Modeli strojnog učenja u inženjerstvu materijala

Martinović, Marinela

SVEUČILIŠTE U ZAGREBU
FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE
SVEUČILIŠNI PREDDIPLOMSKI STUDIJ

Marinela Martinović

ZAVRŠNI RAD

Zagreb, rujan 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE
SVEUČILIŠNI PREDDIPLOMSKI STUDIJ

Marinela Martinović

METODE STROJNOG UČENJA U INŽENJERSTVU MATERIJALA

ZAVRŠNI RAD

Voditelj rada: doc. dr. sc. Miroslav Jerković
Članovi ispitnog povjerenstva: doc. dr. sc. Miroslav Jerković
doc. dr. sc. Erna Begović Kovač
izv. prof. dr. sc. Dajana Kučić Grgić

Zagreb, rujan 2023.

Zahvaljujem se mentoru doc.dr.sc. Miroslavu Jerkoviću na predloženoj temi, pomoći i savjetima pri izradi rada.

Ovaj rad izrađen je na Fakultetu kemijskog inženjerstva i tehnologije u Zagrebu, Zavod za matematiku, pod mentorstvom doc.dr.sc. Miroslava Jerkovića u rujnu 2023. godine.

SAŽETAK

Svaki dan ljudi udahnu oko dvadeset tisuća puta, što bi iznosilo otprilike trinaest do petnaest tisuća litara zraka. Dakle, neminovno je zaključiti da je glavna komponenta života na Zemlji onda svakako zrak. Kao glavna komponenta, neophodan je za opstanak svih živih bića. Zrak je dakle mješavina plinova i to prvenstveno dušika i kisika, a zatim ugljikova dioksida, argona, neona, vodika i drugih. S obzirom na količinu zraka koju ljudi dnevno udišu važno je voditi računa o kvaliteti zraka koji se udiše. Bilo kakvo onečišćenje u zraku utječe na zdravlje ljudi, pravilan rad i aktivnost tijela, cirkulaciju krvi, pravilan rad mozga, a samim tim i na kvalitetu života.
Velik je broj parametara koji utječu na onečišćenje zraka. Međutim, većina čestica i plinova koji utječu na kvalitetu zraka nastaju ljudskim djelovanjem i uglavnom potječu iz urbanih i industrijskih područja. Tako primjerice, veliki gradovi poput Pekinga ili Delhija zbog velike naseljenosti i industrijalizacije imaju problem s tamnim smogom. Koncentracija onečišćujućih tvari u zraku zajedno sa povezanim zdravstvenim rizicima mogu se prikazati indeksom kvalitete zraka, AQI.

Indeks kvalitete zraka može se odrediti „ručno“ mjerenjem utjecaja pojedinih parametara; temperature, tlaka, vidljivosti i slično te sumiranjem njihovih utjecaja. Ali, kako bi se brže i jednostavnije odredio utjecaj pojedinih parametara na AQI moguća je primjena algoritama strojnog učenja uz korištenje nekog od dostupnih programskih jezika.

Dakle, cilj ovog rada jest primjenom algoritama strojnog učenja i programskog jezika R pronaći model koji najbolje predviđa indeks kvalitete zraka te usporediti mjerne vrijednosti indeksa kvalitete zraka s vrijednostima dobivenim predikcijom.

Ključne riječi: strojno učenje, programski jezik R, zrak, indeks kvalitete zraka (AQI)

SUMMARY

Every day, people breathe in about twenty thousand times, which would amount to approximately thirteen to fifteen thousand liters of air. Therefore, it is inevitable to conclude that the main component of life on Earth is air. As the main component, it is necessary for the survival of all living beings. Air is a mixture of gases, primarily nitrogen and oxygen, followed by carbon dioxide, argon, neon, hydrogen, and others. Considering the amount of air that people breathe in daily, it is important to take care of the quality of air that is inhaled. Any air pollution affects human health, proper body work and activity, blood circulation, proper brain work, and therefore the quality of life.

There are a large number of parameters that influence air pollution. However, most of the particles and gases that affect air quality are human-made and mainly originate from urban and industrial areas. For example, large cities such as Beijing or Delhi have a problem with dark smog due to high population and industrialization. The concentration of pollutants in the air together with the associated health risks can be represented by the air quality index, AQI.

The air quality index can be determined "manually" by measuring the influence of individual parameters; temperature, pressure, visibility, and similar but also summarizing their effects. But, to determine the influence of individual parameters on AQI more quickly and simply, the application of the machine learning algorithms combined with one of the available programming languages is possible.

Therefore, the goal of this work is to find a model that best predicts the air quality index by applying machine learning algorithms and the R programming language and to compare the measured values of the air quality index with the values obtained by prediction.

Keywords: machine learning, programming language R, air, air quality index (AQI).

SADRŽAJ

1. UVOD

Novi materijali bili su najveća dostignuća svakog doba. Svi ti materijali ključni su za rast, sigurnost i kvalitetu života ljudi. Dakle, znanstvenici i inženjeri za materijale utječu na život ljudi svakodnevno. Osim što dizajniraju nove materijale, inženjeri materijala također se bave i istraživanjem svojstava postojećih materijala i tvari kako bi se poboljšala njihova kemijska i fizička svojstva. Kao znanstvenici i inženjeri materijala, oni integriraju kemiju, fiziku, matematiku i biologiju s inženjerstvom kako bi odgovorili na globalne izazove važne za tehnologiju, društvo i okoliš. Jedna od dužnosti inženjera materijala jest procjena mogućih utjecaja postojećih i novih materijala i proizvoda na okoliš i zdravlje.

Zrak je glavna komponenta života na Zemlji. Samim tim neophodan je za opstanak svih živih bića. Zrak je mješavina plinova i to prvenstveno dušika i kisika, a zatim ugljikova dioksida, argona, neona, vodika i drugih. Ugljikov dioksid komponenta je zraka koja može imati i dobar i loš utjecaj. Dakle, kada ljudi dišu ispuštaju ugljikov dioksid, koji biljke kasnije, zajedno sa sunčevom svjetlošću, koriste za proizvodnju hrane. Međutim, velike količine ugljikovog dioksida nastaju primjerice i kada automobili ili elektrane izgaraju naftu, ugljen ili benzin. Takav ugljikov dioksid predstavlja problem, odnosno ima štetan utjecaj na okoliš jer upravo tako nastali ugljikov dioksid izaziva globalno zatopljenje. Kako je jedna od zadaća inženjera materijala procjena utjecaja materijala na okoliš i zdravlje, znanstvenici dolaze na ideju izrade materijala koji pohranjuje ugljik. Prirodni materijali koji smanjuju udio ugljika u atmosferi bili bi drvo, pluto, konoplja i alge. Osim navedenih znanstvenici su kreirali i neke umjetne kao što su bioplastika, tepih pločice, olivinski pijesak, modificirani beton i slično.

Prije kreiranja materijala, za smanjenje ugljika u atmosferi koji šteti okolišu i ljudskom zdravlju, potrebno je napraviti uvid koliko je zapravo okoliš odnosno zrak zagađen. Koncentracija onečišćujućih tvari u zraku zajedno sa povezanim zdravstvenim rizicima mogu se prikazati indeksom kvalitete zraka, AQI. Osim ručnim mjerenjem indeks kvalitete zraka, također ga je moguće predvidjeti primjenom strojnog učenja i programskog jezika R.

Strojno učenje nastoji razviti pogodne računalne metode odnosno matematičke modele koji će računalu omogućiti učenje iz skupa podataka, a zatim testiranje modela i predviđanje izlaznih vrijednosti.

Dakle, primjenom strojnog učenja prvenstveno je određeno onečišćenje zraka, odnosno indeks kvalitete zraka. Kada inženjeri materijala dobiju uvid u podatke o indeksu kvalitete zraka, u mogućnosti su primijeniti nove materijale na nekom određenom području i nakon toga ponovno ispitati indeks kvalitete zraka kako bi dobili uvid u kvalitetu materijala. Pod kvalitetom materijala podrazumijeva se utjecaj materijala na smanjenje količine ugljikovog dioksida, a samim tim i onečišćenja zraka, odnosno utjecaj na zdravlje ljudi.

2. STROJNO UČENJE

Jedna od najjednostavnijih, odnosno možemo reći najlakših definicija strojnog učenja jest da strojno učenje podučava računala ono što je ljudima prirodno; učenje iz iskustva. Računalo samo po sebi nije u mogućnosti učiti ili graditi znanje na temelju ranije pohranjenih podataka s obzirom da mu za to nedostaju neke osnovne biološke strukture živih bića. Stoga, strojno učenje nastoji razviti pogodne računalne metode odnosno matematičke modele koji će računalu omogućiti učenje. [1]

2.1. UVOD U STROJNO UČENJE

Umjetna inteligencija se definira kao sposobnost stroja da oponaša inteligentno ljudsko ponašanje. Sustavi umjetne inteligencije koriste se za rješavanje složenih zadataka na način koji je sličan načinu na koji ljudi rješavaju probleme. [3]

Strojno učenje može se definirati kao podpodručje umjetne inteligencije koje omogućuje strojevima da automatski uče iz podataka te prethodno stečenih iskustava kako bi identificirali obrasce ponašanja podataka te dali predviđanja uz minimalnu ljudsku intervenciju. [5] Zadatak strojnog učenja bio bi da omogući računalu izgradnju određenog modela skupa podataka kako bi ono moglo predvidjeti kamo smjestiti nove podatke. [2] Strojno učenje omogućuje softverima, odnosno softverskim aplikacijama, da postanu preciznije u predviđanju ishoda bez da su za to prethodno eksplicitno programirane. [4] Algoritmi strojnog učenja primjenjuju računalne metode pomoću kojih informacije prikupljaju direktno iz podataka, bez oslonca na unaprijed određene teorijske jednadžbe i modele. Takvi algoritmi pronalaze određene prirodne uzorke u podacima na temelju čega dobivaju uvid, a zatim odlučuju i predviđaju. [1] Dakle, algoritmi strojnog učenja koriste „povijesne“ podatke kao ulaz za predviđanje novih izlaznih vrijednosti. [4] Oni se „hrane“ novim podacima i mogu potpuno samostalno učiti, razvijati se i prilagođavati. [5]

Važno je napomenuti da se metode primijenjene na određeni skup podataka mijenjaju i prilagođavaju svakom problemu, odnosno svakom skupu podataka, baš kako bi postigle najbolje moguće rezultate. Naravno, za postizanje najboljih mogućih rezultata potrebno je odabrati najpogodniji to jest najbolji model za ispitivani skup. [2] Također, za postizanje što boljih rezultata, pogodno bi bilo upotrijebiti veliku količinu podataka jer se algoritmi poboljšavaju povećanjem broja uzoraka dostupnih za učenje. [1]

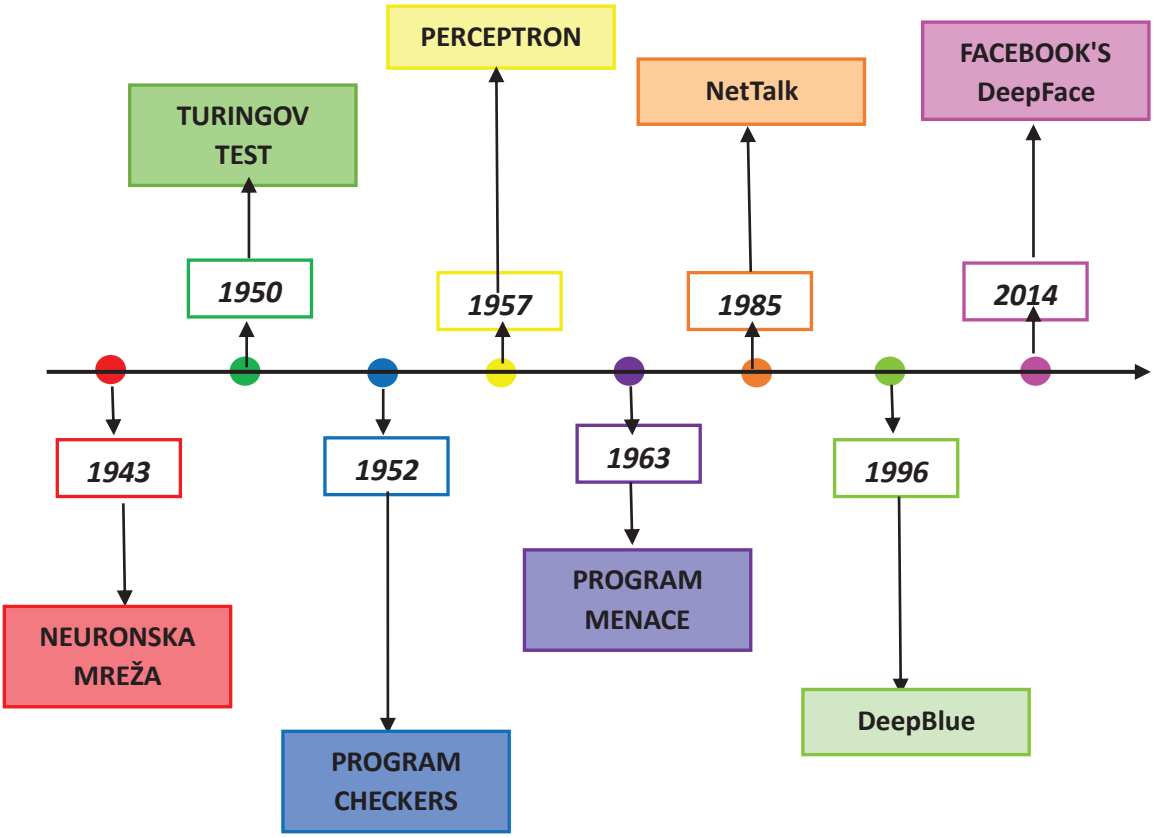
2.2. POVIJEST STROJNOG UČENJA

Povijesni razvoj strojnog učenja započinje 1943. godine kada su logičar Walter Pitts i neuroznanstvenik Warren McCulloch objavili prvo matematičko modeliranje neuronske mreže u svijetu za stvaranje algoritma koji oponaša ljudske misaone procese. Nadalje, 1950. godine, Alan Turing predložio je Turingov test prema kojem, ako stroj može uvjeriti ljudsko biće da je također čovjek onda se smatra „inteligentnim“. Ubrzo nakon toga, ljetni istraživački program na sveučilištu Dartmouth postao je rodnim mjestom umjetne inteligencije. Upravo je taj događaj prekretnica u daljnjim istraživanjima i inovacijama u području strojnog učenja. Dakle, počeli su se pojavljivati razni algoritmi i računalni programi koji su imali široku primjenu. Jedan od primjera takvog programa bio bi program Checkers Arthura Samuela koji je imao mogućnost igranja dame ili program Donalda Michieja pod nazivom MENACE, koji je imao mogućnost naučiti igrati savršenu igru križić-kružić. Godine 1957., Franck Rosenblatt je pokušao dizajnirati prvu računalnu neuronsku mrežu nazvanu „perceptron“ koja prima vizualne ulaze poput slika i stvara izlaze kao što su oznake. [6]

Daljnjim razvijanjem, nastali inteligentni strojevi imali su sve veću i veću primjenu. Neki od najpoznatijih izuma bili bi program NetTalk koji uči izgovarati riječi na sličan način kako to i beba radi, zatim IBM-ov program Deep Blue, nastao 1996. godine, koji pobjeđuje aktualnog svjetskog prvaka u šahu te neuronska mreža nastala od strane Google Brain tima, koja može naučiti prepoznavati mačke putem YouTube videa. [6]

Slika 2.2.1. Šahovska partija između svjetskog prvaka u šahu Garrya Kasparova i IBM-ovog računalnog programa DeepBlue. [9]

Nadalje, jedan od danas možda najpoznatijeg primjera „izuma“ tijekom evolucije strojnog učenja, svakako bi bio Facebookov DeepFace algoritam, koji je u stanju identificirati pojedince na fotografijama jednako precizno kako to može i čovjek. [6]



Slika 2.2.2. Vremenska crta strojnog učenja- povijesni razvoj 1943.-2014.g.

Kao što se može i uočiti strojno učenje je uvelike utjecalo na sve industrijske pravce diljem svijeta. Danas se ono koristi za različite zadatke, od prepoznavanja lica do automatizirane vožnje, od mode do poljoprivrede. Jedan od najvećih uspjeha današnjice na ovom području bila bi mogućnost obrade velike količine podataka u relativno kratkom vremenu, sa sve širom i širom primjenom. [7]

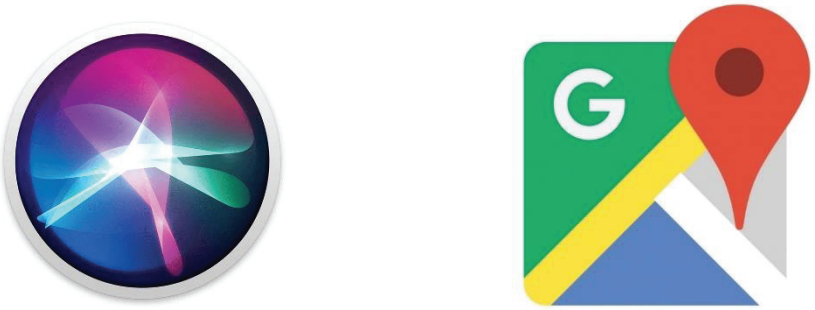
2.3. PRIMJENA STROJNOG UČENJA

Neprestanim porastom broja podataka koji se kontinuirano stvaraju i kruže svijetom, strojno učenje postalo je iznimno važno u sve više područja. [8] U današnjem svijetu, punom tehnologije, strojno učenje se koristi svakodnevno, bez da smo toga i svjesni. Google Maps, Google Assistant i Spotify samo su neke od aplikacija bazirane na strojnom učenju koje se svakodnevno koriste u svijetu. [9]

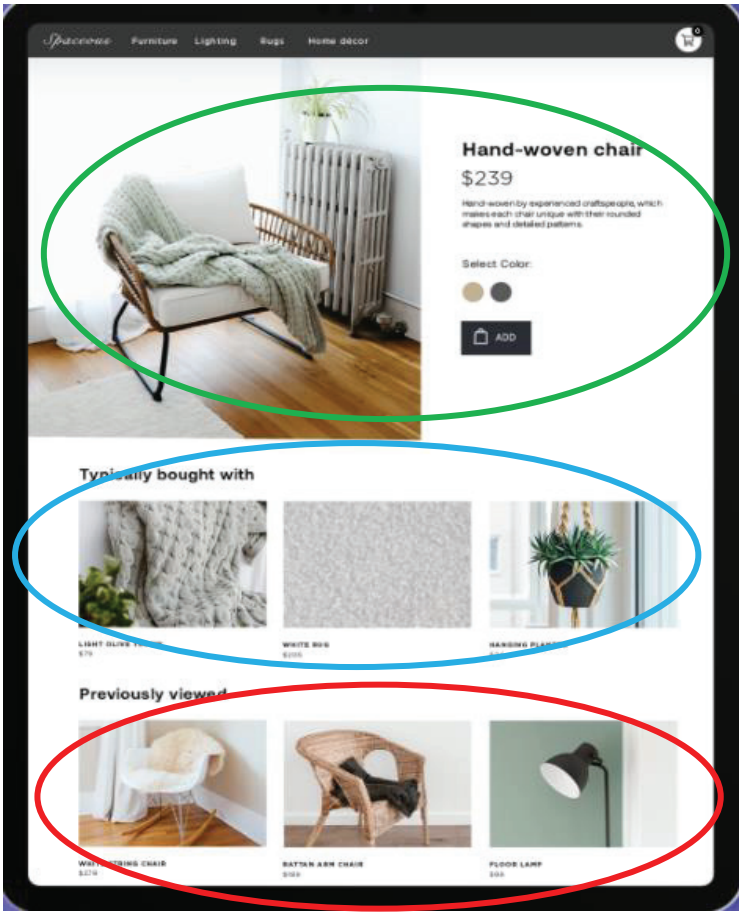
Strojno učenje primjenjuje se kod rješavanja kompleksnih zadataka ili problema koji uključuju veliku količinu podataka i puno varijabli, a za njihovo rješavanje ne postoje jednoznačne formule ili jednadžbe, odnosno modeli. [1]

Neka područja primjene strojnog učenja:

(1) Prepoznavanje slike- identifikacija objekata, osoba, mjesta, automatski prijedlog označavanja prijatelja [9]
(2) Prepoznavanje govora- glasovno pretraživanje (Google Assistant, Siri, Alexa..) [9]
(3) Predviđanje prometa- lokacija vozila u realnom vremenu, prosječno predviđanje vremena putovanja, navigacija u prometu (Google karte) [9]
(4) Preporuka proizvoda- predlaganje proizvoda prema interesu korisnika [9]
(5) Samovozeći automobili- strojno učenje bez nadzora omogućuje automobilu detekciju ljudi i objekata tijekom vožnje [9]
(6) Računalne financije- procjena kreditne sposobnosti [8]
(7) Medicina- detekcija tumora i bolesti povezanih s mozgom [8]
(8) Industrija- procjena i nadziranje kvalitete proizvoda [8]



Slika 2.3.1. Primjeri primjene strojnog učenja.
Lijevo- logo Apple-ove aplikacije Siri za prepoznavanje govora. [11]
Desno- logo Google Maps aplikacije za predviđanje i navigaciju u prometu. [12]



Kao što je već i poznato, internet je vodeći komunikacijski medij današnjice. [13] Dakle, iz dana u dan, svijet se sve više i više digitalizira. Baš iz tog razloga, kupnja proizvoda postala je također online „aktivnost“. Glavna značajka gotovo svake web stranice za e-trgovinu postala je preporuka proizvoda. Drugim riječima, uz pomoć strojnog učenja i umjetne inteligencije, web stranica prati ponašanje korisnika na temelju prethodnih pretraživanja, prethodnih kupnji i povijesti košarica, a zatim daje preporuke za proizvode. Upravo to prikazuje *Slika 2.3.2.* [15]

Slika 2.3.2. Primjer primjene strojnog učenja. [14]

Dakle, *Slika 2.3.2.* prikazuje primjer primjene strojnog učenja, odnosno primjer preporuke proizvoda specificiran za određenog korisnika. Zelenom bojom označen je proizvod koji kupac trenutno gleda. Crvenom bojom, označeni su proizvodi koje je korisnik ranije pretraživao. Kako bi korisnik maksimalno povećao svoje iskustvo u kupovini, preporučeni su proizvodi koji se obično kupuju sa proizvodima koje je korisnik pretraživao (označeno plavo). [14]

3. VRSTE STROJNOG UČENJA

Četiri osnovna tipa strojnog učenja jesu: nadzirano učenje, polunadzirano učenje, nenadzirano učenje te podržano učenje. [16]

3.1. NADZIRANO UČENJE

Nadzirano učenje (*eng. supervised learning*) je tip strojnog učenja koje se temelji, kao što mu i samo ime kaže, na nadzoru. Drugim riječima, u ovom tipu učenja stroj se podučava primjerom. Operater daje algoritmu strojnog učenja poznati skup ulaznih i izlaznih podataka, a algoritam mora naučiti kako od danih ulaza doći do danih izlaza. Dok operater zna točne odgovore na problem, algoritam identificira obrasce, uči iz opažanja te predviđa rezultate. Algoritam zatim daje svoja predviđanja i operater ga, ukoliko nije zadovoljan s danim predviđanjima, ispravlja i proces se nastavlja sve dok algoritam ne postigne visoku razinu preciznosti. [16] Dakle, u ovom načinu učenja algoritmi rade s poznatim skupovima ulaznih i izlaznih podataka i uvježbavaju model za predviđanje izlaznih podataka. [8] Glavni cilj tehnike nadziranog učenja bio bi mapiranje ulazne varijable (x) s izlaznom varijablom (y). [5]

Nadzirno učenje dalje se dijeli na klasifikaciju i regresiju. Klasifikacija se temelji na predviđanju diskretnih odnosno kategoričkih izlaznih podataka, primjerice je li e-pošta originalna ili neželjena. Prilikom klasifikacije ulazni se podatci razvrstavaju u kategorije. [8] Glavni cilj klasifikacije je identificirati odnosno predvidjeti u koju će klasu/kategoriju novi podaci spadati. Neki od algoritama klasifikacije jesu: algoritam slučajne šume, algoritam stabla odlučivanja, algoritam logističke regresije. [5]

Regresijska analiza u strojnom učenju jest metoda koja služi za modeliranje odnosa između zavisne i nezavisne varijable. Drugim riječima, regresijska analiza prikazuje kako se vrijednost zavisne varijable mijenja u skladu s nezavisnom varijablom. Regresija, odnosno regresijski algoritmi se koriste za predviđanje kontinuiranih izlaznih podataka kao što je vremenska prognoza, plaća, cijena i drugo. Ona također uključuje iscrtavanje linije to jest krivulje koja se najbolje uklapa u podatkovne točke. Udaljenost između linija i svake točke minimalizirana je kako bi se postigla što bolja točnost. [18] Neki od algoritama regresije jesu: jednostavni algoritam linearne regresije, algoritam stabla odlučivanja te algoritam multivarijatne regresije. [5]

3.2. NENADZIRANO UČENJE

Nenadzirano učenje (*eng. unsupervised learning*) je tip strojnog učenja u kojem nije potreban nadzor operatera. U ovom učenju ne postoji operater koji daje upute. Umjesto toga, stroj analizira postojeće dostupne podatke te na taj način utvrđuje njihove odnose i podudarnost. [16] Drugim riječima, stroj analizira skupove ulaznih podataka, bez dobivenih izlaznih podataka, pronalazi uzorke, izvodi zaključke te grupira dobivene podatke u manje klastere. [1] Glavni je cilj nenadziranog učenja grupirati skup podataka prema sličnostima, razlikama i raznim obrascima kako bi se postigla njihova što bolja organizacija. [5]

Najpoznatija tehnika učenja bez nadzora jest grupiranje odnosno grupiranje (*eng. clustering*). Grupiranje uključuje razvrstavanje skupova sličnih podataka u grupe (*eng. clusters*) na temelju parametara sličnosti ili razlika između objekata, odnosno na temelju definiranih kriterija. [16]

3.3. POLUNADZIRANO UČENJE

Kao što mu i samo ime kaže, polunadzirano učenje (*eng. semi-supervised learning*) se nalazi između nadziranog i nenadziranog učenja. Ono ima karakteristike i jednog i drugog učenja, odnosno koristi kombinaciju i označenih i neoznačenih skupova podataka tijekom razdoblja obuke. Koristeći obje vrste skupova podataka, ovakav tip učenja prevladava prepreke odnosno nedostatke gore navedenih učenja. Glavni cilj ovakvog učenja bio bi korištenje svih dostupnih podataka, a ne samo označenih kao kod nadziranog učenja. U početku se slični podaci grupiraju zajedno s algoritmom učenja bez nadzora, a zatim pomažu u označavanju neoznačenih podataka u označene podatke. Razlog tome je što su označeni podaci skuplji od neoznačenih. [5]

3.4. PODRŽANO UČENJE

Ovaj tip učenja temelji se na povratnim informacijama. Algoritam strojnog učenja istražuje svoje okruženje, poduzima radnje, uči iz iskustva i poboljšava svoje izvedbe. [5] Podržano učenje (*eng. reinforcement learning*) funkcionira s jasnim ciljem i propisanim skupom pravila za postizanje tog cilja. Definiranjem pravila, algoritam strojnog učenja pokušava istražiti različite opcije i mogućnosti, određujući rezultate dokle god ne nađe onaj optimalan. [16]

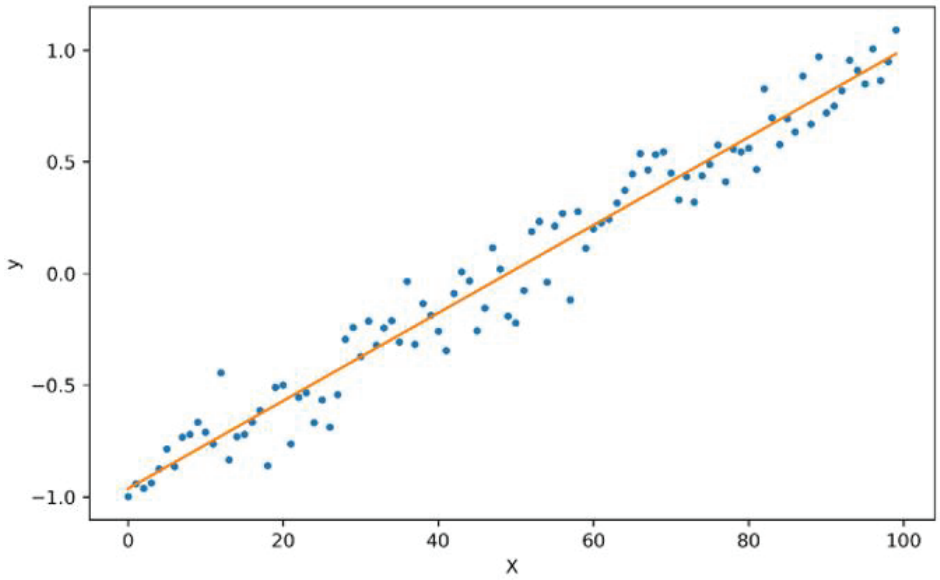
Ovakvi algoritmi su programirani za traženje pozitivnih nagrada koje dobiva kad izvrši radnju koja je korisna za postizanje krajnjeg cilja i za izbjegavanje kazni koje prima kad izvrši radnju koja ga udaljava od krajnjeg cilja. Podržano učenje često se koristi u području robotike i videoigara. [4]

4. OSNOVNI ALGORITMI STROJNOG UČENJA

Algoritmi strojnog učenja omogućuju računalima da uče i daju predviđanja na temelju podataka. Umjesto da računalu damo izričitu naredbu što da radi, primjenom ovih algoritama dajemo mu mogućnost da proučava veliku količinu podataka i puštamo ga da samo otkrije obrasce i odnose između njih. [19] Algoritmi strojnog učenja se dijele na algoritme nadziranog učenja, algoritme za nenadzirano učenje te algoritme podržanog učenja. Algoritmi nadziranog učenja sastoje se od skupa ulaznih i izlaznih varijabli. Koristeći ove skupove varijabli, algoritam predviđa izlazne vrijednosti i to sve do postizanja željene preciznosti. U algoritmu nenadziranog učenja nisu poznate izlazne varijable za predviđanje. Ovi algoritmi se koriste za grupiranje varijabli u različite skupine. Pomoću algoritama podržanog učenja, stroj je osposobljen za donošenje specifičnih odluka, gdje se on kontinuirano uvježbava koristeći pokušaje i pogreške ka postizanju krajnjeg cilja. [5]

4.1. ALGORITAM JEDNOSTAVNE LINEARNE REGRESIJE

Linearna regresija (*eng. linear regression*) najjednostavnija je vrsta regresije. Ona spada u algoritme nadziranog strojnog učenja za predviđanje vrijednosti koje se nalaze unutar kontinuiranog raspona. [20] Linearnom regresijom uspostavlja se odnos između ulazne i izlazne, odnosno nezavisne i zavisne varijable koji se može prikazati ravnom linijom. Drugim riječima, linearna regresija uzima skup točaka s ulaznim i izlaznim varijablama i pronalazi liniju, odnosno pravac koji najbolje odgovara tim točkama. Ova linija poznata je pod nazivom „regresijska linija“ i prikazana je na grafičkom prikazu na *Slika 4.1.1.* [19]



Slika 4.1.1. Grafički prikaz linearne regresije. Prikaz regresijske linije. [21]

Slika 4.1.1. prikazuje skup podatkovnih točaka, označenih plavom bojom. Na temelju zadanih podatkovnih točaka nacrtana je linija koja najbolje definira odnosno opisuje točke. Iz prikazanih podatkovnih točaka moguće je uočiti da su X i Y u linearnoj ovisnosti. Drugim riječima, povećanjem jedne varijable, primjerice X, povećava se i druga, Y varijabla. Linija koja najbolje opisuje dane točke jest regresijska i ona je na grafičkom prikazu iscrtana narančasto. [21]

Regresijsku linija se može modelirati na temelju linearne jednadžbe prikazane u nastavku:

$$y = ax + b \qquad (1)$$

U ovoj jednadžbi: *y*-zavisna (izlazna) varijabla

x-nezavisna (ulazna) varijabla

a,b- koeficijenti [22]

Koeficijenti *a* i *b* izvedeni su minimiziranjem zbroja kvadrata udaljenosti između podatkovnih točaka i regresijske linije. [22] Također, koeficijent *a* u jednadžbi (1) predstavlja nagib pravca, dok koeficijent *b* odsječak na y-osi.

Dakle, u linearnoj regresiji prisutan je skup ulaznih (X) i izlaznih (Y) podataka. Ti prisutni podaci koriste se za učenje funkcije, odnosno algoritma, kako bi se na temelju neke nepoznate zadane ulazne varijable x, mogla predvidjeti to jest pretpostaviti izlazna varijabla y. [20]

4.2. ALGORITAM VIŠESTRUKE LINEARNE REGRESIJE

Linearna regresija se uglavnom dijeli na dva tipa: jednostavna linearna regresija, opisana u podnaslovu prije te višestruka linearna regresija. Kao što je i rečeno, jednostavnu linearnu regresiju karakterizira jedna nezavisna varijabla. Višestruku linearnu regresiju, kako joj i samo ime sugerira, karakterizira više od jedne nezavisne varijable. [23]

Dakle, višestrukom regresijskom analizom može se kontrolirati nekoliko nezavisnih varijabli koje istovremeno utječu na zavisnu varijablu. Kao i linearna regresija, ona spada u nadzirani algoritam strojnog učenja. Također, kao i linearnom regresijom, ovom se metodom analizira odnos između zavisnih i nezavisnih varijabli. [23]

Jednadžba kojom se može opisati višestruka linearna regresija slična je onoj kojom se opisuje jednostavna linearna regresija (1), osim što ona uzima u obzir više od jedne nezavisne varijable.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_3 + \cdots + \beta_n x_n \qquad (2)$$

Pod (2) prikazana je jednadžba kojom se može opisati algoritam višestruke linearne regresije.

U prikazanoj jednadžbi: y- zavisna (izlazna) varijabla

x- nezavisna (ulazna) varijabla

β_0- regresijski koeficijent, odsječak na y-osi

$\beta_1, \beta_2, \ldots, \beta_n$ - regresijski koeficijent koji predstavlja promjenu u y kao rezultat pojedinačnih promjena u x [23]

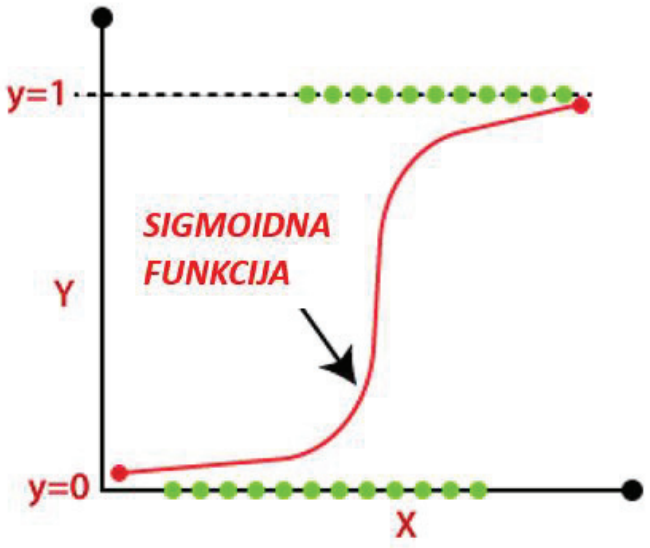
Regresijski koeficijenti $\beta_1, \beta_2, \ldots, \beta_n$ govore koliko svaka nezavisna varijabla doprinosi zavisnoj varijabli.

Višestruka linearna regresija je model koji omogućuje uzimanje u obzir svih potencijalno značajnih varijabli u jednom modelu. Upravo to bi bila jedna prednost ovog modela. Dakle, ovakva analiza uključuje točniji i detaljniji prikaz odnosa između svakog pojedinog čimbenika i ishoda. Još jedna od prednosti algoritma višestruke linearne regresije jest sposobnost objašnjenja nelinearnih odnosa i interakcija između varijabli. [23]

4.3. ALGORITAM LOGISTIČKE REGRESIJE

Jedan od najpopularnijih algoritama strojnog učenja bio bi algoritam logističke regresije. Ovaj algoritam spada u algoritme nadziranog učenja. [24] Logistička regresija (*eng. logistic regression*) zapravo nije regresija, već algoritam učenja klasifikacije. Naziv regresija dolazi zbog činjenice da je matematička formulacija logističke regresije slična onoj linearne regresije. Logistička se regresija koristi za zadatke binarne klasifikacije gdje je ishod binaran poput da ili ne, točno ili netočno, binarne vrijednosti 0/1 i slično. [25] Cilj ovog algoritma jest predvidjeti vjerojatnost da neka varijabla pripada određenoj klasi. Obično se koristi kad želimo odrediti pripada li unos jednoj ili drugoj klasi, kao što je odluka o tome je li prikazana slika auto ili ne. Međutim, u praksi se najčešće izlaz grupira u dvije kategorije; pripada primarnoj klasi ili ne pripada primarnoj klasi. Da bi se to postiglo, logistička regresija zapravo stvara prag ili granicu za binarnu klasifikaciju. Na primjer, svaka izlazna vrijednost između 0 i 0,49 može se klasificirati kao jedna skupina, a svaka izlazna vrijednost između 0,50 i 1,00 kao druga skupina [19] Drugim riječima, ako izlazna vrijednost za neki određeni ulaz bude bliža 0, dodjeljuje joj se negativna oznaka, a ako njena vrijednost bude bliža 1 dodjeljuje joj se pozitivna oznaka. [25]

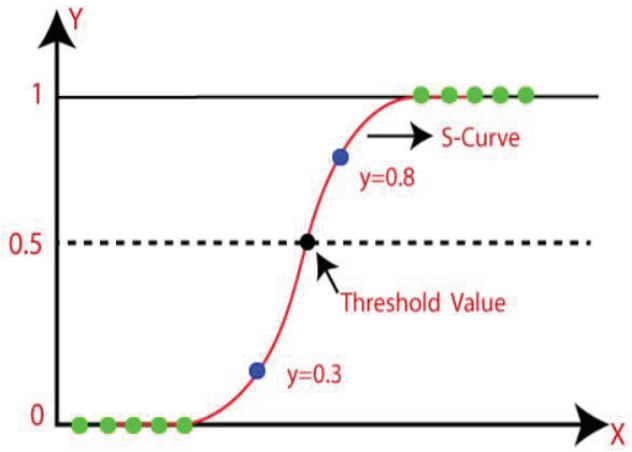
Kao i kod linearne regresije, i logistička regresija ima svoju karakterističnu krivulju. Umjesto korištenja regresijske linije, u logističkoj regresiji koristi se logistička funkcija u obliku slova „S“ koja se naziva sigmoidna funkcija. *(Slika 4.3.1.)* [24]



Slika 4.3.1. Grafički prikaz logističke regresije. Prikaz sigmoidne funkcije („S“ krivulje). [27]

Ova se funkcija koristi za preslikavanje predviđenih vrijednosti u vjerojatnosti. Preslikava bilo koju stvarnu vrijednost u drugu vrijednost unutar intervala <0,1>. Dakle, vrijednost logističke regresije mora biti između 0 i 1, ne smije prijeći tu granicu, tako da tvori „S“ krivulju. [24]

Kao što je već i spomenuto, da bi se izlaz mogao grupirati u dvije kategorije, potrebno je definirati prag ili granicu za binarnu klasifikaciju. [19] Prag, odnosno granica za binarnu klasifikaciju prikazan je na *Slika 4.3.2.* [26]



Slika 4.3.2. Prikaz praga odnosno granične vrijednosti (eng. threshold value) za binarnu klasifikaciju. [26]

Osim granične vrijednosti za binarnu klasifikaciju, na *Slika 4.3.2.* mogu se uočiti i dvije izlazne vrijednosti Y, označene plavom bojom. Ona vrijednost bliža 0 (dolje), odnosno manja od praga, poprima negativnu oznaku, dok ona bliža 1 (gore), odnosno veća ili jednaka pragu, poprima pozitivnu oznaku. [26]

Primjer sigmoidne funkcije, prikazane na *Slika 4.3.1.*, jest logistička funkcija dana jednadžbom:

$$f(x) = \frac{1}{1+e^{-x}} \qquad (3)$$

gdje je e- baza prirodnog logaritma

x- nezavisna (ulazna) varijabla

f(x)- y, zavisna (izlazna) varijabla [25]

Osnovna razlika između linearne i logističke regresije je u tome što je izlazna varijabla linearne regresije kontinuirana vrijednost, dok je izlazna varijabla logističke regresije kategorička varijabla. [19]

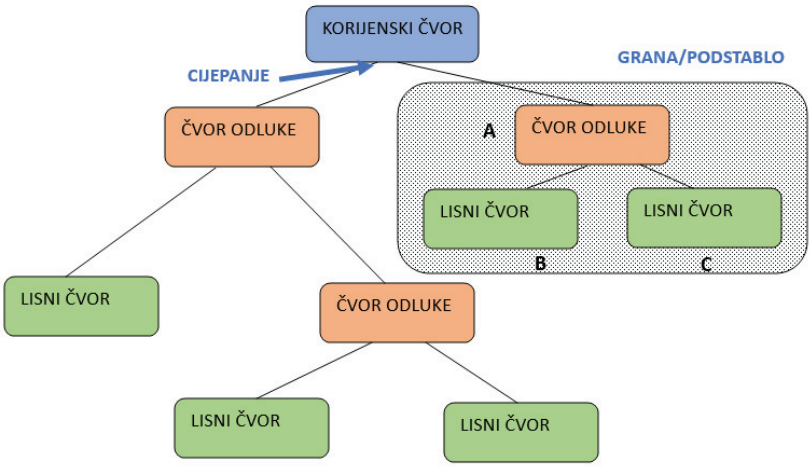
4.4. STABLO ODLUKE

Stablo odluke (*eng. decision tree*) nadzirani je tip strojnog učenja koje se može koristiti i za probleme klasifikacije i probleme regresije, iako se preferira za rješavanje problema klasifikacije. [28]

Ako se zamisli normalno stablo, poznato je da ono osim debla uključuje grane, lišće i korijenje. Slična takva struktura se prati i u stablu odluke. Dakle, stablo odluke sastoji se od korijenskog čvora (*eng. root node)*, čvora grananja ili čvora odluke *(eng. branching node/ decision node)* i lisnih čvorova *(eng. leaf node)*. [29] Unutarnji korijenski čvor je mjesto gdje započinje stablo odluke i on predstavlja cjelokupni skup podataka ili uzorak koji se dalje dijeli na više skupova. [30] Svaka grana se naziva još i podstablo i ona predstavlja čvor odluke. [28] Svaki lisni čvor predstavlja ishod odnosno konačni izlazni čvor, nakon kojeg se stablo ne može dalje odvajati. [30]

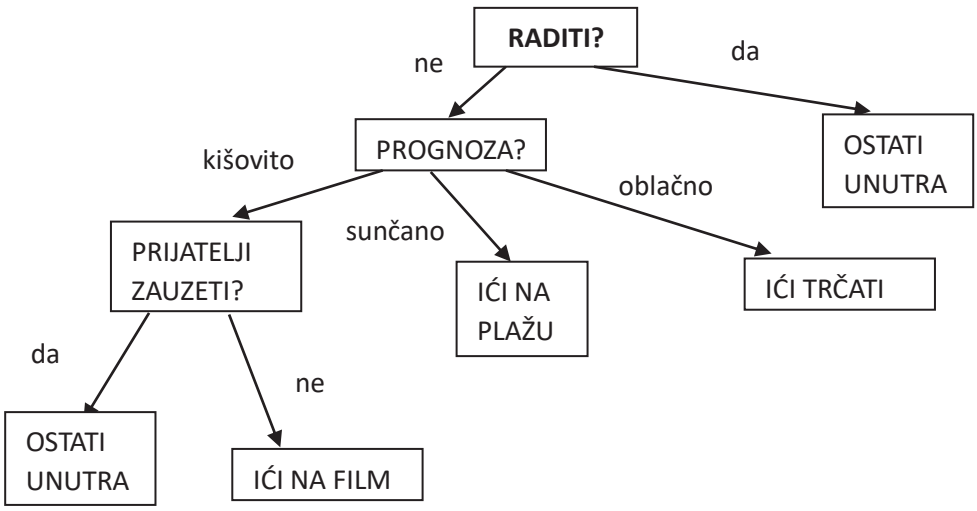
Stablo odlučivanja može se promatrati kao aciklički graf koji se može koristiti za donošenje odluka. [25] Dakle, ovaj bi algoritam zapravo bio grafički prikaz za dobivanje svih mogućih rješenja problema na temelju zadanih uvjeta. [30] Proces donošenja odluke ovim algoritmom sličan je ljudskom procesu odlučivanja, stoga ga je lako razumjeti. Ono se može koristiti za bilo koju situaciju bilo da se radi o diskretnim ili kontinuiranim podacima. [29]

Cijeli proces donošenja odluka algoritmom stabla odluke može se opisati u nekoliko koraka. Dakle, stablo započinje s korijenskim čvorom, koji sadrži kompletan skup podataka. Tu se postavlja određeno pitanje o podacima, odnosno ispituje se određena značajka tih podataka. Na temelju odgovora podaci se usmjeravaju na različite grane do sljedećih čvorova, koji postavljaju dodatna pitanja i vode podatke do sljedećih grana to jest čvorova. [19] Dakle, stablo odluke jednostavno postavlja pitanje i na temelju odgovora (Da/Ne) dalje dijeli stablo na podstabla odnosno grane. [30] Naravno, ako početni skup podataka ima više od jedne značajke, onda će se početni korijenski čvor razdijeliti na onoliko grana koliko ima i značajki. [31] Ovaj proces nastavlja se sve dok podaci ne dođu do krajnjeg lisnog čvora, nakon kojeg se stablo ne može dalje odvajati. [19]



Slika 4.4.1. Prikaz opće strukture stabla odluke. [28]

Prikaz opće strukture stabla odluke predstavljen je na *Slika 4.4.1.* Dakle, plavom bojom naznačen je korijenski čvor. Nadalje slijedi njegovo dijeljenje odnosno cijepanje *(eng. splitting)* u dva čvora grananja to jest čvora odluke, označenih narančastom bojom. Zatim se čvorovi odluke cijepaju na lisne čvorove, označene zelenom bojom. Jedna grana ili podstablo koje predstavlja pododsjek cijelog stabla naznačeno je u točkastom pravokutniku. Slovima A, B i C naznačeni su redom roditeljski čvor *(eng. parent node)* te dva podčvora nazvanih dijete čvor *(eng child node)*. Roditeljski čvor zapravo predstavlja čvor koji je podijeljen na podčvorove. [28]



Slika 4.4.2. Primjer stabla odlučivanja o tome što učiniti pri različitim vremenskim prognozama. [29]

Slika 4.4.2. prikazuje jednostavan primjer stabla odlučivanja, odnosno primjer na kojem je lagano razumjeti proces donošenja odluke ovim algoritmom. Uočljivo je da se na samom

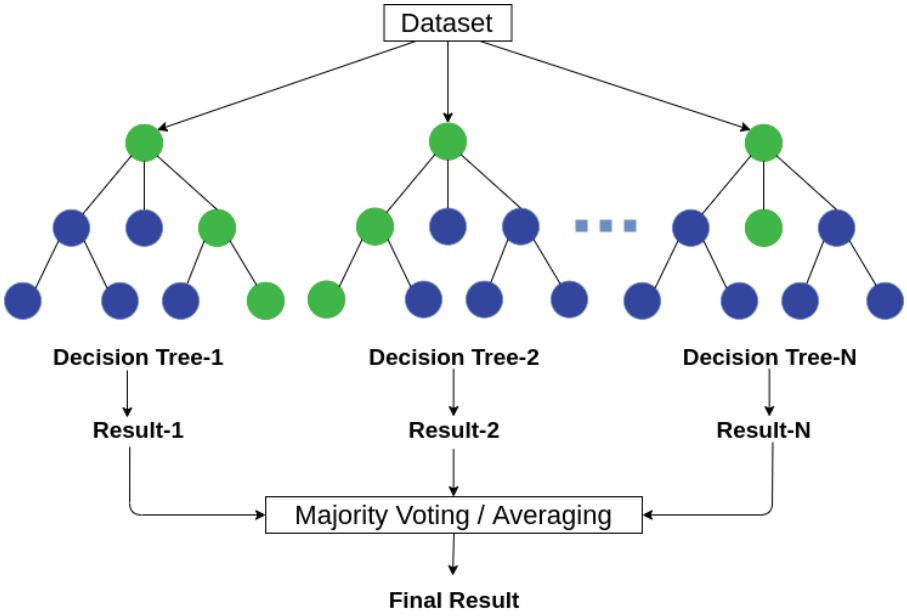
početku stabla odluke korijenski čvor cijepao samo u dva čvora grananja. Time se da zaključiti da je na početku algoritam ispitivao dvije značajke, odnosno da ovo zaista je jednostavan primjer. [29]

Obično problem sadrži velik skup podataka, a samim tim i velik skup značajki. Velik broj značajki automatski rezultira i velikim brojem podjela odnosno cijepanja, što zauzvrat daje veliko i kompleksno stablo. Takva su stabla složen. Iz tog razloga ponekad je potrebno prestati „uzgajati drvo“. Jedan od načina da se smanji kompleksnost stabla jest postaviti minimalan broj ulaza za obuku koji će se koristiti na svakom listu. Primjerice, za donošenje odluke „umro ili preživio“ moguće je ako se upotrijebi najmanje deset putnika. Primjenom postavljana minimalnog broja ulaza svaki list koji ima manje od deset putnika će se zanemariti. Drugi način smanjivanja kompleksnosti stabla jest postavljanje maksimalne dubine modela, odnosno postavljanje najdužeg puta od korijena do lista. Također, jedna od poznatijih metoda povećanje učinkovitosti stabla bilo bi orezivanje *(eng. pruning)*. Orezivanje uključuje uklanjanje grana koje koriste značajke niske važnosti. Na taj način se smanjuje složenost stabla. [31]

4.5. ALGORITAM SLUČAJNE ŠUME

Sljedeći, ali ne manje važan algoritam u strojnom učenju jest algoritam slučajne šume *(eng. random forest)*. Slučajna šuma algoritam je nadziranog strojnog učenja koji se može primijeniti na probleme klasifikacije i regresije. [32]

Kao što mu i samo ime kaže ovaj se algoritam sastoji od skupa stabala. Naravno u ovom slučaju riječ je o stablima odluke. Dakle, algoritam slučajne šume je algoritam koji sadrži nekoliko stabala odluke na različitim podskupovima ili uzorcima određenog promatranog skupa podataka. Sva se ta stabla treniraju korištenjem tih različitih uzoraka iz skupa podataka za obuku. Ova metoda uzorkovanja naziva se *eng. bagging (*kao skraćeno od *eng. bootstrap aggregating)*. Jednom obučena, slučajna šuma uzima iste podatke i unosi ih u svako stablo odlučivanja. Svako stablo daje svoje predviđanje, a slučajna šuma evidentira rezultate. Dakle, umjesto da se oslanja na jedno stablo odluke, slučajna šuma kombinira predviđanja iz više stabala odluke kako bi napravila točnija predviđanja ishoda. Kombiniranjem izlaza svih stabla odluke, algoritam slučajne šume daje konačan rezultat. Naravno, što je veći broj stabala u algoritmu to je njegova točnost i sposobnost rješavanja problema bolja. Najčešća predviđanja među svim stablima odabiru se kao konačno rješenje za skup podataka. [19] [32]



Slika 4.5.1. Shematski prikaz slučajne šume. [33]

Prikaz na *Slika 4.5.1.* predstavlja shematski prikaz algoritma slučajne šume. Prikaz započinje od skupa podataka (*eng. data set*). Promatrani skup podataka podijeljen je na podskupove sa svojim stablima odluke (*eng. decision tree*-1,2,3). Svako stablo predviđa svoje rezultate, a slučajna šuma ih evidentira (*eng. result*-1,2,3). Iza predviđanja rezultata slijedi provjera većinskog glasovanja ili određivanje prosjeka, nakon čega algoritam daje konačni rezultat.

Najjednostavnije objašnjenje rada ovog algoritma može se prikazati na sljedeći način. Dakle, na početku se odabere određeni skup podataka. Iz tog skupa podataka odabiru se nasumični uzorci. Ovaj algoritam će konstruirati stablo odluke za svaki uzorak podataka. Svako stablo odluke dat će konačni rezultat. Prosjek svih tih rezultata, koji bude imao najviše „glasova“ bit će konačni rezultat predviđanja. [34] Ako se pak želi klasificirati novi objekt na temelju neke značajke, svako stablo daje klasifikaciju i kaže se da ono „glasa“. Šuma odabire klasifikaciju koja ima najviše glasova, a u slučaju regresije uzima prosjek izlaza po različitim stablima. [35]

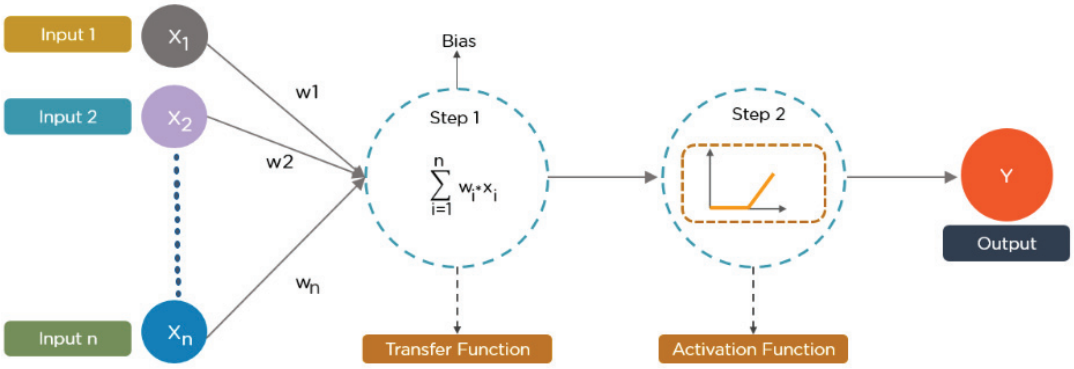
4.6. NEURONSKA MREŽA

Neuronske mreže *(eng. neural networks)* podskup su algoritama strojnog učenja. One oponašaju osnovno funkcioniranje ljudskog mozga i inspirirane su načinom na koji ljudski mozak tumači informacije odnosno način na koji biološki neuroni šalju signale jedni drugima. [36]

Dakle, neuronske se mreže mogu opisati kao skup algoritama, koji su oblikovani prema ljudskom mozgu, a služe za grupiranje i klasificiranje uzoraka odnosno podataka velikom brzinom. Uzorci koje ona prepoznaje su numerički. U taj se oblik moraju prevesti svi podaci iz stvarnog svijeta, bilo da se radi o slikama, zvuku, tekstu ili drugom. [36][37]

Umjetne neuronske mreže *(eng. artificial neural networks, ANNs)* sastoje se od umjetnih neurona odnosno čvorova. [36] Čvor bi zapravo bio mjesto gdje se događa računanje koje se aktivira kad naiđe na dovoljno podražaja. [37] Umjetni neuron može se zamisliti kao jednostavan ili višestruki linearni regresijski model s aktivacijskom funkcijom na samom kraju. Čvorovi su raspoređeni u slojeve. Svaka osnovna neuronska mreža sastoji se od barem tri sloja čvorova; ulaznog sloja *(eng. Input layer),* skrivenog sloja *(eng. Hidden layer),* i izlaznog sloja *(eng. Output layer).* [38]

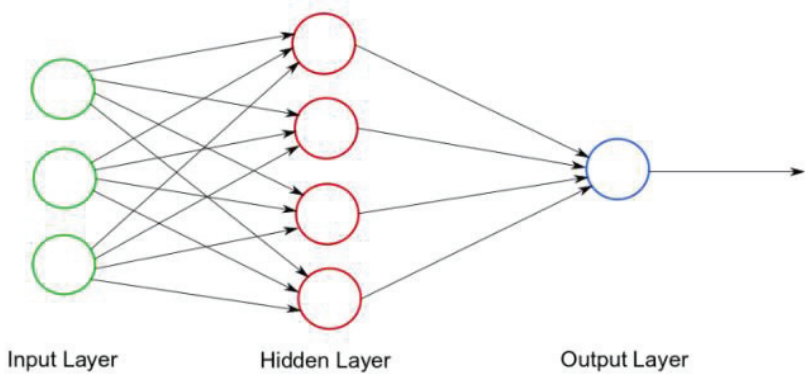
Svaki čvor ili umjetni neuron povezuje se s drugim čvorovima i ima pridruženu težinu *(eng. weights)*, koja sadrži informacije o ulaznom signalu [36] Svaka iteracija i novi unos uzorkuju ažuriranje ovih težina. Dakle, čvor kombinira ulaz iz podataka sa skupom težina, odnosno koeficijenata koji ili pojačavaju ili prigušuju taj unos. Nakon unosa svih podataka iz skupa podataka za obuku, produkti ulaznih podataka i težina se zbrajaju. Taj zbroj se dalje pušta kroz aktivacijsku funkciju čvora. Ovaj se dio odvija kako bi se uvidjelo treba li i u kojoj mjeri taj signal napredovati dalje kroz mrežu kako bi utjecao na krajnji ishod. Može se zaključiti da ove težine pomažu u određivanju važnosti bilo koje ulazne varijable, pri čemu one veće, odnosno značajnije pridonose rezultatu, dok one manje ne. Ako taj signal prođe, čvor je aktiviran. Aktivirani čvor je u mogućnosti slati podatke sljedećem sloju mreže, inače se podaci iz tog čvora ne prosljeđuju na sljedeći sloj mreže. [37] [38] Cijeli tok jednog neurona odnosno čvora prikazan je na *Slika 4.6.1.*



Slika 4.6.1. Shematski prikaz toka jednog neurona. [39]

Dakle, shematski prikaz na *Slika 4.6.1.* prati jedan neuron od ulaza do izlaza. Podaci (*eng. Input 1,2,n)* svakom neuronu pružaju informacije u obliku ulaza (x_1, x_2, x_n). Također svakom su neuronu pridružene težine (W_1, W_2, W_n). U prvom koraku *(eng. step 1)* čvor množi ulaze s nasumičnim težinama i zbraja ih. Nadalje, taj se zbroj pušta kroz aktivacijsku funkciju *(eng. Step 2)* kako bi se odredilo koji neuron aktivirati, odnosno kako bi čvor dao ishod to jest izlaznu vrijednost *(eng. output).* [39]

Rad neuronskih mreža može se opisati na sljedeći način. Dakle, prvi neuron iz prvog skrivenog sloja povezan je sa svim ulazima iz prethodnog sloja. Slično tome, drugi je neuron iz prvog skrivenog sloja također povezan sa svim ulazima iz prethodnog sloja i tako dalje za sve neurone u prvom skrivenom sloju. Neuroni u drugom skrivenom sloju smatraju izlaze prethodno skrivenog sloja ulazima to jest neuroni drugog skrivenog sloja također su povezani s neuronima prethodnog sloja. Cijeli ovaj proces naziva se širenje naprijed *(eng. forward propagation)*. Svi ovi neuroni prolazit će tok prikazan na *Slika 4.6.1.* Dakle, neuron iz sloja *i* uzet će izlaz svih neurona iz sloja *i-1* kao ulaze, uzeti u obzir težine, izračunati zbroj. Zbroj se dalje šalje funkciji aktivacije kako bi čvor dao izlaznu vrijednost. [39]

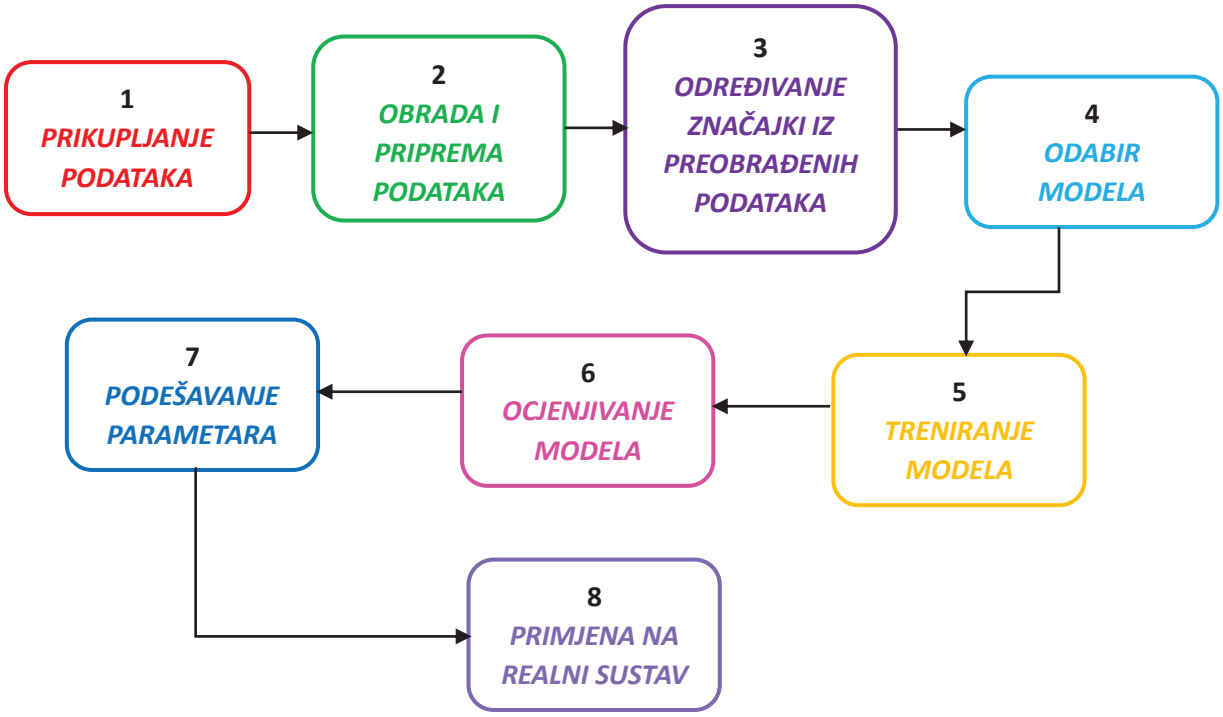


Slika 4.6.2. Shematski prikaz neuronske mreže. [39]

Slika 4.6.2. prikazuje primjer jednostavne neuronske mreže sastavljene od tri sloja; ulaznog, jednog skrivenog i izlaznog sloja. Dakako, u mrežama dubokog učenja, mreže nisu jednostavne. Duboko učenje definirano je neuronskim mrežama s više od tri sloja, uključujući ulazni i izlazni sloj. Dakle, takve mreže imaju ulazni sloj, više od jednog skrivenog sloja te izlazni sloj. Svaki sloj čvorova trenira se na posebnom skupu značajki. Što se dalje napreduje kroz neuronsku mrežu, to su značajke koje čvorovi mogu prepoznati složenije. Zapravo, svaki novi sloj grupira i rekombinira značajke prethodnog sloja. Ovaj postupak je poznat pod nazivom hijerarhija značajki *(eng. feature hierarchy)*. Ova bi hijerarhija bila hijerarhija veće složenosti i kompleksnosti. Ona neuronske mreže dubokog učenja čini sposobnima za rukovanje s velikim skupom podataka. [37]

Jedna od najvećih prednosti neuronskih mreža bila bi mogućnost otkrivanje skrivene strukture unutar skupa neoznačenih, nestrukturiranih podataka. Zapravo je velika većina podataka u svijetu takva, nestrukturirana. Drugi naziv za takve podatke bio bi „sirovi medij“. U takve podatke spadaju slike, tekstovi, video i audio zapisi. Drugim riječima, neuronske mreže imaju sposobnost uzeti milijun slika i grupirati ih prema sličnostima. Primjerice mačke u jednu mapu, pse u drugu, a leptire u treću. [37]

5. PROCES ANALIZE PODATAKA



Slika 5.1. Koraci pri razvoju modela strojnog učenja. [8]

Prikaz na *Slika 5.1.* opisuje korake pri razvoju modela strojnog učenja. Dakle, cijeli proces započinje prikupljanjem podataka. Strojevi uče iz ulaznih podataka. Od najveće je važnosti prikupiti pouzdane podatke kako bi model mogao pronaći ispravne uzorke ali i kako bi se postigli što bolji i točniji rezultati. Pod pouzdane podatke smatraju se podaci koji nisu zastarjeli, koji sadrže vrlo malo ponovljenih vrijednosti i vrlo malo vrijednosti koje nedostaju.

Nakon što se prikupe podaci, potrebno ih je pripremiti. Priprema podataka može se učiniti tako da se svi podaci spoje i zatim nasumično raspoređuju. Također, oni se mogu pripremiti na način da se pročiste radi uklanjanja neželjenih podataka, vrijednosti koje nedostaju te dupliciranih vrijednosti. Nakon pripreme podataka potrebno ih je podijeliti u dva skupa; skup za obuku i skup za testiranje. Skup za obuku bio bi skup podataka iz kojeg model uči, dok bi skup za testiranje bio skup za provjeru točnosti modela nakon obuke. Sljedeći važan korak bio bi određivanje značajki obrađenih podataka. Nakon određivanja značajki potrebno je odabrati model strojnog učenja. Model određuje izlaz koji se dobije nakon pokretanja algoritma strojnog učenja na prikupljenim podacima. Važno je odabrati model koji odgovara ispitivanom problemu. Nadalje, osim odabira modela potrebno je i obučiti odabrani model. Tijekom obuke obrađeni podaci se prosljeđuju odabranom modelu kako bi on pronašao obrazac i napravio predviđanja. Nakon uvježbavanja modela slijedi provjera odnosno ocjenjivanje modela. Ocjenjivanje modela zapravo bi bio postupak provjere kako i da li model dobro radi. To se testiranje provodi na prethodno pripremljenim podacima za testiranje. Sljedeći korak jest podešavanje parametara. U tom koraku provjerava se da li se točnost modela može na bilo koji način poboljšati. To se postiže podešavanjem parametara modela. Krajnji korak jest primjena na realni sustav, odnosno upotreba modela na podacima koji nisu otprije poznati kako bi napravili što točnija predviđanja. [40]

6. PROGRAMSKI JEZIK R

R je programski jezik pogodan za sve vrste izračuna, obradu podataka, znanstvena izračunavanja te grafički prikaz. [41]

6.1. OPĆENITO O PROGRAMSKOM JEZIKU R

R se može opisati kao integrirani softverski paket. Pruža širok izbor statističkih i grafičkih tehnika te je vrlo pogodan za proširivanje. Neke od statističkih tehnika koje sadrži jesu linearno i nelinearno modeliranje, klasični statistički testovi, klasifikacija i drugom.

Kao softverski paket uključuje veliku i sveobuhvatnu zbirku posrednih alata za analizu podataka, razne grafičke mogućnosti za analizu tih istih podataka te njihov prikaz na ekranu

ili papiru, skup operatora za izračune na nizovima, jednostavno rukovanje i skladištenje podataka, mogućnosti unosa i izlaza i drugo. [41]

Programski jezik R razvili su ga Ross Ihaka i Robert Gentleman na Sveučilištu u Aucklandu, Novi Zeland. [43] R je GNU projekt, odnosno GNU operacijski sustav. [41] GNU je operacijski sustav koji je slobodan softver to jest on poštuje slobodu korisnika. [42] Dostupan je kao besplatni softver, dakle svatko ga može instalirati bez kupnje licence. Radi na velikom broju platformi poput Windowsa, Linuxa, macOSa i drugih. [41]

Slika 6.1.1. Logo programskog jezika R. [44]

Glavni nedostatak ovog programskog jezika zapravo bi bio taj što neke naredbe mogu zauzeti veliku količinu memorije. Drugi važan nedostatak je brzina programskog jezika. Programski jezik R puno je sporiji od ostalih programskih jezika kao što su Python i MATLAB. [43]

7. ZRAK

Zemljin plinoviti zaštitni omotač naziva se atmosfera. Atmosfera je mješavina plinova i to kisika, dušika, vode, argona i ugljikovog dioksida. Dakle, atmosfera je ispunjena zrakom koji djeluje kao izolator, drugim riječima sprječava da Zemlja postane prevruća ili pak prehladna. Osim izolatora zrak u atmosferi ima ulogu i zaštite od meteora, koji se vrlo često spale u male komadiće prije nego dođu do Zemljine površine. [48]

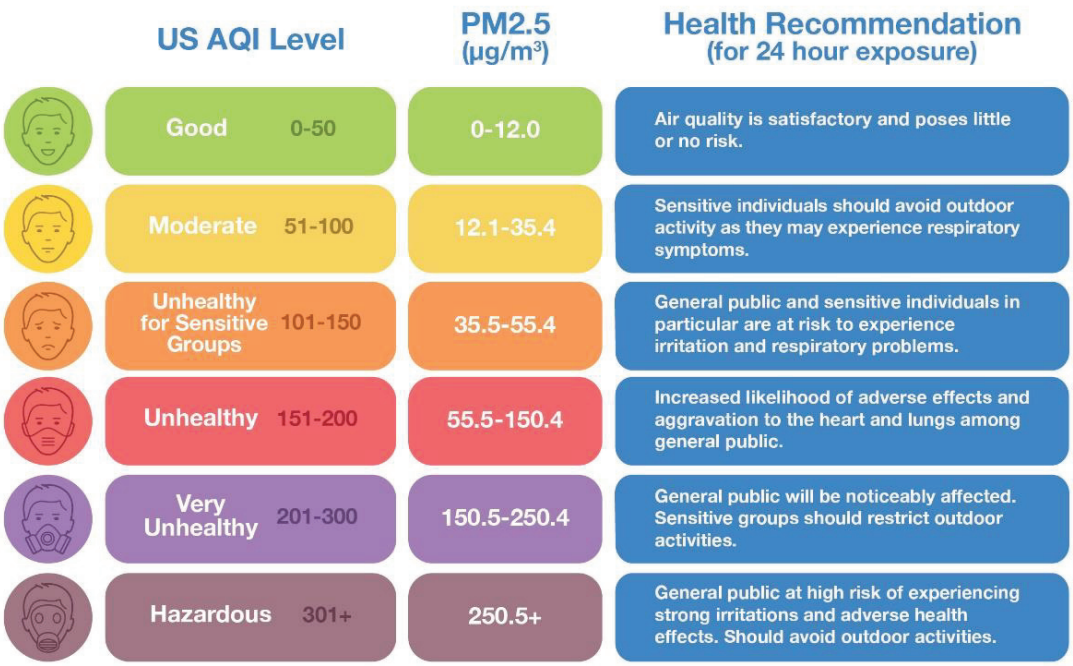
7.1. SVOJSTVA ZRAKA

Zrak je mješavina plinova. Sastoji se približno od 79% dušika i 21% kisika. Naravno, osim dušika i kisika, zrak još sadrži i argon, ugljikov dioksid, vodik, neon, metan, ozon i mnoge druge elemente. Zrak je vrlo važan za život svih živih bića. Ona ga koriste za disanje. Dakle, živa bića udišu kisik, a izdišu ugljikov dioksid. Upravo taj ugljikov dioksid biljke dalje, zajedno sa sunčevom svjetlošću, koriste za proizvodnju hrane u procesu fotosinteze. Zrak također može sadržavati čak 5% vodene pare. Sadržaj vode dakako varira ovisno o temperaturi zraka. Suhi zrak je gušći od vlažnog zraka. Vlažnost zraka obično se mjeri u postotcima, a ona je najviša neposredno prije kiše. [45][46]

Iako je zrak uglavnom plin, on također može sadržavati mnogo sitnih čestica. Te se čestice nazivaju aerosoli. Neki aerosoli jesu prašina ili pelud. Takve aerosole zrak „pokupi“ prirodnim putem kad puše vjetar. No međutim, zrak može sadržavati i čestice koje uzrokuju onečišćenja, poput čađe, dima, ispušnih plinova automobila i slično. Naravno, onečišćenje zraka razlikuje se ovisno o geografskom položaju, količini onečišćujućih tvari, vrsti onečišćujućih tvari i drugom. [45][46]

7.2. INDEKS KVALITETE ZRAKA

Onečišćenje zraka mjeri se indeksom kvalitete zraka *(eng. air quality indeks, AQI)*. Indeks kvalitete zraka jest mjerilo koje se kreće od 0 do 500. Dakle, što je njegova vrijednost viša, veća je razina onečišćenja zraka, a samim tim i veća zabrinutost za zdravlje. AQI je podijeljen u šest kategorija. Svaka kategorija predočava različitu razinu zabrinutosti za zdravlje. Također, kategorije su označene bojama kako bi ljudi lakše i brže uvidjeli kakva je kvaliteta zraka. [47]



Slika 7.2.1. Shematski prikaz indeksa kvalitete zraka (AQI) s odgovarajućim AQI kategorijama, PM2.5 mjerom i zdravstvenim preporukama. [47]

AQI koristi PM2.5 mjerenja kao odrednicu očitanja kvalitete zraka. PM2.5 se mjeri u mikrogramima po kubnom metru (µg/m^3). Osim PM2.5 mjera, AQI je podijeljen u već spomenutih šest kategorija. Prikaz na *Slika 7.2.1.* prikazuje da je 0-50 dobra kvaliteta zraka, dok se mjerenja iznad 300 smatraju opasnima. [47]

8. EKSPERIMENTALNI DIO

Cilj je ovog rada primijeniti algoritme odnosno metode strojnog učenja na neki realni primjer. Realni primjer koji će se obrađivati u ovom radu bit će onečišćenje zraka, odnosno određivanje to jest predviđanje indeksa kvalitete zraka. Drugim riječima, iz danog skupa podataka, primjenom strojnog učenja te programskog jezika R, pokušat će se predvidjeti indeks kvalitete zraka na temelju nekih osnovnih značajki koje utječu na samu kvalitetu zraka.

8.1. OPIS PODATAKA

Podaci korišteni u ovom radu preuzeti su sa online platforme pod imenom Kaggle. Kaggle je zajednica za podatkovnu znanost koja omogućuje korisnicima komunikaciju s drugim znanstvenicima te pronalaženje i objavljivanje skupova podataka. [52]

Preuzeti podaci sadrže indeks kvalitete zraka zajedno s njegovim ovisnim značajkama od 2013. do 2018. za grad Delhi u Indiji. [52]

Dakle, podaci sadrže izmjerene vrijednosti svih značajki koje utječu na indeks kvalitete zraka, uključujući i sam indeks. Značajke koje utječu na indeks jesu:

1. Prosječna temperatura *(eng. average temperature)* - mjerna jedinica: °C
2. Maksimalna temperatura *(eng. maximum temperature)* - mjerna jedinica: °C
3. Minimalna temperatura *(eng. minimum temperature)* - mjerna jedinica: °C
4. Atmosferski tlak na razini mora *(eng. atmospheric pressure at sea level)* - mjerna jedinica: hPa
5. Prosječna relativna vlažnost zraka *(eng. average relative humidity)* - mjerna jedinica: %
6. Prosječna vidljivost *(eng. average visibility)* - mjerna jedinica: Km
7. Prosječna brzina vjetra *(eng. average wind speed)* - mjerna jedinica: Km/h
8. Maksimalna stalna brzina vjetra *(eng. maximum sustained wind speed)* - mjerna jedinica: Km/h

Svih osam značajki predstavljaju ulazne varijable. Jedina izlazna varijabla bit će indeks kvalitete zraka, AQI, koji je izražen po PM2.5 mjeri u $\mu g/m^3$.

8.2. OBRADA I PRIPREMA PODATAKA

Sljedeći ključan korak u procesu obrade i pripreme skupa podataka bio bi pobliže se upoznati s njima. Primjerice, provjeriti ima li u danom skupu neželjenih podataka, vrijednosti koje nedostaju te dupliciranih vrijednosti, povjeriti kakva je ovisnost izlazne varijable o svakoj od ulaznih varijabli te navedeno vizualizirati.

Prvi je korak u ovom procesu instalirati programski jezik R. Programski jezik, točnije njegova inačica RStudio, preuzeta je sa web stranice čija je adresa navedena u prilogu ovog rada.

Nakon instaliranja programskog jezika, potrebno je instalirati pakete koji se koriste u izradi modela strojnog učenja. Paketi u R-u su skup funkcija te kompiliranih kodova pohranjenih u direktoriju pod nazivom „library“ unutar samog programskog jezika. Instaliranje paketa provedeno je pomoću naredbe *install.packages(„ “)*, gdje se unutar zagrade upisuje ime traženog paketa. Paketi koji su instalirani i korišteni za izradu modela primijenjenih u radu jesu:

caret- sadrži funkcije za usmjeravanje procesa obuke modela za probleme regresije i klasifikacije

ggplot2- paket potreban za vizualiziranje podataka

readxl- paket potreban za učitavanje podataka iz Excela u R

caTools- paket za provođenje statističke analize

Dodatan paket potreban za izradu modela višestruke linearne regresije jest *dplyr* - paket za manipulaciju podacima koji rješava najčešće prepreke manipulacije podacima.
Dodatan paket potreban za izradu modela slučajne šume jest *randomForest* – paket za stvaranje i analizu slučajnih šuma.
Dodatan paket potreban za izradu modela neuronske mreže jest *neuralnet*- paket za kreiranje, treniranje i testiranje neuronske mreže.

```
1  #INSTALIRANJE PAKETA
2
3  install.packages("caret")
4  install.packages("ggplot2")
5  install.packages("readxl")
6  install.packages("caTools")
7  install.packages("dplyr")
8  install.packages("randomForest")
9  install.packages("neuralnet")
```

Slika 8.2.1. Prikaz instaliranja paketa potrebnih za izradu modela strojnog učenja.

Instalirani programski paketi pokreću se, odnosno učitavaju iz direktorija pomoću naredbe *library()*, gdje se unutar zagrade upisuje ime paketa koji se želi pokrenuti.

```
11  #UČITAVANJE PAKETA-MODEL VIŠESTRUKA LINEARNA REGRESIJA
12  library(caret)
13  library(ggplot2)
14  library(readxl)
15  library(caTools)
16  library(dplyr)
```

Slika 8.2.2. Učitavanje paketa potrebnih za izradu modela višestruke linearne regresije.

```
11  #UČITAVANJE PAKETA-MODEL SLUČAJNA ŠUMA
12  library(caret)
13  library(ggplot2)
14  library(readxl)
15  library(caTools)
16  library(randomForest)
```

Slika 8.2.3. Učitavanje paketa potrebnih za izradu modela slučajne šume.

```
11  #UČITAVANJE PAKETA-MODEL NEURONSKA MREŽA
12  library(caret)
13  library(ggplot2)
14  library(readxl)
15  library(caTools)
16  library(neuralnet)
```

Slika 8.2.4. Učitavanje paketa potrebnih za izradu modela neuronske mreže.

Nakon učitavanja potrebnih paketa, također je potrebno učitati promatrani skup podataka iz aplikacije za proračunske tablice, Excel u programski jezik R. Učitavanje podataka izvodi se pomoću naredbe *read_excel(„“)*, gdje se unutar navodnih znakova upisuje datoteka na računalu u kojoj se nalazi Excel dokument, te naziv samog dokumenta.

```
17  #UČITAVANJE PODATAKA
18  data <- read_excel("C:/Users/marko/Desktop/Indeks_kvalitete_zraka.xlsx")
```

Slika 8.2.5. Prikaz naredbe za učitavanje podataka iz Excel-a u programski jezik R.

Uvid u učitani set podataka dobiva se naredbom *head(data)*.

```
> head(data)
# A tibble: 6 × 9
  `PROSJEČNA TEMPERATURA` `MAKSIMALNA TEMPERATURA` `MINIMALNA TEMPERATURA`
                    <dbl>                    <dbl>                    <dbl>
1                     7.4                      9.8                      4.8
2                     7.8                     12.7                      4.4
3                     6.7                     13.4                      2.4
4                     8.6                     15.5                      3.3
5                    12.4                     20.9                      4.4
6                    16                       25.2                     10
# i 6 more variables: `ATMOSFERSKI TLAK U RAZINI MORA` <dbl>,
#   `PROSJEČNA RELATIVNA VLAŽNOST` <dbl>, `PROSJEČNA VIDLJIVOST` <dbl>,
#   `PROSJEČNA BRZINA VJETRA` <dbl>, `MAKSIMALNA BRZINA VJETRA` <dbl>,
#   AQI <dbl>
```

Slika 8.2.6. Prikaz varijabli i njihovih vrijednosti pomoću naredbe head.

Funkcija summary(data) dat će cjelokupnu perspektivu statističke distribucije podataka. Drugim riječima, ova funkcija odnosno naredba daje uvid u maksimalnu, minimalnu i srednju vrijednost određene značajke, njen medijan te prvi i treći kvartil za numerički vektor. Prvi kvartil predstavlja vrijednost od koje je 25% podataka manje, a treći kvartil predstavlja vrijednost od koje je 75% podataka manje.

```
> summary(data)
 PROSJEČNA TEMPERATURA MAKSIMALNA TEMPERATURA MINIMALNA TEMPERATURA
 Min.   : 6.70         Min.   : 9.80          Min.   : 0.00
 1st Qu.:19.00         1st Qu.:27.77          1st Qu.:12.10
 Median :27.95         Median :34.40          Median :21.40
 Mean   :25.70         Mean   :32.47          Mean   :19.34
 3rd Qu.:31.40         3rd Qu.:37.00          3rd Qu.:26.00
 Max.   :38.50         Max.   :45.50          Max.   :34.00
 ATMOSFERSKI TLAK U RAZINI MORA PROSJEČNA RELATIVNA VLAŽNOST
 Min.   : 990.4                 Min.   :20.00
 1st Qu.:1001.2                 1st Qu.:54.00
 Median :1007.9                 Median :65.00
 Mean   :1007.9                 Mean   :63.34
 3rd Qu.:1014.8                 3rd Qu.:75.00
 Max.   :1023.2                 Max.   :98.00
 PROSJEČNA VIDLJIVOST PROSJEČNA BRZINA VJETRA MAKSIMALNA BRZINA VJETRA
 Min.   :0.300        Min.   : 0.400          Min.   : 1.90
 1st Qu.:1.600        1st Qu.: 3.500          1st Qu.:11.10
 Median :1.900        Median : 5.900          Median :14.80
 Mean   :2.009        Mean   : 6.488          Mean   :15.66
 3rd Qu.:2.600        3rd Qu.: 8.900          3rd Qu.:18.30
 Max.   :5.800        Max.   :24.400          Max.   :57.60
      AQI
 Min.   :  0.00
 1st Qu.: 43.96
 Median : 83.46
 Mean   :108.26
 3rd Qu.:152.69
 Max.   :404.50
```

Slika 8.2.7. Prikaz podataka pomoću naredbe summary.

Osim navedenih naredbi, u ovom se programskom jeziku mogu koristiti i naredbe za provjeru ima li u danom skupu neželjenih podataka, vrijednosti koje nedostaju te dupliciranih vrijednosti provodi se na sljedeći način.

Naredbom *sum(is.na(data))* provjerava se postoji li u skupu podataka vrijednost koja nedostaje to jest vrijednost koja nije izmjerena. Ako postoji, ova naredba prikazuje ukupan broj takvih vrijednosti, odnosno njihovu sumu. Ono što programski jezik ispisuje kao rezultat prikazano je na *Slika 8.2.8.*

```
> sum(is.na(data))
[1] 0
```

Slika 8.2.8. Rezultat upotrebe sum(is.na(data)) naredbe.

Iz dobivenog rezultata vidljivo je da je ukupan broj vrijednosti koje nedostaju nula.

Naredbom table(duplicated(data)) provjerava se broj dupliciranih vrijednosti u danom skupu podataka. Skup koji se obrađuje u ovom radu nema dupliciranih vrijednosti *(Slika 8.2.9.)*.

```
> table(duplicated(data))

FALSE
 1088
```

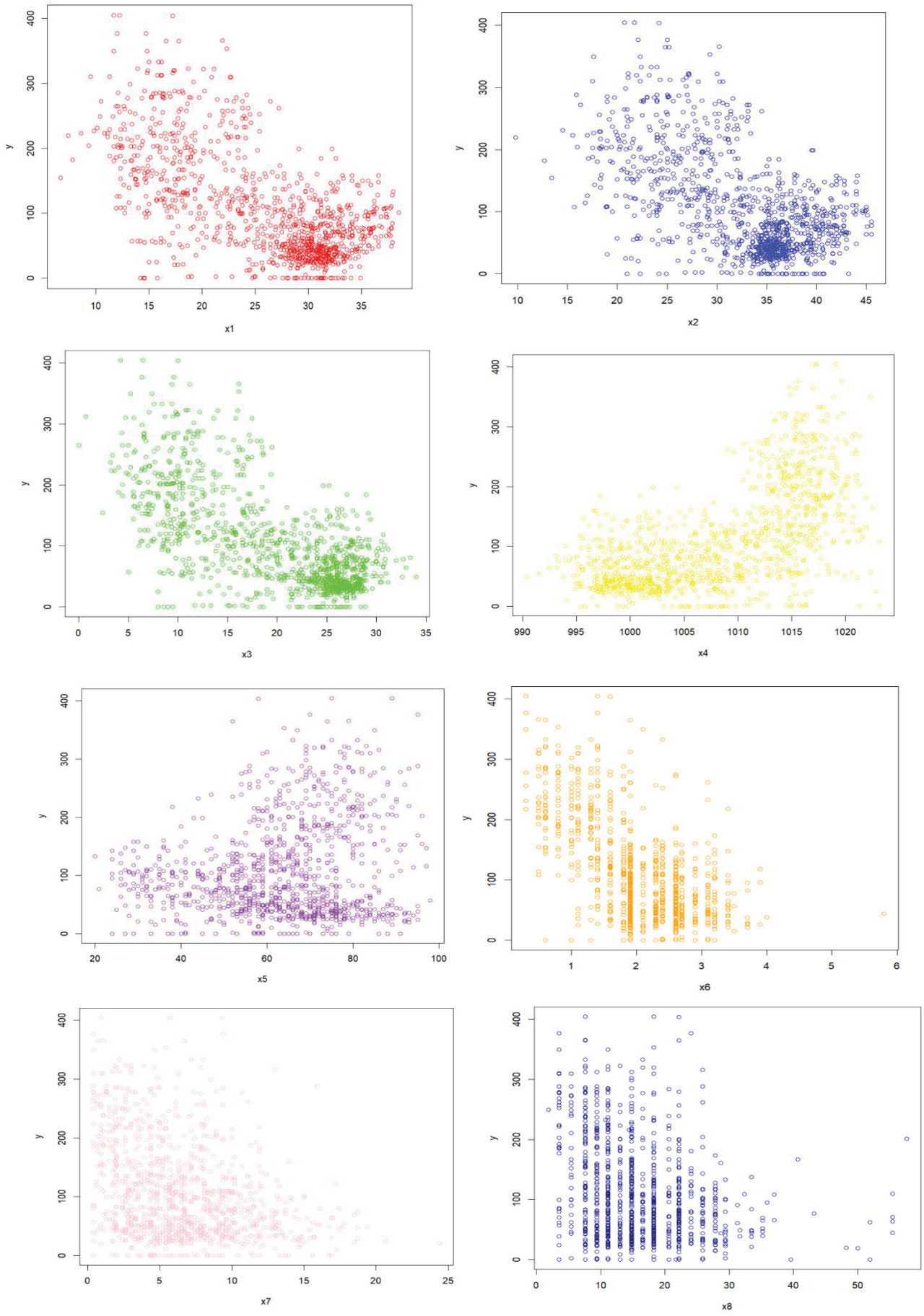
Slika 8.2.9. Rezultat naredbe table(duplicated(data)).

Određivanje dimenzije skupa podataka provodi se funkcijom *dim(data)*..

```
> dim(data)
[1] 1088    9
```

Slika 8.2.10. Određivanje dimenzija promatranog skupa podataka.

Posljednji važan korak u procesu „upoznavanja“ podataka jest njihova vizualizacija.
Na *Slika 8.2.11.* prikazane su ovisnosti izlazne varijable o svakoj ulaznoj varijabli zasebno.



Slika 8.2.11. Grafički prikaz ovisnosti ulaznih varijabli x1-prosječna temperatura, x2- maksimalna temperatura, x3-minimalna temperatura, x4-atmosferski tlak u razini mora, x5- prosječna relativna vlažnost, x6-prosječna vidljivost, x7-prosječna brzina vjetra, x8- maksimalna brzina vjetra o izlaznoj varijabli y-indeks kvalitete zraka, AQI.

Iz prikazanih grafičkih ovisnosti vidljivo je da se indeks kvalitete zraka smanjuje povećanjem temperature. Također se može uočiti da se indeks kvalitete zraka povećava povećanjem atmosferskog tlaka te povećanjem prosječne relativne vlažnosti. AQI nema nikakvu uočenu specifičnu ovisnost o prosječnoj vidljivosti, prosječnoj i maksimalnoj brzini vjetra.

```
#ODREĐIVANJE X I Y OSI ZA CRTANJE GRAFIČKIH PRIKAZA

x1 <- data$`PROSJEČNA TEMPERATURA`
x2 <- data$`MAKSIMALNA TEMPERATURA`
x3 <- data$`MINIMALNA TEMPERATURA`
x4 <- data$`ATMOSFERSKI TLAK U RAZINI MORA`
x5 <- data$`PROSJEČNA RELATIVNA VLAŽNOST`
x6 <- data$`PROSJEČNA VIDLJIVOST`
x7 <- data$`PROSJEČNA BRZINA VJETRA`
x8 <- data$`MAKSIMALNA BRZINA VJETRA`
y <- data$AQI

#CRTANJE GRAFIČKIH PRIKAZA

plot(x1, y, col="red")
plot(x2, y, col="blue")
plot(x3, y, col="green")
plot(x4, y, col="yellow")
plot(x5, y, col="purple")
plot(x6, y, col="orange")
plot(x7, y, col="pink")
plot(x8, y, col="navyblue")
```

Slika 8.2.12. Kodovi za crtanje grafičkih prikaza.

Nakon uvida u promatrani skup podataka, slijedi priprema istih. Priprema podataka provodi se postupcima skaliranja i centriranja. Postupak skaliranja podataka jest tehnika za usporedbu podataka koji se ne mjere na isti način. Skaliranje značajki koje utječu na izlaznu varijablu može se provesti postupkom standardizacije. Standardizacija jest proces u kojem se podaci rekonstruiraju u jedinstven format. Dakle, sve vrijednosti smanjuju se na skalu koja je zajednička svima, obično u rasponu od -3 do +3 te se relativni raspon između vrijednosti održava netaknut. Transformacija podataka u jedinstven format provodi se sljedećom formulom:

$$z = \frac{x_i - \mu}{\sigma} \qquad (4)$$

Gdje je z- rezultat to jest podatak u novom transformiranom formatu

x_i- trenutna vrijednost

μ- srednja vrijednost

σ- standardna devijacija

U R programskom jeziku, funkcija koja se koristi za standardizaciju podataka jest *scale()*.

Dakle funkcija *scale()* koristi se za skaliranje i centriranje vrijednosti u vektoru, matrici ili podatkovnom okviru. Prilikom izrade modela strojnog učenja, ova je funkcija koristila oblik *scale(x, center=TRUE, scale=TRUE)*.

U ovoj funkciji *x* predstavlja naziv objekta za skaliranje. *center* je funkcija koja ispituje treba li oduzeti srednju vrijednost od trenutne pri skaliranju. Zadana vrijednost je *TRUE*, dakle potrebno je oduzeti. *scale* je funkcija koja ispituje treba li podijeliti sve sa standardnom devijacijom prilikom skaliranja. Zadana vrijednost je *TRUE*, dakle treba.

```
44  #SKALIRANJE I CENTRIRANJE PODATAKA
45
46  data <- scale(data, center = TRUE, scale = TRUE)
47  data <- data.frame(data)
```

Slika 8.2.13. Kod za skaliranje i centriranje podataka u R programskom jeziku.

Osim naredbe za skaliranje, upotrebljena je i naredba *data.frame(data)*. Ova naredba služi za izradu „tablice“ u kojoj svaki stupac sadrži vrijednost jedne varijable, a svaki red jedan skup vrijednosti iz svakog stupca.

```
> data <- scale(data, center = TRUE, scale = TRUE)
> data <- data.frame(data)
> head(data,10)
   PROSJEČNA.TEMPERATURA MAKSIMALNA.TEMPERATURA MINIMALNA.TEMPERATURA ATMOSFERSKI.TLAK.U.RAZINI.MORA PROSJEČNA.RELATIVNA.VLAŽNOST PROSJEČNA.VIDLJIVOST
1             -2.535937             -3.393062            -1.921026                    1.2813852                    1.8716166          -2.0722211
2             -2.480511             -2.958979            -1.973861                    1.4007949                    1.4930483          -1.9349395
3             -2.632932             -2.854200            -2.238035                    1.5202045                    1.1775747          -1.9349395
4             -2.369659             -2.539864            -2.119156                    1.4273303                    0.5466276          -1.6603762
5             -1.843113             -1.731572            -1.973861                    1.2415820                   -0.1474143          -0.9739679
6             -1.344281             -1.087932            -1.234174                    0.6976049                    0.9882906          -1.9349395
7             -1.704549             -1.716603            -1.339844                    0.9496918                    1.4930483          -2.0722211
8             -1.579841             -1.477109            -1.683270                    1.1089047                    0.7990064          -1.6603762
9             -1.801544             -2.030939            -1.590809                    1.7590237                    0.7990064          -0.2875596
10            -1.912396             -2.030939            -1.722896                    1.6130786                    0.7990064          -1.2485312
   PROSJEČNA.BRZINA.VJETRA MAKSIMALNA.BRZINA.VJETRA         AQI
1              -0.56210317               -0.8275160  1.354568552
2              -0.53641026               -0.6028743  0.898448465
3              -0.43363864               -0.6028743  0.556358400
4               0.41422727                0.6524762  1.396950047
5               0.56838471                0.8639037  1.122761518
6              -0.43363864               -0.6028743  2.150601398
7              -1.28150455               -1.0653719  1.562425230
8              -1.56412652               -1.6071548  2.049533937
9              -0.09963086                0.1106934 -0.003107723
10              0.46561308                0.6524762 -0.007664873
```

Slika 8.2.14. Skalirani podaci (prvih 10 vrijednosti svake varijable).

Na prikazu na *Slika 8.2.14.* vidljivi su skaliranu podaci. Svi podaci, uključujući i one ne prikazane nalaze se unutar intervala od -3 do 3.

Tek nakon što su podaci skalirani, može se započeti s izradom modela. Obučavanje modela s neskaliranim podacima može trajati dosta vremena, što nikako nije cilj.

8.3. IZRADA MODELA I PRIMJENA NA REALNI SUSTAV

8.3.1. VIŠESTRUKA LINEARNA REGRESIJA

Dakle, model višestruke linearne regresije koristi se za obradu podataka u kojima postoji nekoliko ulaznih, odnosno nezavisnih varijabli i jedna izlazna odnosno zavisna varijabla. Kao što je već i napomenuto skup podataka koji se obrađuje u ovom radu sadrži osam nezavisnih i jednu zavisnu varijablu. Detaljniji opis modela prikazan je u podnaslovu 4.2.

Nakon instaliranja i učitavanja potrebnih paketa, obrade i pripreme podataka te upoznavanja s podacima slijedi izrada modela.

```
#UČITAVANJE PAKETA-MODEL VIŠESTRUKE LINEARNE REGRESIJE

library(caret)
library(ggplot2)
library(readxl)
library(dplyr)
library(caTools)

#UČITAVANJE PODATAKA
data <- read_excel("C:/Users/marko/Desktop/Indeks_kvalitete_zraka.xlsx")

#SKALIRANJE I CENTRIRANJE PODATAKA
data <- scale(data, center = TRUE, scale = TRUE)
data <- data.frame(data)

#IZRADA MODELA
set.seed(1)
split = sample.split(data$AQI, SplitRatio = 0.8)
train = subset(data, split == TRUE)
test = subset(data, split == FALSE)
model <- lm(AQI ~ . ,data = train)
summary(model)
```

Slika 8.3.1.1. Prikaz koda za višestruku linearnu regresiju.

Funkcija *set.seed(1)* koristi se za stvaranje ponovljivih rezultata prilikom pisanja koda koji uključuje stvaranje varijabli koje poprimaju nasumične vrijednosti. Točnije, ovom se funkcijom jamči da će kod proizvoditi svaki put iste nasumične vrijednosti kad god se on pokrene.

Drugim dijelom koda, skup podataka je podijeljen na dva dijela. Prvi dio podataka (80%) koristi se za treniranje modela- train podaci, dok se drugi dio podataka (20%) koristi za testiranje modela- test podaci.

Samo treniranje modela pokrenuto je naredbom *lm(AQI~. ,data = train).*

```
> summary(model)

Call:
lm(formula = AQI ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.32924 -0.39556 -0.04899  0.31388  2.84890

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   0.003329   0.022530   0.148  0.88258
PROSJEČNA.TEMPERATURA         0.150770   0.179742   0.839  0.40181
MAKSIMALNA.TEMPERATURA        0.027480   0.096697   0.284  0.77634
MINIMALNA.TEMPERATURA        -0.631237   0.129661  -4.868 1.34e-06 ***
ATMOSFERSKI.TLAK.U.RAZINI.MORA -0.001230   0.058207  -0.021  0.98315
PROSJEČNA.RELATIVNA.VLAŽNOST  -0.148189   0.047205  -3.139  0.00175 **
PROSJEČNA.VIDLJIVOST          -0.375289   0.030429 -12.333  < 2e-16 ***
PROSJEČNA.BRZINA.VJETRA       -0.079574   0.040087  -1.985  0.04746 *
MAKSIMALNA.BRZINA.VJETRA      -0.032436   0.037839  -0.857  0.39157
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.664 on 861 degrees of freedom
Multiple R-squared:  0.5552,    Adjusted R-squared:  0.5511
F-statistic: 134.3 on 8 and 861 DF,  p-value: < 2.2e-16
```

Slika 8.3.1.2. Prikaz rada modela.

Nakon treniranja i testiranja modela, izvedeni su kodovi za predikciju indeksa kvalitete zraka.

```
#IZDVAJANJE Y VRIJEDNOSTI DOBIVENIH IZ POČETNIH MJERNIH PODATAKA
AQI <- test$AQI

#BRISANJE Y VRIJEDNOSTI IZ TEST SKUPA PODATAKA
test <- test[, -which(names(test) == "AQI")]

#RAČUNANJE PREDIKCIJA - Y VRIJEDNOST - INDEKS KVALITETE ZRAKA
predictions <- predict(model, test)

# PRETVARANJE NAMED NUM U NUM PODATKE
names(predictions) <- NULL

#PRIKAZ Y VRIJEDNOSTI
AQI
predictions
head(AQI)
head(predictions)
```

Slika 8.3.1.3. Kodovi za predikciju indeksa kvalitete zraka, AQI.

```
> AQI
  [1]  1.396950047  2.150601398  1.562425230  0.861130469 -0.063869724
  [6] -0.345400328  0.113352779  0.187786230  0.115378179 -1.315566942
 [11]  0.241459330 -0.119568225 -0.150961925 -0.437556029 -0.523635531
 [16] -1.315566942 -0.887701186 -0.457810030 -0.812761385 -0.970742587
 [21] -0.308436778 -0.557560981 -0.844155085  0.345767432  1.408596097
 [26]  0.710339437  0.827812639  0.944273141  0.158924279  2.137740108
 [31]  0.738188688  1.793928453  3.600078929 -1.315566942  1.286565746
 [36]  1.248589495  0.266270481  0.180697329  0.268295881  0.029805027
 [41] -0.099820575 -0.412744879  0.157405229  1.101241643 -0.579840382
 [46] -0.061844324  0.548813785 -1.010744238 -1.044163338 -0.792507385
 [51] -0.716554884 -0.806685185 -0.623892832 -0.805672485 -1.315566942
 [56] -1.315566942 -0.759594634 -1.174801640  0.580207485  1.303781646
 [61]  1.593920200 -0.200584226  0.139176629  1.158459194  0.969590641
 [66]  1.397456397  1.991404956  1.266311745  1.416191347  0.346273782
 [71]  0.145252829  0.706288637 -0.891245636 -0.738327934  0.250067281
 [76] -0.229446176 -0.083617374 -0.669970683 -0.928715537 -0.736302534
 [81] -0.730732684 -0.631994432 -0.726175534 -0.568194331 -0.823394735
 [86] -0.418314729 -0.968717187 -0.821875685 -1.213284241 -1.074544339
 [91] -0.841116985 -0.808710585 -0.960615587 -0.779342284 -0.168684176
 [96] -0.638576982  0.049046328 -0.323627278  1.807093553  0.833888839
```

Slika 8.3.1.4. Prikaz y vrijednosti-indeks kvalitete zraka dobivenih iz početnih mjernih podataka (prvih 100 vrijednosti).

```
> predictions
  [1]  1.4001875761  1.1827243076  1.2367143763  0.9884288116  0.4879720517
  [6]  0.8679981741  0.3284431091  0.5233314772  0.2391818024 -0.1652663714
 [11]  0.0035386496 -0.2937915806 -0.4408589237 -0.1378980381 -0.5729632306
 [16] -0.6270427387 -0.4746950898 -0.2869345812 -0.4657419878 -0.5668833981
 [21] -0.3197879878 -0.6957408200 -0.4353584581  0.5501392955  1.0744531549
 [26]  0.7204830175  1.0949961714  1.1478988470  0.4973188407  0.8982701076
 [31]  1.0938053431  0.8790300013  1.5213348502  0.9675545645  0.7356567248
 [36]  0.8140952740  0.6237064710  0.6475117411  0.3520943865 -0.0309867065
 [41] -0.0298440032  0.1970506175 -0.0281574271  0.0019939058 -0.3717117153
 [46]  0.0267892706 -0.5610220933 -0.4867885184 -0.8540129602 -1.1076934378
 [51] -0.5624051904 -0.4069698971 -0.4899700317 -0.6670848288 -0.4729373068
 [56] -0.4962904613  0.0509543393 -0.2178787882  0.3950951707  0.6177452489
 [61]  0.5949581953  0.4530323362  0.8962626964  0.8457123176  1.1371231830
 [66]  1.1557002249  1.3935680985  0.9639884105  1.1902340834  0.9660774779
 [71]  0.4288272279  0.1812648127 -0.0382523639 -0.1506542893  0.5463109186
 [76] -0.0186008289  0.0125857964 -0.3283038543 -0.3363610517 -0.4366861457
 [81] -0.4816851134 -0.3254613786 -0.5279681434 -0.2149151795 -0.1694488901
 [86] -0.5465692013 -0.2954656969 -0.3980261926 -0.4557634245 -0.3664220921
 [91] -0.4502806081 -0.6569727997 -0.8821051935 -0.6676846137 -0.2646241399
 [96] -0.4541140291 -0.2394151819 -0.2875887233  0.5912065258  0.5946391176
```

Slika 8.3.1.4. Prikaz y vrijednosti-indeks kvalitete zraka dobivenih predikcijom pomoću modela višestruke linearne regresije (prvih 100 vrijednosti).

```
> head(AQI)
[1]  1.39695005  2.15060140  1.56242523  0.86113047 -0.06386972 -0.34540033
> head(predictions)
[1] 1.4001876 1.1827243 1.2367144 0.9884288 0.4879721 0.8679982
```

Slika 8.3.1.5. Usporedba indeksa kvalitete zraka dobivenog iz početnih mjernih podataka- AQI te predikcijom pomoću modela višestruke linearne regresije- predictions.

Ako se usporedi prvih nekoliko vrijednosti indeksa kvalitete zraka dobivenih iz početnih mjernih podataka- AQI te predikcijom pomoću modela višestruke linearne regresije- predictions uočljivo je da su neke od vrijednosti vrlo dobro predviđene pomoću strojnog učenja, primjerice prva i četvrta vrijednost dok neke jako odstupaju jedne od drugih, primjerice zadnja i predzadnja.

```
43  #ODREĐIVANJE KORIJENA SREDNJE KVADRATNE POGREŠKE
44  RMSE(predictions,AQI)
45
46  #GRAFIČKI PRIKAZ
47  plot (predictions,AQI, col="blue")
48  abline(0,1, col="red")
```

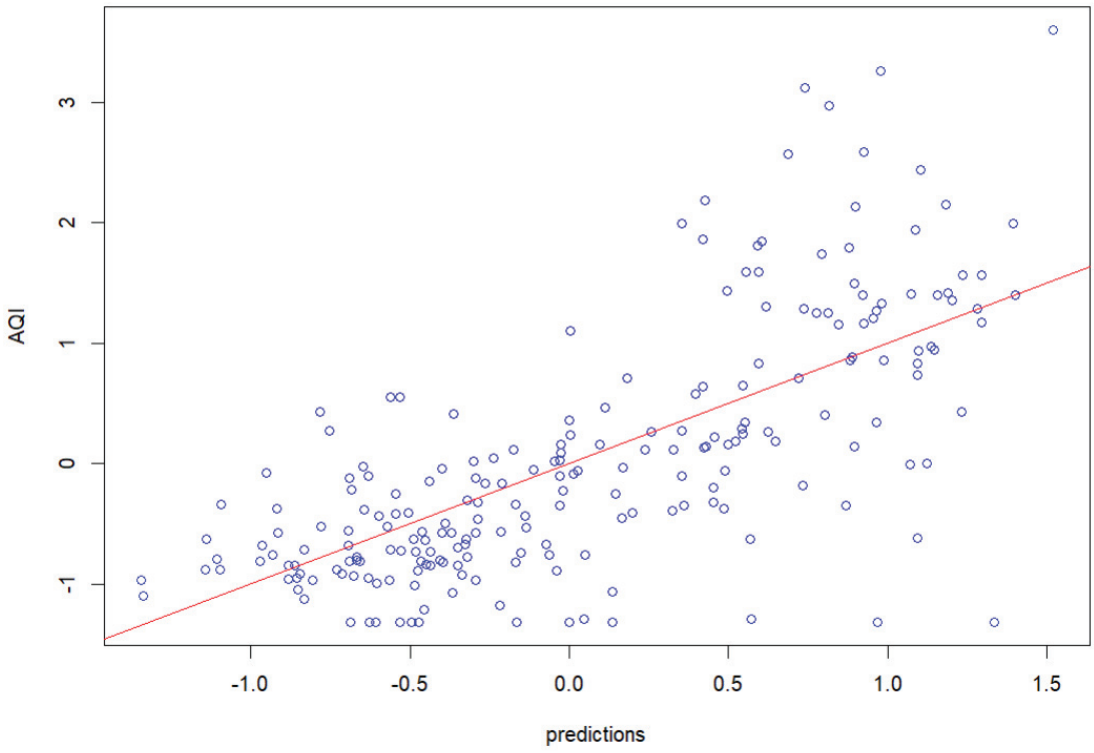
Slika 8.3.1.6. Kodovi za određivanje RMSE te grafički prikaz rezultata.

RMSE, odnosno korijen srednje kvadratne pogreške mjeri prosječnu razliku između predviđenih vrijednosti modela i stvarnih vrijednosti. Dakle, ona određuje koliko su rezultati dispergirani. Kako se podatkovne točke približavaju regresijskoj liniji, model ima manje pogreške, a RMSE se smanjuje. Model s manjom pogreškom daje preciznija predviđanja. Vrijednost 0 znači da predviđene vrijednosti savršeno odgovaraju stvarnim vrijednostima, ali naravno u praksi se to nikad neće vidjeti.

```
> #ODREĐIVANJE KORIJENA SREDNJE KVADRATNE POGREŠKE
> RMSE(predictions,AQI)
[1] 0.7279787
```

Slika 8.3.1.7. Prikaz rezultata RMSE.

RMSE iznosi 0,7279787.



Slika 8.3.1.7. Grafički prikaz ovisnosti stvarnih i predviđenih vrijednosti.

Iz grafičkog prikaza na *Slika 8.3.1.7.* vidljivo je da se stvarne i predviđene vrijednosti ne podudaraju u potpunosti. Drugim riječima, uočljivo je da model višestruke linearne regresije ne predviđa podatke sa 100%-tnom točnošću. Upravo to i potvrđuje pogreška RMSE, koja iznosi 0,7279787.

8.3.2. SLUČAJNA ŠUMA

Model slučajne šume sastoji se od velikog broja stabala odluke. Svako stablo odluke daje izlaz. Kombiniranjem izlaza svih stabla odluke, algoritam slučajne šume daje konačan rezultat. Detaljniji opis ovog algoritma prikazan je u podnaslovu 4.5.

Kao i kod modela višestruke linearne regresije, nakon instaliranja i učitavanja potrebnih paketa, obrade i pripreme podataka te upoznavanja s podacima slijedi izrada modela.

```
#UČITAVANJE PAKETA- MODEL SLUČAJNE ŠUME

library(caret)
library(randomForest)
library(ggplot2)
library(readxl)
library(dplyr)
library(caTools)

#UČITAVANJE PODATAKA
data <- read_excel("C:/Users/marko/Desktop/Indeks_kvalitete_zraka.xlsx")

#SKALIRANJE I CENTRIRANJE PODATAKA
data <- scale(data, center = TRUE,scale = TRUE)
data <- data.frame(data)
```

Slika 8.3.2.1. Prikaz početka koda za model slučajne šume.

Pomoću naredbi prikazanih na *Slika 8.3.2.1.* učitani su programski paketi potrebni za izradu modela slučajne šume. Također učitan je skup podataka koji se promatra danim algoritmom te je provedeno skaliranje i centriranje tih podataka. Prilikom izrade bilo kojeg od modela, potrebno je prije skaliranja i centriranja podataka, pobliže se upoznati s njima kako je prikazano u podnaslovu 8.2.

```r
#IZRADA MODELA
set.seed(1)
split = sample.split(data$AQI, SplitRatio = 0.8)
train = subset(data, split == TRUE)
test = subset(data, split == FALSE)
trC <- trainControl(method = "cv",
                    number = 10,
                    search = "grid")

tuneGrid <- expand.grid(.mtry = c(1: 10))
rf_mtry <- train(AQI ~ .,
                 data = data,
                 method = "rf",
                 tuneGrid = tuneGrid,
                 trControl = trC,
                 importance = TRUE,
                 nodesize = 14,
                 ntree = 300)

best_mtry <- rf_mtry$bestTune$mtry

tuneGrid <- expand.grid(.mtry = best_mtry)

fit_rf <- train(AQI ~ .,
                data,
                method = "rf",
                tuneGrid = tuneGrid,
                trControl = trC,
                importance = TRUE,
                nodesize = 14,
                ntree = 800,
                maxnodes = 24)

print(fit_rf)
```

Slika 8.3.2.2. Prikaz koda za model slučajne šume.

Kao i kod modela višestruke linearne regresije, funkcija *set.seed(1)* jamči da će kod proizvoditi svaki put iste nasumične vrijednosti kad god se pokrene.

Drugim dijelom koda, skup podataka je podijeljen na dva dijela. Prvi dio podataka (80%) koristi se za treniranje modela- train podaci, dok se drugi dio podataka (20%) koristi za testiranje modela- test podaci.

Naredba *trainControl()* se koristi za provođenje unakrsne provjere valjanosti. Unakrsna provjera valjanosti odnosi se na skup metoda za mjerenje izvedbe određenog prediktivnog modela na novim testnim skupovima podataka. Ona se temelji na podjeli podataka u dva skupa, kao što je i prikazano na *Slika 8.3.2.2.*, na podatke za treniranje i podatke za testiranje. Također je poznata kao metoda ponovnog uzorkovanja jer uključuje prilagođavanje iste metode više puta koristeći različite podskupove podataka. Metoda koja se koristi za ponovno uzorkovanje skupa podataka jest „cv“. Dakle, ovom se naredbom određuju kontrolni parametri.

Funkcijom *expand.grid()* stvara se okvir podataka sa svim vrijednostima koje se mogu formirati kombinacijom svih vektora ili faktora proslijeđenih kao argument.

Nadalje, slučajna šuma jest model koji nastaje kombinacijom modela stabala odluke. Različita stabla odluke treniraju se neovisno jedno o drugom, a zatim se kombiniraju izlazi svih stabala kako bi se stvorilo konačno predviđanje. Kako stabla koja se treniraju ne bi izgledala potpuno isto, potrebno je dodati malo slučajnosti u proces stvaranja modela slučajne šume. Parametar *mtry* čini upravo to. Ovaj parametar kontrolira koliko se slučajnosti dodaje u proces izrade stabala odluke. Dakle, ovaj parametar kontrolira koliko ulaznih značajki stablo odluke ima na raspolaganju za razmatranje u bilo kojem trenutku. Drugim riječima, prilikom formiranja svakog razdvajanja u stablu, algoritam nasumično odabire drugačiji slučajni skup varijabli unutar kojih se bira najbolja točka podjele odnosno razdvajanja u stablu.

Također, za izradu modela slučajne šume važni su još i parametri *ntree*, *nodesize*, *maxnodes* te *importance*. *ntree* parametar je koji određuje broj stabala u šumi. *nodesize* određuje broj čvorova u stablu odluke, dok *maxnodes* određuje maksimalan broj terminalnih čvorova u šumi. Posljednji parametar, *importance* ispituje treba li procijeniti važnost nezavisnih varijabli u slučajnoj šumi.

Na kraju izvoda koda za algoritam slučajne šume, prikazan je rad samog modela pomoću naredbe *print()*.

```
> print(fit_rf)
Random Forest

1088 samples
   8 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 980, 980, 977, 978, 980, 979, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.6312635  0.6063362  0.4518272

Tuning parameter 'mtry' was held constant at a value of 4
```

Slika 8.3.2.3. Prikaz rada modela.

Kako bi se provjerila važnost svih ulaznih varijabli koje utječu na ishod, odnosno izlaznu varijablu koristi se naredbe *varImp(fit_rf)*.

```
51  #VAŽNOST VARIJABLI
52  varImp(fit_rf)
```

Slika 8.3.2.4. Prikaz koda u programskom jeziku R za provjeru važnosti varijabli.

```
> #VAŽNOST VARIJABLI
> varImp(fit_rf)
rf variable importance

                                 Overall
PROSJEČNA.VIDLJIVOST              100.00
PROSJEČNA.RELATIVNA.VLAŽNOST       54.50
MINIMALNA.TEMPERATURA              43.94
PROSJEČNA.TEMPERATURA              42.81
ATMOSFERSKI.TLAK.U.RAZINI.MORA     27.82
PROSJEČNA.BRZINA.VJETRA            26.85
MAKSIMALNA.TEMPERATURA             18.19
MAKSIMALNA.BRZINA.VJETRA            0.00
```

Slika 8.3.2.5. Rezultat važnosti varijabli.

Iz prikaza na *Slika 8.3.2.5.* može se vidjeti kako na indeks kvalitete zraka najviše utječe prosječna vidljivost, a zatim prosječna relativna vlažnost. Maksimalna temperatura i maksimalna brzina vjetra varijable su koje najmanje utječu na indeks kvalitete zraka.

Nakon treniranja i testiranja modela, izvedeni su kodovi za predikciju indeksa kvalitete zraka.

```
54  #IZDVAJANJE Y VRIJEDNOSTI DOBIVENIH IZ POČETNIH MJERNIH PODATAKA
55  AQI <- test$AQI
56
57  #BRISANJE Y VRIJEDNOSTI IZ TEST SKUPA PODATAKA
58  test <- test[, -which(names(test) == "AQI")]
59
60  #RAČUNANJE PREDIKCIJA - Y VRIJEDNOST - INDEKS KVALITETE ZRAKA
61  predictions <- predict(fit_rf, test)
62
63  # PRETVARANJE NAMED NUM U NUM PODATKE
64  names(predictions) <- NULL
65
66  #PRIKAZ Y VRIJEDNOSTI
67  AQI
68  predictions
69  head(AQI)
70  head(predictions)
```

Slika 8.3.2.6. Kodovi za predikciju indeksa kvalitete zraka, AQI.

```
> #PRIKAZ Y VRIJEDNOSTI
> AQI
  [1]  1.396950047  2.150601398  1.562425230  0.861130469 -0.063869724
  [6] -0.345400328  0.113352779  0.187786230  0.115378179 -1.315566942
 [11]  0.241459330 -0.119568225 -0.150961925 -0.437556029 -0.523635531
 [16] -1.315566942 -0.887701186 -0.457810030 -0.812761385 -0.970742587
 [21] -0.308436778 -0.557560981 -0.844155085  0.345767432  1.408596097
 [26]  0.710339437  0.827812639  0.944273141  0.158924279  2.137740108
 [31]  0.738188688  1.793928453  3.600078929 -1.315566942  1.286565746
 [36]  1.248589495  0.266270481  0.180697329  0.268295881  0.029805027
 [41] -0.099820575 -0.412744879  0.157405229  1.101241643 -0.579840382
 [46] -0.061844324  0.548813785 -1.010744238 -1.044163338 -0.792507385
 [51] -0.716554884 -0.806685185 -0.623892832 -0.805672485 -1.315566942
 [56] -1.315566942 -0.759594634 -1.174801640  0.580207485  1.303781646
 [61]  1.593920200 -0.200584226  0.139176629  1.158459194  0.969590641
 [66]  1.397456397  1.991404956  1.266311745  1.416191347  0.346273782
 [71]  0.145252829  0.706288637 -0.891245636 -0.738327934  0.250067281
 [76] -0.229446176 -0.083617374 -0.669970683 -0.928715537 -0.736302534
 [81] -0.730732684 -0.631994432 -0.726175534 -0.568194331 -0.823394735
 [86] -0.418314729 -0.968717187 -0.821875685 -1.213284241 -1.074544339
 [91] -0.841116985 -0.808710585 -0.960615587 -0.779342284 -0.168684176
 [96] -0.638576982  0.049046328 -0.323627278  1.807093553  0.833888839
```

Slika 8.3.2.7. Prikaz y vrijednosti-indeks kvalitete zraka dobivenih iz početnih mjernih podataka (prvih 100 vrijednosti).

```
> predictions
  [1]  1.302109449  1.293372525  1.517845486  0.916888661  0.540111231
  [6]  0.994813119  0.004750111  0.112558477 -0.182587340 -0.582734272
 [11] -0.402214956 -0.238496792 -0.306136732 -0.409189431 -0.550867677
 [16] -0.637118158 -0.750858797 -0.665374122 -0.765506782 -0.750889278
 [21] -0.674539579 -0.778053500 -0.768295507  0.514360352  1.136284177
 [26]  1.008610765  1.173019573  1.176052966  0.690025127  1.158890621
 [31]  1.111500475  1.037538694  1.974170423  0.494977767  1.074138581
 [36]  1.012188712  0.971261801  0.758339232  0.231654905  0.006043968
 [41] -0.121767425 -0.222794259 -0.328484510 -0.125694918 -0.404390855
 [46] -0.431995332 -0.443202978 -0.608253790 -0.770251466 -0.441823833
 [51] -0.747543475 -0.731864576 -0.719552477 -0.586332061 -0.678386122
 [56] -0.693736411 -0.625130877 -0.460001725  0.215827132  1.047431678
 [61]  0.967413514  0.046971567  0.835253825  1.037864189  1.104549472
 [66]  1.192293091  1.720722887  1.151424286  1.072902752  0.995410735
 [71]  0.346234340  0.618170489 -0.216129354 -0.006892784  0.742020302
 [76] -0.183825730 -0.160842476 -0.335725035 -0.688850301 -0.718513929
 [81] -0.465854775 -0.474989628 -0.603728918 -0.562911567 -0.707389641
 [86] -0.395142121 -0.640730152 -0.731067674 -0.768987691 -0.739845642
 [91] -0.726446865 -0.738336961 -0.758810928 -0.717205195 -0.508225832
 [96] -0.732866709 -0.665976129 -0.592800584  1.200946096  1.275591609
```

Slika 8.3.2.8. Prikaz y vrijednosti-indeks kvalitete zraka dobivenih predikcijom pomoću modela slučajne šume (prvih 100 vrijednosti).

```
> head(AQI)
[1]  1.39695005  2.15060140  1.56242523  0.86113047 -0.06386972 -0.34540033
> head(predictions)
[1] 1.3021094 1.2933725 1.5178455 0.9168887 0.5401112 0.9948131
```

Slika 8.3.2.9. Usporedba indeksa kvalitete zraka dobivenog iz početnih mjernih podataka- AQI te predikcijom pomoću modela višestruke linearne regresije- predictions.

Ako se usporedi prvih nekoliko vrijednosti indeksa kvalitete zraka dobivenih iz početnih mjernih podataka- AQI te predikcijom pomoću modela slučajne šume- predictions uočljivo je da su neke od vrijednosti vrlo dobro predviđene pomoću strojnog učenja, primjerice prva i treća vrijednost dok neke jako odstupaju jedne od drugih, primjerice zadnja i predzadnja.

```
73  #ODREĐIVANJE KORIJENA SREDNJE KVADRATNE POGREŠKE
74  RMSE(predictions,AQI)
75
76  #GRAFIČKI PRIKAZ
77  plot (predictions,AQI, col="blue")
78  abline(0,1, col="red")
```

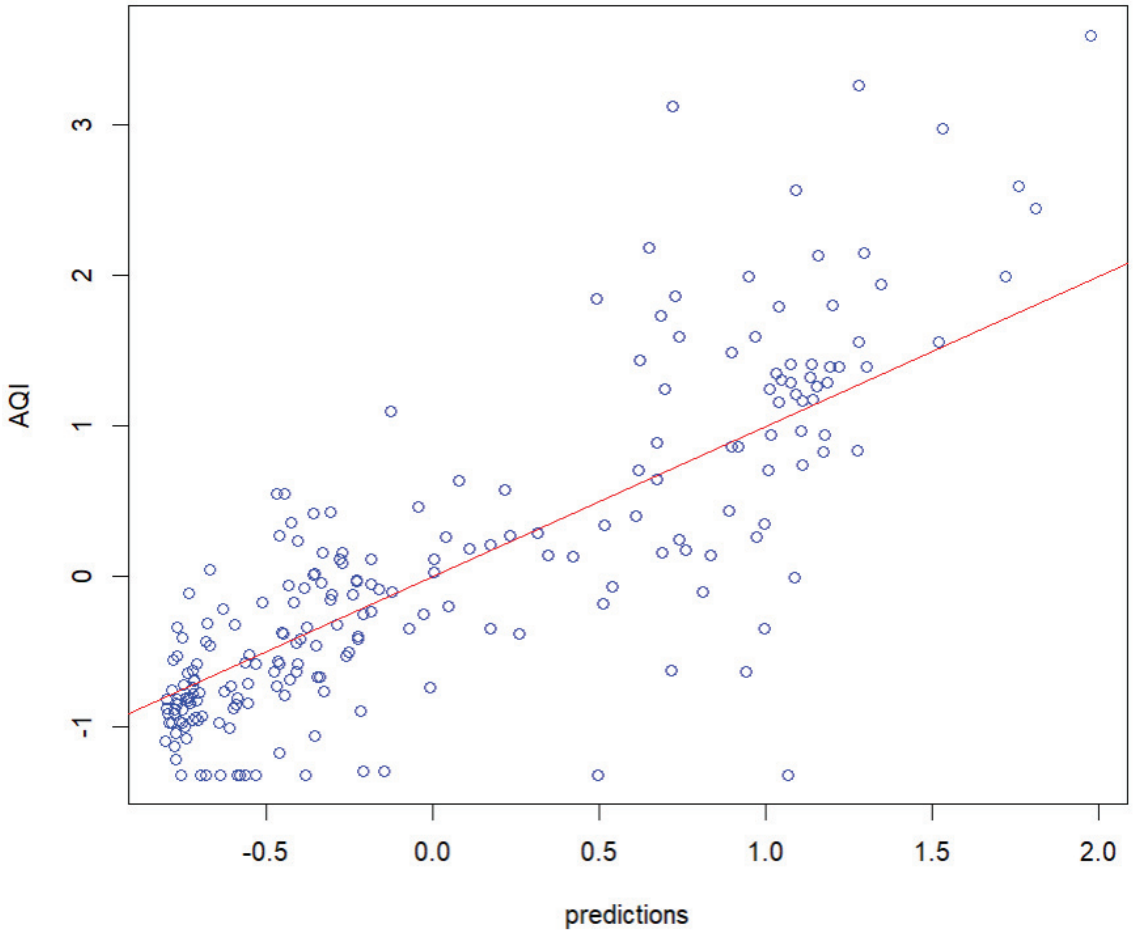
Slika 8.3.2.10. Kodovi za određivanje RMSE te grafički prikaz rezultata.

Kako je već i spomenuto, RMSE, odnosno korijen srednje kvadratne pogreške mjeri prosječnu razliku između predviđenih vrijednosti modela i stvarnih vrijednosti. Kako se podatkovne točke približavaju regresijskoj liniji, model ima manje pogreške, a RMSE se smanjuje.

```
> RMSE(predictions,AQI)
[1] 0.596345
```

Slika 8.3.2.11. Prikaz rezultata RMSE.

RMSE iznosi 0,596345.



Slika 8.3.2.12. Grafički prikaz ovisnosti stvarnih i predviđenih vrijednosti.

Iz grafičkog prikaza na *Slika 8.3.2.12.* vidljivo je da se stvarne i predviđene vrijednosti ne podudaraju u potpunosti. Drugim riječima, uočljivo je da model slučajne šume ne predviđa podatke sa 100%-tnom točnošću. Upravo to potvrđuje pogreška RMSE koja iznosi 0,596345. Također, može se uočiti da model slučajne šume predviđa podatke s većom točnošću nego model višestruke regresije.

8.3.3. NEURONSKA MREŽA

Svaka osnovna neuronska mreža sastoji se od barem tri sloja čvorova; ulaznog sloja, skrivenog sloja i izlaznog sloja. U ovom će se primjeru obrađivati neuronska mreža koja se sastoji od jednog ulaznog sloja, dva skrivena sloja i jednog izlaznog sloja. Detaljniji opis modela prikazan je u podnaslovu 4.6.

Kao i kod prethodna dva modela, nakon instaliranja i učitavanja potrebnih paketa, obrade i pripreme podataka te upoznavanja s podacima slijedi izrada modela.

```
1  #UČITAVANJE PAKETA- MODEL NEURONSKE MREŽE
2
3  library(caret)
4  library(neuralnet)
5  library(ggplot2)
6  library(readxl)
7  library(dplyr)
8  library(caTools)
9
10 #UČITAVANJE PODATAKA
11 data <- read_excel("C:/Users/marko/Desktop/Indeks_kvalitete_zraka.xlsx")
12
13 #SKALIRANJE I CENTRIRANJE PODATAKA
14 data <- scale(data, center = TRUE,scale = TRUE)
15 data <- data.frame(data)
```

Slika 8.3.3.1. Prikaz početka koda za model neuronske mreže.

Pomoću naredbi prikazanih na *Slika 8.3.3.1.* učitani su programski paketi potrebni za izradu modela neuronske mreže. Također, učitan je skup podataka koji se promatra danim algoritmom te je provedeno skaliranje i centriranje tih podataka. Prilikom izrade bilo kojeg od modela, potrebno je prije skaliranja i centriranja podataka, pobliže se upoznati s njima kako je prikazano u podnaslovu 8.2.

```
17 #IZRADA MODELA
18 set.seed(500)
19
20 split = sample.split(data$AQI, SplitRatio = 0.8)
21 train = subset(data, split == TRUE)
22 test = subset(data, split == FALSE)
23
24 nn <- neuralnet(AQI ~ .,
25                  data = train, hidden = c(5, 3),
26                  linear.output = TRUE,
27                  stepmax = 1e+05,
28                  threshold = 0.2)
29
30 set.seed(500)
```

Slika 8.3.3.1. Prikaz koda za model neuronske mreže.

Naredbom *set.seed(500)* kao što je već i navedeno u prethodna dva modela, jamči se da će kod proizvoditi svaki put iste nasumične vrijednosti kad god se on pokrene.

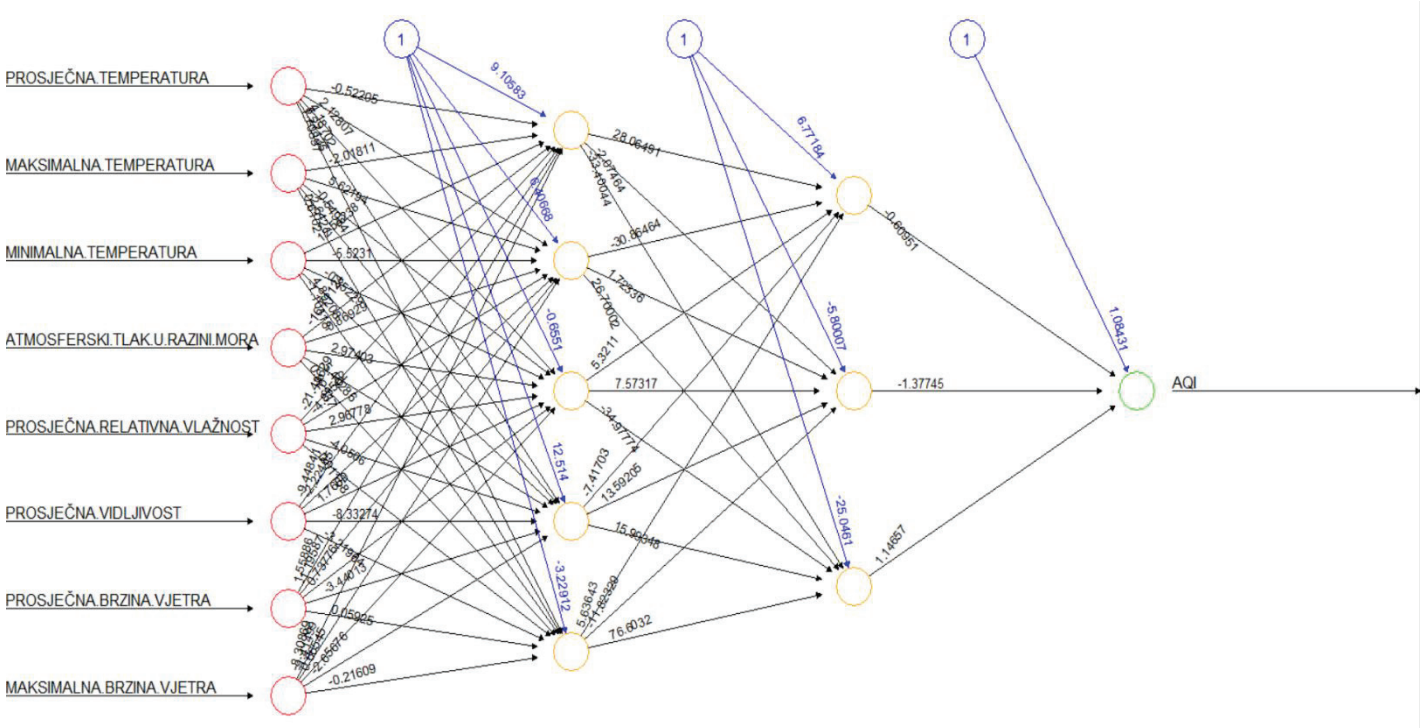
Sljedećim dijelim koda, skup podataka je podijeljen na dva dijela. Prvi dio podataka (80%) koristi se za treniranje modela- train podaci, dok se drugi dio podataka (20%) koristi za testiranje modela- test podaci.

Funkcija *hidden = c(5,3)* specificira broj skrivenih slojeva i broj neurona u svakom sloju. Dakle ova neuronska mreža sadrži dva skrivena sloja s četiri odnosno dva neurona.

Također, argument *linear.output = TRUE* koristi se za određivanje da li algoritam radi regresiju ili klasifikaciju (*linear.output = FALSE*).

stepmax = 1e+05 parametar je kojom se određuje broj koraka tijekom izvršavanja algoritma. Ponekad je potrebno više od sto tisuća koraka da bi algoritam konvergirao. U takvim slučajevima parametar *stepmax* se povećava dok algoritam ne konvergira. Drugim riječima, ovaj parametar određuje koliko dugo se neuronska mreža trenira.

threshold bio bi prag za podatke koji određuje kada model staje sa daljnjom optimizacijom.



Slika 8.3.3.2. Prikaz izračunate neuronske mreže.

Slika 8.3.3.2. prikazuje neuronsku mrežu za promatrani skup podataka. Dakle, ova neuronska mreža sadrži jedan ulazni sloj, dva skrivena sloja i to s četiri odnosno dva neurona te jedan izlazni sloj. S lijeve strane neuronske mreže prvo su prikazane sve ulazne varijable. Zatim je, crvenom bojom, prikazan ulazni sloj neurona. Skriveni slojevi neurona prikazani su narančastom bojom, sa četiri i dva neurona. Na samom kraju prikazan je izlazni sloj s jednom izlaznom varijablom, AQI, zelenom bojom. Crne strelice sa brojevima između slojeva predstavljaju veze između neurona zajedno s njihovim težinama, dok plave strelice s brojevima prikazuju pristranost (*eng. bias*).

```
32  #PRIKAZ NEURONSKE MREŽE
33  plot(nn , col.entry = "red", col.hidden = "orange", col.out = "green" )
```

Slika 8.3.3.3. Kod za prikaz neuronske mreže.

```
33  #IZDVAJANJE Y VRIJEDNOSTI DOBIVENIH IZ POČETNIH MJERNIH PODATAKA
34  AQI <- test$AQI
35
36  #BRISANJE Y VRIJEDNOSTI IZ TEST SKUPA PODATAKA
37  test <- test[, -which(names(test) == "AQI")]
38
39  #RAČUNANJE PREDIKCIJA - Y VRIJEDNOST - INDEKS KVALITETE ZRAKA
40  predictions <- neuralnet::compute(nn, test)
41
42  predictions <- predictions$net.result
43
44  # PRETVARANJE NAMED NUM U NUM PODATKE
45  predictions <- as.numeric(predictions)
46
47  #PRIKAZ Y VRIJEDNOSTI
48  AQI
49  predictions
50  head(AQI)
51  head(predictions)
```

Slika 8.3.3.4. Kodovi za predikciju indeksa kvalitete zraka, AQI.

```
> #PRIKAZ Y VRIJEDNOSTI
> AQI
  [1]  1.354568552  0.556358400  1.122761518  2.049533937
  [5] -0.003107723  0.884169395  0.291081631  0.335134082
  [9] -0.468949730  0.175633829  0.044995528 -0.377300379
 [13] -0.679591333  0.243991080  0.115378179 -1.315566942
 [17]  0.153354429  0.416150083  0.480962884  0.026260577
 [21] -0.371730529 -0.461860830 -0.915044086 -0.718580284
 [25] -0.811748685 -0.912512336 -0.850737635 -0.812761385
 [29] -0.927702837 -0.881624986 -0.308436778 -0.526673631
 [33] -1.315566942 -1.315566942 -0.579840382  0.636918686
 [37]  1.021744692  1.408596097  0.710339437  0.827812639
 [41]  0.945285841  3.600078929  0.975160491  0.686540987
 [45] -1.315566942  1.674429851 -0.106403125  1.341757896
 [49]  1.248589495 -0.626930932 -0.355527328 -0.247674777
 [53] -0.196533426 -0.642121432  0.151329029 -0.668451633
 [57] -0.523635531 -0.532243481 -0.336792378  0.049046328
 [61] -0.578827682 -0.329703478 -0.680604033 -0.107922175
 [65] -1.025428388 -0.737315234 -0.806685185 -0.609208682
 [69] -0.651742083 -0.680604033 -1.315566942 -0.759594634
 [73] -1.157585740  2.113435308  0.532104235 -0.200584226
 [77]  2.929165169  2.454715212  1.168079844  1.494675599
 [81]  1.285553046  1.266311745  1.147319494  0.625272636
 [85]  1.416191347 -0.713010433 -0.891245636 -0.229446176
 [89] -0.691237383 -0.169190526 -0.730732684 -0.371224179
 [93] -0.541864131 -0.549965731 -0.418314729 -0.919601236
 [97] -0.968717187 -0.821875685 -0.942386987 -1.166700040
```

Slika 8.3.3.5. Prikaz y vrijednosti-indeks kvalitete zraka dobivenih iz početnih mjernih podataka (prvih 100 vrijednosti).

```
> predictions
  [1]  0.75353398  1.54097597  2.22927699  1.66147428
  [5]  0.19568010  1.03995558  1.27347068 -0.78605847
  [9] -0.62195353 -0.33245766 -0.20260400 -0.19912086
 [13]  0.21449731 -0.07286367 -0.24243204 -0.54644693
 [17]  0.01318577 -0.31956105 -0.31241591 -0.31543271
 [21] -0.58065210 -0.32125897 -0.59843151 -0.90264204
 [25] -0.70722752 -0.90211084 -0.37493291 -0.60381664
 [29] -0.90262367 -0.90263543 -0.42525669 -0.58111869
 [33] -0.80824867 -0.74686564 -0.84183845  0.73262024
 [37]  1.72689714  1.14861280  0.85685723  1.08845699
 [41]  0.68906624  2.07910687  1.63194323  0.98844008
 [45]  0.36211964  1.98697113  0.10412373  1.94366972
 [49]  0.93878969 -0.58359726 -0.41634036 -0.32267236
 [53] -0.24886962 -0.33083163  0.78776655 -0.39518173
 [57] -0.34407492 -0.87167734 -0.40208594 -0.33172753
 [61] -0.33089738 -0.36787614 -0.85953221 -0.33583217
 [65] -0.90264574 -0.40065119 -0.67588167 -0.59221702
 [69] -0.59314789 -0.69984282 -0.67799294  0.45020645
 [73] -0.29312793  1.40739575  0.76159334  0.85367656
 [77]  2.14927218  1.93403851  1.62136780  1.54619076
 [81]  1.99116270  1.04228340  1.01139455  0.32658112
 [85]  0.87621545 -0.59038996 -0.65114769 -0.29244403
 [89] -0.28847913 -0.32735227 -0.51800248 -0.32367302
 [93] -0.31294876 -0.32579238 -0.39186636 -0.38233859
 [97] -0.34066440 -0.48420501 -0.88817136 -0.90264599
```

Slika 8.3.3.6. Prikaz y vrijednosti-indeks kvalitete zraka dobivenih predikcijom pomoću modela neuronske mreže (prvih 100 vrijednosti).

```
> head(AQI)
[1]  1.354568552  0.556358400  1.122761518  2.049533937
[5] -0.003107723  0.884169395
> head(predictions)
[1] 0.7535340 1.5409760 2.2292770 1.6614743 0.1956801
[6] 1.0399556
```

Slika 8.3.3.7. Usporedba indeksa kvalitete zraka dobivenog iz početnih mjernih podataka- AQI te predikcijom pomoću modela višestruke linearne regresije- predictions.

Ako se usporedi prvih nekoliko vrijednosti indeksa kvalitete zraka dobivenih iz početnih mjernih podataka- AQI te predikcijom pomoću modela slučajne šume- predictions uočljivo je da vrijednosti većinom odstupaju jedna od druge.

```
54  #ODREĐIVANJE KORIJENA SREDNJE KVADRATNE POGREŠKE
55  RMSE(predictions, AQI)
56
57  #GRAFIČKI PRIKAZ
58  plot(predictions, AQI, col="blue")
59  abline(0,1, col("red")
```

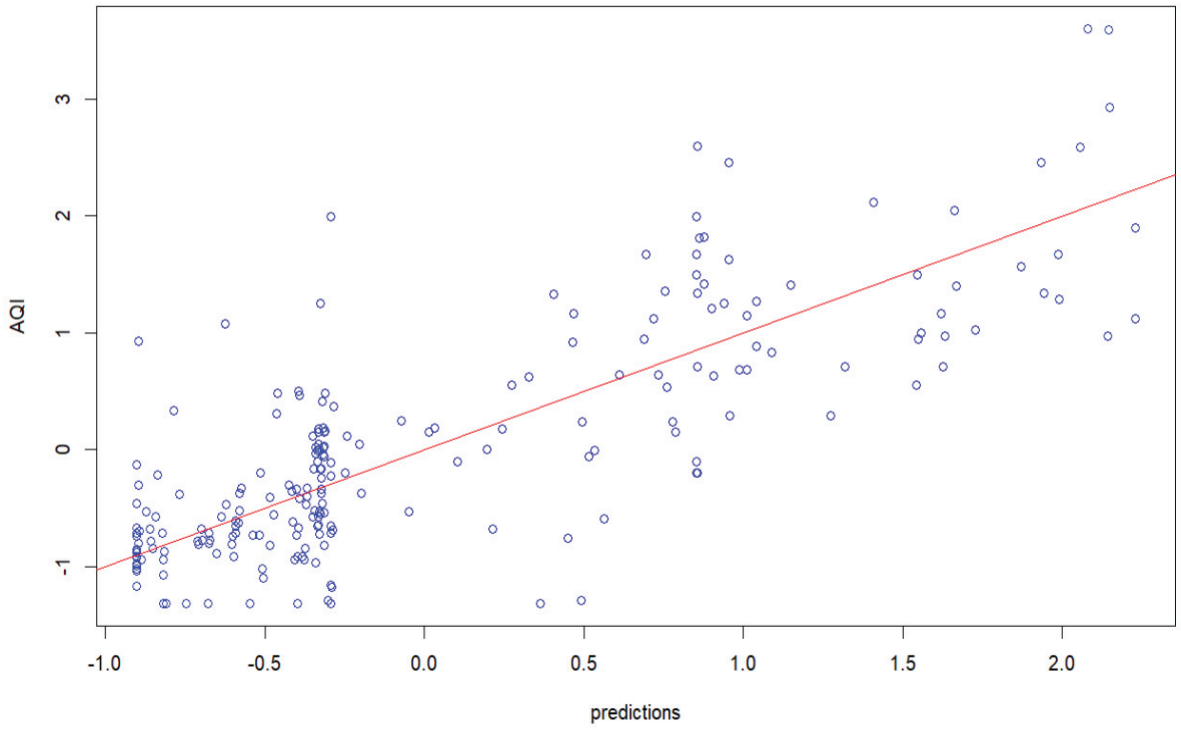
Slika 8.3.3.8. Kodovi za određivanje RMSE te grafički prikaz rezultata.

Korijen srednje kvadratne pogreške, RMSE mjeri prosječnu razliku između predviđenih vrijednosti modela i stvarnih vrijednosti. Kako se podatkovne točke približavaju regresijskoj liniji, model ima manje pogreške, a RMSE se smanjuje.

```
> #ODREĐIVANJE KORIJENA SREDNJE KVADRATNE POGREŠKE
> RMSE(predictions, AQI)
[1] 0.6102752
```

Slika 8.3.3.9. Prikaz rezultata RMSE.

RMSE iznosi 0,6102752.

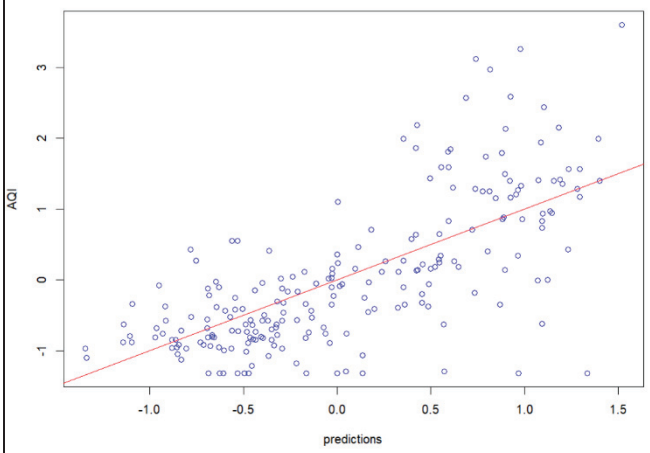


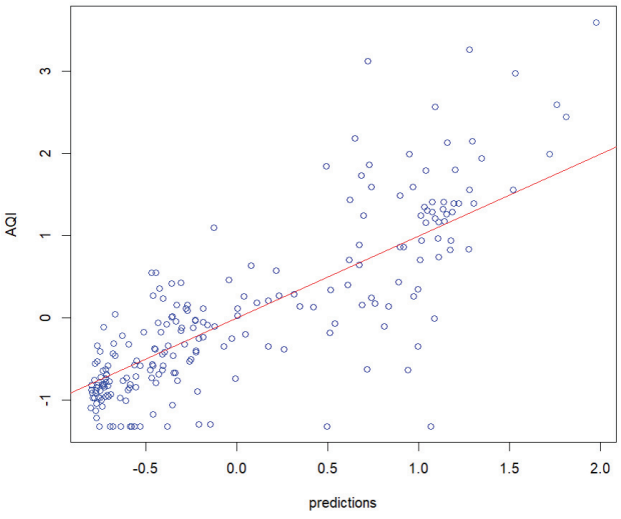
Slika 8.3.3.10. Grafički prikaz ovisnosti stvarnih i predviđenih vrijednosti.

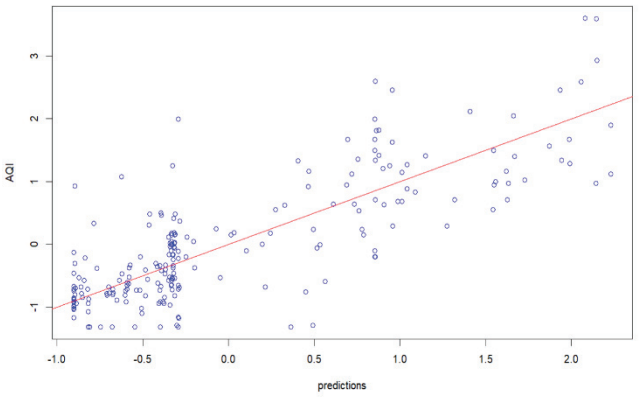
Iz grafičkog prikaza na *Slika 8.3.3.10.* vidljivo je da se stvarne i predviđene vrijednosti ne podudaraju u potpunosti. Drugim riječima, uočljivo je da model slučajne šume ne predviđa podatke sa 100%-tnom točnošću. Upravo to potvrđuje pogreška RMSE koja iznosi 0,6102752.

8.4. REZULTATI

Radi bolje preglednosti i lakše usporedbe rezultata, svi su rezultati koji prikazuju primjenu algoritama strojnog učenja na realan sustav prikazani u tri tablice. Indeks kvalitete zraka dobiven iz početnih mjernih podataka označen je s AQI, dok je indeks kvalitete zraka dobiven predikcijom označen s PREDICTIONS. Uspoređuje se prvih deset mjerenih vrijednosti indeksa kvalitete zraka s prvih deset predviđenih vrijednosti.

MODEL	AQI	PREDICTIONS	RMSE	GRAFIČKI PRIKAZ
VIŠESTRUKA LINEARNA REGRESIJA	1,396950047 2,150601398 1,562425230 0,861130469 -0,063869724 -0,345400328 0,113352779 0,187786230 0,115378179 -1,315566942	1,4001875761 1,1827243076 1,2367143763 0,9884288116 0,4879720517 0,8679981741 0,3284431091 0,5233314772 0,2391818024 -0,1652663714	0,7279787	

MODEL	AQI	PREDICTIONS	RMSE	GRAFIČKI PRIKAZ
SLUČAJNA ŠUMA	1,396950047 2,150601398 1,562425230 0,861130469 -0,063869724 -0,345400328 0,113352779 0,187786230 0,115378179 -1,315566942	1,302109449 1,293372525 1,517845486 0,916888661 0,540111231 0,994813119 0,004750111 0,112558477 -0,182587340 -0,582734272	0,596345	

MODEL	AQI	PREDICTIONS	RMSE	GRAFIČKI PRIKAZ
NEURONSKA MREŽA	1,396950047 2,150601398 1,562425230 0,861130469 -0,063869724 -0,345400328 0,113352779 0,187786230 0,115378179 -1,315566942	0,75353398 1,54097597 2,22927699 1,66147128 0,19568010 1,03995558 1,27347068 -0,78605847 -0,62195353 -0,33245766	0,6102752	

8.5. ZAKLJUČAK

Zrak može sadržavati čestice koje uzrokuju onečišćenja, poput čađe, dima, ispušnih plinova automobila i slično. Naravno, onečišćenje zraka razlikuje se ovisno o geografskom položaju, količini onečišćujućih tvari, vrsti onečišćujućih tvari i drugom. Onečišćenje zraka osim što uzrokuje zdravstvene probleme, stvara veliki problem i u inženjerskoj industriji. Dakle, poprilično je teško jednom inženjeru materijala izumiti materijal koji smanjuje količinu ugljika u atmosferi, a time prilikom proizvodnje samog materijala ne proizvesti dodatni ugljikov dioksid. Drugim riječima, teško je pronaći balans između proizvodnje materijala i ne uzrokovanja dodatnog zagađenja zraka.

Zahvaljujući strojnom učenju inženjeri materijala u mogućnosti su na brz i jednostavan način odrediti indeks kvalitete zraka prilikom procjene kvalitete novog materijala. Kada se računalo jednom nauči kako određeni parametri utječu na indeks kvalitete zraka, ono je pogodno za primjenu za bilo koji novi skup podataka. Dakle, inženjeri materijala prilikom izuma novog materijala za smanjenje ugljika u atmosferi te puštanja materijala u upotrebu, u mogućnosti su primjenom već naučenog računala na brz način predvidjeti nove indekse kvalitete zraka i dobiti uvid koliko je materijal zapravo smanjio onečišćenje zraka.

U ovom je radu primjenom algoritama strojnog učenja i programskog jezika R pronađen model koji najbolje predviđa indeks kvalitete zraka te su uspoređene mjerne vrijednosti indeksa kvalitete zraka s vrijednostima dobivenim predikcijom. Podaci korišteni u ovom radu preuzeti su sa online platforme pod imenom Kaggle. Algoritmi strojnog učenja korišteni za obradu podataka i provođenje predikcije jesu algoritam višestruke linearne regresije, algoritam slučajne šume te algoritam neuronskih mreža.

U rezultatima prikazani su grafički prikazi ovisnosti stvarnih (AQI) i predviđenih vrijednosti (predictions) indeksa kvalitete zraka. Za model višestruke linearne regresije, vidljivo je da se stvarne i predviđene vrijednosti ne podudaraju u potpunosti. Drugim riječima, uočljivo je da model višestruke linearne regresije ne predviđa podatke sa 100%-tnom točnošću. Upravo to i potvrđuje pogreška RMSE, koja iznosi 0,7279787. Za model slučajne šume, na grafičkom je prikazu vidljivo da model ne predviđa podatke sa 100%-tnom točnošću, što potvrđuje pogreška RMSE koja iznosi 0,596345. Osim toga moguće je uočiti da model slučajne šume predviđa podatke s većom točnošću nego model višestruke linearne regresije. Grafički prikaz modela neuronskih mreža također prikazuje ovisnost stvarnih i predviđenih vrijednosti indeksa kvalitete zraka koji se ne podudaraju u potpunosti. Ovaj model ima pogrešku RMSE 0,6102752. Model s najmanjom pogreškom RMSE, odnosno pogreškom bližom 0, ali i onaj kojem su podaci u grafičkom prikazu bliži regresijskoj liniji zapravo predstavlja najbolji model za opis ponašanja ovih podataka. Dakle, iz navedenog da se zaključiti da je model koji najbolje predviđa indeks kvalitete zraka za dani skup podataka zapravo model slučajne šume.

Uspoređujući prvih deset mjerenih vrijednosti indeksa kvalitete zraka s prvih deset predviđenih vrijednosti za sva tri modela, potvrđen je gornji navod. Vidljivo je da su prvih deset mjernih vrijednosti najbolje predviđene metodom slučajne šume, dok je metoda višestruke linearne regresije, metoda koja najlošije opisuje podatke. Ta metoda ima najveću pogrešku i njeni podaci su najudaljeniji odnosno najviše dispergirani oko regresijske linije.

9. LITERATURA

1) Prof.dr.sc. Nenad Bolf, Strojno učenje. URL: https://www.fkit.unizg.hr/_download/repository/MUI_21-22_Strojno_ucenje.pdf (pristup: 1.4.2023.)
2) Machine Learning. URL: https://www.sas.com/en_us/insights/analytics/machine-learning.html (pristup: 5.4.2023.)
3) Machine Learning, explained. URL: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (pristup: 10.4.2023.)
4) Ed Burns, Machine Learning. URL: https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML?vgnextfmt=print (pristup: 11.4.2023.)
5) What Is Machine Learning? Definition, Typesm Applications and Trends for 2022. URL: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/ (pristup: 11.4.2023.)
6) Ed Burns, Machine Learning. URL: https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML (pristup: 17.4.2023.)
7) History of Machine Learning- A Journey through the Timeline. URL: https://www.clickworker.com/customer-blog/history-of-machine-learning/ (pristup: 24.4.2023.)
8) Prof.dr.sc. Nenad Bolf, Strojno učenje. URL: https://hrcak.srce.hr/file/382926 (pristup: 24.4.2023.)
9) Deep Blue vs Garry Kasparov. URL: https://www.lanacion.com.ar/lifestyle/hace-24-anos-computadora-deep-blue-le-nid2332356/ (pristup: 24.4.2023.)
10) What are Machine Learning Applications? Top 10 Industry and Real-World Use Cases. URL: https://emeritus.org/blog/machine-learning-what-are-machine-learning-applications/ (pristup: 24.4.2023.)
11) Siri Logo. URL: https://www.pngwing.com/en/search?q=siri (pristup: 24.4.2023.)
12) Google Maps Logo. URL: https://logowik.com/google-maps-vector-logo-3251.html (pristup: 24.4.2023.)
13) Internet. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2021. URL: https://www.enciklopedija.hr/natuknica.aspx?ID=27653 (pristup: 25.4.2023.)
14) Product Recommendation Examples. URL: https://www.nosto.com/blog/product-recommendations-examples/ (pristup: 1.8.2023.)
15) Top 10 Machine Learning Applications and Examples in 2023. URL: https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-applications (pristup: 1.8.2023.)
16) Machine Learning Algorithms. URL: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html (pristup: 1.8.2023.)
17) Top 10 Machine Learning Algorithms. URL: https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/ (pristup: 2.8.2023.)

18) Dianne Castillo, Machine Learning Regression Explained. URL: https://www.seldon.io/machine-learning-regression-explained (pristup: 2.8.2023.)
19) 10 Machine Learning Algorithms to Know in 2023. URL: https://www.coursera.org/articles/machine-learning-algorithms (pristup: 2.8.2023.)
20) Mohit Gupta, Linear Regression in Machine learning. URL: https://www.geeksforgeeks.org/ml-linear-regression/ (pristup: 2.8.2023.)
21) Joey Hejna, Linear Regression. URL: http://joeyhejna.com/mlbook/content/03/intro.html (pristup: 2.8.2023.)
22) Simon Tavasoli, Top 10 Machine Learning Algorithms For Beginners: Supervised, and More. URL: https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article (pristup: 2.8.2023.)
23) What is Multiple Linear Regression in Machine Learning?. URL: https://www.simplilearn.com/what-is-multiple-linear-regression-in-machine-learning-article (pristup: 3.8.2023.)
24) Logistic Regression in Machine Learning. URL: https://www.geeksforgeeks.org/understanding-logistic-regression/ (pristup: 7.8.2023.)
25) Andriy Burkov, The Hundred-Page Machine Learning Book, 2019., (pristup: 7.8.2023.), str. 25-27
26) Peter Karas, Logistic Regression in Depth. URL: https://ai.plainenglish.io/logistic-regression-543c8424595d (pristup: 9.8.2023.)
27) Top 10 Machine Learning Algorithms In 2023. URL: https://www.synergisticit.com/machine-learning-algorithms/ (pristup: 9.8.2023.)
28) Nagesh Singh Chauhan, Decision Tree Algorithm, Explained. URL: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html (pristup: 9.2.2022.) (pristup: 10.8.2023.)
29) Pateli, H., Prajapati, P. Listopad 2018. Study and Analysis of Decision Tree Based Classification Algorithms. ResearchGate. URL: https://www.researchgate.net/profile/Purvi-Prajapati/publication/330138092_Study_and_Analysis_of_Decision_Tree_Based_Classification_Algorithms/links/5d2c4a91458515c11c3166b3/Study-and-Analysis-of-Decision-Tree-Based-Classification-Algorithms.pdf (pristup: 10.8.2023.)
30) Kajal Kiran, Decision Tree. URL: https://www.linkedin.com/pulse/decision-tree-kajal-kiran (pristup: 10.8.2023.)
31) Prashant Gupta, Decision Trees in Machine Learning. URL: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 (pristup: 10.8.2023.)
32) Understand Random Forest Algorithms With Examples. URL: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/ (pristup: 14.8.2023.)
33) Abihishek Sharma, Random Forest vs Decision Tree. URL: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/ (pristup: 14.8.2023.)
34) Random Forest Algorithm. URL: https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm (pristup: 14.8.2023.)
35) Tree Based Algorithms: A Complete Tutorial from Scratch (in R&Python). https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/ (pristup: 14.8.2023.)
36) What are neural networks? URL: https://www.ibm.com/topics/neural-networks (pristup: 15.8.2023.)

37) A Beginner's Guide to Neural Networks and Deep Learning. URL: https://wiki.pathmind.com/neural-network (pristup: 15.8.2023.)
38) Introduction to Neural Network in Machine Learning. URL: https://www.analyticsvidhya.com/blog/2022/01/introduction-to-neural-networks/ (pristup: 15.8.2023.)
39) Top 10 Deep Learning Algorithms You Should Know in 2023. URL: https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm (pristup: 15.8.2023.)
40) Machine Learning Steps: A Complete Guide. URL: https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps (pristup: 17.8.2023.)
41) R programming language. URL: https://www.r-project.org/about.html (pristup: 17.8.2023.)
42) What is GNU? URL: https://www.gnu.org/ (pristup: 17.8.2023.)
43) R Programming Languade- Introduction. URL: https://www.geeksforgeeks.org/r-programming-language-introduction/ (pristup: 17.8.2023.)
44) R Programming Language Logo Vector. URL: https://vectorseek.com/vector_logo/r-programming-language-logo-vector/ (pristup: 17.8.2023.)
45) NASA, 10 interesting things about air. URL: https://climate.nasa.gov/news/2491/10-interesting-things-about-air/ (pristup: 21.8.2023.)
46) The Chemical Composition of Air. URL: https://www.thoughtco.com/chemical-composition-of-air-604288 (pristup: 21.8.2023.)
47) What is the air quality indeks (AQI)? URL: https://www.iqair.com/newsroom/what-is-aqi (pristup: 21.8.2023.)
48) Aniruddha Pal, Air Quality Indeks. URL: https://www.kaggle.com/datasets/aniruddhapa/air-quality-index (pristup: 21.8.2023.)

10. PRILOG

- **PROGRAMSKI JEZIK R**
  URL: https://posit.co/download/rstudio-desktop/

- INSTALIRANJE PAKETA
  install.packages(„caret“)
  install.packages(„ggplot2“)
  install.packages(„readxl“)
  install.packages(„caTools“)
  install.packages(„dplyr“)
  install.packages(„randomForest“)
  install.packages(„neuralnet“)

- UČITAVANJE PAKETA ZA SVE MODELE
  library(caret)
  library(ggplot2)
  library(readxl)
  library(caTools)
  library(dplyr)
  library(randomForest)
  library(neuralnet)

- UČITAVANJE PODATAKA
  data <- read_excel(„C:/Users/marko/Desktop/Indeks_kvalitete_zraka.xlsx“)

- UVID U SET PODATAKA
  head(data)

- UVID U STATISTIČKU DISTRIBUCIJU PODATAKA
  summary(data)

- PROVJERA VRIJEDNOSTI KOJE NEDOSTAJU
  sum(is.na(data))

- PROVJERA DUPLICIRANIH VRIJEDNOSTI
  table(duplicated(data))

- ODREĐIVANJE DIMENZIJE SKUPA PODATAKA

```
dim(data)
```

- ODREĐIVANJE X I Y OSI ZA CRTANJE GRAFIČKIH PRIKAZA

```
x1 <- data$`PROSJEČNA TEMPERATURA`
x2 <- data$`MAKSIMALNA TEMPERATURA`
x3 <- data$`MINIMALNA TEMPERATURA`
x4 <- data$`ATMOSFERSKI TLAK U RAZINI MORA`
x5 <- data$`PROSJEČNA RELATIVNA VLAŽNOST`
x6 <- data$`PROSJEČNA VIDLJIVOST`
x7 <- data$`PROSJEČNA BRZINA VJETRA`
x8 <- data$`MAKSIMALNA BRZINA VJETRA`
y <- data$AQI
```

- CRTANJE GRAFIČKIH PRIKAZA

```
plot(x1, y, col="red")
plot(x2, y, col="blue")
plot(x3, y, col="green")
plot(x4, y, col="yellow")
plot(x5, y, col="purple")
plot(x6, y, col="orange")
plot(x7, y, col="pink")
plot(x8, y, col="navyblue")
```

- SKALIRANJE I CENTRIRANJE PODATAKA

```
data <- scale(data, center = TRUE, scale = TRUE)
data <- dana.frame(data)
```

- **MODEL VIŠESTRUKE LINEARNE REGRESIJE**

```
#UČITAVANJE PAKETA-MODEL VIŠESTRUKE LINEARNE REGRESIJE
library(caret)
library(ggplot2)
library(readxl)
library(dplyr)
library(caTools)
#UČITAVANJE PODATAKA
data <- read_excel("C:/Users/marko/Desktop/Indeks_kvalitete_zraka.xlsx")
#SKALIRANJE I CENTRIRANJE PODATAKA
data <- scale(data, center = TRUE, scale = TRUE)
data <- data.frame(data)
#IZRADA MODELA
set.seed(1)
split = sample.split(data$AQI, SplitRatio = 0.8)
train = subset(data, split == TRUE)
test = subset(data, split == FALSE)
model <- lm(AQI ~ . , data = train)
summary(model)
#IZDVAJANJE Y VRIJEDNOSTI DOBIVENIH IZ POČETNIH MJERNIH PODATAKA
AQI <- test$AQI
#BRISANJE Y VRIJEDNOSTI IZ TEST SKUPA PODATAKA
test <- test[, -which(names(test) == "AQI")]
#RAČUNANJE PREDIKCIJE-Y VRIJEDNOST- INDEKSA KVALITETE ZRAKA
predictions <- predict(model, test)
#PRETVARANJE NAMED NUM U NUM PODATKE
names(predictions) <- NULL
#PRIKAZ Y VRIJEDNOSTI
AQI
```

```
predictions
head(AQI)
head(predictions)
#ODREĐIVANJE KORIJENA SREDNJE KVADRATNE POGREŠKE
RMSE(predictions,AQI)
#GRAFIČKI PRIKAZ
plot (predictions,AQI, col="blue")
abline(0,1, col="red")
```

- **MODEL SLUČAJNE ŠUME**

```
#UČITAVANJE PAKETA-MODEL VIŠESTRUKE LINEARNE REGRESIJE
library(caret)
library(ggplot2)
library(readxl)
library(dplyr)
library(caTools)
#UČITAVANJE PODATAKA
data <- read_excel("C:/Users/marko/Desktop/Indeks_kvalitete_zraka.xlsx")
#SKALIRANJE I CENTRIRANJE PODATAKA
data <- scale(data, center = TRUE, scale = TRUE)
data <- data.frame(data)
#IZRADA MODELA
set.seed(1)
split = sample.split(data$AQI, SplitRatio = 0.8)
train = subset(data, split == TRUE)
test = subset(data, split == FALSE)
trC <- trainControl(method = "cv",
                    number = 10,
```

```
                search = "grid")

tuneGrid <- expand.grid(.mtry = c(1: 10))
rf_mtry <- train(AQI ~ .,
                 data = data,
                 method = "rf",
                 tuneGrid = tuneGrid,
                 trControl = trC,
                 importance = TRUE,
                 nodesize = 14,
                 ntree = 300)
best_mtry <- rf_mtry$bestTune$mtry
tuneGrid <- expand.grid(.mtry = best_mtry)
fit_rf <- train(AQI ~ .,
                data,
                method = "rf",
                tuneGrid = tuneGrid,
                trControl = trC,
                importance = TRUE,
                nodesize = 14,
                ntree = 800,
                maxnodes = 24)
print(fit_rf)
#VAŽNOST VARIJABLI
varImp(fit_rf)
#IZDVAJANJE Y VRIJEDNOSTI DOBIVENIH IZ POČETNIH MJERNIH PODATAKA
AQI <- test$AQI
#BRISANJE Y VRIJEDNOSTI IZ TEST SKUPA PODATAKA
test <- test[, -which(names(test) == "AQI")]
```

```
#RAČUNANJE PREDIKCIJE-Y VRIJEDNOST- INDEKSA KVALITETE ZRAKA
predictions <- predict(fit_rf, test)
#PRETVARANJE NAMED NUM U NUM PODATKE
names(predictions) <- NULL

#PRIKAZ Y VRIJEDNOSTI
AQI
predictions
head(AQI)
head(predictions)
#ODREĐIVANJE KORIJENA SREDNJE KVADRATNE POGREŠKE
RMSE(predictions,AQI)
#GRAFIČKI PRIKAZ
plot (predictions,AQI, col="blue")
abline(0,1, col="red")
```

- **MODEL NEURONSKE MREŽE**

```
#UČITAVANJE PAKETA-MODEL VIŠESTRUKE LINEARNE REGRESIJE
library(caret)
library(ggplot2)
library(readxl)
library(dplyr)
library(caTools)
#UČITAVANJE PODATAKA
data <- read_excel("C:/Users/marko/Desktop/Indeks_kvalitete_zraka.xlsx")
#SKALIRANJE I CENTRIRANJE PODATAKA
data <- scale(data, center = TRUE, scale = TRUE)
data <- data.frame(data)
```

```
#IZRADA MODELA
set.seed(500)
split = sample.split(data$AQI, SplitRatio = 0.8)
train = subset(data, split == TRUE)
test = subset(data, split == FALSE)

nn <- neuralnet(AQI ~ .,
                data = train, hidden = c(5, 3),
                linear.output = TRUE,
                stepmax = 1e+05,
                threshold = 0.2)

set.seed(500)

#PRIKAZ NEURONSKE MREŽE
plot(nn, col.entry="red", col.hidden= „orange“, col.out= „green“)
#IZDVAJANJE Y VRIJEDNOSTI DOBIVENIH IZ POČETNIH MJERNIH PODATAKA
AQI <- test$AQI
#BRISANJE Y VRIJEDNOSTI IZ TEST SKUPA PODATAKA
test <- test[, -which(names(test) == "AQI")]
#RAČUNANJE PREDIKCIJE-Y VRIJEDNOST- INDEKSA KVALITETE ZRAKA
predictions <- neuralnet::compute(nn, test)
predictions <- predictions$net.result
#PRETVARANJE NAMED NUM U NUM PODATKE
Predictions <- as.numeric(predictions)
#PRIKAZ Y VRIJEDNOSTI
AQI
predictions
head(AQI)
head(predictions)
```

```
#ODREĐIVANJE KORIJENA SREDNJE KVADRATNE POGREŠKE
RMSE(predictions,AQI)
#GRAFIČKI PRIKAZ
plot (predictions,AQI, col="blue")
abline(0,1, col="red")
```