**UNIVERZITET METROPOLITAN BEOGRAD**

**FAKULTET FEFA**

Klod Kolaro

**Komparativna analaiza klasifikacionih algoritama mašinskig učenja i njihov uticaj na tačnost finansijskih modela predikcije**

Doktorska disertacija

Beograd, 2022

UNIVERSITY METROPOLITAN BELGRADE

FACULTY OF FEFA

Klod Kolaro

**Comparative analysis of clasiffication machine learning algorithms and their impact on the accuracy of financial prediction models**

Doctoral Dissertation

Belgrade, 2022

Mentor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. dr Milan Nedeljković, redovni profesor

Univerzitet Metropolitan, FEFA

Članovi komisije: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. dr

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof . dr

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. dr

Datum odbrane:

**Komparativna analaiza klasifikacionih algoritama mašinskig učenja i njihov uticaj na tačnost finansijskih modela predikcije**

**Rezime:**

Predmet istraživanja disertacije je komparativna analiza najčešće primenjivanih[[1]](#footnote-2) klasifikacionih algoritama mašinskog učenja (*ML*), koja će biti zasnovana na poređenju senzitivnosti svakog od analiziranih *ML* modela. U studiji slučaja, algoritmi su primenjeni, sa ciljem predviđanja stečaja kompanije, na osnovu grupe finansijskih indikatora poslovanja. Rešavanjem ovog binarno klasifikacionog problema, pokazaće se uticaj koji mašinsko učenje (*ML*) ima na tačnost predikcije, , odnonsno broju tačno klasifikovanih kompanija u stečaju, kao i uticaj koji *ML* ima na bolje razumevanju podataka, koji opisuju određene ekonomske fenomene i događaje.

U disertaciji su analizirani inferencijalni klasifikacioni algoritmi, koji pored predikcije, imaju i funkciju da opišu međusobnu zavisnost, i uticaj koji varijable (atributi) modela imaju na rezultat predikcije. Tumačenjem ovakvih modela, postiže se bolje razumevanju date pojave ili događaja. Pored ovih, predmet istraživanja su i tzv. *black box* algoritmi mašinskog učenja, čija je osnovna namena postizanje maksimalne tačnosti predikcije, dok zbog svoje kompleksnosti ove prediktivne modeli nije moguće tumačiti.

Klasifikacioni algoritmi *ML* , koji su predmet ove disertacije, spadaju u grupu diskriminativnih modela, kojima se na direktan način dolazi da uslovne raspodele verovatnoće zavisne promenjive, u funkciji nezavisnih varijabli (prediktora). Funkcija koja opisuje prediktivni model, određuje položaj granične hiperavni, koja na najbolji način razdvaja uzorke različitih klasa (kategorija). U odnosu na položaj u odnosu na hiperravan, ispitivani (novi uzorak) biće dodeljen odgovarajućoj klasi. Pored diskriminativnih, neki od analiziranih algoritama, spadaju u grupu generativnih modela, kod kojih se na indirektana način primenon Bajesove teoreme, na osnovu zajedničke distribucije verovatnoće varijabli različitih klasa, dolazi do predikcije klase kojoj ispitivani uzorak pripada. Jedan od ciljeva istraživanja je da pokaže koja od ove dve grupe algoritama, više odgovara rešavanju binarno klasifikacionih problema, u slučaju velikog broja međuzavisnih varijabli, koji nemaju normalnu distribuciju, što su osnovne karakteristike podataka, iz studije slučaja.

Poseban akcenat istraživanje je primena mašinskog učenja u ekonomiji. Pokazano je na primeru linearnog Altman z -score ekonomkog modela (Edward Altman, 1967), kako se primenom mašinskog učenja može povećati tačnost predikcije i postići bolje razumevanje podataka, otkrivanjem njihove međusobne zavisnosti i šablona pojavljivanja.

U uvodnom poglavlju, predstavljen je pojam digitalne transformacije – sveobuhvatne promene u poslovanju priemenom novih tehnologija, kako bi kompanije sačuvale konkurentsku prednost. Ovo strateška odluka, dovodi do promene u kompanijskom lancu vrednosti, kriranjem novih poslovnih modela, transformacijom operacija poslovanja i korisničkog iskustva.

Digitalni podaci i napredna analitika kojom se podaci pretvaraju u korisne informacije i znanje, su ključni akceleratori i pokretači ovih promena[[2]](#footnote-3). Sve veća dostupnost, kao i razvoj analitičkih softverskih alata, učinili su podatke ključnim asetom kompanije. Nestrukturirani podaci, koji najviše doprinose njihovom eksponencijalnom rastu (količine digitalnih podataka se u proseku dupliraju na dve godine[[3]](#footnote-4) ), nije mogće analizirati klasičnim analitičkim alatima. Mašinsko učenje kao podgrupa veštačke inteligencije, primenjuje se u analizi podataka svih struktura. Tako se primenom *ML* u ekonomiji, mogu analizirati i podaci, koji mogu imati uticaj ne određenu ekonomsku pojavu ili dogoađaj, a koji pre pojave *ML*, nisu mogli biti obuhvaćeni analizom.

Dominantni metodi odlučivanja u ekonomiji su linerna i logistička regresija, iz razloga velike interpretabilnosti modela. U poglavlju dva, linearni regresioni model je detaljno opisan, zajedno sa odgovarajućim metodama optimizacije i regularizacije.

U poglavlju tri, detaljno su analizirani klasifikacioni algoritmi mašinskog učenja. Pored logističke regresije, koja zajedno sa linearnom spada u porodicu *Generalized* linearnih modela *GLM* (Nelder i Wedderburn, 1972 ), analizirani su klasifikacioni modeli na bazi stabla odlučivanja, model *K* najbližih suseda, *Naive Bayes* i *Support Vector Machine*. Kako bi reazultati analize bili uporedivi, za određivanje parametara ovih modela, primenjen je metod kros validacije (eng. *cross validation*), sa istim brojem podskupova podataka, kao i brojem ponavljanja.

Tačnost svakog modele, određena je matricom konfuzije (eng. *confusion matirix*) ili matricom greške. Elementi matrice, predstavljaju broj tačno i netačno klasifikovanih uzoraka različitih klasa, iz kojih se mogu dobiti vrednosti senzitivnosti i specifičnosti, svakog modela (poglavlje četiri).

Primenjene metode pretprocesiranja izvornih podataka, opisane su u poglavlju pet. Kako su podaci iz studije slučaja nekompletni, pristupilo se dijagnozi i imputaciji nepostojećih podataka, zameni ekstremnih vrednosti, kao i sklairanju podataka. Zbog postojanja velikog broja varijabli (atributa) i njihove međuzavisnosti, opisan je postupak Analize principalnih komponenti - *PCA* (eng. *Principal Compnent analysis*), kojim je moguće smanjiti dimenzionalnost podataka, kao i eiliminisati pojavu multikolinearnosti.

Rezultati komparativne analize dati su u zaključku.

Izvorni podaci, studije slučaja, detaljno su opisani u prilogu A.1. U prilogu A.2 prikazani su razvijeni klasifikacioni algoritmi, primenom *R* statitističkog softvera.

**Ključne reči**: mašinsko učenje, klasifikacija, predikcija, stečaj kompanija, ekonomija

**Comparative analysis of clasiffication machine learning algorithms and their impact on the accuracy of financial prediction models**

**Abstract:**

**Keywords:** machine learning, classification, prediction, corporate bankruptcy, economy

**SADRŽAJ**

**1. Digitalna transformacija**

1.1 Nauka o podacima i veštačka inteligencija

1.2 Mašinsko učenje

1.2.1 Osnovni tipovi mašinskog učenja

1.2.2 Primena mašinskog učenja u ekonomiji

1.2.3 Altman z-score regresioni model

**2. Linearna regresija, *LR***

2.1 Metoda najmanjim kvadrata, *OLS*

2.2 Metoda maksimalne izglednosti, *MaxL*

2.3 Regularizacije regresionih modela

**3. *ML* klasifikacioni modeli**

3.1 Logistička regresija

3.1.1 Generalni linearni modeli, *LDA*

3.1.2 Gaussian diskriminantna analiza, *GDA*

3.2 *Naive Bayes*, NB

3.3 K - najbližih suseda , *K-NN*

3.4 *Support Vector Machines, SVM*

3.4.1 Određivanje položaja granične hiperravni*, SVM*

3.5Klasifikacioni modeli na bazi stabla odlučivanja

3.5.1 Klasifikaciona stabla, *DT*

3.5.*2 Random Forest*, *RF*

3.5.3 *ADA Boost-* binarni klaasifikator

3.5.4 *Gradient tress boosting, GTB*

**4. Analiza tačnosti klasifikacionih modela**

**5. Pretprocesiranje i transformacija podataka**

**6. Zaključak**

6.1 Regresioni modeli

6.2 Modeli na bazi stabla odlučivanja

6.3 *KNN*

*6.4 SVM*

6.5Uporedna analiza performansi modela *ML* i *Altman Z-score*

**7. Literatura**

**Prilog A.1 Opis podataka**

A.1.1 Priprema podataka

**Prilog A.2 Prikaz razvijenih softverskih algoritama**

**1. DIGITALNA TRANSFORMACIJA**

Poslovanje kompanija u novom dinamičnom digitalnom svetu, karakteriše visok stepen konkurencije, koja nudi visoko personalizovane, inovativne proizvode i usluge, nastale primnom novih digitalnih tehnologija (Atkinson, 2005). Nove tehnologije, kreiraju nove poslovne mogućnosti, ali u isto vreme predstavljaju opasnost za kompanije, koje na vreme ne uoče njihov transformacijski potencijal. Nemogućnost sagledavanja uticaja eksponencijalnog razvoja digitalnih tehnologija, za kompanije nastale pre početka komercijalizacije Interneta (zvaćemo ih tradicionalne ), znači zadržavanje *status quo*. Nastavak poslovanja, koji je baziran na strateškim asetima i znanjima, koje u digitalnom svetu više ne mogu biti osnov konkurentske prednosti, može dovesti do nestanka komapnija sa tržišta (eng. *digital* *Darwinism)* ( Roger, 2016). Tradicionalne kompanije, tako mogu postati taoci svog vlastitog uspeha, planirajući budućnost, samo na osnovu prethodnog iskustva (eng. *competence trap*) (Roger, 2016)**.**

Kada u poslovnom smislu, govorimo o disruptivnim promenama na tržištu prouzrokovanim digitalnim tehnologijama, potrebno je napraviti razliku između inovacije i disrupcije. Pod disruptivnom promenom, smatra se promena u postojećoj industriji, gde konkurencija (eng. *challanger*) kreira značajno veću vrednost za kupca, na jedinstven način, koju tradicionalna kompanija direktno ne može ponuditi ( Roger, 2016). Dok se inovacija, može relativno lako iskopirati i primeniti, to nije slučaj sa disrupcijom. Kako je disrupcija po prirodi asimetrična, odnosno ne dolazi iz industrije u kojoj tradicionalna kompanija posluje, postoje značajne razlike u lancu vrednosti (eng. *value chain*) i strukturi troškova između *challanger* - a i tradicionalne kompanije. Iz tog razloga, disruptivna promena se ne može jednostavno primeniti, već zahteva transformaciju kompanije.

U zavisnosti od prioriteta i prirode konkurencije, razlikujemo tri pristupa digitalnoj transformaciji:

* Eksterni, gde kompanija primenjuje nove tehnologije kao bi unapredila korisničko iskustvo, kroz sve raspoložive digitalne kanale komunikacije. Fokus je pre svega na dizajniranju novog načina interakcije i kolaboracije sa klijentima, koji je bazirana na personalizovanoj ponudi proizvoda i servisa, kao i digitalnih sadržaja, koji su u skladu sa potrebama, inetersima i navikama kupca.
* Interni, gde je cilj povećanje efikasnosti proizvodnih operacija, poslovnog odlučivanja i organizacione strukture. Primenom robotike, veštačke inteligencije, Interenta stvari (eng. *IoT*), omogućava se *m2m* (eng. *machine to machine*) komunikacija i kreiranje sajber- fizičkih sitema, kojima se postiže visok stepen automatuzacije i povećanje operativne efikasnosnti. Velika raspoloživost digitalnih podataka, uz napredak *ICT* infrastrukture i softverskih analitičkih platformi, omogućavaju pametno upravljanje proizvodnjom, kao i donošenje poslovnih odluka, na bazi preporuka prediktivnih i preskriptivnih (eng. *prescribing* ) algoriatama.
* Holistički, svobuhvatna transformacija poslovanja, kojom se utiče na sve segmente i funkcije, te se tako dolazi do novog digitalnog poslovnog modela (Kaufman i Horton, 2015).

U doktorskoj disertaciji, razmatraju se interni aspekti digitalne transformacije, koji pre svega imaju za cilj kvalitetno i efikasno odlučivanje, na osnovu anlize relevantnih podataka. Sveopšta digitalizacija i razvoj tehnologija, omugućili su jednostavan pristup, i brzu obradu velikih količina internih i eksternih podataka svih struktura. Dok se poslovnom analitikom (eng*. business intlligence*), analizira šta se i zbog čega u prošlosti desilo, primenom prediktivnih algoritama mašinskog učenja, moguće je predvideti buduće događaje. U disertaciji anliziramo klasifiakcione algoritme mašinskog učenja, koji na osnovu grupe finansijskih indikatora poslovanja, mogu sa određenom verovatnoćom predvideti stečaj kompanije.

**1.1 Nauka o podacima i veštačka inteligencija**

Automatsko ili mašinsko učenje (eng. *machine learning* – *ML*), predstavlja tehnolgiju kojom se kompijuterski softverski sistemi samostalno razvijaju, odnosno ’uče’ , na osnovu obrade velike količine ulaznih podatka, bez potrebe da se programiraju eksplicitno (Samuel , 1959). Pod samostalnim učenjem, podrazumeva se postupak optimizacije kompijuterskog programa, sa povećanjem količine procesiranih podataka (T. Mitchell, 1998).

Razvoj i paktična primena *ML* algoritama, postaje moguća sa rastom količine raspoloživih podataka, na kojima se ovi algoritmi mogu razvijati, ’učiti’. Digitizacija, postupak kojim se se konvertuju fizički objekti i atributi u digitalne, i tako stvara veza fizičkog sveta i kompijuterskih sistema, započet je još krajem sedamdestih godina. Ovo je dovelo do eksponencijalnog rasta količina digitalnih podataka. Najveći uticaj na ovaj rast, ima primena senzora, odnosno veliki broj *IoT* (eng. *Interent of Things*) konektovanih uređaja. Pored ovih, mašinski generisanih podataka, ljudi svojim danas uobičajenim aktivnostima, a pre svega preko socijalnih mreža, značajno utiču na ovaj eksponencijalni trend rasta. Oko 80% kreiranih podatak na ovaj način, spada u gupu nestrukturiranih (podaci neodređenih struktura). Zbog velikog obima, brzine nastajanja i svoje raznolikosti, ovakve podatke nazivamo Velikim podacima (eng. *Big data*) (Provost & Fawcett, 2013).

Paralelno sa fenomenom Velikih podataka, sličnom brzinom razvijaju se i tehnologije koje omogućavaju njihovo procesiranje, prenos i skladištenje (čuvanje).

Performanse procesora, koji izvršavaju programske instrukcije, dupliraju se u proseku svaka 24 meseca (Moor, 1975). Razvojem kvantnih kompijutera, eliminisan je osnovni ograničavajući faktor daljeg rasta procesorske snage, usled fizičkog ograničenja prostora za smeštaj tranzistora na procesorkom čipu (eng. *CPU*).

Kapaciteti za prenos podataka optičkim kablovima, u proseku se dupliraju svakih 9 meseci (Butter,2001), dok se kapaciteti diskova za smeštaj podataka, udvostručuju u proseku na 13 meseci (Kryder, 2015).

Velike količine podataka, kao i mogućnost efikasnog dostupa i obrade, dovelo je do široke primene poslovne *data* analitike (eng. *data analytics*), koja je postala sastavni deo udlučivanja na svim nivoima upravljanja i rada. Kada se poslovna *data* analitika koristi, kako bi se objasnio događaj i odgovorilo na pitanje šta se i zbog čega dogodilo, govorimo o opisnoj (eng. *describing*), odnosno dijagnostičkoj (eng. *diagnostic*) analitici (Provost i Fawcett, 2013). Kada se strukturirani i nestrukturirani podaci, analiziraju sa ciljem da se predvide dogođaji, otkriju zavisnosti i šabloni u podacima, primenom statističkih metoda i alogoritama *ML*, govorimo o prediktivnoj analitici (Provost i Fawcett, 2013).

**1.2 Mašinsko učenje**

*ML* i duboko učenje su podgupe veštačke inteligencje (eng. *Artificail Intelligence*-*AI*). Pod pojmom veštačke inteligencije, podrazumevamo autonomne inteligente sisteme, koji deluju samostalno, bez ljudske intervencije. Pojam *AI* prvi put je upotrebio John McCarthy 1955 godine, a već sledeće godine, održana je prva konferencija o neuronskim mrežama, koje su bile osnovni predmet istraživanja iz oblasti *AI*, sve do kraja 1970 godine ( Slika 1).

Slika 1. Razvoj veštačke inteligencije i *ML[[4]](#footnote-5)*

Graphical user interface, application, Word

Description automatically generated

Kasnije, primat u razvoju *AI,* preuzimaju algoritmi *ML*, specijalno modeli višeslojnih nuronskih mreža. Oni se zbog svog značaja i široke primene, navode kao posebna podgrupa *ML* – Duboko učenje (eng. *Deep learning* – *DL*).

*DL* se primenjuje u slučajevima klasifikacije kompleksnih podataka. Razvijeni su po analogiji sa neuronskom mrežom ljudskog mozga. Ulazni podaci se iz prvog nivo neuronske mreže(eng. *input layer*), prenose ka skrivenim nivoima (eng. *hidden layers*), gde se sukcesivno transformišu primenom nelinearnih funkcija (eng. *activation* *function*). Sa rastom broja nivoa *hidden* mreže, odnosno sa svakom novom transformacijom, povećava se tačnost modela. U poslednjem, izlaznom nivou mreže (eng. *output layer*), na osnovu primenjinih aktivacionih funkcija, dobijamo konačnu predikciju.

Slika 2. Arhitektura neuronske mreže[[5]](#footnote-6)



Za razliku od tradicionalnog programiranja, gde se na osnovu ulaznih podataka i softverskog programa (logike), dobija rezultat (autput), kod mašinskog učenja se na osnovu ulaznih podataka, iterativnim postupkom učenja algoritma, samostalno (bez potrebe za eksplicitnim programiranjem) kreira *ML* model (Slika 4).

Slika 4. Tradicionalno programiranje (a), Mašinsko učenje (b)

(a) (b)

Ulazni podaci Ulazni podaci

Autput Program

Program Autput

Ovim iterativnim postupkom optimizacije *ML* modela, određuju se vrednosti njegovih parametara, za koje model daje optimalan rezultat. Kada je namena *ML* modela predikcija, konačne vrednosti parametara, koji finalno određuju oblik ciljne funkcije, su one za koje je greška predikcije najmanja (James et.al , 2013). Rast kompleksnosti prediktivnog modela, uobičajeno vodi ka povećanju njegove tačnost, ali se istovremeno povećava rizik od *overfitinga*, čime se smanjuje mogućnosti generalizacije modela, kao i njegovog tumačenje.

U slučaju kada je tačnost predikcije osnovni cilj, a ne razumevanje *ML* modela, kažemo da on predstavljaju crnu kutiju (eng. *black box*) (James et al , 2013). Višeslojne neuronske mreže, spadaju u *black box* algoritme, koji daju veliku tačnost u regresionim i klasifikacionim problemima, ali zbog velikog broja iterativnih nelinearnih transformacija, ovaj model je nemoguće razumeti.

S druge strane, ukoliko je pored predikcije cilj istraživanja i razumevanje zavisnosti između ulaznih i izlaznih varijabli, govorimo o inferencijalnim (eng. *Inference*) *ML* modelima (James et al , 2013).

U doktorskoj disertaciji, bavićemo se komparativnom analizom tačnosti najčešće primenjivanih *ML* klasifikacionih prediktivnih modela, od kojih neki spadaju u *black box*, dok su drugi inferencijalni.

*ML* modele, razlikujemo i prema postupku kojim se dolazi do ciljne funkcije, odnosno konačnog oblika modela. Kada se njen početni oblik pretpostavi, a potom iterativnim postupkom učenja, odrede optimalne vrednosti parametara , keficijenata ciljne funkcije, govorimo o parametarskim metodama *ML* (eng. *parametric*) (Garet et.al , 2014). Kako se iterativnim postupkom učenja, određuju samo vrednosti parametara, a ne i oblik funkcije, ovakvi *ML* modeli su efikasni. Za učenje modela, ne zahteva se velika količina podataka (Shwartz i Shai, 2014). Međutim, ukoliko pretpostavljena funkcija ne opisuje dobro stvarnu relaciju između promenjivih, tačnost predikcije ovih modela je niska.

Kada se oblik funkcije ne pretpostavlja, već se do njega dolazi postupkom učenja, govorimo o neparametarskim metodama *ML* (eng. *non paramtric* ) (Garet et al, 2014). Ovaj potupak zahteva značajne količine trening podataka (podataka za učenje), kako bi se došlo do oblika funkcije, koja najbolje opisuje zavisnost promenjivih. Neparametrski modeli, imaju veću tačnost, ali su skloni *overfiting-*u. Iz tog razloga, moguće je postupkom regularizacije, ograničiti kompleksnost ovakvih modela (Garet et al, 2014).

Podaci na kojima se model razvija, uči, nazivamo trening podacima, dok su test podaci oni, na kojima se proverava njegova tačnost. Trening i test podaci, predstavljaju podskup tzv. univerzalnog skupa (*US*) podataka, iz kojeg se slučajnim uzrokovanjem, osigurava njihova nezavisnosnt, dok je verovatnoća njihovog izbora iz *US*, jednaka za svaki uzorak (eng. *indipendant and indenticaly sampled* ) (Garet, J. et al., 2014). Trening i test podaci postoje u dva osnovna oblika:

* Obeleženi *(*eng*. labeled)* podaci - kod kojih su poznati i ulazni i izlazni podaci. Odnosno kažemo da određuje, ili da *Y* daje kontekst ili značenje podacima *X.* Tako se uzorak (ili observacija), može predstaviti u obliku (). Ulazni podatak () , je vektor sa emenata, koje nazivamo prediktorima ( ili nezavisnim promenjivima), dok izlazni podatak , nazivamo zavisnom ili target promenjivom. Uvodimo sledeću notaciju:

, ,

, ,

- je kolonski vektor zavisne promenjive, reda n x 1

- matrica prediktora, reda n x m :

- uzorak, kolonski vektor reda m x 1

*n* – ukupan broj uzoraka

*m* – ukupan broj prediktora

*R -* skup realnih brojeva

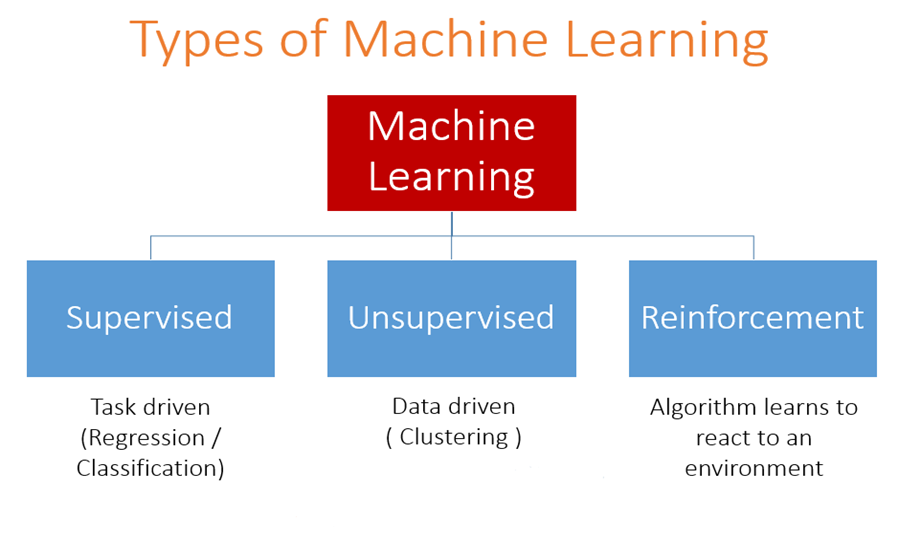
* Neobeleženi *(*eng*. unlabeled data)* podaci, su oni kod kojih ne postoji target promenjiva , već samo ulazni podaci:

, ,

1.2.1 Osnovni tipovi *ML*

Prema problemu koji izučavamo, kao i strukturi podataka nad kojim se model razvija ( obeleženi ili neobeleženi), razlikujemo tri osnovna tipa mašinskog učenja i to: nadgledano (eng. *supervised learning*), nenadgledano ( eng. *unsupervised learning*) i ojačano učenje (eng. *reinforcement learning - RL*) (Slika 5) (Harrington, 2016).

Slika 5. Tipovi *ML*



* Nadgledano učenje

Ukoiko su podaci obeleženi, poznate su vrednosti prediktora i zavisne promenjive , tada govorimo o **nadgledanom učenju**, kojim se razvija model ciljne funkcije , koja na osnovu uzorka prediktora , ima cilj da predvidi vrednost output , što približnije njenoj stvarnoj vrednosti (Slika 6).

Ukoliko je numerička (kvantitativna) promenjiva, onda prediktivni *ML* algoritam predviđa njenu vrednost i nazivamo ga regresionim. Ukoliko je kategorička (može pripadati ograničenom broju kategorija ili klasa), model na osnovu vrednosti , svrstava uzorak u jednu od mogućih klasa zavisne promenjive. Ovakve prediktivne modele zovemo klasifikacionim.

Slika 6. Algoritam nadgledanog učanja

Trening podaci

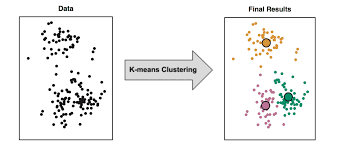
Algoritam učenja ili trening algoritam

)

* Nenadgledano učenje

Kada su podaci neobeleženi, govorimo o nenadgledanim algoritmima *ML*, kojima se analizira struktura podataka i slični uzorci grupišu zajedno u klastere . Kako se svaki uzorak može prikazati kao tačka u *m* dimenzionalnom prostoru (, sličnost uzoraka se najčešće meri njihovom međusobnom razdaljinom. Najbliži uzorci su najsličniji i pripadaju istom klasteru, koji je određen brojem uzoraka i svojim težištem (eng. *centroid*). Ispitivani uzorak, dodeljuje se klasteru čijem je težištu najbliži.

Slika 7. Grupisanje uzoraka u tri klastera[[6]](#footnote-7)

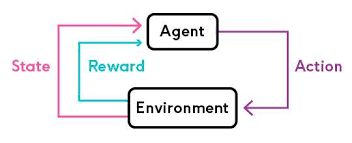


Drugu grupu algoritama nenadgledanog učenja čine oni, kojima se redukuje dimenzionlnost podataka i tako dobijaju ciljne funkcije manje kompleksnosti (eng. *dimension reduction*). Podaci se projektuju u novi prostor, određen osama koje predstavljaju pravce maksimalne varijabilnosti. Ovom transformacijom, eliminiše se eventualno postojanje međuzavisnosti prediktora.

* Ojačano učenje

Ojačano učenje (eng. *reinforcement learning- RL*) ili učenje sa podrškom, bazira se na postupku učenja algoritma, kroz uzastopne pokušaje da se napravi očekivana akcija, odluka (eng. *trial and error*). Inteligentni softverski agent, preduzima autonomno akciju u okruženju, bivajući nagrađen za očekivanu (pravilnu) akciju, dok u suprotnom slučaju biva sankcionisan. Na ovaj način, učeći iz vlastitog iskustva, teži se postizanju maksimalnog broja pravilnih-očekivanih akcija inteligentnog agenta (Slika 8). *RL* se pre svega primenjuje u robotici, autonomnoj vožnji, kao i tzv. *recomendation* sistemima (Sutton i Barto, 2018).

Slika 8. Učenje sa podrškom[[7]](#footnote-8)



1.2.2 Primena *ML* u ekonomiji

Suština ekonomske nauke jeste tumačenje ekonomskih činjenica i istraživanje njihovih međusobnih relacija. Teorijski koncepti i matematički metodi, se često zbog neadekvatnih i nedovoljnih empirijskih istrživanja, ne suočavaju sa činjeničnim podacima, kako bi ekonomska teorija mogla biti proverena u praksi. Ekonometrija ima upravo ulogu da pruži empirijsku verifikaciju ekonomske teorije, koja je u osnovi bazirana na kvalitativnim istraživanjima (Gujarati i Porter, 2009). Jedan od razloga nedovoljne zastupljenosti kvantitatinih empiriskih istraživanja u ekonomiji, jeste nedostatak podataka, njihov upitan kvalitet, nestandardni formati i nemogućnost njihove obrade i analize.

Sa ubrzanom digitalizacijom i atomatizacijom poslovanja, došlo je do eksponencijalnog rasta količina elektronskih podataka. Nove digitalne tehnologije i infrastruktura, omogućavaju danas jednostavan pristupa, prenos , transformaciju i obradu podataka različitih struktura, čime su se stvorili uslovi za razvoj i ekonometrije, ali i sve veću praktičnu primenu *ML* algoritama, u emprijskim istraživanjima u ekonomiji. Tako Varian (2014), opisuje nove mogućnosti primene *ML* u ekonomskoj analizi Velikih podataka, Mullainathan i Spiess (2017) konstatuju sve veću primenu *ML* predikcionih algoritama u ekonomiji.

Kako bi se objasnio ekonomski problem ili pojava, empirijskim istraživanjem, analizira se relacija i uticaj koji ekonomske varijable, imaju na datu pojavu. Dominantni metod u ekonomiji, koji se zbog svoje visoke interpretabilnosti koristi, je linearna regresija, kojom se objašnjava pravac i statistički zanačaj, uticaja nezavisnih ekonomskih varijabli (), na vrednost zavisne promenjive (). Kako je dobijeni eksplanatorni model linerare regresije oblika , kažemo da se tradicionalnim ekonomskim metodama, dolazi do vrednsoti parametr , koji odrđeuje prirodu i intezitet uticaja svake finansijske varijable pojednično, na ekonomski problem koji je predmet istraživanja. Mullainathan i Spiess (2017), navode tako da se tradicionalnim ekonomskim metodama rešava problem.

Primenom nadgledanih metoda *ML*, moguće je značajno proširiti obuhvat ovakve analize. Pored objašnjena pojedinačnih uticaja prediktora (ekonomskih varijabli), na zavisno promenjivu, moguće je na osnovu vrednosti prediktora (), predvideti (, uzimajući u obzir njihovu nelinernu ( ili linearnu) zavisnosnt, kao i sinergetski efekat prediktora, na vrednost . Postupkom regularizacije *ML* modela, moguće je takođe model dodatno pojednostaviti, svođenjem vrednosti linearnih koeficijenata regresije na vrednost nula, za one prediktore koji manje utiču na varijabilnost zavisne promenjive. Ovime se dodatno može uticati na povećanje tačnosti predikcije, kao i na mogućnost generalizacije modela (Hastie et al, 2009). Tako primenom nadgledanih *ML* algoritama u ekonomiji, kažemo da je moguće rešavanje problema (Mullainathan i Spiess, 2017).

Nenadlgledane algoritme *ML* u ekonomskoj analizi, moguće je primeniti za detaljnu strukturnu analizu empirijskih podataka. Ovim postupkom, slični uzorci se grupišu u klastere ili grupe, na način da su razdaljine sličnih uzoraka u klasteru najmanje moguće (minimalna varijabilnost podataka u grupi), a razdaljine težišta klastera različitih grupa uzoraka najveće moguće ( maksimalna varijablinost podataka različitih grupa). Nenadgledanim *ML* algoritmima, kažemo da se mogu rešavati problemi (Mullainathan and Spiess, 2017) i da se na osnovu uočenih sličnosti i razlika, mogu bolje razumeti podaci koji su predmet istraživanja.

Treća moguća primena *ML* algoritama, u ekonomskim istraživanjima, jeste analiza nestrukturiranih podataka, kao što su tekst, video ili slika, koji u klasičnim ekonomskim analizama nisu uzimani u razmatranje, odnosno nije ih bilo moguće analizirati na inteligentan način.

Tabela 1. Razlika između tradicionalne ekonomske analize i dva tipa *ML*: nadgledanog i nenadgledanog. Ekonomska analiza objašnjava ekonomski fenomen, dok *ML* omogućava tačniju predikciju i bolje razumevanje strukture podataka

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pristup | Podaci | Metod | Rezultat | Svrha |
| Tradicionala ekonometrija | Obeleženi ( | Lineran regresija | Model zavisnosti i statističkog značaja promenjivih | problem |
| Nadgledano učenje | Obeleženi ( | *ML*, Nadgledano učenje | Predikcija | problem |
| Nenadgledano učenje | Neobeleženi ( | *ML*, Nenadgledano | Sličnosti, šabloni podataka | problem |

U doktorskoj disertaciji, analiziraćemo *Atman Z-Score* (Edward Altman, 1968), tradicionalni ekonomski model, kojim se opisuje uticaj finansijskih indikatora poslovanja, na mogućnost stečaja kompanije. Pokazaće se, kako se primenom *ML* može povećati tačnost predikcije ( problem), postići bolje razumevanje uticaja većeg broja finansijskih indikatora, kao i njihove međusobne zavisnosti (problem).

1.2.3 Altman Z-score regresioni model

*NYU Stern*, profesor finansija Edward Altman, je primenom diskriminantne analize, razvio model kojim se opisuje uticaj finansijkih indikatora poslovanja, na verovatnoću stečaja kompanije, u periodu od dve godine. Nastao je 1967, kao rezultat analize finansijskih izveštaja 66 kompanija, od kojih je polovina bila u stečaju. Radi se o proizvodnim kompanijama, čija vrednost aktive nije manja od 1mil $. Do 1975 godine, analizirano je dodatnih 70 kompanija, a u periodu od 1996 do 1999 još njih 120, kada je ovaj linearni model dobio svoj konačni oblik.

Altman Z-Score = 1.2A + 1.4B + 3.3C + 0.6D + 1.0E

A = working capital / total assets

B = retained earnings / total assets

C = earnings before interest and tax / total assets

= market value of equity / total liabilities

E = sales / total assets

U zavisnosti od dobijene Z- score vrednosti, može se utvrditi izglednosti stečaja (Tabela 2)

Tabela 2. Granične Z- score vrednosti

|  |  |  |
| --- | --- | --- |
|  | Z Score |  |
| Z<1.8 | 1.8<Z<3 | Z>3 |
| Izgledan stačaj | ‘siva’ zona | Ne postoji rizik od stečaja |

U 2012 godini, novi Z- score model, razvijen je za neproizvodne i kompanije sa tržišta u razvoju:

Z-score (neproizvodne kompanije):

Altman Z-Score = 6.56*A* + 3.26B + 6.72C + 1.05D

Z-score kompanija sa tržišta u razvoju:

Altman Z-Score = 3.25 + 6.56*A*+ 3.26B + 6.72C + 1.05D

D= book value of equity / total liabilities

Tabela 3. Granične Z- score vrednosti za neproizvodne firme i tržišta u razvoju

|  |  |  |
| --- | --- | --- |
|  | Z Score |  |
| Z<1.1 | 1.1<Z<2.6 | Z>2.6 |
| izgledan stečaj | ‘siva’ zona | ne postoji rizik od stečaja |

**2. LINEARNA REGRESIJA, *LR***

Regresioni modeli spadaju u grupu osnovnih algoritama, najčešće primenjivanih u ekonometrijskim analizama. Njima se objašnjava zavisnosnt jedne promenjive (), od jedne ili više nezavisnih promenjivih - prediktora (). Radi se o analizi zavisnosti u jednom smeru, uticaja prediktora na vrednost zavisne promenjive (). Kada je cilj da se regresionim modelom, opiše zavisnost varijabli modela, kažemo da se *LR* primenjuje u svrhu inferencijalne analize (Gareth et al, 2017). Regresioni algoritmi, se takođe primenjuju i u predikciji vrednosti zavisne promenjive, na osnovu poznatih vrednosti prediktora.

Regresioni model, dat je funkcijom , koja daje očekivanu vrednost zavisne promenjive, uslovno od vrednosti prediktora - :

Postupkom optimizacije, na osnovu raspoloživih trening podataka, određuju se optimalne vrednosti regresionih parametara funkcije . U slučaju jednog prediktora ova funkcija ima sledeći oblik:

(1)

Izraz (1) nazivamo determinističkim modelom regresije (Mendenhall et al, 2013). U slučaju jednog prediktora, funkcija pretstavlja regresionu liniju. Na osnovu zakona o univerzalnoj regresiji (Galton, 1885), vrednosti slučajne (stohastičke) promenjive , za date fiksne vrednosti regresiraju ka svojoj srednoj vrednosti. Tako možemo pretpostaviti, da optimalni regresioni mode, prolazi kroz srednju vrednost zavisne promenjive . Kako se radi o slučajnoj promenjivoj, pretpostavlja se da je njena distribucija *Gaussian* normalna (Gudjarati i Poretr, 2009).

Slika 9. Normalna distribucija zavisne promenjive za fiksne vrednosti *x*[[8]](#footnote-9)

Graphical user interface

Description automatically generated with medium confidence

Regresioni model (1), ne opisuje relaciju zavisne promenjive i prediktora na idealan način, odnosno postoje odstupanja očekivanih vrednosti zavisne pormenjive  , od njenih stvarnih vrednosti . Zato se ovaj model prikazuje u *probabalistic* obliku (Mendenhall i dr. 2013), koji nazivamo i regresionom funkcijom populacije (eng. *Population regression function* – *PRF*) (Gudjarati i Poretr 2009) :

(2)

slučajna greška ( eng. *random error*), koju još nazivamo rezidualom, je mera odstupanja stvarne vrednosti zavisne promenjive od očekivane,. Očekivana vrednost izraza (2) dalje je jednaka:

Kako je , sledi da je

Pretpostavlja se da je vrednost prediktora fiksna i nezavisna od reziduala. Tako je njihova kovarijansa, mera njihove linearne zavisnosti, jednaka nuli , . Kako je varijansa reziduala jednaka,

a kako smo pokazali da je , sledi i da je , pa dobijamo da je varijansa reziduala konstantna i jednaka,

Na osnovu pretpostavke da regresiona prava, prolazi kroz srednje vrednosti simetrične distribucije zavisne promenjive, dobili smo da je , dok je varijansa reziduala konstantna i jednaka . Pretpostavljamo dalje da reziduali imaju normalnu distribuciju, te sledi da je ).

Kakao nam podaci iz cele populacije njačešće nisu poznati, cilj je da primenom statističkih metoda na slučajno uzrokovanim podacima, dođemo do najbloljeg modela ( eng. *best fitting line*) linije srednjih očekivanih vrednosti zavisne promenjive koja najbolje aproksimira funkciju . Model, opisan funkcijom nazivamo regresionom funkcijom uzorka ( eng. *Sample regression function*-*SRF*) (Gudjarati i Poretr 2009):

Slika 10. Regresiona funkcija uzorka (*SRF*) i regresiona funkcija populacije (*PRF*)[[9]](#footnote-10)

Graphical user interface, chart

Description automatically generated

Vrednosti regresionih parametara , koji će biti najbolja moguća (optimalna) aproksimacija parametara , regresione funkcije populacije (*PRF*), određujemo primenom dve metode (Gudjarati i Poretr 2009):

* Najmanjih kvadrata ( eng. *Ordinary least squares-OLS*)
* Maksimalne izglednosti (eng. *Maximum Likelihood- MaxL*)

**2.1. Metoda najmanjih kvadrata, *OLS***

Na osnovu pretpostavke da je srednja vrednost reziduala nula, pretpostavljeni oblik multivarijantne regresione funkcije *SRF*, ima oblik:

+++.... (3)

Regresioni model, podrazumeva samo linearnu zavisnost promenjive od regresionih parametara, dok dozvoljavaju nelinearnu zavisnost od prediktora. Ukoliko zavisnost između i prediktora nije linearna, potrebno je transformisati prediktore, najčešće stepenovanjem. Tada govorimo o polinominalnoj regresiji. Pored stepenovanja, često se prediktori transformišu i logaritmovanjem. Moguće je takođe primeniti i postupak tzv. *spline* regresije, kada postoji više modela polinominalne regresije, svaki između prelomnih tačaka (eng. *knots*), najčešće kvartila , vrednosti prediktora (Alboukadel K. , 2017).

Izraz (3) koji definiše *SRF* funkciju, se može pojednostaviti,

(4)

pretpostavili samo da je .

U matričnom obliku izraz (4) postaje,

(5)

gde su:

- vektor prediktora; – vektor patametara linearnog regresionog modela

Kako jednačina (3), definiše hiperravan u *m* dimenzionlanom prostoru, njen optimalan položaj se određuje tako, da suma kavadrata razlika rastojanja između i stvarnih vrednosti , bude minimalna (Slika 10). Funkcija sume kvadrata reziduala, naziva se eng. *cost* ili *objective* funkcija linearnog modela (Alboukadel, 2017) i ima oblik,

(6)

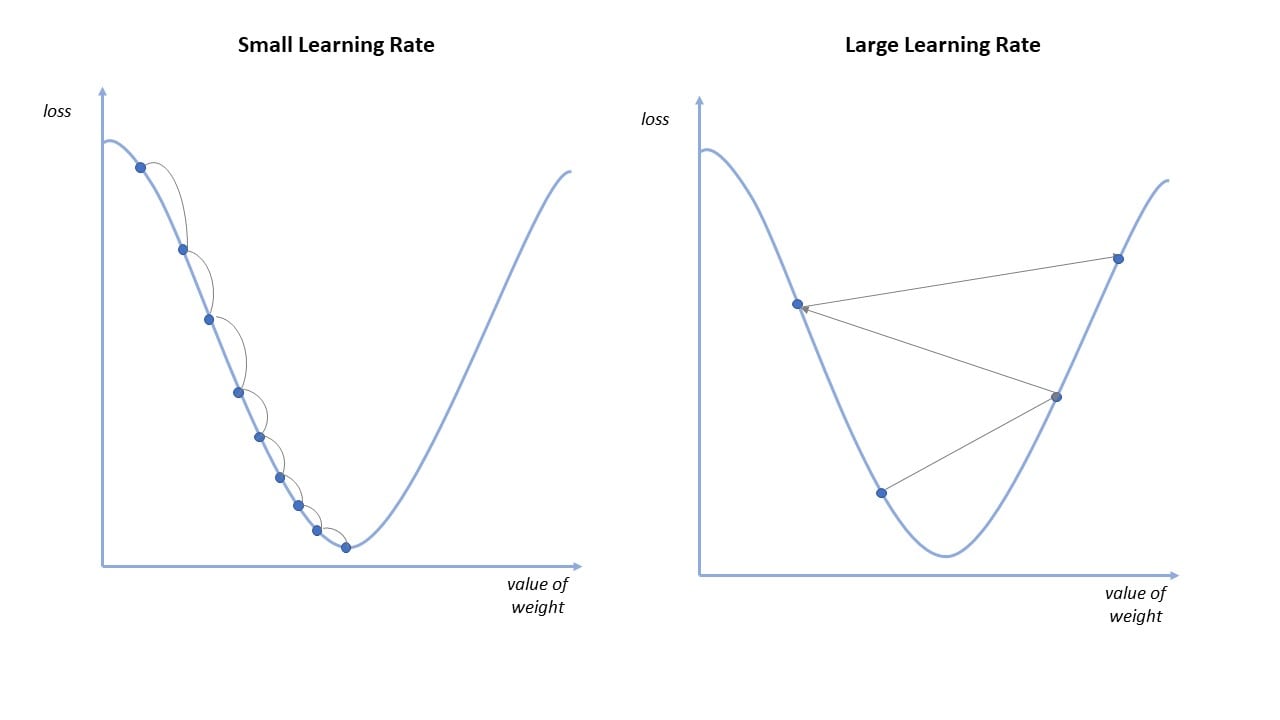
Do traženih vrednosti vektora parametara , za koju *cost* funkcija , ima minimalnu vrednost, dolazi se iterativnim postupkom. Početna, pretpostavljena vrednost parametara , smanjuje se dok ne započne da konvergira ka minimalnoj vrednosti *cost* funkcije. Kako je cilj, da se u što manjem broju iteracija, dođe do ove optimalne vrednosti vektora parametra , a kako gradinet vektor (vektorsko polje), određuje pravace najbrže promene funkcije , primenjuje se optimizacijski algoritam eng. *gradient descen* (Porter i Gujarati, 2009), koji možemo prikazati kao:

*j = 0,1,.....m*

Promene dešavaju se istovremeno za sve vrenosti . Koeficijent učenja povećava intezitet gradinet vektora i tako utiče na veličinu svakog inkrementa promene vrednosti parametra. Male vrednost koeficijenta učenja, povećavaju tačnost pronalaženja minimalne vrednosti *cost* funkcije, jer se ka njoj dolazi u malim koracima, iteracijama, promena parametra .Treba imati u vidu, da je posledica male vrednosti koeficijenta, produženo vreme procesiranja. Veća vrednost ovog koeficijenta, s druge strane, povećava preformanse modela, ali može dovesti do toga da se netačno odredi minimalna vrednost *cost* funkcije (Slika 11).

Slika 11. *Gradinet descent* sa manjim i većim koeficijentom učenja

Male vrednost Veća vrednost



U slučaju jednog uzorka *()*, parcijalni izvod *cost* funkcije po parametru , jednak je:

*=*

Dobijeni izraz nazivamo inkrement promene parametra (eng. *update rule*)(Gareth, et al. 2017). Za uzorak *()* dalje važi,

*j = 0,1,.....m*  (7)

Izraz (7), se naziva Widrow Huff pravilo učenja (Gareth , 2017). Vidimo da je inkrement promene proporcionalan vrednosti residuala . Inkrement promene je tako veći u slučaju većih vrednosti reziduala, odnosno manja sa smanjivanjem vrednosti

Kada primenimo izraz (7) na sve uzorke trening podataka dobijamo,

(8)

Ovakav postupak dobijanja optimalne vrednosti za nazivamo *batch gradient descent* (Gareth, 2017). Kako se pri svakoj iteraciji, razlika residuala računa za sve trening podatke, ovaj postupak je veoma zahtevan (vremenski i resursno).

Alternativa ovom postupku, je *stohastic gradinet descent*, gde se svaki inkrement promene vrednosti parametara modela, bazira na rezidualu jednog trening uzorka (L´eon Bottou, 2010). Tako algoritam prolazi kroz trening podatke samo jednom, a ne onoliko puta koliko ima inkrementa do postizanja minimalne vrednosti :

For i in 1:n

(9)

, za svako *j*

Stohastički pristup je brži, manje resursno zahtevan, ali manje tačan. Preporučuje se u slučajevima velikog broja uzoraka.

Kakoje je konveksna kvadratna funkcija, ona ima jedinstvenu minimalnu vrednost. Tako će postupkom *gradient descent* ,minimalna vrednosti biti jednoznačno određena.

U slučaju linearne regresije, do parametara možemo doći na jednostavniji način od postupka *gradient discent*, direktno bez potrebe za gore opisanim iterativnim postupkom.

Ako vrednosti reziduala napišemo u matričnom obliku:

=

je matrica prediktora (često se u literaturi naziva *design* matricom). Imajući u vidu da za matrice, oblika kolonskog vektora, npr. matricu , važi da je sledi da *cost* funkciju možemo napisati kao:

(10)

je gradient vektor, koji oderđuje pravac tengente na *cost* funkcije *Cost* funkcija, ima minimalnu vrednosti, za vrednost parametra , za koju je izvod funkcije po jednak nuli:

)

=

Kako je je i , sledi:

=

(11)

(podrazumeva se da je matrica invertibilna)

Na ovaj direktan način, u jednom koraku, za prediktivne probleme koji su opisani modelom linerne regresije, možemo doći do vrednosti parametara

**2.2 Metod maksimalne izglednosti, *MaxL***

Pored metode najmanjih kvadrata, do vrednosti parametara možemo doći probalističkom metodom, Maksimalne izglednosti (eng. *Maximum Likelihood*). Ovaj metod se koristi za optimizaciju velikog broja različitih *ML* algoritama.

Izglednost u statistici, je uslovna verovatnoća da poznati slučajni uzorak (), dolazi iz populacije ili nekog nadskupa podataka, čija je distribucija određena parametrom (). Izglednost u opštem obliku možemo napisati kao . S druge strane, verovatnoća da slučajnu uzorci, dolaze iz nadskupa podataka, čiji je parametar distribucije () poznat, označavamo sa . U oba slučaja radi se dakle o uslovnoj verovatnoći.

Ako postoje nezavisni događaji , prema multiplikativnoj teoriji , verovatnoća zajedničkog dešavanja nezavisnih događaja, jednaka je proizvodu njihovih verovatnoća, tako sledi da je,

Ako su vrednosti () poznate, nepoznati parametar distribucije , određujemo primenom metode maksimalne izglednosti (eng. *Maximum Likelihood ).* Logaritmovanjem ovog izraza, dobijamo funkciju izvesnosti (eng. *Likelihood function* – *LF*):

Traženu vrednost , je ona za koju funkcija ima maksimalnu vrednost i dobijamo je iz uslova .

Za primenu metode maksimalne izglednosti, u slučaju linearne regresije date izrazom , uslovnu zajedničku verovatnoću zavisne promenjive možemo napisati u obliku:

Na osnovu pretpostavljene normalne distribucije reziduala sledi da je gustina raspodele (eng. *Cumulative density function – CDF*) reziduala jednaka,

Iz ovog dalje proizilazi,

Vrednosti zavisne promenjive su poznate, dok su vrednosti parametara njene distribucije nepoznati. Za određeivanje ovih parametara, primenjujemo metod maksimalne izglednosti, za koju funkcija izglednosti *L* je jedanka

Vrednost za koju ima maksimalnu vrednost, ekvivalentan je uslovu dobijanju minimalne vrednosti izraza . Vidimo da je ovaj uslov identičan uslovu postizanjem minimallne vrednosti *cost* funkcije metodom najmanjih kvadrata jednačina (10 ). Iz ovog sledi, da je u slučaju linearne regresije metoda najmanjih kvadrata, specijalni slučaj metode maksimalne izglednosti i da oba pristupa daju istu vrednost parametrar , dat jednačinom (11) .

Dobijene vrednosti , primenom *OLS* i *MaxL* metoda, imaju svojstva slučajne promenjive normalne distribuciju, što je posledica pretpostavke da reziduali imaju normalnu distribuciju i da je . Tako na osnovu *central limit* teoreme , očekivana vrednost parametra jednaka je parametru , *PRF* modela, tj. . Tačnost ove predikcije merimo standardnom greškom (SE), te možemo na osnovu njene vrednosti definisati i interval u kojem sa velikom sigurnošću možemo očekivati da se nalazi vrednost (eng. *Interval estimate).*

. Najčešće je to vrednost 95%, verovatnoća da se nalzi u intevalu .

**2.3 Regularizacija regresionih modela**

Određivanjem regresionih parametara , metodom *OLS*, često dovodi do pojave *overfitinga* modela, što uzrokuje nizak nivo generalizacije. U slučajevima velikog broja prediktora, a specijalno kada je *m* , regresioni modeli nisu dovoljno stabilni i tačnost predikcije je niska. Iz tog razloga se pristupa regularizaciji, kojom se vrednost regresionih parametara za prediktore, koji minimalno utiču na zavisno promenjivu, smanjuju do vrednosti približno ili jednakoj nuli. Tako se ovim postupkom, ‘sankcioniše’ regresioni model, usled velike kompleksnosti ( velikog broja prediktora u modelu).

Najčešće primenjivane metode regularizacije regresionih modela su:

1*. Ridge* regresioni model (Hoerl i Kennard, 1970), primenjuje L2 regularizaciju. Cilj je da se za prediktore koji manje utiču na vrednost zavisne promenjive, njihovi regresioni koeficijenti svedu na vresnosti približno jednake nuli. Regresioni parametri ovog modela dobijaju se iz uslova postizanja minimalne vrednosti *cost* funkcije sledećeg oblika:

+

Za , *Ridge* regularizacija daje iste vrednosti regresionih parametara, kao one koje dobijamo metodom najmanjih kvadrata (*OLS*). Kada je , jedini način da se vrednost *cost* funkcije smanji, je da se smanje vrednosti regresionih parametara.

2. *Lasoo* regresija (Tibshirani, 1996), primnjuje L1 formu regularizacije. Parametri ovog modela određuju se iz sličnog uslova kao kod *Ridge* regresije, sa tom razlikom da tzv. *penalty term* ( drugi sabirak u izrazu), nije jednak zbiru kvadrata regresionih parametara, već zbiru njihovih apsolutnih vrednosti,

+

Osnovna razlika između ove dve metode regularizacije (*Ridge i Lasoo*) je u tome, što u slučaju *Ridge* regresije, vrednosti regresionih parametara ne mogu imati vrednost nula, tako da svi prediktori ostaju u modelu. U slučaju *Lasoo* regresije, parametri mogu imati vrednost nula, pa tako u finalnom modelu, prediktori sa najmanje uticaja na vrednost zavisne promenjive se iz modela izostavljaju.

Parametar je *shrinkige* koeficijent, koji može imati vrednosti u rangu . Do optimalne vrednosti dolazi se najčešće postupkom *cross* validacije (CV). Ukoliko je vrednost ovog parametra velika, može doći do značajnog pojednostavljenja modela i pojave *undrefiting-*a.

3. *Elastic Net* regresija, je kombinacija prethodna dva modela.

**3. *ML* KLSIFIKACIONI MODELI**

U desertaciji su analizirani najčešće primenjivani parametarski i neparametarski *ML* klasifikatori. Prema načinu kako se određuje vrednost target kategoričke promenjive, odnosno uslovne verovatnoće da ispitivani uzorak pripada jednoj od klasa zavisne promenjive, ovi algoritmi mogu biti diskriminativni i generativni.

Diskriminativni (eng. *Discriminative*) klasifikacioni algoritmi, su oni kod kojih se na direktan način dolazi do uslovne verovatnoće , verovatnoće da uzorak , pripada jednoj od klasa zavisne promenjive. Ne pretpostavlja se distribucija verovatnoće promenjive , za različite klase , Kod ovih klasifikacionih algoritama, iterativnim postupkom se pretpostavljaju različite granične vrednosti (eng. *threshold value*), koje određuju položaj hiperavni u prostoru , koja na najbolji način razdvaja uzorke različitih klasa. Optimalna vrednost je ona, za koju je broj misklasifikovanih uzoraka najmanji. Ispitivani uzorak se tako dodeljuje klasi, prema položaju u odnosu na graničnu hiperravan.

U primeru zavisne kategoričke promenjive sa dve klase, i jednim prediktorom, ovaj se uslov može napisati na sledeći način:

Slika 12. Optimalni položaj granične prave, određen je sa , za koju je broj misklasifikovanih uzoraka najmanji. Za vrednost , broj pogrešno klasifikovanih uzoraka je 3, za je 3, a za je 2.

.

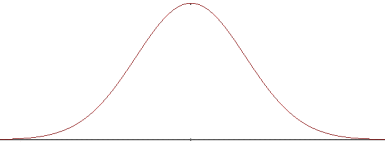
U slučaju generativnih (eng. *Generative*) algoritama, uzorak se dodeljuje klasi, za koju je zajednička verovatnoća najveća. Kako su zavisne, na osnovu multiplikativne teorije verovatnoće sledi da je,

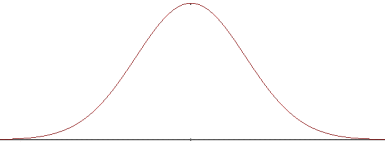
U slučaju dve klase imamo da je:

U primeru sa slike 13, kako je za novi uzorak , , a sledi da će uzorak biti klasifikovan kao

Slika 13. Normalna distribucija uzoraka klase 0 i 1. Novi uzorak , dodeljuje se klasi za koju je najveće









Do verovatnoće možemo doći i primenom *Bayes* teoreme (T. Bayes, 1702-1761) na indirektan način,

Sledi da će takođe biti dodeljen klasi, za koju ima najveću vrednost,



Tabela 4. Diskriminativni i generativni *ML* modeli



|  |  |  |
| --- | --- | --- |
|  | Diskriminativni model | Generativni model |
| Pristup | Direktna procena | Procena potom sledi |
| Učenjem *ML* modela dobijamo | Graničnu liniju(hiperavan) | Distribuciju verovatnoće podataka po klasama |
| Prikaz | https://miro.medium.com/max/1335/1*vUgvyc3Y__v2TL_lsRTQhw.png | https://miro.medium.com/max/1335/1*vUgvyc3Y__v2TL_lsRTQhw.png |
| *ML* algoritmi | Logistička regresija  Stablo odlučivanja  Random Forest  SVM | Naive Bayes  KNN |

**3.1 LOGISTIČKA REGRESIJA, *LogR***

Logistička regresija je parametarski klasifikacioni diskriminativni metod, nadgledanog učenja. Spada u grupu osnovnih klasifikacionih algoritama. Primenjuje se u slučajevima, kada je potrebno predvideti vrednost zavisne kategoričke promenjive, koja može pripadati jednoj od dve ili više kategorija ili klasa. *LogR* kao probabilistički klasifikator, daje uslovnu raspodelu verovotnoče oblika . Kako se u našem primeru (studiji slučaja), radi o binarno zavisno promenjivoj, govorimo o primeni logističke regresije, u rešavanju binarno klasifikacionog problema. U našem primeru, kategorička zavisna promenjiva može pripadati jednoj od dve klase i to:

Model uslovne raspodele verovatnoće, odnosno prediktivni model, za uzorke u stečaju, , teorijski je moguće definisati kao linearnu funkciju prediktora, na sličan način kao u *LR*,

(1)

Model linearne regresije (1), daje vrednosti autputa u granicama . Kako verovatnoća ima pozitvne vrednost u rasponu , očigledno je da se linerna regresiona funkcija, ne može primeniti kao model raspodele verovatnoće. Pored ovog očiglednog razloga, problem sa linearnom regresijom je i u tome, da je ona podložna uticaju ekstremnih vrednosti (eng. *outliers*). Pod pretpostavkom, da je granična vrednost verovatnoće 0.5, tako de je za , sa slike 14, možemo videti kako pojava ekstremnih vrednosti utiče na promenu modela i njegovu tačnost .

Slika 14. Model linearne regresije, za binarni klasifikacioni problem. Regresioni model , kada je uzorak pripada klasi 0, a za vrednosti *a*, klasi 1Broj misklasifikovanih uzoraka je nula. Ukoliko postoji *outlier* ( ), regresioni model postaje , koji klasifikuje uzorke kao , za vrednosti . Broj misklasifikovanih uzoraka je u tom slučaju 2.

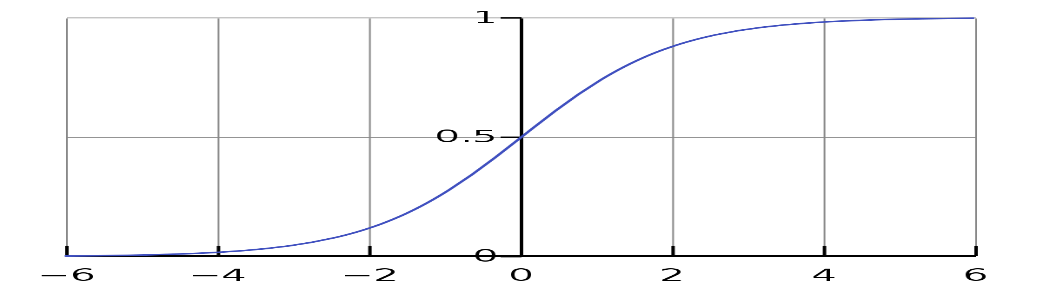
p 1

0.5

0 a b x

Kako bi ogrančili vrednosti modela linearne regresije, primenićemo logističku funkcijom , koje daje vrednosti u očekivanim granicma (James et al, 2013).

Slika 15. Logistička funkcija

x

Posle ove transformacije, promenom oblika funkcije ), u dobijamo,

(2)

Sledi da je verovatnoća da , pripada klasi 0, jednaka,

*=* (3)

Izrazi (2) i (3), napisani u jedinstvenom obliku, predstavljaju *Bernoulli* diskretnu distribuciju verovatnoće, zavisne binarne kategoričke promenjive ,

(4)

Daljom transformacijom izraza (2) dobijamo:

(5)

Količnik , predstavlja izglednost događaja. Odnosno, koliko je puta jedan događaj više izgledan od drugog. Logaritmovanjem izraza (5) sledi,

(6)

Leva stran izraza (6) je *logit* funkciju od , koja je sad linearno zavisna od . Na ovaj način se problem predikcije, u slučaju klasifikacionih problema, ponovo svodi na model identičnog oblika kao u slučaju linearne regresije (slika 16). Ovaj model tumačimo na način, da se povećanjem vrednosti prediktora za jedničnu vrednost (pretpostavljajući da su ostali prediktori konstantni), logaritam izglednosti događaja da je , povećava se približno za .

Slika 16. Logit funkcija, linerano zavisnoa od predikotora ( model sa jednim prediktorom)

*u*

Sada slično kao u linernoj regresiji, iterativnim postupkom, primenom odgovarajućeg optimizacijskog algoritama, dolazimo do optimalnih vrednosti parametara , a time i optimalnog položaja granične hiperavni,koja na najbolji način razdvaja uzorke različitih klasa.

Optimalni pravac hiperravni u slučaju linearne regresije, određen je primenom *OLS* optimizacijskog algoritma, tako da suma kvadrata reziduala bude minimalna. U slučaju logističke regresije, ovaj algoritam se ne može koristiti, jer za svaku vrednost parametra , reziduali imaju beskonačnu vrednost (od do projekcije na regresionu pravu, slika 16). Zato se u slučaju logističke regresije primenjuje algoritam Maksimalne izglednosti (Hastie et al. 2008) Imajući u vidu *Bernoulli,* binarnu rasraspodelu verovatnoća zavisne promenjive, funkcija izglednosti ima oblik:

(7)

Lagaritmovanjemovog izraza dobijamo:

(8)

Što je veća vrednost , to je bolja, tačnija predikcija, zbog čega se i postavlja uslov postizanja njene maksimalne vrednosti, koju dobijamo iz uslova  *,*

U slučaju slučaj jednog uzorka imamo:

\*

( = (9)

*\** Prvi izvod proizvoljne *sigmoid* funkcije *g(x)* jednak:

Primećujemo da je *update rule* (9), ima sličan oblik kao u slučaju linearne regresije. Ovo proizilazi iz činjenice da linerana i logistička regresija , pripadaju istoj familiji algoritama Generalnih linearnih modela – *GLM* (Hastie et al. 2008) .

Za razliku od linerne regresije gde smo tražili minimalnu vrednost *cost* funkciije (OLS) metodom *gradient descent,* ovde tražim maksimalnu vrednost funkcije Maksimalne izglednosti, postupkom *gradient ascent* Tako vrednost parametra , svakom iteracijom raste, dok ne počne da konvergira ka vrednosti , za koju , ima maksimalnu vrednost:

*j= (0,1,......m )* (10)

Odnosno,

(11)

Na osnovu dobijenih optimalnih vrednosti parametra , iz jednačine (2), dobijamo konačno verovatnoću *p(x)* , da je .

Onovne pretpostavke modela logističke regresije su da je logit funkcija linearno zavisna od prediktora, da su prediktori međusobno nezavisni (da ne postoji multikolinearnost), da u podacime nema ekstremnih vrednosti. Kada su ovi preduslovi ispunjeni, ovim modelom se postiže zavidan tačnost predikcije u binarno klasifikacionim problemima (Porter i Gujarati, 2009).

Na sličan način kao kod linearne regresije, kako bi izbegla mogućnost overfiting, kao i nestabilnost modela usled manjeg broja uzoraka, u odnosu na broj prediktora, primenjuje se postupak regularizacije ( Redge, Laasso ili Elastic Net).

3.1.1 Generalni linerni modeli, *GLM*

Modeli linerane i logističke regresije su specijalni slučajevi Generalnih linearnih modela mašinskog učenja - *GLM* (*Generalized linear models*), čiji su autori Nelder and Wedderburn (1972). Primenom *GLM* , generalizujemo model kojim se zavisna promenjiva prikazuje kao linerna funkcija prediktora :

(1)

Tako linerni prediktvni model dat jednačinom (1), nije isključivo ograničen na slučajeve linerne i logističke regredije, već se može primeniti uvek, kada raspodela verovatnoće zavisne promenjive, pripada nekoj od eksponencijalnih distribucija (P.McCullagh, J.A. Nelder, 1983)*.* Transformacijom očekivanih vrednosti zavisne promenjive, primenom tzv. *link* funkcije, dobijamo prediktivni model koji je linearna funkcija prediktora.

Familija eksponencijalnih distribucija se možemo prikazati u opštem obliku (P.McCullagh, J.A. Nelder, 1983),

*Pr*= (2)

funkcija

*T*

, prirodni parameter distribucije

Svaka distribucija koja pripada eksponencijalnoj familiji, je ona koja može biti prikazana izrazom (2) .

Kako smo prethodno naveli i Normalna distribucija, pripadaju familiji eksponencijalnih distribucija.

Kako distribucije raspodele verovatnoće ima oblik:

Transformacijom, ovaj izraz možemo svesti na eksponencijalni oblik :

exp (

(

Poređenjem ovog izraza, sa izrazom koji definiše eksponencijalnu familiju distribucija (2), dobijamo da je:

b(y) = 1

T(y) = y

= , odnosno da je

a() ===

Dobijene vrednosti koeficijenata *b,T* i *a*, su karakteristične za distribuciju, dok vrednost prirodnog parametra , odgovara konkretnoj distribuciju iz familije (P.McCullagh, J.A. Nelder, 1983).

Normalna distribucija zavisne promenjive , data je sledećim izrazom :

Transformacijom ovog izraza takođe možemo pokazati da noramalna distribucija pripada eksponencijalnoj familiji. Odnosno, gornji izraz jednak je:

Sledi da je:

*T(y) y*

(srednja vrednosti)

a()

Dobijene vrednosti koeficijenata *b,T* i *a*, su karakteristične za Normalnu distribuciju, dok je konkretna distribucija koja pripada ovoj familiji, jednoznačno određena svojom srednjom vrednošću, koja je jednaka prirodnom parametru (P.McCullagh, J.A. Nelder, 1983).

Pretpostavke koje su potrebno da budu ispunjene u slučaju primene prediktivnog GLM modela su sledeće (P.McCullagh, J.A. Nelder, 1983):

1. , za date vrednosti prediktora i parametra , zavisno promenjiva , ima eksponencijalnu distribuciju, određenu parametrom .
2. Linearni prediktivni model *f(x*; ), daje očekivanu vrednost zavisne promenjive, odnosno:

1. Prirodni parameter i prediktori su linearno zavisni:

Primenom *GLM*, u slučaju predikcije vrednosti kvantitativne zavisne promenjive, normalne raspodele verovatnoće, za koju važi , zaključujemo da je na osnovu pretpostavke (2):

Kako normalna dostribucija pripada eksponencijalnoj familiji , gde je , a kako iz pretpostavke (3), sledi da je , vidimo da smo primenom GLM algoritma, dobili model identičan modelu linearen regresije:

Slično primenom *GLM*, za slučaj klasifikacionih problema, gde je očekivana vrednost jednaka verovatnoći pozitivnog ishoda slučajnog događaja, na osnovu druge pretpostavke dobijamo da je:

Kako distribucija pripada eksponencijanoj familiji, za nju važi da je =, odnosno da je Na osnovu pretpostavke (3), , sledi da je:

(17)

Primenom *GLM* , za slučaj binarnog klasifikacionog problema, dobili smo isti prediktivni model kao u slučaju logističke regresije. Sledi da izbor *Sigmoid* funkcije u logističkoj regresiji, kojom smo osigurali da vrednost predikcije budu u rangu 0 do 1, nije bio proizvoljan, već rezultat činjenice da logistička regresija spada u *GLM*.

Transformacije očekivanih vrednosti *GLM* modela, *link* funkcijom, često pišemo u obliku g(), gde je očekivana vrednost, svake od eksponencijalnih distribucija (P.McCullagh, J.A. Nelder, 1983). U Tabeli 4, vidimo prikaz link funkcija za najznačajnije eksponencijalne distribucije.

Tabela 4. Link funkcije, za najčešće primenjivane eksponencijalne distribucije

|  |  |
| --- | --- |
| Ditribucija | Link funkcija  , |
| *Normalna* |  |
|  |  |
| *Poisson* |  |
| *Exponential* |  |
| *Gamma* |  |

,

Opšti oblik *GLM* modela možemo napisati kao:

Gde je , očekivana vrednost odgovarajuće distribucije

Jediničnim povećanjem funkcijom , transformisana očekivana vrednost autputa, menja se za vrednost parametra .

Parametri inearnog *GLM* modela se određuju metodom Maksimalne izgednosti (*eng*. maximum liklihood), *gradient descent* određuju se vrednosti parametara linearng modela :

Za testni uzorak , na osnovu poznatih vrednosti i , dobijamo vrednost *link* funkcije . Iverzijom, , se dobijaju očekivane vrednosti zavisne promenjive.

Primenom *GLM*, generalizujemo linearni model, tako da se može primeniti u slučajevima kada zavisna promenjiva ima neku od eksponencijalnih distribucija, primenom odgovarajuće *link* funkcije, kojom se očekivana vrednst zavisne promenjive transformiše.

3.1.2. *Gaussian* diskriminantna analiza *,GDA*

*Gaussian* diskriminantna analiza, *GDA* je generativni klasifikacioni algoritam, koji se primenjije u klasifikacionim problemima, kada je zavisna kategorička promenjiva ima više od dve klase. Za razliku od logističke regresije, gde smo direktno modelovali uslovnu raspodelu verovatnoće da zavisna promenjiva pripada klasi , u slučaju *GDA*, do ove verovatnoće dolazimo na indirektan način. Kako se radi o generativnom algoritmu, modeluje se distribucija pradiktora, za različite klase zavisne promenjive, , pa primenom *Bajesove* teoreme dolazimo do tražene ,

(1)

, klasa proemenjive

Do vrednosti prethodne (eng. *prior*) verovatnoće koja pretstavlja verovatnoću da slučajni uzorak pripada klasi *k*, dolazimo na jednostavan način iz trening podataka,

broj uzorka klase k

Kako bi došli do distribucije verovatnoće pretpostavljamo multivarijantnu normalnu distribuciju prediktora (), za svaku klasu zavisne promenjive. Tako se podrazumeva da svaki prediktor pojedinačno ima normalnu raspodelu, dok između parova prediktora postoji neki stepen korelacije. Parmateri *Gaussian* multivarijantne distribucije, su vektor srednjih vredsnoti prediktora i kovarijantna matrica. Dijagonalni elementi ove matrice predstavljaju varijansu prediktora, a van diagonalni elementi, kovarijansu između prediktora.

*GDA* algoritam postoji u dva oblika, za koje važe različite pretpostavke za kovarijantnu matricu prediktora Kada podrazumevamo da je ova matrica različita za uzorke različitih klasa (manje restiriktivna pretpostavka), govorimo o *Quadratic dicriminante analysis* - *QDA* ( Gareth et al, 2014). Funcija raspodele verovatnoće predikotora u slučaju normalne multivarijantne distribucije tada ima oblik:

(11)

Kada pretpostavimo da je kovariantna matrica prediktora ista, za uzorke različitih klasa, govorimo o *Linerna discriminatne analysis* – *LDA* ( Gaber et al, 2017) . Kako u našem slučaju zavisna promenjiva može biti klase 0 ,za kompanije koje nisu u stečaju i 1 za kompanije koje jesu, a na osnovu pretpostavke o istim kovarijantnim matricama koja važi za *LDA*, imamo sledeća dva oblika multivarijantne distribucije:

,

Tako će u slučaju *LDA*, novi uzorak biti dodeljen klasi , za koju je broilac iz *Bajesove* formule (jedančina 1), odnosno proizvod , najveći. Vrednost imenioca istog količanika je je konstantna. T

(2)

Kako nezavisi od *k*, zameneićemo ovaj količnik konstantom *C*,

Logaritmovanja dobijamo,

Kako je logC konstanta, možemo je izostaviti,

Iz dalje sledi,

nezavisi od *k*,

(3)

Ovaj izraz nazivamo linernom diskrimintnom, funkcijom (Gareth et al, 2014). Testni ili novi uzorak, pripadaće klasi za koju je ima veću vrednost.

U sličaju *QDA* algoritma, uzorak će biti dodeljen klasi za koju *quadratic dicriminant function* *QDF* , ima maksimalnu vrednost . Po analogiji sa *LDA,* imajući u vidu različite kovarijantne matrice uzoraka različitih klasa, tražimo maksimalnu vrednost izraza,

C =

Izostavljanjem konstante C i logaritmovanjem, prethodni izraz postaje,

Odnosno:

+ (4)

Testni uzorak biće dodeljen klasi za koju *QDF* im najveću vrednost (Gareth et al, 2014).

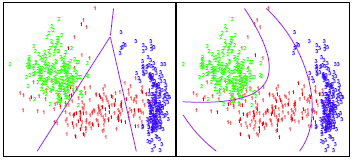
*QDA* ili *Quadratic discriminant analysis* ,kao podgrupa GDA , smatra se fleksibilnijom metodom u odnosu na *LDA*, iz razloga da pretpostavka jednakih kovarijantnih matrica prediktora za različite klase ne važi. Granice između klasa koje se generišu ovim algoritmom nisu linearna funkcije prediktora (kao u slučaju *LDA*) (Slika 17 ) .

Linije razgraničenja klasa ( npr. klase *k* i klase *l*) u slučaju *LDA,* za različite vrednosti prediktora ispunajva uslov:

Kada se radi o *QDA*, linije raygraničenj klase *k* i *l* ispunjava uslov:

Ove granične linije se još i nazivaju Bayes granicama odlučivanja (James et al., 2013)

Slika 17. Granične linije za *LDA* i *QDA*[[10]](#footnote-11)



*LDA* *QDA*

Kad je broj uzoraka ograničen, bolja tačnost predviđanja postiže primenom *LDA*, jer je za ‘učenje’ ovog modela *ML*, potrebna manja količina podataka, u odnosu na *QDA*. Neophodno je takođe, da broj prediktora bude značajno manji od broja uzoraka. Ukoliko taj uslov nije ispunjen, drastičano se smanjuje tačnost predikcije ovim modelima. Iskustveno pravilo je da se *LDA* i *QDA* ,primenjuju u slučajevima kada je ispunjen uslov [[11]](#footnote-12).

U poređenju sa logitičkom regresijom, oba modela su stabilnija, u slučajevima kada je moguća separacija uzoraka različitih klasa. Ovi generativni algoritmi, se koriste uvek u klasifikacionim problemima, kada je broj klasa zavisne kategoričke promenjive veći od dva (eng. *multiclass problem*).

Pored rešavanja klasifikacionih problema, *LDA* se koristi u cilju smanjivanja dimenzionalnosti prediktivnog modela. Uzorci prediktora, kao tačke u originalnom *m* dimenzionalnom prostoru ( ), transformišu se u novi tzv. *LD* prostor, određenom kordinatnim osama , (eng. *linear desriptors*- *LD*), tako da se maksimalna separacija uzoraka različitih klasa, poststiže njihovom projekcijom na pravac , a najmanja na (Slika 18)(Gaber et al, 2017). Tako se končan broj *LD* , na kojima će bit baziran prediktivni model, ograniičava samo na one, koji određuju pravce najveće separacije podataka različitih klasa (pravce minimalne disperzije uzoraka unutar klase i maksimalne između klasa), dok se pravci minimalne separacije mogu zanemariti.

Slika 18. Uzorci u originalnim prostoru, odrđeni su koordinatama prediktroa . Novi *LD* proostor određen je osama maksimalne i minimalne separacije uzoraka. Prediktivni model, tako može bit predstavljen kao , čime se smanjena dimenzionalnost prediktivnog modela za jednu varijablu[[12]](#footnote-13).

Chart, scatter chart

Description automatically generated

Utvrđivanje ovih novih pravaca, odnosno koordintanih osa *LD*, određuje se u tri koraka.

Prvo se računa separablinost između uzoraka različitih klasa (eng. *between class variance* - Sb ), mereći razdaljinu njihovih srednjih vrednosti, od ukupne srednje vrednosti svih uzoraka

Sb =T

srednja vrednosti uzoraka klase *i*

, srednja vrednosti svih uzoraka

*k -* broj klasa

– broj uzoraka klase *i*

Sb - matrica

U drugom koraku, računamo variansu uzoraka svake klase (eng. *within class variance*, Sw)

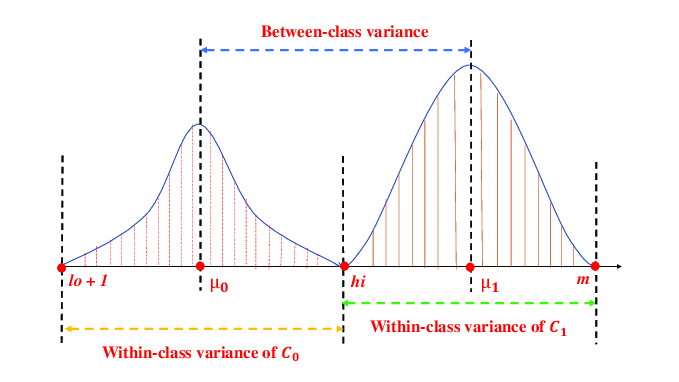
Sw = T

srednja vrednost uzoraka klase *i*

Sw - matrica

Projekcijom uzoraka na nove *LD* pravce, cilj je postići maksimalnu vrednost Sb i minimalnu Sw.

Slika 19. Sw i Sb, za uzorke koje pripadaju različiitim klasama i [[13]](#footnote-14)



Ako sa *V* označimo matricu transformacije, kojiom se vektori uzoraka projektuju u novi u *LD* prostor, onda je projekcija vektora uzoraka i vektora srednjih jednkaka i . Iz ovog dalje sledi da , posle projekcije u novi LD prostor postaje:

Sledi da *Sw*, posle projekcija vektora uzoraka i srednjih vrednosti , na *LD* postaje:

=

Po istoj analogiji dobijamo da je:

Finalno, optimalan *LD* pravac, određen transformacijkom matricom *V ,*dobijamo iz uslova koji se naziva *Fisher* kriterijum:

= (5)

Iz uslova da je izvod sledi:

(6)

Jednačina (6) predstavlja tzv. problem sopstvenih vrednosti (eng. *eigenvalue* ) (Gaber et al, 2017), čije su rešenja sopstvene vrednosti i sopstveni vektori (eng. *eigenvectors*), matrice (Gaber, 2017). Iz (17) dalje sledi:

Kako kvadratna matrica ), nema inverznu matricu (eng. *not invertable*), sledi da je njena deterimnanta jednaka nuli. Iz ovog uslova dobijamo sopstvene vrednosti jediničnih vektora:

Sa poznatim vrednostima iz (6) dobijamo sopestvene vektore

Sopstveni vektori, određuju pravce novih osa u novom *LDA* prostoru. Pravac određen je sopstvenim vektorom, čija je odgovarajuća sopstvena vrednost najveća, Preostali sopsteveni vektori se zanemaruju, odnosno samo sopsteveni vektori , se koriste pri određivanju novog *LDA* prostora. Na ovaj način smo početni prostor transformiasli u novi *LDA*, manje dimenzionalnosti , gde je k<m. Sad se prediktivni model može predstaviti kao funkcija linekarnih deskriptora

**3.2 NAIVE BAYES, *NB***

NB je generativni *probabilaistic* klasifikacioni algoritam, koji zbog svoje jednostavnosti ima široku primenu. Njegova relativna jednostiavnost, proističe iz pretpostavke, da su prediktori međusobno nezavisni. Iz ovoga dalje sledi, da je uslovna verovatnoća da uzorak pripada klasi *k*, na osnovu multiplikativne teorije verovatnoće, jednaka proizvodu uslovnih verovatnoća prediktora tog uzorka (Shai , 2014),

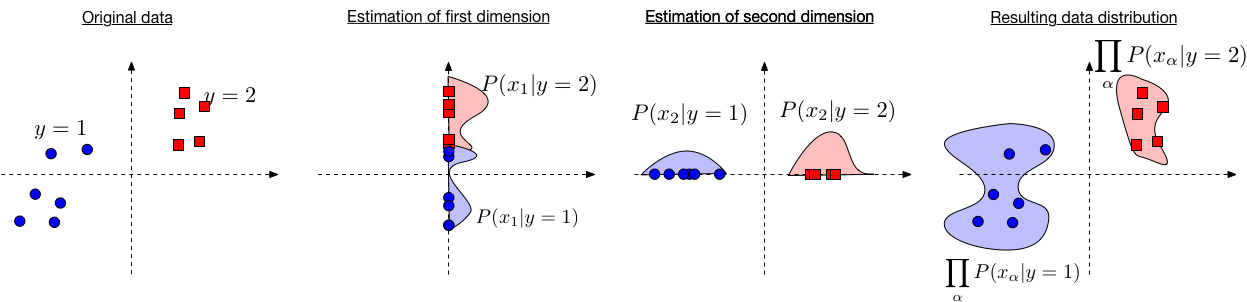
(1)

= () – vektor prediktora

U slučaju dva prediktora*,* njihovazajednička distribucija verovatnoće, jedanka je proizvodu pojedinačnih. Primenom izraza (1) dobijamo,

*k -* klasa zavisne promenjive *y*

Slika 20. Zajdnička distribucija prediktora , jednaka je proizvodu njihovih pojedinačnih distribucija(Alboukadel, 2017*)*



*NB* kao generativni algoritam, baziran je na *Bajesian* teoremi, koja u slučaju *n* uzoraka i *m* prediktora , se može napisati :

(2)

Borojilac jednačine (2), predstavja zajedničku verovarnoću međusobno zavisnih događaja, ,

Na osnovu pretpostavke o nezavisnosti prediktora, sledi de je:

= (3)

Tako dobijamo da je zajednička verovatnoća jednaka :

(4)

Tako na osnovu (4) , jednačina (2) postaje,

(5)

Novi uzorak se tako dodeljuje klasi, za koju će imati maksimalnu vrednost. Ovaj uslov je ispunjen za maksimalnu vrednost brojioca jednačine (5), koji se može napisati u sledećem obliku, :

(

Do vrednosti ), verovatnoće pojavljivanja uzoraka klase *k*, dolazimo jednostavno,:

1 < k < ukupan broj klasa

Vrednost odnosno distribuciju verovoatnoća prediktora, za različite klase zavisne promenjive, određujemo u zavisnosti od tipa prediktora, pa prema tome razlikujemo tri *NB* modela (Advait, 2020):

*Gaussina Naive Bayes*, u slučaju kada su prediktori kvantitativne promenjive, pretpostavlja se njihova normalna distribucija:

srednja vrednost prediktora klase *k*

varijansa prediktora klase *k*

*Bernoulli Naive bayes,* prediktori su binarne kategoričke promenjiva, njena diskretna raspodela verovatnoće jednaka je,

Ovaj model, se najčešće primenjuje, pri klasifikaciji dokumenata prema sadržaju. Binarna vrednost prediktora , označava da li u dokumnetu postoji *i*-tareč, sa liste od m elemenata, dok je verovatnoća da klasa dokumenata *k* ,sadrži datu reč. Unapred definisna lista reči, od m elemenata (eng. *bag of words*), sastavljena je tako, da najbolje karakteriše, određuje, različite vrste dokumenata (Singh et al, 2019)

Dokumnet će tako biti dodeljen klasi, za koju je zajednička verovatnoća pojavljivanja reči iz . *bag of words* u dokumentu najveća:

*Multinominalni Naive Bayes,* za razliku od *Bernoulli* NB, vrednost kategoričke promnjive nije binarna, već prediktori pokazuju učestalost pojavljivanja nezavisnog događaja (reči u dokumentu). Ovaj algoritam se takođe najčešće primenjuje u problemima klasifikacije dokumenata. Tako svaki uzorak, predstavlja histogram, u kojem vrednost npr. , predstavlja frekvenciju pojavljivanja događaja *i* (*i*-te reči sa liste od *m* ) u dokumentu:

Dokumnet se se dodeljuje klasi za koju je:

Kada su prediktori kategoričke promenjive, može se pojaviti slučaj, ‘*Zero frequency problem*’ (Alboukadel Kassambara, 2017*),* kada za neku klasu zavisne promenjive, ne postoji niti jedan uzorak sa prediktorom određene klase.

. Sledi da je , što je malo verovatan slučaj u populaciji, iz koje su ovi podaci uzorkovani .

|  |  |  |
| --- | --- | --- |
|  | *y* =1 | *y*=0 |
|  | 5 | 4 |
|  | 0 | 1 |

U ovakvim slučajevima, najjednostavnije je povećati broj pojavljivanja klasa prediktora i klasa zavisne promenjive za jedan (Tabela 6).

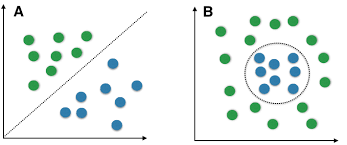
Tabela 6.

|  |  |  |
| --- | --- | --- |
|  | y=1 | y=0 |
|  | 5 (+1) | 4 (+1) |
|  | 0 (+1) | 1 (+1) |

*NB* algoritam, daje bolje rezultate u slučaju kategoričkih prediktora u odnosu na numeričke. Razlog tome je što pretpostavka u normalnoj distribuciji numeričkih prediktora, najčešće nije ispunjena (Advait, 2020). *NB* algoritam ne treba primenjivati, u slučajevima kada uzorci različitih klasa nisu jasno razdvojeni. Odnosno, ako su centri disribucije srednja vrednost) i disperzija, približno iste za uzorke različitih klasa. Oblik granica razdvajanja, koje se kreiraju primenom *NB*, zavise od lokacije centra njihove distribucije i disperzije (Advait, 2020). Ako uzorci različitih klasa, imaju različite srednje vrednosti, a istu disperziju, granica razdvajanja je prava (Slika 21a). Kada su srednje vrednosti iste, a varijanse različite, granična linija je kružnica ili elipsa (Slika 21b). Granična linija je parabola kada su razlikuju i srednje vrednosti i disperzije (Slika 21c).

Slika 21. Granice razdvajanja, *NB* modela, u slučaju dva prediktora, u binarnim klasifikacionim problemima

(a) (b) (c)



**3.3 K - najbližih suseda , *K-NN***

*K - najbližih suseda (*eng*. K-nearest neighbor*), je diskriminativni nadgledani *ML* algoritam, koji se može primenjivati u slučaju klasifikacionih i regresionih problema. Spada u grupu neparametarskih (eng. *non parametric)* modela, tako da ne pretpostavlja oblik funkcije zavisnosti prediktora i target promenjive,

Kada se *K-NN* primenjuje u slučaju klasifikacionih problema, odlučivanje je bazirano na sličnosti uzoraka (Alboukadel, 2017). Algoritam bira *K* broj najsličnijih uzoraka novom uzorku, za koji želimo predvideti kojoj klasi zavisne promenjive pripada. Većinska klasa, kojoj *K* broj najsličnijih uzoraka pripada, jeste klasa novog uzorka. Ovaj način klasifikacije naziva se većinsko odlučivanje (eng. *majority voting* ) (Harrington,2106).

U slučaju regresionih problemima, predviđena vrednost zavisne promenjive novog uzorka, jednaka srednjoj vrednosti zavisne promenjive, *K* najsličnijih uzoraka (Alboukadel, 2017).

Kako se uzorak, može predstaviti tačkom u *m* dimenzionalnom prostoru ( *m* je broj prediktora), sličnost uzoraka se može odrediti na osnovu njihovih međrastojanja. Tako se geometrijski najbliži uzorci, smatraju najsličnijim.

Slika 22. Novi uzorak , se dodeljuje klasi kojoj većina od K=5, suseda pripada- klasi

Diagram, schematic

Description automatically generated

Izbor *K* najbližih suseda, a time i najsličnijih, najčešće se određuje primenom *Euclidean* formule (Harrington, 2016). Ako *n* uzoraka označimo sa (),(),...,() , gde je svaki uzorak određen vektorom u *m* dimenzionalnom prostoru , tada je *Euclidean* rastojanje *d*, prava linija koja spaja tačke :

(

*K-NN* je u slučaju primene euklidske razdaljine, vrlo osetljiv na postojanje ekstremnih vrednosti, specijalno u slučajevima male vrednosti *K.* Takođe, kada se radi o većoj količini podataka, sa velikim brojem prediktora, računanje razdaljine na ovaj način je resursno izuzetno zahtevno.

Alternativa primeni *Euclidean,* je *Manhhatan* formula za izračunavanje razdaljine (Altman, 1992), koja ima sledeći oblik*:*

(

Ovakav pristup daje najkraću razdaljinu između dve tačke, u pravcima koji su paralelni koordinatnim osama prostora, u kojem se tačke nalaze (Slika 23).

Slika 23. Primer *Manhhatan* i *Euclidean* razdaljine tačaka A i B u dvodimenzionalnom prostoru

B Manhhatan

Euclidean

Kada je broj prediktora veliki, kao i u slučaju postojanja ekstremnih vrednosti, *K-NN* je stabilniji kad se primenjuje *Manhhatan* formula (Harrington, 2016).

Kako se sličnost uzoraka, meri međusobnim rastojanjem, ovaj algoritam je osetljiv na postojanje razlika u rangu vrednosti prediktora, te ih je iz tog razloga potrebno transformisati, kako bi njihove vrednosti bile uporedive (eng. *feature scaling*). Moguće je primeniti *min- max* normalizaciju,

Ukoliko prediktori imaju noramalnu distribuciju, predlaže se standardizacija (*z* Normalizacija) podataka (Gareth, 2017), kojom srednja vrednost svodi na nulu a, standardna devijacija jedinici.

Pored razdaljine uzoraka, poređenjem pravaca vektora, koji te uzorke određuju, može se takođe odrediti njihova sličnost. Kada su vektori dva uzoraka približno upravni, oni se smatraju različitim. Kada se pravci poklapaju, uzorci su slični. Metodom *Cosine*, meri se kosinus ugla koji dva vektora zahvataju,

, za vrednosti

Sličnost dva uzoraka, gde je uzorak određen skupom svojih prediktora, može se predstaviti, kao količnik broja elemenat koji su u oba skupa isti (njihov presek) i ukupnog broja različitih elemenata skupova (njihova unija). Ovakav način merenja sličnosti, naziva se *Jaccard (*Alboukadel, 2017),

,

Veći broj istih elemenata dva skupa, znači i veću sličnost uzoraka. Kada su uzorci isti, *Jaccard* sličnost je 100%.

Ovaj način merenje sličnosti, se često koristi u analizi dokumenata. Za razliku od *Cosine*, ključne reči se ne predstavljaju vektorom, već skupom. Primenjuje se kao i *Cosine*, u slulaju velikih količina podataka za analizu.

Potrebno je još napomenuti da veći broj susednih uzoraka (*K*), daje veću tačnost *K-NN* algoritma, ali i veću kompleksnost. Do optimalnog broja suseda, uobičajeno dolazimo postupkom *cross* *validation*, odnosno postizanja minimalne *cross validation* greške.

*K-NN* je jednostavam algoritam, jedini hiperparameter koji je potrebno odrediti je *K*. Sa druge strane, ukoliko se radi, o podacime velike dimenzionalnosti, performanse ovog algoritma se zanačajno smanjuju.

**4.4 SUPPORT VECTOR MACHINE - *SVM***

*SVM* pripada grupi diskriminativnih, nadgledanih algoritama, koji se najčešće primenjuje u klasifikacionim, a manje u regresionim problemima. Razvio ge je Vapnik (1995), te tako spada u novije algoritme mašinskog učenja. Primenom *SVM* , kao i kod ostalih diskriminativnih algoritama, dolazimo do optimalne, granične hiperravni (eng. *hiperplane*), koja na najbolji način razdvaja uzorke (u multidimenzionalnom prostoru) različitih klasa. Kako se uzorak može predstaviti tačkom u prostoru , uzorak se klasifikuje prema svom položaju u odnosu na graničnu hiperravan, dimenzija m -1 . Broj mogućih graničnih hiperavni he neograničen.

Slika 24: Primer dvodimenzionalnog prostora, , u kojem je granična hiperravan prava

Za razliku od modela logističke regresije, gde se iz skupa mogućih graničnih hiperravni, optimalna određuje metodom maksimalne izglednosti (eng. *maximum likelihood – ML*), kojom se ne uzima u obzir razdaljina tačaka (uzoraka) od hiperravni, u *SVM* optimalna granična hipperavan je ona, za koju je rastojanje (margina) između maeđusobno najbližih tačaka različitih klasa najveće. Položaji ovih tačaka određeni su vektorima , koji se nazivaju *support* vektorima. Optimalna hiperravan prolazi sredinom margine i naziva se hiperravna maksimalne margine (eng. *maximum margine hiperplane*). Pouzdanost, tačnost SVM modela, je utoliko veća ukoliko je margina šira, čime se verovatnoća pogrešne klasifikacije uzoraka smanjuje. Algoritam prepoznaje tačke (uzorke) koje su najbliži hiperravni, određene *support* vektorima, koje jedino utiču na njen položaj. Ovo je jedan od razloga zašto *SVM* ima dobru mogućnost generalizacije.

Slika 25: Granična heiperavan, je prava linija u dvodimenzionalnom prostoru,koja razdvaja uzorke dve klase

margina

[Cite your source here.]

pozitivna hiperravna

negativna hiperravan

granična hiperravna

**2**

**1** pozitivna klasa,

negativna klasa,

3.4.1 Određivanje položaja granične hiperavni, *SVM*

Neka svaki trening uzorak određen vektorom , pripada jednoj od dve klase binarne promenjive = . Ako je vektor , upravan na hiiperavan, a vektor određuje položaj tačke testnog uzorka (kojeg želimo da klasifikujemo), prvi potreban uslov da uzorak npr. pripada pozitivnoj klasi (Slika 25) data je izrazom (Winston, 2010),

(1)

Ukoliko je skalarni proizvod ova dva vektora (projekcija vektrora veći i jednak od neke konstantne vrednosti *c*, uzorak će biti klasifikovati kao pozitivan. Odnosno što je vrednost skalarnog proizvoda ova dva vektora veća, uzorak će biti udaljenija ( mereno u pravcu vektora ) od koordinatnog početak, te je verovatnoća da je uzorak pozitivan veća.

Ovaj izraz možemo transformisati,

(2)

Konstanta b =-c

Izraz (2), nazivamo osnovnim pravilom odlučivanja u *SVM* , za pozitivne uzorke (pripadaju pozitivnoj klasi) (Winston,2010). Kako bi došli do vrednosti nepoznatih , postavljamo još dva dodatna uslova (Winston,2010):

, za uzorke koji pripadaju pozitivnoj klasi (3)

, za uzorke koji pripadaju negativnoj klasi

(osim pozitivnog uzorka određenog *support* vektorom)

vektor uzorka negativen klase (osim negativnog uzoraka određenoh *support* vekotorom)

Za različite vrednosti , izrazi (3) se mogu napisati u sledećem obliku,

(4)

Dalje ćemo za uzorke koji se nalaze na granicama margina (pozitivna ili negativna hiperravna) i koji su određeni *support* vektorima, pretpostaviti da je (Winston,2010),

(5)

Kako je već rečeno, položaj granične hiperravni određuje se tako da je širina margine (koju ova hiperravan polovi) najveća. Margina je jednaka skalarnom proizvodu, razlike *support* vektora i jediničnog vektora upravnog na hiperravan

Širina margine (6)

Za uzorke na granicama margine ( određene *support* vektorima), iz (5) sledi da za različite vrednosti važi:

Sada jednačinu (6) možemo napisati u obliku,

Širina margine (7)

Iz (7) sledi, da uslov maksimalne širine margine, koju smo definaisali jednačinom (6), se dobija za minimalne vrednosti inteziteta vektora minimalnu vrednost ,odnsno minimalna vrednost (ovako definsan uslov za min vrednost je uveden kako bi se pojednostavilo računanja parcjlnih izvoda u daljem postupku), uz ispunjenje uslova datog jednačinom (5). Ovo predstavlja optimizacijski problem, sa Lagranžovom funkcijom oblika,

(8)

Tražimo dakle minimlanu vrednost ove funkcije u odnosu na iz uslova da su sledeći parcijalni izvodi jednaki nuli:

dobijamo da je:

(9a)

(9b)

Kada (9a i 9b), zamenimo u jednačinu (8), sledi,

= (10)

Ovo je novi optimizacijski problem, čijim rešavanjem, dobijamo vrednosti za , uz usov . Potom iz (9a) dobijamo .

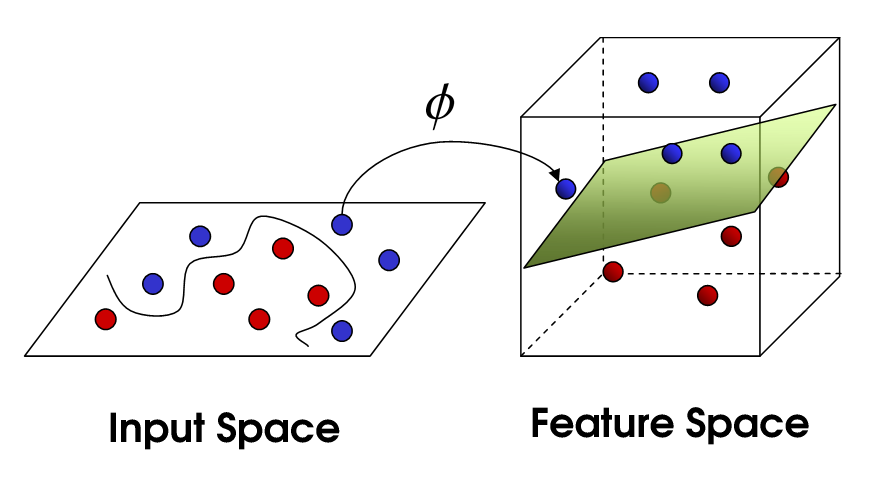
Kako je , iz (5) dobijamo vrednost preostale nepoznate *b*,

(11)

Sa poznatim b i , primenomo osnovnog pravila odlučivanja (2), možemo utvrditi da li testni uzorak određen vektorom pripada pozitvnoj klasi ili ne.

Ukoliko nije moguće napraviti linearnu separaciju uzoraka dve (ili više) klase kao u gornjem primeru (kada govorimo o eng. *linear support vector machine*), pristupa se transformaciji vektora prediktora *x*, iz početnog *p* dimenzionalnog prostora ( tzv. *input space*), u prostor *m* dimenziija, gde je *m>p* (eng. *feature space*). Odnosno, poterbno je transformisati originalne podatke trening seta, primenom neke funkcije *f*(, tako da u novom, *m* dimenzionalnom prostoru, separacija uzoraka različitih klasa lineranom hiperravnom postane moguća.

Slika 26: Transfomacija prediktora funkcijom iz originalnog prostora u prostor veće dimenzionalnosti[[14]](#footnote-15)



Kako bi ovakva transformacija trening podataka bila veoma zahtevna ( veliko vreme procesiranja), *SVM* algoritam primenjuje postupak nazvan *Kernel trick* (Alboukadel,2017). Tako , karnel funkcija , koja kao ulazne atribute, ima vektore prediktora originalnog, *input* prostora, , kao output daje skalarni proizvod transformisanih vektora () u novom prostoru više dimenzije (*feature space*). Na ovaj način podaci su iz prostora , *m>p* ,

(12)

Na ovaj način bez potrebe za transformacijom podataka, primenom karnel funkcije, dobijamo nove vrednosti vektora prediktora, u prostoru veće dimenzionlanosti.

Kako smo i videli iz (10), Lagranžova funkcija, koja je linearno zavisna od skalarnog proizvoda vektora , primenom Karnel trika se može napisati kao,

= = (13)

Neopznata vrednost *b*, na osnovu (11) je jednaka,

(14)

Rešavanjem optimizacijskog problema (13) dobijamo , a iz (14) dobijamo *b*. Transformacijsku funkciju *f*, nije potrebno poznavati (odnosno transformisati uzorke), već samo primeniti odgovarajuću Karnel funkciju *K* . Sa poznatim i *b*, proveravamo ispunjenost uslova (2),

(15)

Ako je nejednačina (15) tačna, uzorak određen vektorom

Najčešće korišćene Kernel funkcije su (Alboukadel,2017):

* Polinomalna funkcija stepena *d* oblika,
* *Radial basis* funkcijaoblika*,*

Iako je razvijeno dosta teorijskih metoda za izbor odgovarajuće *Kernel* funkcije, do optimalne, u praksi najčešće dolazimo postupcima *bootstraping-*a i *cross validation*.

U poređenju sa ostalim klasifiakcionim algoritmima, *SVM* se ne preporučuje u slučajevima velikih setova podataka, kada se ne može postići dobra separacija uzoraka različitih klasa.

*SVM* se uobičajeno primenjuje u slučajevima visoko dimenzionalnih podataka, specijalno kada je broj prediktora veći od broja uzoraka. Ovaj algoritam ima dobru mogućnost generalizacije.

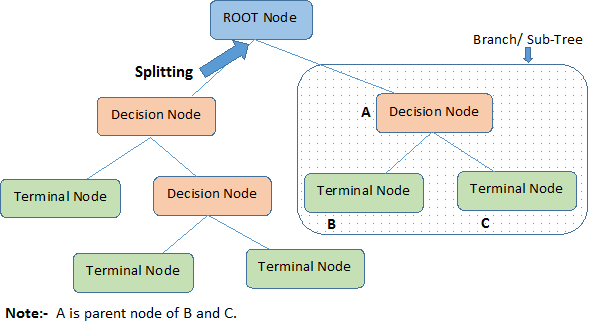
**3.5 KLASIFIKACIONI MODELI NA BAZI STABLA ODLUČIVANJA**

4.5.1 Stablo odlučivanja, *DT*

Stablo odlučivanja (*decission tree* - *DT*) spada u grupu nadgledanih klasifikacionih diskriminantnih modela *ML*, koji se uspešno može primenjivati i u regresionim problemima. Model ’Klasifikaciona i regresiona stabla’ (eng. *Clasification and Regression Tree* ), prvi je predstavio Breiman et al (1984).

*DT* se razvija sukcesivnim particionisanjem podataka (eng. *data splitting*), na osnovu graničnih vrednosti prediktora. Iz polaznog noda ili čvora (*root node*), koji predstavlja grafički prikaz početnog skupa trening podataka, isti se deli u manje podgrupe (particije). Svaka nova particija, predstavljena je novim čvorom odlučivanja (eng. *decision node* ). Postupak granjanja stabla, završava se kad se postigne zadovoljavajuća homogenost uzoraka u krajnjim (terminalnim) čvorovima stabla (u slučaju klasifikacionih problema), postigne unapred projektovana dubina stabla ili maksimalni broj nodova . Kreirano stablo, predstavlja model odlučivanja.

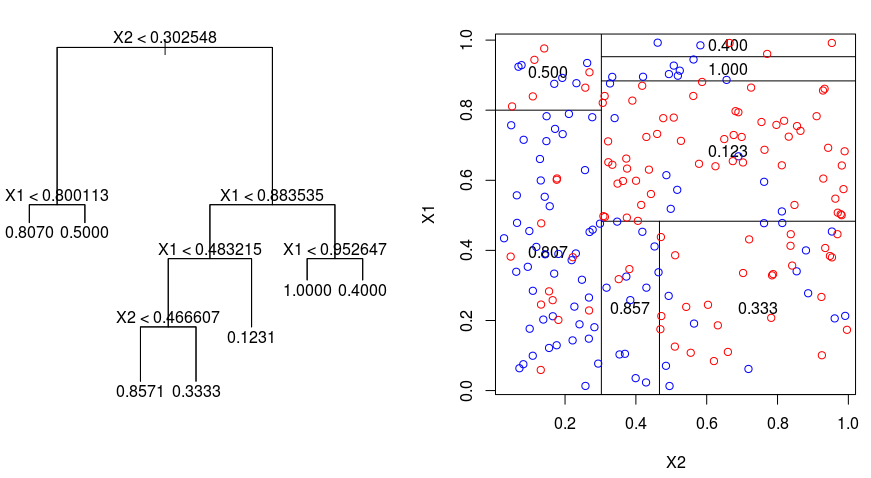
Slika 27. Rekursivno particionisanje stabla[[15]](#footnote-16)



Novi uzorak, prolaskom kroz model stabla, ispunjavajući pravila particionisanja, dolazi do odgovarajućeg terminalnog noda. U slučaju klasifikacionih problema, uzorak će biti dodeljen, većinskoj klasi uzoraka u nodu. Kada je problem regresioni, zavisna promenjva novog uzorka, jednaka je srednjoj vrednosti zavisne promenjive trening uzoraka terminalnog noda.

Na slici 28, dat je primer stabla odlučivanja, na bazi dva prediktora, koje je rezultiralo sa sedam terminalnih čvorova. Svakom terminalnom čvoru odgovara jedna particija, određena graničnim vrednostima prediktora. Novi uzorak će biti dodeljen većinskoj klasi u odgovarajućem terminalnom nodu.Generalizacijom ovog pristupa na veći broj prediktora (*m*), zaključujemo da je svaka particija ograničen u *m* dimenzionalnom prostoru.

Slika 28. (a) Stablo odlučivanja (b) Uzorci u prostoru (dimenzija prostora određena brojem prediktora). Svakom terminalnom čvoru odgovara jedna particija, sub – prostor, odrđena grančnim vrednostima prediktora . Zavisna promenjiva je bionominalna.



Izbor i redosled prediktora, na osnovu kojih se podaci particionišu, kao i njihove granične vrednosti, određuju se na osnovu njihove informativnosti (eng. *information gain*).

U slučaju klasifikacionih problema, informativnost prediktora je mera povećanja homogenosti (eng. *purity,* čistoća) uzoraka u nodu, nastalih particionisanjem na bazi njegove granične vrednosti. Pod homogenosti uzoraka noda, podrazumeva se nivo uređenosti skupa, odnosno stepen prisustva uzoraka iste klase. Mera homogenosti noda ili skupa, naziva se entropijom (eng. *entropy*) (Provost i Fawcett, 2013), i može imati vrednosti u rangu od 0 do 1. Nodovi, kod kojih svi uzorci pripadaju istoj klasi, su potpuno homogeni i imaju entropiju jednaku nuli. Maksimalnu entropiju, imaju heterogeni skupovi, u kojim je broj uzoraka različitih klasa jednak.

Slika 29: Entropija kao mera uređenosti skupa uzoraka dve klase (+ i -). p(+), veorvatnoća da uzorak pripda pozitivnoj klasi[[16]](#footnote-17)

Diagram

Description automatically generated

Entropiju skupa, odnosno uzoraka u nodu, jednaka je,

, verovatnoća da uzorak pripada klasi

*k* , broj klasa zavisne promenjive

Entropija noda, koji particionisanjem nastaje (eng. *child node*), na osnovu granične vrednosti prediktora, ne daje podatak o ukupnoj informativnosti tog prediktora (Provost i Fawcett,2013). Pokazatelj za koliko se entropija noda (eng.  *parent nod* ) smanjila, nakon particionisanja, je osnovna mera informativnosti prediktora i naziva se *Information Gain* - *IG* ( Gareth, J. et, 2014). Tako za prediktor na osnovu kojeg se nakon particionisanja uzoraka *paretn* noda, postigne najveća uređenost *child* nodova, kažemo da ima najveći *IG*.

*parent* nod

*child* nod

- broj uzoraka u

ukupan broj uzoraka parent noda *P*

*m* – broj *child* nodova , koji particionisanjem nastaje

Prediktor sa kojim se postiže maksimalna vrednost *IG*, je onaj sa sa kojim particonisanje, odnosno granjanje stabla započinje (Provost i Fawcett, 2013). Dalje particionisanje se nastavlja sa prediktorima prema opadajućim vrednostima *IG*, koji se njima postižu.

Pored gore opisanog postupka, informativnost prediktora u binarnim klasifikacionim problemima, je moguće odrediti na osnovu vrednosti *Gini* *impurity* indeksa (Provost i Fawcett, 2013). Ovaj indeks je mera neuređenosti skupa. Za uzorke *child* noda, jednak je,

*k*, klase zavisne promenjive

, verovatnoća da uzorci pripadaju klasi *i*

Minimalna vrednost vrednost *Gini* indeksa je 0, u slučaju kada svi uzorci pripadaju istoj klasi. Najveća moguća vrednost je 0.5 , kada je broj uzoraka dve klase jednak.

Vrednost *Gini* indeksa *parent* noda, jednaka je ponderisanoj vrednosti sume *Gini* indeksa *child* nodova,

*m* – broj *child* nodova , koji particionisanjem nastaje

- broj uzoraka *child* noda  *i*

ukupan broj uzoraka parent noda *P*

Predikotor sa najmanjom vrednošću indeksa, je najinformativniji.

Postupak sukcesivnog particionisanja, razvoj stabla, nastavljamo sve dok je vrednost *IG* značajna (Gareth et al, 2014). Ukoliko je dobitak u *IG*, sa novim particionisanjem minimalan, razvoj stabla se zaustavlja. Postupak ograničavanja rasta stabla odlučivanja, naziva se postupkom regularizacije modela *DT* . Tako prema Zhang (2016), potreba za daljim particionisanjem prestaje, kada se ispune neki od sledećih uslova:

* Svi teminalni nodovi su potpuno homogeni
* Postignuta je predefinisna vrednost minimalnog broja uzoraka u terminalnom nodu

* Prekoračena je maksimalna dubina stabla
* Postignut je maksimalan broj nodova u stablu
* Ograničavanje broja prediktora na osnovu kojih se moguće particionisati

U slučaju regresionog stabala odlučivanja, izbor prediktora i graničnih vrednosti, bazira se na vrednosti sume kvadrata reziduala. Odnosno, sumi kvadrata razlika vrednosti zavisne promenjive uzoraka *child* noda i njene srednje vrednosti ( Gareth et al, 2014),

*RSS* – suma kvadtata rezidualnih vrednosti

c, broj *child* nodova koji particionisanjme nastaje

*l*, broj uzoraka u *child* nodu

, srednja vrednost zavisne promenjive u nodu *i*

, vrednost zavisne

Najinformativniji prediktor je onaj, za koji je vrednost *RSS* najmanja.

Jasno je da što je dubina stabla veća, ono je kompleksnije i teže za interpretaciju. Do optimalne dubine, dolazimo uvođenjem *cost* *complexity* parametra , kojim se penalizuje vrednost *RSS* (Peter Bruce i Andrew Bruce, 2017).

Tako uslov postizanja minimlane vrednosti *RSS* postaje,

*T*, broj terminalnih nodova stabla

Mala vrednost , proizvodi kompleksnija stabla, dok veće vrednosti daje stabla manje dubine.

Iz ovog sledi, da stablo treba razivijati sve dok je smanjenje *RSS*, koje se postiže rastom dubine stabla, veće od vrednosti za koji se model penalizuje (). Do optimalne vrednosti , dolazimo postupkom *cross* validacije ( Gareth, et al 2014).

U slučaju regresionog stabla, informativnost prediktora se može odrediti i na osnovu varijanse zavisne promenjive u terminalnim čvorovima ( Gareth et al, 2014),

*Va*r

*n* broj uzoraka u nodu

Za izabrani prediktor (), računa se varijansa zavisne promenjive svakog *child* noda, koji nastaje račvanjem na osnovu njegove granične vrednosti. Varijansa parent noda jednaka je ponderisanoj vrednosti zbira varijansi svakog noda,

broj uzoraka u prvom nodu

broj uzoraka u drugom nodu

ukupan broj uzoraka

varijansa zavisne promenjive uzoraka u prvom nodu

varijansa zavisne promenjive uzoraka u drugom nodu

varijansa zavisne promenjive *parent* noda

Redosled prediktora, na bazi kojih se stablo grana, određuje se prema rastućoj vrednosti varijanse. Odnosno, najinformativniji prediktor, na osnovu čijih graničnih vrednosti počinje razvoj stabla ( početni čvor), je onaj za koga je vrednost varijanse najmanja (Peter Bruce i Andrew Bruce, 2017)

Regresioni i klasifikacioni *DT* algoritmi, se mogu dobro interpretirati i to je jedna od njihovih osnovnih prednosti. Kako *DT* model, sadrži samo jedno stablo odlučivanja, predikcije ovim modelom nisu stabilne. Opasnost od overfitinga (u slučaju kompleksnog stabla) i slabe generalizacije, jesu osnovni nedostaci ovog modela.

*DT*, se primenjuje u slučajevmia, kada je relacija zavisne promenjive i prediktora kompleksna i kad se ne može aproksimirati linearnom funkcijom. Ovaj algoritam, ne zahteva transformaciju podataka što dodatno pojednostavljuje njegovu primenu.

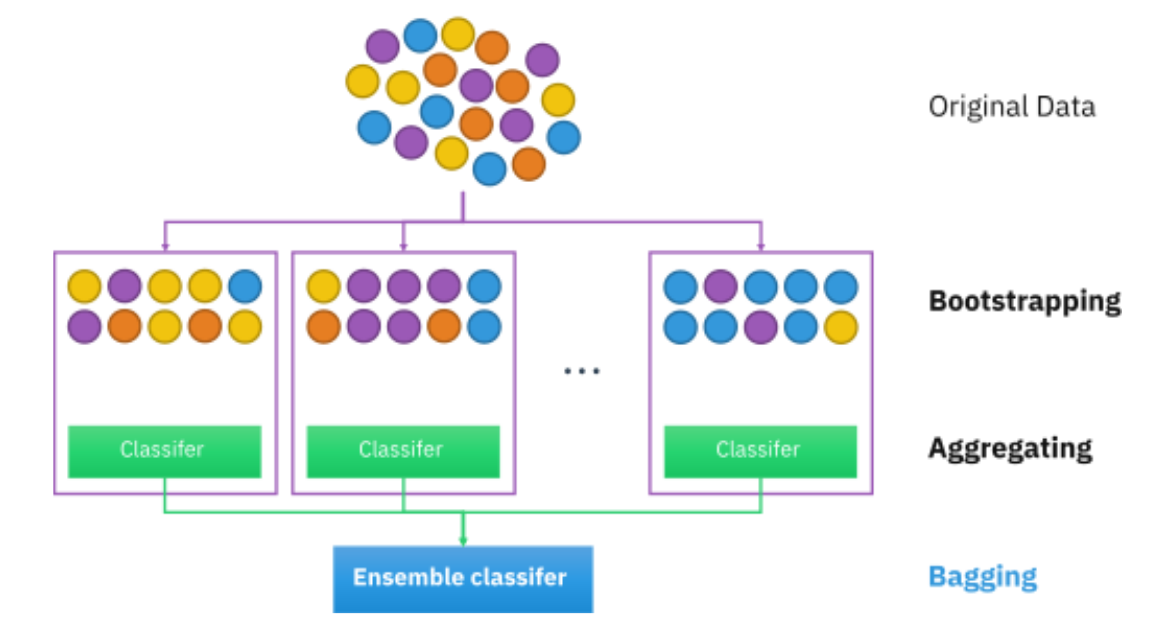
4.5.2 RANDOM FOREST- *RF*

*RF* je sličan *DT*, sa razlikom de se ovim modelom kreira više stabala odlučivanja. Na taj način se rešava osnovni problem nestabilnosti predikcije *DT*, uzrokovan postojanjem samo jednog stabla odlučivanja (Bruce, P. i Bruce A. , 2017) .

Svako pojedinačno stablo, kao element *RF*, razvija se na skupovima podataka istog broja uzoraka, koji su podskupovi originalnog trening *data* seta i nazivaju se *bootstrap*. Uzorci *bootstrapa*, biraju se postupkom uzorkovanja sa zamenom *(*eng*. with replacement)*, tako da se svaki uzorak može jednom ili više puta pojaviti u istom *bootstrap*-u. Uzorci koji su ‘izostavljeni’ i ne pripadaju niti jednom *bootstrap*-u, sačinjavaju tzv. *out of bag dataset - OBD*, koji se koristi za testiranje i proveru tačnosti prediktivnog *RF* modela.

Postupak kojim se agregiraju rezultati, dobijeni primenom pojedinačnih algoritama, u slučaju *RF* pojedinačnih stabala, na zasebnim *bootstrap* podacima , naziva se *bagging,*  što je skraćenica od *bootstrap aggregating* (Leo Breiman, 1994).

Slika 30. *Bootstrap aggregating*[[17]](#footnote-18)



Ovim postupkom, postiže se veća stablinost i tačnost predikcija *ML* algoritama, smanjuje mogućnost overfitinga i povećava stepen generalizacije (Breiman, 1994). *RF* spada u grupu *bagging* algoritama, gde svako kreirano stablo predstavlja zaseban klasifikator, dok je finalna predikcija rezultat agregacije predikcija svakog pojedinačnog stabla.

U slučaju regresionih problema, konačni rezultat, jednak je srednjoj vrednosti zavisne promenjive u odgovarajućim terminalnim nodovima, svakog kreiranog stabla. Kod klasifikacionih problema, konačna predikcija modela, jednaka je preovlađujućoj klasi odgovarajućegih terminalnih nodova (eng. *majority* *voating)* svakog stabla.

Slika 31: *RF* klasifikacioni model[[18]](#footnote-19)



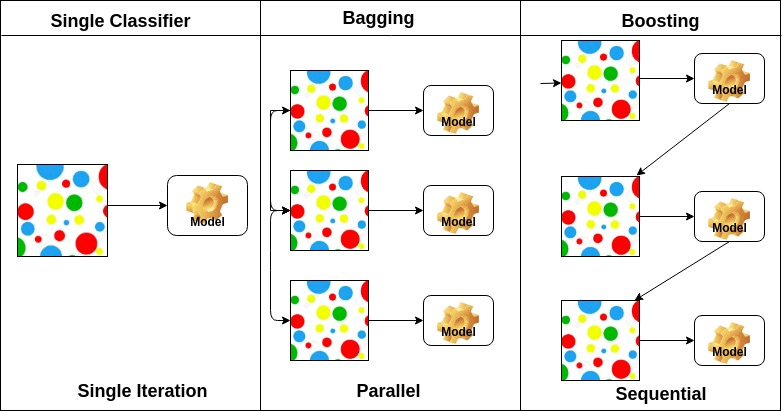
Kada *RF* kreira novo stablo, koristi se samo ograničen broj proizvoljno izabranih prediktora. Ako je *m* njihov ukupan broj, u slučaju klasifikacionih problema, za razvoj pojedinačnog stabla koristi se , a u slučaju regresionih problema *m*/3, proizvoljno izabranih prediktora (Breiman i Cutler, 2018). Kako je ovaj broj prediktora, značajno manji od njihovog ukupnog broja, model svakog stabla biće različit, a stabala međusobno nezavisna. Kao i kod *DT*, redosled prediktora i njihove granične vrednosti, na osnovu kojih se stablo račva, određuju se prema opadajućoj vrednosti *IG* *(*eng*. inforamtion gain)*. Korišćenje samo ograničenog broja prediktora i velikog broja stabala, jest osnovni razlog veće tačnosti predikcije i manje mogućnosti overfitinga, u odnosu *DT* . S druge strane, za ‘učenje ’ ovog modela, potreban je veliki broj uzoraka i resursa, imajući u vidu da je uobičajen broj stabala *RF* veći od 500 (Breiman and Cutler, 2018).

Za razliku od standardnih pristupa, rešavanju problema nedostajućih podataka, kao što su maksimalna izvesnost i regresija, *RF* algoritam rekonstruiše ili procenjuje podatke koji nedostaju, primenom tzv. *proximity* matrice (kvadratne matrice, reda koji odgovara broju uzoraka). Elementi ove martice, predstavljaju meru međusobne sličnosti uzoraka. Pod sličnim uzorcima, smatraju sa oni, koji pripadaju istom terminalnom nodu, nakon ‘prolaska’ uzorka kroz *RF* stabo. Vrednosti *proximity* matrice se kreću u rangu od 0 za različite, do 1 za slične uzorke. Maksimalno slični uzorci u svakom stablu *RF*, pripadaju istom terminalnom nodu. Suprotno, uzorci koji se u svim stablima, nalaze u različitim teminlanim nodovima, su različiti. Kako se *NA* prediktora, može rekonstruisati na osnovu postojećih podataka preostalih uzoraka, pojedinačni uticaj svakog od tih uzoraka, biće srazmeran njegovoj sličnosti, sa uzorkom za koji procenjujemo (rekonstuišemo) nedostajući podatak. Tako će najsličniji uzorci imati najveći uticaj na njegovu rekonstruisanu vrednost.

*RF* spada u jedne od najtačnijih prediktivnih algoritama, kako za klasifikacione, tako i regresione probleme. Ne zahteva pripremu, odnosno transformaciju podataka. Algoritam nije osetljiv na ekstremne vrednosti. Kako su stabla ovog modela bazirana na manjem, ograničenom broju prediktora, *RF* se primenjuje u slučajevima kada je broj prediktora velik. Osnovni nedostatak *RF*, je njegova kompleksnonst, odnosno nemogućnost tumačenja, zbog velikog broja stabala.

U *RF* *bagging* algoritmu, svako stablo odlučivanja ima isti značaj (eng. *weight* ) u odlučivanju. Stabla su nezavisna i kreirana istovremeno. Kada se stabla kreiraju sukcesivno i kada su ona međusobno zavisna, tako da svako novo stablo, uzima u obzir grešku predikcije načinjenu prethodnim, govorimo o novoj grupi klasifikacionihalgoritama nazvanih *Boosting*.

Slika 32.  *Bagging* i *Bossting* klasifikacioni algoritmi[[19]](#footnote-20)



Postupkom *boosting-a* , suksecivno kreirani tzv. slabi klasifikatori (eng. *weak classifiers - WC)*, vrlo jednostavna stabla odlučivanjanja, niske tačnosti predikcije, kombinuju se u jake klasifikatore (eng.  *strong classifiers - SC*), kojma se postiže zadovoljavajuća tačnost (Bottou, 2010).

*Boosting* algoritmi se uspešno primenjuju u rešavanju binarnih klasifikacionih problema. U slučaju kada je zavisna promenjiva kategorička, sa više od dve klase, problem se svodi na višestruko rešavanje binarnih problema (Freund i Schapire, 1997; Schapire i Singer, 1999).

Ako za binarni klasifikacioni problem gde je , slab binarni klasifikator označimo sa a jak sa , onda osnovni princip *boosting*  algoritama, po kojem *SC* klasifikuje uzorak na osnovu rezultata sukcesivnih klasifikacija više *WC*, možemo napisati kao:

(1)

*b,* ukupan broj slabih klasifikatora *WC*

, koeficijent koji određuje značaj svakog *WC* pri odlučivanju

U zavisnosti od strukture *WC* i načina na koji se određuje njihov uticaj (factor ), razlikujemo sledeće *boosting* algoritme: *Adaboost, Gradinet tree boost* i *Extreme gradient boost*.

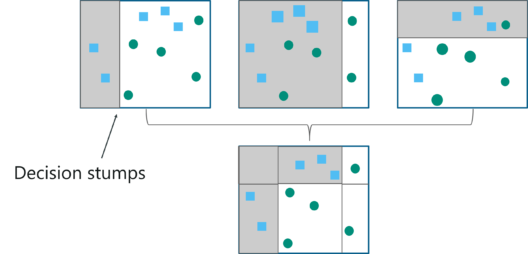
3.5.3 ADABoost – Binarni klsifikator

*Adaboost* algoritama, se primenjuju u binarnim kalsifikacionim problemima. Za razliku of *RF* modela, stabla koja se kreiraju ovim modelom, imaju samo jednu granu (eng. *decision* *stump)* (Freund i Schapire ,1997). Tako se pri kreiranju *stump-a*, koristi samo jedan prediktor. Broj *stump*-ova, odgovara broju prediktora. Onaj sa kojim se postigne najveća tačnost predikcije, u datoj iteraciji, postaje *WC* najvećeg značaja.

Za uzorke, dvodimenzionalnog prostora, granične vrednosti prediktora, određuju položaj prave, koja razdvaja uzorke različitih klasa. Značaj (eng. *weight*) svakog pojedinačnog uzorka, koji je pogrešno klasifikovan primenom *WC* , se povečava u narednoj iteraciji. Tako se sa svakim novim *WC*, povećava verovatnoća, da uzorci sa većim značajem, budu pravilno klasifikovani. Za razliku od *RF*, gde se stabla kreiraju nezavisno, u *Adaboost* -u, svako novo stablo, uzima u obzir grešku nastalu primenom prethodnog *WC*. Kombinovanjem svih *WC,* dobijamo konačni *SC* ,kojim se postiže bolja tačnost predikcije.

Slika 33. Tri *WC* binara klasifikatora, kombinovani u SC[[20]](#footnote-21).

WC WC WC



SC

Ako sa označimo značaj (‘*težinu’*) *i* uzorka, u trenutku *t* = 0, u prvoj iteraciji ona će biće ista za sve uzorke,

*n,* ukupan broj uzoraka

Postavićemo dalje uslov da suma ‘težina’ svih uzoraka, u svakoj iteraciji ( za svako *t*), mora biti jedan,

Ukupna greška klasifikacije (eng. *total error*) slabog klasifikatora *WC,* u trenutku *t* = 0, jedanka je,

*j,*  broj pogrešno klasifikovanih uzoraka

U svakoj iteraciji, slabi klasifikatori *(WC)* se kreiraju za svaki prediktor. Onajza koji je broj pogrešno klasifikovanih uzoraka najmanji, ononsno ima najnižu vrednost, postaje izabrani *WC* u datojiteraciji,

odnosno *WC* najvećeg značaja.

Primenom tako izabranog *WC*, dobijamo prediktivnu vredsnot .

Težine uzoraka u narednoj iteraciji funkcija su prethodnih težina i tačnosti predikcije u trenutku *t* (Winston, 2010),

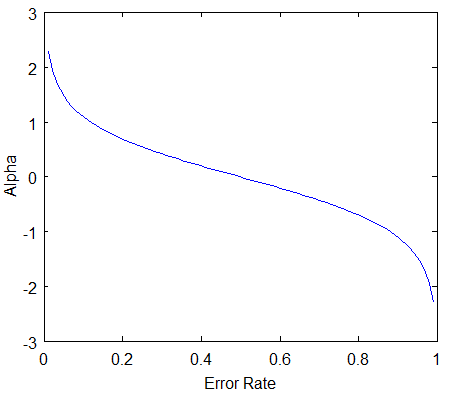
(1)

z je tzv. noramlizacioni faktor, kojim se obezbeđuje ispunjavanje uslova da je zbir težina u svakoj iteracije jednak jedan

, je faktor značaja slabog klasifikatora (eng*. amount of say)* (Winston, 2010),

Iz grafičkog prikaza funkcije , vidimo da manjim vrednostima ukupne greške , odgovaraju veće vrednosti

Slika 34. Rastom ukupne greške klasifikacije , značaj slabog klasifikatora se smanjuje (Masashi & Sugiyama , 2016)



Proizvod ima vrednost + 1, kada je predikcija sa *WC* tačna i -1, kad je netačna (). Sledi da (1) možemo napisati u sledećem obliku,

– za pravilno klasifikovane uzorke (2a)

– za pogrešno klasifikovane uzorke (2b)

Eksponencijalne funkcije date izrazima (2a i 2b), pokazuju da za pravilno klasifikovane uzorke, rastom se smanjuje njihov značaj, . Za pogrešno klasifikovane, vrednost raste sa rastom .

Slika 35. (a) Eksponencijalna funkcija pravilno klasifikovanih uzorka (b) eksponencijalna funkcija za nepravilno klasifikovane uzorke

1. (b)

Jednačinama (2a i 2b), daljom transformacijom postaju:

– za pravilno klasifikovane uzorke (3a)

– za pogrešno klasifikovane uzorke (3b)

Kako je zbir ’težina’ svih uzoraka jednak 1, sledi,

Odnosno:

= z

*k*, broj pravilno klasifikovanih uzoraka

*j*, broj pogrešno klasifikovanih uzoraka

Kako je , a , sledi da je normalizacioni faktor jednak,

(4)

Kada vrednost  *z*, uvrstimo u 3a i 3b, dobijamo finalno (Winston, 2010),

- za pravilno klasifikovane uzorke (5a)

- za pogrešno klasifikovane uzorke (5b)

Na slici 36, prikazan je model *Adaboost* algoritma. U prvom *bootstrap* – u *D1*, svi uzorci imaju isti značaj . Model *M1*, prvi *WC* na podacim *D1*, je slab klasifikator najvčeg značaja (za koji faktor ima najveću vrednost u prvoj iteraciji). Uzorci koji su primenom *M1*, pogrešno klasifikovani, u sledećoj iteraciji dobijaju veći značaj, od uzoraka koji su pravilno klasifikovani. Na osnovu novih vrednosti formira se novi *bootstrap* (*D2*), u kojem veću zastupljenost imaju pogrešno klasifikovani uzorci, iz prethodne iteracije. Postupak se dalje nastavlja kreiranjem novog *WC* , model *M2*. Na osnovu rezultata klasifikacije ovim modelom, računaju se nove vrednosti , a na osnovu njih formira *bootstrap* *D3*, itd. Ovaj postupak kreiranja *WC* se nastavlja dok se ne postigne njihov unpared definisani maksimalni broj ili minimalna projektovana dozvoljena greška predviđanja, uz uslov da je broj kreiranih *WC* neparan. Konačna predikcijs *SC*, je rezultat kombinovanja svih kreiranih *WC* i jednaka je većinskoj klasi iz skupa predikcija, svakog *WC* pojedinačno,

*t,* ukupan broj *WC*

Slika 36. Adaboost model

*SC*

Bootstrap, *D3*

Predikcija primenom *WC*, Model 3, na podacima D1

*WC3*- Model 3

Bootstrap, *D2*

Predikcija primenom *WC*, Model 2, na podacima D1

*WC2*- Model 2

Bootstrap, *D1*

Predikcija primenom *WC*, Model 1, na podacima D1

*WC1*- Model 1

*AdaBoosting* je relativno jednostavan za implementaciju i nije podložan overfitingu. Treba imati u vidu da je dosta osetljiv na ekstremne vrednosti prediktora.

3.5.4 Gradient tree boosting, *GTB*

*GTB* takođe spada u grupu *boosting* algoritama, kod kojeg se kreira veći broj stabala odlučivanja - *WC*, pri čemu svako stablo uzima u obzir grešku predikcije prethodnog. Za razliku od *AdBoost*-a, gde stabla odlučivanja imaju strukturu *stump-a*, kod *GTB* algoritama, stablo se razvija na osnovu graničnih vrednosti više prediktora.Takođe, predviđena vrednost zavisne promenjive u prvoj iteraciji, koja je kod *Adaboost* algoritma rezultat primene prvog *WC* , kod *GTB* modela je u slučaju regresionih problema, srednja vrednost zavisine promenjive *bootstrapa*. U slučaju kategoričke zavisne promenjive, njena predviđena vrednost, kojom se inicijalzuje *GTB* model, () jedanka je,

(1)

izglednost pozitvnog događaja (

se takođe može prikazati i kao verovatnoća pozitivnog događaja (),

(2)

Ako usvojimo da je granična vrednost verovatnoće 0.5, svaki novi uzorak biće klasifikovan kao pozitivan događaj, ako je vrednost .

Može se pokazati da je inicijalna vrednost *GTB* modela, data jednačinama (1) ili (2) , optimalna (Friedman, 2001).

Kako skup datih vrednosti zavisne promenjive , predstavlja skup nezavisnih događaja, koji imaju binomnu distribuciju, ona je određena brojem uzoraka i parametrom *p*, verovatnoćom pozitivnog događaja, *.*

Zajednička distribucija binarne zavisne promenjive jednaka je,

.

Logaritmovanjem ovog izraza dobijamo,

(3)

Izraz (3), predstavlja logaritam izvesnosti, odnosno verovatnoće, da su vrednosti , uzrokovane iz binomne distribucije, određene verovatnoćom pozitivnog događaja

Funkcija gubitaka *LF* (eng. *loss function*) koja se koristi u *GTB*, a čiju vrednost želimo minimizirati, kako bi predikcija modelom imala najveću tačnost, u slučaju binarno klasifikacionih problema, jednaka je negativnoj vrednosti logaritma izglednosti binomne distribucije (3). Odnosno, uslov potizanja maksimalne vrednosti (3), je ekvivalentan uslovu postizanja minimalne vrednosti *LF*,

(4)

Iz razloga jednostavnijeg računanja parcijalnih izvoda, *LF* se prikazuje u funkciji , , odnonso , , izraz (4) se može napisati kao,

(5)

Potrebno ja tako naći verdnost odnono *p*, za koju funkcija gubitaka ima minimalnu vrednost (Friedman, 2001),

Iz uslova dobijamo,

0 (6)

Iz (6) se dobija de je,

broj negativnih događaja

ukupan broj događaja ili uzoraka

Tako se minimalna vrednost funkcije gubitaka, postiže za , odnosno kako je , sledi , tj. ) . Ovime je dokazano da je (1), najbolja aproksimacije za predikciju zavisne promenjive u prvoj iteraciji.

Model *GTB* dalje se razvija dodavanjem stabala odlučivanja (*WC*). Za razliku od *Adaboost* modela, gde se stabla razvijaju, nad podacima strukture (,), kod *GTB*, se formira nova struktura podataka, u kojoj se zavisna promenjiva, zamenjuje rezidulnim vrednostima, prethodne iteracije. Odnosno, struktura podataka postaje (,), gde je *m* broj iteracije u modelu. Ove rezidulane vrednosti dobijamo iz uslova (Friedman, 2001),

,

Tako posle nulte iteracije, )

Tako je ,

(7)

Kako se svako naredno stablo razvija na podacima oblika (, rezidualne vrednosti koje pripadaju istom terminalnom nodu, je potrebno transformisati u jedinstvenu vrednost. Ove vrednosti, predstavljaju meru sličnosti reziduala u terminalnom nodu i predstavljaju output vrednost , koja se stablom (*WC*) predviđa. U slučaju regresionih problema, do vrednosti dolazi se jednostavnim uzimanjem srednje vrednosti reziduala u terminalnim nodovima. U slučaju klasifikacionih preblema, ova vrednost , se određuje iz uslova minimalne vrednosti funkcije gubitaka (Fredamn, 2001),

(8)

, ukupan broj terminalnih nodova stabla

uzorci koji pripadaju terminalnom nodu

funkcija gubitaka

Na osnovu funkcije gubitaka za *GTB* (5), uslova (8) postaje,

Iz uslova , dobijamo

– rezidualne vrednosti uzorka *i ,*terminalnog čvora *j,* stabla *m*

- prediktivna verovatnoća, prethodne iteracije, uzoraka *i* ,terminalnog čvora *j*

Postupak kreiranja stabala, nastavljamo dok ne postignemo njihov maksimalni predviđeni broj ili kad se sa novim stabla, tačnost predikcije samo minimalno poveća (male rezidualne vrednosti ).

Slika 37. Razvoj *GTB* modela[[21]](#footnote-22)

Diagram

Description automatically generated

Tako da finalno prediktivni model, odnosno *SC* ima oblik:

N, maksimalni broj stabala *GTB* modela

, uzorak kojeg želimo da klasifikujemo, primenom *SC*

, parametar učenja

Broj stabala (N) i parametar učenja su hiperparameti *GTB* algoritma, koji značajno mogu uticati na tačnost prediktivnog modela. Veći broj stabala može dovesti do veće tačnosti, ali i do pojave *overfitinga*. Ukoliko se odlučimo za veće vrednosti parametra , potreban je manji broj stabala u modelu. Pravilnim izborom, ovih parametara (najčešće postupkom eng. *cross validation*), možemo doći do modela zadovoljavajuće tačnosti i dobre sposobnosti generalizacije (Friedman , 2001).

Pored *Adaboost* i *GTB*, često primenjivani *boosting* klasifikacioni algoritam je i *XGBoost* (eng. *Extreme gradient boosting* ). Sličan je *GTB*, ali kako omogućava paraleleno procesiranje, ima bolje performanse. *XGBoost,* za razlikuod *GDB,* u prvoj iteraciji ima fiksnu vrednost inicijalne prediktivne verovatnoće jednaku 0.5. Stabla odlučivanja se isto granaju, na osnovu graničnih vrednosti predikotra, za koje je se maksimalno poveća homogenost uzoraka, u nodovima koji granjanjem nastaju. Jedinstvena vrednosti, output vrednosti *R*, u terminalnim nodovima regresionog stabala, kod *XGBoost model,* jednaka je:

regularizacioni parametar

Sa rastom vrednosti parametre , povećava se stabilnost modela i može se dodatno uticati na njegovu mogućnost generalizacije*.*

Primenom *boosting* metoda, dodatno se povećava tačnost *DT* prediktivnih modela. Ove algoritme preporučeno je primenjivati u slučajevima velikog broja uzoraka (više od 1,000) i broja prediktora (do 100) (Kassambra,2017).

**4. ANALIZA TAČNOSTI KLASIFIKACIONIH ALGORITAMA**

Tačnost klasifikacionih algoritama merimo primenom matrice konfuzije (eng. *confusion matrix*), koja prikazuje broj tačno i netačno klasifikovanih uzoraka, za svaku klasu zavisne promenjive (Gareth et al, 2014). U slučaju binarno klasifikacionih problema (pozitivna i negativna klasa), matrica je kvadratna drugog reda. Broj uzoraka pozitivne klase, koje prediktivni model tačno klasifikuje, nazivamo tačnim pozitivima (eng. *true positives*, *TP*). Netačni pozitivi (eng. *false positives*, *FP*) je broj uzoraka negativne klase, koje je prediktivni model klasifikovao kao pozitivne. Uzorci negativne klase koji su tačno klasifikovani, su tačni negativi (eng*. true negatives*, *TN*). Broj netačnih negativa (eng. *false negatives*, *FN*) su pozitivni uzorci, pogrešno klasifikovani kao negativni. Ukupan broj pozitvnih uzoraka jednak je .

Tabela 7. Matrica konfuzije

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Tačna klasa | |
| Pozitivna | Negativna |
| Predviđena klasa | Pozitivna | TP | FP |
|  | Negativna | FN | TN |

Performanse klasifikacionih modela, primenom ove matrice možemo meriti na više načina (Tabela 8).

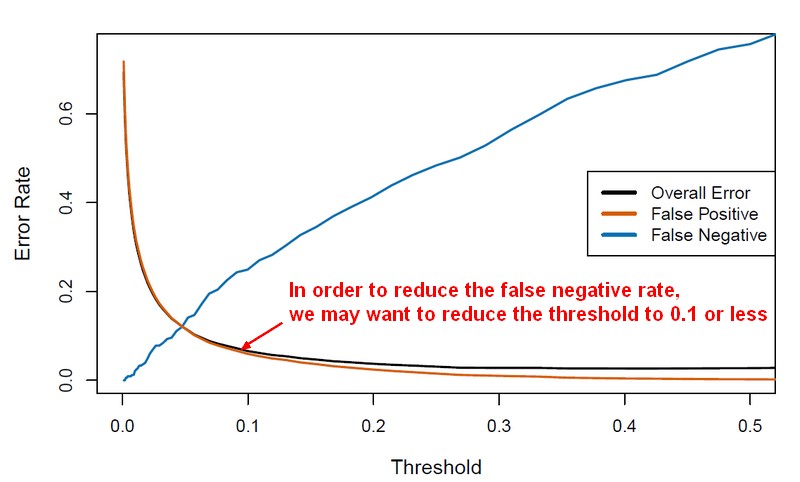
Tabela 8. Mere performansi binarnog klasifikacionog modela

|  |  |
| --- | --- |
| Mere performansi klasifikacionog modela |  |
| Tačnost (*Accuracy*) |  |
| Greška ( *Error rate*) | 1-Tačnost |
| Stopa tačnih pozitiva *TP* , (*Sensitivity*) |  |
| Stopa netačnih pozitiva, *FP* |  |
| Preciznost (*Precision*) |  |
| Specifičnost (*Specificity*) |  |

U zavisnosti od predmeta istraživanja, treba imati u vidu da pored minimiziranja ukupne greške predikcije, cilj može biti postizanje maksimalne senzitivnosti modela. U studiji slučaja, koja je predmet ove disertacije (kompanije u stečaju, pripadaju pozitvnoj klasi) prioritet je postizanje maksimalnog broja pravilno klasifikovanih uzoraka pozitivne klase (*TP*). Promenom graničnih vrednost verovatnoće, čija je podrazumevana vrednost 0.5, možemo uticati na senzitivnost prediktivnog modela. Smanjivanjem granične vrednosti, raste broj *TP* (smanjuje *FN*), a time i senzitivnost model. Istovremeno, ukupna tačnost modela se smanjuje, usled smanjenja specifičnosti, odnosno broja *TN (*raste *FP).*

Senzitivnost i Specifičnost, su metrike koje se pre svega koriste u evaluaciji prediktivnih algoritama, u tzv. nebalansiranim klasifikacionim problemima, kada je distribucija klasa asimetricna ili iskrivljena (eng. *skewness*). U podacima koji su predmet ovog istaživanja, broj uzoraka negativne klase (), značajno je veći od broja uzoraka pozitivne klase (1 ,kompanije u stečaju). Iz tog razloga, se ne koristi *Accuaracy* , kao mera tačnosti modela, jer se njime podrazumeva isti značaj obe klase. Kako smo u našem slučaju, zaintereovani za maksimalnu vrednost *TP*, odnosno tačno klasifikovanih kompanija u stečaju, preformanse ispitivanih modela *ML*, meriće se prema njihovoj Senzitivnosti.

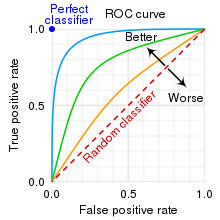
Slika 38. Smanjivanjem granične vrednosti verovatnoće, raste broj tačno klasifikovanih pozitivnih uzoraka (*TP*),odnosno samnjuje *FN*, dok se broj tačno klasifikovanih negativnih uzoraka smanjuje (TN), a time i ukupna tačnost modela (Kassambra, 2017).



Optimalnu vrednost granične verovatnoće, za koju bi stopa *TP* bila optimalna, dobijamo primenom metode *ROC* krive ( eng. *Receiver operating charactheristic* *curve*), koja predstavlja grafičku simulaciji vrednosti stopa *TP* i *FP*, za različite granične vrednosti verovatnoće (Kassambara,2017). U početnom delu *ROC* kriva je velikog nagiba, kada za minimalne vrednosti rasta *FP,* imamo značajan rast senzitivnosti modela. U svom drugom delu, dolazi do zaravnenja *ROC* krive, tako za minimalne dobitke, odnosno rast stope *TP*, imamo značajan rast *FP*. Optimalna vrednost granične verovatnoće modela je ona, za koju je zbir vrednost senzitivnosti i specifičnosti modela najveća (Kassambara,2017).

*ROC* krivu primnjujemo i za poređenje performasi različitih klasifiakcionih algoritama. Površina ograničena *ROC* krivom (eng. *Area under the curve, AUC*), ima maksimalnu vrednost jedan i predstavlja procenat pravilno klasifikovanih uzoraka (Kassambara,2017). Tako klasifikacioni model sa najvećom vrednošću *AUC*, predstavlja model najveće tačnosti.

Slika 39: ROC krive, prikazuje kako se menja stopa *TP* (senzitivnost) i stopa *FP*, za različite vrednosti granične verovatnoće[[22]](#footnote-23)



**5.PREDPROCESIRANJE I TRANSFORMACIJA PODATAKA**

U zavisnosti od vrste klasifikacionog algoritma, strukture i kavaliteta podataka, primenjuju se različite metode predprocesiranja i transformacija, kojima se podaci dovode u oblik, koji će omogućiti pouzdanu preikdciju. U prvom koraku potrebno je proveriti da li struktura i format podataka odgovara zahtevima softversog programa koji se koristi u analizi i razvoju *ML* algoritama. U ovoj doktorskoj disertaciji, primenjen je *R* statistički programski jezika, tako da su formati podataka usklađeni sa zahtevima programskih bibliotaka koje su u anlizi korišćene.

Kako podaci imaju različite jedinice mere, kao i raspone vrednosti, podaci se transformišu primenom s*caling* metoda (skaliranje), kojima se svode na isti rang vrednosti ili na istu srednju vrednost (jednaku nuli). Ovim postupkom, uticaj svake promenjive u modelu, postaje nezavisan od njene apsolutne vrednosti. Ukoliko bi se ova transformacija izostavila, *ML* algoritam, bi veći značaj u modelu ( npr. veća vrednost odgovarajućeg regresionog koeficijenta) dao promenjivoj koja ima veću vrednost. Najčešće primenjivani *scaling* postupci su:

* *Normalizacija*, kojom se numeričke vrednosti prediktora svode na isti rang, dok se oblik distribucije ne menja. Najčešće se koristi *min-max* normalizacija, kojom se vrednosti promenjive svode na rang vrednosti ,

Primenjujemo je u slučajevima nepoznate distribucije predikora ili kad ta distribucija nije *Gaussian*, u algoritmima koji su bazirani na merema rastojanja između uzoraka ( *SVM, KNN*),

* *Standardizacija* ili *z* Normalizacija, je transformacijski metod kojim se srednja vrednost promenjive svodi na 0 i standardnu devijaciju 1 .Preporučuje se u *ML* modelima u kojima se kao optimizaciski algoritam koristi *gradient descent*  (Regresioni modeli, *LDA*).

Usled postojanja *NA* (eng. *no available*), nedostajućih podataka, koji su rezultat propusta prilikom unosa podataka, greške u opermi kojom se podaci prikupljaju ili nepotpunih merenja (Bernard, 1999), efikasnost *ML* algoritma može biti umanjena, a može se prouzrokovati i algoritamska pristrasnost (eng. *algorithmic bias*). Na osnovu rapoloživih podataka, koji sadrže veći broj uzoraka sa *NA*, *ML* algoritam u postupku učenja, raspoznaje strukturu i relacije između podataka, koju generalizuje na nove podatke. Ovakva predikcija, će biti pristrasna kada novi podaci nemaju ili imaju značajno manje *NA*. U zavisnosti od toga, da li je postojanje *NA*, uslovljeno vrednostima drugih promenjivih, razlikujemo sledeće slučajeve (Allison, 2001 i Schafer iet al, 2002):

* *MCAR* (eng. *Missing clompletely at random*) , kada ne postoji zavisnost između vrednosti promenjive (koja sadrži *NA*) i ostalih promenjivih, uključujući i ,
* *MAR* ( eng. *Missing at random*), kada na pojavljivanje *NA* promenjive, utiču druge promenjive, ali ne preostale vrednosti . Tako se na osnovu vrednosti drugih prediktora, može predvideti pojavljivanje *NA*, promenjive
* *NIM* ( eng. *non ignorable missing*), kada pojavljivanje *NA* promenjive samo zavisi od preostalih vrednosti
* *MNAR* (eng. *Missing not at random*), kada postoji relacija između *NA,* promenjive i drugih promenjivih, uključujući i .

Kako u našoj studiji slučajeva, u podacim postoji značajno prisustvo *NA* ( više od 6 000), pristupilo se imputaciji nedostajućih podataka.

U slučaju *MCAR*, nedostajući podaci se mogu zameniti srednjom vrednošću te promenjive. Na ovaj način, srednja vrednost promenjive ostaje ista. Njena varijansa se međutim menja, usled promene distribucije , kao posledica veće koncentracije podataka oko srednje vrednosti. Little et al (1989) tako smatraju da je bolje uzorke sa nedostajućim vrednostima izostaviti, nego ih zameniti sa srednjom vrednosti.

U slučajevima *MAR*, *NIM* i *MNAR*, kada je pojava *NA* uslovljena vrednostima drugih promenjivih, rekonstrukciji nedostajućih podataka, najčešće se pristupa drugim tehnikama, i to pre svega metodama maksimalne izglednosti – *MaxL* i Regresije (Little i Rubin, 1987).Primenom metode *MaxL*, dolazimo do najverovatnijih vrednosti parametara distribucije promenjive koja sadrže *NA*. Kad nam je distribucija verovatnoće poznata, *NA* zemenjujemo vrednošču koja ima najveću verovatnoću pojavljivanja. Kada je količina podataka za analizu velika, za rešavanje problema *NA*, preporučuje se *MaxL* metod (Schafer i Graham, 2002). Primenom regresionig modela, zavisna promenjive, postaje prediktor sa *NA,* . Tako regresioni model, predviđe vrednosti *NA*, na osnovu poznatih podataka preostalih nezavsnih promenjivih.

U slučaju klasifikacionih problema, praktikuje se primena *Random Forest* algoritma, u rekosnstrukciji *NA*. Ovaj postupak je primenjen u studiji slučaja,ove doktorske disertacije (postupak je opisan u 3.6)

Prisustvo ekstremne vrednosti, može značajno uticati na stabilnost i tačnost prediktivnih modela i to pre svega: Linerane i Logističke regresije, *SVM*, *KNN*, *PCA*. Autlajerima (eng. *outliers*), se nazivaju ekstremne vrednosti zavisne promenjive, dok u slučaju ekstremnih vrednosti nezavisne promenjive, govorimo o podacima sa velikim uticajem (eng. *high leverage*). Uzorci sa ekstremnim vrednostima, se najčešće isključuju iz analize, ili se te vrednosti, zamenjuju svojom srednjom vrednošću (ili medijanom). Postoji više metoda, kojima se utvrđuje postojanje ekstremnih vrednosti.

* *Box and whisker* grafikon, prikazuje raspodelu podataka po kvartilima, tako se ekstremne vrednosti mogu lako uočiti. Interkvartilna razlika jednaka je,

je vrednost veća od 25% podataka te promenjive

je vrednost, veća od 75% podataka te promenjive

Vrednosti manje od smatraju se ekstremno malim, dok vrednosti veće od *,* smatraju se ekstremno velikim. Ovaj postupak je primenjen u studiji slučaja.

* U slučaju *ML* modela, koji pretpostavljaju normalnu distribuciju prediktora, postojanje ekstremnih vrednosti se može utvrditi primenom *Z- scores* metode. Vrednosti prediktrora za koje je mogu se smatrati ekstremnim,

srednja vrednost

standardna devijacija

Slika 40. Određivanje ekstremno velikih/malih vrednosti podataka primenom z- Scores[[23]](#footnote-24)

Chart, histogram

Description automatically generated

* U slučaju regresionih modela, uticajne ekstremne vrednosti, određujemo primenom *Cook* rastojanja (eng. *Cook’s distance*). Vrednost rastjanja (), pokazuje za koliko bi se promenio regresioni model, kada bi uzorak bio isključen iz modela,

, broj prediktora

, ukupan broj uzoraka

, predikcija modela, bez uzorka *i*

*MSE* = , (eng. *mean square error*)

Uobičajeno je da se vrednosti za koje je , smatraju ekstremnim (eng. *rule of thumb*).

Postojanje korelacije između prediktora, takođe negtivno utiče na regresione modele, tako što se smanjuje njihova stabilnost i tačnost predikcije. Kako parametri regresionih modela, pokazuju uticaj predikotra na vrednost zavisne promenjive, zbog postojanja korelacije sa drugim predikotorima, ovaj uticaj se ne može posmatrati izolovano. Postjanje korelacije, može se utvrditi računanjem *Pearson* koeficijenta korelacije , koji ima vrednost -1 i +1. Kada koeficijent ima vrednost nula, korelacija ne postoji. Problem multikolinearnosti ( kada se jedan prediktor može prikazati kao funkcija više drugih prediktora), može se rešiti regularizacijom regresionih modela – *Lasso* i *Ridge*, kada se smanjuje uticaj zavisnog prediktora ili se isti isključuje iz modela. Pored ovog pristupa, moguće je problem multikolinearnosti rešiti kreiranjem novih promenjivih, koje su međusobno nezavisne, a izražene su u funkciji originalnih prediktora. Ove promenjive se nazivaju principalne komponente (*PC*), a postupak Analiza principalnih komponenti - *PCA* (eng. *Principal Compnent analysis*).

* *PCA* je nenadgledani, neparametarski metod mašinskog učenja, kojim je moguće redukovati dimenzionalnost podataka, kako bi prediktivni regresioni i klasifikacioni modeli, bili funkcija manjeg broja novo formiranih promenjivih, kojima se na dovoljno dobar načina može objasniti varijabilnost zavisne promenjive. Primenom *PCA*, originalni podaci (prediktori) se linernom transformacijom, mapiraju u prostor definisan novim koordinatnim osama - Principalnim komponentama (eng. *principal components* , *PC*), koje određuju pravce maksimalnih varijacije podataka. Redukcija dimenzionalnosti se postiže tako, što se prediktivni model ‘uči’ samo na onim *PC* , koje u najvećoj meri objašnajvaju varijabilnost podataka.

*PCA*, primenjujemo u slučajevima kada zbog velikog broja prediktora, postoji mogućnost pojave *overfitinga*, te slabe mogućnost generalizacije modela. Velika dimenzionlanost podataka, čini modele kompleksnim i teškim za interpretaciju, dok je procesiranje, ‘učenje’ takvih modela, veoma zahtevno (vremenski i resursno). Lineranom *PCA* transformacijom prediktora, rešavamo i problem nestabilnosti modela, nisku tačnost, prouzrokovanu postojanjem korelacije između prediktora.

Metoda *PCA*, daje važne informacije o varijabilnosti podataka, utvrđivanjem pravaca u multidimenzionalnom prostoru, u kojima podaci imaju najveću disperziju. Pravci principalnih komponenti, u kojima je varijabilnosti podataka najveća, smatraju se najinformativnijim, najvažnijim, eng. *‘the most principal’* ( Jolliffe, 2002). Linearnu transformaciju kojom se uzorci iz originalnog *m* dimenzionalnog prostora, u kojem je svaki uzorak jednoznačno određen vektorom , mapiraju u novi prostor, određen pravcima principalnih kompnenti

(1)

.... .... .... ....

Koeficijenti ove transformacije (), koji se nazivaju i *loading* koeficijenti principalnih komponenti (Gareth et al, 2014), su jedinični sopstveni vektori (eng. *eigenvectors*), kovarijantne matrice uzoraka, kojima je srednja vrednost nula ( originalne podatke je potrebno transformisati, na način da srednja vrednost svih prediktora bude jedanka nuli).

Kovarijantna matrica je jednak,

*COV* (*X*), kovarijantna matrica podataka *X*

Dijagonalni elementi ove matrice pokazuje diperziju svakog od predikotora, njihovu varijansu,

,

Vandijagonalni elementi, predstavljaju kovarijansu između prediktora. Kada je ona pozitivna, prediktori su pozitivno linearno zavisni, kada je negativna, prediktori su negativno linearno zavisni. Sopstveni vektori (*eignevectors*) ove matrice, određuju pravce najveće disperzije podataka. Odgovarajuće sopstvene vrednosti (*eigenvalues*) predstavljaju magnitudu jediničanih vektora, odnosno intezitet dipserzije.

Ako označimo sa , sopstvene vektore kovarijantne matrice *COV(X)* i njihove sopstvene vrednosti sa , sledi,

odnosno,

(

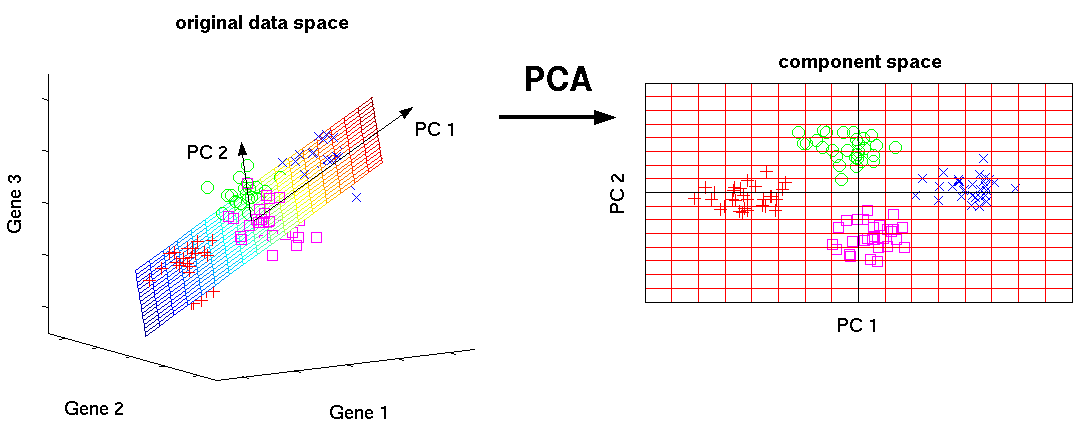
Kaklo *V* nije nula vektor, imamo da je

*I,*  jedinična matrica

Iz poslednja dva uslova dobijamo prvo vrednosti za , a potom . Ako dobijene sopstevene vrednosti uredimo kao niz opadajućeg poretka , tada predstavlja intezitet sopstvenog vektora , koji određuje pravac maksimalne disperzije uzoraka, odnonosno pravac prve principalne komponente *PC1*. Sopstvena vrednost , je intezitet sopstvenog vektora , koji određuje pravac principalne komponente *PC2* (pravac manje dispezije, od one definisanem sa ) . Pravac je pravac najmanje disperzije podataka. Jedinične vrednosti sopstvenih vektori (), predstavljaju keficijente linerne transformacije .

Kako su sopstevene vrednosti , mera varijabilnosti podataka, one se koriste i kako bi se odredio konačni broj *PC* u transformisanom prostoru, manje dimenzionalnosti (Kaiser, 1961). Tako samo one *PC,* za koje su vrednosti odgovarajućih sopstvenih vrednosti veće od jedan (to su pravci veće disperzije podataka, od disperzije u pravacima osa originalnog prostora), definišu novi prostor definisan osama .

Slika 41. Primer transformacije podataka iz trodimenzionalnog prostora (m=3, ), u dvodimenzionalan prostor određen osama (*p=2*). pravac maksimalne disperzije podataka ()[[24]](#footnote-25)



Kako je svaka *PC* upravna na prethodnu (), *PC* su međusobno nezavisne. Tako *PCA* pored smanjivanja dimenzionalnosti, eliminiše i problem multikolinearnosti prediktora, koja u regresionim modelima uzrokuje netačnu parametarizaciju modela, kao i moguće izostavljanje iz modela važnog prediktora (usled niskog statističkog značaja )(Graham,2003).

Primenom *PCA,* postižemo i smanjenje uticaja autlajera na prediktivne modele. Kako je , pravac najveće disperzije, to je ujedno i pravac sa najmanje autlajera. Svaka naredna principalna komponenta, određuje pravac manje disperzije. Tada je mogućnost pojave *outliera* veća , ali je i uticaj takve *PC* na model manji*.*

Važno je napomeut, da linernom transformacijom predikotra u *PC* i redukcijom dimenzionalnosti, iz originalnog prostora u prostor manje dimenzionalnosti , neće doći do gubitka originalnih podataka, jer je svaka *PC* linerna funkcija svih raspoloživih prediktora i njihovih vrednosti.

**6. Zaključak**

Komparativna analiza prediktivnih algoritama *ML*, koji su predmet ove doktorske disertaciji, urađena je prema njihovoj senzitivnosti (eng. *Sensitivity*), korišćenjem svih raspoloživih nezavisnih varijabli, njih ukupno 64.

Tabela 9. Postignute tačnosti, senzitivnost i specifičnost *ML* modela, sa 64+1 varijabli

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Vrsta algoritma | | Klsifikacioni algoritam | Broj atributa | Accuracy | Sensitivity | Specificity |
| Regresioni | Logistička regresija |  |  |  |  |  |
|  |  | Log regresija (*model\_lr*) | 64 | 0. 7815 | 0. 6255 | 0. 8267 |
|  | Log regresija (*model\_lr1*) | 54 | 0.7805 | 0.6255 | 0.8255 |
|  |  | Log regresija (*model\_lrPC*) | 40 | 0.789 | 0.6151 | 0.8394 |
|  | *Penalized* logistička regresija |  |  |  |  |  |
|  |  | Lasso (*model\_lrlasso*) | 54 | 0.789 | 0.6297 | 0.8352 |
|  | Ridge (*model\_lrridge*) | 64 | 0.7876 | 0.6255 | 0.8345 |
|  | Ealstic Net(*model\_elasnet*) | 56 | 0.7881 | 0.6297 | 0.8339 |
|  | Diskriminantna analiza |  |  |  |  |  |
|  |  | LDA (*model\_lda*) | 54 | 0.781 | 0.5774 | 0.840 |
|  | QDA (*model\_qda*) | 54 | 0.7439 | 0.4937 | 0.8164 |
| Modeli na bazi stabla odlučivanja |  | | | | | |
|  | Decision tree (*model\_tree1*) |  |  | 0.851 | 0.6464 | 0.9103 |
|  | RF (*model\_rf*) |  |  | 0.8553 | 0.8577 | 0.8545 |
|  | GTB (*model\_xgboost*) |  |  | 0.8938 | 0.8117 | 0.9176 |
| NB |  |  |  |  |  |  |
|  | NB (*model\_nb*) |  |  | 0.6734 | 0.7385 | 0.6545 |
| NB with PC (*model\_nbpc*) |  |  | 0.734 | 0.5126 | 0.7982 |
| KNN |  | | | | | |
|  | KNN, Acc (*model\_knn1*) |  |  | 0.7688 | 0.50 | 0.8467 |
|  | KNN, ROC (*model\_knn2*) |  |  | 0.7707 | 0.5795 | 0.8261 |
| SVM |  |  |  |  |  |  |
|  | SVM, Lin (model\_svmlin) |  |  | 0.7824 | 0.5335 | 0.8545 |
|  | SVM, Radial (model\_svmradial) |  |  | 0.8177 | 0.6276 | 0.8727 |
|  | SVM, Poli (model\_svmpoli) |  |  | 0.8158 | 0.6130 | 0.8745 |

\* Primenjen je postupak *cross* *valitation* e (*CV* ), sa 5 skupova (eng. *folds*), brojem ponvaljanja 3, za određivanje optimalne vrednosti hiperparametara. Granična vrednost verovatnoće pozitivnog događaja (*y=1*), je 0.3.

**7.1 REGRESIONI MODELI**

Algoritam logističke regresije (*model\_lr*), model sa svih 64 prediktora, ima tačnost *Accuracy* = 0.7815, *Sensitivity* = 0.6255 i *Specificity* = 0.8267. Eliminacijom prediktora, čiji regresioni koeficijenti nisu statistički značajni (p > 0.05), dobijen je model logističke regresije sa 54 prediktora (*model\_lr1)*. Iako je porođenjem vrednosti *AIC* (Akaike informacionog kriterijuma), pokazano da *model\_lr*, bolje opisuje realacije između varijabli, veća tačnost predikcije ovog modela je samo neznatno bolja od *model\_lr1*, koji ja mnogo manje kompleksan. Iz tog razloga, njega je lakše tumačiti, a ujedno se smanjuje mogućnost overfitinga.

Pokazano je i da se transformacijom varijabli u *PC*, dimenzionalnost podataka može smanjiti. Model logističke regresije *model\_lrPC* , ima 40 međusobno nezavisnih *PC*, sa kojima se može objasniti 96% kumulativne varijabilnosti prediktora. Ovim modelom je postignuta veća tačnost (ali ne i senzitivnost) od prethodna dva (*model\_lr i model\_lr1*). Potvrđeno je takođe, da se eliminacijom problema multikolinearnosti, malo utiče na tačnost klasifikacionih algoritama, a više ne vrednosti regresionih koeficijenata (Bruce, 2017).

Primenom metoda regularizacije u modelima logističke regresije, pokazano je da se dodatno povećava njihova tačnost. *Lasso* regularizacijom (*model\_lrlasso*) primenom *L1 penalty term*-a , iz modela je eliminisano deset varijabli, svođenjem odgovarajućih regresionih koeficijenata na vrednost nule, postignuta je najveća tačnost i senzitivnost modela logističke regresije. Vidimo da je *Ridge* regresijom (*model\_lrridge)*, primenom *L2 penalty term*–a, postignuta manja tačnost, iako model ima svih 64 prediktora. *Elastic Net (model\_elasnet* ), metod koji kombinuje  *L1* i *L2 penalty term,* ima najmanje kompleksan model, sa samo neznatno manjom tačnošću u odnosu na *Lasso* regresiju.

Algoritmi *LDA* i *QDA*, koji se pre svega primenjuju u slučaju klasifikacionim problemima, u kojima zavisna promenjiva ima više od dve klase (kategorije), imaju najmanju tačnost (Hastie et al, 2009).

Zaključujemo da u binarnim klasifikacionim problemim, sa velikim brojem međuzavisnih prediktora, sa *Lasso* regresijom se postiže najtačnija predikcija i senzitivnost, uz značajno smanjenje kompleksnosti modela. Rezultati su takođe potvrdili, da kada se vrednosti regresionih koeficijenata značajno razlikuju, bolje preformanse se postižu *Lasso*, u odnosu na druge metode regularizacije (Kassambara, 2017). Očekivano, pet najznačajnijih (informativnijih) prediktora identično je za *Lasso* i *Elastic* *Net* modele, dok je Ridge regresijom dobijeno da je pet najinformativnijih prediktora . U odnosu na *Lasso* i *Elastic* *Net*, prediktor zamenjen je sa *.*

Kako osnovne dve pretpostavka regresionih modela, da postoji linearna zavisnost prediktora u modelu sa log i da nema multikolinearnosti, nisu ispunjene, tačnost ovih modela je relativno niska. Regularizacijom je postignuto povećanje tačnosti, uz smanjenje kompleksnosti modela, čime je omogućeno njegovo lakše tumačenje i bolja generalizacija.

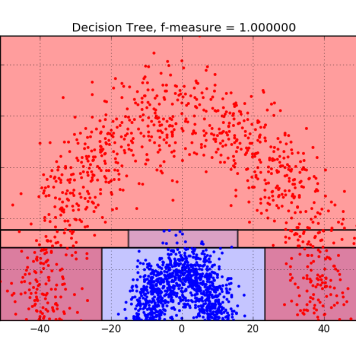
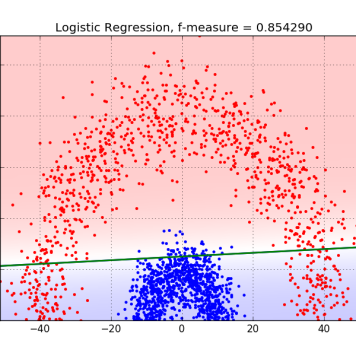
Tabela 10. Pet najznačajnij prediktori *Lasso* i ElasticNet regresije

|  |  |  |
| --- | --- | --- |
| ML Model | Najznačajnij prediktori | Opis |
| Log regresija(*model\_lrlasso \_lr i model\_elasnet*) | X46 | (current assets - inventory) / short-term liabilities |
| X63 | sales / short-term liabilities |
| X38 | constant capital / total assets |
| X24 | gross profit (in 3 years) / total assets |
| X58 | total costs /total sales |

**7.2 MODELI NA BAZI STABLA ODLUČIVANJA**

Algoritmi na bazi stabla odlučivanja, postigli su ukupno najveću tačnost. Očekivano u odnosu na regresione algoritme, imajući u vidu da se ovim modelima na pretpostavlja linearna relacija varijabli sa transformisanom zavisnom promenjivom, kao i da na njih ne utiče postojanje multikolinearnosti. Rezultati su takođe pokazali, da u slučaju velike dimenzionalnosti podataka, ovi algoritmi postižu veču tačnost u odnosu na regresione (Zhang, 2016). Isto tako, kada se uzorci različitih klasa ne mogu razdvojiti lineranom graničnom linijom (ili sa hiperrvani u višedimenzionalnom prostoru), algoritmi na bazi stabla prepoznaju složene šablone između podataka i mogu razdvojiti podatke različitih klasa u više sekcija (Slika 42). Rezultati studije slučaja su pokazali, da je ovim modelima postignuta veća tačnost od regresionih, u proseku za 10%.

Slika 42. Razgraničavanje podataka različitih klasa u slučaju stabla odlučivanja i logističke regresije [[25]](#footnote-26)

Diagram

Description automatically generated

Iz dobijenih rezultata vidimo da *RF (model\_rf)*, sa kojim se razvija više stabala, sa različitim grupama slučajno izabranih prediktora, postiže veću tačnost u odnosu na *DT(model\_tree1)* . Poznato je takođe, da je *DT* zbog postojanja samo jednog stabla, kreiranog primenom svih raspoloživih prediktora, sklon overfiting. Iz tog razloga *DT* , ima lošiju mogućnost generalizacije u odnosu na *RF* (Cutler, 2007).

Međutim, za razliku of *RF*, kojeg je zbog velikog broja stabala (u našem slučaju 500), praktično nemoguće interpretirati, *DT* model se jednostavno tumači, što je i njegova osnovna prednost. Studija slčaja je dalje pokazala, da je primenom *RF*, postignuta značajno veća senzitivnost modela, čak za 36% u odnosu na *DT.* Ovo jei i ujedno najveća postignuta senzitivnost od svih modela *ML*, koji su predmet ove doktorske disertacije.

Ekstremni *boosting* algoritam *GTB (model\_xgboost)* , takođe razvija više stabala, ali za razliku od *RF*, stabla su zavisna. Odnosno, svako novo stablo se uči na grešci u predikciji, načinjenom prethodnim. Kako ovaj algoritam, kombinuje pri odlučivanju jednostavna *stump* stabla (eng. *weak learners*), ima visok *bias* i nisku varijabilnost, zbog čega je generalzacija modela bolja nego u slučaju *RF*. Iz ovih razloga, *GTB* postiže veću tačnosti od *RF* i ukupno najveću tačnost i specifičnosti u odnosu na ispitivane modele. Zbog mogućnosti paralelnog procesiranja, *GTB* je potrebno manje vremena za ’učenje’ modela od *RF*. Intersantno je da je sa *RF* postignuta veća senzitivnost modela od *GB*.

Primećujemo da su tri najznačajnija prediktora modela ista za sva tri algoritma- *DT, RF i GTB.*

Tabel 11. Pet najznačajnijih prediktori *DT, RF i GB* modela

|  |  |  |
| --- | --- | --- |
| ML Model | Najznačajnij prediktori | Opis |
| DT (model\_tree1) | X27 | profit on operating activities / financial expenses |
| X46 | (current assets - inventory) / short-term liabilities |
| X34 | operating expenses / total liabilities |
| X4 | current assets / short-term liabilities |
| X26 | (net profit + depreciation) / total liabilities |
| RF (model\_rf) | X27 | (current assets - inventory) / short-term liabilities |
| X46 | sales / short-term liabilities |
| X34 | operating expenses / total liabilities |
| X44 | (receivables \* 365) / sales |
| X6 | retained earnings / total assets |
| GTB (*model\_xgboost*) | X27 | (current assets - inventory) / short-term liabilities |
| X46 | sales / short-term liabilities |
| X34 | operating expenses / total liabilities |
| X24 | gross profit (in 3 years) / total assets |
| X9 | sales / total assets |

Samo se X46 (eng. *current assets - inventory) / short-term liabilities*) i X24 (eng. *gross profit (in 3 years) / total assets*) , pojavljuju se među pet najznačajnijih prediktora i u *GTB* i u Lasso regresiji.

**7.3 KNN**

*KNN* koji spada u jednostavnije algoritme, bolju tačnost postižu sa podacima manje dimenzionalnosti i sa brojem uzoraka ne većim od 50,000 (Altman, 1992). Osetljiv je na prisustvo ekstremnih vrednosti, tako da ih je potrebno eliminisati, ili kako je urađeno u našoj studiji slučaja, zameniti odgovarajućim središnjim vrednostima. *KNN* algoritmom se ne mogu dobiti informacije o pojedinačnom uticaju prediktora. Model nije moguće tumačiti, već isključivo koristiti u svrhu predikcije. Testirana su dava modela. Prvi ( *model\_knn1*) u kojem je broj suseda K, odeređen na bazi tačanosti modela (K =7) i drugi (*model\_knn2 )*na osnovu vrednosti *ROC (dobijena vrednost K=19)*.

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Primenom *KNN* dobili smo rezultat najmanje tačnosti i senzitivnosti. Kako ovaj algoritam nije dobar izbora kada postoji velika dimenzionalnost podataka i kako je vrlo ostljiv na ekstremne vrednosti, ovi rezultati su očekivani ( Hastie, 2008).

|  |  |  |
| --- | --- | --- |
| ML Model | Najznačajnij prediktori | Opis |
| K-NN (*model\_knn1 i model\_knn2*) | X46 | (current assets - inventory) / short-term liabilities |
| X16 | (gross profit + depreciation) / total liabilities |
| X26 | (net profit + depreciation) / total liabilities |
| X24 | gross profit (in 3 years) / total assets |
| X17 | total assets / total liabilities |

**7.4 SVM**

*SVM* modeli se najviše primenjuju za analize podataka koji su nestrukturirani. Mogu se koristiti i u slučaju klasifikacionih problema sa strukturiranim podacima, ali je potrebno imati u vidu, da se vreme učenja ovih algoritama značajno povećava, rastom količine podataka i njihove dimenzionalnosti (Vapnik i Cortes, 1995). U studiji slučaja, *SVM* sa plinominalnim kernelom, je zahtevao neuporedivo više vremena, u odnosu na sve presotale algoritme. Očekivano *SVM* sa linearnim kernelom, ima najmanju tačnost, iz razloga da se uzorci različitih klasa ne mogu razgraničiti linearnom hiperavni. *SVM* sa Radail kernel funkcijom, daje najveću tačnost od *SVM* modela, a približno isti rezultat postignut je primenom *SVM Poli* algoritma. U skladu sa teorijom, u slučaju nebalansiranih podatakaka, senzitivnost *SVM* modela je dosta niska ( Baruwita et al., 2013).

Tabel 12. Pet najznačajnijih prediktora *SVM linear, SVM radial i SVM poli* algoritama

|  |  |  |
| --- | --- | --- |
| ML Model | Najznačajnij prediktori | Opis |
| SVM Linear (*model\_svmlin*) | X46 | (current assets - inventory) / short-term liabilities |
| X16 | (gross profit + depreciation) / total liabilities |
| X26 | (net profit + depreciation) / total liabilities |
| X24 | gross profit (in 3 years) / total assets |
| X17 | total assets / total liabilities |
| SVM Radial *(model\_svmradial*) | X46 | (current assets - inventory) / short-term liabilities |
| X16 | (gross profit + depreciation) / total liabilities |
| X26 | (net profit + depreciation) / total liabilities |
| X24 | gross profit (in 3 years) / total assets |
| X17 | total assets / total liabilities |
| SVM Poli *(model\_svmpoli*) | X27 | (current assets - inventory) / short-term liabilities |
| X46 | sales / short-term liabilities |
| X34 | operating expenses / total liabilities |
| X24 | gross profit (in 3 years) / total assets |
| X5 | [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] \* 365 |

Iz dobijenih rezulta, zaključujemo da od analizirnih modela, algoritmi na bazi stabla odlučivanja, postižu najveću tačnost predikcije. Sa *Random Forest* postignuta je najveća senzitivnost. Odnosno, zaključujemo da su primenom diskriminativnih modela *ML* , postignute predikcije najveće senzitivnosti i ukupne tačnosti.

*Gradine boost*, ima najeveću tačnost i specifičnost, a neznatno nižu *Random Forest* i *Decission Tree*

Očekivano najmanju tačnost postigao je *Naive Bayes*, zbog ne ispunjenosti osnovne pretpostavke ovog modela, da su prediktori date klase međusobno nezavisni.

**7.5 UPOREDNA ANALIZA PERFORMANSI MODELA *ML* i Altman *Z- score***

Na kreiranom podskupu podataka, sa samo pet nezavisnih promenjivih, koje odgovaraju Altman *Z- score* modelu ( X3,X6,X7,X8 i X9 ), proverena je tačnost predikcije *ML* algoritama u odnosu na Altman *Z- score* model. Dobijeni rezultati potvrđuju da se primenom *ML* u ekonomiji, može povećati tačnost predikcije.

Tabela 13. Performanse *Altman z score* i *Gradient boosting* modela

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Sensitivity | Specificity |
|  |  |  |  |
| Z score | 0. 6964 | 0. 3746 | 0. 8443 |
| Z score emerging | 0.7758 | 0.51 | 0.783 |
| RF | 0.8553 | 0.8577 | 0.8545 |
| GTB | 0.8938 | 0.8117 | 0.9176 |

Primenom *point biserial* metode, dobijeni su koificijenti korelacije, kao mere uticaja prediktora na vrednost kategoričke zavisne promenjive. Isto tako, do uticaja prediktora došli smo i kako je već pokazano, primenom *ML* algoritama, onih koji imaju svojstvo da daju značaj svakog prediktora u modelu.

Tabel 14. Prediktori , uređeni po opadajućoj vrednosti uticaja na vrednost zavisne promenjive

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Promenjiva | Z score emerging | GB | RF | Lasso | Point biserial |
| X7max uticaj) |  |  |  |  |  |
| X3 |  |  |  |  |  |
| X6 |  |  |  |  |  |
| X8 |  |  |  |  |  |
|  |  |  |  |  |  |

Zaključujemo de se ekonomske metode i modele, treba primenjivati u rešavanju problema (Mullainathan, 2017), kojima se određuje priroda i intezitet uticaja finansijskih indikatora na neku ekonomsku pojavu, iz razloga nekonzistentnih podataka o značaju prediktora dobijenog *ML* algoritmima najveće tačnosti, kao i statističkog metoda *Point biserial*.

**X46 najvazniji u svim modelima**

**X24, u lin,tree, svm i knn**

**X26, najvaznijei**

**Granicna verovatnoca 0.3 za sve**

Chart, scatter chart

Description automatically generated

**8. LITERATURA**

1. Abdi, H. (2010). *Principal Compnent Analysis* i John Wiley & Sons, Inc.
2. Advait, J. (2020). *Naïve Bayes Classifier Advanced concept*. Technical Publications. Safari, O'Reilly Media Company.
3. Barnard, J. i X.L. Meng (1999). *Applications of multiple imputation in medical studies: From AIDS to NHANES.*Stat. Methods.
4. Batuwita, R. i Palade, V. (2013). *Class Imbalance Learning Methods for Support Vector Machines*. Wiley on line library.
5. Breiman, L. (1994). [*Bagging*](https://www.stat.berkeley.edu/~breiman/bagging.pdf) *Predictors.*  Department of Statistics, University of California Berkeley (421). Tekst dostupan na internetu: <https://www.stat.berkeley.edu/~breiman/bagging.pdf>.
6. Butte, G.  *Butter’s Law of Photonics* . Tekst dostupan na: <https://roundtables.abl.org/four-laws-enabling-telemedicine-better-faster-cheaper/> [pristupljeno: 3. Jun 2022].
7. Cutler, R. et al. (2007). .*Random Forests for Classification and Regression*. Ecology- Ecological Society of America, 88(11), s.2783- 2792.
8. Friedman, J. (2001). *A Gradient Boosting Machin.* The Annals of Statistics, 29(5), s. 1189-1232*.*
9. Friedman, J. (2001). *Greedy function approximation: A gradient boosting machine*. Ann. Statist,29(5), s. 1189-1232.
10. Gaber, T. et al. (2017). *Liner discriminant Analysis*. Manchester, University of Salford.
11. Gantz, J. i Reinsel, D. (2012). *The Digital Universe in 2020: Big data, Bigger Digital Shadows, and Biggest Growth in the Far East*. IDC., sponsored by EMC.
12. Graham, M. (2003). *Confronting muliticolinearity in ecological multiple regression*. Ecology, 84(11), s. 2809–2815.
13. Hastie, T. et al. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Drugo izdanje.Luxemburg, Springer Science & Business Media.
14. Hoang, D. i Wiegratz, K. (2021). *Machine Learning Methods in Finance: Recent Applications and Prospects*. Tekst dostupan na internetu<https://finance.fbv.kit.edu/rd_download/Machine%20Learning%20Methods%20in%20Finance.pdf> . [pristupljeno: 20. Jun 2022].
15. Huang, A.H. et.al. (2014). *Evidence on the Information Content of Text in Analyst Reports*. The Accounting Review, 89, s. 2151–2180.
16. Jolliffe , I.T. (2002). *Principal Compnent Analysis*. Druga edicija. New York,Springer - Verlag.
17. Kaufman, I. i Horton, C. (2015) . *Digital Transformation: Leveraging Digital Technology with Core Values to Achieve Sustainable Business Goals*. The European Financial Review (December–January), s. 63–67.
18. Kryder, M. (2005). *Kryder’s Law*. Tekst dostupan na:<https://www.techtarget.com/searchstorage/definition/Kryders-Law#:~:text=Kryder%27s%20Law%20is%20the%20assumption,improves%2C%20storage%20will%20become%20cheaper.> [pristupljeno: 3. Jun 2022].
19. Lanzolla, G. i Anderson, J. (2008). *Digital Transformation*. Business Strategy Review, 19(2), s.72-76.
20. L´eon Bottou (2010). *Large-Scale Machine Learning with Stochastic Gradient Descent*. Princeton, NJ, NEC Labs America.
21. Little, R. i Rubin, D. (1989). *The analysis of social science data with missing values*. Sociological Method and Research, SAGE Journals,18(2-3).
22. McCullagh, P. i Nelder, J.A. (1983).  *Generalized Liner Models*. Monographs on Statistics and Applied Probability, London, Chapman and Hall.
23. Mitchell, T. (1997), *Machine Learning*, McGraw Hill.
24. Mullainathan, S. i Spiess, J. (2017). *Machine Learning: An Applied Econometric Approach*. Journal of Economic Perspectives, 31, s. 87–106.
25. Moore, G. (1975). *Moore’s Law* Tekst dostupan na <https://en.wikipedia.org/wiki/Moore%27s_law> [pristupljeno: 3. Jun 2022].
26. Porter, D. i Gujarati, D. (2009). *Basic Econometrics.* NY*,*The McGraw-Hill Series Economics.
27. Provost, F. I Fawcett, T. (2013).*Data Science for Business*. New York, O’Reilly Media Inc.
28. Roger, D. (2016).*The Digital transformation playbook.* Columbia Business School.
29. Samuel, A. (1959). *Some studies in Machine learning using the game of checkers*. IBM.
30. Schafer, JL. i Graham, JW. (2002). *Missing data: Our view of the state of the art*. Psychol Methods*, 7(2), s.*147–177. US, National Library of Medicine.
31. Shihao, G. et al. (2019). *Empirical Asset Pricing via Machine Learning.* USA, University of Chicago
32. Shwartz, S. i Shai Ben David (2014). *Understandin Machine Learning : From Theory to Algorithms*. Cambridge University Press.
33. Singh, G. (2019). *Comparison between Multinomial and Bernoulli for Naive Bayes for Text Classification.* London, IEEE.
34. Sugiyama, M. (2016). Introduction to Statistical Machine Learning. Prva edicija. USA, MK.
35. Vapnik, V. I Cortes, C. (1995).*Support vector networks.* Springer.
36. Vapni, V. (1995). *The Nature of Statistical Learning Theory*. New York,Springer.
37. Varian, H.R. (2014). *Big Data: New Tricks for Econometrics*. Journal of Economic Perspectives ,28, s. 3–28.
38. Winston, P. (2010). *SVM.* Massachusetts Institute of Technology .Tekst dostupan na <https://ocw.mit.edu/courses/6-034-artificial-intelligence-fall-2010/> [pristupljeno: 3. Jun 2022].
39. Winston, P. (2010).*Boosting*, Lecture 17. Massachusetts Institute of Technology .Tekst dostupan na <http://ocw.mit.edu/6-034F10> [pristupljeno: 3. Jul 2022].
40. Winston, P. *Support Vector machines.* MIT. Dostupno na internetu <https://www.youtube.com/watch?v=_PwhiWxHK8o&t=1779s>. [pristupljeno: 3. Jul 2022].

**Prilog A.1 Opis podataka**

*UCI Machine Learning Repository[[26]](#footnote-27)* (osnivač *Universiy of California, Irvine*) je kolekcija baza i generatora podataka, koji se koriste za empirijske analize algoritama *ML*. Od osnivanja 1987, *UCI* predstavlja osnovni izvor podataka za studente, predavače, istraživače, koji ih primenjuju u *ML*. Citiran je kao izvor podataka preko 1000 puta, što prema broju citat spada u top 100 radova iz oblasti *ICT*.

Originalan naziv seta podataka koji se u ovoj dizertaciji korisiti je ‘*’Polish companies bankruptcy Data Set*’’[[27]](#footnote-28). Autor je Sebastian Tomczak, *Department of Operations Research, Wroclaw University of Science and Technology*, koji je kao izvor koristio *EMIS* - *Emerging Markets Information Service[[28]](#footnote-29).* EMIS sakuplja i analizira podatke sa preko 125 tržišta (eng. *emerging markets*)

Podaci predstavljaju skup finansijskih indikatora poslovanja poljskih kompanija, koje su otišle u stečaj u periodu od 2000-2012 godine, kao i onih koje su nastavile nesmetano da posluju do 2013. Podaci su strukturirani, numerički. Istraživanjem je obuhvaćeno 64 različitih finansijskih indikatora, koji utiču na verovatnoću stečaja kompanija. Zavisna promenjiva je binarna kategorička, sa vrednostima 0 (za kompanije koje nisu u stečaju) i 1 (za kompnije u stečaju). Primenjena je sledeca notacija, kojom se podaci predstavljaju u matričnom obliku:

Prediktori, ,

Zavisna (target) promenjiva, ,

|  |  |
| --- | --- |
| **Prediktori** | **Značenje** |
| X1 | net profit / total assets |
| X2 | total liabilities / total assets |
| X3 | working capital / total assets |
| X4 | current assets / short-term liabilities |
| X5 | [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] \* 365 |
| X6 | retained earnings / total assets |
| X7 | EBIT / total assets |
| X8 | book value of equity / total liabilities |
| X9 | sales / total assets |
| X10 | equity / total assets |
| X11 | (gross profit + extraordinary items + financial expenses) / total assets |
| X12 | gross profit / short-term liabilities |
| X13 | (gross profit + depreciation) / sales |
| X14 | (gross profit + interest) / total assets |
| X15 | (total liabilities \* 365) / (gross profit + depreciation) |
| X16 | (gross profit + depreciation) / total liabilities |
| X17 | total assets / total liabilities |
| X18 | gross profit / total assets |
| X19 | gross profit / sales |
| X20 | (inventory \* 365) / sales |
| X21 | sales (n) / sales (n-1) |
| X22 | profit on operating activities / total assets |
| X23 | net profit / sales |
| X24 | gross profit (in 3 years) / total assets |
| X25 | (equity - share capital) / total assets |
| X26 | (net profit + depreciation) / total liabilities |
| X27 | profit on operating activities / financial expenses |
| X28 | working capital / fixed assets |
| X29 | logarithm of total assets |
| X30 | (total liabilities - cash) / sales |
| X31 | (gross profit + interest) / sales |
| X32 | (current liabilities \* 365) / cost of products sold |
| X33 | operating expenses / short-term liabilities |
| X34 | operating expenses / total liabilities |
| X35 | profit on sales / total assets |
| X36 | total sales / total assets |
| X37 | (current assets - inventories) / long-term liabilities |
| X38 | constant capital / total assets |
| X39 | profit on sales / sales |
| X40 | (current assets - inventory - receivables) / short-term liabilities |
| X41 | total liabilities / ((profit on operating activities + depreciation) \* (12/365)) |
| X42 | profit on operating activities / sales |
| X43 | rotation receivables + inventory turnover in days |
| X44 | (receivables \* 365) / sales |
| X45 | net profit / inventory |
| X46 | (current assets - inventory) / short-term liabilities |
| X47 | (inventory \* 365) / cost of products sold |
| X48 | EBITDA (profit on operating activities - depreciation) / total assets |
| X49 | EBITDA (profit on operating activities - depreciation) / sales |
| X50 | current assets / total liabilities |
| X51 | short-term liabilities / total assets |
| X52 | (short-term liabilities \* 365) / cost of products sold) |
| X53 | equity / fixed assets |
| X54 | constant capital / fixed assets |
| X55 | working capital |
| X56 | (sales - cost of products sold) / sales |
| X57 | (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation) |
| X58 | total costs /total sales |
| X59 | long-term liabilities / equity |
| X60 | sales / inventory |
| X61 | sales / receivables |
| X62 | (short-term liabilities \*365) / sales |
| X63 | sales / short-term liabilities |
| X64 | sales / fixed assets |
| **Target promenjiva** | **Značenje** |
| Class | 0 kompanija nije u stečaju, 1 kompanija u stečaju |

\* Imena varijabli originalnih podataka su Attr1....Attr64 i Class

**A 1.1 Priprema podataka**

Primenom metoda deskriptivne statistike, opisani su podaci studije slučaja. Pored mera centralne tendencije i varijabilnosti, analizirana je povezanosti, odnosno korelacija varijabli. Ovaj postupak je neophodan, kako bi se moglo sprovesti statističko zaključivanje, potrebno za primenu odabranih *ML* algoritama.

Izračunate su srednje i središnje vrednosti (medijana), kvartili Q1 i Q3, kao i vrednosti *skewness* i *kurtosis* za sve numeričke varijable. Kako samo X29, ima vrednosti *skewness* između -2 i 2, a *kurtosis* -7 and 7, može se smatrati da je to jedina varijabla koja ima približno normalnu distribuciju (Bryne, 2010).

Slika 11. Histogram promenjive X29[[29]](#footnote-30)

Chart, histogram

Description automatically generated

Staistički značajna (*p-values* < 0.05) korelacija, postoji između 25 parova varijabli. Parovi sa najvećom korelacijom (njih 10), dati su su dati u Tabeli 1. Ovo ukazuje na značajno postojanje multikolinearnosti.

Slika 12. Parovi varijabli sa pozitivnom (plava boja) i negativnom korelacijom[[30]](#footnote-31)

Chart

Description automatically generated

Tabela 1. Opis varijabliablie sa najvećom korelacijom

|  |  |
| --- | --- |
| Pozitivna korelacija | |
| Parovi varijabli | Opis |
| X39 & X56 | Profit on sale/sales & sales-cost of product sold/sales |
| X8 & X17 | book values eq/TL & TA/TL |
| X19 & X23 | gross profit/sales & net profit/sales |
| X45 & X60 | net profit/INV & sales/INV |
| X16 & X26 | (gross profit + depreciation) / TA & (net profit + depreciation) /TL |
| X22 & X35 | (profit on operating activities)/TA & profit on sales / total assets |
| X22 & X48 | (profit on operating activities / TL & EBITDA/TA |
| X35 & X48 | Profit sales/TA & EBITDA/TA |
| X33 & X63 | Operating exp/short term liab & sales/short term liab |

|  |  |
| --- | --- |
| Negativna korelacija | |
| Parovi varijabli | Opis |
| Attr3 & Attr51 | Working capital/ TA & short term liab/ TA |

TA - *total assets*

TL- *total liabilities*

Liab – *liabilities*

Slika 13. Prikaz korelacija svih 64 varijabli[[31]](#footnote-32)

Chart, scatter chart

Description automatically generated

Nedostajućih podataka (*NA*) ima ukupno 6,565, što predstavlja % 1.4 ukupnih podataka. Broj kompletnih uzoraka, bez nedostajućih vrednosti, je 3,254. Priroda pojavljivanja *NA,* je *MAR* ( eng. *Missing at random*). Iz tog razloga, *NA* nisu zamenjene srednjom (ili središnjom) vrednošću, vać se ona predviđaju primenom *RF* algoritma. Najviše nedostajućih vrednosti, imaju varijable X37, X45 i X28.

Slika 22. Mapa nedostajućih vrednosti[[32]](#footnote-33)

Chart, treemap chart

Description automatically generated

Ekstremnim vrednostima, smatramo one koje se nalaze van granica i . Ove vrednosti, zamenjene su odgovarajućim medijanama.

Kako je varijabla X55 (eng. *working capital* ) data u apsolutnoj vrednosti, a preostale varijable, finansijski indikatori, u realtivnom iznosu, postupkom min - max normalizacije, podaci su svedeni na isti rang vrednosti [0,1].

Usled postojanja multikolinearnosti, podaci su transformisani u *PC*, čime je ujedno omogućeno smanjivanje dimenzionalnosti modela. U algoritmima koji su osetljivi na postojanje linearne zavisnosti varijabli, kao u slučaju logističke regresije i *NB,* *ML* modeli su razvijeni na podacima oblika *PC*.

Slika 22. Korelacije posle transformacije verijabli u principalne komponente[[33]](#footnote-34)

Chart, histogram

Description automatically generated

Podaci su podeljeni u trening i test podatke u odnosu 70:30.

**Prilog A.2 Prikaz razvijenih softverskih algoritama**

(kod dostupan na [kkolaro/doktorat (github.com)](https://github.com/kkolaro/doktorat))

ml\_final.R (sa rezultatima)

Klod Kolaro

2022-12-03

# ANALIZA PERFORMANSI KLASIFIKACIONIH ALGORITAMA  
  
library(foreign)  
library(mice)

library(mda)# fda regresiju

library(randomForest)

library(caret)#train

library(RWeka)

library(psych)

library(lares)

library(xgboost)  
library(tibble)  
library(Amelia)

library(rpart)  
library(factoextra)

library(FactoMineR)  
library(devtools)

library(corrplot)

library(ROCR)

library(magrittr)  
library(dplyr)# %>%

library(Matrix)  
library(class)  
library(tidyr)

library(leaps)  
library(glmnet)

library(e1071) # svm

library(rpart.plot)

library(psych)  
library(stats)  
library(MASS)

library(AICcmodavg)

library(car)

tek\_seting <- getOption("warn")  
  
options(warn = -1)

# 1. FORMIRANJE DATA SETA

# Import podataka  
  
stecaj\_1<-read.arff(choose.files())# RWeka, file 5  
stecaj\_2<-read.arff(choose.files())# file 1  
stecaj\_3<-read.arff(choose.files())# file 2  
stecaj\_4<-read.arff(choose.files())# file 4, stecaj posle 2 godine  
  
# Kreiranje data seta. Binarna promenjiva class = (0,1). Za kompanije u stecaju class= 1  
  
samo\_stecaj2<-stecaj\_2[stecaj\_2$class==1,]  
samo\_stecaj3<-stecaj\_3[stecaj\_3$class==1,]  
samo\_stecaj4<-stecaj\_4[stecaj\_4$class==1,]  
  
stecaj<-rbind(stecaj\_1,samo\_stecaj2,samo\_stecaj3,samo\_stecaj4)  
  
  
# Random preraspodela uzoraka  
  
set.seed(1)  
  
stecaj<-stecaj[sample(1:nrow(stecaj)),]  
  
# Promena imena kolone zavisne binarne promenjive 'class', u 'Stecaj'  
  
colnames(stecaj)[65]<-'Stecaj'  
  
# Provera balansiranosti podataka  
  
round (prop.table( as.table(table(stecaj$Stecaj))),3)

##   
## 0 1   
## 0.775 0.225

# 2. OPIS PODATAKA  
  
# Descriptivna statistika data seta.  
# Mean, Median, Q1, Q3, Min, Max, prisutvo NA's  
  
summary(stecaj)

## Attr1 Attr2 Attr3 Attr4   
## Min. :-463.8900 Min. :-430.8700 Min. :-479.9600 Min. : -0.403   
## 1st Qu.: 0.0003 1st Qu.: 0.2755 1st Qu.: 0.0145 1st Qu.: 1.030   
## Median : 0.0395 Median : 0.4865 Median : 0.1931 Median : 1.532   
## Mean : -0.0328 Mean : 0.6458 Mean : 0.0254 Mean : 4.677   
## 3rd Qu.: 0.1107 3rd Qu.: 0.7006 3rd Qu.: 0.4002 3rd Qu.: 2.725   
## Max. : 87.4590 Max. : 480.9600 Max. : 28.3360 Max. :6845.800   
## NA's :3 NA's :3 NA's :3 NA's :23   
## Attr5 Attr6 Attr7   
## Min. :-1076400.0 Min. :-508.4100 Min. :-517.4800   
## 1st Qu.: -52.1 1st Qu.: -0.0002 1st Qu.: 0.0014   
## Median : -5.5 Median : 0.0000 Median : 0.0466   
## Mean : -39.2 Mean : -0.1447 Mean : -0.1058   
## 3rd Qu.: 43.2 3rd Qu.: 0.0776 3rd Qu.: 0.1295   
## Max. : 1250100.0 Max. : 543.2500 Max. : 9.8037   
## NA's :11 NA's :3 NA's :3   
## Attr8 Attr9 Attr10 Attr11   
## Min. : -3.735 Min. : -3.496 Min. :-479.9100 Min. :-463.8900   
## 1st Qu.: 0.401 1st Qu.: 1.010 1st Qu.: 0.2817 1st Qu.: 0.0081   
## Median : 1.000 Median : 1.141 Median : 0.4889 Median : 0.0631   
## Mean : 6.212 Mean : 1.663 Mean : 0.3696 Mean : -0.0065   
## 3rd Qu.: 2.528 3rd Qu.: 1.876 3rd Qu.: 0.6974 3rd Qu.: 0.1438   
## Max. :6868.500 Max. :109.500 Max. : 339.8500 Max. : 9.8135   
## NA's :19 NA's :1 NA's :3 NA's :39   
## Attr12 Attr13 Attr14 Attr15   
## Min. :-6331.800 Min. :-310.3400 Min. :-517.4800 Min. :-9632400   
## 1st Qu.: 0.004 1st Qu.: 0.0171 1st Qu.: 0.0014 1st Qu.: 183   
## Median : 0.131 Median : 0.0595 Median : 0.0466 Median : 879   
## Mean : 0.047 Mean : 0.3968 Mean : -0.1058 Mean : 1495   
## 3rd Qu.: 0.485 3rd Qu.: 0.1243 3rd Qu.: 0.1300 3rd Qu.: 2427   
## Max. : 2470.300 Max. :2340.2000 Max. : 9.8037 Max. : 1341700   
## NA's :23 NA's :8 NA's :3 NA's :6   
## Attr16 Attr17 Attr18   
## Min. :-6331.800 Min. : -0.002 Min. :-517.4800   
## 1st Qu.: 0.048 1st Qu.: 1.426 1st Qu.: 0.0014   
## Median : 0.196 Median : 2.052 Median : 0.0466   
## Mean : 0.174 Mean : 7.305 Mean : -0.0980   
## 3rd Qu.: 0.574 3rd Qu.: 3.610 3rd Qu.: 0.1300   
## Max. : 2837.400 Max. :6869.500 Max. : 55.1250   
## NA's :19 NA's :19 NA's :3   
## Attr19 Attr20 Attr21 Attr22   
## Min. :-310.80000 Min. : -29.34 Min. :-135.150 Min. :-431.5900   
## 1st Qu.: 0.00105 1st Qu.: 17.48 1st Qu.: 0.973 1st Qu.: 0.0000   
## Median : 0.02934 Median : 38.26 Median : 1.108 Median : 0.0549   
## Mean : -0.07557 Mean : 58.78 Mean : 2.361 Mean : -0.0028   
## 3rd Qu.: 0.08185 3rd Qu.: 68.20 3rd Qu.: 1.260 3rd Qu.: 0.1320   
## Max. : 129.02000 Max. :10620.00 Max. :7661.500 Max. : 15.5410   
## NA's :8 NA's :8 NA's :479 NA's :3   
## Attr23 Attr24 Attr25   
## Min. :-310.89000 Min. :-463.8900 Min. :-500.9300   
## 1st Qu.: 0.00021 1st Qu.: 0.0022 1st Qu.: 0.1346   
## Median : 0.02442 Median : 0.1304 Median : 0.3812   
## Mean : -0.08162 Mean : 0.1163 Mean : 0.1752   
## 3rd Qu.: 0.06988 3rd Qu.: 0.3288 3rd Qu.: 0.6080   
## Max. : 129.02000 Max. : 252.3400 Max. : 266.8600   
## NA's :8 NA's :135 NA's :3   
## Attr26 Attr27 Attr28 Attr29   
## Min. :-6331.800 Min. :-158130.0 Min. :-1089.700 Min. :0.006359   
## 1st Qu.: 0.045 1st Qu.: 0.0 1st Qu.: 0.025 1st Qu.:3.609800   
## Median : 0.180 Median : 1.0 Median : 0.454 Median :4.130300   
## Mean : 0.121 Mean : 406.3 Mean : 8.948 Mean :4.101424   
## 3rd Qu.: 0.513 3rd Qu.: 4.2 3rd Qu.: 1.460 3rd Qu.:4.624300   
## Max. : 2689.100 Max. : 565940.0 Max. :21701.000 Max. :9.698300   
## NA's :19 NA's :816 NA's :154 NA's :3   
## Attr30 Attr31 Attr32 Attr33   
## Min. : -23.060 Min. :-310.80000 Min. : -255 Min. : -1.433   
## 1st Qu.: 0.096 1st Qu.: 0.00307 1st Qu.: 51 1st Qu.: 2.678   
## Median : 0.240 Median : 0.03598 Median : 85 Median : 4.304   
## Mean : 2.615 Mean : -0.04615 Mean : 1891 Mean : 8.291   
## 3rd Qu.: 0.432 3rd Qu.: 0.09091 3rd Qu.: 137 3rd Qu.: 7.147   
## Max. :8518.800 Max. : 164.10000 Max. :4277200 Max. :7590.500   
## NA's :8 NA's :8 NA's :54 NA's :23   
## Attr34 Attr35 Attr36 Attr37   
## Min. : -38.524 Min. :-431.5900 Min. : 0.000 Min. : -89.39   
## 1st Qu.: 0.234 1st Qu.: 0.0030 1st Qu.: 1.077 1st Qu.: 1.27   
## Median : 1.708 Median : 0.0535 Median : 1.581 Median : 3.49   
## Mean : 5.104 Mean : -0.0123 Mean : 2.116 Mean : 102.58   
## 3rd Qu.: 3.965 3rd Qu.: 0.1323 3rd Qu.: 2.315 3rd Qu.: 13.39   
## Max. :7590.500 Max. : 15.5410 Max. :965.660 Max. :67922.00   
## NA's :19 NA's :3 NA's :3 NA's :3090   
## Attr38 Attr39 Attr40 Attr41   
## Min. :-479.9100 Min. :-7522.000 Min. : -9.069 Min. :-269.990   
## 1st Qu.: 0.3910 1st Qu.: 0.002 1st Qu.: 0.045 1st Qu.: 0.026   
## Median : 0.5924 Median : 0.034 Median : 0.150 Median : 0.092   
## Mean : 0.4752 Mean : -1.067 Mean : 2.129 Mean : 1.768   
## 3rd Qu.: 0.7579 3rd Qu.: 0.084 3rd Qu.: 0.604 3rd Qu.: 0.218   
## Max. : 467.7700 Max. : 4.968 Max. :4303.200 Max. :5043.300   
## NA's :3 NA's :8 NA's :23 NA's :84   
## Attr42 Attr43 Attr44 Attr45   
## Min. :-143.52000 Min. :-3975.60 Min. :-3946.20 Min. : -3037.3   
## 1st Qu.: 0.00000 1st Qu.: 73.45 1st Qu.: 37.38 1st Qu.: 0.0   
## Median : 0.03511 Median : 105.60 Median : 57.83 Median : 0.2   
## Mean : -0.00297 Mean : 163.26 Mean : 104.48 Mean : 57.5   
## 3rd Qu.: 0.08431 3rd Qu.: 148.23 3rd Qu.: 84.29 3rd Qu.: 0.7   
## Max. : 160.11000 Max. :62210.00 Max. :51591.00 Max. :366030.0   
## NA's :8 NA's :8 NA's :8 NA's :364   
## Attr46 Attr47 Attr48 Attr49   
## Min. : -9.049 Min. : -18.66 Min. :-542.5600 Min. :-515.2700   
## 1st Qu.: 0.560 1st Qu.: 18.25 1st Qu.: -0.0390 1st Qu.: -0.0290   
## Median : 0.976 Median : 41.08 Median : 0.0145 Median : 0.0090   
## Mean : 3.756 Mean : 128.95 Mean : -0.0830 Mean : -0.1512   
## 3rd Qu.: 1.806 3rd Qu.: 73.33 3rd Qu.: 0.0910 3rd Qu.: 0.0566   
## Max. :6845.800 Max. :185610.00 Max. : 15.5410 Max. : 16.8660   
## NA's :24 NA's :35 NA's :3 NA's :8   
## Attr50 Attr51 Attr52 Attr53   
## Min. : -0.012 Min. : -0.1866 Min. : -0.6976 Min. :-1088.70   
## 1st Qu.: 0.773 1st Qu.: 0.1988 1st Qu.: 0.1390 1st Qu.: 0.67   
## Median : 1.204 Median : 0.3590 Median : 0.2316 Median : 1.19   
## Mean : 3.970 Mean : 0.5851 Mean : 0.8480 Mean : 22.18   
## 3rd Qu.: 2.183 3rd Qu.: 0.5544 3rd Qu.: 0.3705 3rd Qu.: 2.19   
## Max. :6845.800 Max. :480.9600 Max. :666.1100 Max. :86795.00   
## NA's :19 NA's :3 NA's :37 NA's :154   
## Attr54 Attr55 Attr56 Attr57   
## Min. :-1088.70 Min. :-1118500 Min. :-7522.100 Min. :-1236.3000   
## 1st Qu.: 0.92 1st Qu.: 3 1st Qu.: 0.006 1st Qu.: 0.0084   
## Median : 1.36 Median : 1318 Median : 0.047 Median : 0.1009   
## Mean : 23.16 Mean : 9211 Mean : -1.027 Mean : -0.2474   
## 3rd Qu.: 2.33 3rd Qu.: 6259 3rd Qu.: 0.119 3rd Qu.: 0.2435   
## Max. :87092.00 Max. : 4212200 Max. : 1.651 Max. : 147.1900   
## NA's :154 NA's :8 NA's :4   
## Attr58 Attr59 Attr60 Attr61   
## Min. :-0.1644 Min. :-184.98000 Min. : -12 Min. : -0.0925   
## 1st Qu.: 0.8807 1st Qu.: 0.00000 1st Qu.: 5 1st Qu.: 4.3114   
## Median : 0.9546 Median : 0.00456 Median : 9 Median : 6.2994   
## Mean : 0.9784 Mean : 0.28912 Mean : 793 Mean : 11.9619   
## 3rd Qu.: 0.9936 3rd Qu.: 0.21715 3rd Qu.: 18 3rd Qu.: 9.7280   
## Max. :60.8870 Max. : 308.15000 Max. :4818700 Max. :1615.8000   
## NA's :5 NA's :4 NA's :364 NA's :17   
## Attr62 Attr63 Attr64 Stecaj   
## Min. : -236.5 Min. : -1.543 Min. : -3.73 0:5500   
## 1st Qu.: 46.6 1st Qu.: 2.889 1st Qu.: 2.13 1:1596   
## Median : 77.9 Median : 4.665 Median : 4.16   
## Mean : 465.6 Mean : 8.910 Mean : 77.83   
## 3rd Qu.: 125.8 3rd Qu.: 7.771 3rd Qu.: 9.37   
## Max. :1073400.0 Max. :7641.300 Max. :158180.00   
## NA's :8 NA's :23 NA's :154

# 3. ZAMENA NEPOSTOJECIH PODATAKA NAs   
  
# Graficka provera postojanja NAs i sablona pojavljivanja,primenom mice package   
  
md.pattern(stecaj, plot = F) # Sablon pojavljivanja NAs. mice package.

bserved,0 missing. Prva kolaona pokazuje ucestlost paterna, a zadnja broj NA u paternu. Output funkcije md.pattern, nije data zbog velicine  
  
  
missmap(stecaj)# Amelia package

Chart, treemap chart

Description automatically generated

# Najveci broj nedostajucih vrednosti Attr37, Attr60, Attr64,...  
  
broj\_NAs<-sum(is.na(stecaj))  
  
if (broj\_NAs != 0) {  
   
 cat("Ukupan broj NAs =",broj\_NAs,"\n")  
   
}

## Ukupan broj NAs = 6566

# Provera da li zavisna promenjiva 'Stecaj' ima NA's  
  
if (sum(is.na(stecaj[,65]))==0) {   
   
 cat("Zavisna promenjiva 'Stecaj' nema NA","\n")  
}

## Zavisna promenjiva 'Stecaj' nema NA

# Unos zamenskih vrednosti za NAs primenom RF   
  
stecaj\_bezNA<-rfImpute(Stecaj~.,data=stecaj, iter=2)

## ntree OOB 1 2  
## 300: 9.63% 3.09% 32.14%  
## ntree OOB 1 2  
## 300: 11.06% 3.73% 36.34%

sum(is.na(stecaj\_bezNA))

## [1] 0

missmap(stecaj\_bezNA)

Chart, treemap chart

Description automatically generated

# 4. PREGLED DISTRIBUCIJA PREDIKTORA   
  
  
opis<-describe(stecaj\_bezNA) # psych  
head(opis)

normal\_prediktor<-subset(opis,(skew<=2&skew>=-2)&(kurtosis<7&kurtosis>-7),select=c(skew,kurtosis))  
normal\_prediktor

## skew kurtosis  
## Stecaj\* 1.32 -0.26  
## Attr29 -0.21 1.02

cat("Prediktori sa normalnom distribucijom su ",row.names(normal\_prediktor)[2],"\n")

## Prediktori sa normalnom distribucijom su Attr29

hist(stecaj\_bezNA$Attr29, main="Histogram varijable X29", xlab="X29")

Chart, histogram

Description automatically generated  
  
  
# 5. PROVERA POSTOJANJA MULTIKOLINEARNOSTI   
  
  
corelacija<-cor(stecaj\_bezNA[,2:64])# korelacija prediktora  
corrplot(corelacija,method='number',type="upper",order="hclust",tl.col='black',tl.pos='n') # plava boja, pozitivna korelacija, crvena negativna. Intezitet boje odgovara intezitetu lin zavisnosti

Chart, histogram

Description automatically generated

corr\_cross(stecaj\_bezNA[,2:64], max\_pvalue = 0.05, top = 10)# package lares. Parovi 10 znacajnih korelacija(p-value<0.05) U plavoj boji su pozitivne korelacije. Korelacije vece od 5%

## Returning only the top 10. You may override with the 'top' argument

Table

Description automatically generated

# Sledi da postoji korelacija (pozitivna i negativna),izmedju prediktora  
  
# Graficki prikaz korelacija (primer za prvih deset prediktora)  
  
pairs.panels(stecaj\_bezNA[,2:11], gap=0, bg=c("red","blue")[stecaj\_bezNA$Stecaj],pch=21)

Diagram, schematic

Description automatically generated

# 6. ANALIZA POSTOJANJA EKSTREMNIH VREDNOSTI   
  
# Vrednosti prediktora koje se nalaze van granice min i max whisker-a, ce se smatrati outliers i bice zamenjenjene  
# sa median vrednosti odgovarajuceg prediktora  
  
stecaj\_bezekstrema\_bezNA<-stecaj\_bezNA  
  
for(i in 2:65){  
   
 out<-boxplot.stats(stecaj\_bezNA[,i],coef = 1.5)$out  
 out\_ind <- which(stecaj\_bezNA[,i] %in% c(out))  
 stecaj\_bezekstrema\_bezNA[c(out\_ind),i]<-median(stecaj\_bezNA[,i], na.rm = TRUE)  
}  
  
summary(stecaj\_bezekstrema\_bezNA)

## Stecaj Attr1 Attr2 Attr3   
## 0:5500 Min. :-0.164610 Min. :0.0000 Min. :-0.56401   
## 1:1596 1st Qu.: 0.009111 1st Qu.:0.2756 1st Qu.: 0.03655   
## Median : 0.039455 Median :0.4865 Median : 0.19308   
## Mean : 0.050101 Mean :0.4865 Mean : 0.21723   
## 3rd Qu.: 0.090696 3rd Qu.:0.6685 3rd Qu.: 0.39613   
## Max. : 0.276170 Max. :1.3369 Max. : 0.97573   
## Attr4 Attr5 Attr6 Attr7   
## Min. :-0.4031 Min. :-195.030 Min. :-0.1171 Min. :-0.19060   
## 1st Qu.: 1.0313 1st Qu.: -38.440 1st Qu.: 0.0000 1st Qu.: 0.01102   
## Median : 1.5356 Median : -5.447 Median : 0.0000 Median : 0.04657   
## Mean : 1.7059 Mean : -4.588 Mean : 0.0117 Mean : 0.05952   
## 3rd Qu.: 2.0837 3rd Qu.: 30.481 3rd Qu.: 0.0000 3rd Qu.: 0.10799   
## Max. : 5.3207 Max. : 185.150 Max. : 0.1948 Max. : 0.32125   
## Attr8 Attr9 Attr10 Attr11   
## Min. :-2.0032 Min. :0.000 Min. :-0.3416 Min. :-0.19513   
## 1st Qu.: 0.4041 1st Qu.:1.010 1st Qu.: 0.3121 1st Qu.: 0.01996   
## Median : 1.0040 Median :1.141 Median : 0.4890 Median : 0.06315   
## Mean : 1.2348 Mean :1.321 Mean : 0.4926 Mean : 0.07202   
## 3rd Qu.: 1.6628 3rd Qu.:1.569 3rd Qu.: 0.6956 3rd Qu.: 0.12361   
## Max. : 5.7573 Max. :3.173 Max. : 1.0000 Max. : 0.34615   
## Attr12 Attr13 Attr14 Attr15   
## Min. :-0.72078 Min. :-0.14255 Min. :-0.19089 Min. :-3173.1   
## 1st Qu.: 0.01507 1st Qu.: 0.02635 1st Qu.: 0.01102 1st Qu.: 358.3   
## Median : 0.13117 Median : 0.05951 Median : 0.04660 Median : 879.3   
## Mean : 0.16963 Mean : 0.06779 Mean : 0.05955 Mean : 1083.2   
## 3rd Qu.: 0.28495 3rd Qu.: 0.10445 3rd Qu.: 0.10811 3rd Qu.: 1682.0   
## Max. : 1.21420 Max. : 0.28538 Max. : 0.32194 Max. : 5782.6   
## Attr16 Attr17 Attr18 Attr19   
## Min. :-0.74539 Min. :-0.002321 Min. :-0.19089 Min. :-0.120320   
## 1st Qu.: 0.05776 1st Qu.: 1.426400 1st Qu.: 0.01102 1st Qu.: 0.008452   
## Median : 0.19706 Median : 2.055375 Median : 0.04660 Median : 0.029261   
## Mean : 0.24043 Mean : 2.284888 Mean : 0.05954 Mean : 0.037862   
## 3rd Qu.: 0.35485 3rd Qu.: 2.727725 3rd Qu.: 0.10808 3rd Qu.: 0.066264   
## Max. : 1.37880 Max. : 6.929400 Max. : 0.32194 Max. : 0.202740   
## Attr20 Attr21 Attr22 Attr23   
## Min. :-29.34 Min. :0.5495 Min. :-0.197420 Min. :-0.104280   
## 1st Qu.: 17.50 1st Qu.:0.9959 1st Qu.: 0.007937 1st Qu.: 0.007138   
## Median : 38.33 Median :1.1140 Median : 0.054864 Median : 0.024344   
## Mean : 42.30 Mean :1.1086 Mean : 0.064626 Mean : 0.032364   
## 3rd Qu.: 59.48 3rd Qu.:1.2161 3rd Qu.: 0.113375 3rd Qu.: 0.055224   
## Max. :144.50 Max. :1.6918 Max. : 0.329830 Max. : 0.174380   
## Attr24 Attr25 Attr26 Attr27   
## Min. :-0.4941 Min. :-0.5733 Min. :-0.66011 Min. :-10.4250   
## 1st Qu.: 0.0209 1st Qu.: 0.1821 1st Qu.: 0.05438 1st Qu.: 0.2666   
## Median : 0.1341 Median : 0.3812 Median : 0.18099 Median : 1.1458   
## Mean : 0.1586 Mean : 0.3846 Mean : 0.21667 Mean : 1.5837   
## 3rd Qu.: 0.2926 3rd Qu.: 0.6072 3rd Qu.: 0.31952 3rd Qu.: 1.5859   
## Max. : 0.8352 Max. : 1.0000 Max. : 1.22110 Max. : 17.4210   
## Attr28 Attr29 Attr30 Attr31   
## Min. :-2.31320 Min. :2.089 Min. :-0.4050 Min. :-0.12875   
## 1st Qu.: 0.06307 1st Qu.:3.640 1st Qu.: 0.1076 1st Qu.: 0.01235   
## Median : 0.47267 Median :4.130 Median : 0.2405 Median : 0.03585   
## Mean : 0.55974 Mean :4.120 Mean : 0.2467 Mean : 0.04462   
## 3rd Qu.: 0.81768 3rd Qu.:4.605 3rd Qu.: 0.3571 3rd Qu.: 0.07486   
## Max. : 3.93300 Max. :6.134 Max. : 0.9360 Max. : 0.22259   
## Attr32 Attr33 Attr34 Attr35   
## Min. : 0.00 Min. :-1.433 Min. :-3.0109 Min. :-0.19030   
## 1st Qu.: 51.21 1st Qu.: 2.682 1st Qu.: 0.2387 1st Qu.: 0.01243   
## Median : 85.63 Median : 4.313 Median : 1.7189 Median : 0.05347   
## Mean : 90.21 Mean : 4.719 Mean : 2.0796 Mean : 0.06326   
## 3rd Qu.:115.66 3rd Qu.: 6.122 3rd Qu.: 3.1631 3rd Qu.: 0.11155   
## Max. :268.32 Max. :13.905 Max. : 9.6071 Max. : 0.32615   
## Attr36 Attr37 Attr38 Attr39   
## Min. :0.000 Min. :-89.39 Min. :-0.1596 Min. :-0.12081   
## 1st Qu.:1.078 1st Qu.: 2.77 1st Qu.: 0.4267 1st Qu.: 0.01000   
## Median :1.581 Median : 23.76 Median : 0.5925 Median : 0.03360   
## Mean :1.658 Mean : 75.36 Mean : 0.5766 Mean : 0.04160   
## 3rd Qu.:2.111 3rd Qu.:138.40 3rd Qu.: 0.7570 3rd Qu.: 0.07005   
## Max. :4.168 Max. :371.81 Max. : 1.1701 Max. : 0.20620   
## Attr40 Attr41 Attr42 Attr43   
## Min. :-0.43106 Min. :-0.27227 Min. :-0.126110 Min. : 0.00   
## 1st Qu.: 0.04471 1st Qu.: 0.04093 1st Qu.: 0.007706 1st Qu.: 73.51   
## Median : 0.15139 Median : 0.09437 Median : 0.035053 Median :105.63   
## Mean : 0.24503 Mean : 0.10881 Mean : 0.042468 Mean :108.36   
## 3rd Qu.: 0.29711 3rd Qu.: 0.15642 3rd Qu.: 0.071790 3rd Qu.:137.82   
## Max. : 1.47360 Max. : 0.52439 Max. : 0.210430 Max. :260.59   
## Attr44 Attr45 Attr46 Attr47   
## Min. : 0.00 Min. :-1.25000 Min. :-0.3429 Min. :-18.66   
## 1st Qu.: 37.39 1st Qu.: 0.04918 1st Qu.: 0.5618 1st Qu.: 18.34   
## Median : 57.87 Median : 0.22719 Median : 0.9795 Median : 41.27   
## Mean : 59.21 Mean : 0.28762 Mean : 1.0746 Mean : 45.11   
## 3rd Qu.: 76.22 3rd Qu.: 0.44132 3rd Qu.: 1.3516 3rd Qu.: 63.52   
## Max. :155.01 Max. : 2.09400 Max. : 3.7069 Max. :157.24   
## Attr48 Attr49 Attr50 Attr51   
## Min. :-0.23331 Min. :-0.157770 Min. :-0.01218 Min. :-0.1866   
## 1st Qu.:-0.01952 1st Qu.:-0.010676 1st Qu.: 0.77419 1st Qu.: 0.1989   
## Median : 0.01453 Median : 0.008843 Median : 1.20608 Median : 0.3589   
## Mean : 0.02448 Mean : 0.015345 Mean : 1.33281 Mean : 0.3787   
## 3rd Qu.: 0.07241 3rd Qu.: 0.045619 3rd Qu.: 1.62652 3rd Qu.: 0.5195   
## Max. : 0.28517 Max. : 0.184430 Max. : 4.33330 Max. : 1.0873   
## Attr52 Attr53 Attr54 Attr55   
## Min. :0.0000 Min. :-1.7093 Min. :-1.313 Min. :-9358.0   
## 1st Qu.:0.1394 1st Qu.: 0.7353 1st Qu.: 0.976 1st Qu.: 105.3   
## Median :0.2322 Median : 1.2120 Median : 1.377 Median : 1318.3   
## Mean :0.2445 Mean : 1.2676 Mean : 1.449 Mean : 2048.3   
## 3rd Qu.:0.3141 3rd Qu.: 1.5846 3rd Qu.: 1.710 3rd Qu.: 2936.7   
## Max. :0.7242 Max. : 4.6760 Max. : 4.694 Max. :15615.0   
## Attr56 Attr57 Attr58 Attr59   
## Min. :-0.16382 Min. :-0.34437 Min. :0.7119 Min. :-0.301380   
## 1st Qu.: 0.01538 1st Qu.: 0.03403 1st Qu.:0.9105 1st Qu.: 0.000000   
## Median : 0.04732 Median : 0.10103 Median :0.9546 Median : 0.004603   
## Mean : 0.05658 Mean : 0.11941 Mean :0.9444 Mean : 0.062493   
## 3rd Qu.: 0.09105 3rd Qu.: 0.19018 3rd Qu.:0.9865 3rd Qu.: 0.061672   
## Max. : 0.28918 Max. : 0.59576 Max. :1.1629 Max. : 0.542110   
## Attr60 Attr61 Attr62 Attr63   
## Min. :-12.440 Min. :-0.09249 Min. : 0.00 Min. :-1.543   
## 1st Qu.: 5.335 1st Qu.: 4.32005 1st Qu.: 46.68 1st Qu.: 2.892   
## Median : 9.523 Median : 6.30610 Median : 77.99 Median : 4.679   
## Mean : 10.852 Mean : 6.65797 Mean : 82.26 Mean : 5.116   
## 3rd Qu.: 12.895 3rd Qu.: 8.22705 3rd Qu.:107.45 3rd Qu.: 6.603   
## Max. : 44.113 Max. :17.91300 Max. :245.37 Max. :15.193   
## Attr64   
## Min. :-3.727   
## 1st Qu.: 2.170   
## Median : 4.288   
## Mean : 4.952   
## 3rd Qu.: 5.762   
## Max. :22.381

# Kako je jedino prediktor X55, vrednost Working capital, dat u svojoj apsolutnoj vrednosti, a ne kao 'ratio'  
# pristupicemo normalizaciji podataka, svodjenjem na isti rang vrednosti ( rang od 0 do 1)  
  
# Min max normalizacija  
  
preproc <- preProcess(stecaj\_bezekstrema\_bezNA[,-1], method=c("range"))# default range 0-1  
stecaj\_bezekstrema\_bezNA\_scaled <- predict(preproc, stecaj\_bezekstrema\_bezNA[,-1])  
stecaj\_bezekstrema\_bezNA\_scaled$Stecaj<- as.factor(stecaj\_bezekstrema\_bezNA$Stecaj)  
  
  
# 7. KREIRANJE TRENING/*TESTING* PODATAKA (podaci bez NA, ekstremnih vrednosti i standardizovani mean nula i sd 1)  
  
  
# Train/test podaci u odnosu 70/30  
  
set.seed(2)  
  
split <- createDataPartition (stecaj\_bezekstrema\_bezNA\_scaled$Stecaj, p = .7, list = F)# caret  
train <- stecaj\_bezekstrema\_bezNA\_scaled[split,]# Attr1....Attr64, Stecaj  
test <- stecaj\_bezekstrema\_bezNA\_scaled[-split,]  
  
  
# Proporcionalnost podataka razlicitih klasa  
  
prop.table(table(train$Stecaj))

##   
## 0 1   
## 0.7749597 0.2250403

prop.table(table(test$Stecaj))

##   
## 0 1   
## 0.7753759 0.2246241

# 7. KREIRANJE PRINCIPALNIH KOMPONENTI pc.  
  
train\_pc <- stecaj\_bezekstrema\_bezNA[split,]# Stecaj, Attr1 ....Attr64  
test\_pc <- stecaj\_bezekstrema\_bezNA[-split,]  
  
  
pc<-prcomp(train\_pc[,-1],center=T, scale=T) # kreiranje PC1...PC64, u funkciji Attr1....Attr64  
  
summary(pc)$importance[3,1:40][40]# Kumulativna varijacija prvih 30 PC's (Cumulative proportion)

## PC40   
## 0.96054

pc$rotation # linerna transformacija prediktora u PC. PC= f(Attr1.....Attr64). Izostavljen je output zbog velicine

# Korelacija izmedju PC je nula, nema linearn zavisnosti  
  
corelacija\_pc<-cor(pc$x)  
corrplot(corelacija\_pc,type="full",order="hclust",tl.col='black',tl.pos='n') # plava boja, pozitivna korelacija, crvena negativna. Intezitet boje odgovara intezitetu lin zavisnosti

Chart

Description automatically generated

#Train/test podaci u funkciji PCs  
  
predict\_pctrain<-predict(pc, train\_pc[,-1])[,1:40]# ne potrebno to je isto kao pc$x  
  
train\_pc<-data.frame(predict\_pctrain, Stecaj=train\_pc$Stecaj)  
  
predict\_pctest<-predict(pc,test\_pc[,-1])[,1:40]  
  
test\_pc<-data.frame(predict\_pctest, Stecaj=test\_pc$Stecaj)  
  
  
# 8. PREDIKTIVNI MODELI  
  
  
# 8.1 LOGISTICKA REGRESIJA  
  
# Kreiranje modela  
  
model\_lr<-glm(Stecaj~ ., data=train, family = binomial)  
summary(model\_lr)

##   
## Call:  
## glm(formula = Stecaj ~ ., family = binomial, data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2252 -0.6443 -0.3962 -0.1273 3.2732   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.54580 1.12175 7.618 2.57e-14 \*\*\*  
## Attr1 0.05866 0.54664 0.107 0.914541   
## Attr2 -0.12406 0.92469 -0.134 0.893276   
## Attr3 0.28377 0.41166 0.689 0.490621   
## Attr4 0.40889 0.50693 0.807 0.419894   
## Attr5 0.51663 0.41433 1.247 0.212435   
## Attr6 0.12234 0.29534 0.414 0.678689   
## Attr7 -2.94893 3.46993 -0.850 0.395407   
## Attr8 -2.15967 1.02879 -2.099 0.035797 \*   
## Attr9 -1.00828 0.26590 -3.792 0.000149 \*\*\*  
## Attr10 1.94590 0.98580 1.974 0.048389 \*   
## Attr11 0.01244 0.60287 0.021 0.983533   
## Attr12 0.27080 0.50157 0.540 0.589255   
## Attr13 -0.96470 0.49447 -1.951 0.051061 .   
## Attr14 3.11381 3.47695 0.896 0.370490   
## Attr15 -0.45473 0.30244 -1.504 0.132694   
## Attr16 0.30095 0.89519 0.336 0.736731   
## Attr17 -0.78377 0.86799 -0.903 0.366539   
## Attr18 NA NA NA NA   
## Attr19 0.71133 0.64980 1.095 0.273651   
## Attr20 0.87270 0.45596 1.914 0.055623 .   
## Attr21 -1.29712 0.23788 -5.453 4.95e-08 \*\*\*  
## Attr22 0.27043 0.73562 0.368 0.713155   
## Attr23 -0.20318 0.54836 -0.371 0.710989   
## Attr24 -2.09454 0.32647 -6.416 1.40e-10 \*\*\*  
## Attr25 -1.37123 0.32261 -4.250 2.13e-05 \*\*\*  
## Attr26 -1.08687 0.88154 -1.233 0.217602   
## Attr27 1.08323 0.38068 2.846 0.004434 \*\*   
## Attr28 -0.30457 0.77835 -0.391 0.695576   
## Attr29 -1.39015 0.26417 -5.262 1.42e-07 \*\*\*  
## Attr30 -0.30762 0.39233 -0.784 0.432982   
## Attr31 0.14203 0.59007 0.241 0.809793   
## Attr32 1.05465 0.64491 1.635 0.101979   
## Attr33 1.76564 0.55909 3.158 0.001588 \*\*   
## Attr34 -0.42041 0.33872 -1.241 0.214543   
## Attr35 -1.03824 0.52065 -1.994 0.046139 \*   
## Attr36 0.72655 0.30264 2.401 0.016362 \*   
## Attr37 -2.17261 0.32991 -6.586 4.53e-11 \*\*\*  
## Attr38 -3.00142 0.70973 -4.229 2.35e-05 \*\*\*  
## Attr39 -0.64322 0.51659 -1.245 0.213082   
## Attr40 0.27424 0.36522 0.751 0.452719   
## Attr41 -1.33247 0.32168 -4.142 3.44e-05 \*\*\*  
## Attr42 -0.69979 0.66828 -1.047 0.295023   
## Attr43 -0.50510 0.36653 -1.378 0.168184   
## Attr44 -0.31503 0.31947 -0.986 0.324088   
## Attr45 -0.11753 0.37424 -0.314 0.753491   
## Attr46 -3.19903 0.48761 -6.561 5.36e-11 \*\*\*  
## Attr47 -0.87783 0.43831 -2.003 0.045201 \*   
## Attr48 1.04326 0.52000 2.006 0.044828 \*   
## Attr49 1.19093 0.48298 2.466 0.013671 \*   
## Attr50 1.09109 0.48268 2.260 0.023790 \*   
## Attr51 -1.29045 0.64793 -1.992 0.046408 \*   
## Attr52 -1.89423 0.68038 -2.784 0.005368 \*\*   
## Attr53 0.02881 0.68310 0.042 0.966361   
## Attr54 -0.45968 0.93453 -0.492 0.622805   
## Attr55 -1.27851 0.37436 -3.415 0.000637 \*\*\*  
## Attr56 -0.44558 0.48170 -0.925 0.354961   
## Attr57 -0.76229 0.32785 -2.325 0.020067 \*   
## Attr58 -1.73934 0.46698 -3.725 0.000196 \*\*\*  
## Attr59 -0.48711 0.29809 -1.634 0.102231   
## Attr60 -0.15492 0.35430 -0.437 0.661920   
## Attr61 0.01952 0.27262 0.072 0.942922   
## Attr62 1.25711 0.40321 3.118 0.001823 \*\*   
## Attr63 -1.98695 0.59966 -3.313 0.000922 \*\*\*  
## Attr64 0.12850 0.30092 0.427 0.669374   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5298.0 on 4967 degrees of freedom  
## Residual deviance: 4057.3 on 4904 degrees of freedom  
## AIC: 4185.3  
##   
## Number of Fisher Scoring iterations: 5

# Predikcija  
  
predict\_lr<-model\_lr%>%predict(test,type="response")  
  
# Tacnost modela  
  
predicted\_lr<-ifelse(predict\_lr>0.3,"1","0")  
confusionMatrix(factor(predicted\_lr), test$Stecaj, positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1321 165  
## 1 329 313  
##   
## Accuracy : 0.7679   
## 95% CI : (0.7493, 0.7857)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.8046   
##   
## Kappa : 0.406   
##   
## Mcnemar's Test P-Value : 2.239e-13   
##   
## Sensitivity : 0.6548   
## Specificity : 0.8006   
## Pos Pred Value : 0.4875   
## Neg Pred Value : 0.8890   
## Prevalence : 0.2246   
## Detection Rate : 0.1471   
## Detection Prevalence : 0.3017   
## Balanced Accuracy : 0.7277   
##   
## 'Positive' Class : 1   
##

# Model\_lr samo sa znacajnim prediktorima  
  
model\_lr1<-glm(Stecaj ~ Attr21 + Attr24 + Attr25 + Attr37 + Attr38 +Attr41 +Attr46 +Attr55 , data=train, family = binomial)  
summary(model\_lr1)

##   
## Call:  
## glm(formula = Stecaj ~ Attr21 + Attr24 + Attr25 + Attr37 + Attr38 +   
## Attr41 + Attr46 + Attr55, family = binomial, data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2253 -0.6772 -0.4540 -0.1930 2.9368   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.5397 0.2393 18.968 < 2e-16 \*\*\*  
## Attr21 -1.5164 0.2189 -6.928 4.28e-12 \*\*\*  
## Attr24 -1.9396 0.2525 -7.680 1.59e-14 \*\*\*  
## Attr25 -1.2599 0.2666 -4.725 2.30e-06 \*\*\*  
## Attr37 -1.1292 0.2620 -4.310 1.63e-05 \*\*\*  
## Attr38 -1.1173 0.2666 -4.191 2.78e-05 \*\*\*  
## Attr41 -1.9742 0.2376 -8.309 < 2e-16 \*\*\*  
## Attr46 -2.6844 0.3319 -8.088 6.06e-16 \*\*\*  
## Attr55 -1.5889 0.3221 -4.933 8.09e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5298.0 on 4967 degrees of freedom  
## Residual deviance: 4352.2 on 4959 degrees of freedom  
## AIC: 4370.2  
##   
## Number of Fisher Scoring iterations: 5

# Multikolinearnost, ne utice na tacnost klasifikacionog modela, ali utice na vrednost koeficijenata i tako odnos izmedju outputa i nezavisne promenjive  
# Vrednosti Variance inflation factor (VIF) u rangu 1-5 oznacavaju srednji nivo kolinearnosti  
  
vif\_prediktora<-data.frame(vif(model\_lr1)) # car paket. VIF prediktora pokazuje koliko dobro se taj prediktor moze 'objasniti' drugim prediktorom  
# Moze se pojaviti greska kada su dve ili vise varijabli mnogo (ili perfektno) correlated.  
  
vif\_prediktora<-rownames\_to\_column(vif\_prediktora)  
  
max\_koli<-vif\_prediktora[order(vif\_prediktora$vif.model\_lr1., decreasing = T),]  
max\_koli[which(max\_koli$vif.model\_lr1>5),]

## [1] rowname vif.model\_lr1.  
## <0 rows> (or 0-length row.names)

# Predikcija  
  
predict\_lr1<-model\_lr1%>%predict(test,type="response")  
  
  
# Tacnost modela  
  
predicted\_lr1<-ifelse(predict\_lr1>0.3,"1","0")  
confusionMatrix(factor(predicted\_lr1), test$Stecaj, positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1327 186  
## 1 323 292  
##   
## Accuracy : 0.7608   
## 95% CI : (0.7421, 0.7788)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.9483   
##   
## Kappa : 0.3768   
##   
## Mcnemar's Test P-Value : 1.659e-09   
##   
## Sensitivity : 0.6109   
## Specificity : 0.8042   
## Pos Pred Value : 0.4748   
## Neg Pred Value : 0.8771   
## Prevalence : 0.2246   
## Detection Rate : 0.1372   
## Detection Prevalence : 0.2890   
## Balanced Accuracy : 0.7076   
##   
## 'Positive' Class : 1   
##

# Poredjenje log reg modela prema AIC  
  
# Model koji je vise od 2 AIC (Akaike iformation criteria, pokazuje koliko dobro model opisuje podatke) jedinice nizi od drugog modela , je statisticki znacajno bolji  
  
models<-list(model\_lr,model\_lr1)  
model.names<-c('model\_lr','modela\_lr1')  
aictab(cand.set = models, modnames = model.names)

##   
## Model selection based on AICc:  
##   
## K AICc Delta\_AICc AICcWt Cum.Wt LL  
## model\_lr 64 4186.95 0.00 1 1 -2028.63  
## modela\_lr1 9 4370.20 183.25 0 1 -2176.08

# Najvazniji, najinformativniji prediktori  
  
coef(model\_lr1)# Intercept i 64 prediktora

## (Intercept) Attr21 Attr24 Attr25 Attr37 Attr38   
## 4.539671 -1.516351 -1.939614 -1.259921 -1.129191 -1.117313   
## Attr41 Attr46 Attr55   
## -1.974199 -2.684390 -1.588924

df\_regkoeficijenti<-data.frame(coef(model\_lr1))# uklanjam intercept  
df\_regkoeficijenti<-rownames\_to\_column(data.frame(df\_regkoeficijenti))# package tibble  
df\_regkoeficijenti<-df\_regkoeficijenti[-1,]  
colnames(df\_regkoeficijenti)<-c('Prediktori','RegKoeficijenti')  
df\_regkoeficijenti[order(-abs(df\_regkoeficijenti$RegKoeficijenti)),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## Prediktori RegKoeficijenti  
## 8 Attr46 -2.684390  
## 7 Attr41 -1.974199  
## 3 Attr24 -1.939614  
## 9 Attr55 -1.588924  
## 2 Attr21 -1.516351

# 8.1.1 Log regresija sa PC  
  
model\_lrpc<-glm(Stecaj~ ., data=train\_pc, family = binomial)#model sa PC  
summary(model\_lrpc)

##   
## Call:  
## glm(formula = Stecaj ~ ., family = binomial, data = train\_pc)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2522 -0.6504 -0.4121 -0.1501 2.9574   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.669918 0.048090 -34.725 < 2e-16 \*\*\*  
## PC1 -0.230328 0.011205 -20.556 < 2e-16 \*\*\*  
## PC2 0.087319 0.014611 5.976 2.28e-09 \*\*\*  
## PC3 -0.196688 0.018163 -10.829 < 2e-16 \*\*\*  
## PC4 -0.033789 0.022716 -1.487 0.136898   
## PC5 -0.142715 0.027398 -5.209 1.90e-07 \*\*\*  
## PC6 0.325811 0.027322 11.925 < 2e-16 \*\*\*  
## PC7 -0.019872 0.033451 -0.594 0.552458   
## PC8 -0.142855 0.034763 -4.109 3.97e-05 \*\*\*  
## PC9 0.008635 0.035746 0.242 0.809128   
## PC10 0.068871 0.036561 1.884 0.059600 .   
## PC11 -0.093966 0.039542 -2.376 0.017485 \*   
## PC12 0.010371 0.038129 0.272 0.785624   
## PC13 -0.187435 0.040266 -4.655 3.24e-06 \*\*\*  
## PC14 -0.175036 0.042464 -4.122 3.76e-05 \*\*\*  
## PC15 -0.159181 0.040110 -3.969 7.23e-05 \*\*\*  
## PC16 -0.037627 0.046365 -0.812 0.417063   
## PC17 -0.200637 0.047894 -4.189 2.80e-05 \*\*\*  
## PC18 -0.032745 0.048060 -0.681 0.495665   
## PC19 -0.135715 0.048392 -2.804 0.005040 \*\*   
## PC20 -0.198901 0.049056 -4.055 5.02e-05 \*\*\*  
## PC21 0.155830 0.055161 2.825 0.004728 \*\*   
## PC22 0.084523 0.051326 1.647 0.099598 .   
## PC23 0.161381 0.052430 3.078 0.002084 \*\*   
## PC24 -0.273308 0.056404 -4.846 1.26e-06 \*\*\*  
## PC25 -0.146821 0.054369 -2.700 0.006924 \*\*   
## PC26 0.178070 0.058954 3.021 0.002524 \*\*   
## PC27 -0.161805 0.060480 -2.675 0.007465 \*\*   
## PC28 0.009037 0.060087 0.150 0.880456   
## PC29 0.052463 0.063157 0.831 0.406155   
## PC30 -0.305334 0.064545 -4.731 2.24e-06 \*\*\*  
## PC31 -0.358891 0.068033 -5.275 1.33e-07 \*\*\*  
## PC32 0.019822 0.068219 0.291 0.771390   
## PC33 0.272249 0.065605 4.150 3.33e-05 \*\*\*  
## PC34 0.169941 0.067170 2.530 0.011406 \*   
## PC35 0.192277 0.070029 2.746 0.006038 \*\*   
## PC36 0.149981 0.074856 2.004 0.045113 \*   
## PC37 0.171244 0.075860 2.257 0.023985 \*   
## PC38 -0.268343 0.079448 -3.378 0.000731 \*\*\*  
## PC39 -0.250287 0.080882 -3.094 0.001972 \*\*   
## PC40 0.008728 0.081720 0.107 0.914943   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5298.0 on 4967 degrees of freedom  
## Residual deviance: 4144.6 on 4927 degrees of freedom  
## AIC: 4226.6  
##   
## Number of Fisher Scoring iterations: 5

# Predikcija  
  
predict\_lrpc<-model\_lrpc%>%predict(test\_pc,type="response")# sa PC  
  
# Tacnost sa PC  
  
predicted\_lrpc<-ifelse(predict\_lrpc>0.3,"1","0")  
confusionMatrix(factor(predicted\_lrpc), test\_pc$Stecaj, positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1321 176  
## 1 329 302  
##   
## Accuracy : 0.7627   
## 95% CI : (0.744, 0.7806)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.9227   
##   
## Kappa : 0.3883   
##   
## Mcnemar's Test P-Value : 1.343e-11   
##   
## Sensitivity : 0.6318   
## Specificity : 0.8006   
## Pos Pred Value : 0.4786   
## Neg Pred Value : 0.8824   
## Prevalence : 0.2246   
## Detection Rate : 0.1419   
## Detection Prevalence : 0.2965   
## Balanced Accuracy : 0.7162   
##   
## 'Positive' Class : 1   
##

# 8.1.1 PENALIZED LOGISTIC REGRESSION, regularizacija modela (ne proredimo prema AIC)  
  
#Lasso regresija  
  
# Kreiranje matrice prediktora   
  
x\_train<- model.matrix(Stecaj~., data=train,)[,-1]# U slucaju postojanja factor promenjivih, kreiraju se dummy promenjive  
x\_test<-model.matrix(Stecaj~., test,)[,-1]  
  
y\_output<-train$Stecaj  
  
# Koriscenjem CV trazimo najoptimalniju vrednost lambda, kojom se odredjuje stepen 'sankcionisanja' kompleksnosti modela tj. umanjenje reg koeficijenata   
  
lasso\_param<-cv.glmnet(x\_train,y\_output,alpha=1,family="binomial")# glmnet package. Za alpha 0=1, radi se o Lasso reg  
  
# Kreiranje modela  
  
model\_lrlasso<- glmnet(x\_train,y\_output, alpha = 1, family = "binomial",lambda = lasso\_param$lambda.min)  
  
# Predikcija  
  
predict\_lrlasso<-model\_lrlasso%>%predict(newx=x\_test,type="response")  
  
# Tacnost modela  
predicted\_lrlasso<-ifelse(predict\_lrlasso>0.3,"1","0")  
confusionMatrix(factor(predicted\_lrlasso), test$Stecaj, positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1321 172  
## 1 329 306  
##   
## Accuracy : 0.7646   
## 95% CI : (0.7459, 0.7825)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.8885   
##   
## Kappa : 0.3947   
##   
## Mcnemar's Test P-Value : 3.179e-12   
##   
## Sensitivity : 0.6402   
## Specificity : 0.8006   
## Pos Pred Value : 0.4819   
## Neg Pred Value : 0.8848   
## Prevalence : 0.2246   
## Detection Rate : 0.1438   
## Detection Prevalence : 0.2984   
## Balanced Accuracy : 0.7204   
##   
## 'Positive' Class : 1   
##

# Najvazniji, najinformativniji prediktori  
  
coef(model\_lrlasso)# Intercept i 64 prediktora

## 65 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) 7.449074544  
## Attr1 .   
## Attr2 -0.344497346  
## Attr3 0.245708020  
## Attr4 0.182748223  
## Attr5 0.254618390  
## Attr6 0.017048673  
## Attr7 .   
## Attr8 -1.992398912  
## Attr9 -0.935923353  
## Attr10 1.133890461  
## Attr11 .   
## Attr12 .   
## Attr13 -0.779627627  
## Attr14 0.111045709  
## Attr15 -0.418095808  
## Attr16 .   
## Attr17 -0.420995012  
## Attr18 0.036994710  
## Attr19 0.340522707  
## Attr20 0.410516818  
## Attr21 -1.220701914  
## Attr22 .   
## Attr23 .   
## Attr24 -1.995397018  
## Attr25 -1.123041779  
## Attr26 -0.631039834  
## Attr27 0.947259470  
## Attr28 .   
## Attr29 -1.384440504  
## Attr30 -0.286119931  
## Attr31 .   
## Attr32 .   
## Attr33 1.025845770  
## Attr34 -0.170979978  
## Attr35 -0.778698681  
## Attr36 0.634417726  
## Attr37 -1.863351796  
## Attr38 -1.868358931  
## Attr39 -0.733195948  
## Attr40 0.122564296  
## Attr41 -1.300590676  
## Attr42 -0.136278262  
## Attr43 -0.287361900  
## Attr44 -0.369998117  
## Attr45 -0.044493279  
## Attr46 -2.725446845  
## Attr47 -0.533502948  
## Attr48 0.963351856  
## Attr49 0.878723413  
## Attr50 0.709470684  
## Attr51 -0.327636218  
## Attr52 -0.850615960  
## Attr53 .   
## Attr54 -0.527876161  
## Attr55 -1.096801797  
## Attr56 -0.207638907  
## Attr57 -0.583810629  
## Attr58 -1.409056925  
## Attr59 -0.432804751  
## Attr60 -0.104452937  
## Attr61 .   
## Attr62 0.971209235  
## Attr63 -1.279205481  
## Attr64 0.002255009

df\_regkoeficijenti<-data.frame(summary(coef(model\_lrlasso)))# uklanjam intercept  
df\_regkoeficijenti[,1]<- df\_regkoeficijenti[,1]-1  
df\_regkoeficijenti<-df\_regkoeficijenti[-1,]  
  
df\_regkoeficijenti[order(-abs(df\_regkoeficijenti$x)),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## i j x  
## 37 46 1 -2.725447  
## 18 24 1 -1.995397  
## 7 8 1 -1.992399  
## 29 38 1 -1.868359  
## 28 37 1 -1.863352

# Ridge regresija  
  
ridge\_param<-cv.glmnet(x\_train,y\_output,alpha=0,family="binomial")# glmnet package. Za alpha =0 , radi se o Ridge reg  
  
# Kreiranje modela  
  
model\_lrridge<- glmnet(x\_train,y\_output, alpha = 0, family = "binomial",lambda = ridge\_param$lambda.min)  
  
# Predikcija  
  
predict\_lrridge<-model\_lrridge%>%predict(newx=x\_test,type="response")  
  
# Tacnost modela  
  
predicted\_lrridge<-ifelse(predict\_lrridge>0.3,"1","0")  
confusionMatrix(factor(predicted\_lrridge), test$Stecaj, positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1320 172  
## 1 330 306  
##   
## Accuracy : 0.7641   
## 95% CI : (0.7455, 0.782)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.8979   
##   
## Kappa : 0.3939   
##   
## Mcnemar's Test P-Value : 2.43e-12   
##   
## Sensitivity : 0.6402   
## Specificity : 0.8000   
## Pos Pred Value : 0.4811   
## Neg Pred Value : 0.8847   
## Prevalence : 0.2246   
## Detection Rate : 0.1438   
## Detection Prevalence : 0.2989   
## Balanced Accuracy : 0.7201   
##   
## 'Positive' Class : 1   
##

# Najvazniji, najinformativniji prediktori  
  
coef(model\_lrridge)# Intercept i 64 prediktora

## 65 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) 7.278919025  
## Attr1 0.076463042  
## Attr2 -0.541499593  
## Attr3 0.248217448  
## Attr4 0.107941476  
## Attr5 0.260336358  
## Attr6 0.073198761  
## Attr7 0.001944018  
## Attr8 -1.358188845  
## Attr9 -0.878203404  
## Attr10 0.625325506  
## Attr11 -0.006234887  
## Attr12 0.125892952  
## Attr13 -0.779893721  
## Attr14 0.126612270  
## Attr15 -0.482898691  
## Attr16 -0.203763800  
## Attr17 -0.806626200  
## Attr18 0.108329051  
## Attr19 0.398913116  
## Attr20 0.481577382  
## Attr21 -1.170441810  
## Attr22 0.139174763  
## Attr23 -0.033838144  
## Attr24 -1.800753148  
## Attr25 -1.093181888  
## Attr26 -0.688351400  
## Attr27 0.952395216  
## Attr28 -0.186810526  
## Attr29 -1.325565268  
## Attr30 -0.324758533  
## Attr31 0.071962622  
## Attr32 0.151874597  
## Attr33 0.754317417  
## Attr34 -0.186518381  
## Attr35 -0.804929773  
## Attr36 0.585025881  
## Attr37 -1.607785022  
## Attr38 -1.456527648  
## Attr39 -0.652767240  
## Attr40 0.061139854  
## Attr41 -1.181952747  
## Attr42 -0.327947319  
## Attr43 -0.321470859  
## Attr44 -0.426597612  
## Attr45 -0.187134272  
## Attr46 -2.198845275  
## Attr47 -0.612849295  
## Attr48 0.784564445  
## Attr49 0.861761645  
## Attr50 0.510668302  
## Attr51 -0.175963056  
## Attr52 -0.817888999  
## Attr53 0.046976673  
## Attr54 -0.554709408  
## Attr55 -1.105853041  
## Attr56 -0.248401907  
## Attr57 -0.644460897  
## Attr58 -1.293302016  
## Attr59 -0.426767517  
## Attr60 -0.226899211  
## Attr61 0.021433049  
## Attr62 0.853163000  
## Attr63 -1.052004084  
## Attr64 0.098389862

df\_regkoeficijenti<-data.frame(summary(coef(model\_lrridge)))# uklanjam intercept  
df\_regkoeficijenti[,1]<- df\_regkoeficijenti[,1]-1  
df\_regkoeficijenti<-df\_regkoeficijenti[-1,]  
  
df\_regkoeficijenti[order(-abs(df\_regkoeficijenti$x)),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## i j x  
## 47 46 1 -2.198845  
## 25 24 1 -1.800753  
## 38 37 1 -1.607785  
## 39 38 1 -1.456528  
## 9 8 1 -1.358189

# Elastic net regresija  
  
elasticnet\_param<-cv.glmnet(x\_train,y\_output,alpha=0.7,family="binomial")# glmnet package. Za alpha izmedju 0 i 1, radi se o Elastic Net regresiji  
  
# Kreiranje modela  
  
model\_elasnet<- glmnet(x\_train,y\_output, alpha = 0.7, family = "binomial",lambda = elasticnet\_param$lambda.min)  
  
# Predikcija  
  
predict\_elasnet<-model\_elasnet%>%predict(newx=x\_test,type="response")  
  
# Tacnost modela  
  
predicted\_elasnet<-ifelse(predict\_elasnet>0.3,"1","0")  
confusionMatrix(factor(predicted\_elasnet), test$Stecaj, positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1320 170  
## 1 330 308  
##   
## Accuracy : 0.765   
## 95% CI : (0.7464, 0.7829)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.8784   
##   
## Kappa : 0.3971   
##   
## Mcnemar's Test P-Value : 1.155e-12   
##   
## Sensitivity : 0.6444   
## Specificity : 0.8000   
## Pos Pred Value : 0.4828   
## Neg Pred Value : 0.8859   
## Prevalence : 0.2246   
## Detection Rate : 0.1447   
## Detection Prevalence : 0.2998   
## Balanced Accuracy : 0.7222   
##   
## 'Positive' Class : 1   
##

# Najvazniji, najinformativniji prediktori  
  
coef(model\_elasnet)# Intercept i 64 prediktora

## 65 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) 7.6140518423  
## Attr1 .   
## Attr2 -0.4021084048  
## Attr3 0.2554502940  
## Attr4 0.2130740296  
## Attr5 0.2822071693  
## Attr6 0.0327938609  
## Attr7 .   
## Attr8 -1.9231041492  
## Attr9 -0.9417852588  
## Attr10 1.1330558899  
## Attr11 .   
## Attr12 0.0001558746  
## Attr13 -0.7914642666  
## Attr14 0.1029040234  
## Attr15 -0.4249010127  
## Attr16 .   
## Attr17 -0.5269275713  
## Attr18 0.0748012150  
## Attr19 0.3799105741  
## Attr20 0.4727008618  
## Attr21 -1.2263364582  
## Attr22 .   
## Attr23 .   
## Attr24 -1.9968549097  
## Attr25 -1.1512584173  
## Attr26 -0.6462600480  
## Attr27 0.9669947327  
## Attr28 -0.0034085979  
## Attr29 -1.3836063386  
## Attr30 -0.2874961641  
## Attr31 .   
## Attr32 0.0747302626  
## Attr33 1.0772413203  
## Attr34 -0.1979672679  
## Attr35 -0.8057583979  
## Attr36 0.6432436177  
## Attr37 -1.8881537658  
## Attr38 -1.9585357292  
## Attr39 -0.7223723311  
## Attr40 0.1383806431  
## Attr41 -1.2940644592  
## Attr42 -0.2058074498  
## Attr43 -0.3157753792  
## Attr44 -0.3662012197  
## Attr45 -0.0621566913  
## Attr46 -2.7543019865  
## Attr47 -0.5803956611  
## Attr48 0.9830858976  
## Attr49 0.9194994470  
## Attr50 0.7312927855  
## Attr51 -0.4076007438  
## Attr52 -0.9282338604  
## Attr53 .   
## Attr54 -0.5544663372  
## Attr55 -1.1192562562  
## Attr56 -0.2404376658  
## Attr57 -0.6058703420  
## Attr58 -1.4504658956  
## Attr59 -0.4407137939  
## Attr60 -0.1146088074  
## Attr61 .   
## Attr62 1.0043181855  
## Attr63 -1.3302610223  
## Attr64 0.0230283549

df\_regkoeficijenti<-data.frame(summary(coef(model\_elasnet)))# uklanjam intercept  
df\_regkoeficijenti[,1]<- df\_regkoeficijenti[,1]-1  
df\_regkoeficijenti<-df\_regkoeficijenti[-1,]  
  
df\_regkoeficijenti[order(-abs(df\_regkoeficijenti$x)),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## i j x  
## 40 46 1 -2.754302  
## 19 24 1 -1.996855  
## 32 38 1 -1.958536  
## 7 8 1 -1.923104  
## 31 37 1 -1.888154

# 8.1.2 LDA i QDA (koriscenjem prediktora najtacnijeg lasso regresionog modela)  
  
  
# LDA  
  
model\_lda<-lda(Stecaj~ . -Attr18 -Attr7 -Attr14 -Attr52 -Attr32 -Attr10 -Attr2 -Attr17 -Attr8 -Attr54 , data=train)  
model\_lda

## Call:  
## lda(Stecaj ~ . - Attr18 - Attr7 - Attr14 - Attr52 - Attr32 -   
## Attr10 - Attr2 - Attr17 - Attr8 - Attr54, data = train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.7749597 0.2250403   
##   
## Group means:  
## Attr1 Attr3 Attr4 Attr5 Attr6 Attr9 Attr11  
## 0 0.5051910 0.5303958 0.3913430 0.5199276 0.4224337 0.4205297 0.5124863  
## 1 0.4333736 0.4330470 0.2867893 0.4386853 0.3924450 0.4080481 0.4363359  
## Attr12 Attr13 Attr15 Attr16 Attr19 Attr20 Attr21  
## 0 0.4835421 0.5155737 0.4842694 0.4879320 0.5093777 0.4127294 0.5058552  
## 1 0.3881322 0.4180581 0.4452528 0.3884219 0.4279199 0.4020664 0.4458569  
## Attr22 Attr23 Attr24 Attr25 Attr26 Attr27 Attr28  
## 0 0.5144711 0.5088976 0.5181492 0.6417224 0.4905319 0.4371596 0.4758296  
## 1 0.4456070 0.4308837 0.4051616 0.5091902 0.3874173 0.4160181 0.4080063  
## Attr29 Attr30 Attr31 Attr33 Attr34 Attr35 Attr36  
## 0 0.5221858 0.4710111 0.5134904 0.4167737 0.4060206 0.5089484 0.3926289  
## 1 0.4420791 0.5237968 0.4312614 0.3544491 0.3932435 0.4309621 0.4148854  
## Attr37 Attr38 Attr39 Attr40 Attr41 Attr42 Attr43  
## 0 0.3793062 0.5799073 0.5151413 0.3673599 0.4849751 0.517940 0.4206891  
## 1 0.2835226 0.4712450 0.4317124 0.3118178 0.4591523 0.443811 0.3889325  
## Attr44 Attr45 Attr46 Attr47 Attr48 Attr49 Attr50  
## 0 0.3940295 0.4758131 0.3793110 0.3669867 0.5107666 0.5191915 0.3326468  
## 1 0.3411103 0.4081653 0.2551876 0.3378343 0.4555848 0.4636604 0.2307318  
## Attr51 Attr53 Attr55 Attr56 Attr57 Attr58 Attr59  
## 0 0.4191249 0.4827271 0.4751892 0.5007606 0.5035373 0.5030529 0.4346924  
## 1 0.5219031 0.4127394 0.3954987 0.4413784 0.4647511 0.5531534 0.4212782  
## Attr60 Attr61 Attr62 Attr63 Attr64  
## 0 0.4149889 0.3704864 0.3171707 0.4186878 0.3322471  
## 1 0.4112592 0.4002082 0.3858927 0.3344019 0.3337639  
##   
## Coefficients of linear discriminants:  
## LD1  
## Attr1 0.21219516  
## Attr3 0.12528876  
## Attr4 0.07075968  
## Attr5 0.49495144  
## Attr6 0.01438363  
## Attr9 -0.98528556  
## Attr11 0.14329381  
## Attr12 0.01461426  
## Attr13 -0.64803384  
## Attr15 -0.94942492  
## Attr16 0.01199952  
## Attr19 0.61585845  
## Attr20 0.68396427  
## Attr21 -1.16544609  
## Attr22 0.15755580  
## Attr23 0.06821156  
## Attr24 -1.75638526  
## Attr25 -0.96758736  
## Attr26 -0.87181929  
## Attr27 0.59422092  
## Attr28 -0.54207388  
## Attr29 -1.27790416  
## Attr30 -0.54702880  
## Attr31 0.09423698  
## Attr33 1.00211188  
## Attr34 -0.07297456  
## Attr35 -0.91748708  
## Attr36 0.62490513  
## Attr37 -1.10764206  
## Attr38 -2.21673057  
## Attr39 -0.52283670  
## Attr40 0.06980212  
## Attr41 -1.39816663  
## Attr42 -0.72173700  
## Attr43 -0.32854136  
## Attr44 -0.46762760  
## Attr45 -0.17169166  
## Attr46 -1.70319512  
## Attr47 -0.64824206  
## Attr48 0.91318116  
## Attr49 0.79691045  
## Attr50 0.13214754  
## Attr51 -1.02058785  
## Attr53 0.04639168  
## Attr55 -0.97163819  
## Attr56 -0.59023943  
## Attr57 -0.50559985  
## Attr58 -1.72417974  
## Attr59 -0.55228136  
## Attr60 -0.21425558  
## Attr61 -0.05239345  
## Attr62 0.56183177  
## Attr63 -1.16432976  
## Attr64 0.11283047

par(mar=c(1,1,1,1))  
plot(model\_lda)# x osa je LD1, prvi grafik za klasu 0 , drugi klasu 1

Chart, histogram

Description automatically generated

# Predikcija  
  
predict\_lda<-model\_lda%>%predict(test)  
names(predict\_lda)# class uzorka, posterior pr da uyorak pripda nekoj od klasa, x su LD

## [1] "class" "posterior" "x"

head(predict\_lda$posterior,2)

## 0 1  
## 1017 0.9458374 0.05416260  
## 4567 0.9374137 0.06258625

# Tacnost modela  
  
predicted\_lda<-ifelse(predict\_lda$posterior[,2] >0.3,"1","0")  
  
confusionMatrix(factor(predicted\_lda), test$Stecaj, positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1339 186  
## 1 311 292  
##   
## Accuracy : 0.7664   
## 95% CI : (0.7479, 0.7843)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.8444   
##   
## Kappa : 0.3865   
##   
## Mcnemar's Test P-Value : 2.665e-08   
##   
## Sensitivity : 0.6109   
## Specificity : 0.8115   
## Pos Pred Value : 0.4842   
## Neg Pred Value : 0.8780   
## Prevalence : 0.2246   
## Detection Rate : 0.1372   
## Detection Prevalence : 0.2834   
## Balanced Accuracy : 0.7112   
##   
## 'Positive' Class : 1   
##

# Najvazniji, najinformativniji prediktori  
  
  
coef(model\_lda)# Intercept i 64 prediktora

## LD1  
## Attr1 0.21219516  
## Attr3 0.12528876  
## Attr4 0.07075968  
## Attr5 0.49495144  
## Attr6 0.01438363  
## Attr9 -0.98528556  
## Attr11 0.14329381  
## Attr12 0.01461426  
## Attr13 -0.64803384  
## Attr15 -0.94942492  
## Attr16 0.01199952  
## Attr19 0.61585845  
## Attr20 0.68396427  
## Attr21 -1.16544609  
## Attr22 0.15755580  
## Attr23 0.06821156  
## Attr24 -1.75638526  
## Attr25 -0.96758736  
## Attr26 -0.87181929  
## Attr27 0.59422092  
## Attr28 -0.54207388  
## Attr29 -1.27790416  
## Attr30 -0.54702880  
## Attr31 0.09423698  
## Attr33 1.00211188  
## Attr34 -0.07297456  
## Attr35 -0.91748708  
## Attr36 0.62490513  
## Attr37 -1.10764206  
## Attr38 -2.21673057  
## Attr39 -0.52283670  
## Attr40 0.06980212  
## Attr41 -1.39816663  
## Attr42 -0.72173700  
## Attr43 -0.32854136  
## Attr44 -0.46762760  
## Attr45 -0.17169166  
## Attr46 -1.70319512  
## Attr47 -0.64824206  
## Attr48 0.91318116  
## Attr49 0.79691045  
## Attr50 0.13214754  
## Attr51 -1.02058785  
## Attr53 0.04639168  
## Attr55 -0.97163819  
## Attr56 -0.59023943  
## Attr57 -0.50559985  
## Attr58 -1.72417974  
## Attr59 -0.55228136  
## Attr60 -0.21425558  
## Attr61 -0.05239345  
## Attr62 0.56183177  
## Attr63 -1.16432976  
## Attr64 0.11283047

df\_regkoeficijenti<-data.frame(coef(model\_lda))  
impattr\_lda<-rownames\_to\_column(df\_regkoeficijenti)# package tibble  
colnames(impattr\_lda)<-c('Prediktori','RegKoeficijenti')  
impattr\_lda[order(-abs(impattr\_lda$RegKoeficijenti)),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## Prediktori RegKoeficijenti  
## 30 Attr38 -2.216731  
## 17 Attr24 -1.756385  
## 48 Attr58 -1.724180  
## 38 Attr46 -1.703195  
## 33 Attr41 -1.398167

# QDA  
  
model\_qda<-qda(Stecaj~ . -Attr18 -Attr7 -Attr14 -Attr52 -Attr32 -Attr10 -Attr2 -Attr17 -Attr8 -Attr54 , data=train)  
model\_qda

## Call:  
## qda(Stecaj ~ . - Attr18 - Attr7 - Attr14 - Attr52 - Attr32 -   
## Attr10 - Attr2 - Attr17 - Attr8 - Attr54, data = train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.7749597 0.2250403   
##   
## Group means:  
## Attr1 Attr3 Attr4 Attr5 Attr6 Attr9 Attr11  
## 0 0.5051910 0.5303958 0.3913430 0.5199276 0.4224337 0.4205297 0.5124863  
## 1 0.4333736 0.4330470 0.2867893 0.4386853 0.3924450 0.4080481 0.4363359  
## Attr12 Attr13 Attr15 Attr16 Attr19 Attr20 Attr21  
## 0 0.4835421 0.5155737 0.4842694 0.4879320 0.5093777 0.4127294 0.5058552  
## 1 0.3881322 0.4180581 0.4452528 0.3884219 0.4279199 0.4020664 0.4458569  
## Attr22 Attr23 Attr24 Attr25 Attr26 Attr27 Attr28  
## 0 0.5144711 0.5088976 0.5181492 0.6417224 0.4905319 0.4371596 0.4758296  
## 1 0.4456070 0.4308837 0.4051616 0.5091902 0.3874173 0.4160181 0.4080063  
## Attr29 Attr30 Attr31 Attr33 Attr34 Attr35 Attr36  
## 0 0.5221858 0.4710111 0.5134904 0.4167737 0.4060206 0.5089484 0.3926289  
## 1 0.4420791 0.5237968 0.4312614 0.3544491 0.3932435 0.4309621 0.4148854  
## Attr37 Attr38 Attr39 Attr40 Attr41 Attr42 Attr43  
## 0 0.3793062 0.5799073 0.5151413 0.3673599 0.4849751 0.517940 0.4206891  
## 1 0.2835226 0.4712450 0.4317124 0.3118178 0.4591523 0.443811 0.3889325  
## Attr44 Attr45 Attr46 Attr47 Attr48 Attr49 Attr50  
## 0 0.3940295 0.4758131 0.3793110 0.3669867 0.5107666 0.5191915 0.3326468  
## 1 0.3411103 0.4081653 0.2551876 0.3378343 0.4555848 0.4636604 0.2307318  
## Attr51 Attr53 Attr55 Attr56 Attr57 Attr58 Attr59  
## 0 0.4191249 0.4827271 0.4751892 0.5007606 0.5035373 0.5030529 0.4346924  
## 1 0.5219031 0.4127394 0.3954987 0.4413784 0.4647511 0.5531534 0.4212782  
## Attr60 Attr61 Attr62 Attr63 Attr64  
## 0 0.4149889 0.3704864 0.3171707 0.4186878 0.3322471  
## 1 0.4112592 0.4002082 0.3858927 0.3344019 0.3337639

# Predikcija  
  
predict\_qda<-model\_qda%>%predict(test)  
  
# Tacnost modela  
  
predicted\_qda<-ifelse(predict\_qda$posterior[,2] >0.3,"1","0")  
  
confusionMatrix(factor(predicted\_qda), test$Stecaj, positive = '1')

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1313 177  
## 1 337 301  
##   
## Accuracy : 0.7585   
## 95% CI : (0.7397, 0.7765)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.9702   
##   
## Kappa : 0.3803   
##   
## Mcnemar's Test P-Value : 2.329e-12   
##   
## Sensitivity : 0.6297   
## Specificity : 0.7958   
## Pos Pred Value : 0.4718   
## Neg Pred Value : 0.8812   
## Prevalence : 0.2246   
## Detection Rate : 0.1414   
## Detection Prevalence : 0.2998   
## Balanced Accuracy : 0.7127   
##   
## 'Positive' Class : 1   
##

# 8.2 STABLA ODLUCIVANJA  
  
# Kreiranje modela  
  
model\_tree<-rpart(Stecaj~., data=train, method = "class",control=rpart.control(cp=.0001)) # rpart koristi Ginin index kao meru imfomativnosti prediktora  
model\_tree # zvezdica oznaka terminalnog noda, simbol ispred zagrade je sta taj nod predvidja ili sta bi bilo predvidjanje kad bi bio terminalni, broj posle uslova je broj uzorka

## n= 4968   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 4968 1118 0 (0.774959742 0.225040258)   
## 2) Attr46>=0.2126472 3896 588 0 (0.849075975 0.150924025)   
## 4) Attr24>=0.4112392 2936 295 0 (0.899523161 0.100476839)   
## 8) Attr15>=0.3550753 2846 258 0 (0.909346451 0.090653549)   
## 16) Attr27< 0.413677 1011 17 0 (0.983184965 0.016815035) \*  
## 17) Attr27>=0.413677 1835 241 0 (0.868664850 0.131335150)   
## 34) Attr6< 0.5179692 1653 174 0 (0.894736842 0.105263158)   
## 68) Attr44>=0.2065673 1409 110 0 (0.921930447 0.078069553)   
## 136) Attr25>=0.3495716 1380 97 0 (0.929710145 0.070289855)   
## 272) Attr1< 0.9888833 1373 93 0 (0.932265113 0.067734887)   
## 544) Attr25>=0.7208503 574 14 0 (0.975609756 0.024390244) \*  
## 545) Attr25< 0.7208503 799 79 0 (0.901126408 0.098873592)   
## 1090) Attr49< 0.8015926 706 57 0 (0.919263456 0.080736544)   
## 2180) Attr64< 0.2799885 243 6 0 (0.975308642 0.024691358) \*  
## 2181) Attr64>=0.2799885 463 51 0 (0.889848812 0.110151188)   
## 4362) Attr5>=0.3191067 453 46 0 (0.898454746 0.101545254)   
## 8724) Attr9>=0.3467744 431 38 0 (0.911832947 0.088167053)   
## 17448) Attr46>=0.3235019 294 16 0 (0.945578231 0.054421769) \*  
## 17449) Attr46< 0.3235019 137 22 0 (0.839416058 0.160583942)   
## 34898) Attr5< 0.5059506 128 13 0 (0.898437500 0.101562500)   
## 69796) Attr21>=0.3083357 118 9 0 (0.923728814 0.076271186)   
## 139592) Attr11< 0.8639429 111 6 0 (0.945945946 0.054054054)   
## 279184) Attr29< 0.5298626 86 1 0 (0.988372093 0.011627907) \*  
## 279185) Attr29>=0.5298626 25 5 0 (0.800000000 0.200000000)   
## 558370) Attr5< 0.4573584 18 1 0 (0.944444444 0.055555556) \*  
## 558371) Attr5>=0.4573584 7 3 1 (0.428571429 0.571428571) \*  
## 139593) Attr11>=0.8639429 7 3 0 (0.571428571 0.428571429) \*  
## 69797) Attr21< 0.3083357 10 4 0 (0.600000000 0.400000000) \*  
## 34899) Attr5>=0.5059506 9 0 1 (0.000000000 1.000000000) \*  
## 8725) Attr9< 0.3467744 22 8 0 (0.636363636 0.363636364)   
## 17450) Attr35< 0.4684422 10 0 0 (1.000000000 0.000000000) \*  
## 17451) Attr35>=0.4684422 12 4 1 (0.333333333 0.666666667) \*  
## 4363) Attr5< 0.3191067 10 5 0 (0.500000000 0.500000000) \*  
## 1091) Attr49>=0.8015926 93 22 0 (0.763440860 0.236559140)   
## 2182) Attr56>=0.6187196 86 16 0 (0.813953488 0.186046512)   
## 4364) Attr24>=0.6875174 38 1 0 (0.973684211 0.026315789) \*  
## 4365) Attr24< 0.6875174 48 15 0 (0.687500000 0.312500000)   
## 8730) Attr24< 0.559295 28 2 0 (0.928571429 0.071428571) \*  
## 8731) Attr24>=0.559295 20 7 1 (0.350000000 0.650000000)   
## 17462) Attr36< 0.3494097 10 3 0 (0.700000000 0.300000000) \*  
## 17463) Attr36>=0.3494097 10 0 1 (0.000000000 1.000000000) \*  
## 2183) Attr56< 0.6187196 7 1 1 (0.142857143 0.857142857) \*  
## 273) Attr1>=0.9888833 7 3 1 (0.428571429 0.571428571) \*  
## 137) Attr25< 0.3495716 29 13 0 (0.551724138 0.448275862)   
## 274) Attr25< 0.2838024 9 1 0 (0.888888889 0.111111111) \*  
## 275) Attr25>=0.2838024 20 8 1 (0.400000000 0.600000000)   
## 550) Attr29< 0.3438767 11 4 0 (0.636363636 0.363636364) \*  
## 551) Attr29>=0.3438767 9 1 1 (0.111111111 0.888888889) \*  
## 69) Attr44< 0.2065673 244 64 0 (0.737704918 0.262295082)   
## 138) Attr51< 0.4727689 193 35 0 (0.818652850 0.181347150)   
## 276) Attr32>=0.02432115 182 28 0 (0.846153846 0.153846154)   
## 552) Attr24>=0.4681729 168 21 0 (0.875000000 0.125000000)   
## 1104) Attr38>=0.7038414 95 3 0 (0.968421053 0.031578947) \*  
## 1105) Attr38< 0.7038414 73 18 0 (0.753424658 0.246575342)   
## 2210) Attr35< 0.704134 59 10 0 (0.830508475 0.169491525)   
## 4420) Attr5< 0.6160569 52 6 0 (0.884615385 0.115384615) \*  
## 4421) Attr5>=0.6160569 7 3 1 (0.428571429 0.571428571) \*  
## 2211) Attr35>=0.704134 14 6 1 (0.428571429 0.571428571) \*  
## 553) Attr24< 0.4681729 14 7 0 (0.500000000 0.500000000) \*  
## 277) Attr32< 0.02432115 11 4 1 (0.363636364 0.636363636) \*  
## 139) Attr51>=0.4727689 51 22 1 (0.431372549 0.568627451)   
## 278) Attr5>=0.4090523 42 20 0 (0.523809524 0.476190476)   
## 556) Attr36>=0.4409852 13 2 0 (0.846153846 0.153846154) \*  
## 557) Attr36< 0.4409852 29 11 1 (0.379310345 0.620689655)   
## 1114) Attr3< 0.3799275 8 2 0 (0.750000000 0.250000000) \*  
## 1115) Attr3>=0.3799275 21 5 1 (0.238095238 0.761904762) \*  
## 279) Attr5< 0.4090523 9 0 1 (0.000000000 1.000000000) \*  
## 35) Attr6>=0.5179692 182 67 0 (0.631868132 0.368131868)   
## 70) Attr27>=0.4155606 95 6 0 (0.936842105 0.063157895) \*  
## 71) Attr27< 0.4155606 87 26 1 (0.298850575 0.701149425)   
## 142) Attr34>=0.3598193 24 3 0 (0.875000000 0.125000000) \*  
## 143) Attr34< 0.3598193 63 5 1 (0.079365079 0.920634921) \*  
## 9) Attr15< 0.3550753 90 37 0 (0.588888889 0.411111111)   
## 18) Attr37>=0.4367245 25 3 0 (0.880000000 0.120000000) \*  
## 19) Attr37< 0.4367245 65 31 1 (0.476923077 0.523076923)   
## 38) Attr29>=0.5221621 13 2 0 (0.846153846 0.153846154) \*  
## 39) Attr29< 0.5221621 52 20 1 (0.384615385 0.615384615)   
## 78) Attr2>=0.5962675 13 4 0 (0.692307692 0.307692308) \*  
## 79) Attr2< 0.5962675 39 11 1 (0.282051282 0.717948718)   
## 158) Attr3>=0.6665963 17 8 0 (0.529411765 0.470588235) \*  
## 159) Attr3< 0.6665963 22 2 1 (0.090909091 0.909090909) \*  
## 5) Attr24< 0.4112392 960 293 0 (0.694791667 0.305208333)   
## 10) Attr27< 0.4131383 535 88 0 (0.835514019 0.164485981)   
## 20) Attr27>=0.322224 447 42 0 (0.906040268 0.093959732)   
## 40) Attr34< 0.2761056 284 6 0 (0.978873239 0.021126761) \*  
## 41) Attr34>=0.2761056 163 36 0 (0.779141104 0.220858896)   
## 82) Attr35>=0.3602585 118 14 0 (0.881355932 0.118644068)   
## 164) Attr55< 0.6510892 109 9 0 (0.917431193 0.082568807) \*  
## 165) Attr55>=0.6510892 9 4 1 (0.444444444 0.555555556) \*  
## 83) Attr35< 0.3602585 45 22 0 (0.511111111 0.488888889)   
## 166) Attr58>=0.657516 21 4 0 (0.809523810 0.190476190)   
## 332) Attr47< 0.5716239 14 0 0 (1.000000000 0.000000000) \*  
## 333) Attr47>=0.5716239 7 3 1 (0.428571429 0.571428571) \*  
## 167) Attr58< 0.657516 24 6 1 (0.250000000 0.750000000) \*  
## 21) Attr27< 0.322224 88 42 1 (0.477272727 0.522727273)   
## 42) Attr25>=0.5600092 44 12 0 (0.727272727 0.272727273)   
## 84) Attr24< 0.3034204 20 1 0 (0.950000000 0.050000000) \*  
## 85) Attr24>=0.3034204 24 11 0 (0.541666667 0.458333333)   
## 170) Attr1>=0.260444 14 3 0 (0.785714286 0.214285714) \*  
## 171) Attr1< 0.260444 10 2 1 (0.200000000 0.800000000) \*  
## 43) Attr25< 0.5600092 44 10 1 (0.227272727 0.772727273) \*  
## 11) Attr27>=0.4131383 425 205 0 (0.517647059 0.482352941)   
## 22) Attr34>=0.2501256 325 112 0 (0.655384615 0.344615385)   
## 44) Attr60>=0.3331742 214 52 0 (0.757009346 0.242990654)   
## 88) Attr43>=0.152172 197 40 0 (0.796954315 0.203045685)   
## 176) Attr29< 0.4974167 142 18 0 (0.873239437 0.126760563)   
## 352) Attr22>=0.2805699 120 8 0 (0.933333333 0.066666667) \*  
## 353) Attr22< 0.2805699 22 10 0 (0.545454545 0.454545455)   
## 706) Attr50>=0.2899338 12 2 0 (0.833333333 0.166666667) \*  
## 707) Attr50< 0.2899338 10 2 1 (0.200000000 0.800000000) \*  
## 177) Attr29>=0.4974167 55 22 0 (0.600000000 0.400000000)   
## 354) Attr24< 0.3948408 37 7 0 (0.810810811 0.189189189) \*  
## 355) Attr24>=0.3948408 18 3 1 (0.166666667 0.833333333) \*  
## 89) Attr43< 0.152172 17 5 1 (0.294117647 0.705882353) \*  
## 45) Attr60< 0.3331742 111 51 1 (0.459459459 0.540540541)   
## 90) Attr55>=0.4266107 53 17 0 (0.679245283 0.320754717)   
## 180) Attr5>=0.4454732 36 6 0 (0.833333333 0.166666667)   
## 360) Attr58>=0.532438 27 0 0 (1.000000000 0.000000000) \*  
## 361) Attr58< 0.532438 9 3 1 (0.333333333 0.666666667) \*  
## 181) Attr5< 0.4454732 17 6 1 (0.352941176 0.647058824) \*  
## 91) Attr55< 0.4266107 58 15 1 (0.258620690 0.741379310)   
## 182) Attr1>=0.4702913 8 1 0 (0.875000000 0.125000000) \*  
## 183) Attr1< 0.4702913 50 8 1 (0.160000000 0.840000000)   
## 366) Attr44>=0.6915683 7 3 0 (0.571428571 0.428571429) \*  
## 367) Attr44< 0.6915683 43 4 1 (0.093023256 0.906976744) \*  
## 23) Attr34< 0.2501256 100 7 1 (0.070000000 0.930000000)   
## 46) Attr9>=0.3454193 8 3 0 (0.625000000 0.375000000) \*  
## 47) Attr9< 0.3454193 92 2 1 (0.021739130 0.978260870) \*  
## 3) Attr46< 0.2126472 1072 530 0 (0.505597015 0.494402985)   
## 6) Attr27< 0.4153159 543 189 0 (0.651933702 0.348066298)   
## 12) Attr34< 0.2679236 219 20 0 (0.908675799 0.091324201)   
## 24) Attr63>=0.1077536 205 9 0 (0.956097561 0.043902439)   
## 48) Attr39>=0.3916489 135 1 0 (0.992592593 0.007407407) \*  
## 49) Attr39< 0.3916489 70 8 0 (0.885714286 0.114285714)   
## 98) Attr27< 0.3763456 63 4 0 (0.936507937 0.063492063) \*  
## 99) Attr27>=0.3763456 7 3 1 (0.428571429 0.571428571) \*  
## 25) Attr63< 0.1077536 14 3 1 (0.214285714 0.785714286) \*  
## 13) Attr34>=0.2679236 324 155 1 (0.478395062 0.521604938)   
## 26) Attr27>=0.374127 135 43 0 (0.681481481 0.318518519)   
## 52) Attr47>=0.2297951 108 23 0 (0.787037037 0.212962963)   
## 104) Attr42< 0.4021558 43 0 0 (1.000000000 0.000000000) \*  
## 105) Attr42>=0.4021558 65 23 0 (0.646153846 0.353846154)   
## 210) Attr58>=0.6139322 34 1 0 (0.970588235 0.029411765) \*  
## 211) Attr58< 0.6139322 31 9 1 (0.290322581 0.709677419)   
## 422) Attr35>=0.492189 9 2 0 (0.777777778 0.222222222) \*  
## 423) Attr35< 0.492189 22 2 1 (0.090909091 0.909090909) \*  
## 53) Attr47< 0.2297951 27 7 1 (0.259259259 0.740740741)   
## 106) Attr16< 0.3420937 7 1 0 (0.857142857 0.142857143) \*  
## 107) Attr16>=0.3420937 20 1 1 (0.050000000 0.950000000) \*  
## 27) Attr27< 0.374127 189 63 1 (0.333333333 0.666666667)   
## 54) Attr58>=0.706078 78 39 0 (0.500000000 0.500000000)   
## 108) Attr45< 0.4020757 51 17 0 (0.666666667 0.333333333)   
## 216) Attr27>=0.2895155 34 5 0 (0.852941176 0.147058824) \*  
## 217) Attr27< 0.2895155 17 5 1 (0.294117647 0.705882353) \*  
## 109) Attr45>=0.4020757 27 5 1 (0.185185185 0.814814815)   
## 218) Attr28< 0.2631296 7 3 0 (0.571428571 0.428571429) \*  
## 219) Attr28>=0.2631296 20 1 1 (0.050000000 0.950000000) \*  
## 55) Attr58< 0.706078 111 24 1 (0.216216216 0.783783784)   
## 110) Attr31>=0.3664143 58 21 1 (0.362068966 0.637931034)   
## 220) Attr41>=0.1795433 32 14 0 (0.562500000 0.437500000)   
## 440) Attr28< 0.2731797 8 0 0 (1.000000000 0.000000000) \*  
## 441) Attr28>=0.2731797 24 10 1 (0.416666667 0.583333333)   
## 882) Attr62>=0.4626442 9 2 0 (0.777777778 0.222222222) \*  
## 883) Attr62< 0.4626442 15 3 1 (0.200000000 0.800000000) \*  
## 221) Attr41< 0.1795433 26 3 1 (0.115384615 0.884615385) \*  
## 111) Attr31< 0.3664143 53 3 1 (0.056603774 0.943396226) \*  
## 7) Attr27>=0.4153159 529 188 1 (0.355387524 0.644612476)   
## 14) Attr27>=0.4155372 273 122 0 (0.553113553 0.446886447)   
## 28) Attr60< 0.3707902 143 34 0 (0.762237762 0.237762238)   
## 56) Attr26>=0.3747739 127 23 0 (0.818897638 0.181102362)   
## 112) Attr28>=0.3280699 110 13 0 (0.881818182 0.118181818) \*  
## 113) Attr28< 0.3280699 17 7 1 (0.411764706 0.588235294) \*  
## 57) Attr26< 0.3747739 16 5 1 (0.312500000 0.687500000) \*  
## 29) Attr60>=0.3707902 130 42 1 (0.323076923 0.676923077)   
## 58) Attr5< 0.3922353 61 28 0 (0.540983607 0.459016393)   
## 116) Attr46>=0.1464865 45 15 0 (0.666666667 0.333333333)   
## 232) Attr30< 0.5025614 11 0 0 (1.000000000 0.000000000) \*  
## 233) Attr30>=0.5025614 34 15 0 (0.558823529 0.441176471)   
## 466) Attr33< 0.2817341 18 4 0 (0.777777778 0.222222222) \*  
## 467) Attr33>=0.2817341 16 5 1 (0.312500000 0.687500000) \*  
## 117) Attr46< 0.1464865 16 3 1 (0.187500000 0.812500000) \*  
## 59) Attr5>=0.3922353 69 9 1 (0.130434783 0.869565217)   
## 118) Attr33< 0.1940946 7 1 0 (0.857142857 0.142857143) \*  
## 119) Attr33>=0.1940946 62 3 1 (0.048387097 0.951612903) \*  
## 15) Attr27< 0.4155372 256 37 1 (0.144531250 0.855468750)   
## 30) Attr34>=0.2802762 123 37 1 (0.300813008 0.699186992)   
## 60) Attr60< 0.3037354 25 6 0 (0.760000000 0.240000000)   
## 120) Attr44< 0.3196116 14 0 0 (1.000000000 0.000000000) \*  
## 121) Attr44>=0.3196116 11 5 1 (0.454545455 0.545454545) \*  
## 61) Attr60>=0.3037354 98 18 1 (0.183673469 0.816326531)   
## 122) Attr54< 0.4193454 53 16 1 (0.301886792 0.698113208)   
## 244) Attr10>=0.4409814 26 12 0 (0.538461538 0.461538462)   
## 488) Attr49>=0.2257262 19 5 0 (0.736842105 0.263157895) \*  
## 489) Attr49< 0.2257262 7 0 1 (0.000000000 1.000000000) \*  
## 245) Attr10< 0.4409814 27 2 1 (0.074074074 0.925925926) \*  
## 123) Attr54>=0.4193454 45 2 1 (0.044444444 0.955555556) \*  
## 31) Attr34< 0.2802762 133 0 1 (0.000000000 1.000000000) \*

# Pruning stabla sa Optimalnom vrednosti cp, odnosno optimalne dubine (kompleksnosnti) stabla.  
  
plotcp(model\_tree)   
best\_cp <- model\_tree$cptable[which.min(model\_tree$cptable[,"xerror"]),"CP"]  
cat("Optimalna vrednost cp je ",best\_cp,"\n")

## Optimalna vrednost cp je 0.00313059

model\_tree1<-prune(model\_tree,cp=best\_cp)# ako stavim 0.01 stablo ce biti isto kao u modelu model\_tree.Biram manje kompleksno stablo  
model\_tree1

## n= 4968   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 4968 1118 0 (0.77495974 0.22504026)   
## 2) Attr46>=0.2126472 3896 588 0 (0.84907598 0.15092402)   
## 4) Attr24>=0.4112392 2936 295 0 (0.89952316 0.10047684)   
## 8) Attr15>=0.3550753 2846 258 0 (0.90934645 0.09065355)   
## 16) Attr27< 0.413677 1011 17 0 (0.98318497 0.01681503) \*  
## 17) Attr27>=0.413677 1835 241 0 (0.86866485 0.13133515)   
## 34) Attr6< 0.5179692 1653 174 0 (0.89473684 0.10526316) \*  
## 35) Attr6>=0.5179692 182 67 0 (0.63186813 0.36813187)   
## 70) Attr27>=0.4155606 95 6 0 (0.93684211 0.06315789) \*  
## 71) Attr27< 0.4155606 87 26 1 (0.29885057 0.70114943)   
## 142) Attr34>=0.3598193 24 3 0 (0.87500000 0.12500000) \*  
## 143) Attr34< 0.3598193 63 5 1 (0.07936508 0.92063492) \*  
## 9) Attr15< 0.3550753 90 37 0 (0.58888889 0.41111111)   
## 18) Attr37>=0.4367245 25 3 0 (0.88000000 0.12000000) \*  
## 19) Attr37< 0.4367245 65 31 1 (0.47692308 0.52307692)   
## 38) Attr29>=0.5221621 13 2 0 (0.84615385 0.15384615) \*  
## 39) Attr29< 0.5221621 52 20 1 (0.38461538 0.61538462)   
## 78) Attr2>=0.5962675 13 4 0 (0.69230769 0.30769231) \*  
## 79) Attr2< 0.5962675 39 11 1 (0.28205128 0.71794872) \*  
## 5) Attr24< 0.4112392 960 293 0 (0.69479167 0.30520833)   
## 10) Attr27< 0.4131383 535 88 0 (0.83551402 0.16448598)   
## 20) Attr27>=0.322224 447 42 0 (0.90604027 0.09395973)   
## 40) Attr34< 0.2761056 284 6 0 (0.97887324 0.02112676) \*  
## 41) Attr34>=0.2761056 163 36 0 (0.77914110 0.22085890)   
## 82) Attr35>=0.3602585 118 14 0 (0.88135593 0.11864407) \*  
## 83) Attr35< 0.3602585 45 22 0 (0.51111111 0.48888889)   
## 166) Attr58>=0.657516 21 4 0 (0.80952381 0.19047619) \*  
## 167) Attr58< 0.657516 24 6 1 (0.25000000 0.75000000) \*  
## 21) Attr27< 0.322224 88 42 1 (0.47727273 0.52272727)   
## 42) Attr25>=0.5600092 44 12 0 (0.72727273 0.27272727) \*  
## 43) Attr25< 0.5600092 44 10 1 (0.22727273 0.77272727) \*  
## 11) Attr27>=0.4131383 425 205 0 (0.51764706 0.48235294)   
## 22) Attr34>=0.2501256 325 112 0 (0.65538462 0.34461538)   
## 44) Attr60>=0.3331742 214 52 0 (0.75700935 0.24299065)   
## 88) Attr43>=0.152172 197 40 0 (0.79695431 0.20304569)   
## 176) Attr29< 0.4974167 142 18 0 (0.87323944 0.12676056) \*  
## 177) Attr29>=0.4974167 55 22 0 (0.60000000 0.40000000)   
## 354) Attr24< 0.3948408 37 7 0 (0.81081081 0.18918919) \*  
## 355) Attr24>=0.3948408 18 3 1 (0.16666667 0.83333333) \*  
## 89) Attr43< 0.152172 17 5 1 (0.29411765 0.70588235) \*  
## 45) Attr60< 0.3331742 111 51 1 (0.45945946 0.54054054)   
## 90) Attr55>=0.4266107 53 17 0 (0.67924528 0.32075472)   
## 180) Attr5>=0.4454732 36 6 0 (0.83333333 0.16666667) \*  
## 181) Attr5< 0.4454732 17 6 1 (0.35294118 0.64705882) \*  
## 91) Attr55< 0.4266107 58 15 1 (0.25862069 0.74137931)   
## 182) Attr1>=0.4702913 8 1 0 (0.87500000 0.12500000) \*  
## 183) Attr1< 0.4702913 50 8 1 (0.16000000 0.84000000) \*  
## 23) Attr34< 0.2501256 100 7 1 (0.07000000 0.93000000) \*  
## 3) Attr46< 0.2126472 1072 530 0 (0.50559701 0.49440299)   
## 6) Attr27< 0.4153159 543 189 0 (0.65193370 0.34806630)   
## 12) Attr34< 0.2679236 219 20 0 (0.90867580 0.09132420)   
## 24) Attr63>=0.1077536 205 9 0 (0.95609756 0.04390244) \*  
## 25) Attr63< 0.1077536 14 3 1 (0.21428571 0.78571429) \*  
## 13) Attr34>=0.2679236 324 155 1 (0.47839506 0.52160494)   
## 26) Attr27>=0.374127 135 43 0 (0.68148148 0.31851852)   
## 52) Attr47>=0.2297951 108 23 0 (0.78703704 0.21296296)   
## 104) Attr42< 0.4021558 43 0 0 (1.00000000 0.00000000) \*  
## 105) Attr42>=0.4021558 65 23 0 (0.64615385 0.35384615)   
## 210) Attr58>=0.6139322 34 1 0 (0.97058824 0.02941176) \*  
## 211) Attr58< 0.6139322 31 9 1 (0.29032258 0.70967742)   
## 422) Attr35>=0.492189 9 2 0 (0.77777778 0.22222222) \*  
## 423) Attr35< 0.492189 22 2 1 (0.09090909 0.90909091) \*  
## 53) Attr47< 0.2297951 27 7 1 (0.25925926 0.74074074)   
## 106) Attr16< 0.3420937 7 1 0 (0.85714286 0.14285714) \*  
## 107) Attr16>=0.3420937 20 1 1 (0.05000000 0.95000000) \*  
## 27) Attr27< 0.374127 189 63 1 (0.33333333 0.66666667)   
## 54) Attr58>=0.706078 78 39 0 (0.50000000 0.50000000)   
## 108) Attr45< 0.4020757 51 17 0 (0.66666667 0.33333333)   
## 216) Attr27>=0.2895155 34 5 0 (0.85294118 0.14705882) \*  
## 217) Attr27< 0.2895155 17 5 1 (0.29411765 0.70588235) \*  
## 109) Attr45>=0.4020757 27 5 1 (0.18518519 0.81481481) \*  
## 55) Attr58< 0.706078 111 24 1 (0.21621622 0.78378378) \*  
## 7) Attr27>=0.4153159 529 188 1 (0.35538752 0.64461248)   
## 14) Attr27>=0.4155372 273 122 0 (0.55311355 0.44688645)   
## 28) Attr60< 0.3707902 143 34 0 (0.76223776 0.23776224)   
## 56) Attr26>=0.3747739 127 23 0 (0.81889764 0.18110236) \*  
## 57) Attr26< 0.3747739 16 5 1 (0.31250000 0.68750000) \*  
## 29) Attr60>=0.3707902 130 42 1 (0.32307692 0.67692308)   
## 58) Attr5< 0.3922353 61 28 0 (0.54098361 0.45901639)   
## 116) Attr46>=0.1464865 45 15 0 (0.66666667 0.33333333) \*  
## 117) Attr46< 0.1464865 16 3 1 (0.18750000 0.81250000) \*  
## 59) Attr5>=0.3922353 69 9 1 (0.13043478 0.86956522)   
## 118) Attr33< 0.1940946 7 1 0 (0.85714286 0.14285714) \*  
## 119) Attr33>=0.1940946 62 3 1 (0.04838710 0.95161290) \*  
## 15) Attr27< 0.4155372 256 37 1 (0.14453125 0.85546875)   
## 30) Attr34>=0.2802762 123 37 1 (0.30081301 0.69918699)   
## 60) Attr60< 0.3037354 25 6 0 (0.76000000 0.24000000) \*  
## 61) Attr60>=0.3037354 98 18 1 (0.18367347 0.81632653) \*  
## 31) Attr34< 0.2802762 133 0 1 (0.00000000 1.00000000) \*

# Model stabla nakon pruninga  
rpart.plot(model\_tree1)

Diagram

Description automatically generated

# 5 najinfoirmativnijih prediktora sa najvecim mean decrease accuracy   
  
impattr\_tree1<-data.frame(model\_tree1$variable.importance)  
impattr\_tree1<-rownames\_to\_column(impattr\_tree1)# package tibble  
  
colnames(impattr\_tree1)<-c('Prediktori','RegKoeficijenti')  
impattr\_tree1[order(-abs(impattr\_tree1$RegKoeficijenti)),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## Prediktori RegKoeficijenti  
## 1 Attr27 256.90687  
## 2 Attr46 225.11841  
## 3 Attr34 169.56809  
## 4 Attr4 129.34563  
## 5 Attr12 92.84896

# Predikcija primenom DT  
  
predict\_tree<-data.frame(predict(model\_tree1,test,type="prob"))  
  
# Tacnost modela  
  
predicted\_tree <- ifelse(predict\_tree[,2] > 0.3, 1,0)  
confusionMatrix(factor(predicted\_tree),  
 test$Stecaj,positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1516 164  
## 1 134 314  
##   
## Accuracy : 0.86   
## 95% CI : (0.8445, 0.8744)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.5888   
##   
## Mcnemar's Test P-Value : 0.09297   
##   
## Sensitivity : 0.6569   
## Specificity : 0.9188   
## Pos Pred Value : 0.7009   
## Neg Pred Value : 0.9024   
## Prevalence : 0.2246   
## Detection Rate : 0.1476   
## Detection Prevalence : 0.2105   
## Balanced Accuracy : 0.7878   
##   
## 'Positive' Class : 1   
##

# 8.3 RANDOM FOREST  
  
  
trControl=trainControl("cv",number=5)# repeates, izostavljeno zbog performansi  
  
# Kreiranje modela  
  
model\_rf<-train(Stecaj~., data=train,method="rf",trControl= trControl,importance=TRUE, search='random')# Importance, vaznost atributa   
plot(model\_rf)

Chart, line chart

Description automatically generated

model\_rf$bestTune

## mtry  
## 2 33

cat("Optimalan broj prediktora koji ce se koristiti pri izradi stabala je" ,model\_rf$finalModel$mtry )

## Optimalan broj prediktora koji ce se koristiti pri izradi stabala je 33

model\_rf$finalModel

##   
## Call:  
## randomForest(x = x, y = y, mtry = min(param$mtry, ncol(x)), importance = TRUE, search = "random")   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 33  
##   
## OOB estimate of error rate: 11.15%  
## Confusion matrix:  
## 0 1 class.error  
## 0 3695 155 0.04025974  
## 1 399 719 0.35688730

# 5 najinefirmativnijih prediktora sa najvecim mean decrease accuracy   
  
imp\_prediktora<-as.data.frame(randomForest::importance(model\_rf$finalModel))  
  
imp\_prediktora[order(imp\_prediktora$MeanDecreaseAccuracy, decreasing = T),][1:5,]

## 0 1 MeanDecreaseAccuracy MeanDecreaseGini  
## Attr27 79.59906 70.498763 93.94252 184.66793  
## Attr46 51.86745 29.090587 58.98183 166.40522  
## Attr34 30.17643 43.416233 43.09273 120.68571  
## Attr24 23.91271 28.654569 33.23569 66.05432  
## Attr44 28.41806 7.215688 30.03441 42.41965

# Graficki prikaz informativnosti prediktora  
  
VarImpPlot(model\_rf$finalModel,type=1, cex=0.5,n.var=min(5, nrow(model\_rf$importance)), main = "Informativnost prediktora") #Mean decrease accuracy, koliko se smanji tacnost modela ako se dati prediktor izostavi

Table

Description automatically generated  
  
# Predikcija primenom RF modela  
  
predict\_rf<- predict(model\_rf, test,type = "prob")  
  
# Tacnost modela  
  
predicted\_rf <- ifelse(predict\_rf$`1`> 0.3, 1,0)  
confusionMatrix(factor(predicted\_rf),  
 test$Stecaj,positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1407 75  
## 1 243 403  
##   
## Accuracy : 0.8506   
## 95% CI : (0.8347, 0.8655)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6186   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.8431   
## Specificity : 0.8527   
## Pos Pred Value : 0.6238   
## Neg Pred Value : 0.9494   
## Prevalence : 0.2246   
## Detection Rate : 0.1894   
## Detection Prevalence : 0.3036   
## Balanced Accuracy : 0.8479   
##   
## 'Positive' Class : 1   
##

# 8.4 BOOSTING MODEL, xgboost sa xgbTree. Koristi sve corove procesora, paralelno procesiranje  
  
  
trControl=trainControl("cv",number=5)  
  
# Kreiranje modela, xgboost package. Stohastic gradient boosting   
  
model\_xgboost<-train(Stecaj~., data=train,method="xgbTree",trControl=trControl,base\_score=0.3)

# Optimalni parametri modela, nrounds je broj iteracija, eta learning rate,   
#gamma nula sledi nema regularizacije , subsumple je br. uzoraka za svako stablo, ako jedan svi  
  
model\_xgboost$bestTune

## nrounds max\_depth eta gamma colsample\_bytree min\_child\_weight subsample  
## 108 150 3 0.4 0 0.8 1 1

impattr\_xgb<-data.frame(varImp(model\_xgboost)[[1]])  
impattr\_xgb<-rownames\_to\_column(impattr\_xgb)  
  
# Najznacajniji prediktori za tacnost modela  
impattr\_xgb[(1:5),1]

## [1] "Attr27" "Attr46" "Attr34" "Attr24" "Attr5"

# Predikcija xgbTree  
  
predict\_xgboost<-predict(model\_xgboost,test,type = "prob")  
  
# Tacnost modela  
  
predicted\_xgboost <- ifelse(predict\_xgboost$`1`> 0.3, 1,0)  
confusionMatrix(factor(predicted\_xgboost),  
 test$Stecaj,positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1503 95  
## 1 147 383  
##   
## Accuracy : 0.8863   
## 95% CI : (0.872, 0.8995)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6857   
##   
## Mcnemar's Test P-Value : 0.001044   
##   
## Sensitivity : 0.8013   
## Specificity : 0.9109   
## Pos Pred Value : 0.7226   
## Neg Pred Value : 0.9406   
## Prevalence : 0.2246   
## Detection Rate : 0.1800   
## Detection Prevalence : 0.2491   
## Balanced Accuracy : 0.8561   
##   
## 'Positive' Class : 1   
##

# Trazenje optimalne vrednosti granicne vrednosti verovatnoce stecaja, za rad kako bi algorimi bili uporedivi uzete je ista vrednost  
actuals<-test$Stecaj  
  
perf1<-performance(prediction(predicted\_xgboost, actuals),"tpr", "fpr") # ROCR  
  
plot(perf1,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))

Chart, scatter chart

Description automatically generated

# 8.4.1 xgboost  
  
#define predictor and response variables in training set  
  
train\_x = data.matrix(train[, -65])  
train\_y =as.numeric(train[,65])-1  
  
#define predictor and response variables in testing set  
test\_x = data.matrix(test[, -65])  
test\_y = as.numeric(test[,65])-1  
  
#define final training and testing sets  
xgb\_train = xgb.DMatrix(data = train\_x, label = train\_y)  
xgb\_test = xgb.DMatrix(data = test\_x, label = test\_y)  
  
head(getinfo(xgb\_train,'label'))

## [1] 0 0 0 0 0 0

head(getinfo(xgb\_test,'label'))

## [1] 0 0 0 0 0 0

# Kreiranje modela  
  
watchlist = list(train=xgb\_train, test=xgb\_test)#  
train\_par<- xgb.train(data = xgb\_train, max.depth = 3, nrounds = 100, watchlist = watchlist)#training and testing root mean squared error for each round.

model\_xgb<- xgboost(data = xgb\_train, max.depth = 3, nrounds = 100, objective = "binary:logistic")# svi prediktori moraju biti numericki vektori

# 5 najinfoirmativnijih prediktora sa najvecim mean decrease accuracy   
  
importance\_matrica = xgb.importance(colnames(xgb\_train), model = model\_xgb)  
importance\_matrica[1:5,][,1]

## Feature  
## 1: Attr27  
## 2: Attr46  
## 3: Attr34  
## 4: Attr24  
## 5: Attr26

xgb.plot.importance(importance\_matrica[1:5,])  
Chart, bar chart

Description automatically generated

# Predikcija  
  
predict\_xgboost<-predict(model\_xgb, newdata=xgb\_test)  
  
# Tacnost modela  
  
predicted\_xgboost <- ifelse(predict\_xgboost> 0.3, 1,0)  
confusionMatrix(factor(predicted\_xgboost),  
 test$Stecaj,positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1471 91  
## 1 179 387  
##   
## Accuracy : 0.8731   
## 95% CI : (0.8582, 0.887)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6581   
##   
## Mcnemar's Test P-Value : 1.192e-07   
##   
## Sensitivity : 0.8096   
## Specificity : 0.8915   
## Pos Pred Value : 0.6837   
## Neg Pred Value : 0.9417   
## Prevalence : 0.2246   
## Detection Rate : 0.1819   
## Detection Prevalence : 0.2660   
## Balanced Accuracy : 0.8506   
##   
## 'Positive' Class : 1   
##

# 8.5 NB  
  
# Kreiranje modela  
  
model\_nb<-naiveBayes(Stecaj~., data=train)  
model\_nb # vidimo mean i sd za svaki numericki prediktor razlicitih klasa

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.7749597 0.2250403   
##   
## Conditional probabilities:  
## Attr1  
## Y [,1] [,2]  
## 0 0.5051910 0.1693516  
## 1 0.4333736 0.1755471  
##   
## Attr2  
## Y [,1] [,2]  
## 0 0.3351004 0.1913340  
## 1 0.4558867 0.2023241  
##   
## Attr3  
## Y [,1] [,2]  
## 0 0.5303958 0.1726119  
## 1 0.4330470 0.1823627  
##   
## Attr4  
## Y [,1] [,2]  
## 0 0.3913430 0.1792762  
## 1 0.2867893 0.1487273  
##   
## Attr5  
## Y [,1] [,2]  
## 0 0.5199276 0.1646984  
## 1 0.4386853 0.1645352  
##   
## Attr6  
## Y [,1] [,2]  
## 0 0.4224337 0.1566157  
## 1 0.3924450 0.1302354  
##   
## Attr7  
## Y [,1] [,2]  
## 0 0.5079591 0.1719241  
## 1 0.4291442 0.1816878  
##   
## Attr8  
## Y [,1] [,2]  
## 0 0.4390244 0.1635731  
## 1 0.3407992 0.1156411  
##   
## Attr9  
## Y [,1] [,2]  
## 0 0.4205297 0.1851673  
## 1 0.4080481 0.2039358  
##   
## Attr10  
## Y [,1] [,2]  
## 0 0.6502232 0.1898448  
## 1 0.5321653 0.2004092  
##   
## Attr11  
## Y [,1] [,2]  
## 0 0.5124863 0.1693083  
## 1 0.4363359 0.1840517  
##   
## Attr12  
## Y [,1] [,2]  
## 0 0.4835421 0.1604416  
## 1 0.3881322 0.1573774  
##   
## Attr13  
## Y [,1] [,2]  
## 0 0.5155737 0.1662451  
## 1 0.4180581 0.1661737  
##   
## Attr14  
## Y [,1] [,2]  
## 0 0.5075773 0.1719697  
## 1 0.4289004 0.1813426  
##   
## Attr15  
## Y [,1] [,2]  
## 0 0.4842694 0.1485943  
## 1 0.4452528 0.1935576  
##   
## Attr16  
## Y [,1] [,2]  
## 0 0.4879320 0.1532843  
## 1 0.3884219 0.1445044  
##   
## Attr17  
## Y [,1] [,2]  
## 0 0.3548055 0.1878215  
## 1 0.2435360 0.1331956  
##   
## Attr18  
## Y [,1] [,2]  
## 0 0.5075773 0.1719697  
## 1 0.4289004 0.1813426  
##   
## Attr19  
## Y [,1] [,2]  
## 0 0.5093777 0.1720409  
## 1 0.4279199 0.1692361  
##   
## Attr20  
## Y [,1] [,2]  
## 0 0.4127294 0.1824038  
## 1 0.4020664 0.1970056  
##   
## Attr21  
## Y [,1] [,2]  
## 0 0.5058552 0.1685142  
## 1 0.4458569 0.1902449  
##   
## Attr22  
## Y [,1] [,2]  
## 0 0.5144711 0.1623235  
## 1 0.4456070 0.1812406  
##   
## Attr23  
## Y [,1] [,2]  
## 0 0.5088976 0.1719719  
## 1 0.4308837 0.1586086  
##   
## Attr24  
## Y [,1] [,2]  
## 0 0.5181492 0.1714771  
## 1 0.4051616 0.1628986  
##   
## Attr25  
## Y [,1] [,2]  
## 0 0.6417224 0.1814870  
## 1 0.5091902 0.1836614  
##   
## Attr26  
## Y [,1] [,2]  
## 0 0.4905319 0.1562148  
## 1 0.3874173 0.1435916  
##   
## Attr27  
## Y [,1] [,2]  
## 0 0.4371596 0.1148630  
## 1 0.4160181 0.1262585  
##   
## Attr28  
## Y [,1] [,2]  
## 0 0.4758296 0.1402920  
## 1 0.4080063 0.1303893  
##   
## Attr29  
## Y [,1] [,2]  
## 0 0.5221858 0.1828395  
## 1 0.4420791 0.1840824  
##   
## Attr30  
## Y [,1] [,2]  
## 0 0.4710111 0.1637416  
## 1 0.5237968 0.1656765  
##   
## Attr31  
## Y [,1] [,2]  
## 0 0.5134904 0.1735608  
## 1 0.4312614 0.1693609  
##   
## Attr32  
## Y [,1] [,2]  
## 0 0.3226423 0.1943638  
## 1 0.3732305 0.2122205  
##   
## Attr33  
## Y [,1] [,2]  
## 0 0.4167737 0.1889693  
## 1 0.3544491 0.1750658  
##   
## Attr34  
## Y [,1] [,2]  
## 0 0.4060206 0.1779493  
## 1 0.3932435 0.1615792  
##   
## Attr35  
## Y [,1] [,2]  
## 0 0.5089484 0.1660360  
## 1 0.4309621 0.1813444  
##   
## Attr36  
## Y [,1] [,2]  
## 0 0.3926289 0.1919936  
## 1 0.4148854 0.2145633  
##   
## Attr37  
## Y [,1] [,2]  
## 0 0.3793062 0.2153308  
## 1 0.2835226 0.1381466  
##   
## Attr38  
## Y [,1] [,2]  
## 0 0.5799073 0.1682077  
## 1 0.4712450 0.1837438  
##   
## Attr39  
## Y [,1] [,2]  
## 0 0.5151413 0.1687567  
## 1 0.4317124 0.1705178  
##   
## Attr40  
## Y [,1] [,2]  
## 0 0.3673599 0.1685275  
## 1 0.3118178 0.1343423  
##   
## Attr41  
## Y [,1] [,2]  
## 0 0.4849751 0.1427947  
## 1 0.4591523 0.1941998  
##   
## Attr42  
## Y [,1] [,2]  
## 0 0.517940 0.1622570  
## 1 0.443811 0.1706223  
##   
## Attr43  
## Y [,1] [,2]  
## 0 0.4206891 0.1888779  
## 1 0.3889325 0.2077750  
##   
## Attr44  
## Y [,1] [,2]  
## 0 0.3940295 0.1972837  
## 1 0.3411103 0.2042132  
##   
## Attr45  
## Y [,1] [,2]  
## 0 0.4758131 0.1509237  
## 1 0.4081653 0.1541864  
##   
## Attr46  
## Y [,1] [,2]  
## 0 0.3793110 0.1860227  
## 1 0.2551876 0.1453117  
##   
## Attr47  
## Y [,1] [,2]  
## 0 0.3669867 0.1953258  
## 1 0.3378343 0.2010298  
##   
## Attr48  
## Y [,1] [,2]  
## 0 0.5107666 0.1716357  
## 1 0.4555848 0.1822768  
##   
## Attr49  
## Y [,1] [,2]  
## 0 0.5191915 0.1696952  
## 1 0.4636604 0.1783698  
##   
## Attr50  
## Y [,1] [,2]  
## 0 0.3326468 0.1976374  
## 1 0.2307318 0.1619019  
##   
## Attr51  
## Y [,1] [,2]  
## 0 0.4191249 0.1704798  
## 1 0.5219031 0.1886199  
##   
## Attr52  
## Y [,1] [,2]  
## 0 0.3250547 0.1938333  
## 1 0.3710608 0.2124428  
##   
## Attr53  
## Y [,1] [,2]  
## 0 0.4827271 0.1440711  
## 1 0.4127394 0.1359702  
##   
## Attr54  
## Y [,1] [,2]  
## 0 0.4765221 0.1415963  
## 1 0.4053296 0.1323759  
##   
## Attr55  
## Y [,1] [,2]  
## 0 0.4751892 0.1527275  
## 1 0.3954987 0.1212608  
##   
## Attr56  
## Y [,1] [,2]  
## 0 0.5007606 0.1622326  
## 1 0.4413784 0.1742126  
##   
## Attr57  
## Y [,1] [,2]  
## 0 0.5035373 0.1572369  
## 1 0.4647511 0.1642798  
##   
## Attr58  
## Y [,1] [,2]  
## 0 0.5030529 0.1638999  
## 1 0.5531534 0.1766225  
##   
## Attr59  
## Y [,1] [,2]  
## 0 0.4346924 0.1455193  
## 1 0.4212782 0.1415538  
##   
## Attr60  
## Y [,1] [,2]  
## 0 0.4149889 0.1464315  
## 1 0.4112592 0.1518942  
##   
## Attr61  
## Y [,1] [,2]  
## 0 0.3704864 0.1831099  
## 1 0.4002082 0.2100728  
##   
## Attr62  
## Y [,1] [,2]  
## 0 0.3171707 0.1889625  
## 1 0.3858927 0.2167010  
##   
## Attr63  
## Y [,1] [,2]  
## 0 0.4186878 0.1865043  
## 1 0.3344019 0.1764350  
##   
## Attr64  
## Y [,1] [,2]  
## 0 0.3322471 0.1631891  
## 1 0.3337639 0.1637390

train%>%filter(Stecaj=="1")%>%summarise(mean(Attr1),sd(Attr1))

## mean(Attr1) sd(Attr1)  
## 1 0.4333736 0.1755471

# Predikcija  
  
predict\_nbp<-predict(model\_nb,test,type='raw')[,2]  
  
# Tacnost modela  
  
predicted\_nb <- ifelse(predict\_nbp> 0.3, 1,0)  
  
confusionMatrix(factor(predicted\_nb),  
 test$Stecaj,positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1051 120  
## 1 599 358  
##   
## Accuracy : 0.6621   
## 95% CI : (0.6416, 0.6822)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.2846   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.7490   
## Specificity : 0.6370   
## Pos Pred Value : 0.3741   
## Neg Pred Value : 0.8975   
## Prevalence : 0.2246   
## Detection Rate : 0.1682   
## Detection Prevalence : 0.4497   
## Balanced Accuracy : 0.6930   
##   
## 'Positive' Class : 1   
##

# Model sa PC  
  
model\_nbpc<-naiveBayes(Stecaj~., data=train\_pc)  
  
# Predikcija  
  
#predict\_nbpc<-predict(model\_nbpc,test\_pc), daje klase  
predict\_nbpcp<-predict(model\_nbpc,test\_pc,type='raw')[,2]  
  
  
# Tacnost modela  
  
predicted\_nbp <- ifelse(predict\_nbpcp> 0.3, 1,0)  
  
confusionMatrix(factor(predicted\_nbp),  
 test\_pc$Stecaj,positive = "1")# cutoff 0.3

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1280 214  
## 1 370 264  
##   
## Accuracy : 0.7256   
## 95% CI : (0.7061, 0.7444)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.294   
##   
## Mcnemar's Test P-Value : 1.418e-10   
##   
## Sensitivity : 0.5523   
## Specificity : 0.7758   
## Pos Pred Value : 0.4164   
## Neg Pred Value : 0.8568   
## Prevalence : 0.2246   
## Detection Rate : 0.1241   
## Detection Prevalence : 0.2979   
## Balanced Accuracy : 0.6640   
##   
## 'Positive' Class : 1   
##

# 8.6 KNN model  
  
  
#kreiranje sintaticka ispravnih imena, za vrednosti karakter vektora  
  
levels(train$Stecaj)<-make.names(levels(train$Stecaj))  
levels(test$Stecaj)<-make.names(levels(test$Stecaj))  
  
# KNN 1, koristimo Accurancy za izbor optimalnog broja k  
  
# Primena cv, za trazenje optimalne vrednosti k  
  
trControl<-trainControl(method = "repeatedcv", number = 5,repeats=3)  
  
# Kreiranje modela   
  
model\_knn1<-train(Stecaj~., data = train, method='knn',   
 trControl=trControl,  
 tuneLength=10 )# teuneLength broj mogucih vrednosti K   
  
  
model\_knn1$bestTune# optimalna vrednost suseda k

## k  
## 5 13

plot(model\_knn1,xlab="Broj suseda K",ylab='Tacnost u funkciji od K', type='b',col='red',lwd=1.5,pch='o')

Chart, line chart

Description automatically generated  
  
# Najznacajniji prediktori za tacnost modela  
  
impattr\_knn1<-data.frame(varImp(model\_knn1)[[1]])  
impattr\_knn1<-rownames\_to\_column(impattr\_knn1)# package tibble  
  
colnames(impattr\_knn1)<-c('Prediktori','RegKoeficijenti')  
impattr\_knn1[order(-abs(impattr\_knn1$RegKoeficijenti)),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## Prediktori RegKoeficijenti NA  
## 46 Attr46 100.00000 100.00000  
## 26 Attr26 90.83498 90.83498  
## 16 Attr16 90.28367 90.28367  
## 8 Attr8 89.43393 89.43393  
## 17 Attr17 88.48449 88.48449

#Predikcija  
  
predict\_knn1<-predict(model\_knn1,test, type='prob')[,2] # X1 stecaj  
  
  
# Tacnost modela  
  
predicted1 = ifelse(predict\_knn1 > 0.3, 1,0)  
  
confusionMatrix(factor(predicted1),  
 factor(actuals), positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1275 174  
## 1 375 304  
##   
## Accuracy : 0.742   
## 95% CI : (0.7229, 0.7605)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.9999   
##   
## Kappa : 0.3556   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.6360   
## Specificity : 0.7727   
## Pos Pred Value : 0.4477   
## Neg Pred Value : 0.8799   
## Prevalence : 0.2246   
## Detection Rate : 0.1429   
## Detection Prevalence : 0.3191   
## Balanced Accuracy : 0.7044   
##   
## 'Positive' Class : 1   
##

# 8.6.1 KNN2, koristimo ROC za izbor k  
  
trControl<-trainControl(method = "repeatedcv", number = 5,repeats=3,classProbs = T,summaryFunction = twoClassSummary)  
  
# Kreiranje modela   
  
model\_knn2<-train(Stecaj~., data = train, method='knn',tuneLength=10,   
 trControl=trControl,metric="ROC")  
  
plot(model\_knn2,xlab="Broj suseda K",ylab='Tacnost u funkciji K', type='b',col='black',lwd=1.5,pch='o')

Chart, line chart

Description automatically generated

model\_knn2$bestTune

## k  
## 10 23

# Vaznost prediktora  
  
  
impattr\_knn2<-data.frame(varImp(model\_knn2)[[1]])  
impattr\_knn2<-rownames\_to\_column(impattr\_knn2)# package tibble  
  
colnames(impattr\_knn2)<-c('Prediktori','RegKoeficijenti')  
impattr\_knn2[order(-abs(impattr\_knn2$RegKoeficijenti)),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## Prediktori RegKoeficijenti NA  
## 46 Attr46 100.00000 100.00000  
## 26 Attr26 90.83498 90.83498  
## 16 Attr16 90.28367 90.28367  
## 8 Attr8 89.43393 89.43393  
## 17 Attr17 88.48449 88.48449

# Predikcija  
  
predict\_knn2<-predict(model\_knn2,test,type='prob')[,2]  
  
  
# Trazenje optimalne vrednosti granicne vrednosti verovatnoce stecaja  
  
  
perf2<-performance(prediction(predict\_knn2, actuals),"tpr", "fpr")  
  
plot(perf2,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))

Chart, line chart

Description automatically generated

# Tacnost modela, za usvojenu granicunu vrednost verovatnoce 0.3 za pozitivan ishod, kako bi modeli bili uporedivi  
  
  
predicted\_knn2 <- ifelse(predict\_knn2 > 0.3, 1,0)  
confusionMatrix(factor(predicted\_knn2),  
 factor(actuals),positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1306 189  
## 1 344 289  
##   
## Accuracy : 0.7495   
## 95% CI : (0.7305, 0.7678)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.9978   
##   
## Kappa : 0.3552   
##   
## Mcnemar's Test P-Value : 2.55e-11   
##   
## Sensitivity : 0.6046   
## Specificity : 0.7915   
## Pos Pred Value : 0.4566   
## Neg Pred Value : 0.8736   
## Prevalence : 0.2246   
## Detection Rate : 0.1358   
## Detection Prevalence : 0.2975   
## Balanced Accuracy : 0.6981   
##   
## 'Positive' Class : 1   
##

# 8.7 SUPORT VECTOR MACHINE linear  
  
trControl<-trainControl(method = "cv", number = 5, summaryFunction=twoClassSummary,classProbs=TRUE)#za izbor modela na bazi ROC  
  
# Kreiranje modela, SVM sa linearnom Kernel funkcijom  
  
model\_svmlin <- train(Stecaj ~., data = train, method = "svmLinear", trControl=trControl,metric="ROC")  
  
# Prikaz modela  
  
model\_svmlin # c=1, za lin metod.Ova konstanta predstavlja cost, misklasifikacije. Tako se sa vecom

## Support Vector Machines with Linear Kernel   
##   
## 4968 samples  
## 64 predictor  
## 2 classes: 'X0', 'X1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 3975, 3974, 3975, 3974, 3974   
## Resampling results:  
##   
## ROC Sens Spec   
## 0.7989844 0.9568831 0.2925408  
##   
## Tuning parameter 'C' was held constant at a value of 1

# vrednosti konstante C, verovatnoca pogresne klasifikacije je manja.  
  
  
impattr\_svmlin<-data.frame(varImp(model\_svmlin)[[1]])  
impattr\_svmlin<-rownames\_to\_column(impattr\_svmlin)  
  
# Najznacajniji prediktori za tacnost modela  
  
impattr\_svmlin[order(-abs(impattr\_svmlin[,2])),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## rowname X0 X1  
## 46 Attr46 100.00000 100.00000  
## 26 Attr26 90.83498 90.83498  
## 16 Attr16 90.28367 90.28367  
## 8 Attr8 89.43393 89.43393  
## 17 Attr17 88.48449 88.48449

# Predikcija  
  
predict\_svmlin<-predict(model\_svmlin,test,type='prob')[,2]  
  
# Optimalne cutoff vrednosti verovatnoce  
  
perf\_svmlin<-performance(prediction(predict\_svmlin, actuals),"tpr", "fpr")  
  
plot(perf\_svmlin,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))  
  
Chart, line chart, scatter chart

Description automatically generated  
# Tacnost modela, za usvojenu granicnu vrednost verovatnoce 0.3, za pozitivan ishod  
  
predicted\_svmlin <- ifelse( predict\_svmlin> 0.3, 1,0)  
  
confusionMatrix(factor(predicted\_svmlin),  
 factor(actuals),positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1366 192  
## 1 284 286  
##   
## Accuracy : 0.7763   
## 95% CI : (0.758, 0.7939)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.4708   
##   
## Kappa : 0.3989   
##   
## Mcnemar's Test P-Value : 3.033e-05   
##   
## Sensitivity : 0.5983   
## Specificity : 0.8279   
## Pos Pred Value : 0.5018   
## Neg Pred Value : 0.8768   
## Prevalence : 0.2246   
## Detection Rate : 0.1344   
## Detection Prevalence : 0.2679   
## Balanced Accuracy : 0.7131   
##   
## 'Positive' Class : 1   
##

# 8.7.1 SVM Radial. Kreiranje modela, SVM sa nelinearnom kernel funkcijom (Radial)  
  
model\_svmradial <- train(Stecaj ~., data = train, method = "svmRadial", trControl=trControl, tuneLength = 5,metric="ROC")  
  
# Hiperparametri sigma and C, sa kojima se postize maksimalna tacnost modela  
  
model\_svmradial$bestTune

## sigma C  
## 5 0.01056986 4

# Prikaz modela  
  
plot(model\_svmradial)

Chart, line chart

Description automatically generated

impattr\_svmradial<-data.frame(varImp(model\_svmradial)[[1]])  
impattr\_svmradial<-rownames\_to\_column(impattr\_svmradial)  
  
# Najznacajniji prediktori za tacnost modela  
  
impattr\_svmradial[order(-abs(impattr\_svmradial[,2])),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## rowname X0 X1  
## 46 Attr46 100.00000 100.00000  
## 26 Attr26 90.83498 90.83498  
## 16 Attr16 90.28367 90.28367  
## 8 Attr8 89.43393 89.43393  
## 17 Attr17 88.48449 88.48449

# Predikcija  
  
predict\_svmradial<-predict(model\_svmradial,test,type='prob')[,2]  
  
# Tacnost modela, za usvojenu granicunu vrednost verovatnoce 0.3 za pozitivan ishod  
  
predicted\_svmradial <- ifelse( predict\_svmradial> 0.3, 1,0)  
  
confusionMatrix(factor(predicted\_svmradial),  
 factor(actuals),positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1410 175  
## 1 240 303  
##   
## Accuracy : 0.805   
## 95% CI : (0.7875, 0.8216)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 0.0004916   
##   
## Kappa : 0.4659   
##   
## Mcnemar's Test P-Value : 0.0016801   
##   
## Sensitivity : 0.6339   
## Specificity : 0.8545   
## Pos Pred Value : 0.5580   
## Neg Pred Value : 0.8896   
## Prevalence : 0.2246   
## Detection Rate : 0.1424   
## Detection Prevalence : 0.2552   
## Balanced Accuracy : 0.7442   
##   
## 'Positive' Class : 1   
##

# 8.7.2 SVM Poli  
  
# Kreiranje modela, SVM sa nelinearnom kernel funkcijom (polinomalna)  
  
model\_svmpoli <- train(Stecaj ~., data = train, method = "svmPoly", metric="ROC",trControl=trControl)

## line search fails -1.90229 -0.3396459 1.162875e-05 -7.221635e-06 -3.483131e-08 -1.292535e-08 -3.117023e-13line search fails -1.922909 -0.3661386 1.484162e-05 -9.193946e-06 -4.510345e-08 -1.738208e-08 -5.095983e-13

# Hiperparametri sigma and C, sa kojima se postize maksimalna tacnost modela  
  
model\_svmpoli$bestTune

## degree scale C  
## 24 3 0.01 1

# Prikaz modela  
plot(model\_svmpoli)

Chart, line chart

Description automatically generated

impattr\_svmpoli<-data.frame(varImp(model\_svmpoli)[[1]])  
impattr\_svmpoli<-rownames\_to\_column(impattr\_svmpoli)  
  
# Najznacajniji prediktori za tacnost modela  
  
impattr\_svmpoli[order(-abs(impattr\_svmpoli[,2])),][c(1:5),]# 5 najvecih reg koeficijenata po apsolutnoj vrednosti

## rowname X0 X1  
## 46 Attr46 100.00000 100.00000  
## 26 Attr26 90.83498 90.83498  
## 16 Attr16 90.28367 90.28367  
## 8 Attr8 89.43393 89.43393  
## 17 Attr17 88.48449 88.48449

# Predikcija  
  
predict\_svmpoli<-predict(model\_svmpoli,test,type='prob')[,2]  
  
# Tacnost modela, za usvojenu granicunu vrednost verovatnoce 0.3 za pozitivan ishod  
  
  
predicted\_svmpoli <- ifelse( predict\_svmpoli> 0.3, 1,0)  
confusionMatrix(factor(predicted\_svmpoli),  
 factor(actuals),positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1424 181  
## 1 226 297  
##   
## Accuracy : 0.8087   
## 95% CI : (0.7914, 0.8253)  
## No Information Rate : 0.7754   
## P-Value [Acc > NIR] : 9.725e-05   
##   
## Kappa : 0.4687   
##   
## Mcnemar's Test P-Value : 0.02918   
##   
## Sensitivity : 0.6213   
## Specificity : 0.8630   
## Pos Pred Value : 0.5679   
## Neg Pred Value : 0.8872   
## Prevalence : 0.2246   
## Detection Rate : 0.1396   
## Detection Prevalence : 0.2458   
## Balanced Accuracy : 0.7422   
##   
## 'Positive' Class : 1   
##

# 9. Z SCORE predikcija   
  
# Podaci za z score, tacnost predikcije, nisu scaled  
  
test\_noscale<-stecaj\_bezekstrema\_bezNA[-split,c(1,4,7,8,9,10)]  
  
  
# X3 je working capitalk/TA, b je X6 je retain earn/TA, c je X7 je ebita/TA, d je X8 je book equity/TL i e je X9 je sales/TA  
# z funkcija ce vratit 1 za kompanije u stecaju  
  
z\_funkcija<-function(x){  
   
 a=x[1]  
 b=x[2]  
 c=x[3]  
 d=x[4]  
 e=x[5]  
 z=(1.2\*a+1.4\*b+3.3\*c+0.6\*d+1\*e)  
   
 if (z<1.81) {  
   
 return(1)# komanija u stecaju  
   
 }else {  
   
 return(0) # 0 oznacava komanije za koje po modelu ne proizilazi stecaj ili se ne moze odrediti  
 }  
   
}  
  
  
z\_score<-apply(test\_noscale[,-1],1,z\_funkcija)  
  
table(test\_noscale$Stecaj,factor(z\_score))

##   
## 0 1  
## 0 1234 416  
## 1 214 264

confusionMatrix(test\_noscale$Stecaj,  
 factor(z\_score),positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1234 416  
## 1 214 264  
##   
## Accuracy : 0.7039   
## 95% CI : (0.684, 0.7233)  
## No Information Rate : 0.6805   
## P-Value [Acc > NIR] : 0.01034   
##   
## Kappa : 0.261   
##   
## Mcnemar's Test P-Value : 1.166e-15   
##   
## Sensitivity : 0.3882   
## Specificity : 0.8522   
## Pos Pred Value : 0.5523   
## Neg Pred Value : 0.7479   
## Prevalence : 0.3195   
## Detection Rate : 0.1241   
## Detection Prevalence : 0.2246   
## Balanced Accuracy : 0.6202   
##   
## 'Positive' Class : 1   
##

# B emerging market  
  
z\_funkcija<-function(x){  
   
 a=x[1]  
 b=x[2]  
 c=x[3]  
 d=x[4]  
   
 z=(3.25 + 6.56\*a+3.26\*b+6.72 \*c+1.05\*d)  
   
 if (z<1.1) {  
   
 return(1)# kompanija u stecaju  
   
 }else {  
   
 return(0) # 0 oznacava komanije za koje po modelu ne proizilazi stecaj ili se ne moze odrediti  
 }  
   
}  
  
  
  
z\_score<-apply(test\_noscale[, -c(1,6)],1,z\_funkcija)  
  
table(test\_noscale$Stecaj,factor(z\_score))

##   
## 0 1  
## 0 1620 30  
## 1 453 25

confusionMatrix(test\_noscale$Stecaj,  
 factor(z\_score),positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1620 30  
## 1 453 25  
##   
## Accuracy : 0.773   
## 95% CI : (0.7546, 0.7907)  
## No Information Rate : 0.9742   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0498   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.45455   
## Specificity : 0.78148   
## Pos Pred Value : 0.05230   
## Neg Pred Value : 0.98182   
## Prevalence : 0.02585   
## Detection Rate : 0.01175   
## Detection Prevalence : 0.22462   
## Balanced Accuracy : 0.61801   
##   
## 'Positive' Class : 1   
##

#######################################################################  
  
  
  
# Vracanje warn messages  
options(warn = tek\_seting)  
  
# Kraj

Lin Regresija

The gradient, represented by the blue arrows, denotes the direction of greatest change of a scalar function. The values of the function are represented in greyscale and increase in value from white (low) to dark (high).

A picture containing diagram

Description automatically generated

he gradient vector can be interpreted as the "direction and rate of fastest increase"If the gradient of a function is non-zero at a point *p*, the direction of the gradient is the direction in which the function increases most quickly from *p*, and the [magnitude](https://en.wikipedia.org/wiki/Magnitude_(mathematics)) of the gradient is the rate of increase in that direction,

he gradient (or gradient vector field) of a scalar function *f*(*x*1, *x*2, *x*3, …, *xn*) is denoted ∇*f* or ∇→*f*

[How To Find The Directional Derivative and The Gradient Vector - YouTube](https://www.youtube.com/watch?v=CnVes9TdnPo)

[Probability VS Likelihood. This blog aims to explain the… | by Harshit Dawar | The Startup | Medium](https://medium.com/swlh/probability-vs-likelihood-cdac534bf523)

[12 Types of Neural Networks Activation Functions: How to Choose? (v7labs.com)](https://www.v7labs.com/blog/neural-networks-activation-functions)

[What are Neural Networks? | IBM](https://www.ibm.com/cloud/learn/neural-networks)

[Neural Networks Pt. 1: Inside the Black Box - YouTube](https://www.youtube.com/watch?v=CqOfi41LfDw&list=PLblh5JKOoLUIxGDQs4LFFD--41Vzf-ME1&index=1)

[ANN (Artificial Neural Network) Models in R: Code & Examples on How to Build Your NN | DataCamp](https://www.datacamp.com/tutorial/neural-network-models-r)

’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’

[Gradient Boosting Classification explained through Python | by Vagif Aliyev | Towards Data Science](https://towardsdatascience.com/gradient-boosting-classification-explained-through-python-60cc980eeb3d)

Eigen vlale [ML | Linear Discriminant Analysis - GeeksforGeeks](https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/) ,[Microsoft PowerPoint - l10.ppt (iitkgp.ac.in)](http://www.facweb.iitkgp.ac.in/~sudeshna/courses/ml08/lda.pdf)

### [Using machine learning, neural networks, and statistics to predict corporate bankruptcy](https://onlinelibrary.wiley.com/doi/abs/10.1111/0885-9507.00062)

### [A Brief Introduction to Supervised Learning | by Aidan Wilson | Towards Data Science](https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590)

* [Artificial Intelligence, Machine Learning and Big Data in Finance: Opportunities, Challenges, and Implications for Policy Makers (oecd.org)](https://www.oecd.org/finance/financial-markets/Artificial-intelligence-machine-learning-big-data-in-finance.pdf)

[Commonly Used Machine Learning Algorithms | Data Science (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/)

[A guide to the types of machine learning algorithms | SAS UK](https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html)

[get (unibl.org)](https://fedora.unibl.org/fedora/get/o:1755/bdef:Content/get)

[n18.pdf (eecs189.org)](https://www.eecs189.org/static/notes/n18.pdf)

<https://mathformachines.com/images/quadratic-linear.png>

[22be93da16dc0e708bfcda1eafd9a1fdc62a.pdf](file:///C:\Users\kkolaro\Desktop\Doktorat\Books\22be93da16dc0e708bfcda1eafd9a1fdc62a.pdf)

[Six Varieties of Gaussian Discriminant Analysis (mathformachines.com)](https://mathformachines.com/posts/discriminant-analysis/)

[Machine Learning Tutorial.pdf (ntu.edu.tw)](http://disp.ee.ntu.edu.tw/~pujols/Machine%20Learning%20Tutorial.pdf) , dobri elementi za pocetak

* + [07.pdf (berkeley.edu)](https://people.eecs.berkeley.edu/~jrs/189/lec/07.pdf)

XGBoost stands for “Extreme Gradient Boosting”, where the term “Gradient Boosting” originates from the paper Greedy Function Approximation: A Gradient Boosting Machine, by Friedman.

[Understand your dataset with XGBoost — xgboost 1.4.0-SNAPSHOT documentation](https://xgboost.readthedocs.io/en/latest/R-package/discoverYourData.html)

[17. Learning: Boosting - YouTube](https://www.youtube.com/watch?v=UHBmv7qCey4)

[Regression Trees · UC Business Analytics R Programming Guide (uc-r.github.io)](https://uc-r.github.io/regression_trees)

[Decision Trees in R - DataCamp](https://www.datacamp.com/community/tutorials/decision-trees-R)

Primenom SVM algoritama

[Support Vector Machine (SVM) Algorithm - Javatpoint](https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm)

[Support-vector machine - Wikipedia](https://en.wikipedia.org/wiki/Support-vector_machine)

[SUPPORT VECTOR MACHINES(SVM). Introduction: All you need to know… | by Ajay Yadav | Towards Data Science](https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589)

[Building A Neural Net from Scratch Using R - Part 1 · R Views (rstudio.com)](https://rviews.rstudio.com/2020/07/20/shallow-neural-net-from-scratch-using-r-part-1/)

[SVM | Support Vector Machine Algorithm in Machine Learning (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/)

[Support Vector Machine (SVM) Algorithm - Javatpoint](https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm)

[linear algebra - Why is the SVM margin equal to $\frac{2}{\|\mathbf{w}\|}$? - Mathematics Stack Exchange](https://math.stackexchange.com/questions/1305925/why-is-the-svm-margin-equal-to-frac2-mathbfw)

[Support-vector machine - Wikipedia](https://en.wikipedia.org/wiki/Support-vector_machine)

[16. Learning: Support Vector Machines - YouTube](https://www.youtube.com/watch?v=_PwhiWxHK8o)

**Excellent interview**

[Gaussian Discriminant Analysis an example of Generative Learning Algorithms | by Parag Radke | Towards Data Science](https://towardsdatascience.com/gaussian-discriminant-analysis-an-example-of-generative-learning-algorithms-2e336ba7aa5c)

[Gaussian Discriminant Analysis. Generative learning algorithm | by Golam Mortuza | Medium](https://gmortuza.medium.com/gaussian-discriminative-analysis-e5701f12f3e9)

[Multivariate normal distribution (peterroelants.github.io)](https://peterroelants.github.io/posts/multivariate-normal-primer/)

[Gaussian Discriminant Analysis an example of Generative Learning Algorithms | by Parag Radke | Towards Data Science](https://towardsdatascience.com/gaussian-discriminant-analysis-an-example-of-generative-learning-algorithms-2e336ba7aa5c)

**E**

**Between**

[Naive Bayes Classifiers - GeeksforGeeks](https://www.geeksforgeeks.org/naive-bayes-classifiers/)

[Learn Naive Bayes Algorithm | Naive Bayes Classifier Examples (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/)

[Lecture 5: Bayes Classifier and Naive Bayes (cornell.edu)](https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote05.html)

[Naïve Bayes Algorithm: Everything You Need to Know - KDnuggets](https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html#:~:text=The%20Zero%2DFrequency%20Problem,all%20the%20probabilities%20are%20multiplied.)

[understanding-machine-learning-theory-algorithms.pdf](file:///C:\Users\kkolaro\Desktop\Doktorat\uvod\understanding-machine-learning-theory-algorithms.pdf) ,24.7

[Naive Bayes classifier - Wikipedia](https://en.wikipedia.org/wiki/Naive_Bayes_classifier) argmax

[cs229-notes2.pdf](file:///C:\Users\kkolaro\Desktop\Doktorat\stanford\cs229-notes2.pdf)

**Super , R na jednom mestu** [Chi-Square Test in R | Explore the Examples and Essential concepts! - DataFlair (data-flair.training)](https://data-flair.training/blogs/chi-square-test-in-r/#:~:text=Chi%2DSquare%20test%20in%20R%20is%20a%20statistical%20method%20which,Green%2C%20Yes%2FNo%20etc.)

1. [Analytics Vidhya](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/) [↑](#footnote-ref-2)
2. Izvor:[Gartner](https://www.gartner.com/smarterwithgartner/why-data-and-analytics-are-key-to-digital-transformation) [↑](#footnote-ref-3)
3. Izvor:[Gantz, J. i Reinsel, D. IDC (2012)](https://www.cs.princeton.edu/courses/archive/spring13/cos598C/idc-the-digital-universe-in-2020.pdf) [↑](#footnote-ref-4)
4. [Artificial Intelligence (AI): What it is and why it matters | SAS](https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html) [↑](#footnote-ref-5)
5. Izvor: [Researchgate](https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051) [↑](#footnote-ref-6)
6. Izvor: [K Means Clustering. In the previous story we understood… | by Ayush Kalla | DataDrivenInvestor](https://medium.datadriveninvestor.com/k-means-clustering-4a700d4a4720) [↑](#footnote-ref-7)
7. Izvor: [What is reinforcement learning](https://medium.com/@vishnuvijayanpv/what-is-reinforcement-learning-e5dc827c8564) [↑](#footnote-ref-8)
8. Izvor: Porter i Gujarati (2009 ).*Basic* *Econometrics,* s. 37 [↑](#footnote-ref-9)
9. Izvor: Porter i Gujarati (2009 ).*Basic* *Econometrics,* s. 45 [↑](#footnote-ref-10)
10. Izvor: James et al., 2013 [↑](#footnote-ref-11)
11. Izvor : [Linear & Quadratic Discriminant Analysis · UC Business Analytics R Programming Guide (uc-r.github.io)](http://uc-r.github.io/discriminant_analysis) [↑](#footnote-ref-12)
12. Izvor: [Dimensionality-Reduction LDA](https://saltfarmer.github.io/blog/machine%20learning/Dimensionality-Reduction/) [↑](#footnote-ref-13)
13. [Scientific Diagram (researchgate.net)](Within-and%20between-class%20variances.%20|%20Download%20Scientific%20Diagram%20(researchgate.net)) [↑](#footnote-ref-14)
14. Izvor: [The Kernel Trick in Support Vector Classification | by Drew Wilimitis | Towards Data Science](https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f) [↑](#footnote-ref-15)
15. Izvor: [Decision Tree Algorithm, Explained - KDnuggets](https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html) [↑](#footnote-ref-16)
16. Izvor: Foster Provost i Tom Fawcett,2013. *Data Science for Business*. s. 52 [↑](#footnote-ref-17)
17. Izvor: [(Wikipedia, Bagging)](https://en.wikipedia.org/wiki/Bootstrap_aggregating) [↑](#footnote-ref-18)
18. Izvor: [Edureca: Random forest in R](https://www.youtube.com/watch?v=gmmV4drPTS4&list=PL9ooVrP1hQOGR57Y4g1LFhn1JXVgn1lkX&index=17) [↑](#footnote-ref-19)
19. Izvor: [AdaBoost Classifier Algorithms using Python Sklearn Tutorial | DataCamp](https://www.datacamp.com/tutorial/adaboost-classifier-python) [↑](#footnote-ref-20)
20. Izvor: [A Beginners Guide To Boosting Machine Learning Algorithms | Edureka](https://www.edureka.co/blog/boosting-machine-learning/) [↑](#footnote-ref-21)
21. Izvor: [Ml-gradient-boosting](https://www.geeksforgeeks.org/ml-gradient-boosting/) [↑](#footnote-ref-22)
22. Izvor: <https://commons.wikimedia.org/wiki/File:Roc_curve.svg> [↑](#footnote-ref-23)
23. **Izvor:** : <http://slideplayer.com/slide/6394283/> [↑](#footnote-ref-24)
24. Izvor: [Nlpca.org](http://www.nlpca.org/fig_pca_principal_component_analysis.png) [↑](#footnote-ref-25)
25. Izvor: [Logistic Regression versus Decision Trees – The Official Blog of BigML.com](https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees/) [↑](#footnote-ref-26)
26. Izvor: [UCI Machine Learning Repository: About](http://archive.ics.uci.edu/ml/about.html) [↑](#footnote-ref-27)
27. Izvor: <https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data> [↑](#footnote-ref-28)
28. Izvor: [EMIS](https://www.emis.com/) [↑](#footnote-ref-29)
29. Izvor: [kkolaro/doktorat (github.com)](https://github.com/kkolaro/doktorat) [↑](#footnote-ref-30)
30. Izvor: [kkolaro/doktorat (github.com)](https://github.com/kkolaro/doktorat) [↑](#footnote-ref-31)
31. Izvor: [kkolaro/doktorat (github.com)](https://github.com/kkolaro/doktorat) [↑](#footnote-ref-32)
32. Izvor: [kkolaro/doktorat (github.com)](https://github.com/kkolaro/doktorat) [↑](#footnote-ref-33)
33. Izvor: [kkolaro/doktorat (github.com)](https://github.com/kkolaro/doktorat) [↑](#footnote-ref-34)