Umjetne neuronske mreže u kemijskom inženjerstvu

Krhlanko, Karlo

SVEUČILIŠTE U ZAGREBU

FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE

SVEUČILIŠNI PRIJEDIPLOMSKI STUDIJ

Karlo Krhlanko

ZAVRŠNI RAD

Zagreb, rujan 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE
POVJERENSTVO ZA ZAVRŠNE ISPITE

Kandidat **Karlo Krhlanko**

Predao je izrađen završni rad dana: 18. rujna 2024.

Povjerenstvo u sastavu:

doc. dr. sc. Miroslav Jerković, Sveučilište u Zagrebu Fakultet kemijskog inženjerstva i tehnologije
doc. dr. sc. Matija Cvetnić, Sveučilište u Zagrebu Fakultet kemijskog inženjerstva i tehnologije
doc. dr. sc. Dragana Vuk, Sveučilište u Zagrebu Fakultet kemijskog inženjerstva i tehnologije
izv. prof. dr. sc. Erna Begović Kovač, Sveučilište u Zagrebu Fakultet kemijskog inženjerstva i tehnologije (zamjena)

povoljno je ocijenilo završni rad i odobrilo obranu završnog rada pred povjerenstvom u istom sastavu.

Završni ispit održat će se dana: 23. rujna 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE

SVEUČILIŠNI PRIJEDIPLOMSKI STUDIJ

KEMIJSKO INŽENJERSTVO

Karlo Krhlanko

UMJETNE NEURONSKE MREŽE U KEMIJSKOM INŽENJERSTVU

ZAVRŠNI RAD

Voditelj rada: doc. dr. sc. Miroslav Jerković

Članovi ispitnog povjerenstva: doc. dr. sc. Miroslav Jerković

doc. dr. sc. Matija Cvetnić

doc. dr. sc. Dragana Vuk

Zagreb, rujan 2024.

Završni rad izrađen je na Zavodu za matematiku Fakulteta kemijskog inženjerstva i tehnologije Sveučilišta u Zagrebu

Zahvaljujem svom mentoru doc. dr. sc. Miroslavu Jerkoviću na ukazanoj prilici, strpljenju i korisnim uputama prilikom izrade ovoga rada. Zahvaljujem svojim prijateljima, a posebno svojim roditeljima na svoj podršci i razumijevanju tijekom cijelog mojeg školovanja.

SAŽETAK

Topljivost različitih organskih spojeva u vodi je važna za razvoj lijekova u farmaceutskoj i novih kemijskih spojeva u kemijskoj industriji. Pomoću umjetnih neuronskih mreža možemo predvidjeti kako će različiti organski spojevi biti topljivi u vodi na temelju njihovih karakteristika i na taj način odlučiti kakva vrsta spoja je potrebna za projekte. Umjetne neuronske mreže djeluju na sličan način kao ljudski mozak. Umjetne neuronske mreže u kemijskom inženjerstvu se koriste za predviđanje rezultata nepoznatih vrijednosti nakon procesa treniranja. U ovom radu fokus je na primjeni umjetnih neuronskih mreža za predviđanje topljivosti organskih spojeva u vodi.

SUMMARY

Solubility of various organic compounds in water is very important for drug development in pharmaceutical and the creation of new chemical compounds in the chemical industry. Using artificial neural networks, we can anticipate how the solubility of different organic compounds will be based on the characteristics of said compounds and which type of compound is needed for specific projects. Artificial neural networks work on a similar principle to the human brain, enabling the prediction of unknown values after the training process. This paper focuses on the use of artificial neural networks to predict the solubility of organic compounds in water.

SADRŽAJ

1. UVOD

Topljivost organskih spojeva u vodi jedna je od najvažnijih kemijskih karakteristika te ima značajan utjecaj na razna područja kemije, farmacije i biologije. Topljivost se definira kao sposobnost spoja da se otopi u vodi i stvori homogen sustav. Na topljivost utječu brojni faktori, uključujući polaritet molekule, veličinu molekule, prisutnost funkcionalnih skupina i pH vrijednost otopine. U kemijskoj industriji, loša topljivost u vodi može dovesti do većih troškova razvoja i smanjene efikasnosti procesa. Iz tog se razloga dugi niz godina razvijaju modeli koji mogu predvidjeti topljivost spojeva u vodi kako bi se unaprijedile industrijske prakse i optimizirali procesi. Ovaj rad koristi podatke iz skupa podataka AqsolDB preuzetih sa stranice Kaggle, kako bi se usporedio i prikazao način rada i rezultati algoritama umjetnih neuronskih mreža.

Cilj ovog rada je istražiti učinkovitost primjene umjetnih neuronskih mreža za predviđanje topljivosti organskih spojeva u vodi. Kroz analizu podataka o kemijskim spojevima, modeli umjetnih neuronskih mreža bit će korišteni za identifikaciju ključnih karakteristika koje utječu na topljivost te za izradu prediktivnog modela. Osim umjetnih neuronskih mreža, rezultati će se usporediti s drugim algoritmima strojnog učenja, poput višestruke linearne regresije i slučajne šume, kako bi se utvrdilo koja metoda daje najpreciznije rezultate.

2.TOPLJIVOST ORGANSKIH SPOJEVA

Topljivost organskih spojeva u vodi je jedna je od najbitnijih kemijskih karakteristika koja ima značajan utjecaj na različita područja kemije, farmacije i biologije. Topljivost je sposobnost spoja da se otopi u vodi kako bi se stvorio homogen sustav. Topljivost spoja ovisi o mnogim faktorima, a najbitniji faktori su: polaritet molekule, veličina molekule, prisutnost funkcionalnih skupina, pH vrijednost otopine. [1]

Voda je polarna molekula te se iz tog razloga drugi polarni organski spojevi poput alkohola i karboksilnih kiselina dobro otapaju. Nepolarni spojevi poput ugljikovodika ne mogu stvarati vodikove veze i ne mogu se otapati. Manje molekule imaju bolju topljivost od većih jer se mogu integrirati s molekulama vode. Veće molekule s dugim ugljikovim lancima imaju manju topljivost zbog hidrofobnosti. Različite funkcionalne skupine drukčije djeluju na topljivost u vodi, skupine poput hidroksilnih (-OH), karboksilnih (-COOH) i aminokiselinskih (-NH2) povećaju polarnost spoja i time olakšavaju stvaranje vodikovih veza i povećavaju topljivost. Utjecaj pH na topljivost u vodi ovisi o vrsti tvari koja se otapa. Kiselim spojevima poput karboksilnih kiselina više odgovara pH viši od njihove pKa jer onda lakše stvaraju anionske oblike s vodom i bolje se otapaju. Bazičnim spojevima poput amina više odgovara pH koji je niži od njihove pKa, jer se u tom slučaju stvara kationski oblik koji je topljiviji u vodi. pKa je broj koji pokazuje koliko je jaka ili slaba određena kiselina. Kiselina je jača i bolje donira elektrone u slučaju da je pKa manji, dok je kod baza bolje da je pKa veći jer onda lakše prima elektrone. [2]

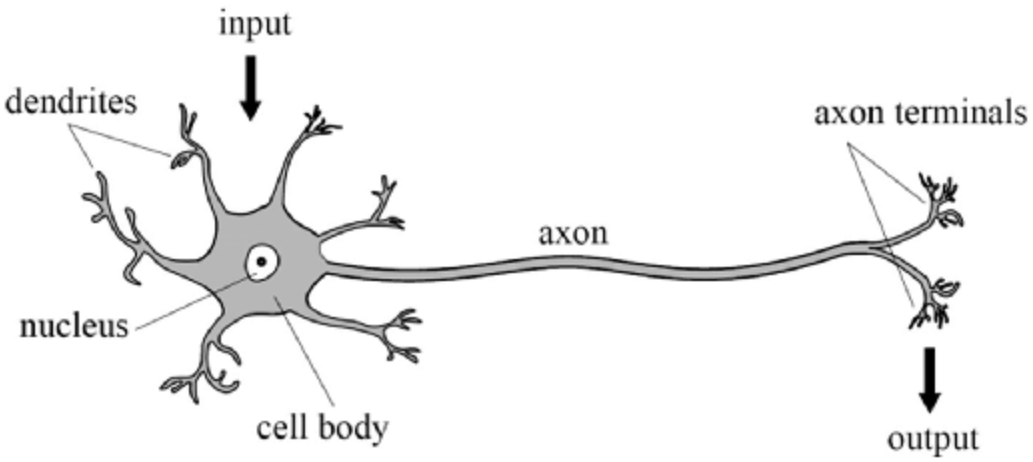
U ovom radu cilj je predvidjeti topljivost organskih spojeva u vodi, jer je voda jedno od ključnih otapala u industriji i životu. Topljivost u vodi ili vodena topljivost se definira kao maksimalna količina spoja, tj. otopljene tvari, koja se može otopiti u određenom volumenu vode i ovisi o fizičkim uvjetima kao što su temperatura i tlak. U kemijskoj industriji loša vodena topljivost može dovesti do loših rezultata, većih troškova razvoja tijekom razvoja. Zbog važnosti predviđanja topljivosti u vodi se dugi niz godina pokušavaju napraviti modeli koji će na temelju podataka previdjeti topljivost u vodi. [3]

Skup podataka o parametrima koji utječu na topljivost tvari naziva se AqsolDB. Skup je preuzet sa Kaggle stranice, i koristeći njega cilj je usporediti rad i rezultate različitih algoritama umjetnih neuronskih mreža.[4]

3.UMJETNE NEURONSKE MREŽE

Umjetne neuronske mreže (engl. Artificial neural networks, ANN) su analitički i računalni modeli koji modeliraju odnos ulaznih signala (engl. input) i izlaznog signala (engl. output). One djeluju na istom principu na koji ljudski mozak prihvaća razne podražaje i sukladno njima reagira. U ljudskom mozgu postoji mreža međusobno povezanih stanica, neurona koji omogućuju veliku mogućnost učenja, dok u umjetnim neuronskim mrežama postoji mreža umjetnih neurona odnosno čvorova (engl. node) koji omogućuju učenje i rješavanje složenih problema. Umjetne neuronske mreže se koriste posljednjih 50 godina kako bi simulirale način na koji ljudski mozak rješava problem. Razvitkom računala i računalnih programa porasla je snaga i složenost problema koje umjetne neuronske mreže mogu rješavati. U današnje vrijeme se koriste za razne probleme koji obuhvaćaju programe za prepoznavanje teksta i slika, do mreža koje će predviđati meteorološke uvjete. Umjetne neuronske mreže su samo neke od metoda umjetne inteligencije.[5]

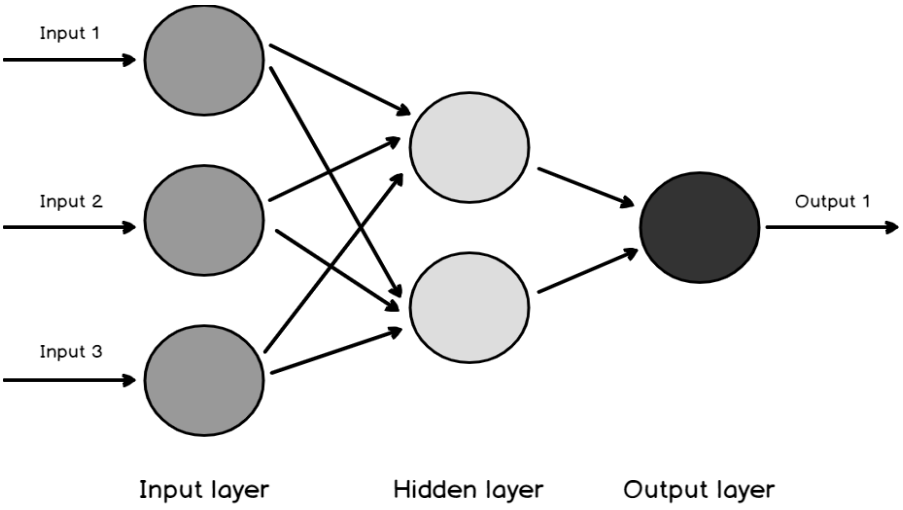
Umjetne neuronske mreže su građene od umjetnih neurona. Umjetni neuroni su izrađeni prema biološkim neuronima koji grade biološke neuronske mreže koje se nalaze u mozgu i leđnoj moždini. Slika 1. prikazuje biološki neuron, informacije dolaze putem dendrita, soma obrađuje pristigle informacije i prenosi ih dalje putem aksona. Kod umjetnih neurona informacije dolaze putem ulaznog sloja, obrađuju se u skrivenom sloju te izlaze kroz izlazni sloj. Na slici 2. prikazani su slojevi koji čine neuronsku mrežu, slojevi omogućuju neuronskoj mreži da uči i da generalizira na složene zadatke. Ulazni slojevi su prvi sloj neuronske mreže, on prima sirove ulazne podatke, a svaki neuron u ulaznom sloju odgovara jednoj značajci. Ulazni podatci se potom prosljeđuju skrivenom sloju bez obrade. Skriveni slojevi izvode većinu obrade ulaznih podataka te koriste matematičke modele kako bi dobili informacije iz ulaznog signala. Izlazni sloj je posljednji sloj mreže i on generira konačni izlaz, a ima izlaznih neurona koliko ima i klasa. [6] [7]



Slika 1. Prikaz prirodnog neurona [8]

Umjetne neuronske mreže možemo podijeliti na nekoliko vrsta. Statičke (unaprijedne, engl. Feedforward) i dinamičke (povratne, engl. Feedback). Statičke, unaprijedne umjetne neuronske mreže su osnovni tip mreža gdje informacija teče u jednom smjeru, od ulaza prema izlazu bez povratnih veza. Neki od vrsta unaprijednih veza su MLP (engl. Multilayer Perceptors) mreža koja se sastoji od jednog ulaza, više skrivenih slojeva i jednog izlaza te RBFN (engl. Radial Basis Function Network) koja koristi radijalne bazne funkcije kao funkcije aktivacije u skrivenim slojevima. Osnovne unaprijedne mreže (engl. Simple Neural Network) se koriste za obrazovanje i razumijevanje osnovnih principa neuronskih mreža. Povratne (engl. Feedback) neuronske mreže su vrsta mreža kod kojih je moguće da se informacije vraćaju u mrežu te na taj način mreža može pamtiti prethodne informacije i koristiti ih za donošenje daljnjih odluka. Neke od najvažnijih vrsta povratnih mreža su RNN, CNN, LSTM mreže i GRU mreže. RNN (engl. Reccurent Neural Networks) imaju povratne vezu zbog kojih informacije mogu teći u petlji unutar mreže. To omogućuje mreži da čuva podatke o prethodnim ulazima. CNN (engl. Convolutional Neural Networks) su specijalizirane mreže dizajnirane za rad s podacima u obliku mreža, kao što su slike. One koriste seriju konvolucijskih i potpuno povezanih slojeva kako bi automatski prepoznale značajke u složenim podacima i poboljšali točnost modela. LSTM mreže (engl. Long Short-Term Memory) su posebne vrste RNN mreža koje su dizajnirane kako bi se riješio problem dugoročne ovisnosti i zaborava u standardnim RNN-ovima. Jedinice LSTM imaju strukturu koja uključuje “zaboravne” i “ulazne” sklopke koje pomažu u održavanju važnih informacija kroz duže

intervale. GRU mreža (engl. Gated Recurrent Units) je slična mreža LSTM-u, ali je jednostavnije strukturirana. [5][9]



Slika 2. Prikaz slojeva u umjetnim neuronskim mrežama [10]

4.UMJETNA INTELIGENCIJA

4.1. UMJETNA INTELIGENCIJA

Umjetna inteligencija je sposobnost računala da imitira ljudske sposobnosti uz korištenje algoritama. Umjetna inteligencija koristi razne tehnologije i algoritme kako bi strojevima dala mogućnost da planiraju i uče s inteligencijom sličnom ljudskoj. Sistemi umjetne inteligencije na taj način mogu prepoznavati predmete i okoliš, učiti iz prijašnjih iskustava, donositi odluke, rješavati kompleksne probleme i uočavati obrasce.[11]

Termin umjetna inteligencija (engl. Artificial Intelligence, AI) skovan je 1956. godine od strane američkog znanstvenika John-a McCarthy-a na akademskoj konferenciji. Prvi program umjetne inteligencije dizajnirali su 1955. godine Allen Newell i Herbert A. Simon te su ga nazvali “Logic Theorist”. Njihov rad se temeljio na dokazivanju matematičkih teorema i dokazao je preko 40 različitih teorema. [12] [13]

Umjetne neuronske mreže jedna su od najbitnijih komponenti umjetne inteligencije, a druga je strojno učenje.

4.2. STROJNO UČENJE

Strojno učenje (engl. Machine Learning) je područje umjetne inteligencije koje omogućuje programima da uče i poboljšavaju svoj rad pomoću kolekcije podataka. Strojno učenje služi da se rješavaju problemi uz skupljanje skupa podataka i razvijanje računskih programa koji koriste skupove podataka kako bi se koristili za učenje i rješavanje problema.[14]

Strojno učenje se primjenjuje u medicini, ekonomiji, dubinskoj analizi podataka, razvoju videoigara, računalnoj sigurnosti. Strojno učenje se koristi u farmaceutskoj i kemijskoj industriji. Pomoću strojnog učenja se mogu poboljšati procesi kemijskih reakcija kako bi se povećala efikasnost, smanjio otpad i greške u mjerenjima. Strojno učenje na temelju skupa podataka i algoritama može predvidjeti kako i gdje će određeni protein reagirati s lijekom i na taj način znatno ubrzati procese razvoja lijekova.[15]

Cilj strojnog učenja nije samo ponavljanje naučenih podataka, već i predviđanje novih primjera. Želja je naučiti generalni model koji nadilazi primjere iz skupa za učenje tako da daje dobra predviđanja za primjere koji nisu viđeni tijekom učenja. To je primarna moć strojnog učenja, a naziva se generalizacijska sposobnost modela i algoritma za učenje. Generalizacijska sposobnost ovisi o: prikladnosti modela zadatku, količini podataka i o tome koliko su dobro parametri modela optimizirani. [14]

Uz predviđanje novih primjera, cilj strojnog učenja jest i prilagodba podataka modelu. U idealnom slučaju jedan model dovoljan je za primjenu svih dostupnih podataka.

Proces rada strojnog učenja počinje prikupljanjem i pripremom podataka. Tijekom pripreme podataka uklanjaju se nepotpuni podaci, normaliziraju vrijednosti kako bi podaci bili prikladni za analizu. Potrebno je izabrati pravilan algoritam rada s obzirom na problem koji je potrebno riješiti. Treniranje modela je važno jer tokom treniranja model uči odnose između ulaznih značajki i ciljane varijable. Uspješnost modela strojnog učenja se provjerava validacijom. U slučaju da rezultati nisu jednaki ciljanim rezultatima, potrebno je poboljšati izabrani model na neke od sljedećih načina: provjerom korištenih podataka, odlučivanjem koje podatke treba

ukloniti iz modela kako bi rezultat bio bolji provjerom ima li koji drugi model koji će točnije prikazati rezultate od onog korištenoga i prilagodbom parametara. [16]

4.3. VRSTE STROJNOG UČENJA

Strojno učenje se dijeli na: nadzirano učenje, nenadzirano učenje, polunadzirano i ojačano učenje.

4.3.1. NADZIRANO STROJNO UČENJE

Nadzirano strojno učenje je najjednostavnija i najpopularnija vrsta strojnog učenja. Kod nadziranog učenja skup podataka je kolekcija označenih primjera. Svaki element x_i se naziva vektor značajke. Vektor značajke je vektor u kojem svaka dimenzija j opisuje primjer na neki način. Na primjer, ako primjer u skupu podataka x označava osobu, onda značajka $x^{(1)}$ može označavati težinu osobe u kg dok značajka $x^{(2)}$ može označavati visinu osobu u cm i tako možemo dodavati niz značajki. Nadzirano učenje se dijeli u dvije kategorije: klasifikaciju i regresiju. Klasifikacija je problem automatskog dodjeljivanja oznake neoznačenom primjeru. Zadatak klasifikacijskog algoritma je razvrstati dobivene vrijednosti ulaza u kategorije kojima ti podaci pripadaju, na temelju odrađenog treninga nad podacima. Regresija je problem predviđanja stvarne vrijednosti oznake na temelju neoznačenog primjera. Primjer toga je procjena cijene kuće na temelju značajki kuće poput: površine, lokacije, broja soba i sličnih parametara. Problem regresije se rješava algoritmom za učenje regresije koji za ulazne podatke uzima skup označenih primjera i proizvodi model koji može uzeti neoznačen primjer kao ulaz i dati cilj kao izlaz. Najbitniji algoritmi za regresiju su: linearna regresija, logistička regresija i polinomna regresija. Cilj nadziranog strojnog učenja je iskoristiti skup podataka kako bi se napravio model koji će uzeti vektor značajki x kao ulaz, a kao izlazni signal će dati određenu informaciju, npr. vjerojatnost [14]

4.3.2. NENADZIRANO STROJNO UČENJE

U nenadziranom strojnom učenju skup podataka je kolekcija neoznačenih primjera. Cilj nenadziranog učenja je stvoriti model koji uzima vektor značajke x kao ulaz i pretvara ga u drugi vektor značajke ili u vrijednost koja se može iskoristiti za rješavanje problema.[14]

4.3.3. POLUNADZIRANO STROJNO UČENJE

Kod polunadziranog strojnog učenja skup podataka sadrži i označene i neoznačene primjere, a najčešće je broj neoznačenih primjera veći od broja označenih. Cilj polunadziranog strojnog učenja je isti kao kod nadziranog učenja, to jest ono uči algoritam da na temelju podataka koji su već naučeni predviđa izlaz za nikad viđene ni naučene podatke. U polunadziranom učenju neoznačeni primjeri djeluju tako da daju nove informacije te se na taj način stvara bolji model. [14]

4.3.4. OJAČANO STROJNO UČENJE

Ojačano strojno učenje je podpodručje strojnog učenja u kojem je stroj "živ" u okolišu te doživljava okoliš kao vektor značajki. Stroj može odrađivati različite radnje koje donose različite nagrade. Cilj ojačanog strojnog učenja jest da stroj nauči pravila. Pravila su funkcija koja uzima vektor značajke kao ulaz, a izlaz je optimalna akcija koju je potrebno izvršiti. Akcija je optimalna ako maksimizira očekivanu nagradu. Ojačano strojno učenje rješava probleme gdje je donošenje odluka uzastopno, a cilj dugoročan, npr. probleme u logistici, robotici i igranju igrica. [14]

4.4. PROGRAMSKI JEZIK R

Programski jezik R jedan je od najkorištenijih i najpopularnijih jezika namijenjen statističkom računanju i grafičkom prikazivanju. Stvorili su ga Ross Ihaka i Robert Gentleman na Sveučilištu u Aucklandu 1993., a ime je dobio po prvom slovu imena oba autora. Program je dostupan kao besplatan softver na velikom broju operativnih sustava. R program je vodeći alat za strojno učenje, statistiku, umjetnu inteligenciju i analizu podataka, omogućuje jednostavno stvaranje objekata, funkcija. Pruža širok raspon statističkih i grafičkih tehnika koje se lako prilagođavaju proširivanju. Velika prednost R-a je lakoća kojom se mogu stvarati kvalitetno dizajnirani grafikoni, uključujući pri tome matematičke simbole i formule na mjestima gdje su potrebni. [17]

Program R je integrirani paket softverskih alata za manipulaciju podacima, izračunavanje i grafički prikaz. Uključuje: učinkovitu mogućnost rukovanja podacima i pohrane, skup operatora za izračune na poljima, grafičke mogućnosti za analizu podataka te veliku, koherentnu i integriranu zbirku srednjih alata za analizu podataka. R je jednostavan, dobro razvijen i učinkovit programski jezik koji uključuje uvjetne izraze, funkcije, petlje koje definira korisnik i mogućnosti unosa i izlaza. R je dizajniran oko stvarnog računalnog jezika te omogućuje korisnicima definiranje novih funkcija. [18]

Slika 3. Logo R programskog jezika [19]

5.ALGORITMI UMJETNIH NEURONSKIH MREŽA

5.1. REGRESIJA

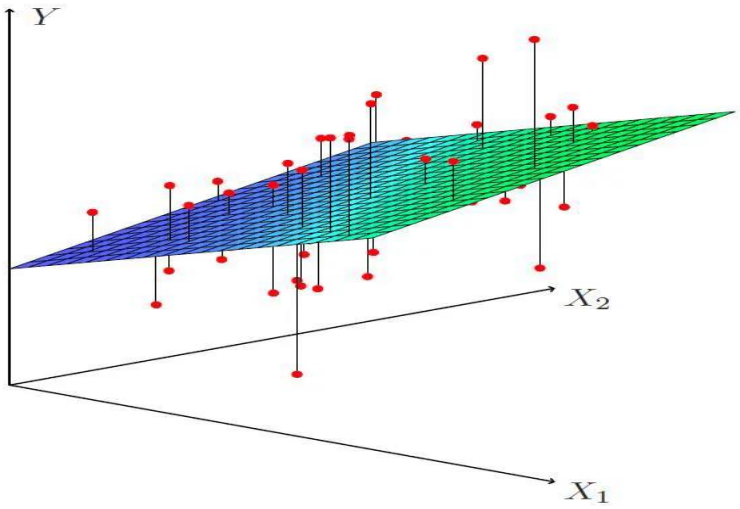
Regresija je statistička metoda kojoj je cilj odrediti karakter odnosa između zavisne varijable i niza drugih, nezavisnih varijabli. Pokazuje jesu li promjene opažene kod zavisne varijable povezane s promjenama kod nezavisnih varijabli, a to radi na način da pronalazi krivulju koja najbolje odgovara podacima te promatra kako su podaci raspoređeni oko linije. [20]

Regresija se dijeli na linearnu, logističku i polinomnu. Linearna regresija može biti jednostavna i višestruka. [11]

Linearna regresija je model, odnosno analiza koja računa vrijednost izlazne varijable u odnosu na poznate vrijednosti ulaznih varijabli, ona podrazumijeva linearnu ovisnost između ulaznih i jedne izlazne varijable. Jednostavna regresija ima samo jednu ulaznu varijablu dok višestruka sadrži više ulaznih varijabli i jednu izlaznu. Prikaz višestruke regresije dan je slikom 4. [21]

Logistička regresija je algoritam koji se primjenjuje na probleme binarnih klasifikacija, a za rješenje daje binarni ishod ograničen na dva ishoda: točno/netočno, da/ne, 1/0. Logistička regresija analizira odnos između jedne ili više nezavisnih varijabli i klasificira podatke u posebne klase. Primjenjuje se u prediktivnom modeliranju u kojem model procjenjuje matematičku vjerojatnost da će se neki događaj dogoditi npr. jesu li osobe glasale ili ne na izborima. [22]

Polinomna regresija se koristi ako odnos među danim podacima nije linearan. Naime, u tom slučaju ne možemo koristit linearnu regresiju jer rezultat neće biti točan. Za nelinearne podatke koristimo polinomnu regresiju jer ona povezuje zavisne i nezavisne varijable i pomoću nje možemo nelinearne podatke prikazati krivuljom.[23]



Slika 4. Prikaz višestruke linearne regresije [24]

Regresija se smatra jednim od najbitnijih algoritama strojnog učenja, a jednako često se koristi kao algoritam umjetnih neuronskih mreža.

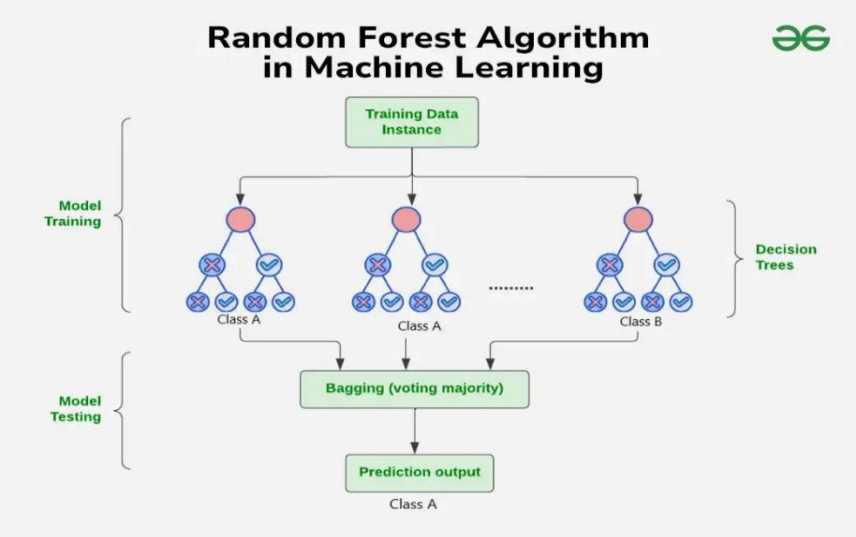
5.2. SLUČAJNA ŠUMA

Slučajna šuma je algoritam nadziranog strojnog učenja koji se sastoji od algoritama stabla odluke. Ona sadrži velik broj algoritama stabla odluke koji rade kao skup. Stvorena šuma uči se kroz pakiranje ili tzv. bootstrap nagomilavanje. Stablo odluke je algoritam nadziranog strojnog učenja, a primjenjuje za rješavanje regresijskih ili klasifikacijskih problema. Cilj stabla odluke je stvoriti model učenja koji može predvidjeti vrijednost ili klasu zavisne varijable učenjem jednostavnih pravila odlučivanja iz podataka za učenje. Svako stablo u šumi predviđa klasu u slučaju klasifikacije ili vrijednost u slučaju regresije na temelju zadanog skupa podataka. Konačna odluka slučajne šume se donosi prosjekom ili glasanjem svih predviđanja pojedinačnih stabala, što znači da će s većim brojem stabala u šumi predviđanje biti točnije. [25]

Na slici 5. prikazan je primjer modela slučajne šume. Najveća razlika između slučajne šume i stabla odluke jest da se u slučajnoj šumi cijepanje i uspostavljanje korijenskih članova vrši nasumično. Slučajna šuma koristi metodu pakiranja (engl. bagging) za stvaranje predviđanja. Pakiranje omogućuje individualnim stablima odluke da iz šume slučajnim odabirom uzimaju podatke za učenje, što uzrokuje različite izlaze individualnih stabala. To znači da svako uzima

jedan određeni dio podataka, a ne sve dostupne podatke i stvara predviđanje na temelju dijela podataka.[14] [25]

Prednosti slučajne šume su: da može raditi s velikim brojem podataka, može rukovati s podacima koji nedostaju jer zbog velikog broja stabala iz prosjeka može dobiti predviđanje, robusna je i točna te manje osjetljiva na promjene podataka. Nedostaci slučajne šume su: računalna složenost jer zbog velikog broja stabala odluke raste složenost i potrebno vrijeme izvršenja rada. U odnosu na stablo odluke, slučajnu šumu je teže interpretirati jer uzima u obzir velik broj stabala. [25]

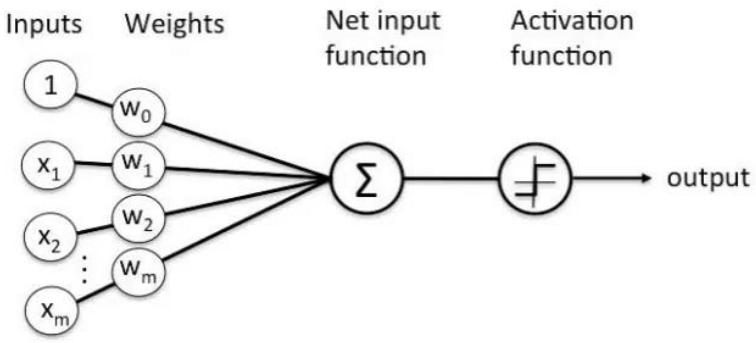


Slika 5. Prikaz slučajne šume [26]

5.3. UNAPRIJEDNA NEURONSKA MREŽA

FNN (engl. Feed Forward Neural Network) ili unaprijedne neuronske mreže su jedna od najjednostavnijih vrsta umjetnih neuronskih mreža. U ovoj mreži informacije se kreću samo u jednom smjeru, od ulaznih čvorova preko skrivenih do izlaznog čvora. U ovoj mreži nema povratnih veza niti petlji. FNN mreže su jednostavnije od RNN (engl. Reccurent Neural Networks) i od CNN (engl. Convolutional Neural Networks). Unaprijedne mreže se sastoje od 3 sloja: ulaznog, skrivenih (jednog ili više) i izlaznog sloja. Ulazni sloj se sastoji od neurona koji primaju podatke i šalju ih idućem sloju, ali ih ne mijenjaju na bilo koji način. Broj ulaznih neurona ovisi o broju značajki u skupu podataka. Skriveni slojevi nisu izloženi ulazima niti izlazu te se mogu

smatrati računskim motorom neuronske mreže. Skriveni slojevi izvode većinu obrade ulaznih podataka te koriste matematičke modele kako bi dobili informacije iz ulaznog signala, i šalju te podatke idućem sloju. Izlazni sloj proizvodi izlaz za dane ulaze, a broj neurona u izlaznom sloju ovisi o broju mogućih izlaza koje je mreža dizajnirana proizvesti. Svaki neuron u jednom sloju povezan je sa neuronom u svakom sljedećem sloju, što ovu mrežu čini potpuno povezanom. [28] Rad FNN-a sastoji se od dvije faze: feedforward faze i fazu povratnog širenja (engl. backpropagation phase). U feedforward fazi ulazni se podaci unose i prosljeđuju dalje kroz mrežu. Na ulazne podatke djeluje težina, a zbrojevi umnožaka ulaznih vrijednosti i odgovarajućih im težina daju neto ulaznu funkciju koja se označava grčkim slovom sigma i prolazi kroz aktivacijsku funkciju. [27]



Slika 6. Dijagram jednog čvora [28]

Slika 6. prikazuje kako se na svakom skrivenom sloju izračunava neto ulazna funkcija te se provodi kroz aktivacijsku funkciju koja uvodi nelinearnost u model. Aktivacijske funkcije svojim uvođenjem nelinearnosti omogućuju modelu da uči. To se provodi dok se ne dosegne izlazni sloj u kojem se provodi predviđanje. Nakon što se napravi predviđanje, počinje faza povratnog širenja, tj. backpropagation faza u kojoj se računa pogreška. Ta se greška vraća povratno u mrežu, a težine prilagođavaju kako bi se pogreška smanjila. Taj proces prilagodbe težine se najčešće provodi pomoću optimizacijskog algoritma gradijentnog spuštanja. Treniranje FNN-a je važno jer uključuje korištenje skupa podataka za prilagodbu težine veze između neurona, a provodi se kroz iterativni proces. Što se više puta provede više će se smanjiti greške u predviđanju. [27]

6. PRIMJENA

6.1. OPIS PODATAKA

Skup podataka koji se primjenjuje u ovome radu preuzet je s online platforme Kaggle namijenjene znanstvenicima koji se bave analizom podataka i područjima umjetne inteligencije. [4]

Skup podataka koji je korišten u radu sadrži 26 varijabli i informacije o 9982 različita organska spoja. Od 26 varijabli 25 ih je ulazno, a jedina izlazna varijabla je topljivost (engl. Solubility).

Ulazne varijable su predstavljene s imenom i mjernim jedinicama:

- Identifikacijski broj izvora spoja (engl. ID)
- Ime spoja (engl. Name of compound)
- Međunarodni kemijski identifikator (engl. The IUPAC International Chemical Identifier, InChI)
- Ključ međunarodnog kemijskog identifikatora (engl. Hashed InChI value, InChiKey)
- Tekstualni prikaz kemijske strukture (engl. SMILES)
- Datoteka s podacima o strukturi (engl. Standard deviation of multiple occurrences, SD)
- Pojavljivanje (engl. Occurrence)-broj pojavljivanje pojedinog spoja
- Kategorije pouzdanosti (engl. Group)
- Molekulska masa (engl. Molecular weight)-MolWt-g/mol
- Molarna refraktivnost (engl. Molar refractivity)- Mol MR- cm^3/mol
- Broj teških atoma (engl. Number of heavy atoms)
- Broj donora vodikovih veza (engl. Hydrogen bond donors)
- Broj akceptora vodikovih veza (engl. Hydrogen bond acceptors)
- Molekulski logaritamski koeficijent podjele (engl. Molecular Logarithm of Partition Coefficient) -MolLogP
- Broj heteroatoma (engl. Number of heteroatoms)
- Broj rotacijskih veza (engl. Number of rotatable bonds)
- Broj aromatskih prstena (engl. Number of aromatic rings)

- Broj zasićenih prstena (engl. Number of saturated rings)
- Broj alifatskih prstena (engl. Number of aliphatic rings)
- Ukupan broj prstena (engl. Total number of rings)
- Topološka polarna površina (engl. Topological polar surface area)-TPSA- A^2
- Labuteova približna površina (engl. Labute's Approximate Surface Area)- A^2
- Balabanov J indeks (engl. Balaban J index)
- Bertzov kemijsko topološki index (engl. Bertz Chemical Topological Index, Bertz CT)

Izlazna varijabla je:

- Topljivost (engl. Solubility)-g/mol

Nakon što se učita bazu podataka u R programski jezik pomoću naredbe

df <-read.csv("C:\\Users\\Korisnik\\Desktop\\curated-solubility-dataset.csv"), naredbom *head(df)* dobije se uvid u set podataka.

Tablica 1. Prikaz varijabli i njihovih vrijednosti

```
> head(df)
         ID        Name        InChI   InChIKey     SMILES  Solubility         SD Ocurrences
1 -1.2452721  1.07778501  0.005725788  0.8173996 -1.7140927 -0.30666003 -0.2873798 -0.3694084
2 -1.0249160  0.04752724 -1.215775610 -0.9374676  1.3868552 -0.15406862 -0.2873798 -0.3694084
3 -0.7979666 -0.38991237  1.076274598 -1.6155397  0.4547664  0.30100691 -0.2873798 -0.3694084
4 -0.5023077  1.71539699 -1.710969500  1.4160047 -1.3760977 -0.43683818 -0.2873798 -0.3694084
5 -0.4818337 -0.48579913  0.129958032 -1.1564356 -1.1560886 -0.74832779 -0.2873798 -0.3694084
6 -1.7317905  1.70594829  1.431273441 -0.4665650 -0.4551134 -0.09849083 -0.2873798 -0.3694084
       Group      MolWt     MolLogP      MolMR HeavyAtomCount NumHAcceptors  NumHDonors NumHeteroato
1 -0.4895766  0.6832703 0.56255836  0.7663046      0.4595278    -0.9967334 -0.74453698     -0.6749
2 -0.4895766 -0.5292836 0.12119517 -0.3201296     -0.3573632    -0.7108724 -0.07293314     -0.6749
3 -0.4895766 -0.6846434 0.04927395 -0.6438768     -0.6841196    -0.7108724 -0.74453698     -0.6749
4 -0.4895766  2.6580663 1.74456798  2.8784891      2.9102007     0.7184327  0.59867071      0.3806
5 -0.4895766  0.8462367 0.14390863  1.1237750      1.1130405     0.7184327 -0.74453698      0.1695
6 -0.4895766 -0.8062099 0.18729447 -0.5486444     -0.6841196    -0.9967334 -0.74453698     -1.0972
  NumRotatableBonds NumValenceElectrons NumAromaticRings NumSaturatedRings NumAliphaticRings
1         2.2892049           0.7375695       -0.81587082        -0.3326821        -0.4244046
2        -0.7212831          -0.4979792        0.71151525        -0.3326821         0.5237617
3        -0.5441955          -0.7450889       -0.05217778        -0.3326821        -0.4244046
4         1.0495922           2.6217812        3.76628739        -0.3326821        -0.4244046
5         1.4037673           1.0773453        0.71151525         4.2148466         3.3682605
6        -0.5441955          -0.7450889       -0.05217778        -0.3326821        -0.4244046
   RingCount        TPSA  LabuteASA   BalabanJ     BertzCT
1 -0.9219105 -0.98595534  0.6487869 -2.1924195 -0.47007784
2  0.9025365 -0.52659026 -0.4411172  0.1748631  0.08029623
3 -0.3137615 -0.71649273 -0.6624333  0.5660065 -0.48419387
4  2.7269834  0.91969938  2.8097723 -2.1924193  2.73916038
5  2.7269834 -0.09248237  0.9713319 -1.1985559  0.55350459
6 -0.3137615 -0.98595534 -0.6941416  0.6218927 -0.46887797
```

Za dobivanje boljeg uvida u distribuciju varijabli koristi se naredba *summary*(df)

Tablica 2. Prikaz distribucije varijabli

```
> summary(df)
       ID.V1              Name.V1             InChI.V1          InChIKey.V1
 Min.   :-1.7317905   Min.   :-1.7368764   Min.   :-1.7317905   Min.   :-1.7317905
 1st Qu.:-0.8658953   1st Qu.:-0.8654092   1st Qu.:-0.8658953   1st Qu.:-0.8658953
 Median : 0.0000000   Median : 0.0046582   Median : 0.0000000   Median : 0.0000000
 Mean   : 0.0000000   Mean   : 0.0000000   Mean   : 0.0000000   Mean   : 0.0000000
 3rd Qu.: 0.8658953   3rd Qu.: 0.8659768   3rd Qu.: 0.8658953   3rd Qu.: 0.8658953
 Max.   : 1.7317905   Max.   : 1.7248457   Max.   : 1.7317905   Max.   : 1.7317905
     SMILES.V1          Solubility.V1          SD.V1          Ocurrences.V1          Group.V1
 Min.   :-1.7317905   Min.   :-4.341774   Min.   :-0.287380   Min.   :-0.36941   Min.   :-0.4895766
 1st Qu.:-0.8658953   1st Qu.:-0.606555   1st Qu.:-0.287380   1st Qu.:-0.36941   1st Qu.:-0.4895766
 Median : 0.0000000   Median : 0.114746   Median :-0.287380   Median :-0.36941   Median :-0.4895766
 Mean   : 0.0000000   Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.00000   Mean   : 0.0000000
 3rd Qu.: 0.8658953   3rd Qu.: 0.709487   3rd Qu.:-0.287380   3rd Qu.:-0.36941   3rd Qu.:-0.4895766
 Max.   : 1.7317905   Max.   : 2.122999   Max.   :16.202213   Max.   :35.78191   Max.   : 2.9446847
     MolWt.V1           MolLogP.V1          MolMR.V1        HeavyAtomCount.V1   NumHAcceptors.V1
 Min.   :-1.398932   Min.   :-12.181797   Min.   :-1.435732   Min.   :-1.337632   Min.   :-0.996733
 1st Qu.:-0.566785   1st Qu.: -0.386446   1st Qu.:-0.562993   1st Qu.:-0.520741   1st Qu.:-0.425011
 Median :-0.206234   Median : -0.008513   Median :-0.175423   Median :-0.193985   Median :-0.139150
 Mean   : 0.000000   Mean   :  0.000000   Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000
 3rd Qu.: 0.291945   3rd Qu.:  0.409315   3rd Qu.: 0.325504   3rd Qu.: 0.296150   3rd Qu.: 0.146711
 Max.   :27.325533   Max.   : 18.921811   Max.   :29.072857   Max.   :30.276048   Max.   :23.587315
   NumHDonors.V1     NumHeteroatoms.V1  NumRotatableBonds.V1 NumValenceElectrons.V1
 Min.   :-0.744537   Min.   :-1.097266   Min.   :-0.721283   Min.   :-1.455529
 1st Qu.:-0.744537   1st Qu.:-0.463857   1st Qu.:-0.544196   1st Qu.:-0.559757
 Median :-0.072933   Median :-0.252721   Median :-0.190020   Median :-0.189092
 Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000
 3rd Qu.: 0.598671   3rd Qu.: 0.169552   3rd Qu.: 0.164155   3rd Qu.: 0.274239
 Max.   :16.717163   Max.   :17.693873   Max.   :24.248058   Max.   :29.618519
 NumAromaticRings.V1 NumSaturatedRings.V1 NumAliphaticRings.V1    RingCount.V1          TPSA.V1
 Min.   :-0.815871   Min.   :-0.33268     Min.   :-0.424405    Min.   :-0.921910   Min.   :-0.985955
 1st Qu.:-0.815871   1st Qu.:-0.33268     1st Qu.:-0.424405    1st Qu.:-0.921910   1st Qu.:-0.570790
 Median :-0.052178   Median :-0.33268     Median :-0.424405    Median :-0.313761   Median :-0.185303
 Mean   : 0.000000   Mean   : 0.00000     Mean   : 0.000000    Mean   : 0.000000   Mean   : 0.000000
 3rd Qu.: 0.711515   3rd Qu.:-0.33268     3rd Qu.: 0.523762    3rd Qu.: 0.294387   3rd Qu.: 0.283060
 Max.   :25.913385   Max.   :33.77378     Max.   :28.020584    Max.   :20.971453   Max.   :18.183302


    LabuteASA.V1         BalabanJ.V1         BertzCT.V1
 Min.   :-1.326246   Min.   :-2.192423   Min.   :-0.85494
 1st Qu.:-0.553065   1st Qu.:-0.355453   1st Qu.:-0.55630
 Median :-0.204187   Median : 0.135035   Median :-0.21165
 Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.00000
 3rd Qu.: 0.264257   3rd Qu.: 0.586787   3rd Qu.: 0.25470
 Max.   :27.749109   Max.   : 4.697097   Max.   :37.05041
```

Naredba *summary* ispisuje minimalnu i maksimalnu vrijednost varijable, medijan i kvantile kao informacije o distribuciji raspodjele, prosjek kao srednju vrijednost distribucije. Prvi i treći kvantili označavaju podatke od kojih je 25% podataka manje i od kojih je 75% podataka manje.

6.2. IZRADA MODELA

Za primjenu modela korišten je R programski jezik u RStudio sučelju. Izrađeni modeli su prikazani grafički, te je prikazana RMSE pogreška koja pokazuje koji od modela najbolje opisuju i procjenjuju topljivost spojeva.

Prije početka korištenja modela u RStudio je potrebno instalirati pakete koji sadrže potrebne alate i funkcije kako bi se algoritam mogao interpretirati. Paketi se učitavaju pomoću funkcije *install.packages(“”),* a između znakova zagrada i navodnika se upisuje ime paketa koji se koristi. Paketi koji su se koristili su:

- Caret- – paket s funkcijama za stvaranje predviđanja
- dplyr – paket za interpretaciju podataka
- randomForest- paket za izradu slučajne šume
- ggplot2 – paket s funkcijama za stvaranje grafova
- nnet – paket za izradu neuronske mreže
- NeuralNetTools – paket s funkcijama za interpretaciju i prikaz neuronske mreže

Paketi se u RStudio učitavaju pomoću funkcije *library (),* a unutar zagrade se upisuje ime paketa. Nakon učitavanja podataka, preuzimanja i učitavanja paketa, potrebno je podesiti podatke. Za rad s algoritmima je bilo potrebno da svi podaci iz naše baze podataka budu u numeričkom obliku kako bi se uspjeli provesti modeli regresije i ostali korišteni modeli za ovaj skup podataka. Provjera i promjena u numeričke podatka se provela pomoću naredbe *mutate ()* koja je bila dio veće funkcije. Svi ovi koraci su potrebni prije početka korištenja modela, a njihov izgled u RStudio je prikazan na slici 7.

```r
#INSTALIRAVANJE POTREBNIH PAKETA
install.packages("nnet")
install.packages("dplyr")
install.packages("caret")
install.packages("ggplot2")
install.packages("NeuralNetTools")
install.packages("randomForest")
#UČITAVANJE PAKETA
library(dplyr)
library(caret)
library(nnet)
library(NeuralNetTools)
library(randomForest)
#UČITAVANJE PODATAKA
df<- read.csv("C:\\Users\\Korisnik\\Desktop\\curated-solubility-dataset.csv")
#PRIPREMA PODATAKA
df <- df %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), as.numeric)) %>%
  na.omit()
df <- df %>%
  mutate(across(where(is.numeric), scale))
```

Slika 7. Prikaz početka koda i pripreme podataka

6.2.1. VIŠESTRUKA LINEARNA REGRESIJA

U ovom slučaju se koristi višestruka linearna regresija jer skup podataka u kojem su korišteni podaci sadrži više ulaznih varijabli.

Na slici 8. prikazan je kod koji se koristio da pokaže način rada modela i grafički prikaz podataka.

```r
#VIŠESTRUKA LINEARNA REGRESIJA
set.seed(42)
trainIndex <- createDataPartition(df$Solubility, p = 0.8, list = FALSE)
trainData <- df[trainIndex, ]
testData <- df[-trainIndex, ]
input_columns <- setdiff(names(df), "Solubility")
#MODEL
formula <- as.formula(paste("Solubility ~", paste(input_columns, collapse = " + ")))
lm_model <- lm(formula, data = trainData)
summary(lm_model)
predictions <- predict(lm_model, newdata = testData)
mse <- mean((actuals - predictions)^2)
rmse <- sqrt(mse)
sst <- sum((actuals - mean(actuals))^2)
ssr <- sum((actuals - predictions)^2)
r2 <- 1 - (ssr / sst)
first_10_actuals <- actuals[1:10]
first_10_predictions <- predictions[1:10]
mse10 <- mean((first_10_actuals - first_10_predictions)^2)
rmse10 <- sqrt(mse10)
cat("Root Mean Squared Error (RMSE):", round(rmse, 4), "\n")
cat("Root Mean Squared Error for the first 10 observations (RMSE10):", round(rmse10, 4), "\n")
cat("R-squared (R²):", round(r2, 4), "\n")
#GRAFIČKI PRIKAZ
comparison <- data.frame(Actual = actuals, Predicted = predictions)
ggplot(comparison, aes(x = Actual, y = Predicted)) +
  geom_point(color = 'blue') +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  ggtitle('Actual vs. Predicted Values') +
  xlab('Actual Solubility') +
  ylab('Predicted Solubility') +
  theme_minimal()
```

Slika 8. Kod za višestruku linearnu regresiju

Za dobivanje uvida u model koristimo naredbu *summary(lm_model)*

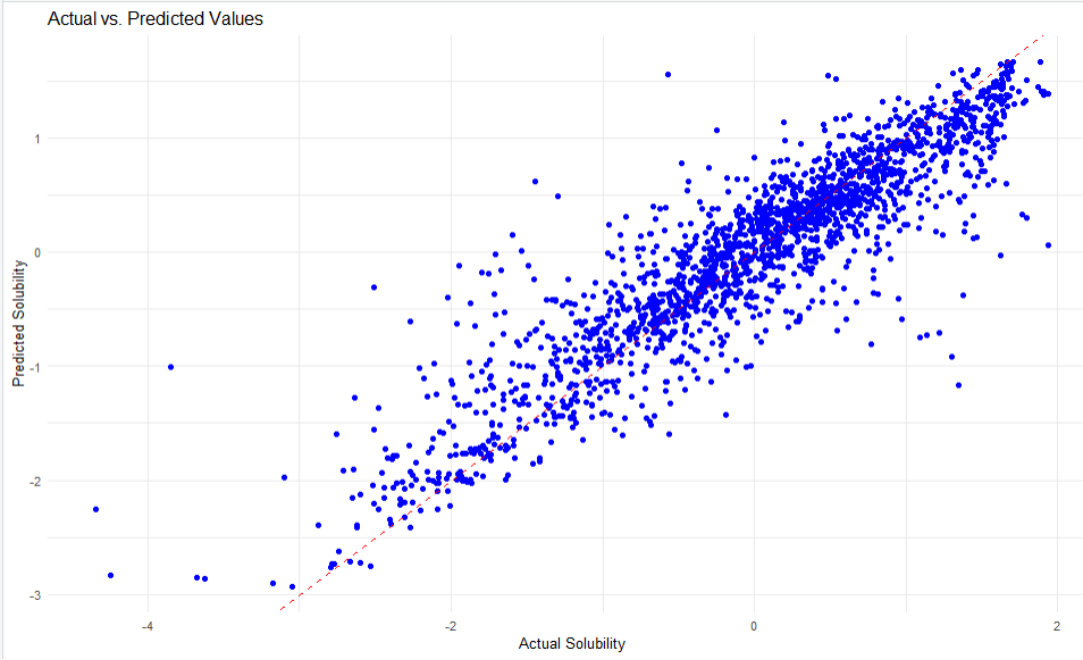
Tablica 3. Prikaz rada model višestruke linearne regresije i korištenja naredba *summary ()*

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.9082 -0.3722  0.0336  0.4050  6.6417

Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -0.0002695  0.0075646  -0.036 0.971577
ID                    0.0161688  0.0083521   1.936 0.052917 .
Name                 -0.0008380  0.0079109  -0.106 0.915645
InChI                 0.1212691  0.0081022  14.967  < 2e-16 ***
InChIKey              0.0015009  0.0075655   0.198 0.842752
SMILES                0.0489402  0.0087712   5.580 2.49e-08 ***
SD                   -0.0568147  0.0087883  -6.465 1.07e-10 ***
Ocurrences           -0.0627358  0.0117430  -5.342 9.43e-08 ***
Group                 0.0500089  0.0116105   4.307 1.67e-05 ***
MolWt                -0.3453528  0.0414583  -8.330  < 2e-16 ***
MolLogP              -0.6339358  0.0207636 -30.531  < 2e-16 ***
MolMR                 0.1506132  0.0729110   2.066 0.038888 *
HeavyAtomCount       -2.2159644  0.2243162  -9.879  < 2e-16 ***
NumHAcceptors         0.2029268  0.0256497   7.911 2.89e-15 ***
NumHDonors            0.0855373  0.0123023   6.953 3.86e-12 ***
NumHeteroatoms       -0.3037737  0.0282727 -10.744  < 2e-16 ***
NumRotatableBonds     0.0328784  0.0205051   1.603 0.108881
NumValenceElectrons   2.0124991  0.1787356  11.260  < 2e-16 ***
NumAromaticRings     -0.3742578  0.0271662 -13.777  < 2e-16 ***
NumSaturatedRings     0.0470090  0.0198515   2.368 0.017907 *
NumAliphaticRings    -0.1960595  0.0235858  -8.313  < 2e-16 ***
RingCount                  NA         NA      NA       NA
TPSA                 -0.1041237  0.0269676  -3.861 0.000114 ***
LabuteASA            -0.2147378  0.0844941  -2.541 0.011058 *
BalabanJ             -0.0440904  0.0104651  -4.213 2.55e-05 ***
BertzCT               0.9374893  0.0445055  21.065  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6759 on 7961 degrees of freedom
Multiple R-squared:  0.5398,    Adjusted R-squared:  0.5384
F-statistic:   389 on 24 and 7961 DF,  p-value: < 2.2e-16
```

U tablici 3. najprije su ispisani podaci deskriptivne statističke analize koji su potrebni za analizu modela. Nakon njih je prikaz izračunatih koeficijenata za svaku ulaznu varijablu, uz vrijednosti standardnih pogrešaka, t – vrijednosti i Pr(> ItI) vrijednosti. t – vrijednost računa se kao omjer procijenjene vrijednosti koeficijenta i standardne pogreške. Njena vrijednost upućuje na to može li se odbaciti nulta hipoteza, a koristi se za izračun Pr(> ItI) vrijednosti koja predstavlja individualnu p – vrijednost svakog parametra i određuje može li se odbaciti nulta hipoteza. Nakon toga slijedi ispis signifikantnih kodova koji ukazuju na to koliko je koji parametar značajan. Što je manja p – vrijednost parametra, to je on većeg značaja. U zadnjem dijelu tablice dane su vrijednosti preostale standardne pogreške (RSE), kvadratne pogreške te F – statistika zajedno s p – vrijednosti. Prvi korak u interpretaciji modela proučavanje je F – statistike i njene p – vrijednosti. F – statistika pokazuje odnos između prediktora i odgovora, a što je njena vrijednost veća, to postoji više razloga za odbacivanje nulte hipoteze. p – vrijednost modela iznosi 2.2e-16 što znači da se nulta hipoteza odbacuje i da je ukupni model značajan u stvaranju predviđanja.

RMSE vrijednost predstavlja prosječnu razliku između predviđene i stvarne vrijednost, ukoliko je manja vrijednost RMSE to znači je model precizniji i manja je pogreška. R^2 predstavlja točnost s kojom se predviđa vrijednost izlazne varijable, u ovom slučaju izlazna varijabla je topljivost (Solubility). R^2 se označava u postocima ili decimalnim brojevima u rasponu od 0 do 1. Vrijednost 1 označava 100%, tj. označava izlaz predviđen bez pogreške. U ovom modelu vrijednost R^2 iznosi 51.63%, te se može reći da se 51.63% izlaznih podataka predviđa točno.

Sa slike 9. se može vidjeti da stvarne i predviđene vrijednosti nisu snažno linearno korelirane te da postoji određena pogreška u modelu. Modelom dobivena pogreška iznosi RMSE=0.7098, a pogreška dobivena validacijom iznosi $RMSE_{10}$=0.7436.



Slika 9. Grafički prikaz rezultata modela višestruke regresije

6.2.2. SLUČAJNA ŠUMA

Teorijska pozadina modela slučajne šume je obrađena u prethodnom poglavlju pa se ovdje ne spominje.

Na slici 10. je prikazan kod za razvoj slučajne šume.

```
#KOD ZA RAZVOJ SLUČAJNE ŠUME
set.seed(42)
trainIndex <- createDataPartition(df$Solubility, p = 0.8, list = FALSE)
trainData <- df[trainIndex, ]
testData <- df[-trainIndex, ]
# MODEL
rf_model <- randomForest(Solubility ~ ., data = trainData, importance = TRUE, ntree = 500)
print(rf_model)
varImpPlot(rf_model)
predictions <- predict(rf_model, newdata = testData)
actuals <- testData$Solubility
mse <- mean((actuals - predictions)^2)
rmse <- sqrt(mse)
sst <- sum((actuals - mean(actuals))^2)
ssr <- sum((actuals - predictions)^2)
r2 <- 1 - (ssr / sst)
first_10_actuals <- actuals[1:10]
first_10_predictions <- predictions[1:10]
mse10 <- mean((first_10_actuals - first_10_predictions)^2)
rmse10 <- sqrt(mse10)
cat("Root Mean Squared Error (RMSE):", round(rmse, 4), "\n")
cat("Root Mean Squared Error for the first 10 observations (RMSE10):", round(rmse10, 4), "\n")
cat("R-squared (R²):", round(r2, 4), "\n")
# GRAFIČKI PRIKAZ
comparison <- data.frame(Actual = actuals, Predicted = predictions)
ggplot(comparison, aes(x = Actual, y = Predicted)) +
  geom_point(color = 'blue') +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  ggtitle('Actual vs. Predicted Values') +
  xlab('Actual Solubility') +
  ylab('Predicted Solubility') +
  theme_minimal()
```

Slika 10. Kod za prikaz razvoja slučajne šume

Za uvid u rad modela koristi se naredba *print(rf_model).*

```
> print(rf_model)

Call:
 randomForest(formula = Solubility ~ ., data = trainData, importance = TRUE,      ntree = 500)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 8

          Mean of squared residuals: 0.1880191
                    % Var explained: 81
```

Slika 11. Uvid u model slučajne šume

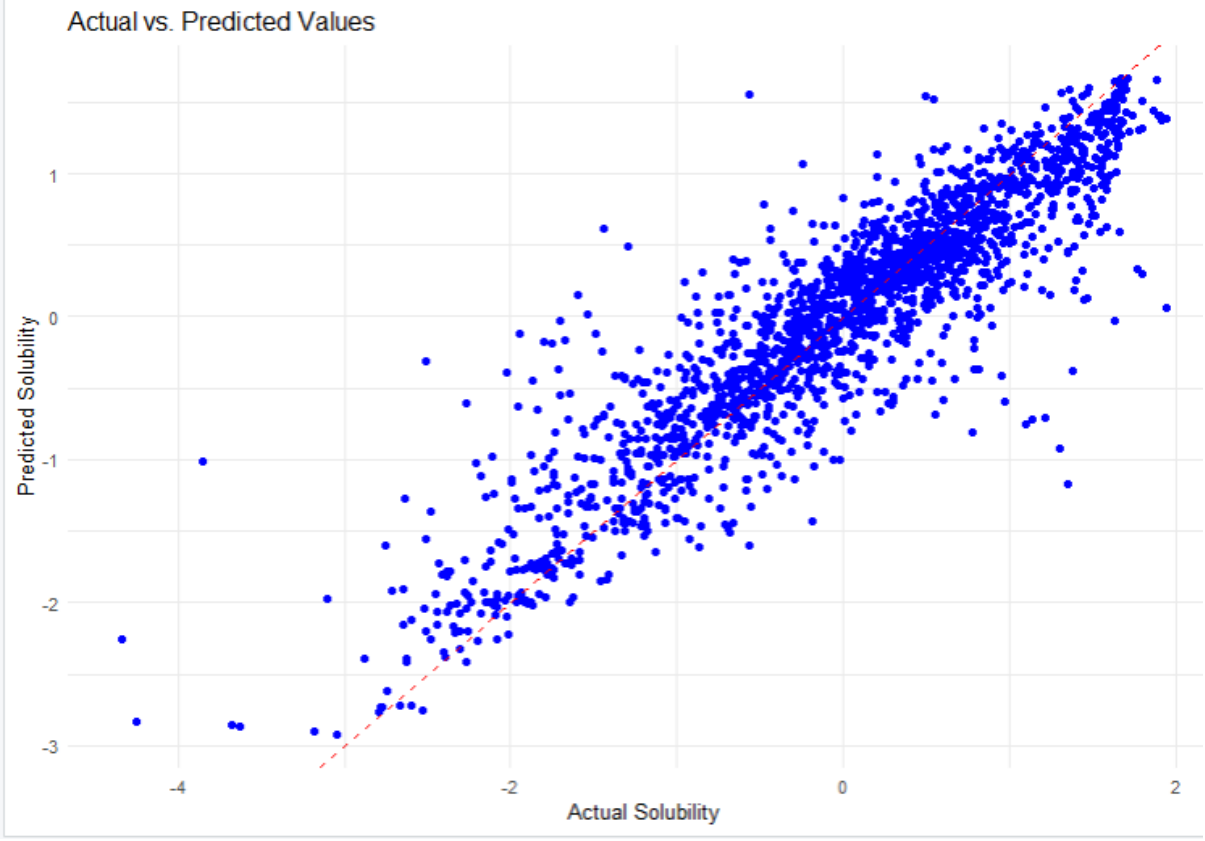
Nakon pripreme podataka prikazana je funkcija modela koja stvara slučajnu šumu. Zatim je stvorena regresijska slučajna šuma koja provodi nelinearnu višestruku regresiju. Slijedi ispis broja stabala u šumi koji se zadaje ručno, te broja varijabli koje se primjenjuje na svakom cijepanju. Naposlijetku, dane su vrijednosti preostale standardne pogreške i varijance iskazane u postotku koje pokazuju koliko dobro model odgovara korištenim podacima. Ostatci su razlika između predviđenih i stvarnih vrijednosti, a njena izuzetno mala vrijednost ukazuje na to da model vrlo kvalitetno stvara predviđanja. Vrijednost od 81.73% pokazuje da model radi dobro na danom skupu podataka.

Naredbom *importance(rf_model)* pokazuje se važnost pojedine varijable na cijeli model.

Tablica 4. Uvid u važnost varijabli

```
> importance(rf_model)
                     %IncMSE IncNodePurity
ID                 62.102631     389.61213
Name               24.332407     122.52524
InChI              25.075267     347.58071
InChIKey            1.272860      92.81182
SMILES             29.445365     215.86613
SD                 15.682918      38.90443
Ocurrences          9.161421      20.41527
Group               9.563552      14.03975
MolWt              31.806566     416.00717
MolLogP            78.690871    2784.30957
MolMR              30.467098     845.99108
HeavyAtomCount     15.655637     284.72285
NumHAcceptors      26.539688     207.40424
NumHDonors         21.617809      86.07791
NumHeteroatoms     33.050963     101.66111
NumRotatableBonds  28.889707      83.42689
NumValenceElectrons 17.274397    177.86135
NumAromaticRings   19.229159      63.94398
NumSaturatedRings  16.783277      14.96736
NumAliphaticRings  17.693233      17.80050
RingCount          18.661471      52.90762
TPSA               45.188429     366.64359
LabuteASA          19.338316     474.14412
BalabanJ           46.535624     244.00014
BertzCT            45.083722     336.18492
```

Iz tablice 4. vidimo da slučajna šuma stvara dva izlaza: smanjenje srednje kvadratne pogreške (%IncMSE) i čistoću čvora (IncNodePurity). Čistoća čvora se definira kao ukupno smanjenje preostalog broja kvadrata, što definira koliko dobro prediktor smanjuje varijancu. MSE je pouzdanija mjera važnosti varijabli. Kod MSE vrijedi pravilo da varijable većeg značaja imaju veću vrijednost MSE, u ovom slučaju to su Molekulski logaritamski koeficijent podjele(MolLogP) i identifikacijski broj spoja(ID).



Slika 12. Grafički prikaz rezultata modela slučajne šume

Kao i kod modela višestruke linearne regresije stvarne i predviđene vrijednosti topljivosti nisu snažno linearno korelirane, te model radi s određenom pogreškom. Pogreška modela iznosi RMSE=0.4362, a pogreška nakon validacije iznosi $RMSE_{10}$=0.5411.

6.2.3. UNAPRIJEDNA NEURONSKA MREŽA (FNN)

Jednako kao u prethodna dva primjera, model neuronske mreže teorijski je obrađen u prethodnom poglavlju i stoga se ovdje ne spominje. U ovom slučaju korišten je model unaprijedne neuronske mreže (FNN) umjesto modela CNN i RNN jer bolje i brže obrađuje oblik podatka kakav se nalazi u korištenom skupu podataka.

Kod za model unaprijedne neuronske mreže je prikazan na slici 13.

```
#NEURONSKA MREŽA
x <- as.data.frame(df %>% select(-Solubility))
y <- df$Solubility
data <- cbind(x, Solubility = y)
set.seed(42)
trainIndex <- createDataPartition(y, p = 0.8, list = FALSE)
trainData <- data[trainIndex, ]
testData <- data[-trainIndex, ]
# Izgradnja i treniranje  modela
set.seed(42)
nn_model <- nnet(Solubility ~ ., data = trainData, size = 10, linout = TRUE, maxit = 500)
summary(nn_model)
plotnet(nn_model, alpha = 0.6)
predictions <- predict(nn_model, testData)
rmse <- sqrt(mean((testData$Solubility - predictions)^2))
sst <- sum((testData$Solubility - mean(testData$Solubility))^2)
ssr <- sum((testData$Solubility - predictions)^2)
r2 <- 1 - (ssr / sst)
cat("Root Mean Squared Error (RMSE):", round(rmse, 4), "\n")
cat("R-kvadrat (R²):", round(r2, 4), "\n")
set.seed(42)
sampleIndex <- createDataPartition(testData$Solubility, p = 0.1, list = FALSE)
testSubset <- testData[sampleIndex, ]
predictions_subset <- predict(nn_model, testSubset)
rmse10 <- sqrt(mean((testSubset$Solubility - predictions_subset)^2))
cat("Root Mean Squared Error (RMSE) za 10% podataka:", round(rmse10, 4), "\n")
#GRAFIČKI PRIKAZ
comparison <- data.frame(Actual = testData$Solubility, Predicted = predictions)
ggplot(comparison, aes(x = Actual, y = Predicted)) +
  geom_point(color = 'blue') +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  ggtitle('Stvarne vrijednosti naspram Predikcija') +
  xlab('Stvarna topljivost') +
  ylab('Predviđena topljivost') +
  theme_minimal()
```

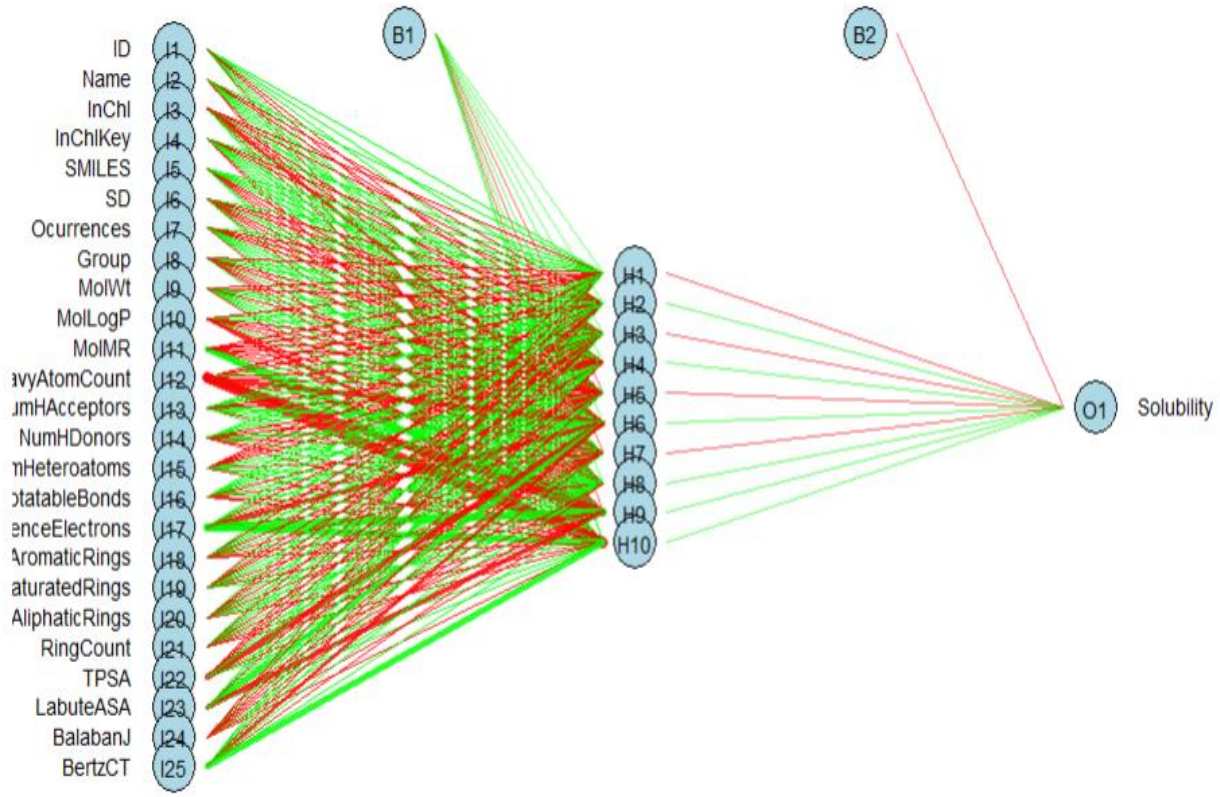
Slika 13. Kod za izgradnju unaprijedne neuronske mreže

Slika 14. prikazuje kod koji stvara neuronsku mrežu iz stvorenog modela

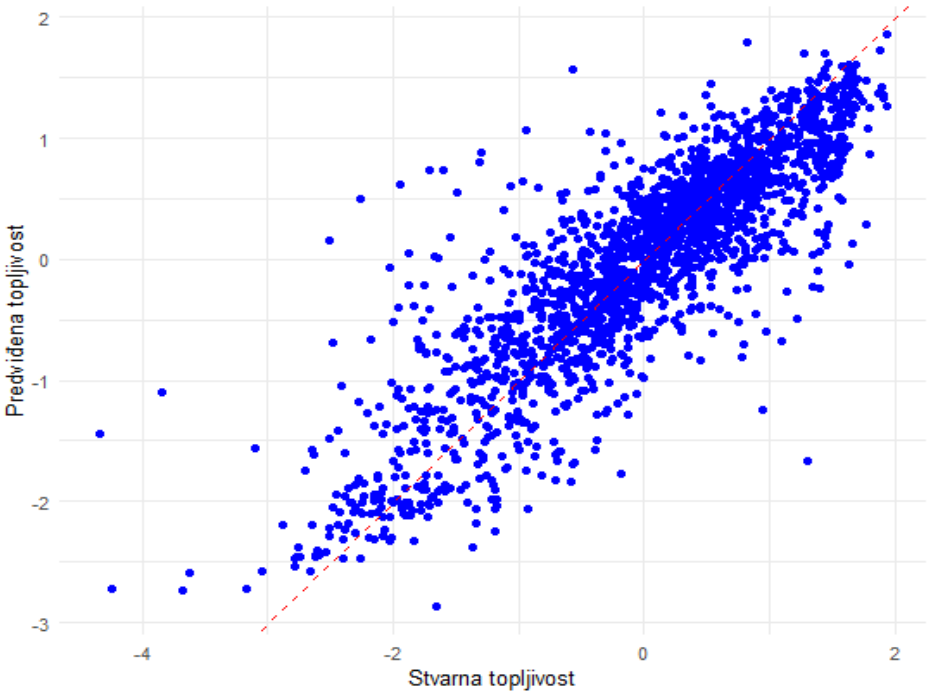
```
#MREŽA
plotnet(nn_model,
        alpha = 0.6,
        circle_col = "lightblue",  # BOJA VARIJABLE
        pos_col = "green",     # BOJA NEGATIVNIH UTEGA
        neg_col = "red",       # BOJA POZITIVNIH UTEGA
        bord_col = "black"     # BOJA GRANICA
)
```

Slika 14. Kod za stvaranje modela neuronske mreže

Na slici 15. prikazana je neuronska mreža koja se sastoji od jednog skrivenog sloja sa 10 neurona. S lijeve strane su prikazane ulazne varijable i one predstavljaju ulazni sloj neuronske mreže. Nakon njih se nalazi skriveni sloj od 10 čvorova u kojima se odvija računanje, a na kraju je izlazni sloj s jednom izlaznom varijablom.



Slika 15. Prikaz neuronske mreže

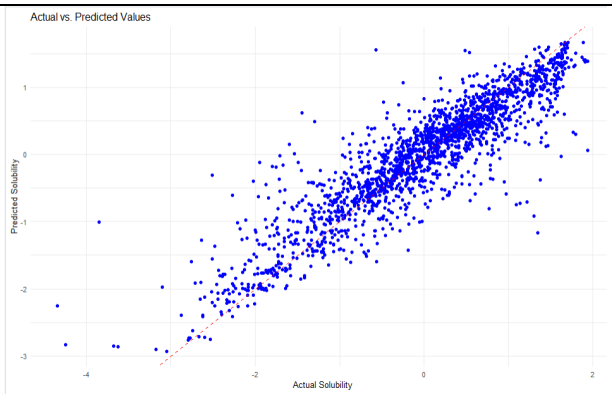
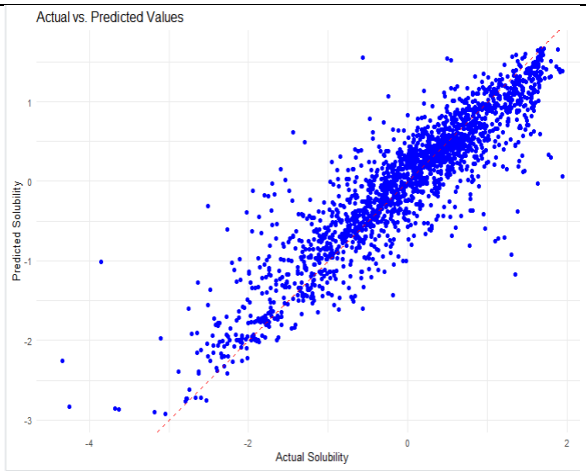
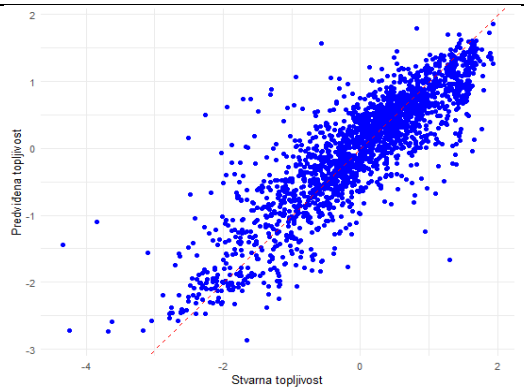


Slika 16. Grafičkih prikaz rezultata modela neuronske mreže

Na grafičkom prikazu sa slike 16. vidi se da odnos predviđene i stvarne topljivost nije u izraženoj linearnoj ovisnosti, te da opet postoji određena pogreška modela. Pogreška dobivena modelom iznosi RMSE=0.52, a pogreška dobivena validacijom iznosi $RMSE_{10}$= 0.5241. Vrijednost R^2 na ovom modelu unaprijedne neuronske mreže iznosi R^2 =0.7404.

6.3. REZULTATI

Tablica 5. Prikaz rezultata

MODEL	GRAFIČKI PRIKAZ	RMSE	$RMSE_{10}$	R^2/%
LINEARNA REGRESIJA		0.7098	0.7436	51.36
SLUČAJNA ŠUMA		0.4362	0.5411	81.73
UNAPRIJEDNA NEURONSKA MREŽA		0.52	0.5241	74.04

7. ZAKLJUČAK

U kemijskom inženjerstvu od velike je važnosti poznavati svojstva tvari, a jedno od najbitnijih svojstava je topljivost. Kako bi se pravilno i točno odredila topljivost pojedinih spojeva koriste se različiti modeli i algoritmi umjetnih neuronskih mreža i umjetne inteligencije. Pomoću umjetnih neuronskih mreža može se brže i jeftinije doći do informacija o topljivosti pojedinih organskih spojeva.

Cilj ovog je rada je bio teoretski predstaviti proces predviđanja topljivosti organskih spojeva u ovisnosti o različitim svojstvima tih spojeva, prikazati najučestalije modele i vrste umjetnih neuronskih mreža te ih primijeniti na odabrani primjer.

Podaci za rad su preuzeti za internetske stranice Kaggle, a njihovi autori su: M. C. Sorkun, A. Khetan i S. Er. Za primjenu algoritama i modela se koristio programski jezik R, te njegovo sučelje RStudio.

Rezultati dobiveni primjenom algoritama ukazuju na visoku primjenjivost ovih algoritama na predviđanje topljivosti organskih spojeva, a to je vidljivo po niskim vrijednostima pogreške dobivene modelom (RMSE). Za ovaj skup podataka najbolja predviđanja o vrijednosti topljivosti daju model slučajne šume i model unaprijednih neuronskih mreža, dok model višestruke linearne regresije stvara predviđanja s nešto većom pogreškom.

Iz dobivenih rezultata može se zaključiti da su korišteni algoritmi umjetnih neuronskih mreža primjenjivi na dani problem, te se njihovom upotrebom dobivaju prihvatljivi rezultati.

8.LITERATURA

[1] Vemula, V.R., Lagishetty, V., Lingala, S., Solubility enhancement techniques, 2010.,

https://globalresearchonline.net/journalcontents/volume5issue1/article-007.pdf

(Pristup: 4.9.2024.)

[2] What is pKa?, https://chemistrytalk.org/what-is-pka/ (Pristup: 4.9.2024.)

[3] Sorkun, M.C., Khetan, A., Er, S., AqSolDB, a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds. *Sci Data* **6**, 143 (2019)., https://www.nature.com/articles/s41597-019-0151-1 (Pristup: 4.9.2024.)

[4] Sohun, M.C., AqSolDB: A curated aqueous solubility dataset, https://www.kaggle.com/datasets/sorkun/aqsoldb-a-curated-aqueous-solubility-dataset (Pristup: 4.9.2024.)

[5] Lantz, B., Machine learning with R, Third edition, 2019., 239-245

[6] izv. prof. dr. sc., Ujević Andrijić, Ž., Umjetne neuronske mreže. Kem. Ind., 2019, 68, 219-220., https://hrcak.srce.hr/file/322233 (Pristup: 4.9.2024.)

[7] Krenker A., Bešter J., Kos A., Introduction to artificial neural networks, University of Ljubljana, 2011.,

https://www.researchgate.net/profile/Kenji-Suzuki-2/publication/319316102_Artificial_Neural_Networks_-_Methodological_Advances_and_Biomedical_Applications/links/59a42f16aca272a6461bb35e/Artificial-Neural-Networks-Methodological-Advances-and-Biomedical-Applications.pdf#page=15

(Pristup: 4.9.2024.)

[8] Prikaz neurona,

https://www.researchgate.net/figure/The-biological-neuron_fig2_320384373

(Pristup: 5.9.2024.)

[9] Introduction to ANN | Set 4 (Network Architectures),

https://www.geeksforgeeks.org/introduction-to-ann-set-4-network-architectures/

(Pristup: 6.9.2024.)

[10] https://www.sqlshack.com/implement-artificial-neural-networks-anns-in-sql-server/

(Pristup: 4.9.2024.)

[11] Vijay Kanade, What is artificial intelligence,

https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ai/

(Pristup: 6.9.2024.)

[12] Flasinski, M., Introduction to artificial intelligence, 2016., https://books.google.hr/books?hl=hr&lr=&id=UpvvDAAAQBAJ&oi=fnd&pg=PR5&dq=flasinski+artificial+intelligence&ots=5x-2gTj4ft&sig=CaXXgqigxokb-tRY52do_9O8jLE&redir_esc=y#v=onepage&q=flasinski%20artificial%20intelligence&f=false

(Pristup: 4.9.2024.)

[13] History of Artificial Intelligence, https://www.javatpoint.com/history-of-artificial-intelligence

(Pristup: 4.9.2024.)

[14] Burkov, A., The hundred-page machine learning, 2019.

[15] Keith, J.A, Vassilev-Galindo, V., Cheng, B., Chmiela, S., Gastegger M., Müller, K.-R., Tkatchenko A., Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems, 2021., https://pubs.acs.org/doi/10.1021/acs.chemrev.1c00107?ref=pdf (Pristup: 4.9.2024.)

[16] Alyapadin E., Introduction to machine learning, Third edition, 2014., https://dl.matlabyar.com/siavash/ML/Book/Ethem%20Alpaydin-Introduction%20to%20Machine%20Learning-The%20MIT%20Press%20(2014).pdf

(Pristup: 4.9.2024.)

[17] R Programming Language, https://www.geeksforgeeks.org/r-programming-language-introduction/ (Pristup: 4.9.2024.)

[18] Službena stranica R programskog jezika, https://www.r-project.org/about.html

(Pristup: 4.9.2024.)

[19] Logo programskog jezika R, https://www.r-project.org/about.html (Pristup: 4.9.2024.)

[20] Beers, B., Regression definition, https://www.investopedia.com/terms/r/regression.asp (Pristup: 4.9.2024.)

[21] IBM, What is linear regression ?, https://www.ibm.com/topics/linear-regression

(Pristup: 4.9.2024.)

[22] Kanade, V., What is logistic regression?, https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/ (Pristup: 4.9.2024.)

[23] Agrawal, R., Polynomnal regression for beginners?, https://www.analyticsvidhya.com/blog/2021/07/all-you-need-to-know-about-polynomial-regression/ (Pristup: 4.9.2024.)

[24] Prikaz višestruke linearne regresije,

https://medium.com/analytics-vidhya/multiple-linear-regression-an-intuitive-approach-f874f7a6a7f9 (Pristup: 5.9.2024.)

[25] Kanth, R., How Does Random Forest Work?, https://www.analyticsvidhya.com/blog/2023/02/how-does-random-forest-work/

(Pristup: 7.9.2024.)

[26] Prikaz modela slučajne šume, https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/ (Pristup: 5.9.2024.)

[27] Feedforward neural network, https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network (Pristup: 6.9.2024.)

[28] Dijagram jednog čvora, https://medium.com/@nlunge786/a-deep-architecture-multi-layer-perceptron-164bc5ff3842 (Pristup:7.9.2024.)

9. PRILOZI

```
# INSTALIRAVANJE POTREBNIH PAKETA

install.packages("randomForest")

install.packages("dplyr")

install.packages("ggplot2")

 install.packages("caret")

install.packages("nnet")

install.packages("NeuralNetTools")

# UČITAVANJE PAKETA

library(nnet)

library(dplyr)

library(caret)

library(ggplot2)

library (NeuralNetTools)

library(randomForest)

# PRIPREMA PODATAKA

df<- read.csv("C:\\Users\\Korisnik\\Desktop\\curated-solubility-dataset.csv")

 df <- df %>%

            mutate(across(where(is.character), as.factor)) %>%

            mutate(across(where(is.factor), as.numeric)) %>%

            na.omit()
```

```
df <- df %>%
        mutate(across(where(is.numeric), scale))
# VIŠESTRUKA LINEARNA REGRESIJA
set.seed(42)
 trainIndex <- createDataPartition(df$Solubility, p = 0.8, list = FALSE)
 trainData <- df[trainIndex, ]
 testData <- df[-trainIndex, ]
 input_columns <- setdiff(names(df), "Solubility")
 formula <- as.formula(paste("Solubility ~", paste(input_columns, collapse = " + ")))
 lm_model <- lm(formula, data = trainData)
 summary(lm_model)
predictions <- predict(lm_model, newdata = testData)
actuals <- testData$Solubility
mse <- mean((actuals - predictions)^2)
rmse <- sqrt(mse)
sst <- sum((actuals - mean(actuals))^2)
ssr <- sum((actuals - predictions)^2)
r2 <- 1 - (ssr / sst)
first_10_actuals <- actuals[1:10]
first_10_predictions <- predictions[1:10]
mse10 <- mean((first_10_actuals - first_10_predictions)^2)
```

```
 rmse10 <- sqrt(mse10)

cat("Root Mean Squared Error (RMSE):", round(rmse, 4), "\n")

cat("Root Mean Squared Error for the first 10 observations (RMSE10):", round(rmse10, 4), "\n")

cat("R-squared (R²):", round(r2, 4), "\n")

comparison <- data.frame(Actual = actuals, Predicted = predictions)

ggplot(comparison, aes(x = Actual, y = Predicted)) +

          geom_point(color = 'blue') +

          geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +

          ggtitle('Actual vs. Predicted Values') +

          xlab('Actual Solubility') +

          ylab('Predicted Solubility') +

          theme_minimal()

# SLUČAJNA ŠUMA

set.seed(42)

 trainIndex <- createDataPartition(df$Solubility, p = 0.8, list = FALSE)

 trainData <- df[trainIndex, ]

testData <- df[-trainIndex, ]

rf_model <- randomForest(Solubility ~ ., data = trainData, importance = TRUE, ntree = 500)

print(rf_model)

varImpPlot(rf_model)

predictions <- predict(rf_model, newdata = testData)
```

```
 actuals <- testData$Solubility

mse <- mean((actuals - predictions)^2)

 rmse <- sqrt(mse)

sst <- sum((actuals - mean(actuals))^2)

ssr <- sum((actuals - predictions)^2)

r2 <- 1 - (ssr / sst)

first_10_actuals <- actuals[1:10]

first_10_predictions <- predictions[1:10]

mse10 <- mean((first_10_actuals - first_10_predictions)^2)

 rmse10 <- sqrt(mse10)

cat("Root Mean Squared Error (RMSE):", round(rmse, 4), "\n")

cat("Root Mean Squared Error for the first 10 observations (RMSE10):", round(rmse10, 4), "\n")

cat("R-squared (R²):", round(r2, 4), "\n")

comparison <- data.frame(Actual = actuals, Predicted = predictions)

ggplot(comparison, aes(x = Actual, y = Predicted)) +

            geom_point(color = 'blue') +

            geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +

           ggtitle('Actual vs. Predicted Values') +

            xlab('Actual Solubility') +

            ylab('Predicted Solubility') +

            theme_minimal()

> print(rf_model)
```

```
# UNAPRIJEDNE NEURONSKE MREŽE (FNN)

x <- as.data.frame(df %>% select(-Solubility))

y <- df$Solubility

data <- cbind(x, Solubility = y)

set.seed(42)

trainIndex <- createDataPartition(y, p = 0.8, list = FALSE)

trainData <- data[trainIndex, ]

testData <- data[-trainIndex, ]

set.seed(42)

nn_model <- nnet(Solubility ~ ., data = trainData, size = 10, linout = TRUE, maxit = 500)

summary(nn_model)

plotnet(nn_model, alpha = 0.6)

predictions <- predict(nn_model, testData)

rmse <- sqrt(mean((testData$Solubility - predictions)^2))

sst <- sum((testData$Solubility - mean(testData$Solubility))^2)

ssr <- sum((testData$Solubility - predictions)^2)

r2 <- 1 - (ssr / sst)

cat("Root Mean Squared Error (RMSE):", round(rmse, 4), "\n")

cat("R-kvadrat (R²):", round(r2, 4), "\n")

set.seed(42)

sampleIndex <- createDataPartition(testData$Solubility, p = 0.1, list = FALSE)
```

```
testSubset <- testData[sampleIndex, ]

predictions_subset <- predict(nn_model, testSubset)

rmse10 <- sqrt(mean((testSubset$Solubility - predictions_subset)^2))

cat("Root Mean Squared Error (RMSE) za 10% podataka:", round(rmse10, 4), "\n")

comparison <- data.frame(Actual = testData$Solubility, Predicted = predictions)

ggplot(comparison, aes(x = Actual, y = Predicted)) +

 geom_point(color = 'blue') +

 geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +

 ggtitle('Stvarne vrijednosti naspram Predikcija') +

 xlab('Stvarna topljivost') +

 ylab('Predviđena topljivost') +

 theme_minimal()

plotnet(nn_model, alpha = 0.6,

        circle_col = “lightblue”, #BOJA VARIJABLE

        pos_col = “green”, #BOJA NEGATIVNIH UTEGA

        neg_col = “red”, #BOJA POZITIVNIH UTEGA

        bord_col = “black”, #BOJA GRANICA

)

predictions <- predict (nn_model, testData)
```