Upotreba ROC krivulja u evaluaciji performansi sustava u inženjerstvu

Došen, Leonarda

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij strojarstva

Završni rad

UPOTREBA ROC KRIVULJA U EVALUACIJI PERFORMANSI SUSTAVA U INŽENJERSTVU

Rijeka, srpanj 2024.

Leonarda Došen

0069092298

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij strojarstva

Završni rad

UPOTREBA ROC KRIVULJA U EVALUACIJI PERFORMANSI SUSTAVA U INŽENJERSTVU

Mentorica: izv. prof. dr. sc. Loredana Simčić

Komentorica: prof. dr. sc. Nelida Črnjarić

Rijeka, srpanj 2024.

Leonarda Došen

0069092298

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 15.03.2024.

Zavod: Zavod za matematiku, fiziku i strane jezike

Predmet: Inženjerska statistika

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Leonarda Došen (0069092298)**

Studij: Sveučilišni prijediplomski studij strojarstva (1010)

Zadatak: **Upotreba ROC krivulja u evaluaciji performansi sustava u inženjerstvu / Using the ROC curve to evaluate the performance of technical systems**

Opis zadatka:

U inženjerskim se problemima često na temelju ulaznih podataka, odnosno poznatih značajki, trebaju donijeti zaključci o pripadnosti određenoj kategoriji, predvidjeti vrijednost koja će se ostvariti ili pak vjerojatnost pojave određenog ishoda. Alati koji se pritom mogu koristiti su brojni modeli koji se temelje na strojnom učenju. Za tako dobivene modele važno je ispitati i procijeniti njihovu kvalitetu i pouzdanost, a jedan od alata za vrednovanje modela je Receiver Operating Characteristic (ROC) krivulja (krivulja osjetljivosti). Zadatak ovog završnog rada je istražiti primjenu modela strojnog učenja za predviđanje performansi sustava u inženjerstvu, uz poseban fokus na evaluaciju tih modela pomoću ROC krivulja. Moguće područje analize može biti predviđanje performansi timova (vozača) u Formuli 1. U tom slučaju cilj bi bio razviti model koji može predvidjeti uspješnost timova u različitim utrkama temeljem raznih podataka uključujući rezultate kvalifikacija, karakteristike staze, vremenske uvjete, povijest performansi vozača i timova i slično. Istraživanje uključuje prikupljanje i analizu relevantnih podataka, razvoj različitih modela strojnog učenja za predviđanje performansi na temelju prikupljenih podataka, evaluaciju modela pomoću ROC krivulja, te usporedbu različitih modela kako bi se odredila njihova pouzdanost i osjetljivost u cilju optimizacije performansi sustava.

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

Zadatak uručen pristupniku: 20.03.2024.

Predsjednik povjerenstva za završni ispit:
izv. prof. dr. sc. Samir Žic

Mentor:
izv. prof. dr. sc. Loredana Simčić

Komentor:
prof. dr. sc. Nelida Črnjarić

IZJAVA O AUTORSTVU ZAVRŠNOG RADA I JAVNOJ OBJAVI OBRANJENOG ZAVRŠNOG RADA

Ime i prezime studenta/studentice: **Leonarda Došen**

Matični broj studenta/studentice: **0069092298**

Naslov rada: **Upotreba ROC krivulja u evaluaciji performansi sustava u inženjerstvu**

Izjavljujem da sam ovaj rad samostalno izradila/o, te da su svi dijelovi rada, nalazi ili ideje koje su u radu citirane ili se temelje na drugim izvorima, bilo da su u pitanju knjige, znanstveni ili stručni članci, Internet stranice, zakoni i sl. u radu jasno označeni kao takvi, te navedeni u popisu literature.

Izjavljujem da kao student–autor završnog rada, dozvoljavam Tehničkom fakultetu Sveučilišta u Rijeci da ga trajno javno objavi i besplatno učini dostupnim javnosti u cjelovitom tekstu u mrežnom digitalnom repozitoriju Tehničkog fakulteta Sveučilišta u Rijeci.

U svrhu podržavanja otvorenog pristupa završnim radovima trajno objavljenim u javno dostupnom digitalnom repozitoriju Tehničkog fakulteta Sveučilišta u Rijeci, ovom izjavom dajem neisključivo imovinsko pravo iskorištavanja, bez sadržajnog, vremenskog i prostornog ograničenja, mog završnog rada kao autorskog djela pod uvjetima *Creative Commons* licencije CC BY Imenovanje, prema opisu dostupnom na http://creativecommons.org/liceses/.

U Rijeci, **9. srpnja 2024.**

Potpis studenta/studentice: _____

ZAHVALA

Želim se zahvaliti svojoj mentorici izv. prof. dr. sc. Loredani Simčić, te posebice komentorici prof. dr. sc. Nelidi Črnjarić koja mi je pružila podršku i pomoć tijekom izrade ovog završnog rada. Hvala Vam na strpljenju i utrošenom vremenu. Također, zahvaljujem se svom treneru i prijatelju mag. edu. math. et inf. Marku Kovačiću, kolegama, posebno Janu Peliću, i ostalima koji su svojom potporom, savjetima i idejama ovaj proces učinili lakšim.

SADRŽAJ:

1. UVOD

U inženjerskom okruženju današnjice značajni zadaci su: donošenje preciznih i točnih zaključaka o kategorizaciji te predviđanje i procjena vrijednosti i vjerojatnosti promatranih ishoda. Modeli strojnog učenja služe u postizanju navedenih ciljeva analizirajući ulazne podatke i generirajući izlazne podatke, odnosno odgovarajuće rezultate. Uz razvoj modela važno je osigurati i njihovu pouzdanost što dovodi do važne komponente strojnog učenja, a to je evaluacija performansi modela.

Postoje različiti alati za evaluaciju performansi modela, a jedan od važnijih je krivulja osjetljivosti ili ROC krivulja (eng. *Receiver Operating Characteristic Curve*). ROC krivulja omogućava detaljan uvid u sposobnost modela da razlikuje između pojedinih klasa, odnosno slučajeva. Također, ROC krivuljom dobiva se analiza osjetljivosti i specifičnosti modela pri različitim pragovima klasifikacije, a što se tiče samih inženjerskih sustava, ROC krivulja ključan je procjenitelj kvalitete modela.

Cilj ovog završnog rada je istraživanje primjene modela strojnog učenja u predviđanju performansi inženjerskih sustava, a posebno je fokusiran na evaluaciju spomenutih modela uz pomoć ROC krivulja. Unutar istraživanja, u radu će se analizirati mogućnost primjene ROC krivulja u evaluaciji performansi određenih inženjerskih sustava. Posebnu pozornost pridat će se području predviđanja uspješnosti timova i vozača u sportu Formula 1.

Formula 1, kao primjer inženjerskog sustava koji je odabran za istraživanje, nudi raznolik i bogat skup podataka i složene parametre koji igraju kompleksnu ulogu u dobivanju rezultata. Zadatak je razviti model koji ima karakteristiku preciznog predviđanja uspješnosti timova i vozača u različitim utrkama. U obzir je potrebno uzeti niz raznovrsnih faktora, odnosno parametara, uključujući rezultate kvalifikacija, obilježja staze, vremenske uvjete, povijest nastupa vozača i timova, i mnoge druge. Nakon prikupljanja odgovarajućih podataka i njihove analize, primjenom strojnog učenja razvijeni su različiti modeli te su njihove performanse vrednovane korištenjem ROC krivulja. Ti su modeli potom uspoređeni i ustanovljeno je koji je od njih najpouzdaniji i najstabilniji.

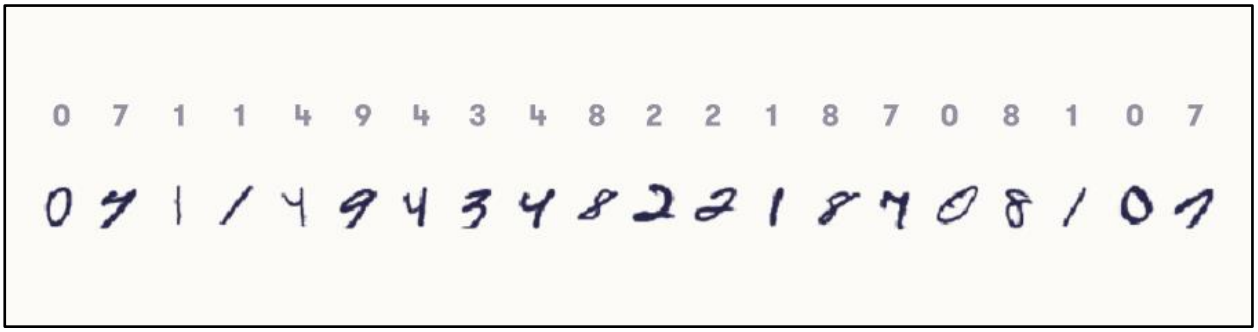
Rad je strukturiran na sljedeći način: najprije su opisani osnovni koncepti strojnog učenja i analiza korištenjem ROC krivulja, zatim je detaljno objašnjena metodologija istraživanja i način prikupljanja i pripreme podataka, te sam razvoj modela korištenjem strojnog učenja. Dan je prikaz rezultata provedenog istraživanja i provedena rasprava o istim i njihovom značaju. Na kraju su dani zaključci i pregled daljnjih mogućih istraživanja u ovom području.

Svrha ovog rada je razumijevanje i primjena ROC krivulja kao alata za evaluaciju performansi modela dobivenih primjenom strojnog učenja. Također su ponuđene praktične smjernice za optimizaciju performansi inženjerskih sustava.

2. STROJNO UČENJE

Odavno se zna da je učenje ključan element za usvajanje različitih znanja i vještina, a time i razvoj inteligencije. To vrijedi, kako za prirodnu inteligenciju, tako i za umjetnu inteligenciju. Strojno učenje predstavlja tehnologiju koja mijenja način na koji se interpretiraju podaci. Uočavanjem određenih zakonitosti među njima, te stvaranjem smislenih veza, razvijaju se nova znanja, koja postaju temelj za donošenje odluka i predviđanje različitih ponašanja sustava. Strojno je učenje već postalo neizostavan dio svakodnevnog života. Osnovna ideja strojnog učenja može se objasniti na jednostavnom primjeru.

Čest primjer koji se koristi kao motivacija za primjenu strojnog učenja, kao u [1], odnosi se na prepoznavanje rukom pisanih znamenki. Na Slici 2.1. prikazane su rukom pisane znamenke iz skupa podataka MNIST (Modified National Institute of Standards and Technology).



Slika 2.1. Rukom pisane znamenke iz skupa podataka MNIST i pripadajuće točne oznake [1]

Iznad svake rukom napisane znamenke prikazana je točna znamenka, odnosno znamenka koju je osoba trebala napisati. Može se primijetiti da postoji mogućnost pogrešnog prepoznavanja nekih rukom napisanih znamenki. Na primjer, nije sigurno je li druga znamenka slijeva zaista broj sedam ili je ustvari osoba originalno napisala broj četiri.

U klasifikacijskim problemima strojnog učenja, poput skupa podataka MNIST, postoji samo jedna ispravna klasifikacija za svaku instancu, u ovom slučaju broj. Cilj je osmisliti metodologiju

umjetne inteligencije koja može analizirati slike slične onima u skupu podataka MNIST i generirati točne oznake, odnosno odgovarajuće brojeve.

2.1. Vrste strojnog učenja

Algoritmi strojnog učenja omogućavaju računalima učenje iz podataka i unaprjeđivanje performansi na temelju iskustva, bez potrebe da su posebno programirani za svaki zadatak. Za razliku od tradicionalnog programiranja, gdje se svaki korak mora detaljno specificirati, strojno učenje omogućava računalima da uče iz primjera, osiguravajući im da samostalno donose zaključke o novim i nepoznatim podacima. Stoga se može zaključiti da je strojno učenje proces pomoću kojeg se identificiraju obrasci i donose odluke s minimalnim ljudskim nadzorom.

Strojno učenje koristi temelje statistike, koja se može opisati kao umijeće „izvlačenja“ informacija iz podataka. Metode poput linearne regresije, koja će biti opisana u narednim poglavljima, i Bayesove statistike, datiraju još iz prošlog stoljeća, a i danas su ključne komponente strojnog učenja. Strojno učenje često se grupira u različite skupine u ovisnosti o vrsti problema koje pokušava riješiti. U nastavku se navodi osnovna podjela strojnog učenja.

Postoje tri osnovne vrste strojnog učenja: nadzirano, nenadzirano i pojačano učenje (vidi [2]). Nadzirano učenje koristi označene podatke, odnosno podatke koji već imaju definirane odgovore, kako bi treniralo modele, omogućavajući im predviđanje odgovora na nove podatke. Nenadzirano učenje radi s neoznačenim podacima, tražeći skrivene obrasce bez unaprijed definiranih odgovora. Pojačano učenje tehnika je u kojoj model uči donositi odluke na temelju nagrada i kazni, optimizirajući svoje ponašanje kroz interakciju s okolinom.

Duboko učenje podskup je strojnog učenja te ono koristi složene modele za obradu podataka s visokim stupnjem apstrakcije, rješavajući tako izuzetno složene zadatke. Duboko učenje koristi kompleksne strukture poznate kao neuronske mreže, koje će biti objašnjene kao jedan od algoritama strojnog učenja u narednim poglavljima. Neuronske mreže služe dubokom učenju kako bi ono obradilo podatke na način koji oponaša ljudski mozak, omogućavajući modelima da obrade i interpretiraju zahtjevne i složene podatke poput slike i zvuka (vidi [3]).

Osnovne razlike između jednostavnijih algoritama strojnog učenja i dubokog učenja očituju se u njihovoj ovisnosti o podacima. Jednostavniji algoritmi strojnog učenja obično pokazuju izvrsne performanse na malom i srednjem skupu podataka, dok duboko učenje briljira na velikim skupovima podataka. Također, jednostavniji se algoritmi strojnog učenja mogu koristiti i na računalima slabijih performansi, pri čemu je obično potrebno razumjeti značajke podataka. S druge strane, duboko učenje zahtijeva računala snažnijih performansi, ali se prednost krije u tome što ono ne mora u potpunosti razumjeti značajke podataka.

Ove kategorije donekle se preklapaju i granice među njima često su nejasne, što može otežati klasifikaciju određene metode u samo jednu kategoriju. Na primjer, djelomično nadzirano učenje, kako i samo ime nalaže, kombinira elemente nadziranog i nenadziranog učenja. Odabir odgovarajućeg algoritma ovisi o vrsti problema koji se pokušava riješiti i vrsti dostupnih podataka.

2.2. Proces strojnog učenja

Proces strojnog učenja ponavljajući je postupak u kojem računalni sustav uči prepoznavati obrasce iz podataka u svrhu donošenja odluka i predviđanja rezultata na temelju novih ulaznih podataka. Sastoji se od nekoliko faza ili koraka koje uključuju prikupljanje i pripremu podataka, odabir i treniranje modela te naposljetku evaluaciju, optimizaciju i primjenu modela. U nastavku su, u skladu s [4], navedene faze procesa strojnog učenja, te su iste prikazane shematski na Slici 2.2.

Prikupljanje podataka smatra se prvom fazom procesa i ono obuhvaća pronalaženje relevantnih podataka koji igraju glavnu ulogu u uspješnom treniranju modela strojnog učenja. Podaci koji se mogu uzimati iz različitih izvora i baza podataka, mogu biti strukturirani, ali to nije uvijek slučaj.

Kako bi sakupljeni podaci postali upotrebljivi, moraju se „očistiti“ i često preformulirati na način da postanu pogodni za analizu i modeliranje. Za to služi druga faza procesa, priprema podataka, koja podrazumijeva uklanjanje neiskoristivih vrijednosti, normalizaciju podataka, kodiranje kategoričkih varijabli te podjelu podataka na skupove za testiranje i učenje.

Nakon uspješne pripreme podataka slijedi treća faza procesa u kojoj se odabire željeni model, odnosno algoritam strojnog učenja. Model se bira ovisno o vrsti problema koji se nastoji riješiti i svojstvima dostupnih podataka. Neki od popularnih modela su linearna i logistička regresija, stabla odlučivanja, slučajne šume, potporni vektori i neuronske mreže. Ovi se modeli koriste u već spomenutom nadziranom strojnom učenju, osim neuronskih mreža, koje se koriste u dubokom učenju. U naredim će se poglavljima opisati navedeni modeli.

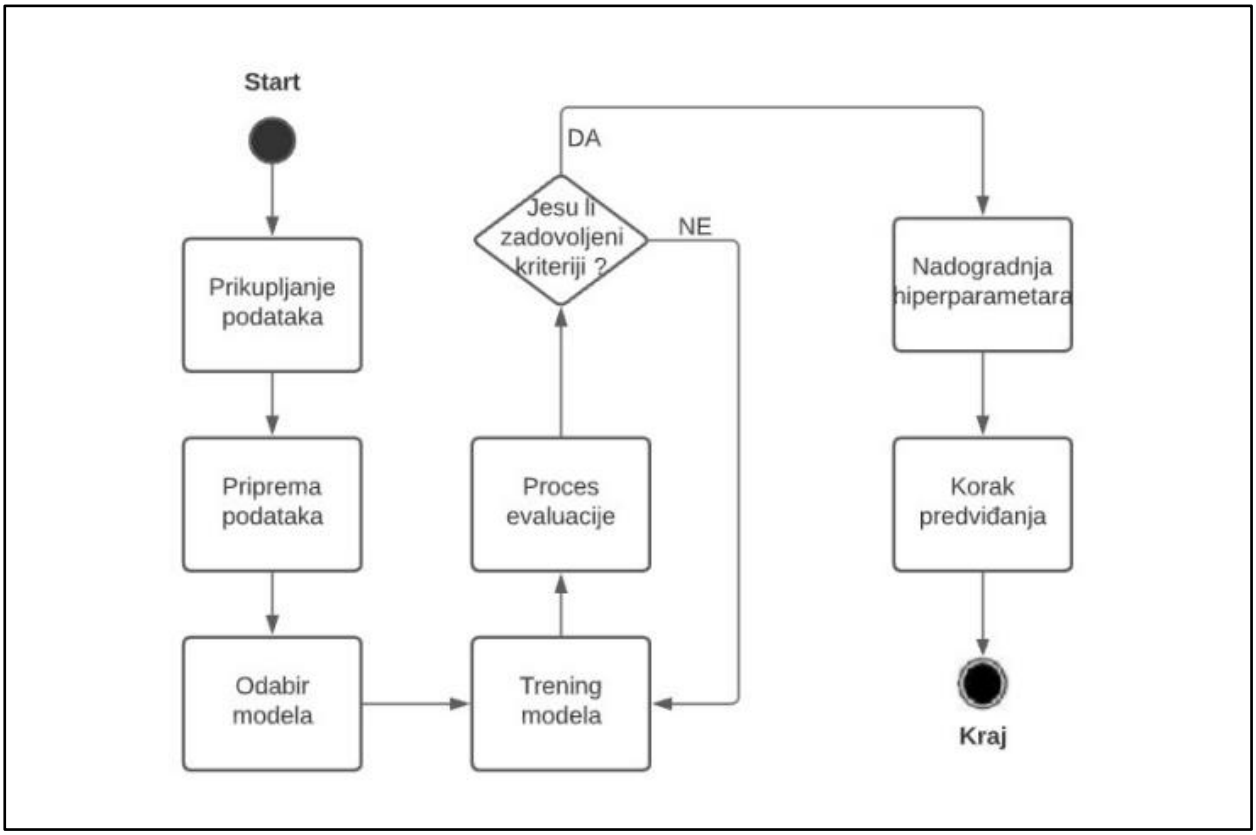
Iduća, četvrta faza, je treniranje modela, u kojoj se vrši prilagodba modela pomoću podataka za učenje kako bi model mogao naučiti odnose i veze između ulaznih i izlaznih podataka s ciljem minimizacije grešaka i optimizacije njegovih performansi.

Peta faza procesa ili evaluacija modela odnosi se na procjenu sposobnosti modela da generalizira na temelju dosad neviđenih podataka. Pritom se koriste različite metrike evaluacije, kao što su točnost, preciznost, osjetljivost, i tako dalje.

U sklopu evaluacije modela postavlja se i pitanje jesu li zadovoljeni određeni kriteriji. U slučaju da kriteriji ne zadovoljavaju, algoritam se vraća na postupak treniranja, a ako kriteriji zadovoljavaju, model prelazi na sljedeći korak, to jest optimizaciju i moguću nadogradnju.

Na temelju dobivenih rezultata evaluacije, model je moguće optimizirati, što obilježava šestu fazu procesa. To može uključivati podešavanje parametara modela, dodatno transformiranje podataka, pa čak i odabir drugog modela.

Po izvršenju svih navedenih faza procesa strojnog učenja, model je spreman za praktičnu primjenu, odnosno predviđanje, što karakterizira sedmu i posljednju fazu procesa.



Slika 2.2. Faze procesa strojnog učenja [4]

U narednim će se poglavljima detaljnije opisati svaka od faza procesa strojnog učenja primjenjujući ih na konkretno odabranim primjerima. Prethodno će se dati pregled algoritama strojnog učenja, a nakon toga i opis metoda za evaluaciju performansi modela strojnog učenja.

2.3. Algoritmi strojnog učenja

Kad je riječ o strojnom učenju, obično se referira na različite algoritme koji čine osnovu tog procesa. Algoritmi strojnog učenja su skupovi pravila i matematičkih modela koje računalo koristi za analizu i učenje iz podataka. Neki od važnih algoritama uključuju već spomenutu linearnu i logističku regresiju, stabla odlučivanja, slučajne šume i potporne vektore, koji čine modele nenadziranog učenja, te neuronske mreže, koje se svrstava u modele dubokog učenja.

Klasifikacija algoritama strojnog učenja ključan je korak u razumijevanju različitih pristupa u ovoj složenoj i dinamičnoj disciplini. U ovom poglavlju, opisat će se različiti algoritmi strojnog učenja,

koji se koriste za rješavanje različitih problema u području inženjerstva, i njihove osnovne karakteristike, kako bi se stekao detaljniji uvid u njihovu primjenu.

2.3.1. Linearna regresija

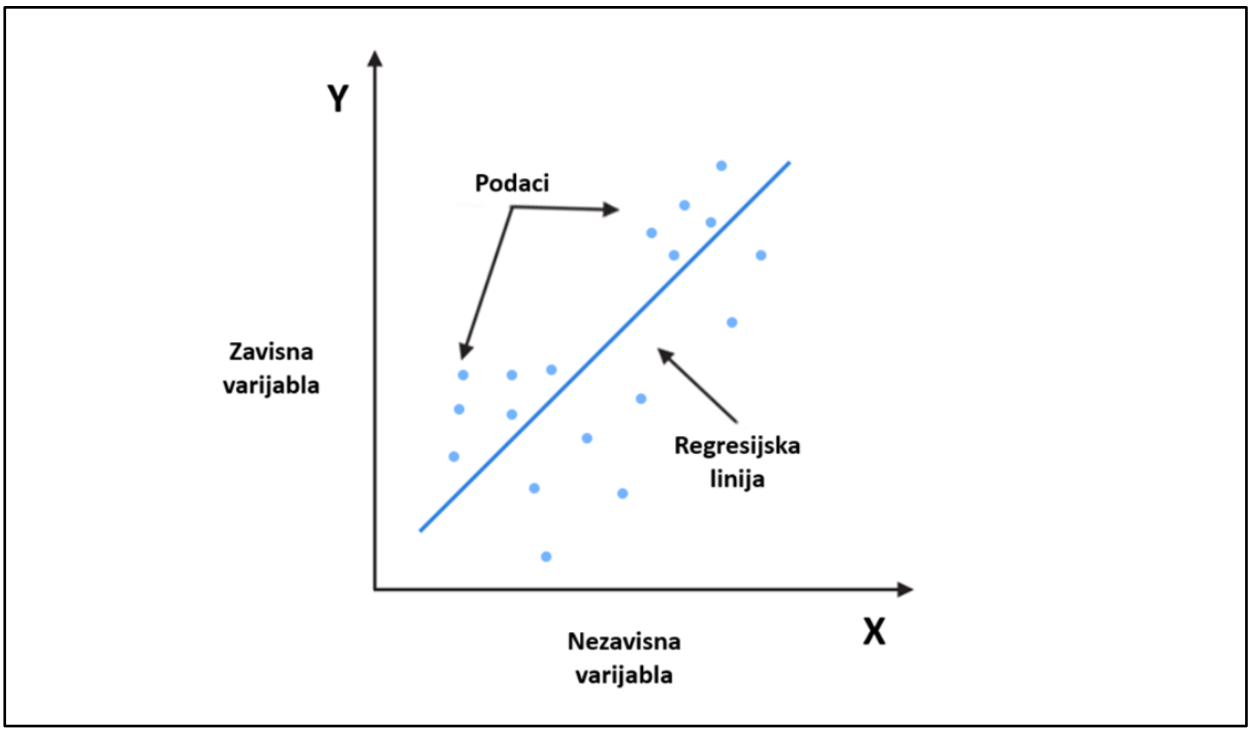
Linearna regresija (eng. *Linear Regression*) statistička je metoda koja se koristi za modeliranje linearnog odnosa između nezavisne, to jest ulazne varijable (eng. *Independent Variable*) i zavisne, to jest izlazne varijable (eng. *Dependent Variable*), s ciljem predviđanja vrijednosti zavisne varijable. Može biti jednostavna, s jednom nezavisnom varijablom, ili višestruka, s više nezavisnih varijabli.

Učenje modela podrazumijeva određivanje parametara koji minimiziraju grešku između stvarnih i predviđenih vrijednosti, a evaluira se pomoću metrika poput srednje kvadratne pogreške MSE (eng. *Mean Squared Error*), koja mora biti što manja. Model se koristi za predviđanje vrijednosti na temelju novih podataka, no zahtijeva pretpostavke o linearnoj neovisnosti i normalnoj distribuciji greške. Linearna regresija značajna je tehnika u disciplinama ekonomije, medicine i inženjerstva.

Kod linearne regresije polazi se od toga da su podaci oblika (X, Y), pri čemu je X nezavisna varijabla i Y zavisna varijabla (Slika 2.3.) Osnovni je zadatak linearne regresije pronaći linearni odnos između ovih varijabli i predvidjeti vrijednost Y na temelju poznate vrijednosti X, to jest, odrediti jednadžbu oblika $\hat{Y} = aX + b$ na način da odstupanje modela od zadanih vrijednosti Y bude najmanje moguće.

Ovdje je parametar a nagib pravca, dok je b odsječak na osi ordinata. Optimalne vrijednosti za parametre a i b određuju se tako da se minimizira srednje kvadratno odstupanje između stvarnih vrijednosti Y i predviđenih vrijednosti modela.

Po završetku određivanja parametara, model je naučen i spreman za korištenje. Predviđanje se vrši tako da se za nove vrijednosti nezavisne varijable *X*, uvrštavanjem u naučeni model dobivaju predviđene vrijednosti zavisne varijable *Y*.



Slika 2.3. Graf linearne regresije [5]

Važno je napomenuti da stvarne primjene linearne regresije često uključuju složenije postupke kao što su analiza podataka i provjera pretpostavki modela, kako bi se osigurala pouzdanost i točnost predviđanja. U svakom slučaju, tehnika linearne regresije moćna je tehnika modeliranja.

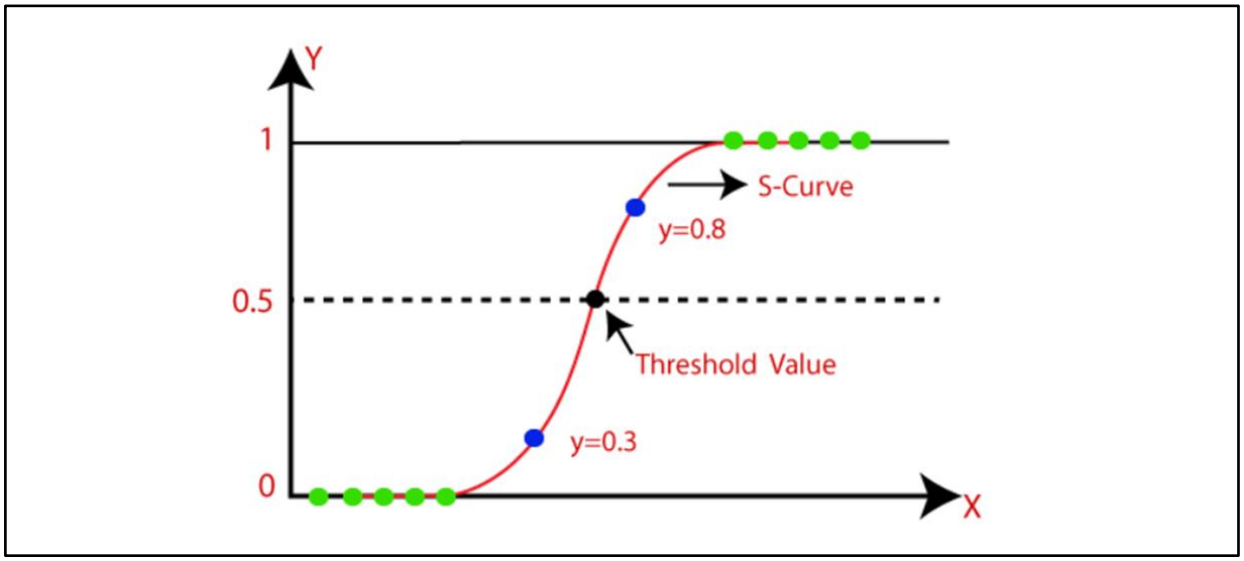
2.3.2. Logistička regresija

Logistička regresija (eng. *Logistic Regression*) također je metoda koja se koristi za modeliranje odnosa između zavisne varijable i jedne ili više nezavisnih varijabli, s ciljem klasifikacije primjera u različite kategorije. Iako govorimo o regresiji, logistička regresija uglavnom se koristi za binarnu klasifikaciju gdje se podaci svrstavaju u jednu od dvije kategorije, kao što su pozitivno i negativno ili prisutno i odsutno. Kategorije općenito označavamo s nulom ili jedinicom.

Temeljna je ideja modeliranje vjerojatnosti pomoću logističke funkcije. Pritom logistička funkcija kao funkcija nezavisne varijable *X* daje vjerojatnost da zavisna varijabla *Y* pripada jednoj od kategorija. Logistička funkcija transformira linearnu kombinaciju nezavisnih varijabli u raspon vrijednosti između nula i jedan. Parametri modela procjenjuju se iz podataka korištenjem metode maksimalne vjerojatnosti. Logistička regresija široko je korištena u problemima klasifikacije, u područjima poput medicinske dijagnostike i analize rizika, zbog svoje efikasnosti i jednostavnosti tumačenja rezultata.

Učenje modela svodi se na optimizacijski postupak u kojem se određuje minimum funkcije maksimalne vjerojatnosti. Pritom se optimalno rješenje određuje korištenjem iterativnih postupaka, najčešće gradijentne metode. Izvršavanjem odabranog broj iteracija, izračunava se linearna kombinacija, vjerojatnost, te gradijent za trenutne vrijednosti parametara modela. Gradijent se potom koristi u optimizacijskoj metodi za minimizaciju greške modela. Za optimalne parametre, dobiva se model.

Sada se za novi skup vrijednosti nezavisne varijable *X*, može koristiti naučeni model za predviđanje pripadajuće kategorije. Ovaj je algoritam osnova za implementaciju logističke regresije, koja se često koristi u problemima klasifikacije zbog svoje jednostavnosti i sposobnosti dobre interpretacije rezultata. Graf logističke funkcije, preuzet iz [6], prikazan je na Slici 2.4. Na slici je prikazana situacija u kojoj je kritična vrijednost (eng. *Threshold Value*) postavljena na 0,5, pa će vrijednosti zavisne varijable *Y* biti jednake 1 ako je vrijednost logističke funkcije (eng. *S-Curve*) veća od 0,5, a 0 ako je ta vrijednost manja od 0,5.

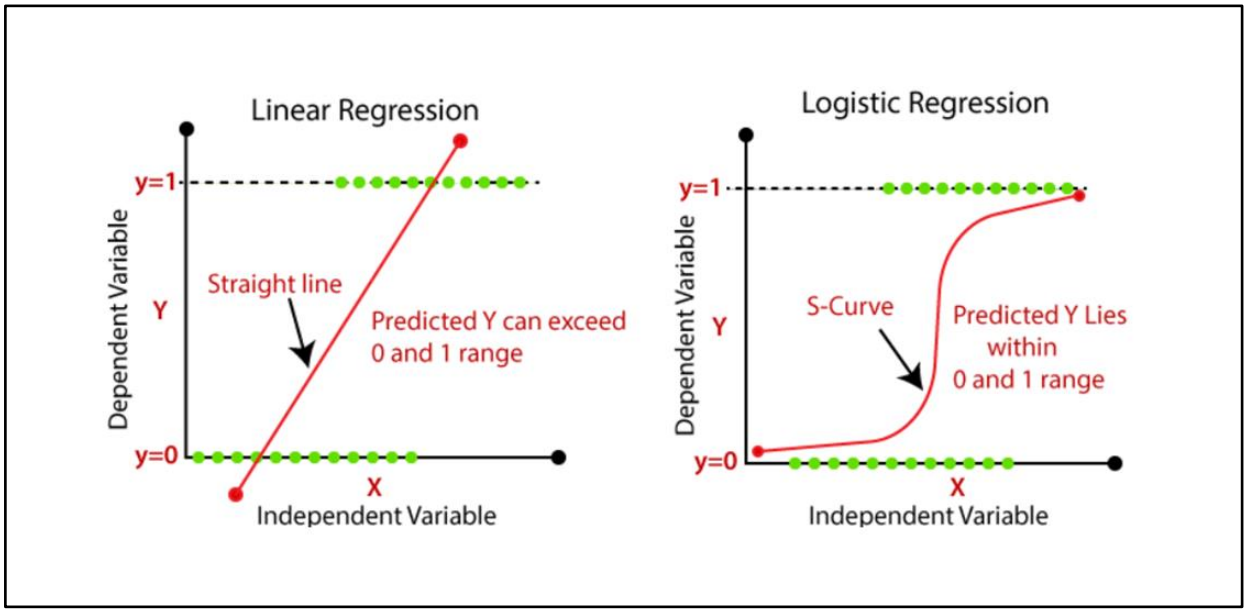


Slika 2.4. Graf logističke regresije [6]

Također, važno je napomenuti da stvarna implementacija obično uključuje dodatne korake poput provjere konvergencije, primjene tehnika regularizacije te provjere i validacije modela kako bi se osigurala njegova pouzdanost i optimalne performanse.

Zaključno, linearna regresija i logistička regresija dva su poznata algoritma strojnog učenja koji spadaju u tehniku nadziranog učenja. Budući da su oba algoritma nadzirane prirode, koriste označeni skup podataka za predviđanje. Glavna razlika među njima leži u tome kako se i u kojim situacijama koriste.

Linearna regresija koristi se za predviđanje numeričkih vrijednosti, a modelira linearni odnos između nezavisnih i zavisnih varijabli. Rezultat linearne regresije kontinuirana je vrijednost. Logistička regresija koristi se za klasifikaciju, a ponajviše binarnu klasifikaciju. Modelira vjerojatnost da će neki podatak pripadati određenoj kategoriji, a rezultat je interval vjerojatnosti između nula i jedan. Usporedba linearne i logističke regresije prikazana je na Slici 2.5.



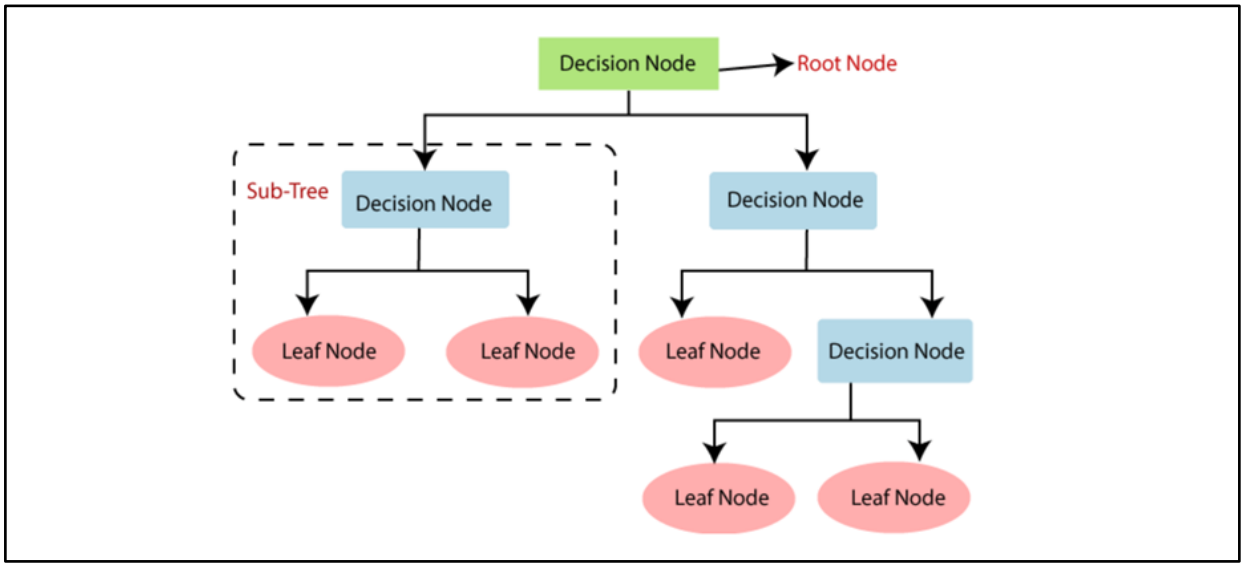
Slika 2.5. Grafički prikaz linearne i logističke regresije [7]

2.3.3. Stabla odlučivanja

Stabla odlučivanja, DT (eng. *Decision Trees*), popularan su algoritam u strojnom učenju koji se koristi za klasifikaciju i regresiju. Osnovna je ideja konstruirati stablo koje se sastoji od čvorova i grana, gdje svaki čvor predstavlja odluku (eng. *Decision Node*) temeljenu na vrijednostima određenih značajki, dok svaka grana predstavlja mogući ishod te odluke.

Proces izgradnje stabla počinje od korijenskog čvora (eng. *Root Node*), podijelivši podatke na temelju kriterija, poput srednje kvadratne greške za regresijske probleme. Nakon podjele, proces se ponavlja za svaki podskup podataka na sljedećoj razini čvorova, uz mogućnost zaustavljanja prema definiranim kriterijima. Stablo odlučivanja lako je objašnjivo i omogućuje ljudima da razumiju donesene odluke. Ovaj pristup ima široku primjenu, ponovno u područjima medicine, ali i u područjima financija.

Stablo odlučivanja hijerarhijska je struktura koja se sastoji od čvorova i grana, gdje svaki čvor predstavlja odluku temeljenu na odabranoj značajki, dok grane vode do podskupova podataka (Slika 2.6.).



Slika 2.6. Shematski prikaz stabla odlučivanja [8]

Algoritam počinje funkcijom izgradnje stabla koja je rekurzivna. Prvo se provjerava jesu li svi podaci u trenutnom čvoru iste kategorije ili je dosegnuta maksimalna dubina stabla. Ako je to slučaj, stvara se list čvora (eng. *Leaf Node*) s najčešćom kategorijom među podacima u tom čvoru. U suprotnom, odabire se najbolja značajka za podjelu podataka. Nakon odabira značajke, stvara se unutarnji čvor, a podaci se dijele na temelju vrijednosti te značajke.

Algoritam zatim rekurzivno poziva sam sebe za svaki podskup podataka, stvarajući podstabla (eng. *Sub-Tree*) i čvorove na dubljoj razini. Ovaj se postupak ponavlja dok se ne dosegnu kriteriji za zaustavljanje, kao što su maksimalna dubina stabla ili minimalni broj uzoraka u čvoru. Konačni je rezultat stablo odlučivanja koje se sastoji od čvorova i grana, kao što je prikazano na Slici 2.6.

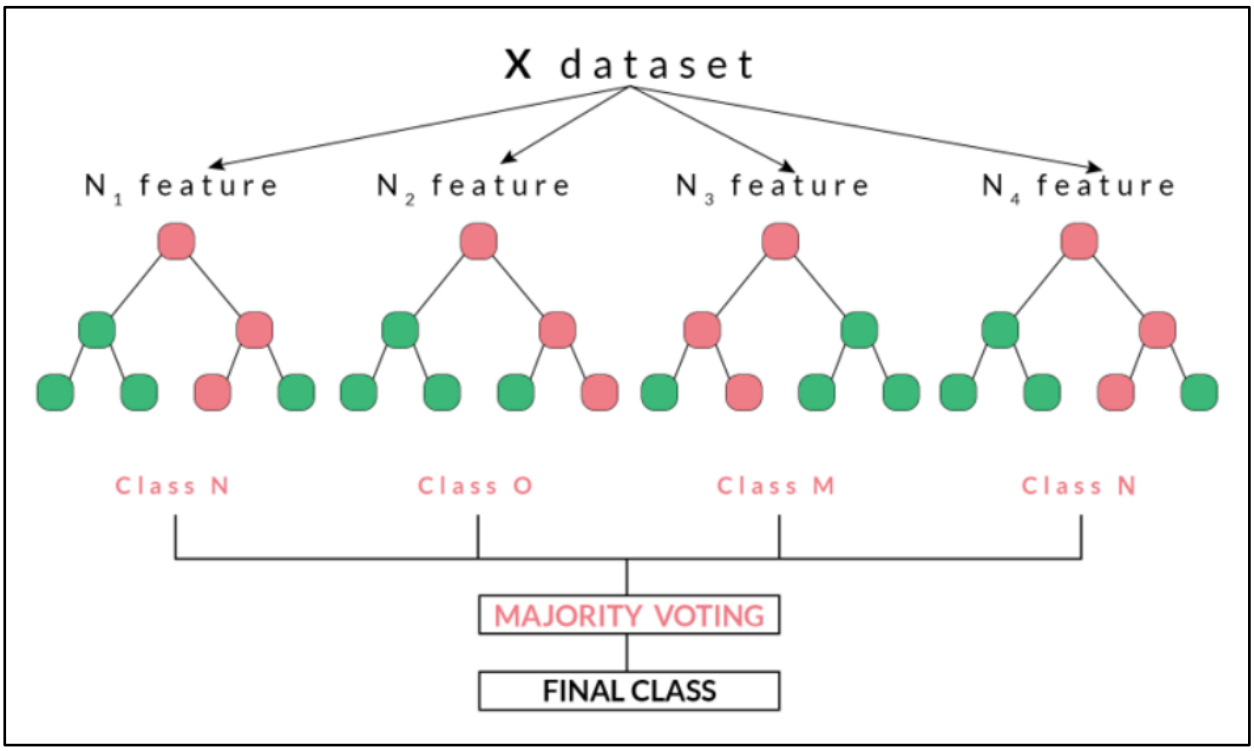
Nakon izgradnje stabla, mogu se koristiti funkcije predviđanja stabla za predviđanje kategorija novih podataka. Ta funkcija prolazi kroz stablo od korijena do lista, donoseći odluke na temelju vrijednosti značajki novih primjera. Stablo odlučivanja čitljiva je metoda koja se često koristi u strojnom učenju za klasifikaciju i regresiju zbog svoje jednostavnosti i sposobnosti razumijevanja donesenih odluka.

2.3.4. Slučajne šume

Slučajne šume, RF (eng. *Random Forest*), algoritam su koji se temelji na stablima odlučivanja. Ovaj algoritam kombinira predikcije više stabala odlučivanja kako bi postigao veću stabilnost i generalizaciju. Slučajnost je ključna komponenta slučajnih šuma, što se očituje u nasumičnom odabiru značajki tijekom izgradnje stabala, što pomaže u sprečavanju prenaučenosti. Konačna predikcija za novi podatak dobiva se većinskim glasanjem svih stabala unutar šume.

Podaci odabrani za trening se slučajnim odabirom podijele u podskupove, a nakon toga se svako stablo gradi na temelju dobivenih podskupova (vidi [9]).

Nakon što su sva stabla izgrađena, predviđanje se vrši za svaki testni podatak i svako pojedino stablo. Konačna predikcija (eng. *Final Result*) dobiva se većinskim glasanjem (eng. *Majority Voting*) među svim stablima. Shematski prikaz prethodno opisanog stabla dan je na Slici 2.7.



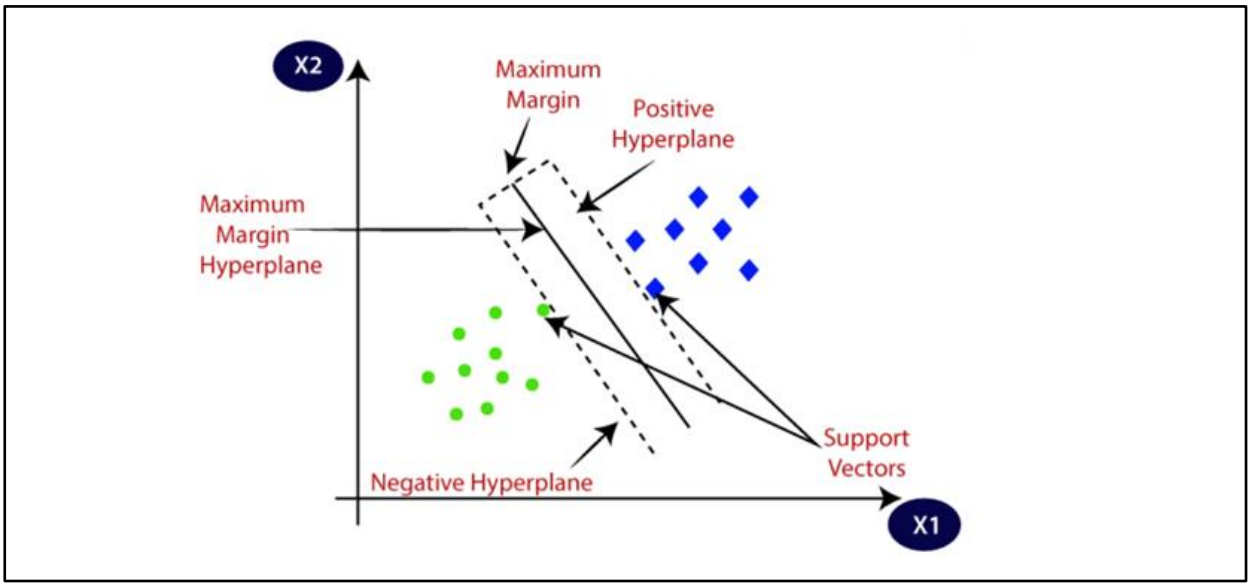
Slika 2.7. Shematski prikaz slučajne šume [9]

Kao što je već rečeno, slučajne su šume robusne i imaju visoku sposobnost generalizacije i smanjenja varijance. Slučajna šuma često pruža bolje performanse od pojedinačnih stabala odlučivanja i koristi se u različitim zadacima strojnog učenja. Ova tehnika omogućava modelima da budu stabilniji i bolje se nose s raznolikim vrstama podataka.

2.3.5. Potporni vektori

Potporni vektori, poznatiji kao SVM (eng. *Support Vector Machine*), algoritam su strojnog učenja koji koristi modele nadziranog učenja kako bi riješio kompleksne probleme klasifikacije, regresije i detekcije odstupanja između podataka provođenjem optimalnih transformacija tih podataka.

Cilj SVM algoritma je stvoriti najbolju liniju ili granicu odluke koja može razdvojiti n-dimenzionalni prostor u klase, tako da se nova točka podatka može smjestiti u ispravnu kategoriju (Slika 2.8.). Ta najbolja granica odluke naziva se hiperravnina (eng. *Hyperplane*). SVM algoritam zapravo odabire ekstremne točke/vektore koji pomažu u definiranju hiperravnina.



Slika 2.8. Grafički prikaz potpornih vektora [10]

Kao i s već opisanim modelima strojnog učenja, podaci se podijele na skup za treniranje i skup za testiranje, pod pretpostavkom da je analiza podataka već provedena.

Algoritam potpornih vektora široko je korišten u strojnom učenju jer može obraditi kako linearne tako i nelinearne klasifikacijske zadatke. Međutim, kada podaci nisu linearno odvojivi, koristi se transformacija podataka u višedimenzijski prostor kako bi se omogućila linearna separacija.

Transformacije određuju granice između podataka na temelju unaprijed definiranih kategorija, oznaka ili izlaza. Potporni vektori široko su prihvaćeni u različitim disciplinama poput zdravstva, obrade prirodnog jezika, primjena obrade signala te prepoznavanja govora i slika.

2.3.6. Neuronske mreže

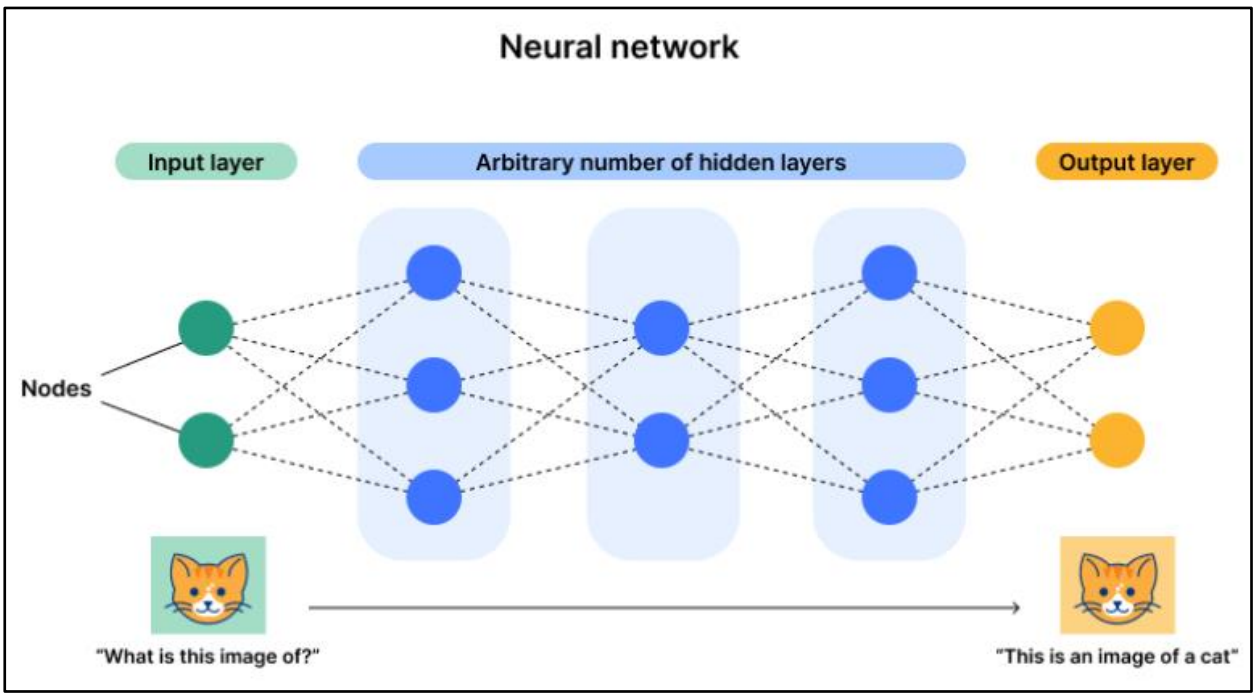
Neuronske mreže (eng. *Neural Networks*), algoritam su strojnog učenja temeljen na modelu funkcioniranja ljudskog mozga. Neuronske mreže sastoje se od skupa procesnih jedinica koje se nazivaju čvorovi (eng. *Nodes*). Ti čvorovi međusobno prenose podatke, slično tome kako neuroni u mozgu prenose električne impulse.

Neuronske mreže koriste se u dubokom učenju, naprednoj vrsti strojnog učenja koja može donositi zaključke iz neoznačenih podataka bez ljudske intervencije. Na primjer, model dubokog učenja izgrađen na neuronskoj mreži i hranjen dovoljno velikim skupom podataka za učenje mogao bi identificirati predmete na fotografiji koje nikada prije nije vidio. Neuronske mreže široko se koriste u različitim aplikacijama, kao što je spomenuto prepoznavanje slika i obrada prirodnog jezika.

Neuronska mreža obično uključuje mnogo procesora koji djeluju paralelno i raspoređeni su u slojevima ili nivoima. Prvi sloj, koji je analogan optičkim živcima u ljudskoj vizualnoj obradi, prima neobrađene ulazne informacije (eng. *Input Layer*). Svaki sljedeći sloj prima izlaz iz prethodnog sloja umjesto sirovih ulaznih podataka, na isti način na koji neuroni udaljeniji od optičkog živca primaju signale od onih bliže njemu. Posljednji sloj proizvodi izlaz sustava (eng. *Output Layer*).

Svaki procesni čvor ima vlastito ograničeno područje znanja, uključujući informacije koje je vidio i bilo kakva pravila s kojima je prvotno programiran ili koja je razvio samostalno. Slojevi su visoko povezani, što znači da će svaki čvor u sloju biti povezan s mnogim čvorovima u nižem i višem sloju. Može postojati jedan ili više čvorova u izlaznom sloju, iz kojih se čita proizvedeni odgovor.

Neuronske mreže poznate su po svojoj prilagodljivosti, što znači da se mijenjaju kako uče iz početnog treninga i dobivaju više informacija. Najosnovniji model učenja usredotočen je na težinsko modeliranje ulaznih tokova, način na koji svaki čvor mjeri važnost ulaznih podataka od svojih prethodnika. Ulazi koji doprinose dobivanju pravih odgovora imaju veće težine. Shematski prikaz jedne neuronske mreže dan je na Slici 2.9.



Slika 2.9. Shematski prikaz neuronske mreže [11]

U ovom su poglavlju opisani osnovni koncepti, vrste, proces i algoritmi strojnog učenja. U sljedećim poglavljima, razmatrat će se evaluacija performansi modela strojnog učenja i proučiti različite metode evaluacije, s naglaskom na ROC krivulje.

3. EVALUACIJA PERFORMANSI MODELA STROJNOG UČENJA

Evaluacija performansi modela strojnog učenja jedan je od važnih koraka u izgradnji učinkovitog modela strojnog učenja. Za evaluaciju performansi ili kvalitete modela koriste se različite metrike, a te metrike poznate su kao metrike performansi ili evaluacijske metrike. Metrike performansi procjenjuju koliko dobro model funkcionira s danim podacima. Podešavanjem parametara ponekad se mogu poboljšati performanse modela. Svaki model strojnog učenja ima za cilj dobro generalizirati na novim podacima, a metrike performansi pomažu utvrditi koliko dobro model generalizira, odnosno predviđa rezultate, koristeći novi skup podataka.

U strojnom učenju, svaki zadatak ili problem dijeli se na klasifikaciju i regresiju. Nisu sve metrike prikladne za sve vrste problema, stoga je važno znati i razumjeti koje metrike treba koristiti. U nastavku se govori o zadacima klasifikacije te pripadajućim metrikama koje mogu poslužiti za analizu.

3.1. Klasifikacijski problem

Klasifikacija je nadzirani proces strojnog učenja kojim se dani skup ulaznih podataka kategorizira u razrede na temelju jedne ili više varijabli. Klasifikacijski problem može se provesti na strukturiranim i nestrukturiranim podacima kako bi se točno predvidjelo hoće li podaci pripasti unaprijed određenim kategorijama. U strojnom učenju, klasifikacija može zahtijevati dvije ili više kategorija u danom skupu podataka (vidi [13]).

Na primjer, problemom klasifikacije može se smatrati određivanje je li osoba oboljela od neke bolesti. Još jedan uobičajen primjer je odluka o kupnji proizvoda na internetskom portalu. Pitanje je kupiti li proizvod sada ili čekati nekoliko mjeseci kako bi iskoristili maksimalni popust. Ili, u slučaju kupnje automobila, koji je automobil, od ponuđenih opcija, najbolji izbor s obzirom na potrebe i proračun.

Za evaluaciju performansi klasifikacijskog modela koriste se različite mjere, uključujući točnost (eng. *Accuracy*), matricu zabune (eng. *Confusion Matrix*), preciznost (eng. *Precision*), senzitivnost

(eng. *Sensitivity*), F-ocjenu (eng. *F-Score*) i ROC krivulju, to jest površinu ispod ROC krivulje (eng. *Area Under the Curve*), koja se označava s AUC. U sljedećim poglavljima, bit će opisane kako pojedine metrike funkcioniraju. Posebna pozornost bit će pridana ROC krivuljama.

3.1.1. Točnost

Točnost se definira kao omjer točno klasificiranih podataka u odnosu na ukupan broj podataka.

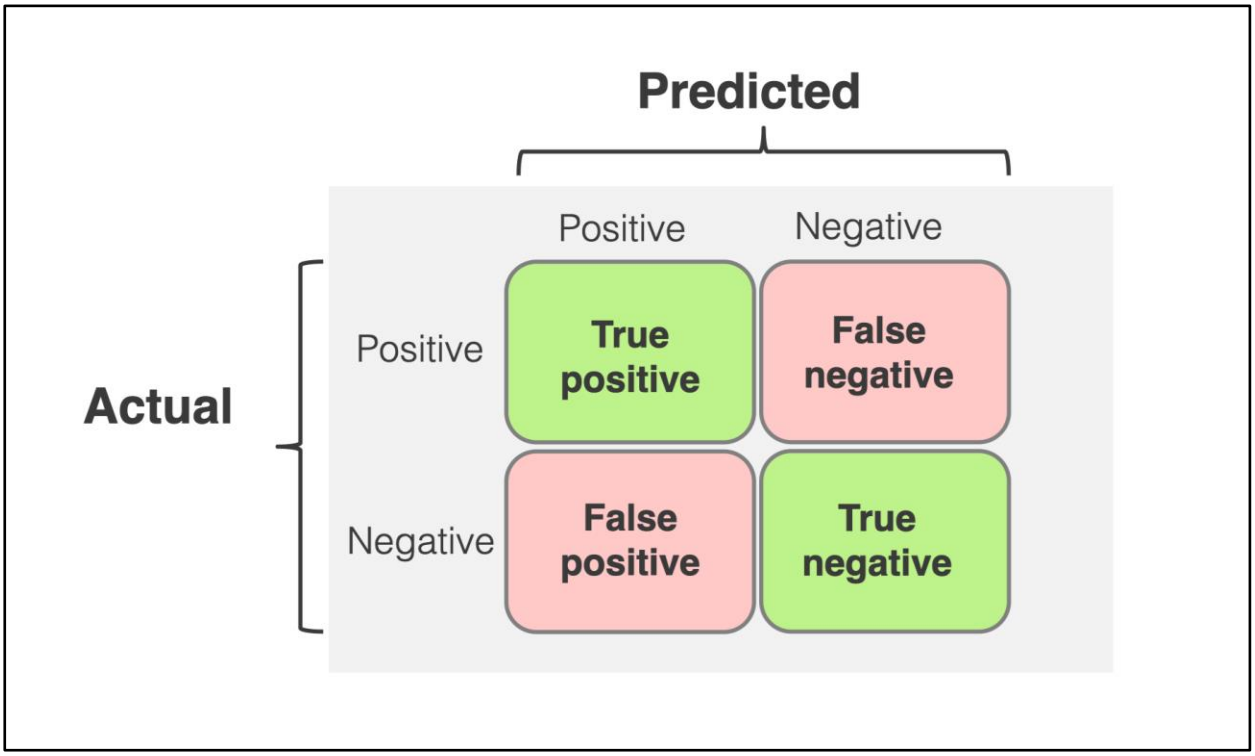
$$\text{Točnost} = \frac{\text{Broj točnih predikcija}}{\text{Ukupan broj predikcija}} \tag{3.1}$$

Iako je jednostavna za korištenje i implementaciju, točnost je prikladna samo za slučajeve u kojima svaka kategorija ima približno jednak broj uzoraka, odnosno kada su kategorije ciljne varijable u podacima uravnotežene po veličini.

Točnost se ne koristi kada ciljna varijabla većinom pripada jednoj kategoriji. Na primjer, neka postoji model za predviđanje bolesti u kojem od velikog broja ljudi samo nekolicina ima bolest, npr. manje od 10%. U ovom slučaju, ako model predviđa da je svaka osoba zdrava, mjera točnosti bit će iznad 90%. Ovakvo je predviđanje neispravno i nekvalitetno, bez obzira na visoku točnost.

3.1.2. Matrica zabune

Matrica zabune tablični je prikaz predikcijskih rezultata bilo kojeg binarnog klasifikatora. Ovaj tablični prikaz koristi se kako bi opisao performanse klasifikacijskog modela na skupu testnih podataka kada su poznate stvarne vrijednosti. Matrica zabune jednostavna je za prikaz, ali terminologija korištena u ovoj matrici može biti zbunjujuća, kako joj i samo ime sugerira. Tipična matrica zabune za binarni klasifikator prikazana je na Slici 3.1.



Slika 3.1. Matrice zabune za binarni klasifikator [12]

Pretpostavlja se da postoji skup podataka za koje se zna njihova ispravna klasifikacija. Model strojnog učenja koji se analizira, dat će za te podatke „svoja“ predviđanja, odnosno pripadnosti pojedinim klasama. U matrici zabune se potom prikazuju sljedeće vrijednosti: TN (eng. *True Negative*) označava stvarno negativne vrijednosti, odnosno broj točno predviđenih negativnih slučajeva. Slično tome, TP (eng. *True Positive*) označava stvarno pozitivne vrijednosti, odnosno broj točno predviđenih pozitivnih slučajeva. S druge strane, FP (eng. *False Positive*) označava lažno pozitivnu vrijednost, to jest broj negativnih slučajeva koji su netočno previđeni kao pozitivni, dok FN (eng. *False Negative*) označava lažno negativnu vrijednost ili broj pozitivnih slučajeva koji su netočno predviđeni kao negativni.

Za bolje razumijevanje ovih pojmova, u nastavku se promatra primjer pravosudnog sustava. Svaki pravosudni sustav želi kažnjavati samo osobe koje su počinile zločin, a ne želi pogrešno optuživati nevine osobe. Može se pretpostaviti da promatrani model opisuje pravosudni sustav koji procjenjuje svaku osobu i predviđa je li ona kriva ili nevina. U tom slučaju, TN predstavlja nevine osobe koje su oslobođene optužbi, dok TP predstavlja uhvaćene kriminalce i zločince. Nasuprot tome, FP označava nevine osobe koje su osuđene, dok FN označava počinitelje zločina koji su proglašeni nevinima.

3.1.3. Preciznost

Preciznost je metrika koja se koristi za prevladavanje ograničenja točnosti. Preciznost predstavlja udio pozitivnih predikcija koje su stvarno točne. Može se izračunati kao omjer ispravno predviđenih pozitivnih predikcija i ukupnih pozitivnih predikcija (koje obuhvaćaju i lažno pozitivne predikcije).

$$\text{Preciznost} = \frac{\text{Stvarno pozitivno}}{\text{Stvarno pozitivno} + \text{Lažno pozitivno}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

U prethodno spomenutom primjeru pravosudnog sustava, preciznost bi predstavljala omjer broja uhvaćenih počinitelja zločina i ukupnog broja zatvorenih osoba, i krivih i nevinih.

3.1.4. Senzitivnost

Senzitivnost ili odziv (eng. *Recall*) metrika je slična preciznosti. Međutim, cilj joj je izračunati udio stvarno pozitivnih vrijednosti u odnosu na ukupne stvarne pozitivne vrijednosti u originalnom skupu podataka. Senzitivnost predstavlja udio ispravno predviđenih pozitivnih vrijednosti, u skupu stvarno pozitivnih. Bilo da su te pozitivne vrijednosti ispravno predviđene kao pozitivne ili neispravno predviđene kao negativne.

$$\text{Senzitivnost} = \frac{\text{Stvarno pozitivno}}{\text{Stvarno pozitivno} + \text{Lažno negativno}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

U primjeru pravosudnog sustava, senzitivnost predstavlja omjer broja uhvaćenih počinitelja zločina i ukupnog broja kriminalaca, što zatvorenih, što slobodnih.

Iz definicija preciznosti i senzitivnosti može se zaključiti da senzitivnost određuje performanse klasifikatora u odnosu na lažno negativne rezultate, dok preciznost pruža informacije o performansama klasifikatora u odnosu na lažno pozitivne rezultate.

Stoga, kako bi se minimizirali lažno negativni rezultati, senzitivnost bi trebala biti što bliže vrijednosti od 100%, a za minimizaciju lažno pozitivnih rezultata, preciznost bi trebala biti što bliža vrijednosti od 100%. Dakle, uz maksimalnu preciznost, minimiziraju se lažno pozitivne pogreške, a uz maksimalnu senzitivnost, lažno negativne pogreške.

3.1.5. F-ocjena

F-ocjena je metrika za evaluaciju binarnog klasifikacijskog modela na temelju predikcija koje su napravljene za pozitivnu kategoriju. Izračunava se uz pomoć preciznosti i senzitivnosti. To je vrsta pojedinačne mjere koja uključuje i preciznost i senzitivnost. F-ocjena može se izračunati kao harmonijska sredina preciznosti i senzitivnosti, dodjeljujući jednaku težinu svakoj od njih.

$$\mathrm{F-ocjena} = 2 * \frac{\mathrm{Preciznost} * \mathrm{Senzitivnost}}{\mathrm{Preciznost} + \mathrm{Senzitivnost}} \tag{3.4}$$

F-ocjena koristi i preciznost i senzitivnost, stoga je treba koristiti kad su oba elementa važna za evaluaciju, ali je jedan od njih nešto važniji od drugog. Na primjer, kada su lažno negativni rezultati relativno važniji od lažno pozitivnih rezultata, ili obrnuto (vidi [12] za detalje).

3.1.6. ROC krivulja (krivulja osjetljivosti)

Ponekad je korisno promatrati odnos pojedinih metrika klasifikacijskog modela i prikazati ih grafički. U tu svrhu može se koristiti ROC krivulja, koja može poslužiti kao važan alat za evaluaciju performansi klasifikacijskog modela.

Zanimljiva povijesna činjenica o ROC krivuljama jest da su prvi put korištene tijekom Drugog svjetskog rata za analizu radarskih signala. Nakon napada na Pearl Harbor, američko vojno vodstvo željelo je detektirati japanske zrakoplove pomoću radarskih signala. ROC krivulje pokazale su se izuzetno korisnima za tu zadaću jer su omogućile operaterima da biraju različite razine praga za razlikovanje pozitivnih i negativnih instanci, odnosno primjera ili slučajeva.

ROC krivulja daje vizualni prikaz sposobnosti modela da razlikuje između kategorija, obično između stvarno pozitivnih i stvarno negativnih instanci. ROC krivulja grafički prikazuje odnos između senzitivnosti, poznate i kao TPR (eng. *True Positive Rate*), i specifičnosti ili TNR (eng. *True Negative Rate*) modela, pri različitim razinama praga klasifikacije. Senzitivnost, odnosno TPR, je već definirana izrazom (3.3). Specifičnost, odnosno TNR je omjer broja ispravno predviđenih negativnih instanci i stvarno negativnih instanci.

$$\text{Specifičnost} = \frac{\text{Stvarno negativno}}{\text{Stvarno negativno} + \text{Lažno pozitivno}} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3.5}$$

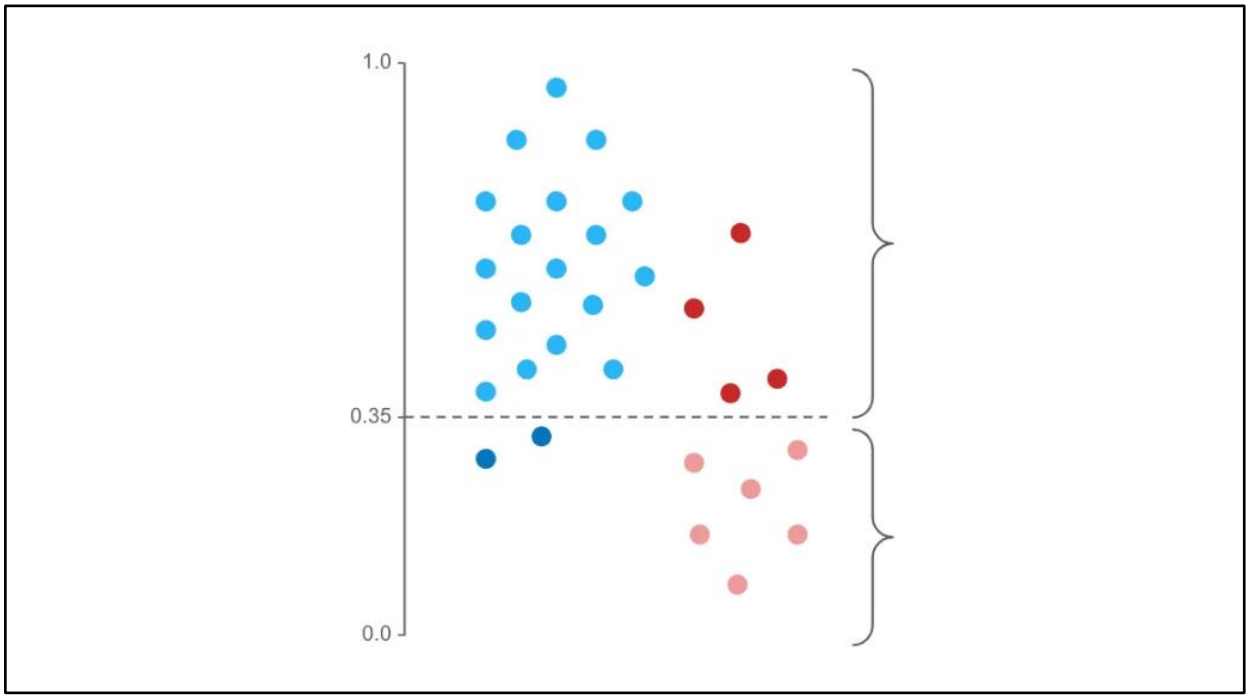
Uobičajeno je umjesto stvarne negativne stope, to jest specifičnosti, koristiti lažnu pozitivnu stopu, FPR (eng. *False Positive Rate*).

$$\text{Lažna pozitivna stopa} = 1 - \text{Specifičnost}$$

$$\text{Lažna pozitivna stopa} = \frac{\text{Lažno pozitivno}}{\text{Lažno pozitivno} + \text{Stvarno negativno}} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{3.6}$$

U nastavku će biti opisana konstrukcija ROC krivulje po koracima. Temeljem dobivenog klasifikacijskog modela, koji je predmet analize, za svaki podatak iz skupa za testiranje dobiva se vjerojatnost da podatak pripada pozitivnoj kategoriji.

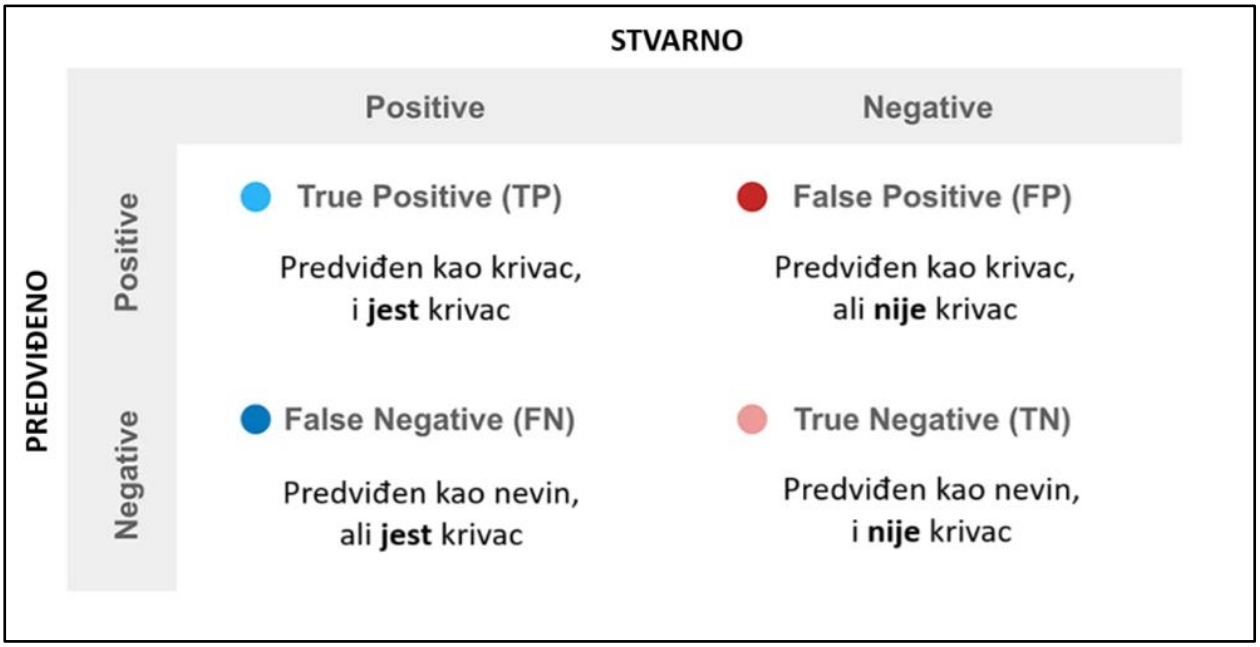
Ako se radi o primjeru pravosudnog sustava, zapravo se dobiva vjerojatnost da je promatrana osoba počinila kazneno djelo. Primjer kad se kao prag za pripadnost pozitivnoj klasi (odnosno proglašavanje osobe krivom) koristi vrijednost 0,35, prikazana je na Slici 3.2. Osobe za koje je procijenjena vjerojatnost da su počinile zločin veća od odabranog praga, smatraju se krivcima, dok se osobe ispod navedenog praga smatraju nevinima. U primjeru prikazanom na slici, prag je postavljen na vrijednost 0,35, te je u nastavku provedena diskusija koje su instance ispravno ili neispravno klasificirane.



Slika 3.2. Primjer klasifikacijskog modela [14]

Na slici se plave točkice odnose na osobe koje su počinile zločin. Svijetlo plave točkice, koje se nalaze iznad odabranog praga, označuju zločince koji su prema klasifikacijskom modelu proglašeni krivima, to jest stvarno pozitivne slučajeve (TP), dok tamno plave točkice, ispod praga, predstavljaju krivce koji su ostali na slobodi, odnosno lažno negativne slučajeve (FN). Crvene točkice prikazuju osobe koje nisu počinile zločin. Svijetlo crvene točkice, ispod praga, označuju nevine osobe koje su i proglašene nevinima, to jest stvarno negativne slučajeve (TN), dok tamno crvene točkice predstavljaju nevine osobe koje su pogrešno proglašene, odnosno lažno pozitivne slučajeve (FP). Može se zaključiti da su pogrešno klasificirani podaci oni koji su tamno plavi ili tamno crveni.

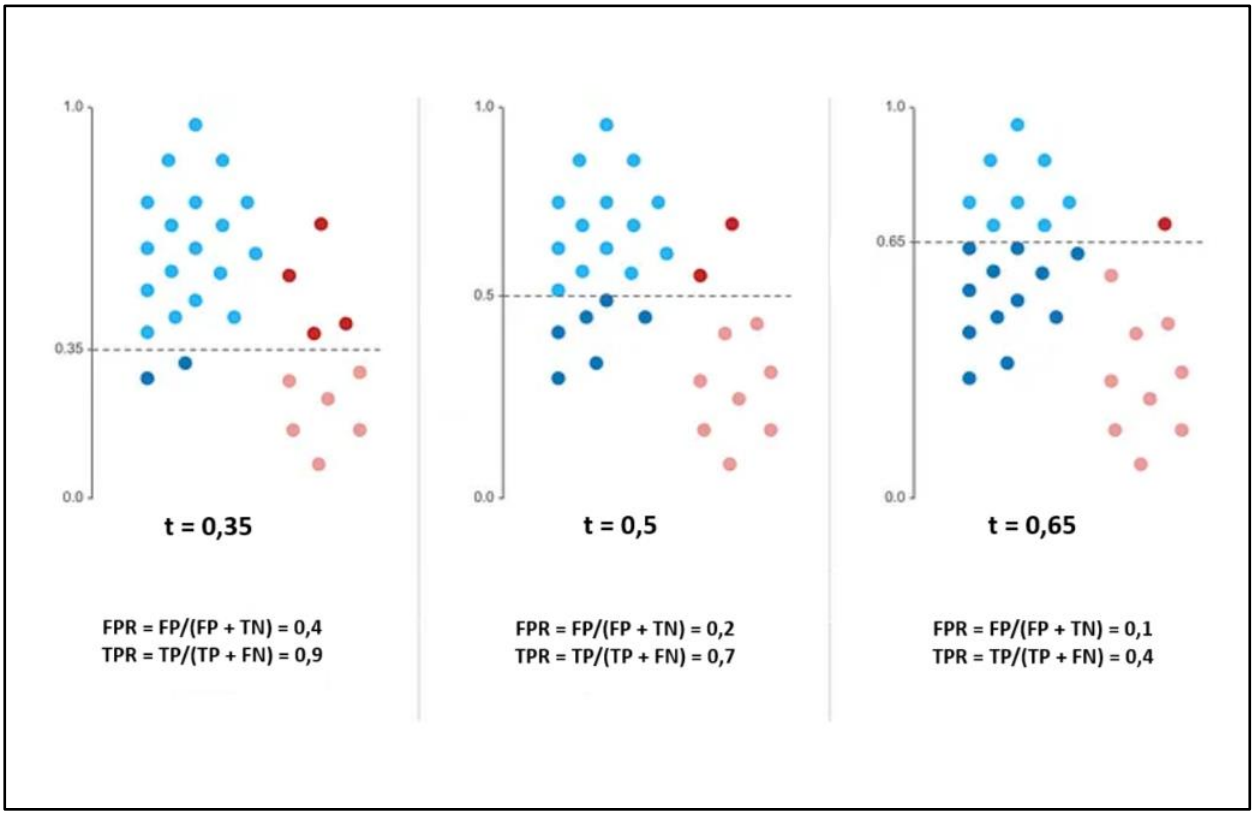
Matrica zabune uz korištenje prethodnih oznaka i za promatrani primjer, prikazana je na Slici 3.3.



Slika 3.3. Primjer matrice zabune

Vezano uz promatrani primjer, TPR ili stvarna pozitivna stopa predstavlja udio krivaca koje su s pravom proglašeni krivim temeljem klasifikacijskog modela. Stvarna negativna stopa, odnosno TNR, je udio nevinih osoba koji su s pravom proglašeni nevinima. Lažna pozitivna stopa, FPR, može se shvatiti kao udio nevinih osoba koje su osuđene pogreškom klasifikatora.

Na Slici 3.4. prikazane su vrijednosti FPR i TPR za različite pragove klasifikacije. Situacija za odabrani prag od 0,35, prikazana je na lijevoj slici. Iz dobivenih se vrijednosti može zaključiti da je točno klasificirano 90% pozitivnih instanci, to jest detektirano je ispravno 90% krivaca. Pogrešno je klasificirano 40% negativnih instanci, što znači da je 40% nevinih osoba pogreškom optuženo. U promatranom se primjeru može primijetiti da rezultati TPR-a i FPR-a opadaju povećanjem praga. Ako je prag jednak nuli, vrijednosti TPR-a i FPR-a obje će biti 100%, a ako je jednak jedan, vrijednosti TPR-a i FPR-a obje će biti 0%.

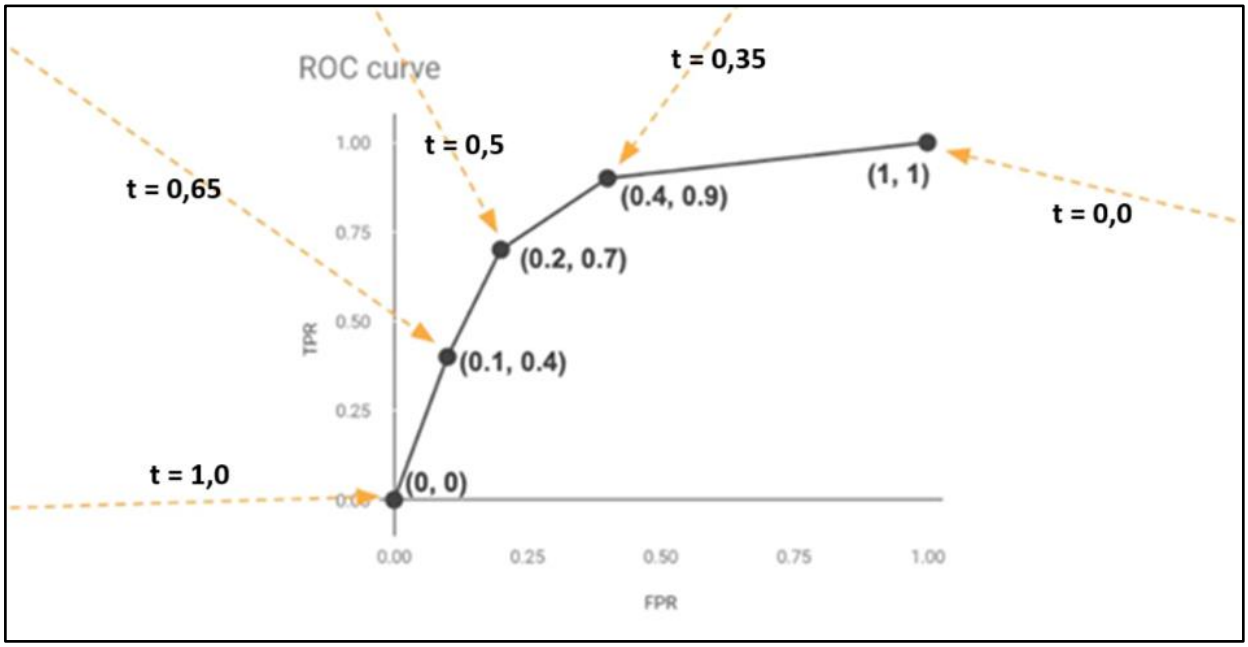


Slika 3.4. Primjer vrijednosti FPR-a i TPR-a za različite pragove

Iz prethodnog je jasno da se povećavanjem praga, bolje klasificiraju negativne instance, ali na račun pogrešne klasifikacije pozitivnih instanci. Odabir praga uvelike ovisi o problemu koji se promatra i rizicima neispravne klasifikacije.

Kod crtanja ROC krivulje, vrijednosti FPR-a prikazuju se na osi apscisa, a vrijednosti TPR-a na osi ordinata. Pritom se za različito odabrane pragove, izračunaju FPR-a i TPR-a te se ucrtaju u graf, kao što je prikazano na Slici 3.5. Za više pragova bit će više točaka, koje se potom spajaju u ROC krivulju.

ROC krivulja za prethodno razmatrani primjer prikazana je na Slici 3.5. Gledano s desna na lijevo, prva točka dobije se kad je prag na nuli, druga za prag jednak 0,35, i tako dalje (vidi [14]).

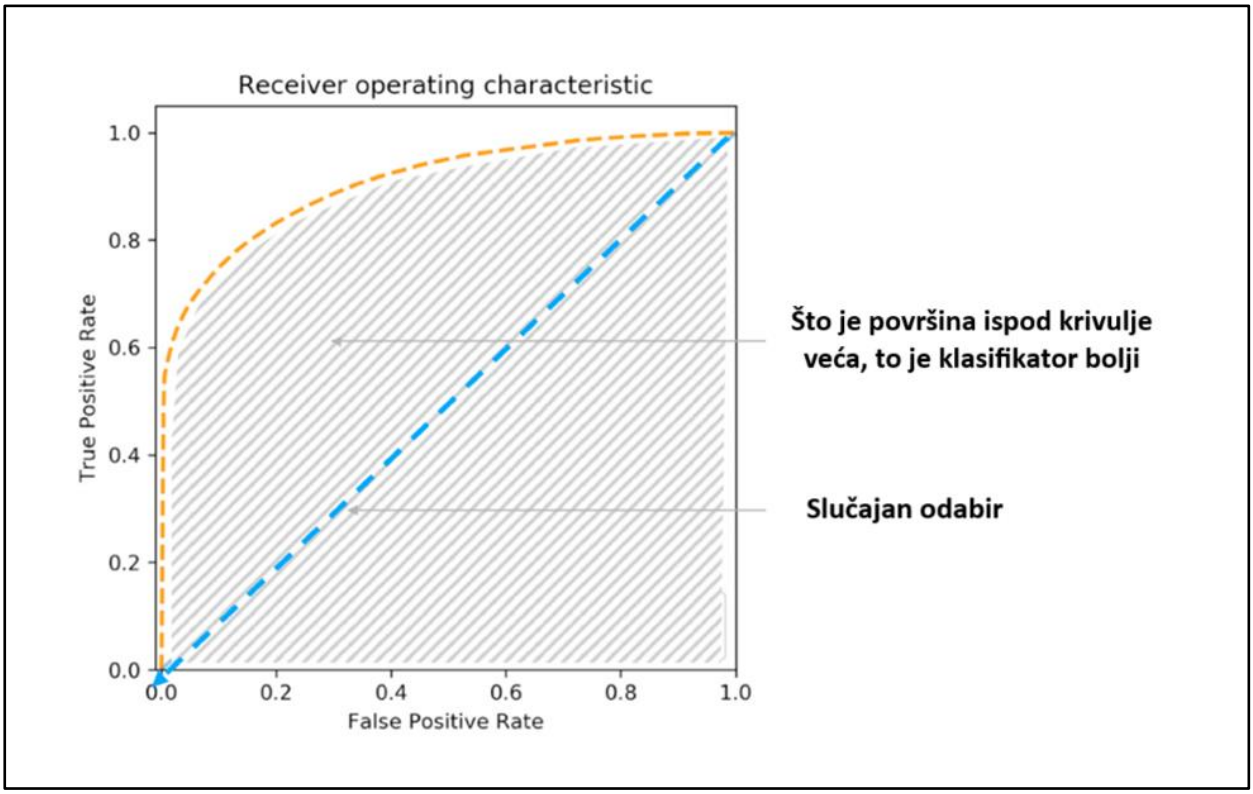


Slika 3.5. Primjer ROC krivulje

ROC krivulja omogućuje analizu performansi modela pri različitim pragovima klasifikacije. Promjenom praga, može se prilagoditi senzitivnost i specifičnost modela prema potrebama problema. ROC krivulja pruža sveobuhvatan pogled na performanse modela, posebno koristan kada postoji potreba za uravnotežavanjem senzitivnosti i specifičnosti. Razumijevanje i interpretacija ROC krivulje ključna je za donošenje informiranih odluka o učinkovitosti klasifikacijskih modela u različitim područjima primjene.

3.1.7. Površina ispod ROC krivulje

Površina ispod ROC krivulje ili AUC numerička je mjera koja kvantificira ukupnu performansu klasifikacijskog modela. Ova mjera koristi se za procjenu sposobnosti modela da razlikuje pozitivne i negativne instance pri odabranim razinama praga klasifikacije (Slika 3.6.).



Slika 3.6. Površina ispod ROC krivulje

AUC mjeri površinu ispod ROC krivulje. ROC krivulja, kao što je rečeno, predstavlja odnos između senzitivnosti i specifičnosti pri različitim pragovima klasifikacije. Ako AUC ima vrijednost blizu 1, to ukazuje na visoke performanse modela. Drugim riječima, model je sposoban razlikovati između pozitivnih i negativnih instanci s visokom točnošću. Vrijednost AUC blizu 0,5 ukazuje na to da je model ekvivalentan nasumičnom klasificiranju, a njegova ROC krivulja izgleda kao dijagonala kvadrata. Vrijednosti manje od 0,5 sugeriraju obrnuto, odnosno da je model lošiji od nasumičnog klasificiranja (vidi [15]).

AUC pruža kumulativnu procjenu performansi modela, bez obzira na zadani prag klasifikacije. Ova mjera je posebno korisna u situacijama gdje je ravnoteža između senzitivnosti i specifičnosti važna, a omogućuje i usporedbu modela na različitim područjima primjene.

4. PREDVIĐANJE USPJEŠNOSTI U FORMULI 1 I EVALUACIJA MODELA

Predviđanje performansi sustava može biti važno iz različitih razloga. Jedan od primjera je planiranje kapaciteta, što omogućuje planiranje potrebne infrastrukture kako bi performanse bile optimalne i troškovi minimalni. Drugi primjer je elastično skaliranje, pri čemu se sustav prilagođava promjenama opterećenja zbog osiguranja pouzdanosti. Ostali razlozi mogu biti otkrivanje anomalija ili nepravilnih ponašanja, optimizacija performansi, unapređenje korisničkog iskustva, te istraživanje i razvoj u svrhu otkrivanja novih tehnologija. Postojeći pristupi predviđanju performansi obično se oslanjaju na dvije različite tehnike, analitičko modeliranje (eng. *Analytical Modeling*) ili AM i strojno učenje (eng. *Machine Learning*) ili ML.

Analitičko modeliranje tradicionalna je tehnika za predviđanje performansi sustava u različitim kontekstima. Ono koristi stručno znanje unutarnje dinamike sustava i pripadajući matematički model koji opisuje preslikavanje parametara na performanse. Nedostatak analitičkog modela je to što precizni analitički model pojedinog sustava često nije poznat, pa se koriste pojednostavljeni modeli, koji dovode do smanjenja točnosti.

S druge strane, strojno učenje promatra stvarno ponašanje sustava pod različitim uvjetima kako bi moglo izgraditi statistički model ponašanja. Modeliranje temeljeno na strojnom učenju postaje sve popularnije za predviđanje performansi složenih sustava. Zbog povećanja računalnih kapaciteta, bez obzira na zahtjevnost proračuna, prednost predviđanja performansi sve se više okreće strojnom učenju (vidi [16]).

U nastavku se istražuje primjena strojnog učenja u predviđanju performansi u Formuli 1. Analizirat će se korištenje strojnog učenja za predviđanje uspješnosti timova i vozača u Formuli 1 na temelju prethodnih rezultata i ostalih parametara. Podaci uključuju rezultate kvalifikacija, karakteristike staze, meteorološke uvjete i povijest performansi vozača i timova.

4.1. Primjena strojnog učenja u Formuli 1

Formula 1, kao jedan od najintenzivnijih sportova, iznimno je kompleksno okruženje gdje se isprepleću vrhunska tehnologija, inženjerske vještine i vozačko umijeće. Uspjeh u Formuli 1 osigurava složen sustav, koji se sastoji od tehnologije i konstrukcije bolida, taktičkih odluka i lukavstva timova, te vozačkih sposobnosti i hrabrosti natjecatelja. Ovakva kombinacija Formulu 1 čini vrhuncem inženjerskih dostignuća i talenta vozača. Formula 1 predstavlja zanimljivo područje za primjenu modela strojnog učenja u predviđanju uspješnosti timova i vozača.

U ovom radu, istražit će se primjena modela strojnog učenja u predviđanju performansi okruženja Formule 1, fokusirajući se na predviđanje rezultata utrka. Rezultati utrka koje će se predviđati odnose se na 2021. godinu tijekom koje se vodila najneizvjesnija borba za naslov svjetskog prvaka posljednjih godina. Postava timova i vozača za 2021. godinu prikazana je na Slici 4.1. Cilj je razviti model koji će predviđati uspjeh timova i vozača na različitim utrkama, na temelju mnogobrojnih i raznolikih podataka. Model bi trebao pružiti uvid u predvidljivost rezultata u ovom sportu, što se može pokazati kao vrlo izazovan zadatak.

Slika 4.1. Postava timova i vozača Formule 1 u 2021. godini

4.1.1. Prikupljanje i analiza podataka

Prikupljanje i analiza podataka ključni su koraci u procesu primjene strojnog učenja za predviđanje performansi u Formuli 1. Potrebno je prikupiti raznovrsne podatke koji obuhvaćaju širok spektar informacija. Takvi će podaci omogućiti razvijanje preciznog modela za predviđanje performansi timova i vozača.

U ovom će se poglavlju opisati postupak prikupljanja podataka, njihova analiza i priprema skupa relevantnih podataka za razvoj modela strojnog učenja. Korištenjem podataka o utrkama, kvalifikacijama, vozačima i konstruktorima, istražit će se mogućnosti predviđanja pobjednika, to jest vozača na pobjedničkom postolju.

Model će biti razvijen u programskom jeziku Python. U nastavku je detaljno opisan korišten kod.

Korištene biblioteke i funkcije i njihov opis dani su u nastavku: biblioteka „pandas“ koristi se za rad s podacima u DataFrame formatu, dok se „matplotlib.pyplot“ koristi za vizualizaciju podataka. „LabelEncoder“ služi za enkodiranje kategoričkih značajki, to jest postupak pretvaranja kategoričkih varijabli u numeričke vrijednosti koje računalni modeli mogu lakše interpretirati. „RandomForestClassifier“, „LogisticRegression“, „SVC“ i „DecisionTreeClassifier“ su modeli strojnog učenja koji će biti korišteni, a opisani su u prethodnim poglavljima. Iz „sklearn.metrics“ uvoze se različite metrike za evaluaciju modela, poput „accuracy_score“, „recall_score“, „f1_score“, „roc_curve“, „auc“, „confusion_matrix“, i tako dalje. Rad s višedimenzionalnim matricama omogućuje „numpy“, dok je za poboljšanje vizualizacije korišten „seaborn“.

Podaci su dobiveni korištenjem Ergast Developer API web servisa (http://ergast.com/mrd/) koji pruža povjesne podatke o utrkama, te su spremljeni u CSV datoteku pod imenom „results_2015_2022.csv“, a odnose se na razdoblje od 2015. do 2022. godine. U skripti u kojoj je definiran proračun, podaci se najprije učitaju. Varijabla „YEAR_TO_ANALYZE“ odnosi se na sezonu koja se želi predviđati, a ovdje je postavljena na 2021. godinu (Slika 4.2.).

```python
# Uvoz svih potrebnih biblioteka
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, roc_curve, auc, confusion_matrix, roc_auc_score,
precision_score, recall_score, f1_score
import numpy as np
import seaborn as sns

# Učitavanje CSV datoteke u DataFrame
data = pd.read_csv('results_2015_2022.csv')
YEAR_TO_ANALYZE = 2021

```

Slika 4.2. Uvoz svih potrebnih biblioteka i učitavanje CSV datoteke u DataFrame

Ova je sezona bila jedna od najuzbudljivijih i najemotivnijih sezona posljednjih godina. Natjecanje za naslov prvaka svijeta obilježila je žestoka borba između Mercedesovog vozača Lewisa Hamiltona i Red Bullovog vozača Maxa Verstappena. Izvanredne vozačke sposobnosti i nepopustljivi karakteri obojice vozača pružili su publici nebrojeno nezaboravnih trenutaka, stoga je 2021. godina najzanimljiviji izbor za predmet istraživanja.

Nakon što su podaci učitani, potrebno ih je obraditi, odnosno preformulirati kako bi se iz njih moglo dobiti podatak o kojem smisleno ovisi predviđanje uspješnosti vozača, odnosno tima. Podaci su obrađeni tako da se izračuna DNF (eng. *Did Not Finish*) omjer. DNF označava slučaj kada vozač nije uspio završiti utrku. Ako utrka nije završena, što je označeno kao „status != 'Finished'“, vozaču se pridružuje zadnje mjesto koje je označeno brojem dvadeset.

Najčešće, vozači neće završiti utrku zbog oštećenja na njihovom bolidu. Unatoč mnogim promjenama u sportu, oštećenje bolida ostaje gotovo neizbježno. Oštećenja se vrlo lako događaju, pogotovo kada se bolidi sudare. Vozači također mogu napraviti kontakt sa zidom ili nekim drugim dijelom staze, čak i pokupiti otpad, što može uzrokovati nepopravljivu štetu.

Oštećenje nije jedini razlog koji može spriječiti vozača da završi utrku. Krhkost bolida čini ih ranjivima u smislu mehaničkih i električnih problema, s obzirom da su napravljeni na način da pruže svoj maksimum do posljednje sekunde utrke. Tehnički kvar ne bi bio krivnja vozača, nego

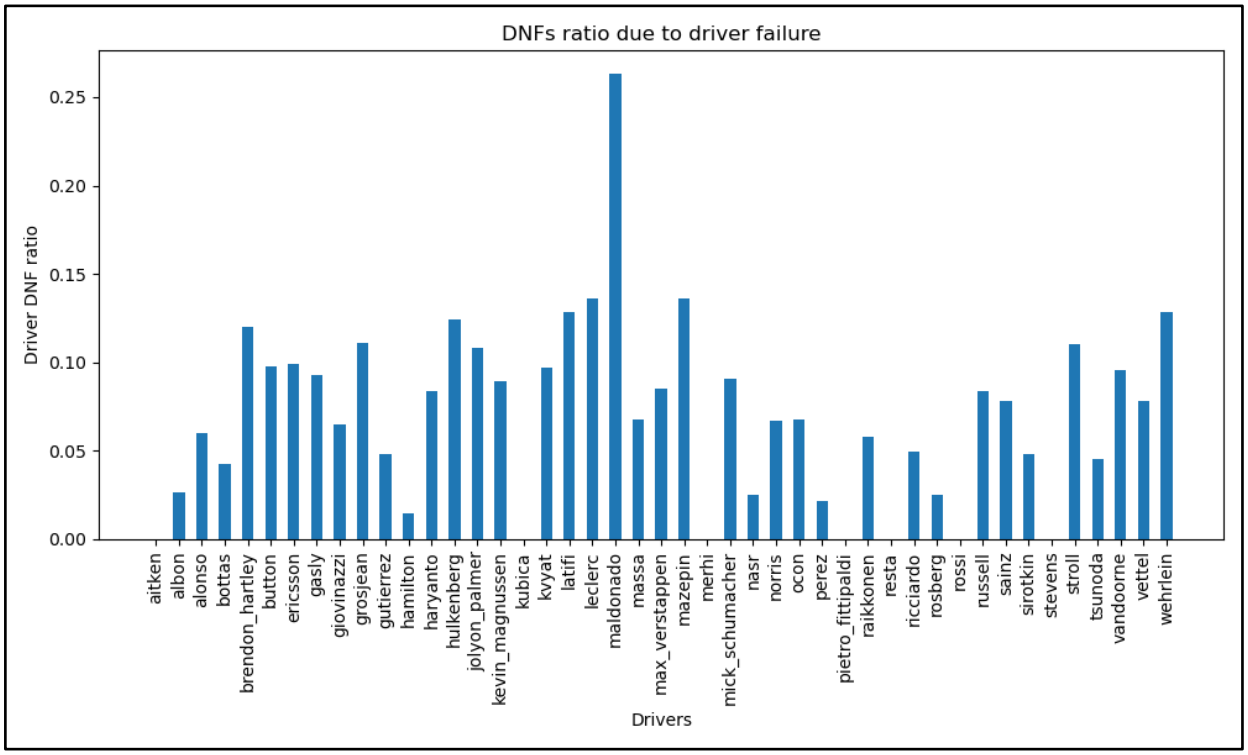
konstruktora, ali će ipak rezultirati DNF oznakom pored vozačevog imena. Greške konstruktora također se mogu očitovati u lošim strategijama utrke, poput loše procjene vremenskih uvjeta, zaustavljanja u boksu i potrošnje guma.

Nakon pojašnjenja mogućih uzroka odustajanja od utrke, mogu se definirati dvije kategorije kvarova, kvar uzrokovan vozačevom greškom i kvar uzrokovan greškom konstruktora. Popis mogućih pogrešaka koje se dogode, a zbog kojih utrka nije završena, klasificiraju se u dva skupa, „driver_failures“ i „constructor_failures“, u matrici podataka. Zatim se računaju varijable „driver_dnf_ratio“ i „constructor_dnf_ratio“ koje predstavljaju udio utrka u kojima vozač nije završio utrku zbog vlastite pogreške, odnosno pogreške konstruktora (Slika 4.3.).

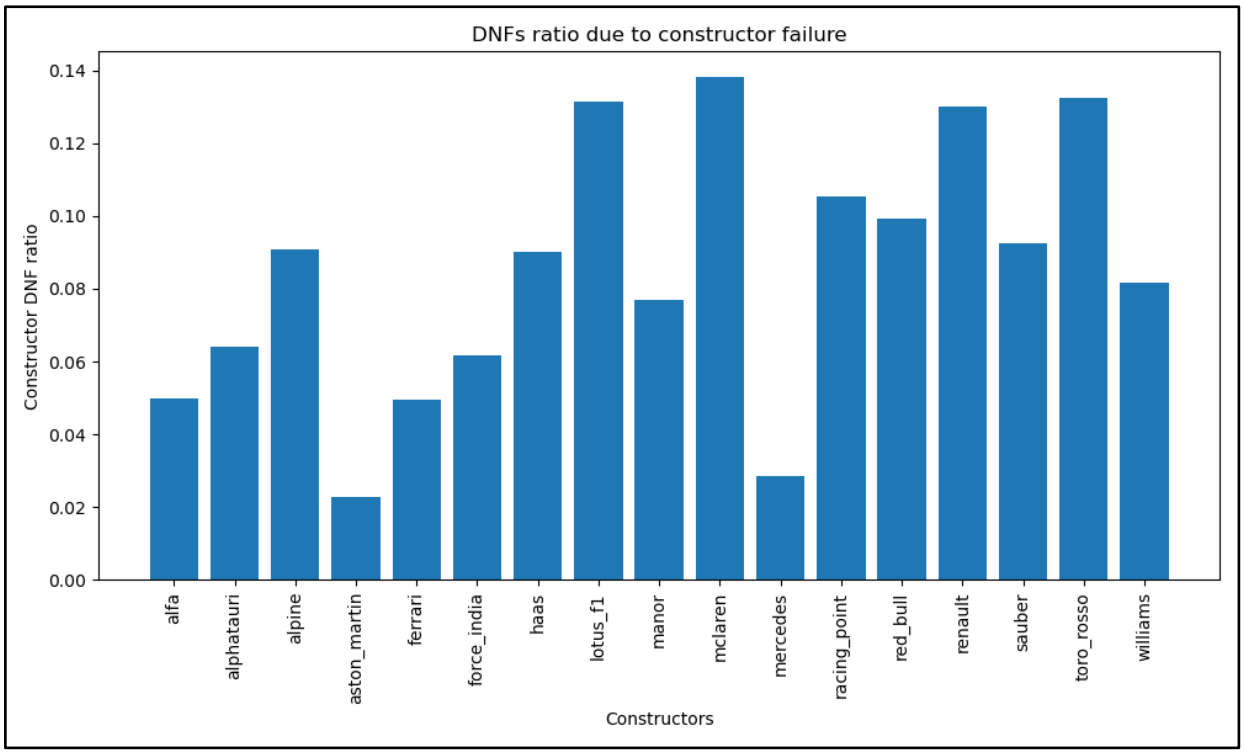
```
21  # Definiranje svih featurea u slučaju da se neka utrka nije završila, bilo to zbog vozača ili
    konstruktora
22  status_values = list(data['status'].unique())
23  driver_failures = ['Accident', 'Collision', 'Collision damage', 'Damage', 'Disqualified',
    'Suspension']
24  constructor_failures = {'Brakes', 'Cooling system', 'Debris', 'Differential', 'Driveshaft',
    'Electrical', 'Electronics', 'Engine', 'Exhaust', 'Fuel leak', 'Fuel pressure', 'Fuel pump',
    'Gearbox', 'Hydraulics', 'Mechanical', 'Oil leak', 'Overheating', 'Power Unit', 'Power loss',
    'Puncture', 'Radiator', 'Rear wing', 'Spun off', 'Steering', 'Technical', 'Transmission',
    'Turbo', 'Undertray', 'Vibrations', 'Water leak', 'Water pressure', 'Water pump', 'Wheel',
    'Wheel nut'}
25
26  # Formiraju se dvije nove kolone u kojima vrijednost 1 poprimaju odgovarajuće pozicije
27  data['driver_failure'] = data['status'].apply(lambda x: 1 if x in driver_failures else 0)
28  data['constructor_failure'] = data['status'].apply(lambda x: 1 if x in constructor_failures
    else 0)
29
30  # Izračun DNF omjera
31  driver_dnf_ratio = data.groupby('driver').sum()['driver_failure'] /
    data.groupby('driver').count()['driver_failure']
32  constructor_dnf_ratio = data.groupby('constructor').sum()['constructor_failure'] /
    data.groupby('constructor').count()['constructor_failure']
33
```

Slika 4.3. Obrada podataka i izračun DNF omjera

Na slikama 4.4. i 4.5. prikazani su navedeni omjeri za svakog vozača, odnosno svaki tim.



Slika 4.4. DNF omjer uzrokovan greškom vozača



Slika 4.5. DNF omjer uzrokovan greškom konstruktora

Najviši stupci u dijagramima odnose se na najnepouzdanije vozače i konstruktore. Samopouzdanje vozača suprotno je njegovoj nepouzdanosti, a pouzdanost konstruktora njihovoj nepouzdanosti, pa pripadajuću vjerojatnost računamo kao 1 – „failure“.

Slijedi definiranje ciljnih značajki za predviđanje. Kreiraju se nove značajke koje označavaju pobjedničke rezultate i pozicije na pobjedničkom postolju (Slika 4.6.).

```
57
58  # Definiranje novog featurea koji ćemo koristiti za predviđanje pobjednika
59  data['win'] = data['position'].apply(lambda x: 1 if x == 1 else 0)
60  data['podium'] = data['position'].apply(lambda x: 1 if x<4 else 0)
61
```

Slika 4.6. Definiranje ciljnih značajki za predviđanje

Za izgradnju modela kategoričke značajke potrebno je pretvoriti u numeričke. U promatranom je primjeru pretvorba kategoričkih značajki, „driver“ i „constructor“ provedena korištenjem Pythonove klase LabelEncoder() (Slika 4.7.).

```
61
62  # Koristimo podatke iz 2021. sezone
63  data_2021 = data.loc[data['season'] == YEAR_TO_ANALYZE]
64  N_rounds = max(data['round'].unique())
65
66  # Enkodiranje kategoričkih featurea
67  label_encoder_d = LabelEncoder()
68  label_encoder_c = LabelEncoder()
69  data_2021['constructor'] = label_encoder_c.fit_transform(data_2021['constructor'])
70  data_2021['driver'] = label_encoder_d.fit_transform(data_2021['driver'])
71
```

Slika 4.7. Enkodiranje kategoričkih značajki

U konačnici potrebno je odabrati značajke koje će biti korištene za treniranje i izgradnju modela. Prethodno je provedena analiza i utvrđeno je da samo neke od značajki imaju smisleni utjecaj na rezultate utrka. Odabrane značajke prikazane su na Slici 4.8., a uključuju vozača, tim, poziciju na kvalifikacijama, samopouzdanje vozača i pouzdanost konstruktora.

Treniranje modela u strojnom učenju znači proces tijekom kojeg algoritam uči prepoznati obrasce u podacima kako bi mogao donositi točne predikcije. Dijelovi koda opisani u ovom poglavlju odnose se na analizu i formulaciju podataka pogodnih za razvoj modela za predviđanje performansi u Formuli 1.

```
71
72  # Priprema i definiranje podataka koje ćemo koristiti za treniranje modela
73  chosen_features = ['driver', 'constructor', 'quali_pos', 'driver_confidence',
    'constructor_reliability']
74
```

Slika 4.8. Priprema podataka za treniranje modela

4.1.2. Razvoj modela

Razvoj modela strojnog učenja važan je korak u predikciji rezultata utrka Formule 1. U ovom će poglavlju biti opisan proces izrade modela strojnog učenja. Zadatak je izraditi model koji predviđa hoće li vozač završiti na podiju ili osvojiti veliku nagradu, to jest pobijediti u utrci.

Nakon pripreme podataka i odabranih značajki, te ciljane varijable, podaci se podijele u skup podataka za treniranje i skup za testiranje.

Kao ciljana varijabla ovdje se odabire varijabla „podium“, pomoću koje se predviđa hoće li vozač završiti na pobjedničkom postolju, ili „win“, ako se predviđa hoće li vozač pobijediti. Dodatno se definiranjem varijable „k_test“ odabire situacija u kojoj će se „k_test“ posljednjih utrka koristiti za testiranje i ti se podaci onda ne koriste u izgradnji modela, već samo za procjenu kvalitete modela. Trenutno je na Slici 4.9. ta varijabla postavljena na 5, što znači da se zadnjih 5 utrki sezone ne koristi za učenje modela već samo za predviđanje i evaluaciju performansi modela.

Na početku se inicijaliziraju instance klasa različitih algoritama: logistička regresija, slučajne šume, potporni vektori i stabla odlučivanja (Slika 4.9.).

```
87
88  # Definiranje modela pomoću kojega ćemo odabrati target, odnosno uvesti user input
89  def run_model(target):
90      X = data_2021[chosen_features]
91      y = data_2021[target]
92
93      # Podjela podataka na trening i test skupove
94      # k_test se može mijenjati ovisno o tome koliko utrka želimo koristiti za testiranje
95      k_test = 5
96      k_train = N_rounds - k_test
97      X_train = X.iloc[:-k_test * 20, :]
98      y_train = y.iloc[:-k_test * 20]
99      X_test = X.iloc[-k_test * 20:, :]
100     y_test = y.iloc[-k_test * 20:]
101
102     # Inicijalizacija modela, ovdje se može dodati još modela ako se želi
103     models = {
104         'LR': LogisticRegression(),
105         'RF': RandomForestClassifier(),
106         'SVC': SVC(probability=True),
107         'DT': DecisionTreeClassifier()
108     }
109
```

Slika 4.9. Definiranje ciljne varijable, podjela podataka na trening i test skupove te inicijalizacija algoritama strojnog učenja

Kao posljednji korak u razvoju modela, slijedi treniranje. Modeli se treniraju na skupu podataka za trening. Za prilagođavanje parametara modela podacima, koristi se funkcija „fit“, koja „trenira“ model koristeći ulazne značajke „X_train“ i ciljnu varijablu „y_train“. Dio koda vezan za treniranje modela bit će prikazan u sklopu idućeg poglavlja, u kojem će se govoriti o evaluaciji i optimizaciji performansi našeg modela.

4.1.3. Evaluacija i optimizacija performansi

Nakon razvoja modela strojnog učenja, provodi se evaluacija modela, nakon koje se dodatno može provesti optimizacija parametara za postizanje boljih perfiormansi. U tu svrhu koristi se ROC krivulju koja pruža uvid u performanse modela. ROC krivulja omogućuje analizu osjetljivosti i specifičnosti te usporedbu različitih modela s ciljem odabira onog najboljeg za predviđanje performansi u Formuli 1.

Nakon treniranja modela na temelju definiranih podataka, slijedi njegova evaluacija. Za svaki model i svaki prag vjerojatnosti izračunavaju se evaluacijske vrijednosti poput točnosti,

preciznosti, odziva, F-ocjene i matrice zabune. Pripadajući kod prikazan je na Slici 4.10. Ove metrike pomažu u procjenjivanju kako različiti pragovi utječu na performanse modela.

```
119
120     # Treniranje i evaluacija modela sa samom preciznošću i križnom validacijom
121     # Ovdje se mogu dodati i drugi metrički podaci poput F1 score, recall, itd.
122     for name, model in models.items():
123         classifier_results = {"Classifier": name}
124         classifier_metrics = []
125
126         # Treniranje modela na temelju iznad definiranih podataka
127         model.fit(X_train, np.array(y_train).ravel())
128
129         # y_pred = model.predict(X_test) #prediction with threshold=0.5
130         y_scores = model.predict_proba(X_test)[:, 1]
131
132         # Adjust threshold and make predictions
133         for threshold in threshold_values:
134             y_pred = (y_scores > threshold).astype(int)
135
136             # Metrics
137             accuracy = accuracy_score(y_test, y_pred)
138             precision = precision_score(y_test, y_pred)
139             recall = recall_score(y_test, y_pred)
140             f1 = f1_score(y_test, y_pred)
141             cm = confusion_matrix(y_test, y_pred)
142
143             # Store metrics
144             classifier_metrics.append({
145                 "Threshold": threshold,
146                 "Accuracy": accuracy,
147                 "Precision": precision,
148                 "Recall": recall,
149                 "F1-score": f1,
150                 "Confusion Matrix": cm
151             })
152
153             classifier_results["Metrics"] = classifier_metrics
154
```

Slika 4.10. Treniranje modela i evaluacija na različitim pragovima

Nakon treniranja, model se evaluira na test skupu podataka. Za svaki model izračuna se vrijednost predikcije „y_scores“, na temelju čega se generiraju ROC krivulje s pripadajućim površinama, odnosno AUC vrijednostima. Uz to se izračunava i Youdenov indeks koji pomaže u određivanju optimalnog praga vjerojatnosti koji maksimizira razliku između stvarne pozitivne stope ili TPR-a i lažne pozitivne stope ili FPR-a (Slika 4.11.).

```
155         # Predikcija i ispis matrice konfuzije
156         predicted_results[f'{name}_prediction'] = y_scores
157         cm = confusion_matrix(y_test, y_pred)
158         print(cm)
159 
160         # ROC i AUC, Youdenov indeks
161         fpr, tpr, thresholds = roc_curve(y_test, y_scores)
162         roc_auc = auc(fpr, tpr)
163 
164         # Crtanje ROC krivulje
165         plt.plot(fpr, tpr, color=colors[i], lw=2, label=f'{name} (AUC = {roc_auc:.2f})')
166 
167         # Youdenov indeks
168         JI = tpr - fpr
169 
170         # Optimalna vrijednost
171         OptimalThreshold = thresholds[np.argmax(JI)]
172 
173         # Spremanje metrika ROC krivulje u classifier_results
174         classifier_results["fpr"] = fpr
175         classifier_results["tpr"] = tpr
176         classifier_results["thresholds"] = thresholds
177         classifier_results["JI"] = JI
178         classifier_results["OptimalThreshold"] = OptimalThreshold
179         classifier_results["AUC"] = roc_auc
180 
181         results.append(classifier_results)
182 
183         i += 1
```

Slika 4.11. Predikcija i evaluacija modela, Youdenov indeks i optimalni prag

U sljedećem poglavlju slijedi prikaz konačnih rezultata i diskusija o dobivenim rješenjima. Analizirat će se dobivene ROC krivulje i odlučiti koji se model pokazao najpouzdanijim i najuspješnijim u predviđanju pobjedničkog postolja, odnosno pobjednika utrke.

4.1.4. Interpretacija i usporedba rezultata

Nakon što su prikupljeni i analizirani podaci, te je razvijen, evaluiran i optimiziran model strojnog učenja, ključno je s razumijevanjem interpretirati dobivene rezultate kako bi se donijeli ispravni zaključci o učinkovitosti modela.

Ovo poglavlje posvetit će se analizi dobivenih ROC krivulja i AUC vrijednosti, uspoređujući performanse modela u slučaju odabira podija i u slučaju odabira pobjede kao cilja istraživanja. Uz to će se prikazati i matrice zabune za svaki cilj kako bi se dobila bolja predodžba o točnosti i pouzdanosti modela.

PREDVIĐANJE PODIJA

Promatrajući prvo slučaj kada je cilj postavljen na predviđanje pobjedničkog postolja, analizirane su performanse nekoliko različitih modela strojnog učenja, kao što su logistička regresija, slučajne šume, potporni vektori i stabla odlučivanja. U evaluaciju svakog pojedinog modela uključeno je i određivanje najboljeg praga vjerojatnosti za postizanje ravnoteže između metrika evaluacije, to jest točnosti, preciznosti, senzitivnosti i F-ocjene. Utvrđuje se da performanse različitih modela variraju ovisno o odabranom pragu vjerojatnosti (Slika 4.12.).

LOGISTIC REGRESSION	RANDOM FOREST	SVC	DECISION TREES
Threshold: 0.15 Accuracy: 0.8800 Precision: 0.5600 Recall: 0.9333 F1-score: 0.7000 Confusion Matrix: [[74 11] [1 14]]	Threshold: 0.15 Accuracy: 0.8300 Precision: 0.4643 Recall: 0.8667 F1-score: 0.6047 Confusion Matrix: [[70 15] [2 13]]	Threshold: 0.15 Accuracy: 0.9200 Precision: 0.7333 Recall: 0.7333 F1-score: 0.7333 Confusion Matrix: [[81 4] [4 11]]	Threshold: 0.15 Accuracy: 0.8300 Precision: 0.4615 Recall: 0.8000 F1-score: 0.5854 Confusion Matrix: [[71 14] [3 12]]
Threshold: 0.3 Accuracy: 0.9000 Precision: 0.6316 Recall: 0.8000 F1-score: 0.7059 Confusion Matrix: [[78 7] [3 12]]	Threshold: 0.3 Accuracy: 0.8700 Precision: 0.5500 Recall: 0.7333 F1-score: 0.6286 Confusion Matrix: [[76 9] [4 11]]	Threshold: 0.3 Accuracy: 0.9200 Precision: 0.7692 Recall: 0.6667 F1-score: 0.7143 Confusion Matrix: [[82 3] [5 10]]	Threshold: 0.3 Accuracy: 0.8700 Precision: 0.5500 Recall: 0.7333 F1-score: 0.6286 Confusion Matrix: [[76 9] [4 11]]
Threshold: 0.5 Accuracy: 0.9400 Precision: 0.9091 Recall: 0.6667 F1-score: 0.7692 Confusion Matrix: [[84 1] [5 10]]	Threshold: 0.5 Accuracy: 0.8900 Precision: 0.6429 Recall: 0.6000 F1-score: 0.6207 Confusion Matrix: [[80 5] [6 9]]	Threshold: 0.5 Accuracy: 0.9400 Precision: 0.9091 Recall: 0.6667 F1-score: 0.7692 Confusion Matrix: [[84 1] [5 10]]	Threshold: 0.5 Accuracy: 0.8700 Precision: 0.5714 Recall: 0.5333 F1-score: 0.5517 Confusion Matrix: [[79 6] [7 8]]

Slika 4.12. Performanse modela za predviđanje podija

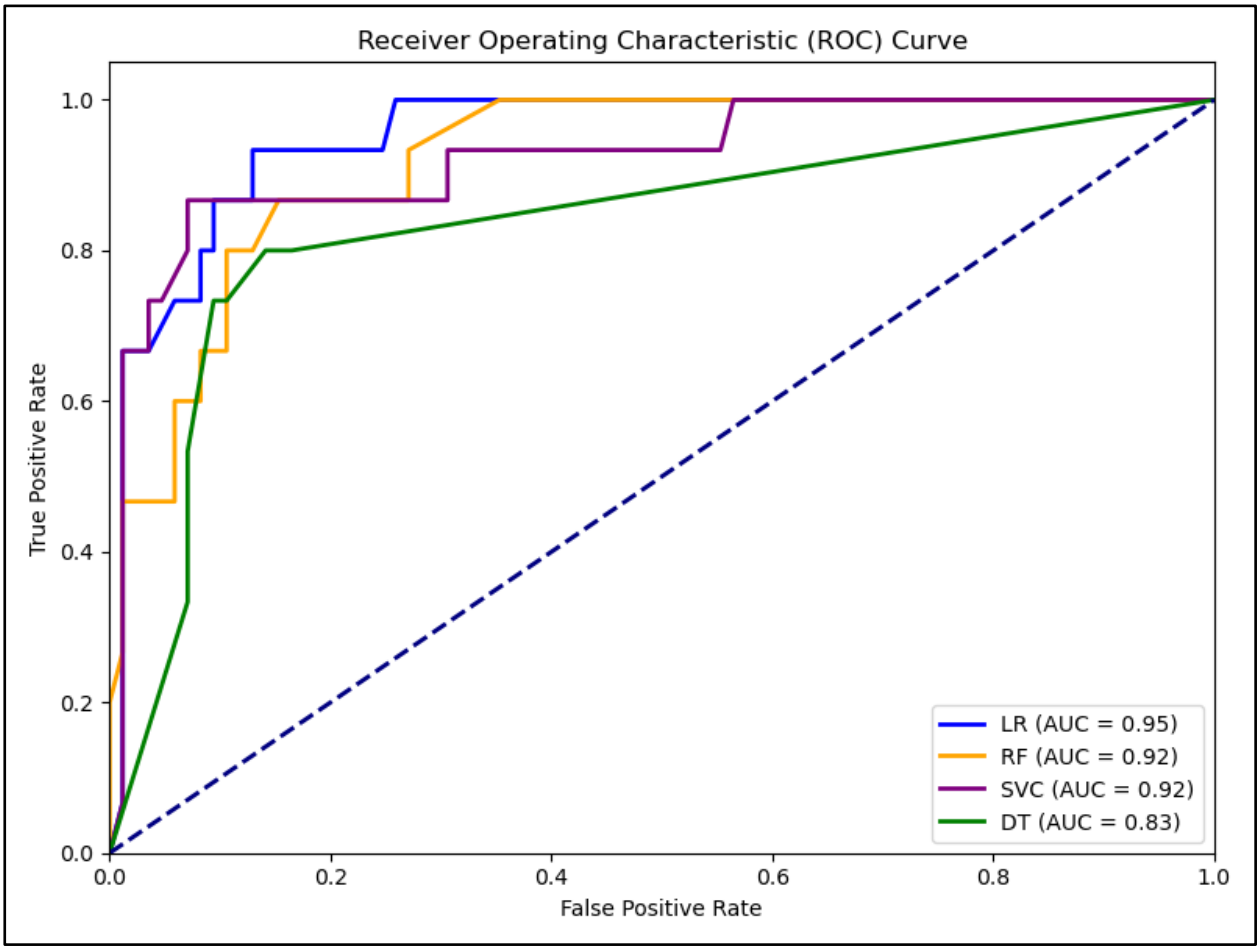
Matrice zabune opisuju performanse modela, pružajući uvid u uspješnost predviđanja podija u odnosu na stvarne ishode. U prikazanim matricama zabune, vrijednost u prvom retku i prvom stupcu predstavlja točno predviđen broj vozača koji nisu završili na podiju (TN), vrijednost u prvom retku i drugom stupcu je broj vozača koji nisu završili na podiju ali je netočno predviđeno

da hoće (FP), vrijednost u drugom retku i prvom stupcu je broj vozača koji jesu završili na podiju iako je netočno predviđeno da neće (FN), dok vrijednost u drugom retku i drugom stupcu predstavlja točno predviđen broj vozača koji jesu završili na podiju (TP).

U promatranom primjeru klase nisu dobro izbalansirane. Fokus je postavljen na ispravno predviđanje pobjedničkog postolja, odnosno traži se da stvarna pozitivna stopa bude čim veća, uz razumno malu lažno pozitivnu stopu.

Iz ispisanih matrica zabune i ostalih metrika, zaključuje se da model logističke regresije pokazuje najbolje performanse, posebno kod nižih pragova vjerojatnosti. Logistička regresija uspješno razlikuje pozitivne i negativne instance, te ostvaruje najbolju ravnotežu između stvarno pozitivnih i stvarno negativnih instanci kod praga vjerojatnosti od 0,15. Međutim, logistička regresija povećava stvarno negativne instance povećanjem praga vjerojatnost. Iako to radi pod cijenu povećanja lažno negativnih instanci, i dalje ostvaruje najbolje rezultate u usporedbi s ostalim modelima. Što se tiče vrijednosti ostalih metrika, točnost, preciznost i F-ocjena povećavaju s povećanjem praga vjerojatnosti, dok senzitivnost opada. Kad je riječ o modelima slučajnih šuma, potpornih vektora i stabla odlučivanja, njihove se metrike ponašaju na približno isti način, ali daju nešto slabije rezultate.

Na temelju ROC krivulja i AUC vrijednosti potvrđuje se zaključak da je logistička regresija najpouzdaniji model za ovaj zadatak. Logistička regresija daje rezultate s najvišim AUC vrijednostima (0,95), što sugerira najbolju sveobuhvatnu performansu u odabiru vozača koji završavaju na pobjedničkom postolju. Na Slici 4.13. može se vidjeti prednost logističke regresije u odnosu na ostale modele.



Slika 4.13. ROC krivulje za predviđanje podija

PREDVIĐANJE PODIJA

Slijedi analiza performansi modela u predviđanju pobjednika utrke (Slika 4.14.). Uspoređujući modele identificirat će se onaj koji ima najveću uspješnost u predviđanju pobjednika te prag vjerojatnosti koji daje najbolje rezultate.

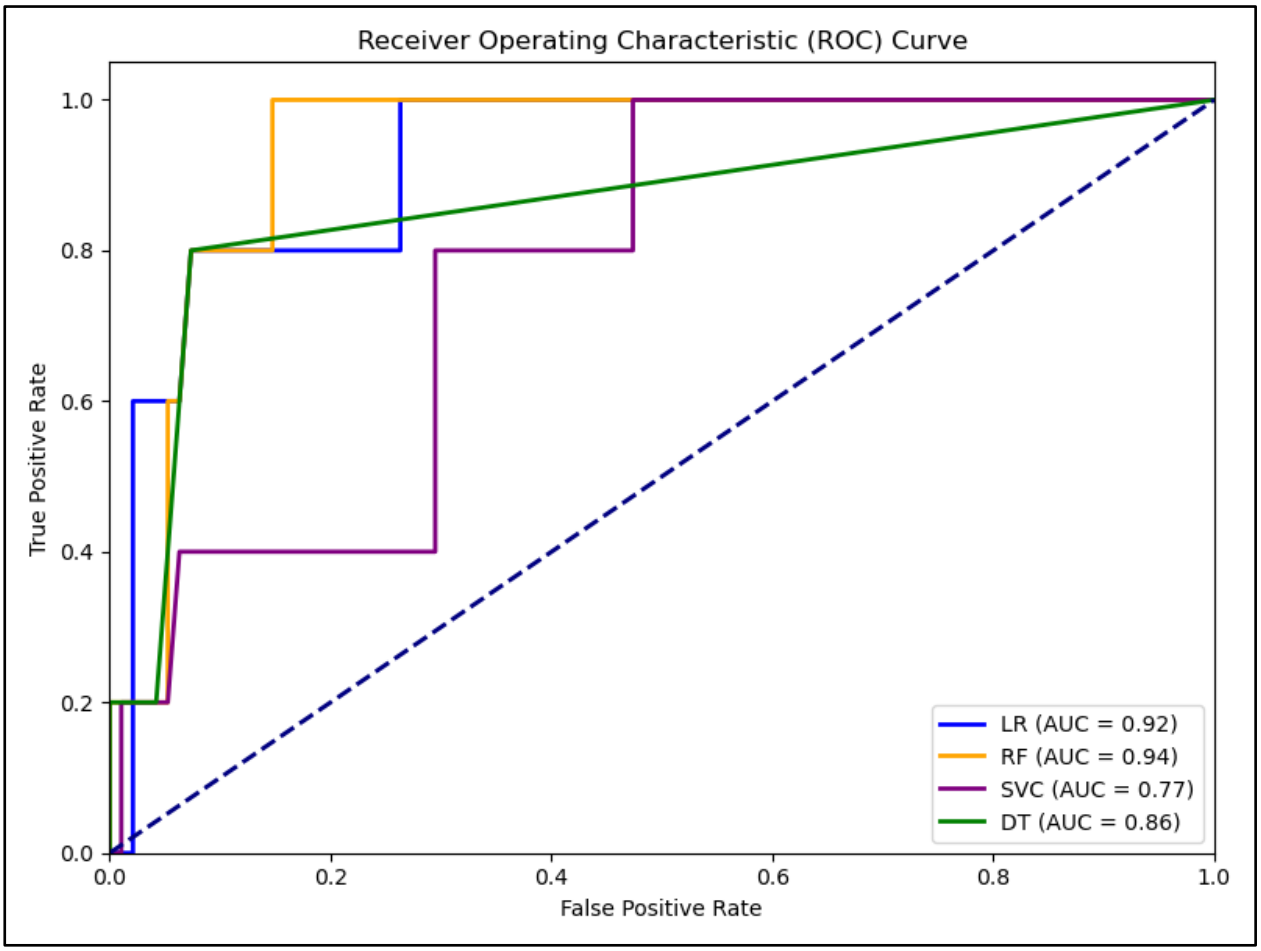
LOGISTIC REGRESSION	RANDOM FOREST	SVC	DECISION TREES
Threshold: 0.15 Accuracy: 0.9200 Precision: 0.3636 Recall: 0.8000 F1-score: 0.5000 Confusion Matrix: [[88 7] [1 4]]	Threshold: 0.15 Accuracy: 0.9000 Precision: 0.3077 Recall: 0.8000 F1-score: 0.4444 Confusion Matrix: [[86 9] [1 4]]	Threshold: 0.15 Accuracy: 0.9500 Precision: 0.0000 Recall: 0.0000 F1-score: 0.0000 Confusion Matrix: [[95 0] [5 0]]	Threshold: 0.15 Accuracy: 0.9200 Precision: 0.3636 Recall: 0.8000 F1-score: 0.5000 Confusion Matrix: [[88 7] [1 4]]
Threshold: 0.3 Accuracy: 0.9600 Precision: 0.6000 Recall: 0.6000 F1-score: 0.6000 Confusion Matrix: [[93 2] [2 3]]	Threshold: 0.3 Accuracy: 0.9200 Precision: 0.3333 Recall: 0.6000 F1-score: 0.4286 Confusion Matrix: [[89 6] [2 3]]	Threshold: 0.3 Accuracy: 0.9500 Precision: 0.0000 Recall: 0.0000 F1-score: 0.0000 Confusion Matrix: [[95 0] [5 0]]	Threshold: 0.3 Accuracy: 0.9200 Precision: 0.3333 Recall: 0.6000 F1-score: 0.4286 Confusion Matrix: [[89 6] [2 3]]
Threshold: 0.5 Accuracy: 0.9500 Precision: 0.0000 Recall: 0.0000 F1-score: 0.0000 Confusion Matrix: [[95 0] [5 0]]	Threshold: 0.5 Accuracy: 0.9400 Precision: 0.3333 Recall: 0.2000 F1-score: 0.2500 Confusion Matrix: [[93 2] [4 1]]	Threshold: 0.5 Accuracy: 0.9500 Precision: 0.0000 Recall: 0.0000 F1-score: 0.0000 Confusion Matrix: [[95 0] [5 0]]	Threshold: 0.5 Accuracy: 0.9600 Precision: 1.0000 Recall: 0.2000 F1-score: 0.3333 Confusion Matrix: [[95 0] [4 1]]

Slika 4.14. Performanse modela za predviđanje pobjednika

U ovom su slučaju klase podataka u još većem disbalansu u odnosu na prethodni jer se nastoji izdvojiti samo jedan pobjednik.

Na temelju matrica zabune, slučajne šume pokazuju najbolju sposobnost razlikovanja između pozitivnih i negativnih instanci. Ovaj model ima dobre performanse kod nižih pragova vjerojatnosti (t = 0,15), s minimalnim brojem lažno negativnih instanci. Iako ima nešto veći broj lažno pozitivnih instanci, još uvijek pruža bolje rezultate od ostalih modela. Logistička regresija povećanjem praga vjerojatnosti smanjuje broj lažno pozitivnih instanci, ali istovremeno povećava broj lažno negativnih instanci. Potporni vektori pokazuju najslabije performanse u odnosu na ostale modele, s velikim brojem lažno pozitivnih i lažno negativnih instanci. Stabla odlučivanja, unatoč boljim performansama kod nižih pragova vjerojatnosti, imaju značajan broj lažno pozitivnih instanci. Točnost, preciznost, senzitivnost i F-ocjena, imaju sličan trend kao i kod predviđanja podija, kod svih modela.

ROC krivulje, prikazane na Slici 4.15., zajedno s ispisanim AUC vrijednostima, još jednom dokazuju da slučajne šume predstavljaju najbolji model za predviđanje pobjednika. S najvišom AUC vrijednosti (0,94), ovaj model ima najbolji omjer stvarno pozitivnih i stvarno negativnih instanci, te može s velikom vjerojatnošću predvidjeti pobjednika utrke.



Slika 4.15. ROC krivulje za predviđanje pobjednika

Prikazi ROC krivulja i AUC vrijednosti omogućuju vizualnu usporedbu performansi modela. Rezultati sugeriraju da je logistička regresija najbolji odabir kod predviđanja pobjedničkih postolja, dok se slučajne šume odabiru u slučaju predviđanja pobjednika utrke. Jasno je da će se teže predvidjeti pobjednik velike nagrade pošto on može biti samo jedan vozač, dok je pobjedničko postolje predviđeno za trojicu.

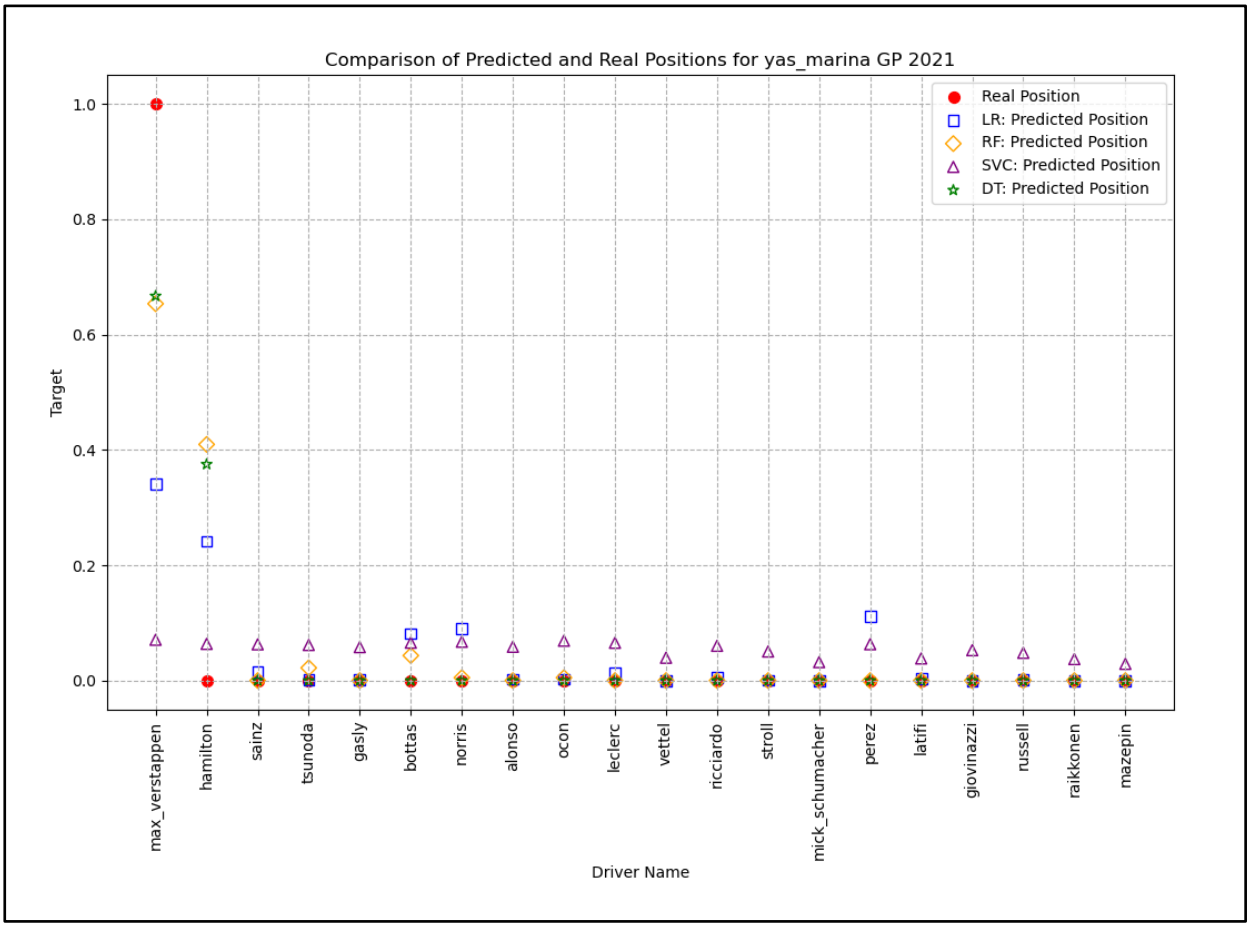
4.1.5. Ograničenja modela

Iako su rezultati modela strojnog učenja impresivni u predviđanju performansi timova i vozača, mora se naglasiti da se korištena znanja i vještine u stvarnosti ne mogu mjeriti s tehnologijom i metodama kojima se služe inženjeri unutar stvarnih momčadi u Formuli 1.

Inženjeri u Formuli 1 imaju na raspolaganju znatno kvalitetnije i detaljnije podatke. Njihovi podaci prikupljeni su pomoću stotina senzora prisutnih na svakom bolidu i obuhvaćaju podatke u stvarnom vremenu. Koriste sustave obrade podataka koji su napredniji i optimizirani za posebne performanse bolida, strategije utrka i različite scenarije. Njihove simulacije uvažavaju aerodinamičke karakteristike bolida, izdržljivost i ponašanje guma u različitim vremenskim uvjetima, i specifična obilježja svake staze (vidi [17]).

Model ovog istraživanja koristi ograničen skup podataka i bavi se generaliziranim predikcijama koje su temeljene na podacima i ishodima prošlih utrka. Iako predstavlja značajan korak u razumijevanju i predviđanju performansi u okruženju Formule 1, tek je mali dio složenog sustava kojeg koriste inženjeri u momčadima Formule 1, što se vidi iz sljedećeg primjera.

Na Slici 4.16. prikazane su vjerojatnosti pobjede pojedinih vozača dobivene korištenjem razmatranih modela, dok crvena točkica s vrijednošću apscise 1 predstavlja stvarnog pobjednika. Svima ostalima, koji nisu pobijedili, pridružena je vjerojatnost 0.



Slika 4.16. Predviđeni i stvarni rezultati za Veliku nagradu Abu Dhabija

Kad je riječ o predviđanju rezultata posljednje utrke sezone, u 2021. godini u Formuli 1, dolazi se do jednog od najdramatičnijih i najkontroverznijih završetaka sezone u povijesti ovog sporta, što će ujedno i pokazati ograničenja upotrebe modela u nepredvidljivim situacijama.

Uoči posljednje utrke, Velike nagrade Abu Dhabija, nizozemski vozač Max Verstappen i britanski vozač Lewis Hamilton, imali su jednak broj bodova ostvarenih tijekom sezone, a upravo je iznenadni i nepredvidljivi ljudski faktor presudio u žestokoj borbi za naslov svjetskog prvaka na stazi Yas Marina.

Jedan od vozača zabio se u barijeru na samom kraju utrke i izazvao izlazak sigurnosnog automobila na stazu, što je dovelo do neočekivanog obrata. Iako je Hamilton bio u vodstvu i sigurno vozio prema svom osmom naslovu svjetskog prvaka, ovaj je incident omogućio Verstappenu da u posljednjem krugu utrke pretekne Hamiltona i prvi put u karijeri postane svjetski prvak.

Verstappenova pobjeda označila je kraj dugogodišnje dominacije Mercedesa i otvorila vrata novoj eri Formule 1, kojom je uvjerljivo zavladao Red Bull.

Na osnovu dobivenih rezultata može se zaključiti da su svi promatrani modeli najveću vjerojatnost pobjede dali Maxu Verstappenu i time uspjeli predvidjeti da će Max Verstappen pobijediti u posljednjoj utrci sezone (Slika 4.16.). No, bez obzira na takvo predviđanje, može se zaključiti da je okruženje Formule 1 izuzetno nepredvidljivo i promjenjivo. Inženjeri u Formuli 1, zahvaljujući sofisticiranijem pristupu istraživanju, uvijek će imati nedvojbenu prednost u predviđanju rezultata i pobjednika u utrkama, ali ipak slučajni faktori još uvijek igraju značajnu ulogu i potpuna sigurnost u predviđanju pobjednika nije moguća.

Slika 4.17. Velika nagrada Abu Dhabija 2021. godine

5. ZAKLJUČAK

Ovaj završni rad predstavlja istraživanje primjene modela strojnog učenja u predviđanju performansi timova i vozača u Formuli 1. Cilj je bio pokazati kako se podaci prikupljeni iz prošlih utrka mogu iskoristiti za stvaranje modela koji predviđa vjerojatnost postizanja odabranih rezultata. Rezultati odabrani za istraživanje su osvajanje podija i pobjede u utrkama.

Prije primjene strojnog učenja, opisane su vrste i koraci procesa strojnog učenja koji uključuju prikupljanje i analizu podataka, odabir i razvoj modela, te evaluaciju i optimizaciju modela. Objašnjeni su algoritmi strojnog učenja, kao što su logistička regresija, slučajne šume, potporni vektori, stabla odlučivanja i drugi. Posebna pažnja posvećena je evaluaciji modela strojnog učenja, pri čemu su promatrani klasifikacijski problemi.

Obrađene su razne metrike evaluacije, poput točnosti, matrice zabune, preciznosti, senzitivnosti i F-ocjene. Naglasak rada je na primjeni i korištenju ROC krivulje i površina ispod ROC krivulje, odnosno AUC. Detaljno je opisan postupak dobivanja ROC krivulje i objašnjen značaj AUC vrijednosti, što će biti odlučujući faktor u evaluaciji modela istraživanja.

Povrh teoretskog dijela, u rad je uključen praktičan primjer sustava u inženjerstvu, koji će demonstrirati teoretske zaključke i omogućiti uvid u implementaciju modela strojnog učenja. Pokazano je kako se algoritmi strojnog učenja mogu primijeniti u realnim situacijama za predviđanje i donošenje odluka.

Model je ostvario zadovoljavajuće rezultate, pri čemu je logistička regresija identificirana kao najbolji model za predviđanje osvajanja podija, a slučajne šume kao najbolji model za predviđanje pobjednika. Važno je napomenuti da su u istraživanju uočena ograničenja te da je model, sukladno očekivanjima, inferioran u odnosu na tehnologije i resurse s kojima raspolažu inženjeri momčadi.

Unatoč ograničenjima, ovo istraživanje pokazuje potencijal korištenja strojnog učenja u predviđanju performansi okruženja Formule 1, kao i drugih sportova. Ovaj rad pridonosi razumijevanju uloge podataka i analitike u strojnom učenju i pruža potrebna znanja za buduće primjene u ovom zanimljivom području.

LITERATURA

[1] Krishnan, B., Hamberg, L., Schulz, N.: "Elements of AI", s Interneta, https://course.elementsofai.com/hr/4/1, 28. travnja 2024.

[2] Dir.hr: "Što je strojno učenje?", s Interneta, https://dir.hr/sto-je-strojno-ucenje/, 28. travnja 2024.

[3] Johnson, D.: "Machine Learning versus Deep Learning", s Interneta, https://www.guru99.com/hr/machine-learning-vs-deep-learning.html, 28. travnja 2024.

[4] Aleksić, D.: "Mogućnosti primjene metoda strojnog učenja", s Interneta, https://repozitorij.fpz.unizg.hr/islandora/object/fpz%3A2355/datastream/PDF/view, 29. travnja 2024.

[5] Trehan, D.: "Linear Regression", s Interneta, https://pub.towardsai.net/linear-regression-explained-f5cc85ae2c5c, 29. travnja 2024.

[6] Tpoint Tech.: "Logistic Regression in Machine Learning", s Interneta, https://www.javatpoint.com/logistic-regression-in-machine-learning, 17. svibnja 2024.

[7] Tpoint Tech.: "Linear Regression versus Logistic Regression", s Interneta, https://www.javatpoint.com/linear-regression-vs-logistic-regression-in-machine-learning, 18. svibnja 2024.

[8] Kharkar, D.: "About Decision Tree Algorithms", s Interneta, https://www.linkedin.com/pulse/decision-tree-algorithms-dishant-kharkar/, 18. svibnja 2024.

[9] Rout, P.: "Random Forest in Machine Learning", s Interneta, https://dotnettutorials.net/lesson/random-forests-inmachine-learning/, 18. svibnja 2024.

[10] Tpoint Tech.: "Support Vector Machine Algorithm", s Interneta, https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm, 19. svibnja 2024.

[11] Cloudflare, Inc.: "What Is Neural Network", s Interneta, https://www.cloudflare.com/learning/ai/what-is-neural-network/, 19. svibnja 2024.

[12] Tpoint Tech.: "Performance Metrics in Machine Learning", s Interneta, https://www.javatpoint.com/performance-metrics-in-machine-learning, 19. svibnja 2024.

[13] Raj, R.: “Classification and Regression in Machine Learning”, s Interneta, https://www.enjoyalgorithms.com/blogs/classification-and-regression-in-machine-learning, 1. lipnja 2024.

[14] Cortez, V.: “Understanding the ROC curve in three visual steps”, s Interneta, https://towardsdatascience.com/understanding-the-roc-curve-in-three-visual-steps-795b1399481c, 1. lipnja 2024.

[15] Sreeram, A.: “A beginner’s guide to understanding and using ROC AUC in machine learning”, s Interneta, https://adithsreeram.medium.com/a-beginners-guide-to-understanding-and-using-roc-auc-in-machine-learning-7b4507be1c99, 2. lipnja 2024.

[16] Didona, D., Quagila F., Romano, P.: “Enhancing Performance Prediction Robustness by Combining Analytical Modeling and Machine Learning”, s Interneta, https://www.dpss.inesc-id.pt/~romanop/files/papers/ICPE15.pdf, 2. lipnja 2024.

[17] Rajvanshi, A.: “Machine Learning In Formula 1 – A Look Into The Future”, s Interneta, https://akshatrajvanshi.medium.com/machine-learning-in-formula-1-a-look-into-the-future-8fa6238aa95d, 2. lipnja 2024.

SAŽETAK

Ovaj završni rad predstavlja istraživanje primjene modela strojnog učenja u predviđanju performansi timova i vozača u Formuli 1 pomoću ROC (eng. *Receiver Operating Characteristic*) krivulja. Cilj je pokazati kako se podaci iz prošlih utrka mogu iskoristiti za stvaranje modela koji predviđa vjerojatnost postizanja odabranih rezultata, poput osvajanja pobjedničkog postolja i pobjede u utrkama. Kroz ovaj rad opisuje se proces strojnog učenja, objašnjavaju se algoritmi strojnog učenja, način prikupljanja i obrade podataka, te evaluacija performansi modela pomoću različitih metrika. Model logističke regresije pokazuje se najboljim za predviđanje osvajanja podija, dok su slučajne šume najuspješnije u predviđanju pobjednika. Unatoč ograničenjima, istraživanje pokazuje potencijal korištenja strojnog učenja u predviđanju performansi inženjerskih sustava, pružajući temelj i potrebna znanja za buduća istraživanja.

Ključne riječi: strojno učenje, algoritmi strojnog učenja, evaluacija performansi modela strojnog učenja, predviđanje rezultata, Formula 1

SUMMARY

This thesis presents a study on the application of machine learning models in predicting the performance of teams and drivers in Formula 1 using ROC (Receiver Operating Characteristic) curves. The aim is to show how data from past races can be used to create a model that predicts the likelihood of achieving selected results, such as winning a podium or a race. This thesis describes the process of machine learning, explains the machine learning algorithms, the methods of data collection and processing, and the evaluation of model performance using various metrics. The logistic regression model is shown to be the best for predicting podium finishes, while random forests are the most successful in predicting winners. Despite the limitations, the research demonstrates the potential of using machine learning in predicting the performance of engineering systems, providing a foundation and necessary knowledge for future research.

Keywords: machine learning, machine learning algorithms, evaluation of machine learning model performance, results prediction, Formula 1