**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Klasifikacija pacijenata na temelju mogućnosti oboljenja od bolesti srca**

**Josip Rizner**

**Osijek, 2022.**

**SADRŽAJ**

[**1. UVOD 3**](#_Toc106973030)

[**2. TEORETSKE OSNOVE KORIŠTENIH MODELA 4**](#_Toc106973031)

[**2.1. Logistička regresija 4**](#_Toc106973032)

[**2.2. Nasumične šume odlučivanja 5**](#_Toc106973033)

[**2.3. Stroj potpornih vektora 6**](#_Toc106973034)

[**2.4. Gaussov Naivni Bayes klasifikator 7**](#_Toc106973035)

[**2.5. K-najbližih susjeda 7**](#_Toc106973036)

[**3. OBRADA PODATAKA 9**](#_Toc106973037)

[**3.1. Opis izvornih podataka 9**](#_Toc106973038)

[**3.2. Obrada izvornih podataka 10**](#_Toc106973039)

[**4. PREGLED PROGRAMSKOG KODA KORIŠTENOG ZA UČENJE I TESTIRANJE MODELA 12**](#_Toc106973040)

[**5. ANALIZA REZULTATA 13**](#_Toc106973041)

[**6. TESTIRANJE APLIKACIJE 14**](#_Toc106973042)

[**7. ZAKLJUČAK 15**](#_Toc106973043)

[**8. LITERATURA 16**](#_Toc106973044)

[**9. POPIS SLIKA 17**](#_Toc106973045)

# UVOD

Bolesti srca i krvnih žila najčešći su uzrok smrti u Hrvatskoj, a među njima, na prvom je mjestu ishemijska bolest srca. Prema statističkim podacima svaki drugi građanin Hrvatske umire zbog bolesti srca i krvnih žila. Uz ishemijsku bolest srca, među 10 vodećih uzroka smrti u Hrvatskoj su još cerebrovaskularne bolesti te zatajenje srca.

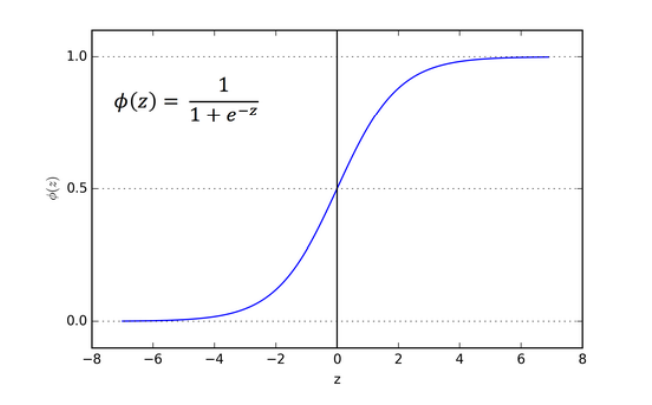
U ovom projektnom zadatku pristupa se problemu klasifikacije pacijenata na temelju mogućnosti oboljenja od bolesti srca pomoću pet modela. To su modeli : stroj potpornih vektora, k najbližih susjeda, logistička regresija, nasumične šume odlučivanja i Gaussov naivni Bayes. Za svaki model provedena je analiza uspješnosti i rezultati modela su uspoređeni. Isto tako, kreirano je korisničko sučelje za pružanje novih podataka modelima u svrhu vršenja predviđanja, a implementirano je u obliku Streamlit aplikacije.

Na početku ovog pisanog rada sažeto su objašnjene teoretske osnove svakog modela i prikazani su koraci koji su napravljeni prije samog rješavanja problema, što uključuje pregled podataka i obradu tih podataka korištenjem programskog jezika Python. Nakon obrade podataka, korištenjem istog programskog jezika i biblioteke scikit-learn, spomenuti modeli primjenjeni su na podatke, točnije, naučeni su na dobivenom skupu za učenje i ispitani na skupu za ispitivanje. Na kraju, rezultati se analiziraju te se testira implementirana aplikacija.

# TEORETSKE OSNOVE KORIŠTENIH MODELA

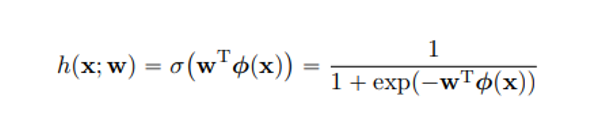
## Logistička regresija

Logistička regresija klasifikacijski je algoritam, ali se naziva regresija jer se koriste jako slične tehnike kao kod linearne regresije. Jedan je od najšire korištenih algoritama u strojnom učenju. Pri takvoj klasifikaciji, izlazne su vrijednosti diskretne (y=0 ili y=1), a vrijednost modela 0 ≤ ℎ𝜃(𝑥) ≤ 1. Logistička može se koristiti i za višeklasnu klasifikaciju. Kako bi definirali model, prvo je potrebno definirati aktivacijsku funkciju koja se naziva sigmoidna ili logistička (logistic) funkcija. Njezina definicija može se vidjeti na slici 2.1.



**Slika 2.1.** Logistička funkcija

Tri karakteristike koje sigmoidu čine dobrim odabirom za aktivacijsku funkciju: funkcija „gnječi“ izlaz na interval od 0 do 1, oblikom je slična funkciji praga i funkcija je derivabilna na cijeloj domeni. Iz navedenog se dolazi do modela logističke regresije koji se može vidjeti na slici 2.2.

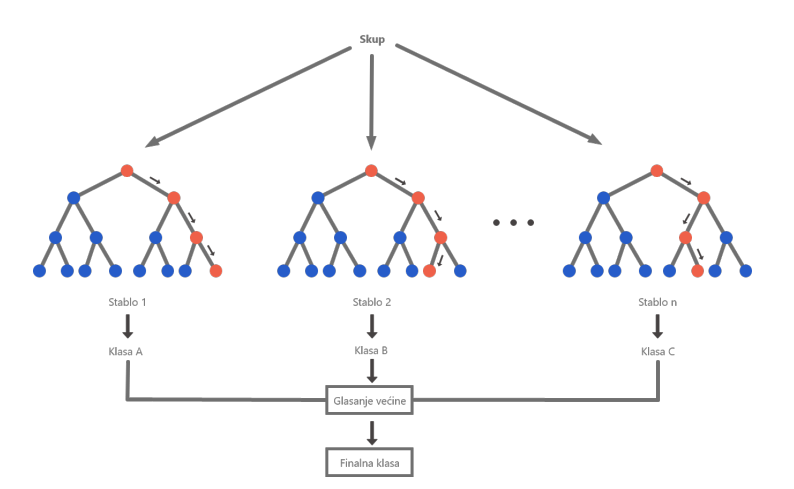


**Slika 2.2.** Model logističke regresije

σ predstavlja aktivacijsku funkciju omotanu oko linearnog modela regresije, odnosno, predstavlja sigmoidu. Vektor w je vektor parametara ili težina, a Φ(x) je funkcija preslikavanja koja kao argument ima primjer x. Klasifikacija se obavlja u prostoru gdje se nalazi granica, to jest, gdje se nalazi linearni model regresije. On predstavlja granicu i ovisno na kojoj strani granice se nalazi primjer spadat će u određenu klasu. Također, udaljenost od same granice, predstavlja sigurnost modela u odabir klase koju je odabrao za odgovarajući primjer.

## Nasumične šume odlučivanja

Algoritam nasumičnih šuma gradi veliki broj nekoreliranih stabala. Pri klasifikaciji stabla predstavljaju glasove te algoritam očitava najmnogobrojniji glas, dok pri regresiji algoritam pronalazi prosječnu vrijednost. Algoritam nasumičnih šuma jednostavan je za treniranje i podešavanje. Kao posljedica, slučajne šume su jako popularan algoritam. Prednosti ovog algoritma su učinkovitost, a nedostatci su sklonost pretjeranoj prilagodbi, pogotovo ako u podacima ima šuma.



**Slika 2.3.** Nasumične šume odlučivanja

## Stroj potpornih vektora

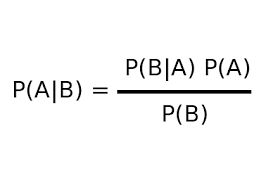
Support Vector Machines je skup nadziranih metoda učenja koje se koriste za klasifkaciju, regresiju i vanjsko otkrivanje (engl. *outlier detection*). Cilj ove metode je odvajanje pozitivnih primjera od negativnih primjera s maksimalnom granicom, a granica je udaljenost od hiperravnine do najbližeg pozitivnog ili negativnog primjera, tj. cilj je pronaći hiperravninu u n-dimenzionalnom prostoru koji jasno razdvaja podatkovne točke. Ti primjeri nazivaju se potporni vektori (engl. *support vectors*) . Ovi algoritmi koriste teoriju minimalizacije strukturnog rizika. Algoritmi bazirani na SVM upotrebljavaju se za klasifikaciju nekoliko primjera u dvije različite klase. Ovaj algoritam nalazi hiperravninu između te dvije klase tako da odvajanje granica između te dvije klase postaje maksimalno. Klasifikacija testnog primjera ovisi o strani hiperravnine, odnosno strani gdje se testni primjer nalazi. Ulazne značajke mogu se preslikati i u prostor visokih dimenzija, ali u tom slučaju, za smanjenje računskih troškova obuke i postupka ispitivanja u prostoru visokih dimenzija, koriste se neke funkcije kernela.



**Slika 2.4.** Stroj potpornih vektora

## Gaussov Naivni Bayes klasifikator

Naivni Bayes-ovi klasifikatori spadaju u obitelj jednostavnih vjerojatnosnih klasifikatora zasnovanih na Bayesovom teoremu sa strogim pretpostavkama neovisnosti među značajkama. Istaknuta slika (značajka) je izraz – s P(A|B) kao posteriornom vjerojatnošću, P(B|A) s likelihood, P(A) je razred priorne vjerojatnosti, a P(B) je prediktor priorne vjerojatnosti.

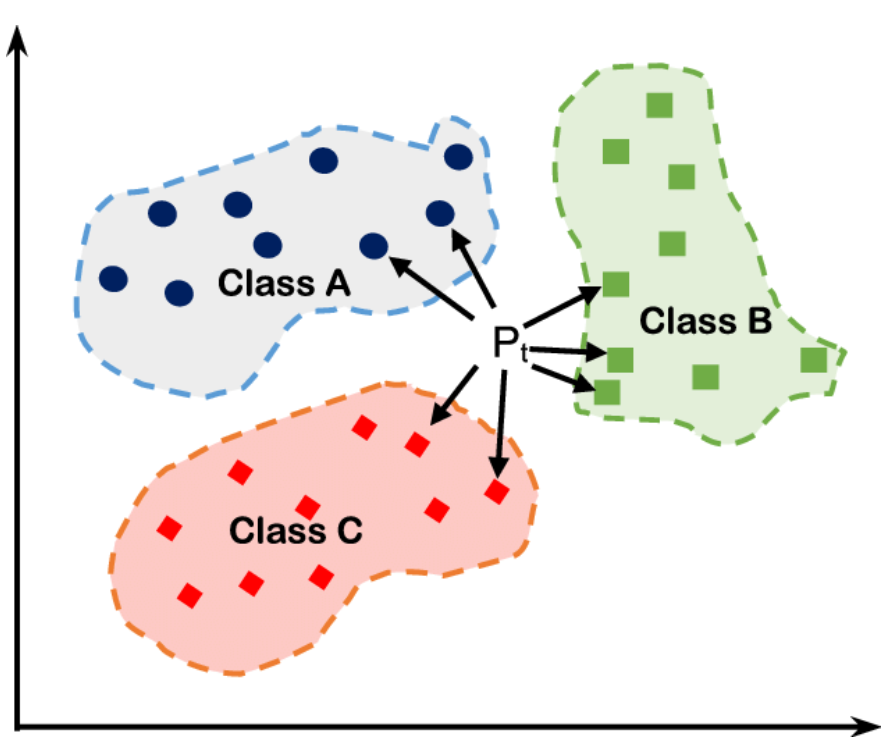


**Slika 2.5.** Naive Bayes - formula

Naive Bayes klasifikatori pokazali su vrlo dobrima u stvarnim situacijama, kao što je klasifikacija dokumenata i filtriranje neželjene pošte. Zahtijevaju malu količinu podataka u skupu za učenje za procjenu potrebnih parametara. Vrlo je jednostavan algoritam za implementaciju i dobri rezultati su postignuti u većini slučajeva. Klasifikatori Naive Bayesa mogu biti iznimno brzi u usporedbi sa sofisticiranijim metodama. Odvajanje raspodjele klasnih uvjetnih značajki znači da se svaka distribucija može samostalno procijeniti kao jednodimenzionalna distribucija.

## K-najbližih susjeda

Algoritam k-najbližih susjeda je jedan od jednostavnijih algoritama za razumijevanje i implementaciju, te se kao takav često koristi. Kada se ispituje klasa nekog novog podatka, tada algoritam provjerava postojeće kategorizirane podatke, te traži k onih koji su mu najsličniji po ulaznim parametrima. Zatim algoritam gleda kojoj klasi pripada najviše susjednih podataka, te na temelju toga dodjeljuje tu kategoriju novom podatku.



**Slika 2.6.** K-najbližih susjeda

# OBRADA PODATAKA

## Opis izvornih podataka

Podaci su preuzeti sa stranice UCI Machine Learning repository [1]. Podaci o dijagnozi bolesti srca prikupljeni su na četiri mjesta, a to su:

* Klinika u Clevelandu
* Mađarski institut za kardiologiju, Budimpešta
* V.A. Medicinski centar, Long Beach, CA
* Sveučilišna bolnica, Zürich, Švicarska

Podaci originalno imaju 76 atributa, ali samo je 14 zapravo iskoristivo te se svi prethodno objavljeni eksperimenti s ovim skupom podataka baziraju na tih 14 atributa. Svi atributi imaju numeričku vrijednost. Koriste se sljedeći atributi:

* Dob pacijenta
* Spol pacijenta
* Tip boli u prsima
* Krvni tlak prilikom mirovanja [mm Hg]
* Kolesterol [mg/dl]
* Šećer u krvi natašte
* Rezultati mjerenja elektrokardiografije prilikom mirovanja
* Postignut maksimalni broj otkucaja srca
* Angina izazvana vježbanjem
* ST depresija izazvana vježbanjem u odnosu na mirovanje
* ST segment – nagib prilikom vježbe
* Broj velikih žila obojenih fluoroskopijom
* Oblik talasemije
* Dijagnoza bolesti srca (atribut koji se predviđa)

U izvornim podacima ima dosta zapisa u kojem nedostaju stupci odnosno atributi. Izgled izvornih podataka može se vidjeti na slici 3.1.

Slika na kojoj se prikazuje stol

Opis je automatski generiran

**Slika 3.1.** Izvorni podatci

## Obrada izvornih podataka

Izvorne podatke prvo trebamo obraditi kako bi ih mogli koristiti za treniranje modela te kako bi dobili što bolje rezultate. Programski kod koje se koristi za obradu podataka može se vidjeti na slici 3.2.

Slika na kojoj se prikazuje tekst

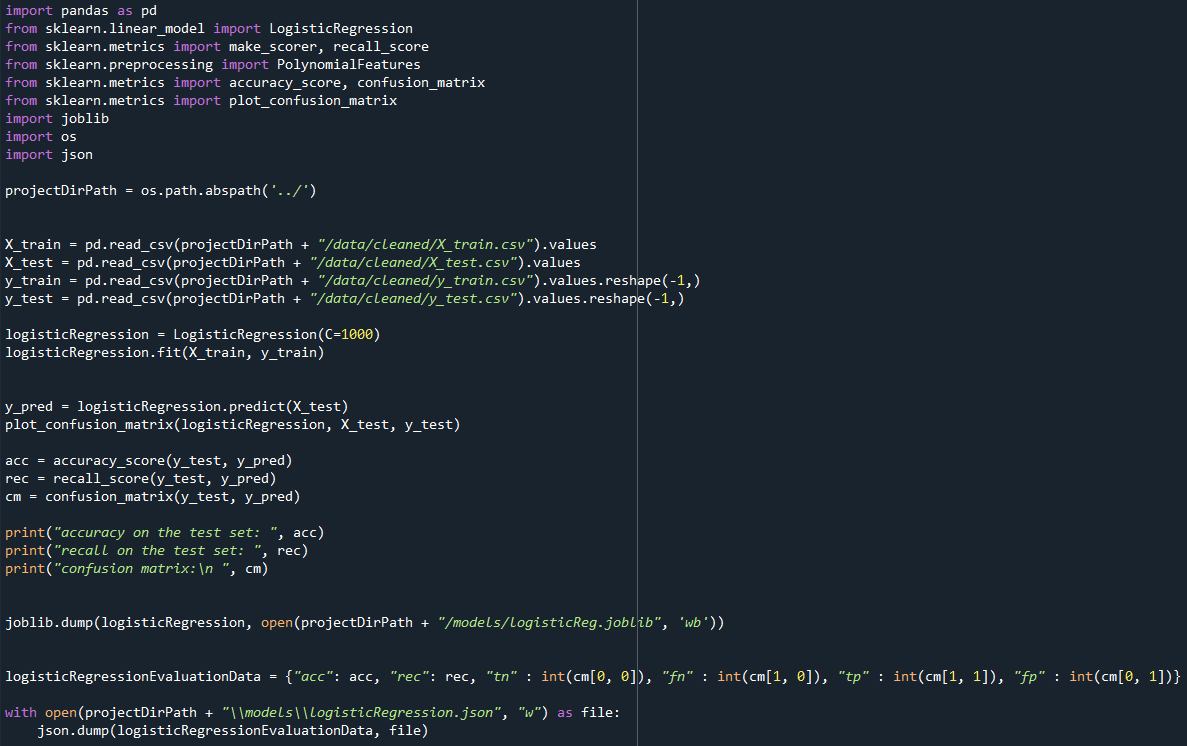
Opis je automatski generiran

**Slika 3.2.** Obrada podataka - programski kod

Za obradu podataka korištene su *python* biblioteke: *pandas*, *numpy*, *sklearn* i os. Na početku je pročitana *csv* datoteka sa izvornim podatcima korištenjem *pandas* biblioteke. Nakon toga su sve vrijednosti koje nedostaju, a označene su sa znakom *?*, zamijenjene s *np.nan*. Nakon toga se podatci razdvajaju na matricu X i vektor y. Dodatno, u vektoru y nalaze se vrijednosti od 0 do 4, a predstavljaju razinu prisutnosti bolesti srca kod pacijenta. Vrijednosti od 1 do 4 zamijenit ćemo s 1 i to će predstavljati dijagnozi oboljelosti od srca, a 0 će ostati 0 i to će značiti da je pacijent zdrav. Nakon toga, pomoću biblioteke *sklearn*, kreira se objekt klase *SimpleImputer* koji se koristi za popunjavanje vrijednosti koje nedostaju. Sada kada imamo sve podatke, kreiramo objekt klase *MinMaxScaler* kojim skaliramo sve podatke svakog stupca u interval između 0 i 1. Sada kada su podatci obrađeni, razdvajamo ih u set za treniranje te set za testiranje pomoću funkcije *train\_test\_split*. Petina podataka se koristi za testiranje dok se ostatak koristi za treniranje modela. Na kraju se setovi podataka za treniranje i testiranje spremaju u obliku *csv* datoteke kako bi im se kasnije moglo pristupi te ih koristiti za treniranje, odnosno testiranje modela.

# PREGLED PROGRAMSKOG KODA KORIŠTENOG ZA UČENJE I TESTIRANJE MODELA

Treniranje i testiranje različitih modela vrlo je slično pa će u ovom poglavlju biti prikazan primjer za samo jedan model, odnosno za logičku regresiju.



**Slika 4.1.** Programski kod korišten za treniranje i testiranje modela

# ANALIZA REZULTATA

# TESTIRANJE APLIKACIJE

# ZAKLJUČAK

# LITERATURA

[1] UC Irvine Machine Learning Repository, dataset -https://archive.ics.uci.edu/ml/datasets/Heart+Disease

# POPIS SLIKA

[**Slika 2.1.** Logistička funkcija 4](#_Toc106973086)

[**Slika 2.2.** Model logističke regresije 4](#_Toc106973087)

[**Slika 2.3.** Nasumične šume odlučivanja 5](#_Toc106973088)

[**Slika 2.4.** Stroj potpornih vektora 6](#_Toc106973089)

[**Slika 2.5.** Naive Bayes - formula 7](#_Toc106973090)

[**Slika 2.6.** K-najbližih susjeda 8](#_Toc106973091)

[**Slika 3.1.** Izvorni podatci 10](#_Toc106973092)

[**Slika 3.2.** Obrada podataka - programski kod 11](#_Toc106973093)