korištenjem parametara (težina i predrasuda) umjetnih neurona. Pomoću funkcije aktivacije koja se primjenjuje na svaki umjetni neuron dobiva se izlaz umjetnog neurona sve dok ne dođemo do izlaznog sloja koji izbacije rezultat. [27]

Faza 4: Funkcija gubitka i propagacije pogrešaka unatrag

Propagacija unatrag je metoda koja se koristi za izračun težina u umjetnim neuronskim mrežama. [26]

Ovdje se saznaje koliko dobro model radi definiranje funkcije gubitaka koja se intrepretira kao razlika između predviđenog i stvarnog izlaza. [27]

Faza 5: Optimizacija

Uključuje odabir optimizacijskog algoritma u kojem se minimizira funkcija gubitka i poboljšava točnost modela. Dok model ne postigne zadovoljavajuću razinu točnosti ova faza se ponavlja. [27]

Gradijentni silazak je matematička tehnika u kojoj se zapravo traže lokalni minimumi koji smanjuju greške tijekom predikcije odnosno poboljšava se postupno točnost predikcije. [26]

Faza 6: Evaluacija modela

U njoj se izvršava testiranje modela na podacima testiranja u kojima se dobivaju para-

metri točnost, preciznost, odaziva i f-1 rezultata. [27]

Prednosti umjetnih neuronskih mreža

- paralelni rad
- koriste se za pohranjivanje informacija na mreži tako da, čak i u nedostatku para podataka, to ne znači da mreža ne daje rezultate
- postupno se raspadaju.

[29]

Nedostaci umjetnih neuronskih mreža

- umjetne neuronske mreže ovise o hardveru zbog mogućnosti paralelnog rada
- težak je izbor ispravne strukture neuronske mreže
- kada umjetna neuronska mreža daje rješenje problema na temelju nečeg nepoznatog ona nije pouzdana.

[29]

Primjeri korištenja umjetnih neuronskih mreža

Prihvaćene se u raznim industrijama zbog svoje sposobnosti da uče iz podataka i daju predviđanja s visokom točnošću. Na primjer ona se koristi u marketingu prilikom segmentacije kupaca u kojoj se zapravo vrši klasifikacija kupaca prema primjerice ponašanju i kod proizvodnje u kvaliteti kontrole u kojoj se klasificira ispravnost proizvoda [27]

3.5.6. Naivna Bayesova metoda *(engl. Naive Bayes method)*

To je metoda koji riješava kompleksne klasifikacijske slučajeve. Može se izračunati dvije vrste vjerojatnosti, a to su: vjerojatnost pojavljivanja određene kategorije ili klase i vjerojatnost pojave te iste klase uzimajući u obzir dodatni modifikator x. Ova metoda naziva se "naivnom" jer pretpostavlja neovisnost atributa kod ulaznih podataka. Metoda naivnog Bayesa najbolje radi kada su podaci normalizirani prilikom kojih i daje precizne rezultate.[30] [25]

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Za Bayesovo predviđanje koriste se 4 varijable:

- 1. P(A): vjerojatnost da se događaj A dogodi
- 2. P(B): vjerojatnost događanja događaja B
- 3. P(A|B): označava vjerojatnost da će se događaj A dogoditi ako se događaj B već dogodio
- 4. P(B|A): vjerojatnost događaja B s obzirom da se događaj A već dogodio.

[31]

Vrste naivnih Bayesovih klasifikatora:

Prema ibm.com su dane vrste naivnih Bayesovih kalsifikatora, a to su : Gaussanov naivni Bayes *(engl. Gaussian Naive Bayes (GaussianNB))*, multinominalni naivni Bayes *(engl. Multinomial Naïve Bayes (MultinomialNB))* i Brenulijev naivni Bayes *(engl. Bernoulli Naive Bayes (BernoulliNB))*.[32]

Gaussanov naivni Bayes : ovdje se koristi Gaussova distribucija, odnosno normalna distribucija kontinuirane varijable. Ovaj model se prilagođava pronalaženjem srednje vrijednosti i standardne devijacije svake klase. [32]

Multinominalni naivni Bayes : ovdje se pretpostavlja da su podaci raspoređeni prema multinominalnoj distribuciji. Primjer ovakve klasifikacije je klasifikacije neželjene pošte. [32]

Brenulijev naivni Bayes : koristi se booleanovim varijablama. [32]

Prednosti naivnog Bayesovog klasifikatora

- ne treba puno podataka za treniranje
- jednostavana implementacija
- vrlo je skalabilan
- može obraditi kontinuirane i kategoričke podatke
- nije osjetljiv na nebitne podatke
- predviđanje u stvarnom vremenu.

[33]

Nedostaci naivnog Bayesovog klasifikatora

- ima problema s nultim vrijednostima
- pretpostavit će da su svi atributi neovisni
- ne treba ozbiljno shvatiti njegove rezultate vjerojatnosti zbog mogućnosti krive procjene

[33]

Primjeri korištenja naivnog Bayesovog algoritma su klasifikacija teksta i filtriranje neželjene pošte. [33]

3.5.7. Stablo odlučivanja *(engl Decision Tree)*

Koristi se u problemima klasifikacije te pripada jednostavnim modelima strojnog nadziranog učenja što podrazumjeva da se gradi model te se unose podatci treniranja koji se podudaraju s točnim rezultatima iz kojih model uči. Modelu se daju novi takozvani testni podatci kako bi provjerila njegova sposobnost klasifikacije. [34]

Prema Brocanu stablo odlučivanja sastoji se od 3 vrste komponenti :

- čvorovi (engl nodes) — daju odluku o vrijednosti određenog atributa
- rubovi (engl edges)— rub je zapravo jedan od odgovora čvora te gradi vezu sa sljedećim čvorovima
- čvorovi listova (engl leaf nodes)— izlazne točke odnosno krajnji rezultat.

[34]

Dakle, predviđanje vrijednosti iz stabla odlučivanja značilo bi početi od vrha stabla i postavljaju se pitanja na svakom čvoru. Ovisno o odgovoru čvora supušta se po ispravnoj grani i nastavlja s tim sve dok se ne dođe do lisnog čvora koji zapravo i predstavlja odluku za klasifikaciju određenog unosnog podatka. [34]

Entropija *(engl. Entropy)*

Entropija je mjera nesigurnosti ili mjera nereda u skupu podataka. Formula entropije je:

$$E(S) = -p_{(+)}log(p_{(+)}) - p_{(-)}log(p_{(-)})$$

Gdje je: $p_{(+)}$ vjerojatnost pozitivne klase $p_{(-)}$ vjerojatnost negativne klase S podskup primjera treniranja [35]

Dobitak informacije *(engl. Information Gain)*

Dobitak informacije kvantificira smanjenje nesigurnosti odnosno mjere nereda s obzirom na neki podatak i također je odlučujući faktor o kojem će biti izabran atribut kao čvor odlučivanja ili korijenski čvor. [35]

Prednosti

- razumljiv odnosno jednostavna intrepretacija rezultata
- izrazito je fleksibilan, ne smetaju mu nulte vrijednosti i nije mu potrebna skoro nikakva priprema podataka
- koristiti se za zadatke klasifikacije i regresije
- neosjetljiv na odnose između atributa odnosno ne smeta mu visoka zavisnost.

[36]

Nedostaci

- sklonost pretjeranom prilagođavanju odnosno ne generaliziraju se dobro na nove podatke
- male varijacije unutar podataka mogu proizvesti vrlo različito stablo odlučivanja
- njihovo treniranje može biti skuplje u usporedbi s drugim algoritmima

[36]

3.5.8. Slučajna šuma *(engl. Random Forest)*

Prema Braiman-u, slučajna šuma je " klasifikator koji sadrži kolekciju klasifikatora stabala odluke $h(x, \theta_k), k = 1, ...$ gdje θ_k su nevisno jednako distribuirani slučajni vektori te svako stablo odluke daje glas za najpopularniju klasu na ulazu x." [37]

Slučajna šuma često je korišten algoritam strojnog učenja u kojem se kombinira izlaz više stabala odlučivanja kako bi se postigao jedan rezultat. [38]

Rad algoritma slučajna šuma

Za bolje razumljevanje rada algoritma slučajnih šuma mora se poćeti od objašnjenja tehnika skupnog učenja. Ta tehnika učenja jednostavno znači kombiniranje više modela odnosno može se reći da se kolekcija modela koristi za izradu predviđanja, a ne pojedinačni model. [39]

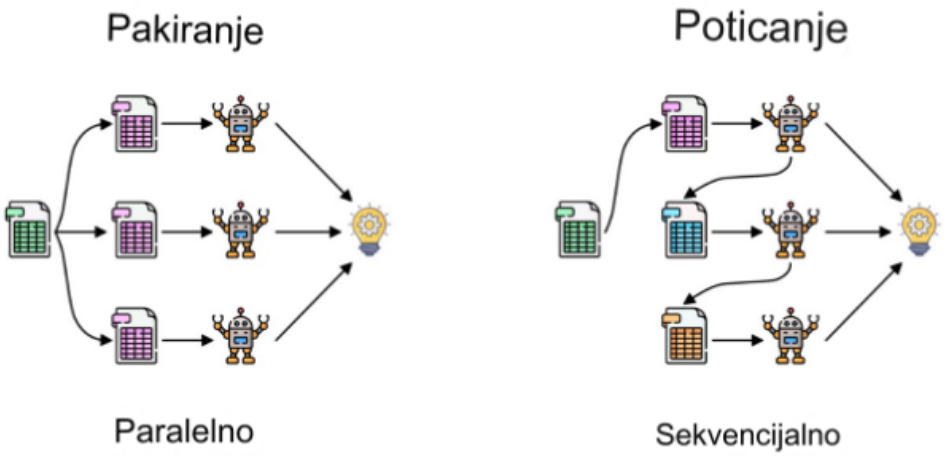
Skupna tehnika učenja *(engl. Ensemble)* koristi dvije vrste metoda:

Pakiranje *(engl. Bagging)*

Stvara drugčiji podskup podataka za treniranje iz izvornih podataka o treniranju, a konačni rezultat se dobiva pomoću većinskog glasovanja. [39]

Poticanje *(engl. Boosting)*

Poticanje koristi koncept skupnog učenja u kojem se kombiniraju slabi ili jednostavnih modeli u snažne ili složene modele stvaranjem sekvencijalnih modela tako da konačni model ima najveću točnost. [39]



Slika 10: Pakiranje i Poticanje([39], slobodan prijevod)

Rad pakiranja

Prema Surthiu pakiranje uključuje sljedeće korake:

1. odabir podskupa: pakiranje počinje odabirom slučajnog uzorka ili podskupa iz cijelog skupa podataka
2. bootstrap uzorkovanje: svaki se model stvara iz uzoraka koji su poznati pod nazivom Bootstrap uzorci. Takvi uzorci se uzimaju iz izvornih podataka sa mogućnošću zamjene. Ovaj proces je poznat kao uzorkovanje redaka *(engl. row sampling)*
3. bootstrapping: korak uzorkovanja retka sa zamjenom naziva se bootstrapping i principu ovdje se dešava izmjena izvornih podataka sa podatcima zamjene
4. nezavisno treniranje modela: svaki se model samostalno trenira prema odgovarajućem uzorku. Ovaj proces treniranja generira rezultate za svaki model
5. glasovanje većinom: konačni rezultat se predviđa kao najčešće predviđani rezultat koji su donijeli modeli.
6. agregacija: ovdje se kombiniraju svih rezultati i generira se rezultata na temelju većinskog glasovanja.

[39]

Koraci uključeni u algoritam slučajna šuma

1. odabire se podskup podataka i podskup atributa za izradu svakog stabla odlučivanja
2. za svaki uzorak generira se stablo odlučivanja
3. svako stablo odlučivanja daje rezultat (izlaz)

4. konačni rezultat se dobiva većinskim glasovanjem ili preko izračunavanja prosjeka te se time provodi klasifikacija ili regresija podataka.

[39]

Prednosti

- nema problema s prekomjernog prilagođavanja zbog većinskog glasovanja
- ne smetaju mu nulte vrijednosti
- javlja se svojstvo paralelizacije zbog neovisnosti stabala odluke.
- vrlo je stabilan
- održava raznolikost jer se ne uzimaju u obzir svi atributi
- imun je na visokodimenzionalnosti
- ne mora razdvajati podatke tjekom treniranja i testiranja jer će uvijek postojati 30 % podataka koji se ne vide u stablu odlučivanja
- precizna predviđanja
- koristiti se za zadatke klasifikacije i regresije
- prikladn za velike skupove podataka.

[39][40]

Nedostaci

- složen algoritam
- vrijeme treninga je veće
- teško je objasniti rezultat u nekim situacijama
- možda neće dobro funkcionirati s vrlo malim skupovima podataka
- zahtijevaju veliku količinu memorije

[39][40]

Primjeri korištenja algoritma slučajna šuma

Koristi se u raznim domenama kao što su zdravstvo, financije, marketing i e-trgovina zbog svoje svestranosti i robusnosti. Algoritam slučajna šuma naširoko se koristi za zadatke klasifikacije u područjima kao što su otkrivanje neželjene e-pošte, analize osjećaja i predviđanja odljeva korisnika. [40]

3.5.9. Algoritam k-najbližih susjeda *(engl. k-nearest neighbours, KNN)*

Pripada nadziranom algoritma učenja koji se se može upotrijebiti za regresiju ili za klasifikaciju. Za testne podatke se izvršava izračun udaljenosti između testnih podataka i svih točaka skupa za treniranje u svrhu određivanja predikcije točne klase. Bira se k koji je najbliži testnim podacima. Vrši se izračun vjerojatnost u kojima se predviđa da testni podaci pripadaju klasama podataka za treniranje 'k' i bit će odabrana klasa koja ima najveću vrijednost vjerojatnosti.[41]

Rad Algoritma k-najbližih susjeda

Prema Christopheru rad algorima k-najbližih susjeda sastoji se od 6 koraka.

- 1: bira se broj k susjeda
- 2: izračuna se euklidska udaljenost između k broja susjeda
- 3: uzima se broj najbližih susjeda prema izračunatoj euklidskoj udaljenosti
- 4: određuju se klase takvih susjeda
- 5: nova podatkovna točka dobiva klasu prema maksimalnom broju klase susjeda koji su uzeti
- 6: model je spreman za klasifikaciju

[41]

Odabir vrijednosti k

- ne postoji metoda koja bira optimalan k
- bira se slučajna vrijednost k
- mala vrijednost k rezultira nestabilnim granicama odlučivanja
- velike vrijednosti k poboljšavaju granice odlučivanja
- među svim testnim vrijednostima k bira se ona koja ima minimalnu stopu pogreške.[41]

[41]

Izračunavanje udaljenosti:

Prema Christopheru najpoznatije metode izračuna udaljenosti između nove podatkovne točke i broja susjeda su : Euklidova udaljenost, Manhattanova udaljenost i Hamingova udaljenost [41]

Prednosti korištenja algoritma k-najbližih susjeda:

- lako ga je razumjeti i jednostavno implementirati
- može se koristiti i za probleme klasifikacije i regresije
- idealan je za nelinearne podatke budući da nema pretpostavki o temeljnim podacima
- prirodno može obraditi slučajeve s više klasa.

[42]

Nedostatci korištenja algoritma K-najbližih susjeda:

- zahtijeva veliku količinu memorije jer pohranjuje sve podatke tijekom treniranja
- treba odrediti vrijednost K
- osjetljiv je na nebitne podatke.

[42]

4. Skupovi podataka

U ovome poglavlju objašnjen je teorijski dio skupova podataka te preuzeti skupovi podataka koji će se korisititi u komparativnoj analizi klasifikacijski metoda. Započinje jednostavnim objašnjenjem osnovnih pojmova iz skupova podataka. Nakon objašnjenja osnovnih pojmova opisana su 3 skupa podataka. U skupovima su objašnjene varijable i dana je vizualana prezentacija samih podataka uz osnovne statističke analize samih podataka.

4.1. Osnovni pojmovi

Podaci su opažanja i mjerenja. Podaci brojeva su kvantitativni podaci, dok su podaci teksta kvalitativni podaci. [43]

Varijabla se može nazivati i podatkovnom stakom koja može poprimiti neke vrijednosti te se može i mijenjati s obzirom na razne okolnosti. [44]

Numeričke varijable mogu biti kontinuirane ili diskretne:

- kontinuirana varijabla primaju vrijednosti iz skupa realnih brojeva npr vrijednost temperature
- diskretna varijabla primaju vrijednosti cijelih brojeva iz nekog skupa koje je prebrojiv npr broj djece u familiji.

[44]

Kategorijalne varijable opisuju kvalitetu ili karakteristiku podatka te su predstavljene s tekstualnim vrijednostima. [44]

Kategorijalne varijable mogu biti ordinalne ili nominalne:

- **ordinalna varijabla** prima vrijednost koja se može logično poredati ili rangirati npr veličina majce
- **nominalna varijabla** prima vrijednost koja se ne može organizirati u logičan slijed npr boja očiju.

[44]

Nezavisna i zavisna varijabla dobivaju se korištenjem bivarijatne analize

- **nezavisna varijabla** (ili prediktorska varijabla) utječe ili predviđa drugu varijablu
- **zavisna varijabla** (ili varijabla ishoda) na nju se utječe ili se predviđa.

[43]

Skup podataka je skup podataka koji se odnose na određenu temu. Skupovi podataka mogu poprimati različite vrste informacija (npr. slike, tekst). Pohranjuju se u raličitim formatima kao što su: CSV, JSON ili SQL. [45]

Prosječna vrijednost *(engl. Mean)* je vrijednost koja se dobiva zbrajanjem svih vrijednost te se takav zbroj podjeli s brojem vrijednosti. **Medijan** *(engl. Median)* je vrijednost broja koja se pojavljuje u sredini sortiranih brojeva (broj u sredini sortirane liste). **Mod** *(engl. Mode)* je vrijednost broja koji se najčešće pojavljuje u nekom skupu brojeva [46]

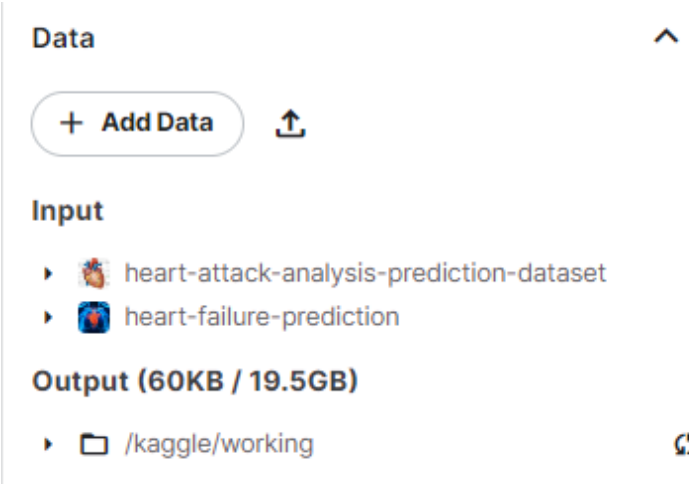
Standardna devijacija *(engl. Standard Deviation)* je vrijednost koja prikazuje koliko su vrijednosti udaljene od prosječne vrijednosti. Ona se označava sa grčkim slovom σ te se dobiva korijenovanjem varijance. [47]

Percentil *(engl. Percentiles)* je vrijednost broja koja opisuje koliko se posto manjih vrijednosti nalazi prije takve postotne vrijednosti. [48]

4.2. Priprema skupova podataka za vizualizaciju

Vizalizaciju podataka te neke statističke analize su napravljene preko Rob Mullerovom youtube poučku `https://www.youtube.com/watch?v=xi0vhXFPegw`. U ovome poučku je dano objašnjenje kako se radi Eksploratorna analiza podataka. Rob Mulla na početku daje objašnjenje o bibliotekama koje se trebaju uključiti: Panda biblioteka je već uključena prilikom izrade nove bilježnice u kaggelu *(engl kaggle notebook)*, numpy je biblioteka koja se koristi za razne matematičke operacije , matplotlib za vizalizaciju podataka, seaborn za vizalizaciju podataka te se uvodi stil ggplot za lijepši prikaz grafikona. Nakon toga se dodaju skupovi podataka i koristi se pd.read_csv(putanjadoskupa) funkcija. Rob Mulla također daje objašnjenje o raznim funkcijama kao što su: shape, head, dtypes, describe, isna().sum(). Nakon objašnjenja o samim funkcijam prikazuje kako se koriste određeni vizalni prikazi podataka. [49]

Na slici ispod je vizualni prikaz kako se dodaju skupovi podataka u kaggleovu bilježnicu. Pritisne se na gumb +add data te se dodaju skupovi podataka. Na slici se može primjetiti da su dodani skupovi: o predviđanju i analizi srčanog udara te predviđanju zatajenja srca.



Slika 11: Prikaz dodavanje skupova podataka u kaggle bilježnici(samostalna izrada)

Na slici ispod može se vidjeti dodane biblioteke numpy, pandas, matplotlib i seaborn te stil prikaza ggplot za grafikone. Nakon dodaje skupova podataka pomoću pd.read_csv daju se detaljna objašnjenja o skupovima koji će se koristiti u klasifikaciji podataka.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pylab as plt
import seaborn as sns
plt.style.use('ggplot')
```

```
zatajenjeSrca = pd.read_csv('../input/heart-failure-prediction/heart.csv')
srcaniUdar= pd.read_csv('../input/heart-attack-analysis-prediction-dataset/heart.csv')
```

Slika 12: Prikaz unosa biblioteka i dohvaćanja skupova podataka (samostalna izrada)

4.3. Skup podataka o predviđanju i analizi srčanog udara *(engl. Heart Attack Analysis & Prediction Dataset)*

Skup podataka o predviđanju i analizi srčanog udara preuzet je s kaggle.com-a te je dostupan preko sljedeće poveznice `https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset`.

Skup podataka se sastoji od 14 atributa:

- dob: Dob osobe *(engl. age: Age of the person)*
- spol: Spol osobe *(sex: Gender of the person)*
- cp: Vrsta boli u prsima *(cp: Chest Pain type)*

  vrijednost 1: tipična angina

  vrijednost 2: atipična angina

  vrijednost 3: neanginozna bol

  vrijednost 4: asimptomatski
- trtbps: krvni tlak u mirovanju (u mm Hg) *(engl. trtbps:resting blood pressure (in mm Hg))*
- khol: kolestoralni u mg/dl dohvaćen preko BMI senzora *(chol: cholestoral in mg/dl fetched via BMI sensor)*
- fbs: (šećer u krvi natašte > 120 mg/dl)*(engl. fbs: (fasting blood sugar > 120 mg/dl))* (1 = točno; 0 = netočno)
- resecg: rezultati elektrokardiografije u mirovanju *(restecg: resting electrocardiographic results)*

  vrijednost 0: normalno

  vrijednost 1: abnormalnost ST-T vala (inverzija T vala i/ili ST elevacija ili depresija > 0,05 mV)

vrijednost 2: pokazuje vjerojatnu ili sigurnu hipertrofiju lijeve klijetke prema Estesovim kriterijima

- thalachh: maksimalni postignuti broj otkucaja srca *(thalachh: maximum heart rate achieved)*
- exng: angina izazvana naporom *(engl. exng: exercise induced angina)*(1 = da; 0 = ne)
- oldpeak: prethodni vrh *(engl. oldpeak: Previous peak)*
- slp: Nagib (slp: Slope)
- caa: broj glavnih tkiva (0-3) *(caa: number of major vessels)*
- thall: Thal stopa *(engl. thall: Thal rate)*
- izlaz: ciljna varijabla *(engl. output: Target variable)* 0= manje šanse za srčani udar 1= veće šanse za srčani udar.

[50]

Na slici ispod može se vidjeti prvih 5 readaka u skupu podataka pomoću funkcije head(). U ovakvom tabličnom prikazu vide se spomenutih 14 atributa koji su prikazani kao stupci.

```
[9]: srcaniUdar.head()
```

[9]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Slika 13: Tablični prikaz skupa podataka o predviđanju i analizi srčanog udara (samostalna izrada)

Na slici ispod pomoću funkcije shape vraćaju se dvije brojke. 303 govori da se skup sastoji od 303 readaka, a 14 govori da se skup sastoji od 14 atributa.

```
[10]: srcaniUdar.shape

[10]: (303, 14)
```

Slika 14: Prikaz rada shape funkcije (samostalna izrada)

Funkcija dtypes vraća tipove atribute. Može se vidjeti da vraća tipove int64 i float64 iz kojih se može zaključiti da se radi o potpuno numeričkom tipu skupa podataka odnosno da su svi podaci kvantitivni.

```
srcaniUdar.dtypes
```

```
age           int64
sex           int64
cp            int64
trtbps        int64
chol          int64
fbs           int64
restecg       int64
thalachh      int64
exng          int64
oldpeak     float64
slp           int64
caa           int64
thall         int64
output        int64
dtype: object
```

Slika 15: Prikaz tipova atributa (samostalna izrada)

Funkcija describe opisuje skup podataka preko različitih parametara, a to su: prosjek, standardna devijacija, minimalna vrijednosti, 25% percentil, 50% percentil, 75% percentil, maksimalna vrijednost. Prema primjeru iz tablice. Dob se može vidjeti da joj je prosječna vrijednost 54.4 godine, standardna devijacija odnosno mijera odstupanja od prosječne vrijednosti iznosi 9.1. Najmlađi pacijent ima 29 godina, a najstariji 77 godina. 25% posto osoba ima manje od 47.5 godine, 50% osoba ima manje od 55 godine i 75% osoba ima manje od 61 godine. Prema opisanim podacima da se naslutiti da osobe imaju normalnu raspodjelu godina. Dok kod primjerice kolesterola smatram da će se pojaviti ekstremne vrijednosti.

```
srcaniUdar.describe()
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Slika 16: Prikaz rada funkcije describe (samostalna izrada)

Funkcija isna().sum() vraća sumu nultih vrijednosti prema atributima. Prema prikazanom vidi se da ne postoji nulta vrijednost u skupu.

```
srcaniUdar.isna().sum()
```

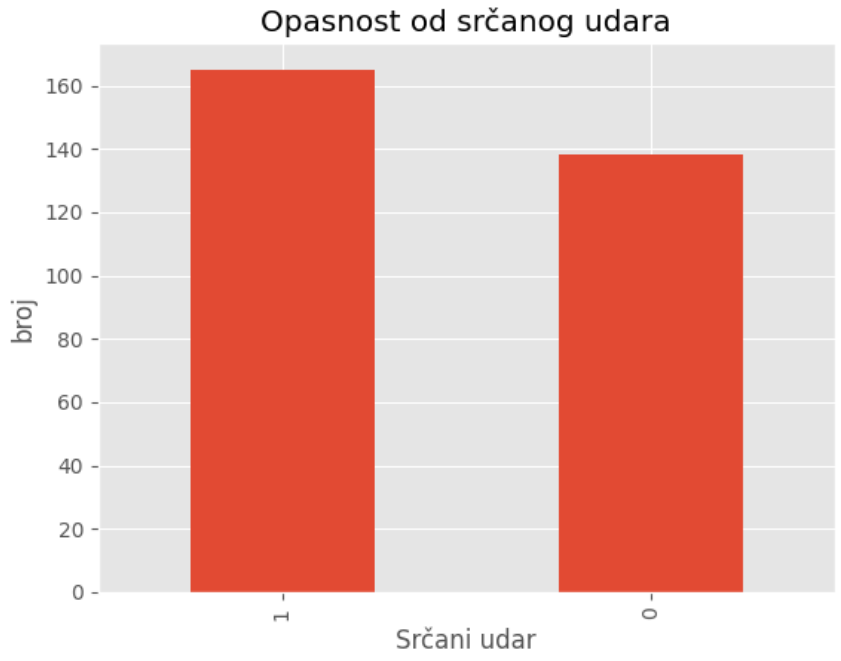
```
age         0
sex         0
cp          0
trtbps      0
chol        0
fbs         0
restecg     0
thalachh    0
exng        0
oldpeak     0
slp         0
caa         0
thall       0
output      0
dtype: int64
```

Slika 17: Prikaz nultih vrijednosti u skupu podataka o predviđanju i analizi srčanog udara (samostalna izrada)

Na stupčastom grafikonu ispod dan je prikaz distribucije ljudi koji su u opanosti od dobivanja srčanog udara. Vrijednost 0 prikazuje da odprilike 140 ljudi imaju nisku razinu opasnosti od dobivanja srčanog udara odnosno vrijednost 1 prikazuje odprilike 160 ljudi koji imaju visoku razinu opasnosti od dobivanja srčanog udara. Može se primjetiti da je distribucija opasnosti od srčanog udara izbalansirana.

```
osiGrafaSrcaniUdar = srcaniUdar['output'].value_counts()\
    .plot(kind='bar', title='Opasnost od srčanog udara')
osiGrafaSrcaniUdar.set_xlabel('Srčani udar')
osiGrafaSrcaniUdar.set_ylabel('broj')
```

```
Text(0, 0.5, 'broj')
```

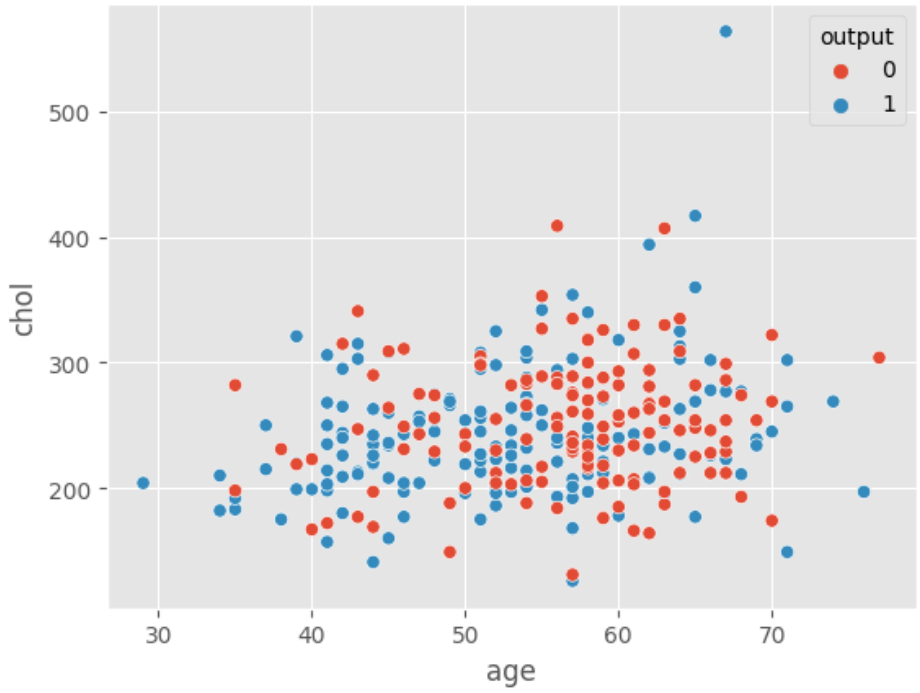


Slika 18: Stupčasti grafikon opasnot od srčanog udara (samostalna izrada)

Na grafikonu raspršenosti može se vidjeti odnos podataka na x osi koji predstavlja dob pacijenta te na y osi kolesterol. Dodavanjem hue="output" dodaje se boja podatkovnim točkama u kojoj plave točke imaju visoku razinu opasnosti od srčanog udara dok crvene točke imaju nisku razinu opasnosti od srčanog udara. Ovakav tip grafikona je dobar zato što se može vidjeti pojava ekstremnih vrijednosti poput ove plave točke koja ima kolesterol preko 500.

```
sns.scatterplot(x='age',
                y='chol',
                hue='output',
                data=srcaniUdar)
```

```
<Axes: xlabel='age', ylabel='chol'>
```



Slika 19: Grafikon raspršenosti(samostalna izrada)

```
sns.pairplot(srcaniUdar, vars=['age','sex','cp','trtbps','chol'],
             hue='output')
```

Slika 20: Funkcija pairplot(samostalna izrada)

Pomoću funkcije sns.pairplot() stvara se dijagram u kojim se vidi odnos parova atributa. U ovom primjeru dani su odnosi između dobi, spola, vrsta boli u prsima, krvnog tlaka u mirovanju i kolesterola. Dodavanjem parametra hue="output" opet se prikazuje kao i na prošlom primjeru opasnost od dobivanje srčanog udara u bojama. Na ovakvom dijagramu mogu se primjetiti zanimljivi uzorci iz odnosa među varijablama primjerice može se primjetiti da u odnosu spola i dobi najveću opasnost od srčanog udara ima spol vrijednosti 0 te da s obzirom na dob se može primjetiti da osobe od 60 godina imaju nisku razinu opasnosti od srčanog udara. Može se primjetiti da kod ekstremnih odnosno vrlo visokih vrijednosti kolesterola uvijek se pojavljuje visok rizik od srčanog udara. Ovakva analiza je korisna za pronalazak zanimljivih uzoraka.

Slika 21: Grafikon odnosa parova(samostalna izrada)

Funkcija corr() stvara tablicu korelacija koja daje informaciju o zavisnostima između atributa. Naravno zbog toga dijagonala koja uspoređuje iste atribute ima uvijek vrijednost 1. U prikazanoj tablici se može primjetiti da su najveće apslutne vrijednosti zavisnosti oko 0.4, a najmanje oko 0.01. Takve zavisnosti se mogu smatrati slabim zavinostima odnosno može se reći da skup podataka sastoji od nezavisnih odnosno jako slabo zavisnih atributa.

[69]:
```
corrSrcaniUdar=srcaniUdar.corr()
corrSrcaniUdar
```

[69]:

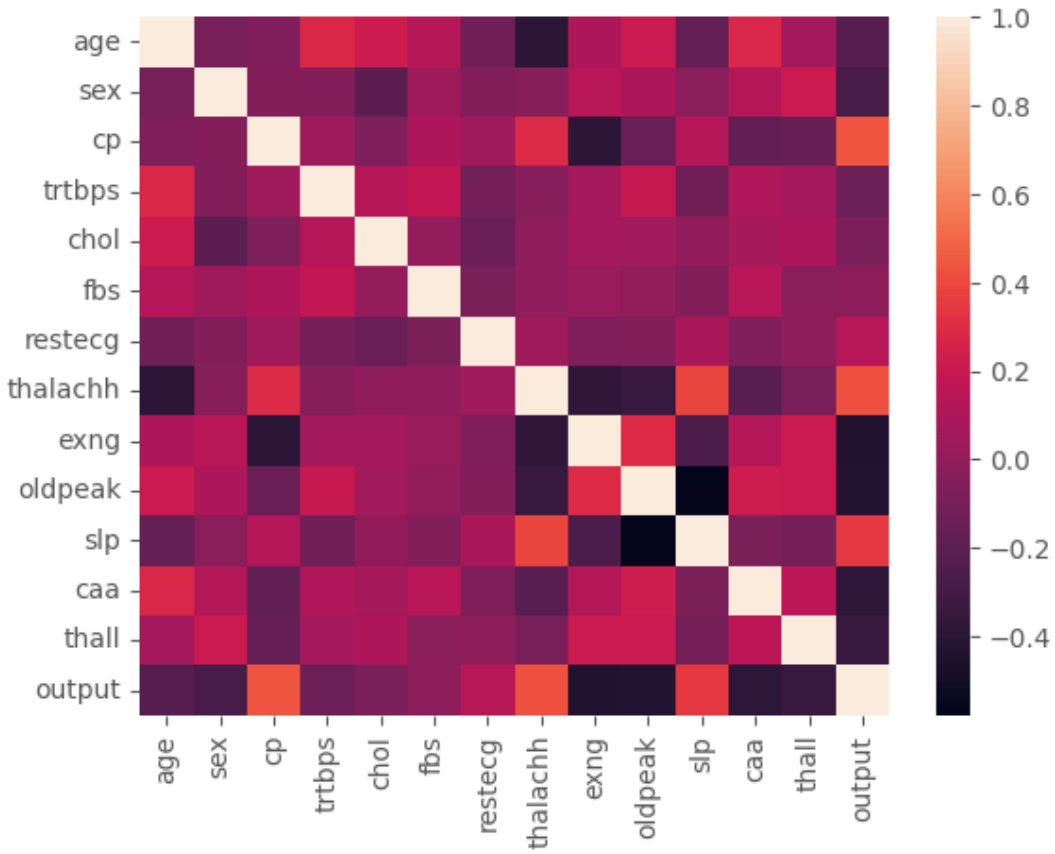
	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522	0.096801	0.210013	-0.168814	0.276326	0.068001	-0.225439
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020	0.141664	0.096093	-0.030711	0.118261	0.210041	-0.280937
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762	-0.394280	-0.149230	0.119717	-0.181053	-0.161736	0.433798
trtbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698	0.067616	0.193216	-0.121475	0.101389	0.062210	-0.144931
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940	0.067023	0.053952	-0.004038	0.070511	0.098803	-0.085239
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567	0.025665	0.005747	-0.059894	0.137979	-0.032019	-0.028046
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123	-0.070733	-0.058770	0.093045	-0.072042	-0.011981	0.137230
thalachh	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000	-0.378812	-0.344187	0.386784	-0.213177	-0.096439	0.421741
exng	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812	1.000000	0.288223	-0.257748	0.115739	0.206754	-0.436757
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187	0.288223	1.000000	-0.577537	0.222682	0.210244	-0.430696
slp	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784	-0.257748	-0.577537	1.000000	-0.080155	-0.104764	0.345877
caa	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177	0.115739	0.222682	-0.080155	1.000000	0.151832	-0.391724
thall	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439	0.206754	0.210244	-0.104764	0.151832	1.000000	-0.344029
output	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741	-0.436757	-0.430696	0.345877	-0.391724	-0.344029	1.000000

Slika 22: Tablica korelacije (samostalna izrada)

sns.heatmap() je funkcija koja omogućuje prikaz prijašnje tablice pomoću takozvane toplinske mape. Iz takve toplinske mape prema bojama po skali sa desne strane može se primjetiti da su najveće apslutne vrijednosti zavisnosti oko 0.4, a najmanje oko 0.01. Takve zavisnosti se mogu smatrati slabim zavinostima odnosno može se reći da skup podataka sastoji od nezavisnih odnosno jako slabo zavisnih atributa. Toplinske mape su možda i ugodniji prikaz zavisnosti prilikom prezentacije podataka.

```
sns.heatmap(corrSrcaniUdar)
```

```
<Axes: >
```



Slika 23: Toplinska mapa (samostalna izrada)

4.4. Skup podataka o predviđanju zatajenja srca *(Heart Failure Prediction Dataset)*

Skup podataka o predviđanju zatajenja srca prezet je s kaggle.com-a te je dostupan preko sljedeće poveznice

`https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction`

Skup podataka se sastoji od 12 atributa:

- Dob: dob pacijenta [godine] *(engl. Age: age of the patient)*
- Spol: spol pacijenta [M: muškarac, Ž: žena] *(engl. Sex: sex of the patient)*
- ChestPainType: vrsta boli u prsima [TA: tipična angina, ATA: atipična angina, NAP: neanginozna bol, ASY: asimptomatska] *(engl.ChestPainType)*
- BP u mirovanju: krvni tlak u mirovanju [mm Hg] *(engl. RestingBP: resting blood pressure [mm Hg])*
- Kolesterol: kolesterol u serumu [mm/dl] *(engl. Cholesterol: serum cholesterol [mm/dl])*
- FastingBS: šećer u krvi natašte [1: ako je FastingBS > 120 mg/dl, 0: inače] *(engl.FastingBS: fasting blood sugar)*
- EKG u mirovanju: rezultati elektrokardiograma u mirovanju [Normalno: normalno, ST: abnormalnost ST-T vala (inverzije T vala i/ili ST elevacija ili depresija > 0,05 mV), LVH: pokazuje vjerojatnu ili sigurnu hipertrofiju lijeve klijetke prema Estesovim kriterijima] *(engl.RestingECG: resting electrocardiogram results)*
- MaxHR: maksimalni postignuti broj otkucaja srca [Brojna vrijednost između 60 i 202] *(engl.MaxHR: maximum heart rate achieved)*
- Angina vježbanja: angina izazvana vježbanjem [Y: Da, N: Ne] *(engl. ExerciseAngina: exercise-induced angina)*
- Oldpeak: oldpeak = ST [Numerička vrijednost izmjerena u depresiji] *(engl. Oldpeak: oldpeak = ST)*
- ST_Slope: nagib ST segmenta vršne vježbe [gore: uzlazni, ravno: ravno, dolje: silazni] *(engl. ST_Slope: the slope of the peak exercise ST segment)*
- HeartDisease: izlazna klasa [1: bolest srca, 0: normalno] *(engl. HeartDisease: output class)*.

[51]

Na slici ispod može se vidjeti prvih 5 readaka u skupu podataka pomoću funkcije head(). U ovakvom tabličnom prikazu vide se spomenutih 12 atributa koji su prikazani kao stupci.

```
zatajenjeSrca.head()
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

Slika 24: Tablični prikaz skupa podataka o predviđanju zatajenja srca (samostalna izrada)

Na slici ispod pomoću funkcije shape vraćaju se dvije brojke. 918 govori da se skup sastoji od 918 readaka, a 12 govori da se skup sastoji od 12 atributa.

```
zatajenjeSrca.shape
```

```
(918, 12)
```

Slika 25: Rad funkcije shape (samostalna izrada)

Funkcija dtypes vraća tipove atributa. Može se vidjeti da vraća tipove int64, float64 i object. Object najčešće predstavlja tekstualni tip podatka. Iz kojih se može zaključiti da se radi o skupu podataka koji sadrži kvantitative i kvalitativne tipove podataka.

```
zatajenjeSrca.dtypes
```

```
Age                int64
Sex               object
ChestPainType     object
RestingBP          int64
Cholesterol        int64
FastingBS          int64
RestingECG        object
MaxHR              int64
ExerciseAngina    object
Oldpeak          float64
ST_Slope          object
HeartDisease       int64
dtype: object
```

Slika 26: Tipovi podataka iz skupa podataka o predviđanju zatajenja srca(samostalna izrada)

Funkcija describe opisuje skup podataka preko različitih parametara, a to su: prosjek, standardna devijacija, minimalna vrijednosti, 25% percentil, 50% percentil, 75% percentil, maksimalna vrijednost. Prema primjeru iz tablice. Dob se može vidjeti da joj je prosječna vrijednost 53.5 godine, standardna devijacija odnosno mijera odstupanja od prosječne vrijednosti iznosi 9.4. Najmlađi pacijent ima 28 godina, a najstariji 77 godina. 25% posto osoba ima manje od 47 godina, 50% osoba ima manje od 54 godine i 75% osoba ima manje od 60 godina. Prema opisanim podacima da se naslutiti da osobe imaju normalnu raspodjelu godina. Dok kod primjerice kolesterola smatram da će se pojaviti ekstremne vrijednosti.

```
zatajenjeSrca.describe()
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

Slika 27: Rad funkcije describe nad skupa podataka o predviđanju zatajenja srca (samostalna izrada)

Funkcija isna().sum() vraća sumu nultih vrijednosti prema atributima. Prema prikazanom vidi se da ne postoji nulta vrijednost u skupu.

```
zatajenjeSrca.isna().sum()
```

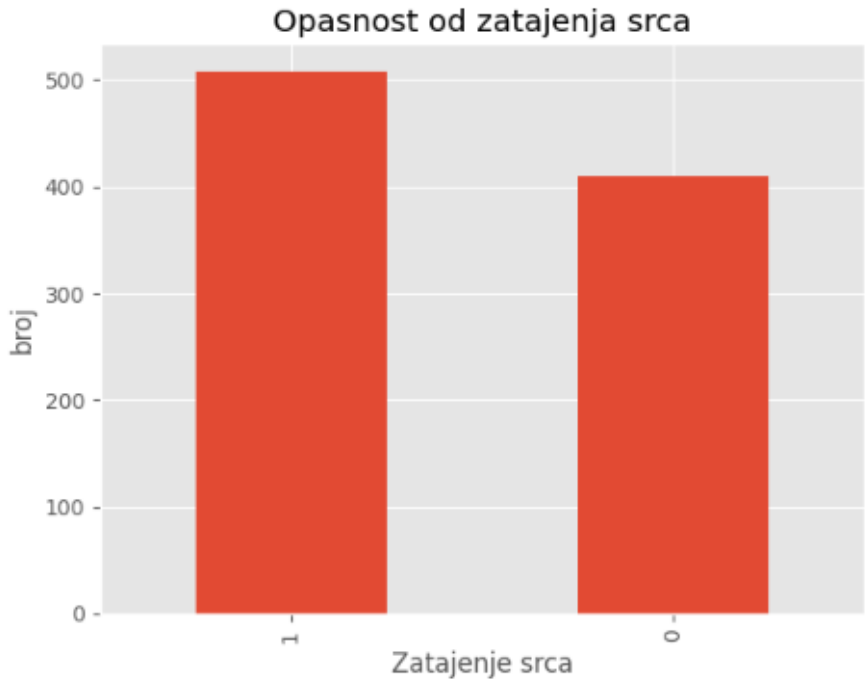
```
Age             0
Sex             0
ChestPainType   0
RestingBP       0
Cholesterol     0
FastingBS       0
RestingECG      0
MaxHR           0
ExerciseAngina  0
Oldpeak         0
ST_Slope        0
HeartDisease    0
dtype: int64
```

Slika 28: Prikaz nultih vrijednosti u skupa podataka o predviđanju zatajenja srca(samostalna izrada)

Na stupčastom grafikonu ispod dan je prikaz distribucije ljudi koji su u opanosti od zatajenja srca. Vrijednost 0 prikazuje da odprilike 400 ljudi imaju nisku razinu opasnosti od zatajenja srca odnosno vrijednost 1 prikazuje odprilike 500 ljudi koji imaju visoku razinu opasnosti zatajenja srca. Može se primjetiti da je distribucija opasnosti zatajenja srca izbalansirana.

```
osiGrafaZatajenjeSrca = zatajenjeSrca['HeartDisease'].value_counts()\
    .head(10) \
    .plot(kind='bar', title='Opasnost od zatajenja srca')
osiGrafaZatajenjeSrca.set_xlabel('Zatajenje srca')
osiGrafaZatajenjeSrca.set_ylabel('broj')
```

```
Text(0, 0.5, 'broj')
```



Slika 29: Stupčasti grafikon o opasnosti zatajenja srca (samostalna izrada)

Na grafikonu raspršenosti može se vidjeti odnos podataka na x osi koji predstavlja dob pacijenta te na y osi kolesterol. Dodavanjem hue="output" dodaje se boja podatkovnim točkama u kojoj plave točke imaju visoku razinu opasnosti od zatajenja srca dok crvene točke imaju nisku razinu opasnosti od zatajenja srca. Ovakav tip grafikona je dobar zato što se može vidjeti pojava ekstremnih vrijednosti poput ovih plavih točaka koje imaju kolesterol preko 500.

```
sns.scatterplot(x='Age',
                y='Cholesterol',
                hue='HeartDisease',
                data=zatajenjeSrca)
```

```
<Axes: xlabel='Age', ylabel='Cholesterol'>
```

Slika 30: Grafikon raspršenost iz skupa podataka o predviđanju zatajenja srca(samostalna izrada)

Funkcija corr() stvara tablicu korelacija koja daje informaciju o zavisnostima između atributa. Naravno zbog toga dijagonala koja uspoređuje iste atribute ima uvijek vrijednost 1. U prikazanoj tablici se može primjetiti da su najveće apslutne vrijednosti zavisnosti oko 0.4, a najmanje oko 0.05. Takve zavisnosti se mogu smatrati slabim zavinostima odnosno može se reći da skup podataka sastoji od nezavisnih odnosno jako slabo zavisnih atributa. Važno je naglasiti da funkcija corr() prima samo numeričke vrijednosti.

```
corrZatajenjeSrca=zatajenjeSrca[['Age','RestingBP','Cholesterol','FastingBS','MaxHR','Oldpeak','HeartDisease']].corr()
corrZatajenjeSrca
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
Age	1.000000	0.254399	-0.095282	0.198039	-0.382045	0.258612	0.282039
RestingBP	0.254399	1.000000	0.100893	0.070193	-0.112135	0.164803	0.107589
Cholesterol	-0.095282	0.100893	1.000000	-0.260974	0.235792	0.050148	-0.232741
FastingBS	0.198039	0.070193	-0.260974	1.000000	-0.131438	0.052698	0.267291
MaxHR	-0.382045	-0.112135	0.235792	-0.131438	1.000000	-0.160691	-0.400421
Oldpeak	0.258612	0.164803	0.050148	0.052698	-0.160691	1.000000	0.403951
HeartDisease	0.282039	0.107589	-0.232741	0.267291	-0.400421	0.403951	1.000000

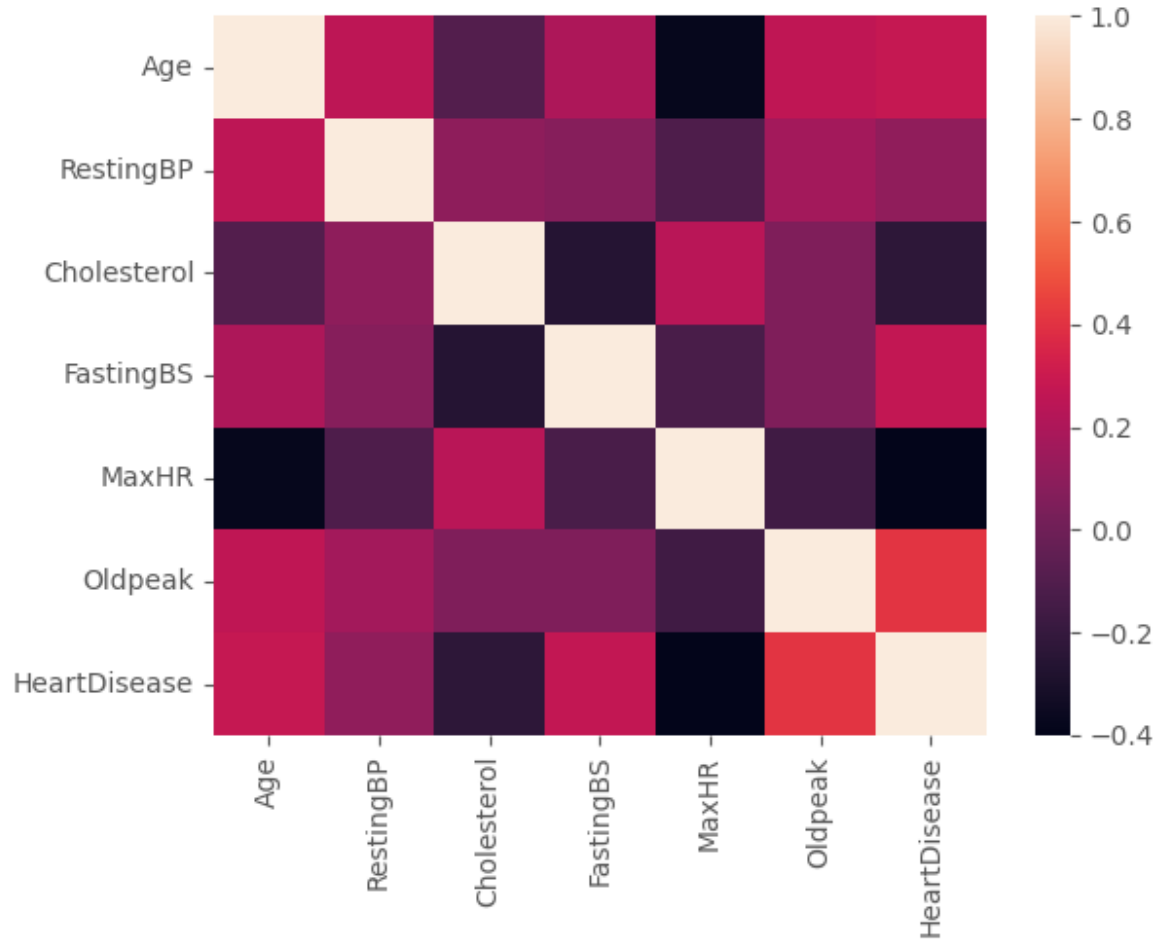
Samo prima numeričke vrijednosti .corr()

Slika 31: Korelacijska tablica iz skupa podataka o predviđanju zatajenja srca(samostalna izrada)

sns.heatmap() je funkcija koja omogućuje prikaz prijašnje tablice pomoću takozvane toplinske mape. Iz takve toplinske mape prema bojama po skali sa desne strane može se primjetiti da su najveće apslutne vrijednosti zavisnosti oko 0.4, a najmanje oko 0.05. Takve zavisnosti se mogu smatrati slabim zavinostima odnosno može se reći da skup podataka sastoji od nezavisnih odnosno jako slabo zavisnih atributa. Toplinske mape su možda i ugodniji prikaz zavisnosti prilikom prezentacije podataka.

```
sns.heatmap(corrZatajenjeSrca)
```

```
<Axes: >
```



Slika 32: Toplinska mapa iz skupa podataka o predviđanju zatajenja srca (samostalna izrada)

5. Komparativna analiza klasifikacijskih metoda

Komparativna analiza se zapravo bavi usporedbom klasifikacijskih metoda preko određenih parametara koje ocjenjuju same klasifikacijske metode nad određenim skupovima podataka.

Točnost *(engl. Accuracy)* predstavlja omjer ukupnog broja ispravnih instanci prema ukupnom broju instanci. Točnost je dana sljedećom formulom: [52]

$$\frac{TP + TN}{TP + TN + FP + FN}$$

gdje je TP oznaka za pozitivnu istinu*(engl. True positives)*,TN je oznaka za negativnu istinu *(engl. True negatives)*, FP je oznaka za pozitivnu laž *(engl. False positives)*, a FN je oznaka za negativnu laž *(engl. False negatives)*. [52]

Preciznost *(engl. Precision)* predstavlja omjer istinskih pozitivnih predviđanja prema ukupnom broju pozitivnih predviđanja. Preciznost je dana sljedećem formulom: [52]

$$\frac{TP}{TP + FP}$$

Odaziv *(engl. Recall)* predstavlja omjer broja istinski pozitivnih predviđanja prema zbroju pravih pozitivnih i lažno negativnih predviđanja. Odaziv je dan sa sljedećom formulom: [52]

$$\frac{TP}{TP + FN}$$

F1-rezultat (engl. F1-score) predstavlja harmonijsku sredinu preciznosti i odaziva.F1-rezultat je dan sljedećem formulom: [52]

$$\frac{2 * Preciznost * Odaziv}{Preciznost + Odaziv}$$

Klasifikacija skupova podataka je napravljena uz pomoću Chanin Nantasenamat youtube poučka `https://www.youtube.com/watch?v=dRqtLxZVRuw`. Chanin Nantasenamat poznat je i pod nazivom Data Professor. U svom poučku on objašnjava kako izgraditi klasifikacijski model u r programskom jeziku. Na početku daje objašnjenje da treba uključiti caret biblioteku koja radi na problemima klasifikacijskog i regresiskog treniranja. Nakon toga daje objašnjenje da treba skupove podataka podjeliti na skupove treniranja i testiranja. Skup treniranja čini većinski dio nad kojim se gradi klasifikacijski model te će se preko takvog modela izvršavati klasifikacija na skupu za testiranje. [53]

Na početku programskog koda uključio sam biblioteku caret te sam ostavio generički kod koji se stvara prilikom kreiranja nove kaggleove bilježnice , a radi se o uključivanju tidyverse biblioteke te o naredbe list files koja ispisuje sve podatkovene skupove koje smo dodali u kaggleovu bilježnicu. Naravno radi se o podatkovnim skupovima koji su opisani u prethodnom poglavlju.

```
library(tidyverse) # metapackage of all tidyverse packages
list.files(path = "../input")
library(caret)
```

Izlaz: 'heart-attack-analysis-prediction-dataset"heart-failure-prediction'

Nakon uključivanja caret biblioteke potrebno je učitavanje skupova podataka pomoću read_csv() naredbe. Skupovi su spremljeni u zastajenjeSrca i srcaniUdar. Potrebna je bila konverzija atributa HeartDisease i output u faktorski tip atributa zbog naredbe train i predict (inače vraća Error)

```
zatajenjeSrca <- read_csv('../input/heart-failure-prediction/heart.csv')
zatajenjeSrca$HeartDisease<-factor(zatajenjeSrca$HeartDisease)
srcaniUdar <- read_csv('../input/heart-attack-analysis-prediction-dataset/heart.csv'
    )
srcaniUdar$output<-factor(srcaniUdar$output)
```

5.1. Klasifikacija u skupom podataka o predviđanju zatajenja srca

5.1.1. Kreiranje particija

Kao što je već spomenuto potrebana je kreiranje particija skupova pomoću naredbe createDataPartition(). U njoj naređujemo da glavni atribut po kojem će se klasificirati skup podataka bude HeartDisease te 70% skupa podataka dodjeljeno skupu pod nazivom skupTreniranjaZatajenjeSrca, a 30% skupu pod nazivom skupTestiranjaZatajenjeSrca. Treba se postaviti sjeme da kad ponovno izvrtimo simulacije skupovi budu jednaki zbog toga što o njima ovise i rezultati same klasifikacije podataka. List=False je postavljen kao zadnji parametar u createDataPartition-u zato što se ne želi dobivati liste.

```
set.seed(3012)
skupZatajenjaSrca <- createDataPartition(zatajenjeSrca$HeartDisease, p=0.7, list=
    FALSE)
skupTreniranjaZatajenjeSrca <- zatajenjeSrca[skupZatajenjaSrca,]
skupTestiranjaZatajenjeSrca <- zatajenjeSrca[-skupZatajenjaSrca,]
```

5.1.2. Logistička regresija

Prilikom izrade klasifikacije logističke regresije prvo se trenira klasifikacijski model pod nazivom ModelLogistickaRegresijaZatajenjaSrca pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da HeartDisease bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaZatajenjeSrca govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaZatajenjeSrca. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva glm te pripada binomialnoj obitelji. Zapravo se radi o binarnoj logističkoj regresiji. Nakon treninga se izvršava klasifikacija nad skupTestiranjaZatajenjeSrca pomoću klasifikacijskog modela ModelLogistickaRegresijaZatajenjaSrca te se dobiva ModelLogistickaRegresijaZatajenjaSrcaTestiranje

pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelLogistickaRegresijaZatajenjaSrcaTestiranje pomoću skupTestiranjaZatajenjeSrca$HeartDisease te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda Logisticka regresija
ModelLogistickaRegresijaZatajenjaSrca <- train(HeartDisease~.,data=
    skupTreniranjaZatajenjeSrca,method="glm",family="binomial")

ModelLogistickaRegresijaZatajenjaSrcaTestiranje <- predict(
    ModelLogistickaRegresijaZatajenjaSrca, skupTestiranjaZatajenjeSrca)
ModelLogistickaRegresijaZatajenjaSrcaTestiranjeConf <- confusionMatrix(
    ModelLogistickaRegresijaZatajenjaSrcaTestiranje, skupTestiranjaZatajenjeSrca$
    HeartDisease, mode = "everything")
print("Matrica zbunjivanja za Logisticku regresiju u skupu podataka Zatajenje Srca")
print(ModelLogistickaRegresijaZatajenjaSrcaTestiranjeConf)
```

Rezultat:

```
[1] "Matrica zbunjivanja za Logisticku regresiju u skupu podataka Zatajenje Srca"
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 106  21
         1  17 131

               Accuracy : 0.8618
                 95% CI : (0.8153, 0.9003)
    No Information Rate : 0.5527
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7214

 Mcnemar's Test P-Value : 0.6265

            Sensitivity : 0.8618
            Specificity : 0.8618
         Pos Pred Value : 0.8346
         Neg Pred Value : 0.8851
              Precision : 0.8346
                 Recall : 0.8618
                     F1 : 0.8480
             Prevalence : 0.4473
         Detection Rate : 0.3855
   Detection Prevalence : 0.4618
      Balanced Accuracy : 0.8618

       'Positive' Class : 0
```

Slika 33: Klasifikacija logističkom regresijom u skupu podataka o predviđanju zatajenja srca(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom logističke regresije daje sljedeće rezultate parametara koji će se koristi u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8618, preciznost = 0.8346, odaziv = 0.8618 i f-1 rezultat = 0.8480.

5.1.3. Metoda potpornih vektora

Prilikom izrade klasifikacije metodom potpornih vektora prvo se trenira klasifikacijski model pod nazivom ModelMetodaPotpornihVektoraZatajenjaSrca pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da HeartDisease bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaZatajenjeSrca govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaZatajenjeSrca. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva svmLinear2. Zapravo se radi o metodi potpornih vektora s linearnom jezgrom. Nakon treninga se izvršava klasifikacija nad skupTestiranjaZatajenjeSrca pomoću klasifikacijskog modela ModelMetodaPotpornihVektoraZatajenjaSrca te se dobiva ModelMetodaPotpornihVektoraZatajenjaSrcaTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelMetodaPotpornihVektoraZatajenjaSrcaTestiranje pomoću skupTestiranjaZatajenjeSrca$HeartDisease te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda potpornih vektora s linearnom jezgrom
ModelMetodaPotpornihVektoraZatajenjaSrca <- train(HeartDisease ~.,data=
    skupTreniranjaZatajenjeSrca,
                              method="svmLinear2")
ModelMetodaPotpornihVektoraZatajenjaSrcaTestiranje <- predict(
    ModelMetodaPotpornihVektoraZatajenjaSrca, skupTestiranjaZatajenjeSrca)
ModelMetodaPotpornihVektoraZatajenjaSrcaTestiranjeConf <- confusionMatrix(
    ModelMetodaPotpornihVektoraZatajenjaSrcaTestiranje, skupTestiranjaZatajenjeSrca$
    HeartDisease, mode = "everything")
print("Matrica␣zbunjivanja␣za␣Metodu␣potpornih␣vektora␣u␣skupu␣podataka␣Zatajenje␣
    Srca")
print(ModelMetodaPotpornihVektoraZatajenjaSrcaTestiranjeConf)
```

```
[1] "Matrica zbunjivanja za Metodu potpornih vektora u skupu podataka Zatajenje Srca"
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 105  19
         1  18 133

               Accuracy : 0.8655
                 95% CI : (0.8193, 0.9035)
    No Information Rate : 0.5527
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7281

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.8537
            Specificity : 0.8750
         Pos Pred Value : 0.8468
         Neg Pred Value : 0.8808
              Precision : 0.8468
                 Recall : 0.8537
                     F1 : 0.8502
             Prevalence : 0.4473
         Detection Rate : 0.3818
   Detection Prevalence : 0.4509
      Balanced Accuracy : 0.8643

       'Positive' Class : 0
```

Slika 34: Klasifikacija metodom potpornih vektora u skupom podataka o predviđanju zatajenja srca(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom potpornih vektora daje sljedeće re-

zultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8111, preciznost = 0.8, odaziv = 0.7805 i f-1 rezultat = 0.7901.

5.1.4. Umjetne neuronske mreže

Prilikom izrade klasifikacije logističke regresije prvo se trenira klasifikacijski model pod nazivom ModelMetodaUmjetneNeuronskeMrežeZatajenjaSrca pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da HeartDisease bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaZatajenjeSrca govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaZatajenjeSrca. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva nnet. Zapravo se radi o umjetnim neuronskim mrežama. Nakon treninga se izvršava klasifikacija nad skupTestiranjaZatajenjeSrca pomoću klasifikacijskog modela ModelMetodaUmjetneNeuronskeMrežeZatajenjaSrca te se dobiva ModelMetodaUmjetneNeuronskeMrežeZatajenjaSrcaTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelMetodaUmjetneNeuronskeMrežeZatajenjaSrcaTestiranje pomoću skupTestiranjaZatajenjeSrca$HeartDisease te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda Umjetne neuronske žmreež
ModelMetodaUmjetneNeuronskeMreeZatajenjaSrca <- train(HeartDisease ~.,data=
    skupTreniranjaZatajenjeSrca,
                              method="nnet")ž
ModelMetodaUmjetneNeuronskeMreeZatajenjaSrcaTestiranje <- predict(
    žModelMetodaUmjetneNeuronskeMreeZatajenjaSrca, skupTestiranjaZatajenjeSrca)ž
ModelMetodaUmjetneNeuronskeMreeZatajenjaSrcaTestiranjeConf <- confusionMatrix(
    žModelMetodaUmjetneNeuronskeMreeZatajenjaSrcaTestiranje,
    skupTestiranjaZatajenjeSrca$HeartDisease, mode = "everything")
print("Matrica␣zbunjivanja␣za␣Metodu␣umjetne␣neuronske␣žmree␣u␣skupu␣podataka␣
    Zatajenje␣Srca")
print(žModelMetodaUmjetneNeuronskeMreeZatajenjaSrcaTestiranjeConf)
print(žModelMetodaUmjetneNeuronskeMreeZatajenjaSrca)
```

Rezultat:

```
[1] "Matrica zbunjivanja za Metodu umjetne neuronske mreže u skupu podataka Zatajenje Srca"
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 105  24
         1  18 128

               Accuracy : 0.8473
                 95% CI : (0.7992, 0.8877)
    No Information Rate : 0.5527
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6925

 Mcnemar's Test P-Value : 0.4404

            Sensitivity : 0.8537
            Specificity : 0.8421
         Pos Pred Value : 0.8140
         Neg Pred Value : 0.8767
              Precision : 0.8140
                 Recall : 0.8537
                     F1 : 0.8333
             Prevalence : 0.4473
         Detection Rate : 0.3818
   Detection Prevalence : 0.4691
      Balanced Accuracy : 0.8479

       'Positive' Class : 0
```

Slika 35: Klasifikacija umjetno neuronskom mrežom u skupom podataka o predviđanju zatajenja srca(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom umjetne neuronske mreže daju se sljedeći rezultati parametara koji će se koristi u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8473, preciznost = 0.8140, odaziv = 0.8537 i f-1 rezultat = 0.8333. Prilikom treniranja metoda je automatski odlučila da će koristiti 5 umjetnih neurona u skrivenom sloju.

5.1.5. Naivni Bayesov algoritam

Prilikom izrade klasifikacije naivnog Bayesovog algoritma prvo se trenira klasifikacijski model pod nazivom ModelMetodaNaivniBayesZatajenjaSrca pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da HeartDisease bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaZatajenjeSrca govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaZatajenjeSrca. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva naive_bayes. Zapravo se radi o naivnom Bayesu. Nakon treninga se izvršava klasifikacija nad skupTestiranjaZatajenjeSrca pomoću klasifikacijskog modela ModelMetodaNaivniBayesZatajenjaSrca te se dobiva ModelMetodaNaivniBayesZatajenjaSrcaTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelMetodaNaivniBayesZatajenjaSrcaTestiranje pomoću skupTestiranjaZatajenjeSrca$HeartDisease te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda naivni Bayes
ModelMetodaNaivniBayesZatajenjaSrca <- train(HeartDisease ~.,data=
    skupTreniranjaZatajenjeSrca,
                                method="naive_bayes")
ModelMetodaNaivniBayesZatajenjaSrcaTestiranje <- predict(
    ModelMetodaNaivniBayesZatajenjaSrca, skupTestiranjaZatajenjeSrca)
```

```
ModelMetodaNaivniBayesZatajenjaSrcaTestiranjeCof <- confusionMatrix(
    ModelMetodaNaivniBayesZatajenjaSrcaTestiranje, skupTestiranjaZatajenjeSrca$
    HeartDisease, mode = "everything")
print("Matrica zbunjivanja za Metodu naivnog Bayes-a u skupu podataka Zatajenje Srca
    ")
print(ModelMetodaNaivniBayesZatajenjaSrcaTestiranjeCof)
```

Rezultat:

```
[1] "Matrica zbunjivanja za Metodu naivnog Bayes-a u skupu podataka Zatajenje Srca"
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 100  16
         1  23 136

               Accuracy : 0.8582
                 95% CI : (0.8113, 0.8972)
    No Information Rate : 0.5527
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7116

 Mcnemar's Test P-Value : 0.3367

            Sensitivity : 0.8130
            Specificity : 0.8947
         Pos Pred Value : 0.8621
         Neg Pred Value : 0.8553
              Precision : 0.8621
                 Recall : 0.8130
                     F1 : 0.8368
             Prevalence : 0.4473
         Detection Rate : 0.3636
   Detection Prevalence : 0.4218
      Balanced Accuracy : 0.8539

       'Positive' Class : 0
```

Slika 36: Klasifikacija naivnim Bayesom u skupom podataka o predviđanju zatajenja srca(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom naivni bayesov algoritam daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8582, preciznost = 0.8621, odaziv = 0.8130 i f-1 rezultat = 0.8368.

5.1.6. Slučajna šuma

Prilikom izrade klasifikacije metodom slučajna šuma prvo se trenira klasifikacijski model pod nazivom ModelMetodaRandomForestsZatajenjaSrca pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da HeartDisease bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaZatajenjeSrca govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaZatajenjeSrca. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva rf. Zapravo se radi o algoritmu slučajna šuma. Nakon treninga se izvršava klasifikacija nad skupTestiranjaZatajenjeSrca pomoću klasifikacijskog modela ModelMetodaRandomForestsZatajenjaSrca te se dobiva ModelMetodaRandomForestsZatajenjaSrcaTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelMetodaRandomForestsZatajenjaSrcaTestiranje pomoću skupTestiranjaZatajenjeSrca$HeartDisease te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda Random Forest
ModelMetodaRandomForestsZatajenjaSrca <- train(HeartDisease ~.,data=
    skupTreniranjaZatajenjeSrca,
                                method="rf")
ModelMetodaRandomForestsZatajenjaSrcaTestiranje <- predict(
    ModelMetodaRandomForestsZatajenjaSrca, skupTestiranjaZatajenjeSrca)
ModelMetodaRandomForestsZatajenjaSrcaTestiranjeCof <- confusionMatrix(
    ModelMetodaRandomForestsZatajenjaSrcaTestiranje, skupTestiranjaZatajenjeSrca$
    HeartDisease, mode = "everything")
print("Matrica␣zbunjivanja␣za␣Metodu␣random␣forest␣u␣skupu␣podataka␣Zatajenje␣Srca")
print(ModelMetodaRandomForestsZatajenjaSrcaTestiranjeCof)
```

Rezultat:

```
[1] "Matrica zbunjivanja za Metodu random forest u skupu podataka Zatajenje Srca"
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 103  15
         1  20 137

               Accuracy : 0.8727
                 95% CI : (0.8275, 0.9097)
    No Information Rate : 0.5527
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7416

 Mcnemar's Test P-Value : 0.499

            Sensitivity : 0.8374
            Specificity : 0.9013
         Pos Pred Value : 0.8729
         Neg Pred Value : 0.8726
              Precision : 0.8729
                 Recall : 0.8374
                     F1 : 0.8548
             Prevalence : 0.4473
         Detection Rate : 0.3745
   Detection Prevalence : 0.4291
      Balanced Accuracy : 0.8694

       'Positive' Class : 0
```

Slika 37: Klasifikacija metodom algoritma slučajna šuma nad skupom podataka o predviđanju zatajenja srca(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom slučajna šuma daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8727, preciznost = 0.8729, odaziv = 0.8374 i f-1 rezultat = 0.8548.

5.1.7. K- najbližih susjeda

Prilikom izrade klasifikacije metodom k-najbližih susjeda prvo se trenira klasifikacijski model pod nazivom ModelMetodaKNNZatajenjaSrca pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da HeartDisease bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaZatajenjeSrca govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaZatajenjeSrca. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva knn. Zapravo se radi o metodi k-najbližih susjeda. Nakon treninga se izvršava klasifikacija nad skupTestiranjaZatajenjeSrca pomoću klasifikacijskog modela ModelMetodaKNNZatajenjaSrca te se dobiva ModelMetodaKNNZatajenjaSrcaTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se

matrica zbunjivanja nad ModelMetodaKNNZatajenjaSrcaTestiranje pomoću skupTestiranjaZatajenjeSrca$HeartDisease te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda KNN
ModelMetodaKNNZatajenjaSrca <- train(HeartDisease ~.,data=
    skupTreniranjaZatajenjeSrca,
                                method="knn")
ModelMetodaKNNZatajenjaSrcaTestiranje <- predict(ModelMetodaKNNZatajenjaSrca,
    skupTestiranjaZatajenjeSrca)
ModelMetodaKNNZatajenjaSrcaTestiranjeCof <- confusionMatrix(
    ModelMetodaKNNZatajenjaSrcaTestiranje, skupTestiranjaZatajenjeSrca$HeartDisease,
     mode = "everything")
print("Matrica zbunjivanja za Metodu KNN-a u skupu podataka Zatajenje Srca")
print(ModelMetodaKNNZatajenjaSrcaTestiranjeCof)
print(ModelMetodaKNNZatajenjaSrca)
```

Rezultat:

```
[1] "Matrica zbunjivanja za Metodu KNN-a u skupu podataka Zatajenje Srca"
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  86  44
         1  37 108

               Accuracy : 0.7055
                 95% CI : (0.6477, 0.7587)
    No Information Rate : 0.5527
    P-Value [Acc > NIR] : 1.466e-07

                  Kappa : 0.4075

 Mcnemar's Test P-Value : 0.505

            Sensitivity : 0.6992
            Specificity : 0.7105
         Pos Pred Value : 0.6615
         Neg Pred Value : 0.7448
              Precision : 0.6615
                 Recall : 0.6992
                     F1 : 0.6798
             Prevalence : 0.4473
         Detection Rate : 0.3127
   Detection Prevalence : 0.4727
      Balanced Accuracy : 0.7049

       'Positive' Class : 0
```

Slika 38: Klasifikacija metodom k-najbližih susjeda nad skupom podataka o predviđanju zatajenja srca(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom k-najbližih susjeda daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.7055, preciznost = 0.6615, odaziv = 0.6992 i f-1 rezultat = 0.6798. Prilikom treniranja automatski je odlučeno da će se koristiti k=7 prilikom klasifikacije podataka.

5.1.8. Komparacija parametara dobivenih analizom klasifikacijskih metoda nad skupom podataka o predviđanju zatajenja srca

Tablica 1: Komparacija parametara dobivenih analizom klasifikacijskih metoda nad skupom podataka o predviđanju zatajenja srca

Parametar/ Klasifikacijska metoda	Logistička regresija	Metoda potpornih vektora	Umjetne neuronske mreže	Naivni Bayes	Slučajna šuma	K-najbližih susjeda
Točnost	0.8618	0.8655	0.8473	0.8582	0.8727	0.7055
Preciznost	0.8346	0.8468	0.8140	0.8621	0.8729	0.6615
Odaziv	0.8618	0.8537	0.8537	0.8130	0.8374	0.6992
F-1 rezultat	0.8480	0.8502	0.8333	0.8368	0.8548	0.6798

(samostalna izrada)

Temeljem podataka iz gornje tablice može se zaključiti kako je metoda k najbližih susjeda najgore izvela klasifikaciju nad skupom podataka o predviđanju zatajenja srca. S obzirom na parametre točnosti, preciznosti i F-1 rezultata najtočiniju klasifikaciju nad skupom podataka dala je metoda slučajna šuma. Promatrajući parametar odaziva najbolje rezultat daje logistička regresija. Može se primjetiti da metode: logistička regresija, potporni vektori, naivni bayesov algoritam, umjetne neuronske mreže te slučajna šuma daju podjednake rezultate te temeljem komparativne analize može se reći da su sve navedene metode prikladne za klasifkaciju podataka nad skupom podataka o predviđanju zatajenja srca osim metode k najbližih susjeda.

5.2. Klasifikacija nad skupom podataka o predviđanju i analizi srčanog udara

5.2.1. Kreiranje particija

Kao što je već spomenuto potrebana je kreiranje particija skupova pomoću naredbe createDataPartition(). U njoj naređujemo da glavni atribut po kojem će se klasificirati skup podataka bude output te 70% skupa podataka dodjeljeno skupu pod nazivom skupTreniranjaSrcaniUdar, a 30% skupu pod nazivom skupTestiranjaSrcaniUdar. Treba se postaviti sjeme da kad ponovno izvrtimo simulacije skupovi budu jednaki zbog toga što o njima ovise i rezultati same klasifikacije podataka. List=False je postavljen kao zadnji parametar u createDataPartition-u zato što se ne želi dobivati liste.

```
skupSrcaniUdar <- createDataPartition(srcaniUdar$output, p=0.7, list=FALSE)
skupTreniranjaSrcaniUdar <- srcaniUdar[skupSrcaniUdar,]
skupTestiranjaSrcaniUdar <- srcaniUdar[-skupSrcaniUdar,]
```

5.2.2. Logistička regresija

Prilikom izrade klasifikacije logističke regresije prvo se trenira klasifikacijski model pod nazivom ModelLogistickaRegresijaSrcaniUdar pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da output bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaSrcaniUdar govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaSrcaniUdar. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva glm te pripada binomialnoj obitelji. Zapravo se radi o binarnoj logističkoj regresiji. Nakon treninga se izvršava klasifikacija nad skupTestiranjaSrcaniUdar pomoću klasifikacijskog modela ModelLogistickaRegresijaSrcaniUdar te se dobiva ModelLogistickaRegresijaSrcaniUdarTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelLogistickaRegresijaSrcaniUdarTestiranje pomoću skupTestiranjaSrcaniUdar$output te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda Logisticka regresija
ModelLogistickaRegresijaSrcaniUdar <- train(output ~.,data=skupTreniranjaSrcaniUdar,
                                method="glm",family="binomial")

ModelLogistickaRegresijaSrcaniUdarTestiranje <- predict(
    ModelLogistickaRegresijaSrcaniUdar, skupTestiranjaSrcaniUdar)
ModelLogistickaRegresijaSrcaniUdarTestiranjeCof <- confusionMatrix(
    ModelLogistickaRegresijaSrcaniUdarTestiranje, skupTestiranjaSrcaniUdar$output,
    mode = "everything")
print("Matrica zbunjivanja za Logisticku regresiju u skupu podataka Srcani Udar")
print(ModelLogistickaRegresijaSrcaniUdarTestiranjeCof)
```

Rezultat:

```
[1] "Matrica zbunjivanja za Logisticku regresiju u skupu podataka Srcani Udar"
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 34 10
         1  7 39

               Accuracy : 0.8111
                 95% CI : (0.7149, 0.8859)
    No Information Rate : 0.5444
    P-Value [Acc > NIR] : 1.061e-07

                  Kappa : 0.6215

 Mcnemar's Test P-Value : 0.6276

            Sensitivity : 0.8293
            Specificity : 0.7959
         Pos Pred Value : 0.7727
         Neg Pred Value : 0.8478
              Precision : 0.7727
                 Recall : 0.8293
                     F1 : 0.8000
             Prevalence : 0.4556
         Detection Rate : 0.3778
   Detection Prevalence : 0.4889
      Balanced Accuracy : 0.8126

       'Positive' Class : 0
```

Slika 39: Klasifikacija logističkom regresijom nad skupom podataka o predviđanju i analizi srčanog udara(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom logističke regresije daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8111, preciznost = 0.7727, odaziv = 0.8293 i f-1 rezultat = 0.8.

5.2.3. Metoda potpornih vektora

Prilikom izrade klasifikacije metodom potpornih vektora prvo se trenira klasifikacijski model pod nazivom ModelMetodaPotpornihVektoraSrcaniUdar pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da output bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaSrcaniUdar govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaSrcaniUdar. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva svmLinear2. Zapravo se radi o metodi potpornih vektora s linearnom jezgrom. Nakon treninga se izvršava klasifikacija nad skupTestiranjaSrcaniUdar pomoću klasifikacijskog modela ModelMetodaPotpornihVektoraSrcaniUdar te se dobiva ModelMetodaPotpornihVektoraSrcaniUdarTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelMetodaPotpornihVektoraSrcaniUdarTestiranje pomoću skupTestiranjaSrcaniUdar$output te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda potpornih vektora
```

```
ModelMetodaPotpornihVektoraSrcaniUdar <- train(output ~.,data=
    skupTreniranjaSrcaniUdar,
                               method="svmLinear2")

ModelMetodaPotpornihVektoraSrcaniUdarTestiranje <- predict(
    ModelMetodaPotpornihVektoraSrcaniUdar, skupTestiranjaSrcaniUdar)
ModelMetodaPotpornihVektoraSrcaniUdarTestiranjeCof <- confusionMatrix(
    ModelMetodaPotpornihVektoraSrcaniUdarTestiranje, skupTestiranjaSrcaniUdar$output
    , mode = "everything")
print("Matrica zbunjivanja za metodu potpornih vektora u skupu podataka Srcani Udar"
    )
print(ModelMetodaPotpornihVektoraSrcaniUdarTestiranjeCof)
```

Rezultat:

```
[1] "Matrica zbunjivanja za metodu potpornih vektora u skupu podataka Srcani Udar"
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 32  8
         1  9 41

               Accuracy : 0.8111
                 95% CI : (0.7149, 0.8859)
    No Information Rate : 0.5444
    P-Value [Acc > NIR] : 1.061e-07

                  Kappa : 0.6185

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.7805
            Specificity : 0.8367
         Pos Pred Value : 0.8000
         Neg Pred Value : 0.8200
              Precision : 0.8000
                 Recall : 0.7805
                     F1 : 0.7901
             Prevalence : 0.4556
         Detection Rate : 0.3556
   Detection Prevalence : 0.4444
      Balanced Accuracy : 0.8086

       'Positive' Class : 0
```

Slika 40: Klasifikacija metodom potpornih vektora nad skupom podataka o predviđanju i analizi srčanog udara(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom potpornih vektora daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8111, preciznost = 0.8, odaziv = 0.7805 i f-1 rezultat = 0.7901.

5.2.4. Umjetne neuronske mreže

Prilikom izrade klasifikacije umjetne neuronske mreže prvo se trenira klasifikacijski model pod nazivom ModelUmjetneNuronskeMrezeSrcaniUdar pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da output bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaSrcaniUdar govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaSrcaniUdar. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva nnet. Zapravo se radi o umjetnim neuronskim mrežama. Nakon treninga se izvršava klasifikacija nad skupTestiranjaSrcaniUdar pomoću klasifikacijskog modela ModelUmjetneNuronskeMrezeSrcaniUdar te se dobiva ModelUmjetneNuronskeMrezeSrcaniUdarTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelUmjetneNuronskeMrezeSrcaniUdarTestiranje pomoću skupTestiranjaSrcaniUdar$output te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda ANN
ModelUmjetneNuronskeMrezeSrcaniUdar <- train(output ~.,data=skupTreniranjaSrcaniUdar
    ,
                                method="nnet")

ModelUmjetneNuronskeMrezeSrcaniUdarTestiranje <- predict(
    ModelUmjetneNuronskeMrezeSrcaniUdar, skupTestiranjaSrcaniUdar)
ModelUmjetneNuronskeMrezeSrcaniUdarTestiranjeCof <- confusionMatrix(
    ModelUmjetneNuronskeMrezeSrcaniUdarTestiranje, skupTestiranjaSrcaniUdar$output,
    mode = "everything")
print("Matrica␣zbunjivanja␣za␣umjetne␣neuronske␣žmree␣u␣skupu␣podataka␣Srcani␣Udar")
print(ModelUmjetneNuronskeMrezeSrcaniUdarTestiranjeCof)
print(ModelUmjetneNuronskeMrezeSrcaniUdar)
```

Rezultat:

```
[1] "Matrica zbunjivanja za umjetne neuronske mreže u skupu podataka Srcani Udar"
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 33 10
         1  8 39

               Accuracy : 0.8
                 95% CI : (0.7025, 0.8769)
    No Information Rate : 0.5444
    P-Value [Acc > NIR] : 3.697e-07

                  Kappa : 0.5984

 Mcnemar's Test P-Value : 0.8137

            Sensitivity : 0.8049
            Specificity : 0.7959
         Pos Pred Value : 0.7674
         Neg Pred Value : 0.8298
              Precision : 0.7674
                 Recall : 0.8049
                     F1 : 0.7857
             Prevalence : 0.4556
         Detection Rate : 0.3667
   Detection Prevalence : 0.4778
      Balanced Accuracy : 0.8004

       'Positive' Class : 0
```

Slika 41: Klasifikacija umjetnih neuronskih mreža nad skupom podataka o predviđanju i analizi srčanog udara(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom umjetnih neuronskih mreža daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8, preciznost = 0.7674, odaziv = 0.8049 i f-1 rezultat = 0.7857. Prilikom treniranje umjetne neuronske mreže automatski je odlučeno da će se koristiti 5 umjetnih neurona u skrivenom sloju.

5.2.5. Naivni Bayesov algoritam

Prilikom izrade klasifikacije naivnim Bayesom prvo se trenira klasifikacijski model pod nazivom ModelNaivniBayesSrcaniUdar pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da output bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaSrcaniUdar govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaSrcaniUdar. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva naive_bayes. Zapravo se radi o naivnom Bayesu. Nakon treninga se izvršava klasifikacija nad skupTestiranjaSrcaniUdar pomoću klasifikacijskog modela ModelNaivniBayesSrcaniUdar te se dobiva ModelNaivniBayesSrcaniUdarTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelNaivniBayesSrcaniUdarTestiranje pomoću skupTestiranjaSrcaniUdar$output te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda naivni Bayes
ModelNaivniBayesSrcaniUdar <- train(output ~.,data=skupTreniranjaSrcaniUdar,
                                    method="naive_bayes")
```

```
ModelNaivniBayesSrcaniUdarTestiranje <- predict(ModelNaivniBayesSrcaniUdar,
    skupTestiranjaSrcaniUdar)
ModelNaivniBayesSrcaniUdarTestiranjeCof <- confusionMatrix(
    ModelNaivniBayesSrcaniUdarTestiranje, skupTestiranjaSrcaniUdar$output, mode = "
    everything")
print("Matrica␣zbunjivanja␣za␣naivni␣Bayes␣u␣skupu␣podataka␣Srcani␣Udar")
print(ModelNaivniBayesSrcaniUdarTestiranjeCof)
```

Rezultat:

```
[1] "Matrica zbunjivanja za naivni Bayes u skupu podataka Srcani Udar"
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 31  6
         1 10 43

               Accuracy : 0.8222
                 95% CI : (0.7274, 0.8948)
    No Information Rate : 0.5444
    P-Value [Acc > NIR] : 2.84e-08

                  Kappa : 0.6387

 Mcnemar's Test P-Value : 0.4533

            Sensitivity : 0.7561
            Specificity : 0.8776
         Pos Pred Value : 0.8378
         Neg Pred Value : 0.8113
              Precision : 0.8378
                 Recall : 0.7561
                     F1 : 0.7949
             Prevalence : 0.4556
         Detection Rate : 0.3444
   Detection Prevalence : 0.4111
      Balanced Accuracy : 0.8168

       'Positive' Class : 0
```

Slika 42: Klasifikacija naivnim Bayesom nad skupom podataka o predviđanju i analizi srčanog udara(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom naivnog Bayesovog algoritma mreža daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8222, preciznost = 0.8378, odaziv = 0.7561 i f-1 rezultat = 0.7949.

5.2.6. Slučajna šuma

Prilikom izrade klasifikacije metodom slučajana šuma prvo se trenira klasifikacijski model pod nazivom ModelRandomForestSrcaniUdar pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da output bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaSrcaniUdar govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaSrcaniUdar. U zadnjim parametrima je dodjeljena metoda pomoću koje se gradi klasifikacijiski model a ona se naziva rf. Zapravo se radi o slučajnoj šuma. Nakon treninga se izvršava klasifikacija nad skupTestiranjaSrcaniUdar pomoću klasifikacijskog modela ModelRandomForestSrcaniUdar te se dobiva ModelRandomForestSrcaniUdarTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelRandomForestSrcaniUdarTestiranje pomoću skupTestiranjaSrcaniUdar$output te je namješten

drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda random forest
ModelRandomForestSrcaniUdar <- train(output ~.,data=skupTreniranjaSrcaniUdar,
                              method="rf")

ModelRandomForestSrcaniUdarTestiranje <- predict(ModelRandomForestSrcaniUdar,
    skupTestiranjaSrcaniUdar)
ModelRandomForestSrcaniUdarTestiranjeCof <- confusionMatrix(
    ModelRandomForestSrcaniUdarTestiranje, skupTestiranjaSrcaniUdar$output, mode = "
    everything")
print("Matrica␣zbunjivanja␣za␣Random␣Forest␣u␣skupu␣podataka␣Srcani␣Udar")
print(ModelRandomForestSrcaniUdarTestiranjeCof)
```

Rezultat:

```
[1] "Matrica zbunjivanja za Random Forest u skupu podataka Srcani Udar"
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 35 11
         1  6 38

               Accuracy : 0.8111
                 95% CI : (0.7149, 0.8859)
    No Information Rate : 0.5444
    P-Value [Acc > NIR] : 1.061e-07

                  Kappa : 0.623

 Mcnemar's Test P-Value : 0.332

            Sensitivity : 0.8537
            Specificity : 0.7755
         Pos Pred Value : 0.7609
         Neg Pred Value : 0.8636
              Precision : 0.7609
                 Recall : 0.8537
                     F1 : 0.8046
             Prevalence : 0.4556
         Detection Rate : 0.3889
   Detection Prevalence : 0.5111
      Balanced Accuracy : 0.8146

       'Positive' Class : 0
```

Slika 43: Klasifikacija metodom slučajna šuma nad skupom podataka o predviđanju i analizi srčanog udara(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom slučajne šume mreža daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.8111, preciznost = 0.7609, odaziv = 0.8537 i f-1 rezultat = 0.8046.

5.2.7. K najbližih susjeda

Prilikom izrade klasifikacije motodom k-najbližih susjeda prvo se trenira klasifikacijski model pod nazivom ModelKNNSrcaniUdar pomoću funkcije train(). U prvom paremetru funkcije train naređujemo da output bude glavni parametar po kojem će se klasificirati. Drugi parametar data=skupTreniranjaSrcaniUdar govori o tome da će se trening obaviti uz pomoću skupa pod nazivom skupTreniranjaSrcaniUdar. U zadnjim parametrima je dodjeljena metoda pomoću koje

se gradi klasifikacijiski model a ona se naziva knn. Zapravo se radi o metodi k-najbližih susjeda. Nakon treninga se izvršava klasifikacija nad skupTestiranjaSrcaniUdar pomoću klasifikacijskog modela ModelKNNSrcaniUdar te se dobiva ModelKNNSrcaniUdarTestiranje pomoću naredbe predict(). Nakon klasifikacije stvara se matrica zbunjivanja nad ModelKNNSrcaniUdarTestiranje pomoću skupTestiranjaSrcaniUdar$output te je namješten drugi parametar mode="everything" koji omogućuje ispis f1 rezultata. Nakon stvaranja matrica zbunjivanja ispisuje se rezultat.

```
#Metoda KNN
ModelKNNSrcaniUdar <- train(output ~.,data=skupTreniranjaSrcaniUdar,
                            method="knn", tuneGrid=data.frame(k=(6)))

ModelKNNSrcaniUdarTestiranje <- predict(ModelKNNSrcaniUdar, skupTestiranjaSrcaniUdar
    )
ModelKNNSrcaniUdarTestiranjeCof <- confusionMatrix(ModelKNNSrcaniUdarTestiranje,
    skupTestiranjaSrcaniUdar$output, mode = "everything")
print("Matrica␣zbunjivanja␣za␣KNN␣u␣skupu␣podataka␣Srcani␣Udar")
print(ModelKNNSrcaniUdarTestiranjeCof)
```

Rezultat:

```
[1] "Matrica zbunjivanja za KNN u skupu podataka Srcani Udar"
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 29 15
         1 12 34

               Accuracy : 0.7
                 95% CI : (0.5943, 0.7921)
    No Information Rate : 0.5444
    P-Value [Acc > NIR] : 0.001859

                  Kappa : 0.3988

 Mcnemar's Test P-Value : 0.700311

            Sensitivity : 0.7073
            Specificity : 0.6939
         Pos Pred Value : 0.6591
         Neg Pred Value : 0.7391
              Precision : 0.6591
                 Recall : 0.7073
                     F1 : 0.6824
             Prevalence : 0.4556
         Detection Rate : 0.3222
   Detection Prevalence : 0.4889
      Balanced Accuracy : 0.7006

       'Positive' Class : 0
```

Slika 44: Klasifikacija metodom k-najbližih susjeda nad skupom podataka o predviđanju i analizi srčanog udara(samostalna izrada)

Prema dobivenim rezultatima klasifikacija metodom k-najbližih susjeda daje sljedeće rezultate parametara koji će se koristiti u komparativnoj analizi klasifikacijskih metoda. Točnost = 0.7, preciznost = 0.6591, odaziv = 0.7073 i f-1 rezultat = 0.6824. K ima vrijednost 6 te je samostalno izabran preko parametra u train funkciji tuneGrid=data.frame(k=(6)). U slučaju automatskog biranja k(vrijednost 9) parametri komparativne analize postaju gori.

5.2.8. Komparacija parametara dobivenih analizom klasifikacijskih metoda nad skupom podataka o predviđanju i analizi srčanog udara

Tablica 2: Komparacija parametara dobivenih analizom klasifikacijskih metoda nad skupom podataka o predviđanju i analizi srčanog udara

Parametar/ Klasifikacijska metoda	Logistička regresija	Metoda potpornih vektora	Umjetne neuronske mreže	Naivni Bayes	Slučajna šuma	K-najbližih susjeda
Točnost	0.8111	0.8111	0.8	0.8222	0.8111	0.7
Preciznost	0.7727	0.8	0.7674	0.8378	0.7609	0.6591
Odaziv	0.8293	0.7805	0.8049	0.7561	0.8537	0.7073
F-1 rezultat	0.8	0.7901	0.7857	0.7949	0.8046	0.6824

(samostalna izrada)

Temeljem parametara iz gornje tablice može se zaključiti kako naivni Bayesov algoritam daje najbolju točnost i preciznost klasifikacije dok metoda slučajne šume daje najbolje rezultate za odaziv i F-1 rezultat prilikom klasifikacije nad skupom podataka o predviđanju i analizi srčanog udara. Metoda k-najbližih susjeda ima najgore rezultate u svim parametrima komparacijske analize prilikom klasifikacije podataka nad skupom podataka o predviđanju i analizi srčanog udara. Promatrajući tablicu može se opet zaključiti da da metode: logistička regresija, potporni vektori, naivni bayesov algoritam, umjetne neuronske mreže te slučajna šuma daju podjednake rezultate te temeljem komparativne analize može se re ci da su sve navedene metode prikladne za klasifkaciju podataka nad skupom podataka o predviđanju i analizi srčanog udara osim metode k najbližih susjeda.

6. Zaključak

U ovome radu je izvršena komparativna analiza različitih klasifikacijskih metoda nad skupom podataka o predviđanju zatajenja srca te nad skupom podataka o predviđanju i analizi srčanog udara.

U komparativnoj analizi uspoređene su sljedeće klasifikacijske metode: logistička regresija, metoda potpornih vektora, umjetne neuronske mreže, naivni Bayes, slučajna šuma i metoda k-najbližih susjeda.

Prema rezultatima komparativne analize nad spomenutim skupovima najbolja metoda je metoda slučajne šume. Metoda slučajnih šuma vrlo je stablina te daje vrlo precizna predviđanja klasifikacije.

Ispada da metoda k-najbližih susjeda daje najgore rezultate prema rezultatima dobivenim prilikom komparativne analize. Metoda k-najbližih susjeda je osjetljiva na nebitne podatke.

Za bolji rad klasifikacijskih metoda potrebna su prilagođavanja određenih metoda prema skupu podataka. Za određene skupove podataka jednostavno neke metode bolje rade zbog svog prirodnog načina rada algoritma. Komparativnom analizom klasifikacijskih metoda može se riješiti nedoumice pri izboru klasifikacijske metode. Kao što je već i spomenuto ispravna klasifikacija može riješiti probleme kao primjerice u medicini kod srčanih bolesti. Ovo je i sam pokazatelj da je klasifikacija podataka vrlo korisna u stvarnom svijetu.

Popis literature

[1] overleaf.com. „Overleaf Official Logos.” (), adresa: https://www.overleaf.com/for/partners/logos (pogledano 25. 8. 2022.).

[2] kaggle.com. „How to Use Kaggle.” (), adresa: https://www.kaggle.com/docs/notebooks (pogledano 13. 2. 2024.).

[3] kaggle.com. „Guidelines for use of Kaggle brand features.” (), adresa: https://www.kaggle.com/brand-guidelines (pogledano 13. 2. 2024.).

[4] sambhav228. „Introduction of Object Oriented Programming.” (11. rujna 2020.), adresa: https://www.geeksforgeeks.org/introduction-of-object-oriented-programming/?ref=gcse (pogledano 25. 8. 2022.).

[5] H. enciklopedija. „klasa.” (11. rujna 2020.), adresa: http://www.enciklopedija.hr/Natuknica.aspx?ID=31754 (pogledano 23. 1. 2023.).

[6] D. Hand, H. Mannila i P. Smyth, „Principles of Data Mining,” The MIT Press, 2001.

[7] T. Terkhedkar. „DATA MINING TECHNIQUES.” (23. travnja 2019.), adresa: https://medium.com/@tanmayct/data-mining-techniques-24d01a8fb71e (pogledano 25. 8. 2022.).

[8] S. Modi. „Classification in Data Mining.” (2. studenoga 2020.), adresa: https://shyamodi.medium.com/classification-in-data-mining-418a08eb325c (pogledano 25. 8. 2022.).

[9] winder.ai. „403: Linear Classification.” (), adresa: https://winder.ai/403-linear-classification/ (pogledano 23. 8. 2023.).

[10] X. Zhai. „How to Choose Different Types of Linear Classifiers?” (8. rujna 2022.), adresa: https://medium.com/mlearning-ai/how-%20to-choose-different-types-of-linear-classifiers-63ca88f5cd3a (pogledano 23. 8. 2023.).

[11] D. Krishna. „A Look at the Maths Behind Linear Classification.” (), adresa: https://towardsdatascience.com/a-look-at-the-maths-behind-linear-classification-166e99a9e5fb (pogledano 23. 8. 2023.).

[12] S. S. Gill. „Linear vs. Non-Linear Classification.” (), adresa: https://www.codingninjas.com/studio/library/linear-vs-non-linear-classification (pogledano 23. 8. 2023.).

[13] A. Karatzoglou, D. Meyer i K. Hornik, „Support Vector Machines in R," *Journal of Statistical Software*, sv. 15, br. 9, str. 1–28, 2006. DOI: 10.18637/jss.v015.i09. adresa: https://www.jstatsoft.org/index.php/jss/article/view/v015i09.

[14] A. Saini. „Guide on Support Vector Machine (SVM) Algorithm." (12. kolovoza 2021.), adresa: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/ (pogledano 23. 8. 2023.).

[15] S. Patel. „Chapter 2 : SVM (Support Vector Machine) — Theory." (3. ožujka 2017.), adresa: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72 (pogledano 23. 8. 2023.).

[16] A. Yadav. „SUPPORT VECTOR MACHINES(SVM)." (2018.), adresa: https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589#:~:text=Pros%20and%20cons%20of%20SVM%3A&text=It%20is%20really%20effective%20in,thus%20outliers%20have%20less%20impact (pogledano 17. 1. 2024.).

[17] sciencedirect.com. „Logistic Regression." (2014.), adresa: https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on (pogledano 17. 1. 2024.).

[18] S. SHARMA. „Activation Functions in Neural Networks." (2017.), adresa: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (pogledano 17. 1. 2024.).

[19] V. Kanade. „What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices." (2022.), adresa: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/ (pogledano 17. 1. 2024.).

[20] ibm.com. „What is logistic regression?" (), adresa: https://www.ibm.com/topics/logistic-regression#:~:text=Resources-,What%20is%20logistic%20regression%3F,given%20dataset%20of%20independent%20variables (pogledano 17. 1. 2024.).

[21] V. Alto. „Understanding the Perceptron Algorithm." (2021.), adresa: https://medium.com/analytics-vidhya/understanding-the-perceptron-algorithm-4a368f493109 (pogledano 23. 8. 2023.).

[22] Shreyak. „A to Z about Artificial Neural Networks (ANN) (Theory N Hands-on)." (2020.), adresa: https://medium.com/analytics-vidhya/a-to-z-about-artificial-neural-networks-ann-theory-n-hands-on-713c12f3351e (pogledano 23. 8. 2023.).

[23] AmyChu. „What is a linear classifier (Perceptron)." (2023.), adresa: https://medium.com/@musicaround/10-what-is-a-linear-classifier-perceptron-a2356a5d86e (pogledano 23. 8. 2023.).

[24] D. Kriesel, *A Brief Introduction to Neural Networks*. David Kriesel, 2007.

[25] M. Vulin. „Umjetne neuronske mreže kao metoda umjetne inteligencije, , Završni rad, Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin." (2020.), adresa: https://urn.nsk.hr/urn:nbn:hr:211:803651 (pogledano 23. 8. 2023.).

[26] M. Islam, G. Chen i S. Jin, „An overview of neural network," *American Journal of Neural Networks and Applications*, sv. 5, br. 1, str. 7–11, 2019.

[27] D. V. Pandey. „Artificial Neural Networks and their applications in Computer Vision, NLP and Robotics." (2023.), adresa: https://www.linkedin.com/pulse/artificial-neural-networks-applications-computer-vision-pandey (pogledano 17. 1. 2024.).

[28] M. Zakaria, A. Mabrouka i S. Sarhan, „Artificial neural network: a brief overview," *neural networks*, sv. 1, str. 2, 2014.

[29] geeksforgeeks.org. „Advantages and Disadvantages of ANN in Data Mining." (2021.), adresa: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-ann-in-data-mining/ (pogledano 17. 1. 2024.).

[30] V. Fedak. „Top 10 Most Popular AI Models." (2018.), adresa: https://dzone.com/articles/top-10-most-popular-ai-models (pogledano 28. 8. 2023.).

[31] MasterClass.com. „Bayes' Theorem: How to Use the Probability Formula." (2022.), adresa: https://www.masterclass.com/articles/bayes-theorem (pogledano 28. 8. 2023.).

[32] ibm.com. „What are Naïve Bayes classifiers?" (), adresa: https://www.ibm.com/topics/naive-bayes#:~:text=The%20Na%C3%AFve%20Bayes%20classifier%20is,a%20given%20class%20or%20category (pogledano 17. 1. 2024.).

[33] turing.com. „An Introduction to Naive Bayes Algorithm for Beginners." (), adresa: https://www.turing.com/kb/an-introduction-to-naive-bayes-algorithm-for-beginners#types-of-the-naive-bayes-model (pogledano 17. 1. 2024.).

[34] M. Borcan. „Decision Tree Classifiers Explained." (2020.), adresa: https://medium.com/@borcandumitrumarius/decision-tree-classifiers-explained-e47a5b68477a (pogledano 28. 8. 2023.).

[35] A. Saini. „Decision Tree – A Step-by-Step Guide." (2024.), adresa: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/ (pogledano 17. 1. 2024.).

[36] ibm.com. „What is a Decision Tree?" (), adresa: https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes (pogledano 17. 1. 2024.).

[37] Breiman. „L. Random Forests. Machine Learning 45, 5–32." (2001.), adresa: https://doi.org/10.1023/A:1010933404324 (pogledano 17. 1. 2024.).

[38] ibm.com. „What is random forest?" (2001.), adresa: https://www.ibm.com/topics/random-forest (pogledano 17. 1. 2024.).

[39] S. E. R. „Understand Random Forest Algorithms With Examples (Updated 2024)." (2024.), adresa: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/ (pogledano 17. 1. 2024.).

[40] P. Choubey. „Random Forest." (2024.), adresa: https://www.wallstreetmojo.com/random-forest/ (pogledano 17. 1. 2024.).

[41] A. Christopher. „K-Nearest Neighbor." (2021.), adresa: https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4 (pogledano 28. 8. 2023.).

[42] A. Joby. „K Nearest Neighbor or KNN Algorithm And It's Essence in ML." (), adresa: https://learn.g2.com/k-nearest-neighbor (pogledano 17. 1. 2024.).

[43] curtin.edu.au. „Introduction to statistics." (), adresa: https://uniskills.library.curtin.edu.au/numeracy/statistics/data-variable-types/ (pogledano 25. 1. 2024.).

[44] abs.gov.au. „Variables." (), adresa: https://www.abs.gov.au/statistics/understanding-statistics/statistical-terms-and-concepts/variables (pogledano 25. 1. 2024.).

[45] B. Data. „What Is a Dataset? Definition, Use Cases, Benefits, and Example." (2023.), adresa: https://medium.com/@Bright-Data/what-is-a-dataset-definition-use-cases-benefits-and-example-9aaf5ecc301e (pogledano 25. 1. 2024.).

[46] w3schools.com. „Machine Learning - Mean Median Mode." (), adresa: https://www.w3schools.com/python/python_ml_mean_median_mode.asp (pogledano 5. 2. 2024.).

[47] w3schools.com. „Machine Learning - Standard Deviation." (), adresa: https://www.w3schools.com/python/python_ml_standard_deviation.asp (pogledano 5. 2. 2024.).

[48] w3schools.com. „Machine Learning - Percentiles." (), adresa: https://www.w3schools.com/python/python_ml_percentile.asp (pogledano 5. 2. 2024.).

[49] R. Mulla. „Exploratory Data Analysis with Pandas Python 2023." (2021.), adresa: https://www.youtube.com/watch?v=xi0vhXFPegw (pogledano 5. 2. 2024.).

[50] R. RAHMAN. „Heart Attack Analysis Prediction Dataset." (2021.), adresa: https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset (pogledano 5. 2. 2024.).

[51] fedesoriano. „Heart Failure Prediction Dataset." (2021.), adresa: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction (pogledano 5. 2. 2024.).

[52] geeksforgeeks.org. „Confusion Matrix in Machine Learning." (), adresa: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/ (pogledano 5. 2. 2024.).

[53] C. N. aka data professor. „Machine Learning in R: Building a Classification Model." (2019.), adresa: https://www.youtube.com/watch?v=dRqtLxZVRuw (pogledano 5. 2. 2024.).

Popis slika

Popis tablica