

Model za predviđanje cijena dionica pomoću metoda strojnog učenja

Iljić, Luka

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Economics and Business in Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Ekonomski fakultet u Osijeku**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:145:130760>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-22**



Repository / Repozitorij:

[EFOS REPOSITORY - Repository of the Faculty of Economics in Osijek](#)



Sveučilište Josipa Jurja Strossmayera
Ekonomski fakultet u Osijeku
Diplomski studij smjer Poslovna informatika

Luka Iljić

**MODEL ZA PREDVIĐANJE CIJENA DIONICA POMOĆU
METODA STROJNOG UČENJA**

Diplomski rad

Osijek, 2024.

Sveučilište Josipa Jurja Strossmayera
Ekonomski fakultet u Osijeku
Diplomski studij smjer Poslovna informatika

Luka Iljić

**MODEL ZA PREDVIĐANJE CIJENA DIONICA POMOĆU
METODA STROJNOG UČENJA**

Diplomski rad

Kolegij: Sustavi poslovne inteligencije

JMBAG: 0010210776

e-mail: itzluka@gmail.com

Mentor: doc.dr.sc. Domagoj Ševerdija

Komentor: Dr. sc. Adela Has

Osijek, 2024.

Josipa Juraj Strossmayer University of Osijek
Faculty of Economics and Business in Osijek
University Graduate Study in Business Informatics


Luka Iljić

**A MODEL FOR FORECASTING STOCK PRICES USING
MACHINE LEARNING METHODS**

Graduate paper

Osijek, 2024.

IZJAVA
O AKADEMSKOJ ČESTITOSTI,
PRAVU PRIJENOSA INTELEKTUALNOG VLASNIŠTVA,
SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA
I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA

1. Kojom izjavljujem i svojim potpisom potvrđujem da je DIPLOMSKI (navesti vrstu rada: završni / diplomski / specijalistički / doktorski) rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljeni literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom *Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska*. 
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucionalnom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o visokom obrazovanju i znanstvenoj djelatnosti, NN 119/2022).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

Ime i prezime studenta/studentice: Luka Iljić

JMBAG: 0010210776

OIB: 31996911181

e-mail za kontakt: itzluka@gmail.com

Naziv studija: Poslovna informatika

Naslov rada: Model za predviđanje cijena dionica pomoću metoda strojnog učenja

Mentor/mentorica rada: Doc. Dr. sc. Domagoj Ševerdija

U Osijeku, 16.09.2024. godine

Potpis



Model za predviđanje cijena dionica pomoću metoda strojnog učenja

SAŽETAK

Umjetna inteligencija je pojam koji se sve češće spominje u zadnjih desetak godina, gotovo da ne postoji osoba koja nije čula ili se pak susrela s tim pojmom. U posljednje vrijeme sve se više daje na važnosti umjetnoj inteligenciji i njenom razvoju. U razvoju umjetne inteligencije važnu ulogu ima strojno učenje. Metode strojnog učenja se uveliko primjenjuju u raznim područjima poznatima čovjeku. Od primjene u medicini, obrazovanju, proizvodnji pa sve do zabavne industrije.

Tako se umjetna inteligencija učestalo koristi u brojnim ekonomskim područjima, npr. za predviđanje rizika, predviđanje cijena određenih proizvoda, predviđanje cijena dionica i slično.

Zato se moderna i tehnološki orijentirana poduzeća uvelike okreću primjeni i unaprjeđenju umjetne inteligencije kako bi što brže i efikasnije predvidjele i minimizirale rizike, te lakše prepoznale prilike koje se pružaju u okolini koja svakim danom postaje sve dinamičnija.

U ovom radu bit će prikazan povijesni razvoj strojnog učenja, metode koje se koriste i duboko učenje. Zatim će biti objašnjene metode strojnog učenja poput logičke regresije, neuronskih mreža i k-najbližih susjeda, te će se na temelju dobivenih rezultata pokušati utvrditi koja metoda je najbolja za predviđanje cijena dionica. Analiza i modeliranje će biti izvršeni uz pomoć programskog alata Python.

Ključne riječi: umjetna inteligencija, strojno učenje, duboko učenje, neuronske mreže, logistička regresija, k-najbližih susjeda

A model for forecasting stock prices using machine learning methods

ABSTRACT

Artificial intelligence is a term that has been mentioned more and more often in the last ten years, there is hardly a person who has not heard or encountered this term. Recently, more and more attention has been given to artificial intelligence and its development. Machine learning plays an important role in the development of artificial intelligence. Machine learning methods are widely applied in various fields known to man. From application in medicine, education, production to the entertainment industry.

Thus, artificial intelligence is often used in a number of economic fields, for example, for predicting risks, predicting the prices of certain products, predicting stock prices, and the like.

That is why modern and technologically oriented companies are largely turning to the application and improvement of artificial intelligence in order to predict and minimize risks as quickly and efficiently as possible, and to more easily recognize the opportunities offered in an environment that is becoming more and more dynamic every day.

In this paper, the historical development of machine learning, the methods used and deep learning will be presented. Then, machine learning methods such as logical regression, neural networks and k-nearest neighbors will be explained, and based on the obtained results, an attempt will be made to determine which method is the best for predicting stock prices. Analysis and modeling of the project will be performed with the help of the Python programming tool.

Keywords: artificial intelligence, machine learning, deep learning, neural networks, logical regression, k-nearest neighbors

SADRŽAJ

1. UVOD.....	1
2. Teorijska podloga i prethodna istraživanja	2
2.1. Pojava i razvoj umjetne inteligencije	2
2.2. Suvremena umjetna inteligencija	5
2.3. Prednosti i nedostaci umjetne inteligencije	8
3. Metodologija rada.....	10
3.1. Strojno učenje.....	10
3.2. Proces strojnog učenja.....	12
3.3. Vrste i modeli strojnog učenja	15
3.3.1. Nadzirano strojno učenje	15
3.3.1.1. <i>KNN algoritam</i>	16
3.3.1.2. <i>Logistička regresija</i>	17
3.3.1.3. <i>Neuronske mreže</i>	18
3.3.2. Nenadzirano strojno učenje.....	19
3.3.2.1. <i>Primjena nenadziranog strojnog učenja</i>	20
3.3.3. Polunadzirano strojno učenje	21
3.4. Duboko učenje.....	23
3.5. Python	26
3.5.1. Python biblioteke	26
3.5.2. Što su Python biblioteke	26
4. Opis istraživanja i rezultati istraživanja	27
4.1. Opis korištenih podataka	27
4.2. Pripremanje podataka.....	28
4.3. Kreiranje modela	40
5. Rasprava	48
6. Zaključak.....	49
7. Literatura	50

1. UVOD

Strojno učenje je jedna od najbrže rastućih tehnologija današnjice koja je našla široku primjenu u mnogobrojnim industrijama. Strojno učenje je grana umjetne inteligencije i računalne znanosti uz pomoć koje se prikupljaju, klasificiraju, sortiraju, te na samom kraju obrađuju podaci pomoću kojih sustav uči na temelju iskustva. Osnova strojnog učenja je da na temelju prikupljenih podataka imitira način na koji ljudi uče i s vremenom povećava preciznost i brzinu kojim radi. Sve to se koristi radi što bržeg i efikasnijeg donošenja odluka, ali i donošenja odluka na temelju informacija koje čovjek nije sposoban sam dobiti. Cilj ovog rada je izrada klasifikacijskog modela za kupnju dionica, koji će na temelju povijesnih podataka i odabranih parametara odrediti i pomoći u donošenju odluka o kupnji dionica. Na temelju dobivenih modela, donijet će se zaključak o tome koji model predloženih modela ima najveću stopu točnosti klasifikacije.

U narednim poglavljima će se najprije objasniti povijest koncepta umjetne inteligencije, njezin razvoj kao suvremena tehnologija i sveopća implementacija u modernom društvu, te koje su njene prednosti i mane. U metodologiji rada će se detaljnije objasniti pojam strojnog učenja, što ono predstavlja, koje su vrste i tipovi i na koji način funkcionira. Zatim će se preći na opis istraživanja i rezultate istraživanja gdje će se definirati podaci korišteni u izradi i objasniti kreirani model na temelju kojeg će se donijeti zaključci.

U poglavlju rasprave će se prokomentirati prethodno dobiveni rezultati, te se na temelju istih donijeti sud o prednostima i nedostacima samog modela, te koja su to poboljšanja koja se mogu implementirati u model kako bi se dobili bolji rezultati u budućnosti. U posljednjem poglavlju će se iznijeti završni zaključci na temelju dobivenih rezultata.

2. Teorijska podloga i prethodna istraživanja

2.1. Pojava i razvoj umjetne inteligencije

Razvoj umjetne inteligencije često se percipira kao relativno nov fenomen, s uvjerenjem da je riječ o tehnologiji koja je nastala u 21. stoljeću. Razumljivo je da postoji takva percepcija u društvu ako uzmemo u obzir činjenicu da se masovna primjena umjetne inteligencije počela dešavati u proteklih desetak godina. Iako je koncept umjetne inteligencije te njene metode i tehnike bile poznate i ranije, u posljednjem desetljeću dolazi do svojevrsne revolucije. Umjetna inteligencija je postala sastavni dio naše svakodnevnice, gotovo svaka aplikacija koju skinemo na svoje pametne mobitele koristi neki oblik takve tehnologije. Od aplikacija koje pružaju usluge ili koje služe za učenje, pa sve do onih koje koristimo u svoje slobodno vrijeme, odnosno za zabavu, sve one koriste tehnologiju koju danas nazivamo umjetnom inteligencijom.

Međutim, kako bi se bolje razumio koncept umjetne inteligencije njezin razvoj se mora sagledati od samog začetka ideje koja seže preko 100 godina unazad na početak 20. stoljeća.

Godine 1921. češki dramatičar Karel Čapek objavio je znanstvenofantastičnu dramu "Rossumovi univerzalni roboti" koja je predstavila ideju "umjetnih ljudi" koje je nazvao roboti. Ovo je bila prva poznata upotreba riječi (Anon., 2022). 1929. Japanski profesor Makoto Nishimura napravio je prvog japanskog robota, nazvanog Gakutensoku. 20 godina nakon prvog robota, 1949. godine, računalni znanstvenik Edmund Callis Berkley objavio je knjigu "Divovski mozgovi ili strojevi koji misle" u kojoj je novije modele računala usporedio s ljudskim mozgovima. (Anon., 2022)

Do sredine 20. stoljeća umjetna inteligencija je još bila u svojem ranom razvoju i do tada strojno učenje još nije postojalo, dakle može se reći da je umjetna inteligencija do 1950. bila poprilično primitivna u odnosu na razdoblje koje dolazi u drugoj polovici 20. stoljeća. Razlog tome je pojava strojnog učenja kojim se uvelike ubrzao proces razvoja umjetne inteligencije. Strojno učenje je prvo začeto iz matematičkog modeliranja neuronskih mreža. Rad logičara Waltera Pittsa i neuroznanstvenika Warrena McCullocha, objavljen 1943., pokušao je matematički mapirati misaone procese i donošenje odluka u ljudskoj spoznaji (Chai, 2020). Godine 1950. Alan Turing predložio je Turingov test, koji je postao lakmus test na temelju kojeg se strojevi smatraju "inteligentnima" ili "neinteligentnima". Kriterij da stroj dobije status "inteligentnog" stroja bio je

da ima sposobnost uvjeriti ljudsko biće da je on, stroj, također ljudsko biće. Ubrzo nakon toga, ljetni istraživački program na Dartmouth Collegeu postao je službeno mjesto rođenja umjetne inteligencije (Chai, 2020).

Od tog trenutka počeli su se pojavljivati "inteligentni" algoritmi za strojno učenje i računalni programi, koji rade sve, od planiranja putnih ruta za prodavače do igranja društvenih igara s ljudima kao što su dame i križić-kružić. Inteligentni strojevi počeli su raditi sve, od korištenja prepoznavanja govora, učenja izgovaranja riječi na način na koji bi to beba učila pa sve do pobjede nad svjetskim prvakom u šahu u igri šaha (Chai, 2020).

Frank Rosenblatt bio je psiholog koji je najpoznatiji po svom radu na strojnom učenju. Godine 1957. razvio je „Perceptron“, koji je algoritam strojnog učenja. Perceptron je bio jedan od prvih algoritama koji je koristio umjetne neuronske mreže, široko korištene u strojnom učenju. (Koch, 2022.) Dizajniran je za poboljšanje točnosti računalnih predviđanja. Cilj Perceptra bio je učiti iz podataka prilagođavanjem njegovih parametara dok ne dođe do optimalnog rješenja. Svrha Perceptra bila je olakšati računalima učenje iz podataka i poboljšati prethodne metode koje su imale ograničen uspjeh (Koch, 2022). 10 godina nakon Perceptra kreiran je poznatiji algoritam K-najbližih susjeda. Algoritam najbližeg susjeda razvijen je kao način za automatsko identificiranje uzoraka unutar velikih skupova podataka. Cilj ovog algoritma je pronaći sličnosti između dviju stavki i odrediti koja je bliža uzorku koji se nalazi u drugoj stavci. To se može koristiti za stvari poput pronalaženja odnosa između različitih dijelova podataka ili predviđanja budućih događaja na temelju prošlih događaja (Koch, 2022).

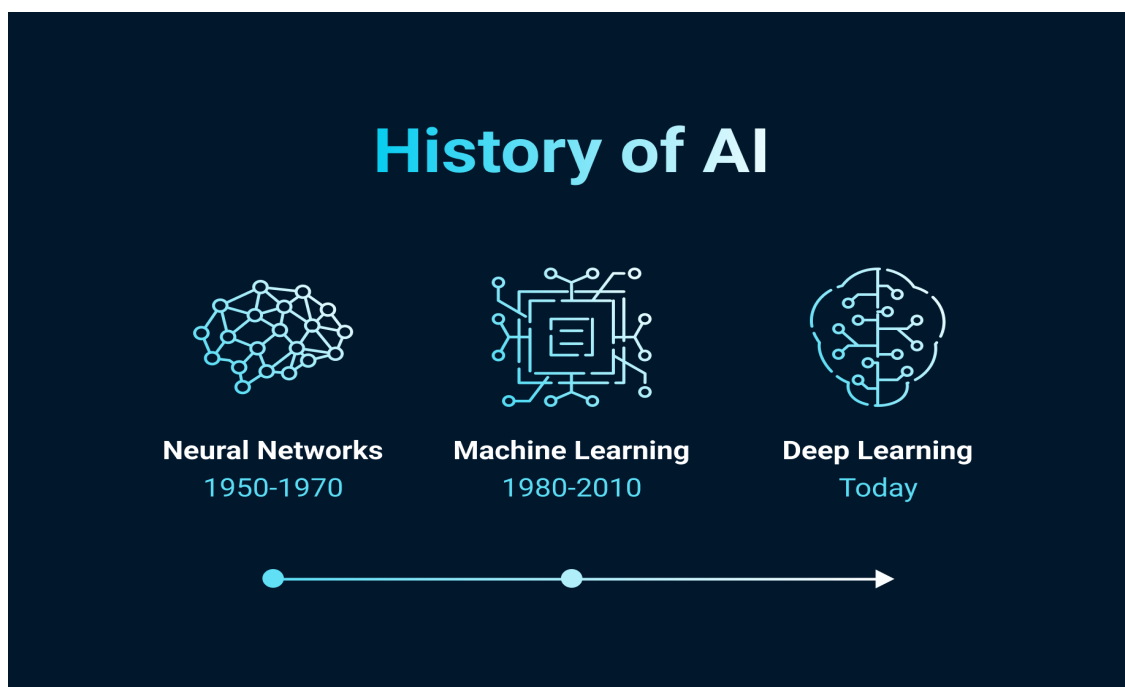
Godine 1967. Cover i Hart objavili su članak o "Klasificiranju uzoraka najbližeg susjeda". To je metoda induktivne logike koja se koristi u strojnom učenju za klasificiranje ulaznog objekta u jednu od dvije kategorije. Uzorak klasificira iste stavke koje su klasificirane u iste kategorije kao i njegovi najbliži susjedi. Ova se metoda koristi za klasificiranje objekata s nizom atributa, od kojih su mnogi kategorički ili numerički i mogu imati preklapajuće vrijednosti. (Koch, 2022.)

Sve do kasnih 1970-ih, strojno učenje bilo je samo još jedna komponenta napretka umjetne inteligencije. Zatim se odvojilo i evoluiralo samo od sebe, jer se strojno učenje pojavilo kao važna

funkcija u računalstvu u oblaku i e-trgovini. Strojno učenje je vitalni pokretač u mnogim područjima vrhunske tehnologije našeg vremena. Znanstvenici trenutno rade na pristupima kvantnog strojnog učenja. (Selman, 2022.)

Dakle, možemo zaključiti da je razvoj umjetne inteligencije uvelike ovisio o strojnom učenju, odnosno, ono je igralo ključnu ulogu u napretku umjetne inteligencije kakve je mi poznajemo danas, te je otključalo vrata mnogih drugih mogućnosti te napredne tehnologije.

Na slici 1. vidi se grafika koja prikazuje povijest umjetne inteligencije kroz njezine razvojne faze, odnosno tehnologije koje su tada bile u primarnom fokusu. Sredinom prošlog stoljeća primarni fokus je bio na neuronskim mrežama, zatim se prešlo na strojno učenje i naposljetku na duboko učenje.



Slika 1. Povijest umjetne inteligencije

Izvor: <https://omq.ai/blog/history-of-ai/>

2.2. Suvremena umjetna inteligencija

Najjednostavnijim riječima ona se može definirati kao tehnologija koja ima sposobnost imitirati način na koji ljudi razmišljaju i zaključuju. Umjetna inteligencija (AI) tehnologija je koja računalima i strojevima omogućuje simulaciju ljudskog učenja, razumijevanja, rješavanja problema, donošenja odluka, kreativnosti i autonomije (Stryker, Kavlakoglu, 2024).

Umjetna inteligencija (AI) je sposobnost digitalnog računala ili računalno kontroliranog robota da obavlja zadatke koji se obično povezuju s inteligentnim bićima. Izraz se često primjenjuje na projekt razvoja sustava obdarenih intelektualnim procesima karakterističnim za ljude, kao što je sposobnost rasuđivanja, otkrivanja značenja, generaliziranja ili učenja iz prošlih iskustava (Copeland, 2023).

Kada se govori o podjeli ili vrstama umjetne tehnologije, ona se dijeli na 4 vrste odnosno razine. Svaka od te 4 vrste umjetne inteligencije funkcionira na različiti način i razlikuju se po stupnju kompleksnosti.

Dakle umjetnu tehnologiju može se podijeliti na 4 vrste: (Marr, 2021)

1. Reaktivni strojevi
2. Umjetna inteligencija s ograničenim pamćenjem
3. Teorija uma
4. Samosvjesna umjetna inteligencija

Najosnovnija vrsta umjetne inteligencije je reaktivna umjetna inteligencija, koja je programirana za pružanje predvidljivog izlaza na temelju inputa koji prima. Reaktivni strojevi uvijek reagiraju na identične situacije na potpuno isti način svaki put i nisu u stanju naučiti radnje ili zamisliti prošlost ili budućnost.

Primjeri reaktivne umjetne inteligencije uključuju:

- Deep Blue, IBM-ovo superračunalo za igranje šaha koje je pobijedilo svjetskog prvaka Garija Kasparova
- Filtri neželjene pošte za našu e-poštu koji sprječavaju promocije i pokušaje krađe identiteta izvan naše pristigle pošte

- Netflixov mehanizam za preporuke

Reaktivna umjetna inteligencija bila je ogroman korak naprijed u povijesti razvoja umjetne inteligencije, ali te vrste umjetne inteligencije ne mogu funkcionirati izvan zadataka za koje su izvorno dizajnirane. To ih čini inherentno ograničenima i zrelima za poboljšanje. Znanstvenici su razvili sljedeću vrstu umjetne inteligencije na temelju toga (Marr, 2021).

AI s ograničenim pamćenjem uči iz prošlosti i gradi iskustveno znanje promatrajući akcije ili podatke. Ova vrsta umjetne inteligencije koristi povijesne podatke promatranja u kombinaciji s unaprijed programiranim informacijama za izradu predviđanja i izvođenje složenih zadataka klasifikacije. To je danas najraširenija vrsta umjetne inteligencije. Primjerice, autonomna vozila koriste AI s ograničenom memorijom za promatranje brzine i smjera drugih automobila, pomažući im da "čitaju cestu" i prilagođavaju se po potrebi. Ovaj postupak za razumijevanje i tumačenje dolaznih podataka čini ih sigurnijima na cestama. Međutim, AI s ograničenom memorijom – kao što mu ime govori – još uvijek je ograničen. Podaci s kojima rade autonomna vozila prolazni su i ne pohranjuju se u dugoročnoj memoriji automobila (Marr, 2021).

Ukoliko se želi smisleno komunicirati s robotom koji se ponaša i izgleda kao ljudsko biće s emocijama, onda se tu javlja umjetna inteligencija bazirana na teoriji uma. S ovom vrstom umjetne inteligencije, strojevi će steći istinske sposobnosti donošenja odluka koje su slične ljudima. Strojevi s teorijom uma moći će razumjeti i zapamtiti emocije, zatim prilagoditi ponašanje na temelju tih emocija dok komuniciraju s ljudima.

Još uvijek postoje brojne prepreke u postizanju teorije uma, jer je proces promjene ponašanja temeljen na brzom promjeni emocija tako fluidan u ljudskoj komunikaciji. Teško ga je oponašati dok pokušavamo stvoriti sve više i više emocionalno inteligentnijih strojeva.

Glava robota Kismet, razvijena od profesorice Cynthije Breazeal, mogla je prepoznati emocionalne signale na ljudskim licima i replicirati te emocije na vlastito lice. Humanoidni robot Sophia, razvijen od strane Hanson Roboticsa u Hong Kongu, može prepoznati lica i reagirati na interakcije vlastitim izrazima lica (Marr, 2021).

Teorija uma je još uvijek u procesu intenzivnog istraživanja i razvoja; stoga će se njezine

karakteristike nastaviti razvijati s vremenom, no na temelju trenutnih podataka mogu se očekivati sljedeće karakteristike: (Arya, 2024.)

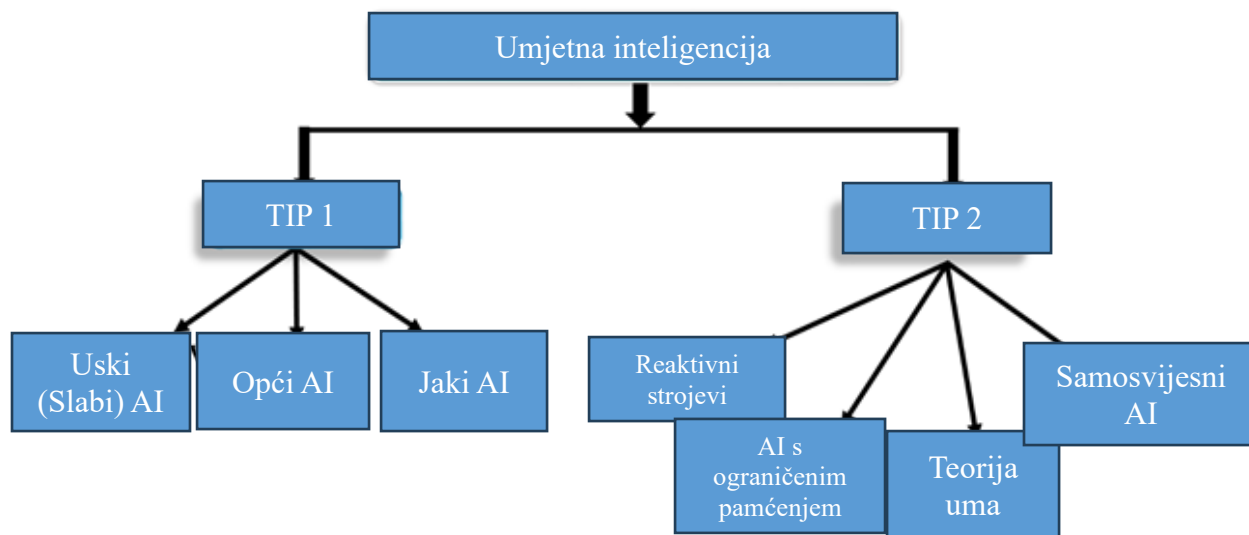
- Sustavi bazirani na teoriji uma će moći razumjeti važnost njihove svijesti i različite posljedice do kojih bi to moglo dovesti
- Roboti će bolje komunicirati s ljudskim bićima od sadašnje generacije umjetne inteligencije, koja ne može objasniti svoje postupke
- Teorija uma bit će implementirana sa sustavom strojnog učenja koji može objasniti odluke na različitim jezicima, pomažući korisniku da bolje razumije
- Roboti bi trebali moći razumjeti namjeru drugih sličnih robota ili sustava temeljenim na teoriji uma

Samosvjesna umjetna inteligencija je najnaprednija vrsta umjetne inteligencije, ona je, barem u teoriji, kao što joj sam naziv sugerira, svjesna svoje okoline i ima sposobnost da doživi subjektivna iskustva u interakciji s okolinom. Ona posjeduje ljudske karakteristike kao što su; samosvijest, kreativnost i sposobnost osjećanja istinskih emocija poput radosti ili straha. Također ima moć učenja, prilagodbe, pa čak i da stvara međuljudske odnose s ljudima i drugim samosvjesnim sustavima (Glover, 2024).

Ovu vrstu iznimno sofisticirane umjetne inteligencije smatra se najnaprednijom i još uvijek je u ranoj fazi razvoja. Naime, još ne postoje dovoljno napredni i moćni hardveri ili algoritmi koji bi podržali ovakvu vrstu umjetne inteligencija.

Slika 2. prikazuje glavnu podjelu umjetne inteligencija na 2 osnovna tipa- Tip 1 i Tip 2.

Tip 1 se dijeli na vrste umjetne inteligencije ovisno na njihovu kompleksnost, dok Tip 2 pokazuje koje su to vrste umjetne inteligencije u primjeni.



Slika 2. Podjela umjetne inteligencije na 2 tipa

Izvor: <https://www.javatpoint.com/types-of-artificial-intelligence>

2.3. Prednosti i nedostaci umjetne inteligencije

U posljednjih desetak godina tehnologija je eksponencijalno napredovala zbog umjetne inteligencije i nedvojbeno je da će nastaviti u istom smjeru koji vodi ka usavršavanju AI-a.

Ne tako davno, do prije par desetljeća, umjetna inteligencija se smatrala kao fenomen iz daleke budućnosti, no ona je ipak, brže nego očekivano, postala sastavni dio stvarnosti. Gdje god se krene, svjesno ili nesvjesno, susreće se s ovim fenomenom. Primjeri poput personaliziranih filmskih preporuka na platformama poput Netflix-a, osobnih asistenata poput Siri ili Alexa, „chatbotova“ na web-sjedištima koja se posjećuju, kao i autonomnih sistema u automobilima kojima se povjerava sigurnost, samo su neki od mnogih aspekata umjetne inteligencije u svakodnevnom životu. Iako se redovito koriste ove tehnologije, često se ne shvaća u potpunosti njihov način funkcioniranja niti utjecaj na život.

Kada se govori o pozitivnim utjecajima umjetne inteligencije jedna od prvih stvari je sigurno njezina pouzdanost. Ukoliko su prikupljeni ulazni podaci točni i dobro selektirani AI se pokazuje kao vrlo koristan alat za otklanjanje faktora ljudske pogreške. Umjetna inteligencija postala je

nezamjenjiv alat za organizacije u različitim industrijama, nudeći širok raspon prednosti koje potiču učinkovitost, inovacije i rast. Ovo su jedni od najbitnijih benefita integracije umjetne inteligencije u poslovne operacije: (Fedorychak, 2024)

- Povećana učinkovitost
- Diskretno donošenje odluka
- Poboljšana korisnička iskustva
- Bolje poslovanje i rukovanje resursima
- Proaktivno upravljanje rizikom i sigurnost
- Brže inovacije
- Skalabilnost i fleksibilnost
- Prednost nad konkurencijom

Usprkos svemu tome, nameće se pitanje koliko je umjetna inteligencija štetna za ljude, odnosno koji su to potencijalni rizici koji se javljaju nezinom prevelikom primjenom. Najveći strah kada se spominje umjetna inteligencija je hoće li ona zamijeniti ljude, hoće li ljudi izgubiti svoje poslove i biti upotpunosti zamijenjeni strojevima? Odgovor leži negdje između. Neki zastarjeli poslovi su već odavno zamijenjeni umjetnom inteligencijom, no ta brojka nije velika u usporedbi koliko se razvojem AI-a javlja potreba za ljudskim znanjem u drugim područjima koja su se razvila upravo zbog iste te tehnologije. Ljudski rad i promišljanje jesu i uvijek će biti najbitniji faktor u razvoju tehnologije i u unaprijeđenju kvalitete života na Zemlji.

Naposljetku, možemo zaključiti da pametnom primjenom umjetnom inteligencije čovječanstvo može uvelike profitirati i napredovati u mnoštvu aspekata, kao što su: napredak robotike, medicine, kvalitete života, produktivnosti i sl.

3. Metodologija rada

3.1. Strojno učenje

Strojno učenje je potpodručje umjetne inteligencije, koja se široko definira kao sposobnost stroja da oponaša inteligentno ljudsko ponašanje. Sustavi umjetne inteligencije koriste se za obavljanje složenih zadataka na način koji je sličan načinu na koji ljudi rješavaju probleme.

Cilj umjetne inteligencije je stvoriti računalne modele koji pokazuju "inteligentna ponašanja" poput ljudi. To znači strojeve koji mogu prepoznati vizualni prizor, razumjeti tekst napisan prirodnim jezikom ili izvesti radnju u fizičkom svijetu.

Strojno učenje jedan je od načina korištenja umjetne inteligencije. 1950-ih godina pionir umjetne inteligencije Arthur Samuel definirao ga je kao "područje proučavanja koje računalima daje mogućnost učenja bez eksplicitnog programiranja."

Definicija vrijedi, prema Mikeyu Shulmanu, predavaču na MIT Sloanu i voditelju strojnog učenja u Kenshou, koji je specijaliziran za umjetnu inteligenciju za financije i američke obavještajne zajednice. Usporedio je tradicionalni način programiranja računala ili "softver 1.0" s pečenjem, gdje recept zahtijeva precizne količine sastojaka i govori pekaru da miješa točno određeno vrijeme. Tradicionalno programiranje na sličan način zahtijeva stvaranje detaljnih uputa koje računalo treba slijediti.

Ali u nekim slučajevima pisanje programa koji bi stroj pratio oduzima puno vremena ili je nemoguće, poput osposobljavanja računala da prepozna slike različitih ljudi. Dok ljudi mogu lako obaviti ovaj zadatak, teško je reći računalu kako to učiniti. Strojno učenje ima pristup dopuštanja računalima da nauče sama programirati kroz iskustvo.

Strojno učenje počinje s podacima — brojevima, fotografijama ili tekстом, kao što su bankovne transakcije, slike ljudi ili čak pekarskih artikala, zapisi o popravcima, vremenski niz podataka iz senzora ili izvješća o prodaji. Podaci se prikupljaju i pripremaju za korištenje kao podaci za obuku ili informacije na kojima će se model strojnog učenja trenirati. Što više podataka, to je model bolji. (Brown, 2021.)

Strojno učenje može se svrstati u 3 kategorije:

- Deskriptivna

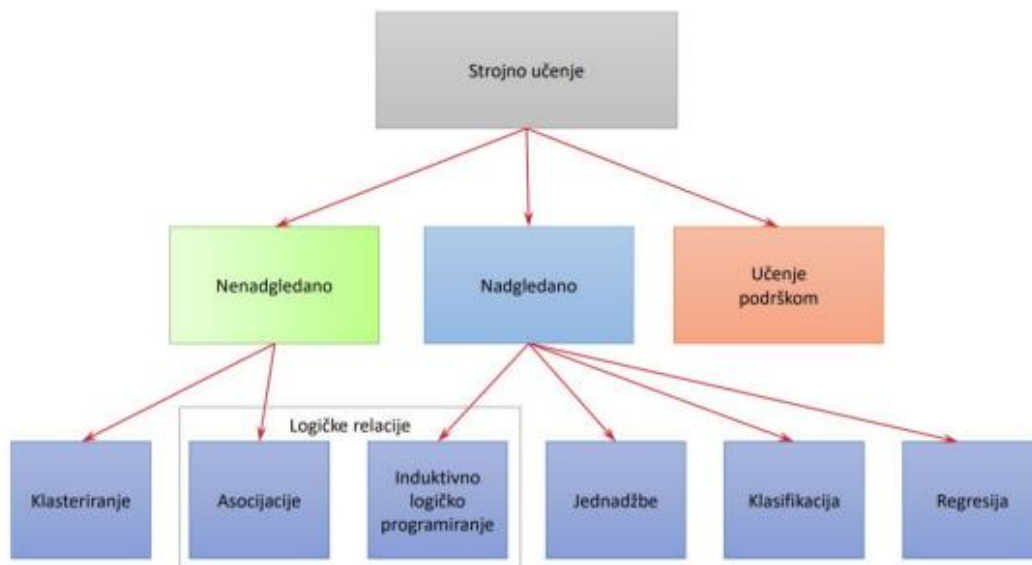
- Prediktivna
- Preskriptivna

Deskriptivno strojno učenje koristi podatke kako bi objasnilo što se dogodilo, kod prediktivnog nam sustav na temelju dostupnih podataka govori što će se dogoditi, dok kod preskriptivnog sustav nam sugerira što da činimo, odnosno koje akcije da poduzmemo na temelju analize podataka.

Osim 3 kategorije, postoje i 3 glavne podjele strojnog učenja;

- Nadzirano
- Nenadzirano
- Učenje podrškom, odnosno polunadzirano

Slika 3. prikazuje grafiku podjele strojnog učenja na 3 glavne kategorije koje detaljnije pokazuju čime se određena kategorije bavi.

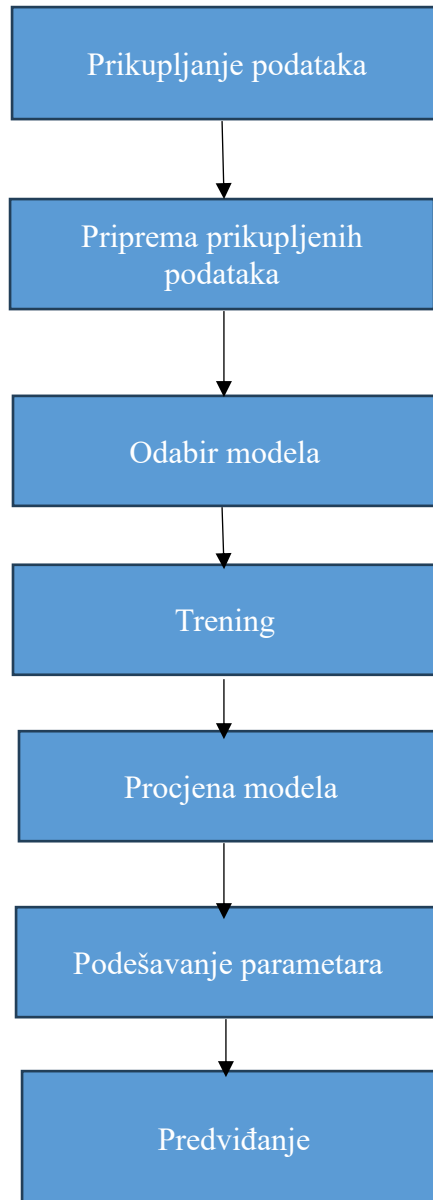


Slika 3. Podjela strojnog učenja na kategorije

Izvor: <https://repositorij.foi.unizg.hr/islandora/object/foi%3A6443/datastream/PDF/view>

3.2. Proces strojnog učenja

Prije nego se navedu vrste strojnog učenja i načina na koji svaka od njih funkcionira, treba objasniti na koji način funkcionira sami proces strojnog učenja. Mnogi smatraju proces strojnog učenja vrlo kompleksnim. U ovom poglavlju pokušat će se u 7 koraka objasniti proces strojnog učenja.



Grafika 1. Koraci u procesu strojnog učenja

Izvor: <https://www.datacamp.com/blog/what-is-machine-learning>

Kako bi se lakše objasnio i razumio cijeli proces, svaki od 7 koraka će biti zasebno objašnjen.

1. Korak – Prikupljanje podataka

Prvi i jedan od najvažnijih koraka u procesu strojnog učenja je prikupljanje podataka. Odavno je poznato da bez dobrih i pouzdanih podataka ne možemo dobiti ni precizne rezultate. Strojevi uče iz podataka koji su im dani na raspolaganje, tako da je od iznimne važnosti da su ti podaci pouzdani kako bi model strojnog učenja pronašao ispravne uzorke. U protivnom, ukoliko uneseni podaci nisu pouzdani ili relevantni, ishodi mogu biti djelomično ili u potpunosti krivi, odnosno to može rezultirati lošim predviđanjem.

2. Korak – Priprema prikupljenih podataka

Nakon što se odabere set podataka koji će se koristiti ti se podaci trebaju pripremiti, a to se može učiniti na više načina:

- Spajanje svih podataka zajedno te njihovo nasumično raspoređivanje. Ovaj korak osigurava da su svi podaci jednako distribuirani i njihov raspored neće utjecati na ishod procesa učenja.
- Čišćenje baze od neželjenih podataka, nedostajućih vrijednosti, stupaca, redaka, podataka koji se ponavljaju, pretvaranje podataka u druge vrijednosti i sl.
- Vizualizacija podataka kako bi se lakše razumjela struktura i povezanost podataka, odnosno povezanost između varijabli i klasa.
- Razdvajanje pripremljenih podataka u dva seta – trening set i testni set. Trening set je set od kojeg model uči, dok se testni set koristi kako bi se provjerila točnost modela nakon treninga.

3. Korak – Odabir modela

Nakon što su svi podaci spremni za korištenje, vrijeme je da se odabere model koji će se koristiti za analizu tih podataka. Od iznimne je važnosti da se odabere model koji je prikladan i relevantan za zadatak koji se želi izvršiti. Postoje različiti modeli prikladni za razne zadatke poput onih za:

prepoznavanje govora, prepoznavanje slika, prognoze i sl. No, osim toga mora se odabrati model koji je prikladan za vrstu podataka s kojima će taj model raditi, odnosno da li su ti podaci numerički ili kategorički.

4. Korak – Trening modela

Nakon odabira, sljedeći korak je trening odabranog modela. Ovo je ujedno i najvažniji korak u cijelom procesu strojnog učenja. U ovom koraku, pripremljeni podaci se ubacuju u model strojnog učenja kako bi se pronašli uzorci i napravila predviđanja. Rezultat ovog procesa je učenje modela na temelju podataka, te njegovo poboljšanje u preciznosti predviđanja.

5. Korak – Procjena modela

Nakon treninga modela treba se provjeriti kako naš model radi. To se izvršava na način da se testira izvedba našeg modela na temelju prethodno neviđenog seta podataka, odnosno onog seta podataka kojeg smo izdvojili ranije. Ukoliko bi se koristio isti set koji se već koristio za trening modela to bi rezultiralo abnormalnom velikoj preciznosti, jer bi model prepoznao identične uzorke kao u prethodnom procesu.

6. Korak – Podešavanje parametara

Nakon što smo napravili i procijenili naš model, vrijeme je da se provjeri preciznost modela i što se može učiniti kako bi se dobila što veća preciznost ukoliko je to moguće. To se postiže podešavanjem parametara u modelu. Primjeri parametara modela uključuju težine i pristranosti u neuronskoj mreži, koeficijente u modelu linearne regresije i vektore potpore u stroju vektora potpore. Na primjer, u jednostavnom linearnom regresijskom modelu danom $y = w_0 + w_1x$, parametri (težinu) i (pristranosti) saznaju se iz podataka. Hiperparametri su, s druge strane, vanjske konfiguracije postavljene prije početka procesa učenja. Ovi se parametri ne uče iz podataka o vježbanju, već ih vježbač ručno postavlja. Hiperparametri upravljaju cjelokupnom strukturom i ponašanjem modela, utječući na način na koji se uče parametri modela. Primjeri hiperparametara uključuju stopu učenja u gradijentnom spuštanju, broj slojeva i neurona u neuronskoj mreži,

dubinu stabla odlučivanja, broj klastera u k-srednjim klasterima i parametar regulacije u logističkoj regresiji (Anon., 2024).

7. Korak – Predviđanje

Nakon što smo obavili svih 6 prethodnih koraka na red dolazi posljednji – Predviđanje. Naš model je napokon spreman za predviđanje, te se u ovom koraku dobiva točnost modela i njegova uspješnost na podacima na kojima nije učio.

3.3. Vrste i modeli strojnog učenja

Razlikuju se 3 glavne vrste modela strojnog učenja:

1. Nadzirano strojno učenje
2. Nenadzirano strojno učenje
3. Polunadzirano strojno učenje

U sljedećim poglavljima će se objasniti svaka navedena vrsta strojnog učenja zasebno.

3.3.1. Nadzirano strojno učenje

Kod nadziranog učenja (engl. supervised learning) modeli se uvježbavaju, odnosno grade primjenom skupa ulaznih i izlaznih podataka prikupljenih eksperimentom ili akvizicijom tako da mogu predviđati buduće izlaze na temelju dostupnih ulaza (Bolf, 2021.)

Nadzirano učenje koristi set za obuku kako bi naučilo modele da daju željeni učinak. Ovaj skup podataka za obuku uključuje ulaze i ispravne izlaze, koji omogućuju modelu da s vremenom uči. Algoritam mjeri svoju točnost kroz funkciju gubitka, prilagođavajući se dok se pogreška dovoljno ne smanji. Nadzirano učenje koristi se postupcima klasifikacije i regresije:

Klasifikacija koristi algoritam za točno dodjeljivanje testnih podataka u određene kategorije. Prepoznaje određene entitete unutar skupa podataka i pokušava izvući neke zaključke o tome kako

te entitete treba označiti ili definirati. Uobičajeni algoritmi klasifikacije su linearni klasifikatori, strojevi za potporu vektora (SVM), stabla odlučivanja, k-najbliži susjed i slučajna šuma (Anon., 2021). Regresija se koristi za razumijevanje odnosa između zavisnih i nezavisnih varijabli. Obično se koristi za izradu projekcija, kao što je prihod od prodaje za određenu tvrtku. Linearna regresija, i polinomna regresija popularni su regresijski algoritmi (Anon., 2021).

Neki od najčešćih primjera gdje se nenadzirano strojno učenje primjenjuje su:

- Kategorizacija teksta
- Prepoznavanje lica
- Prepoznavanje potpisa
- Otkrivanje spama u elektroničkoj pošti
- Prognoza vremena
- Predviđanje cijena dionica

3.3.1.1. KNN algoritam

KNN ili algoritam K-najbližih susjeda je jednostavan, neparametarski algoritam strojnog učenja koji se jednako koristi za klasifikacijske i regresijske zadatke. KNN se temelji na pretpostavci da se slične stavke nalaze međusobno blizu u danom prostoru. Temeljni koncept KNN-a je da identificira k-najbliže susjede za dani upit, te da predvidi rezultate za taj upit na temelju njihovih klasnih oznaka. (LaViale, 2023.)

Neka od najčešćih područja primjene KNN algoritma su: (Hussain, A., 2020.)

- Poljoprivreda
- Financije
- Medicina
- Prepoznavanje lica
- Sustavi za davanje preporuka

Kada se govori o algoritmima za strojno učenje, ne postoji jedan algoritam koji bi udovoljio svim vrstama zahtjeva koji se stavljaju pred njega. Stoga se za svaki model može reći da ima svoje

prednosti i nedostatke u odnosu na ono što pružaju drugi modeli.

Neke od prednosti KNN algoritma su: (Airon, P., 2022.)

- Jednostavnost algoritma za tumačenje
- Brzi izračuni
- Svestranost – koriste se za regresiju i klasifikaciju
- Visoka točnost

Dok su neki od nedostataka: (Airon, P., 2022.)

- Točnost ovisi o kvaliteti podataka
- Predviđanje postaje sporo s povećanjem broja podataka
- Nije relevantno za velike skupove podataka
- Potrebno je pohraniti sve podatke o obuci, stoga zahtijeva veliku količinu memorije

3.3.1.2. Logistička regresija

Logistička regresija je nadzirani algoritam strojnog učenja koji se koristi za zadatke klasifikacije gdje je cilj predvidjeti vjerojatnost da instanca pripada određenoj klasi ili ne, te analizira odnos između dva faktora podataka.

Logistička regresija koristi se za binarnu klasifikaciju gdje se koristi sigmoidna funkcija, koja uzima ulaz kao nezavisne varijable i daje vrijednost vjerojatnosti između 0 i 1.

Ona predviđa izlaz kategoričke zavisne varijable. Stoga ishod mora biti kategorička ili diskretna vrijednost. Može biti Da ili Ne, 0 ili 1, točno ili netočno itd., ali umjesto davanja točne vrijednosti kao 0 i 1, daje vjerojatnosne vrijednosti koje leže između 0 i 1. (Anon., 2024.)

Postoje 3 tipa logističke regresije: (Bobbitt, Z., 2021.)

1. Binarna logistička regresija - Varijabla može pripadati samo jednoj od dvije kategorije.
2. Multinomijalna logistička regresija – Varijabla može pripadati jednoj od tri ili više kategorija, ali ne postoji prirodni poredak među kategorijama.
3. Ordinalna logistička regresija – Varijabla može pripadati jednoj od tri ili više kategorija i postoji prirodni poredak među kategorijama

3.3.1.3. Neuronske mreže

Neuronska mreža ili kako se još naziva umjetna neuronska mreža je model strojnog učenja dizajniran za obradu podataka na način koji oponaša funkciju i strukturu ljudskog mozga. Neuronske mreže su zamršene mreže međusobno povezanih čvorova ili umjetnih neurona koji surađuju u rješavanju kompliciranih problema. Neuronske mreže uvelike se koriste u raznim aplikacijama, uključujući prepoznavanje slika, prediktivno modeliranje, donošenje odluka i obradu prirodnog jezika. Primjeri značajnih komercijalnih primjena u posljednjih 25 godina uključuju prepoznavanje rukopisa za obradu čekova, transkripciju govora u tekst, analizu podataka o istraživanju nafte, predviđanje vremena i prepoznavanje lica. (Yasar, K., 2024.)

Može se zaključiti da se ovaj kompleksan model strojnog bazira na strukturi ljudskog mozga, no kako on zapravo funkcionira u stvarnosti? Kao i ljudski mozak umjetna neuronska mreža prima ulazne signale koji dosežu prag pomoću sigmoidnih funkcija, obrađuju informacije i zatim generiraju izlazni signal. Poput ljudskih neurona, umjetne neuronske mreže primaju više ulaza, zbrajaju ih i zatim obrađuju zbroj sigmoidnom funkcijom. Ako zbroj unesen u sigmoidnu funkciju proizvede vrijednost koja funkcionira, ta vrijednost postaje izlaz neuronske mreže. (Coursera, 2024.)

Dakako, to je struktura samo jednog individualnog neurona u cjelokupnoj mreži neurona, ali one imaju višestruke slojeve i neurone koji kreiraju tu mrežu. Struktura cjelokupne umjetne neuronske mreže sastoji se od: (Coursera, 2024.)

- Ulazni sloj – on prima ulazne podatke i prenosi ih na drugi (skriveni) sloj neurona pomoću sinapsi
- Skriveni sloj – uzima podatke iz ulaznog sloja za kategorizaciju ili otkrivanje željenih aspekata podataka. Čvorovi u skrivenom sloju šalju podatke u više skrivenih slojeva ili, konačno, u izlazni sloj.
- Izlazni sloj - uzima podatke iz skrivenog sloja i ispisuje rezultate, te ima onoliko čvorova koliko model želi.

- Sinapse - povezuju čvorove u slojevima i između slojeva.

3.3.2. Nenadzirano strojno učenje

Nenadzirano strojno učenje je moćan alat pomoću kojeg stječemo važan uvid iz podataka. Za razliku od nadziranog strojnog učenja, nenadzirano učenje ne zahtijeva označene podatke, nego ima za cilj automatsko otkrivanje uzoraka, struktura ili grupiranja u podacima. Koristeći se tehnikama poput grupiranja, pridruživanja ili smanjivanja dimenzionalnost može se dobiti uvid u informacije koje su sakrivene u podacima. (Patrick, 2023.)

Dakle, može se zaključiti da su osnovna 3 zadatka nenadziranog strojnog učenja: grupiranje, pridruživanje i smanjivanje dimenzionalnost. Za bolje razumijevanje tih osnovnih zadataka definirat će se svaka metoda učenja, te će biti navedeni uobičajeni algoritmi i pristupi za njihovo provođenje.

Grupiranje je tehnika rudarenja podataka koja grupira neoznačene podatke na temelju njihovih sličnosti ili razlika. Algoritmi grupiranja koriste se za obradu sirovih, neklasificiranih podatkovnih objekata u grupe predstavljene strukturama ili uzorcima u informacijama. Algoritmi klasteriranja mogu se kategorizirati u nekoliko tipova, posebno isključivi, preklapajući, hijerarhijski i probabilistički (Anon., 2021).

Pravilo pridruživanja metoda je temeljena na pravilu za pronalaženje odnosa između varijabli u danom skupu podataka. Ove se metode često koriste za analizu tržišne košarice, omogućujući tvrtkama da bolje razumiju odnose između različitih proizvoda. Razumijevanje potrošačkih navika kupaca omogućuje tvrtkama da razviju bolje strategije unakrsne prodaje i mehanizme za preporuku. Primjeri ovoga mogu se vidjeti na Amazonovom popisu za reprodukciju "Kupci koji su kupili ovaj artikl također su kupili" ili Spotifyjevom popisu za reprodukciju "Discover Weekly". Iako postoji nekoliko različitih algoritama koji se koriste za generiranje pravila pridruživanja, kao što su Apriori, Eclat i FP-Growth, Apriori algoritam je najčešće korišten (Anon, 2021).

Iako više podataka općenito daje točnije rezultate, to također može utjecati na izvedbu algoritama strojnog učenja, a također može otežati vizualizaciju skupova podataka. Smanjenje dimenzionalnosti je tehnika koja se koristi kada je broj značajki ili dimenzija u danom skupu podataka prevelik. Smanjuje broj unosa podataka na upravljivu veličinu, istovremeno čuvajući cjelovitost skupa podataka što je više moguće. Obično se koristi u fazi predprocesiranja podataka, a postoji nekoliko različitih metoda smanjenja dimenzionalnosti koje se mogu koristiti, kao što su: analiza glavnih komponenti, dekompozicija singularne vrijednosti, autokoderi (Anon., 2021).

3.3.2.1. Primjena nenadziranog strojnog učenja

Tehnike strojnog učenja postale su uobičajena metoda za poboljšanje korisničkog iskustva proizvoda i testiranje sustava za osiguranje kvalitete. Učenje bez nadzora pruža istraživački put za pregled podataka, omogućujući tvrtkama da brže identificiraju obrasce u velikim količinama podataka u usporedbi s ručnim promatranjem. Neke od najčešćih primjena nenadziranog učenja u stvarnom svijetu su: (Anon., 2021)

- Odjeljci vijesti: Google vijesti koriste nenadzirano učenje za kategorizaciju članaka o istoj priči iz različitih internetskih novinskih kuća. Na primjer, rezultati predsjedničkih izbora mogli bi se kategorizirati pod njihovom oznakom za "američke" vijesti.
- Računalno prepoznavanje: Algoritmi za učenje bez nadzora koriste se za zadatke vizualne percepcije, kao što je prepoznavanje objekata.
- Medicinsko oslikavanje: nenadzirano strojno učenje pruža bitne značajke medicinskim uređajima za oslikavanje, kao što su otkrivanje slike, klasifikacija i segmentacija, koje se koriste u radiologiji i patologiji za brzo i točno dijagnosticiranje pacijenata.
- Otkrivanje anomalija: modeli učenja bez nadzora mogu pročešljati velike količine podataka i otkriti netipične podatkovne točke unutar skupa podataka. Ove anomalije mogu podići svijest o neispravnoj opremi, ljudskoj pogrešci ili kršenju sigurnosti.
- Personaliziranje kupaca: Definiranje osobnosti kupaca olakšava razumijevanje zajedničkih osobina i kupovnih navika poslovnih klijenata. Učenje bez nadzora omogućuje tvrtkama da izgrade bolje profile kupaca, omogućujući organizacijama da prikladnije usklade svoje

poruke o proizvodima.

- Preporuke: korištenjem prošlih podataka o kupovnom ponašanju, nenadzirano učenje može pomoći u otkrivanju trendova podataka koji se mogu koristiti za razvoj učinkovitijih strategija dodatne prodaje. Ovo se koristi za davanje relevantnih preporuka za dodatke kupcima tijekom procesa naplate za online trgovce.

3.3.3. Polunadzirano strojno učenje

Polunadzirano strojno učenje koristi mali dio označenih podataka i mnogo neoznačenih podataka kako bi se trenirao model.

Polunadzirano učenje nadilazi tehnike nadziranog učenja i nenadziranog učenja za rješavanje njihovih ključnih izazova. S njim se trenira početni model na nekoliko označenih uzoraka, a zatim se iterativno primjenjuje na veći broj neoznačenih podataka.

- Za razliku od učenja bez nadzora, polunadzirano učenje radi za razne probleme od klasifikacije i regresije do grupiranja i pridruživanja.
- Za razliku od učenja pod nadzorom, metoda koristi male količine označenih podataka, ali i velike količine neoznačenih podataka, što smanjuje troškove ručnog označavanja i skraćuje vrijeme pripreme podataka (Anon., 2022).

Na slici se mogu vidjeti odnosi sva 3 pristupa strojnog učenja, na prikazu je jasno vidljivo da se glavne razlike nalaze u načinu označavanja podataka. U nadziranom učenju svi podaci se označuju, kod nenadziranom se radi u potpunosti bez označavanja podataka, dok se kod polunadziranog učenja kombiniraju oba pristupa, ali se u većem postotku koriste neoznačeni podaci.

SUPERVISED LEARNING vs SEMI-SUPERVISED LEARNING vs UNSUPERVISED LEARNING



Slika 4. Nadzirano učenje vs polunadzirano učenje vs nenadzirano učenje

Izvor: <https://www.altexsoft.com/blog/semi-supervised-learning/>

Za rad s neoznačenim skupom podataka mora postojati odnos između objekata u skupu podataka. Da bismo ovo razumjeli, polunadzirano učenje koristi bilo koju od sljedećih pretpostavki:

- Pretpostavka o kontinuitetu: prema pretpostavci o kontinuitetu, objekti koji su blizu jedan drugog imaju tendenciju da dijele istu grupu ili oznaku. Ova se pretpostavka također koristi u nadziranom učenju, a skupovi podataka odvojeni su granicama odluke. U polunadziranom učenju, granice odluke se dodaju uz pretpostavku glatkoće u granicama niske gustoće.
- Pretpostavke klastera: U ovoj pretpostavci podaci su podijeljeni u različite diskretne klasterne. Nadalje, točke u istom klasteru dijele izlaznu oznaku.
- Pretpostavke mnogostrukosti: Ova pretpostavka pomaže u korištenju udaljenosti i gustoće, a ti podaci leže na mnogostrukosti manjih dimenzija od ulaznog prostora.
- Dimenzionalni podaci nastaju procesom koji ima manji stupanj slobode i može ga biti teško izravno modelirati. Ova pretpostavka postaje praktična ako je visoka (Potrimba, 2022).

3.4. Duboko učenje

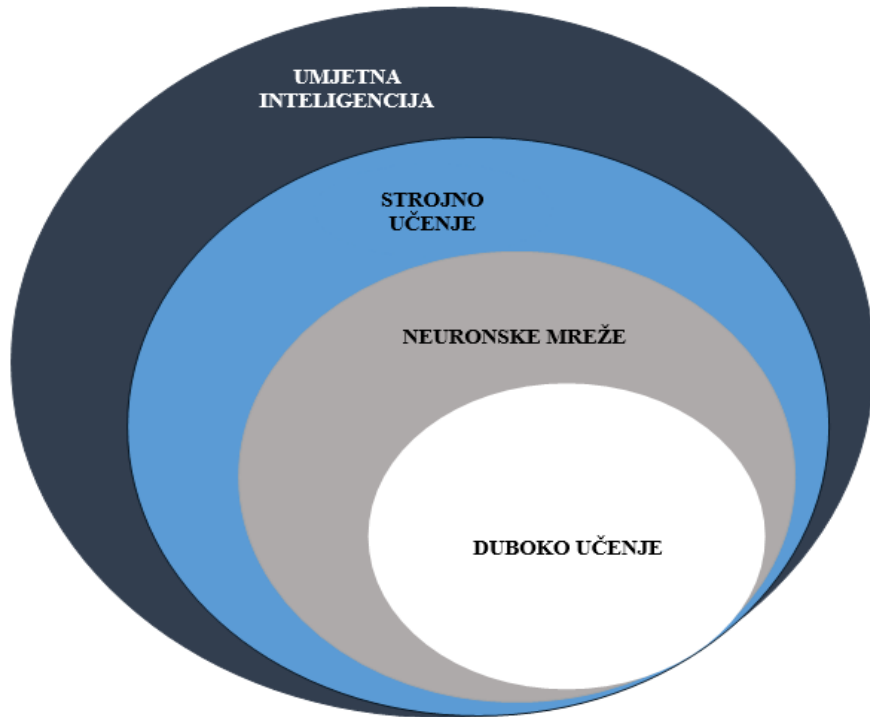
Iako mnogi izdvajaju duboko učenje od strojnog učenja ono je zapravo vrsta strojnog učenja, odnosno naprednija i kompleksnija metoda strojnog učenja. Način na koji duboko učenje funkcionira je da uz pomoću više algoritamskih slojeva uči strojeva da imitiraju način na koji ljudi razmišljaju, uče i zaključuju. Dakle, zadaća dubokog učenja je da imitira ljudski mozak uz pomoć algoritama složenih u barem 3 sloja koja čine neuronsku mrežu. Model dubinskog učenja osmišljen je za kontinuiranu analizu podataka s logičkom strukturom sličnom načinu na koji bi čovjek izvukao zaključke. Kako bi se analiza izvršila, aplikacije temeljene na dubokom učenju koriste slojevitou strukturu algoritama koja se naziva neuronska mreža. Dizajn umjetne neuronske mreže inspiriran je mrežom neurona u ljudskom mozgu, rezultat toga je sustav učenja koji je mnogo sposobniji od uobičajenih modela strojnog učenja.

Proces kreiranja modela dubinskog učenja koji ne donosi netočne zaključke vrlo je težak i kompleksan, te zahtijeva mnogo obuke i vremena, ali kada radi kako je predviđeno, funkcionalno duboko učenje često se doživljava kao znanstveno čudo koje mnogi smatraju okosnicom prave umjetne inteligencije (Grieve, 2023).

Neki od primjera primjene dubokog učenja s kojima se možemo susresti u svakodnevnom životu su:

- Aplikacije za prepoznavanje slika (npr. prepoznavanje vrste ptice na temelju priložene slike)
- Autonomna vozila
- Virtualni asistenti
- Prepoznavanje lica
- Nosivi senzori i uređaji namijenjeni za nadziranje zdravstvenog stanja pacijenata u stvarnom vremenu

Na slici 5. prikazan je odnos dubokog učenja između umjetne inteligencije, strojnog učenja i neuronskih mreža



Slika 5. Odnos između AI-a, strojnog učenja, neuronskih mreža i dubokog učenja

Izvor: Autor (prema izvoru: <https://www.javatpoint.com/deep-learning-vs-machine-learning-vs-artificial-intelligence>)

Osim što je duboko učenje kompleksnija vrsta strojnog učenja, ta činjenica dolazi s pretpostavkom da ono ima neke prednosti u odnosu na tradicionalno strojno učenje.

Zapravo postoji mnoštvo prednosti koje uzdižu duboko učenje razinu više u odnosu na tradicionalne modele strojnog učenja. Neki od primjera su: (Grieve, 2023)

- Automatizacija generiranja značajki: Algoritmi dubokog učenja mogu generirati nove značajke iz ograničenog broja koji se nalaze u skupu podataka za obuku bez dodatne ljudske intervencije. To znači da duboko učenje može obavljati složene zadatke koji često zahtijevaju opsežni inženjering značajki.

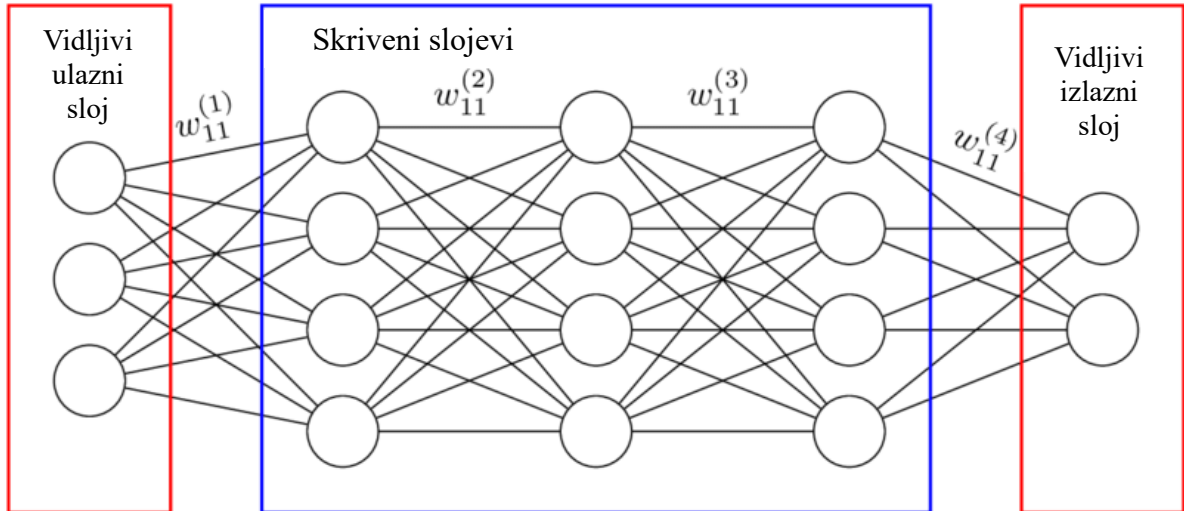
Za tvrtke to znači brže uvođenje aplikacija ili tehnologije koje pružaju vrhunsku točnost.

- Dobro radi s nestrukturiranim podacima: Jedna od najvećih prednosti dubokog učenja je njegova sposobnost rada s nestrukturiranim podacima. U poslovnom kontekstu ovo postaje osobito relevantno ako uzmete u obzir da je većina poslovnih podataka nestrukturirana. Tekst, slike i glas neki su od najčešćih formata podataka koje tvrtke koriste. Klasični

algoritmi ograničeni su u svojoj sposobnosti analiziranja nestrukturiranih podataka, što znači da ovo bogatstvo informacija često ostaje neiskorišteno.

- Bolje sposobnosti samoučenja: Višestruki slojevi u dubokim neuronskim mrežama omogućuju modelima da postanu učinkovitiji u učenju složenih značajki i izvođenju intenzivnijih računalnih zadataka, tj. istovremenom izvršavanju mnogih složenih operacija. Nadmašuje strojno učenje u zadacima strojne percepcije koji uključuju nestrukturirane skupove podataka.
- Isplativost: Dok obuka modela dubokog učenja može biti skupa, jednom obučena može pomoći tvrtkama da smanje nepotrebne troškove. U industrijama kao što su proizvodnja, savjetovanje ili čak maloprodaja, trošak netočnog predviđanja ili kvara proizvoda je ogroman. To često nadmašuje troškove obuke modela dubokog učenja.

Na slici 6. se nalazi shematski prikaz algoritamskih slojeva u dubokom učenju. S lijeve strane se nalaze vidljivi ulazni sloj, dok se u sredini nalaze skriveni slojevi koji imaju ključnu ulogu, na samome kraju prikaza, tj. na desnoj strani je vidljivi izlazni sloj.



Slika 6. Prikaz slojeva dubokog učenja

Autor: Adcock, J., Allen, E., Day, M., 2015.

Izvor: https://www.researchgate.net/figure/An-example-of-a-deep-learning-neural-network-with-3-hidden-layers-For-a-Boltzmann_fig6_286513346

3.5. Python

Python je interpretirani, objektno orijentirani programski jezik visoke razine s dinamičkom semantikom. Njegove visoke razine ugrađenih podatkovnih struktura, u kombinaciji s dinamičkim tipkanjem i dinamičkim vezanjem, čine ga vrlo atraktivnim za brzi razvoj aplikacija, kao i za korištenje kao skriptni ili ljepljivi jezik za povezivanje postojećih komponenti. Pythonova jednostavna sintaksa koju je lako naučiti naglašava čitljivost i stoga smanjuje troškove održavanja programa. Python podržava module i pakete, što potiče modularnost programa i ponovnu upotrebu koda. Python tumač i opsežna standardna biblioteka dostupni su u izvornom ili binarnom obliku bez naknade za sve glavne platforme i mogu se besplatno distribuirati. (Anon., 2020.)

Danas je nedvojebeno Python najpopularniji jezik upravo za strojno učenje, izuzev svojom jednostavnošću i pristupačnošću, Python je popularan među AI developerima iz razloga što je sam jezik konstruiran kao vrlo moćan alat za analizu podataka.

3.5.1. Python biblioteke

Biblioteke napisane u Pythonu igraju ključnu ulogu u strojnom učenju i znanosti o podacima. Svaki modul u Python biblioteci služi određenoj svrsi. Mnogi različiti programi mogu imati koristi od modularnosti koda ove biblioteke. Mnogo je prednosti za programera kao rezultat ovog alata.

Širok raspon operacija može se izvesti pomoću metoda i varijabli koje se nalaze u bilo kojoj Python biblioteci (nizovi, rječnici, objekti itd.). Moguće je automatizirati zadatke, predvidjeti ishode i isporučiti inteligentne uvide uz pomoć SEO alata. (Tuama, 2022.)

3.5.2. Što su Python biblioteke

Biblioteka je zbirka koda koja svakodnevne zadatke čini učinkovitijima. Koristeći Seaborn, na primjer, možete generirati vizualizacije sa samo jednom linijom koda. Da biste stvorili grafikon iz objekta, morali biste napisati mnogo koda bez ovakve biblioteke. Python je popularan izbor za analizu podataka zbog svoje opsežne biblioteke alata za manipulaciju, vizualizaciju i obuku modela strojnog učenja. (Tuama, 2022.)

4. Opis istraživanja i rezultati istraživanja

U sljedećim poglavljima bit će prikazani ishodi istraživanja koja su provedena na temelju baze podataka iz repozitorija Kaggle. Istraživanje je provedeno uz pomoć programskog jezika Python u kojem su izgrađena tri modela strojnog učenja: logistička regresija, kNN i neuronske mreže.

4.1. Opis korištenih podataka

Za sljedeći model će se koristiti baza podataka dionice kompanije Tesla, navedena baza obuhvaća povijest trgovanja dionicom u razdoblju od 10 godina, odnosno od 2010. do 2020. godine.

Baza podataka se sastoji od 2416 opservacija i 7 varijabli, te je pohranjena u Excel tablici.

Naziv varijable	Hrvatski naziv varijable	Opis varijable
Date	Datum	Označava datum trgovanja dionicom u formatu godina, mjesec, dan. Podaci su u numeričkom obliku.
Open	Otvorenje	Cijena po kojoj je prva dionica prodana na određeni dan trgovanja.
High	Najviša	Najviša postignuta cijena po kojoj je dionica prodana na dan trgovanja.
Low	Najniža	Najniža cijena prodane dionice na dan trgovanja.
Close	Završna	Cijena dionice na kraju dana trgovanja.
Adj Close	Korigirana završna	Prilagođena cijena dionice. Korigirana cijena dionice na kraju dana trgovanja
Volume	Promet	Količina prodanih dionica na dan trgovanja.

Tablica 1. Opis svih varijabli iz baze podataka

Izvor: autor

4.2. Priprema podataka

Prije kreiranja modela najprije je potrebno pripremiti i oblikovati podatke kako bi se oni što bolje iskoristili. Prvo se učitala baza podataka koja je u .csv formatu, te se naredbom „head“ dobiva uvid u prvih nekoliko redaka učitane tablice kako bi se bolje vidjelo s kakvim podacima se raspolaže. Dobivena tablica jasno pokazuje da učitana baza podataka ima 7 varijabli koje sadrže podatke za datum trgovanja dionicom, zatim početnu, najvišu, najnižu i završnu cijenu, te korigiranu završnu cijenu i ukupan promet trgovanja dionicom.

Sve cijene u tablici predstavljaju iznose u američkim dolarima. Uz pomoć naredbe „shape“ vidljivo je da baza podataka sadrži 2416 redova i 7 varijabli.

Uz pomoć naredbe „describe()“ dobiva se uvid u neke osnovne podatke deskriptivne statistike.

```
dionice.describe()
```

Slika 7. Naredba za deskriptivnu statistiku

Izvor: autor

	Prva cijena	Najviša cijena	Najniža cijena	Završna cijena	Korigirana završna cijena	Ukupan promet
Zbroj	2416.0000	2416.0000	2416.0000	2416.0000	2416.0000	2416.0000
Aritmetička sredina	186.2711	189.5782	182.9166	186.4036	186.4036	5.5727
Stand. devijacija	118.7402	120.8923	116.8575	119.1360	119.1360	4.9879
Minimum	16.1399	16.6299	14.9800	15.8000	15.8000	1.1850
Donji kvartil	34.3425	34.8975	33.5875	34.4000	34.4000	1.8993
Medijan	213.0350	216.7450	208.8700	212.9600	212.9600	4.5784
Gornji kvartil	266.4500	270.9275	262.1025	266.7749	266.7749	7.3612
Maksimum	673.6900	786.1400	673.5200	780.0000	780.0000	4.7065

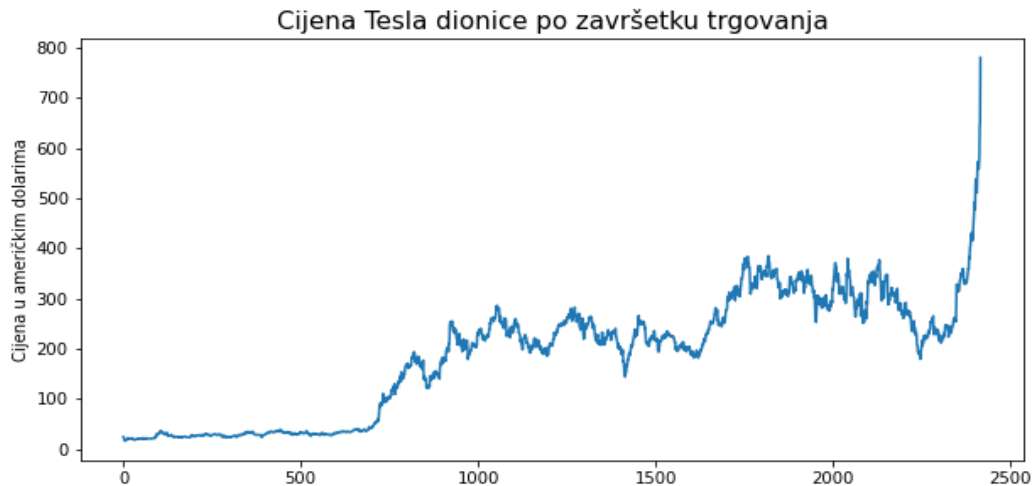
Tablica 2. Deskriptivna statistika baze podataka

Izvor: Autor

Na tablici 2. može se vidjeti deskriptivna statistika kompletne baze podataka kao što su medijan, standardna devijacija, minimum, maksimum, prosjek, te gornji i donji kvartil.

Već po prikazanim podacima može se zaključiti da se u periodu od 10 godina dogodilo puno promjena, prvenstveno u kretanju cijene dionice. Vidljivo je da je najniža cijena dionice iznosila 16.139999, a najviša 673.690002 USD.

Kako bi se jednostavnije prikazalo kretanje cijene dionice Tesle, koristit će se sljedeća naredba pomoću koje se kreira graf kretanja varijable „Close“.



Slika 8. Graf kretanja cijene dionica

Izvor: autor

Na temelju slike 8. može se zaključiti da je Tesla zabilježila blago rastući trend cijene dionica, no i da je došlo do naglog rasta u otprilike posljednjoj godini trgovanja prema podacima baze podataka.

Promatrajući podatke iz baze podataka može se zamijetiti da su podaci za varijable „Close“ i „Adj Close“ u prvih 5 redova jednaki, stoga će se ispitati koliko redaka sadrži identične podatke za te dvije varijable.

```
dionice.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-06-29	19.000000	25.00	17.540001	23.889999	23.889999	18766300
1	2010-06-30	25.790001	30.42	23.299999	23.830000	23.830000	17187100
2	2010-07-01	25.000000	25.92	20.270000	21.959999	21.959999	8218800
3	2010-07-02	23.000000	23.10	18.709999	19.200001	19.200001	5139800
4	2010-07-06	20.000000	20.00	15.830000	16.110001	16.110001	6866900

Slika 9. Prikaz prvih 5 redaka baze podataka

Izvor: autor

```
dionice[dionice['Close'] == dionice['Adj Close']].shape  
(2416, 7)
```

Slika 10. Naredba za usporedbu koliko se redova dviju varijabli podudara

Izvor: autor

Dobivenim rezultatom vidljivo je da varijable „Close“ i „Adj Close“ sadrže u potpunosti jednake podatke u svim redovima. Ukoliko baza podataka sadrži suvišne podatke najbolje bi bilo da ih se ukloni, jer dolazi do redundancije koja nije pogodna za bazu podataka.

Stoga će se stupac „Adj Close“ ukloniti iz tablice uz pomoć naredbe „`dionice = dionice.drop(['Adj Close'], axis=1)`“

```
dionice = dionice.drop(['Adj Close'], axis=1)  
dionice.head()
```

	Date	Open	High	Low	Close	Volume
0	2010-06-29	19.000000	25.00	17.540001	23.889999	18766300
1	2010-06-30	25.790001	30.42	23.299999	23.830000	17187100
2	2010-07-01	25.000000	25.92	20.270000	21.959999	8218800
3	2010-07-02	23.000000	23.10	18.709999	19.200001	5139800
4	2010-07-06	20.000000	20.00	15.830000	16.110001	6866900

Slika 11. Naredba za brisanje stupca „Adj Close“ iz tablice

Izvor: autor

Nakon što su se uredili i uklonili suvišni podaci unutar tablice mogu se izraditi grafički prikazi za preostale varijable.

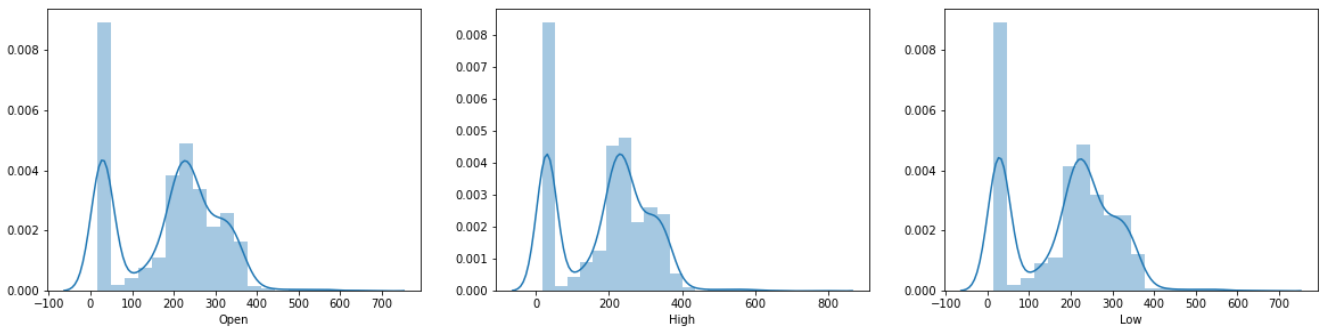
```

features = ['Open', 'High', 'Low', 'Close', 'Volume']
plt.subplots(figsize=(20,10))
for i, col in enumerate(features):
    plt.subplot(2,3,i+1)
    sb.distplot(dionice[col])
plt.show()

```

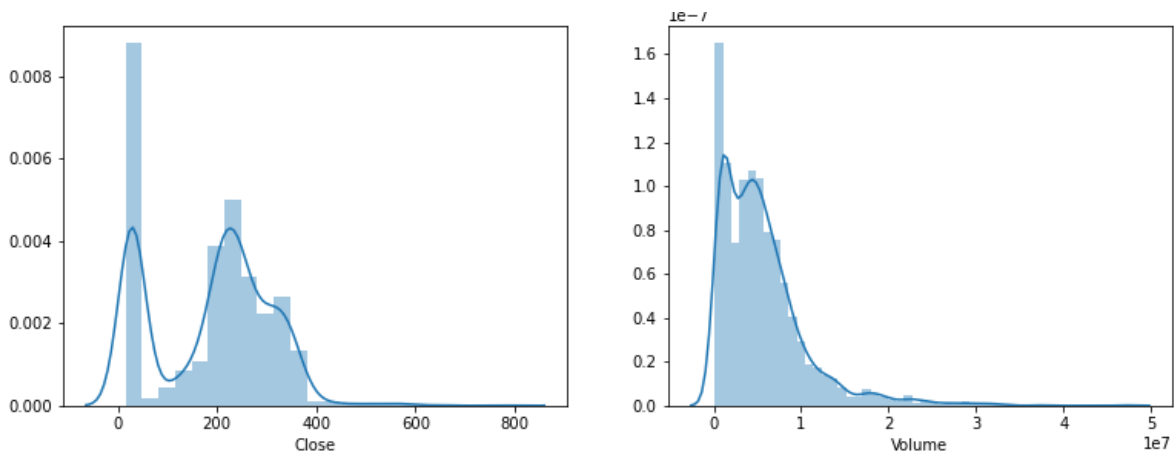
Slika 12. Naredba za kreiranje dijagrama distribucije

Izvor: autor



Slika 13. Dijagrami distribucije za varijable Open, High i Low

Izvor: autor



Slika 14. Dijagrami distribucije za varijable Close i Volume

Izvor: autor

Na dobivenim dijagramima distribucije za Open, High, Low i Close podatke može se vidjeti istaknuta 2 vrha, što ukazuje da su podaci odskakali u dvije regije. Dok su podaci o volumenu

zakrivljeni ulijevo.

Kako bi se lakše upravljalo podacima dobivenih unutar varijable „Date“ istu će se razdvojiti u 3 stupca, to će se izvesti tako što će se format dan/mjesec/godina staviti svaki u poseban stupac kako bi se dobila bolja iskoristivost dobivenih podataka.

```
splitted = dionice['Date'].str.split('-', expand=True)

dionice['day'] = splitted[2].astype('int')
dionice['month'] = splitted[1].astype('int')
dionice['year'] = splitted[0].astype('int')

dionice.head()
```

Slika 15. Naredba kojom se uz pomoć varijable „Date“ kreiraju 3 dodatna stupca „day“, „month“, „year“

Izvor: autor

	Date	Open	High	Low	Close	Volume	day	month	year
0	2010-06-29	19.000000	25.00	17.540001	23.889999	18766300	29	6	2010
1	2010-06-30	25.790001	30.42	23.299999	23.830000	17187100	30	6	2010
2	2010-07-01	25.000000	25.92	20.270000	21.959999	8218800	1	7	2010
3	2010-07-02	23.000000	23.10	18.709999	19.200001	5139800	2	7	2010
4	2010-07-06	20.000000	20.00	15.830000	16.110001	6866900	6	7	2010

Slika 16. Rezultat naredbe split

Izvor: autor

Nakon što se stupac „Date“ razdvojio u 3 zasebna stupca mogu se iskoristiti novonastali stupci za dobivanje novih podataka.

Najprije će se uz pomoć varijable „month“ kreirati novi stupac koji će pokazivati da li je mjesec u kojem se trguje posljednji u kvartalu.

```
dionice['kraj_kvartala'] = np.where(dionice['month']%3==0,1,0)
dionice.head()
```

Slika 17. Naredba za kreiranje stupca „kraj_kvartala“

Izvor: autor

Nakon što se unese gore prikazani kod dobiva se dodatni stupac s podacima koji pokazuju je li mjesec u kojem se trgovalo zadnji u kvartalu (3, 6, 9, 12). Ovaj podatak može biti od iznimne koristi za kreiranje modela iz razloga što se na kraju svakog kvartala pripremaju izvješća o poslovanju kompanija što uvelike utječe na trgovanje vrijednosnim papirima na burzi.

	Date	Open	High	Low	Close	Volume	day	month	year	kraj_kvartala
0	2010-06-29	19.000000	25.00	17.540001	23.889999	18766300	29	6	2010	1
1	2010-06-30	25.790001	30.42	23.299999	23.830000	17187100	30	6	2010	1
2	2010-07-01	25.000000	25.92	20.270000	21.959999	8218800	1	7	2010	0
3	2010-07-02	23.000000	23.10	18.709999	19.200001	5139800	2	7	2010	0
4	2010-07-06	20.000000	20.00	15.830000	16.110001	6866900	6	7	2010	0

Slika 18. Rezultati unosa stupca „kraj_kvartala“

Izvor: autor

Stupac „kraj_kvartala“ ukazuje na to da li je promatrani mjesec posljednji u kvartalu. Broj 0 pokazuje da promatrani mjesec nije posljednji u kvartalu, a broj 1 da jest.

Nakon što je kreiran stupac s podacima koji pokazuje da li je kraj kvartala može se testirati deskriptivna statistika kako bi se vidjelo ima li kakvih razlika u trgovanju i cijeni dionice ovisno o kraju kvartala.

```
dionice.groupby('kraj_kvartala').mean()
```

Slika 19. Naredba za prikaz prosječnih vrijednosti svih varijabli u usporedbi s varijablom „kraj_kvartala“

Izvor: autor

	Open	High	Low	Close	Volume	day	month	year
kraj_kvartala								
0	185.875081	189.254226	182.449499	186.085081	5.767062e+06	15.710396	6.173886	2014.816213
1	187.071200	190.232700	183.860262	187.047163	5.180154e+06	15.825000	7.597500	2014.697500

Slika 20. Rezultati naredbe sa slike 19.

Izvor: autor

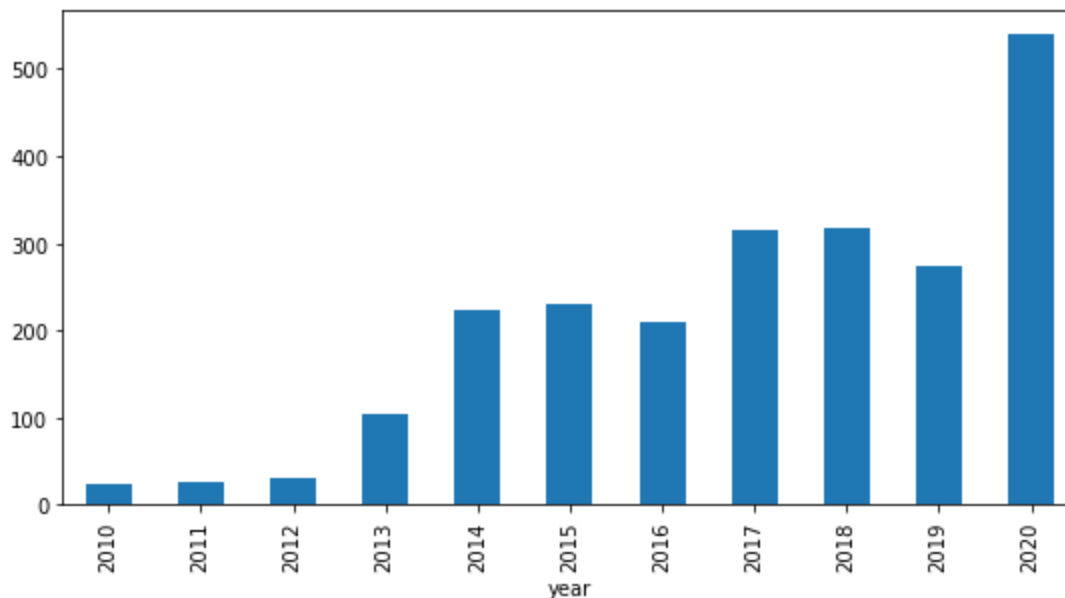
Prema dobivenim podacima može se zaključiti da je prosječna vrijednost dionice u prosjeku viša ukoliko se bliži kraj kvartala, no također je vidljivo da se za isti period u prosjeku manje trguje dionicama u odnosu na mjesece koji nisu posljednji u kvartalu.

```
data_grouped = dionice.groupby('year').mean()
plt.subplots(figsize=(20,10))

for i, col in enumerate(['Close']):
    plt.subplot(2,2,i+1)
    data_grouped[col].plot.bar()
plt.show()
```

Slika 21. Naredba za kreiranje grafičkog prikaza kretanja prosječne cijene dionice po svakoj godini

Izvor: autor



Slika 22. Godišnja prosječna cijena dionice za razdoblje od 10 godina

Izvor: autor

Na temelju gore prikazanog grafa može se zaključiti da se cijena dionica Tesle udvostručila u razdoblju od 2013. do 2014., kao i u razdoblju od 2019. do 2020. godine.

Zatim će se baza podataka proširiti s 4 dodatna stupca. Prva 2 stupca će sadržavati podatke s postotnom promjenom završne/zadnje cijene i količinom trgovanja dionicama u odnosu na prethodni dan trgovanja, dok će druga 2 stupca sadržavati prosječne vrijednosti varijabli „Volume“ i „Close“ u posljednjih 14 dana trgovanja.

```
dionice['%PromjenaClose'] = dionice['Close'].pct_change() * 100
dionice['%PromjenaVolume'] = dionice['Volume'].pct_change() * 100

dionice['14D_Mean_Volume'] = dionice['Volume'].rolling(window=14).mean()
dionice['14D_Mean_Close'] = dionice['Close'].rolling(window=14).mean()

dionice.head()
```

Slika 23. Naredbe pomoću kojih se dobivaju postotne promjene i prosječne vrijednosti dionica

Izvor: Autor

Prije nego se krene na treniranje modela dodat će se još jedan stupac u tablicu pod nazivom „Kupiti“ koji će ujedno i predstavljati izlaznu varijablu. Također se dodaju 2 stupca koji sadrže

podatke oscilacija cijena dionica.

```
dionice['Close-Open'] = dionice['Close'] - dionice['Open']
dionice['High-Low'] = dionice['High'] - dionice['Low']

def Target_var():
    dionice['Kupiti'] = 0

    uvjet1 = dionice['Close'] > dionice['14D_Mean_Close'] * 1.1
    uvjet2 = dionice['Volume'] > dionice['14D_Mean_Volume'] * 1.1
    medijan = dionice['High-Low'].rolling(window=30).median()
    uvjet3 = dionice['High-Low'] * 0.9 > medijan
    uvjet4 = dionice['Close-Open'] * 0.9 > 0

    suma = uvjet1.astype(int) + uvjet2.astype(int) + uvjet3.astype(int) + uvjet4.astype(int)
    dionice.loc[suma >= 2, 'Kupiti'] = 1

Target_var()

dionice.head()
```

Slika 24. Naredba za izračun oscilacija cijena dionica i procjenu kada treba kupovati

Izvor: autor

kraj_kvartala	%PromjenaClose	%PromjenaVolume	14D_Mean_Volume	14D_Mean_Close	Close-Open	High-Low	Kupiti
1	NaN	NaN	NaN	NaN	4.889999	7.459999	0
1	-0.251147	-8.415084	NaN	NaN	-1.960001	7.120001	0
0	-7.847256	-52.180414	NaN	NaN	-3.040001	5.650000	0
0	-12.568297	-37.462890	NaN	NaN	-3.799999	4.390001	0
0	-16.093749	33.602475	NaN	NaN	-3.889999	4.170000	0

Slika 25. Rezultati naredbe sa slike 24.

Izvor: autor

Na slici 25. može se vidjeti 5 novih stupaca, stupci „Close-Open“ i „High-Low“ koji računaju razliku zadnje i početne cijene dionice na određeni dan trgovanja, odnosno razliku najviše i najniže cijene prodane dionice za isti dan trgovanja. Stupci „%PromjenaClose“ i „%PromjenaVolume“ pokazuju vrijednosti postotnih promjena varijabli „Close“ i „Volume“, dok su stupci „14D_Mean_Volume“ i „14D_Mean_Close“ u funkciji indikatora prosječne količine trgovanja, odnosno cijene dionica u posljednjih 14 dana respektivno. Naposljetku u stupcu „Kupiti“ dobiven je izračun kada je povoljno vrijeme za kupnju dionice, broj 0 označava „NE“,

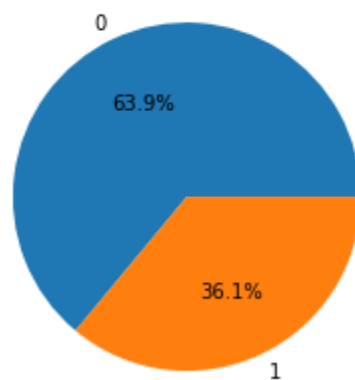
dok 1 označava „DA“. Valja napomenuti da je izlazna varijabla „Kupiti“ uvjetovana sa četiri zasebna uvjeta, te da bi se ostvario pozitivan rezultat (1) moraju se ispuniti najmanje 2 od 4 uvjeta, u protivnom varijabla „Kupiti“ poprima vrijednost 0. Prethodno spomenuti uvjeti su:

- Vrijednost varijable „Close“ mora biti veća od prosječne završne cijene dionice u prethodnih 14 dana
- Vrijednost varijable „Volume“ mora biti veća od prosječne količine trgovanja dionicom u prethodnih 14 dana
- Vrijednost varijable „High-Low“ mora biti veća od medijalne vrijednost razlike najviše i najniže dnevne cijene dionice izračunate za prethodnih 30 dana
- Vrijednost varijable „Close-Open“ mora biti veća od 0

Također za sva četiri uvjeta su dodani težinski koeficijenti.

Na slici 26. je prikazan kružni grafikon raspodjele podataka za izlaznu varijablu. Na njemu se može vidjeti da su podaci blago disbalansirani u korist klase 0.

```
plt.pie(dionice['Kupiti'].value_counts().values,  
        labels=[0, 1], autopct='%1.1f%%')  
plt.show()
```



Slika 26. Raspodjela podataka za varijablu "Kupiti"

Izvor: autor

Prije nego se krene na izgradnju modela iz tablice će se obrisati suvišni stupci koji sadrže podatke

vezane za datume, te stupci koji pokazuju visoki stupanj međusobne korelacije, takvi podaci nisu od koristi za model. Također će se obrisati prvih 14 redaka unutar svih varijabli iz razloga što u navedenim redovima nedostaju određene vrijednosti.

```
dionice = dionice.drop(['Date'], axis=1)
dionice = dionice.drop(['day'], axis=1)
dionice = dionice.drop(['month'], axis=1)
dionice = dionice.drop(['year'], axis=1)
dionice = dionice.drop(['14D_Mean_Close'], axis=1)
dionice = dionice.drop(['Low'], axis=1)
dionice = dionice.drop(['Close'], axis=1)
dionice = dionice.drop(['High'], axis=1)

dionice = dionice.iloc[14:].reset_index(drop=True)

dionice.head()
```

Slika 27. Naredbe za brisanje stupaca i redova

Izvor: autor

	Open	Volume	kraj_kvartala	%PromjenaClose	%PromjenaVolume	14D_Mean_Volume	Close-Open	High-Low	Kupiti
0	21.850000	1825300	0	-7.348247	-26.591595	5.417643e+06	-1.550001	1.800001	0
1	20.660000	1252500	0	-0.394089	-31.381143	4.279457e+06	-0.440001	1.400000	0
2	20.500000	957800	0	3.857572	-23.528942	3.760814e+06	0.500000	0.879999	1
3	21.190001	653600	0	1.380957	-31.760284	3.440371e+06	0.100000	0.500000	1
4	21.500000	922200	0	-1.596994	41.095471	3.015750e+06	-0.549999	1.200001	0

Slika 28. Konačna baza podataka prije kreiranja modela

Izvor: autor

Nakon što se u potpunosti pripremila baza podataka može se krenuti na proces izrade modela.

4.3. Kreiranje modela

Nakon što su se prilagodili svi podaci za korištenje modela može se započeti s odvajanjem podataka i treniranjem modela.

Najprije se odvajaju ulazne i izlazne varijable uz pomoć naredbi prikazanih na slici 29.

```
y1 = dionice.iloc[:, -1]  
y1.shape
```

```
(2402,)
```

```
X1 = dionice.iloc[:, :-1]  
X1.shape
```

```
(2402, 8)
```

Slika 29. Odvajanje izlazne varijable i ulaznih varijabli

Izvor: autor

Na slici 30. se nalazi kod za podjelu podataka na skupove podataka za treniranje i testiranje za ulazne i izlazne varijable. Podjela podataka za treniranje i testiranje je u omjer 75:25, što znači da u skupu podataka za treniranje ima 1801 red podataka, dok se u skupu podataka za testiranje nalazi njih 601.


```
X_train1, X_test1, y_train1, y_test1 = train_test_split(X1, y1, random_state = 0)
```

```
X_train1.shape
```

```
(1801, 8)
```

```
X_test1.shape
```

```
(601, 8)
```

```
y_train1.shape
```

```
(1801,)
```

```
y_test1.shape
```

```
(601,)
```

Slika 30. Razdvajanje podataka na skupove za treniranje i testiranje

Izvor: autor

Za treniranje podataka izabrana su sljedeća 3 modela, a to su: KNN-klasifikator, logistička regresija i neuronska mreža.

Najprije se koristi KNN klasifikator. Na slici 31. je prikazan kod za kreiranje klasifikatora koji se zatim trenira na podacima za treniranje, a prethodno je odabran k=8 kao optimalni broj susjeda.

U daljnjim rezultatima dobivene su predikcije na temelju skupa za testiranje za koji točnost iznosi 0.889.

```

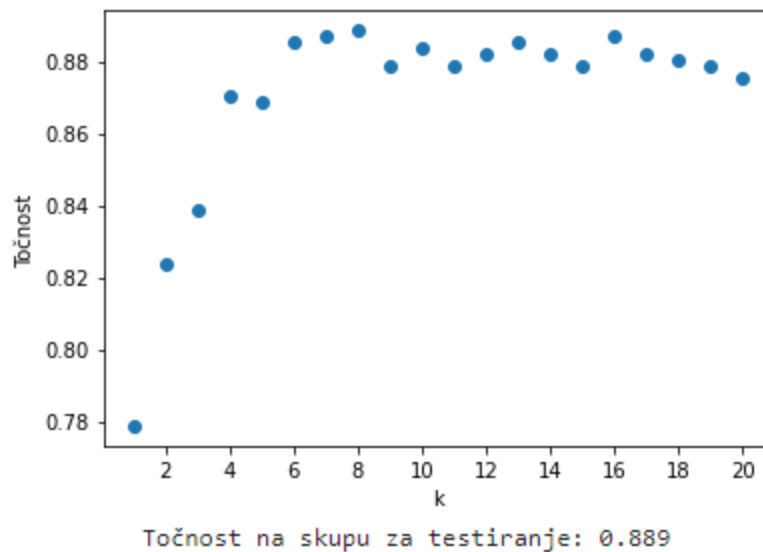
from sklearn.neighbors import KNeighborsClassifier

k_range = range(1,21)
scores = []

for k in k_range:
    knn_class = KNeighborsClassifier(n_neighbors = k)
    knn_class.fit(X_train1, y_train1)
    scores.append(knn_class.score(X_test1, y_test1))

plt.figure()
plt.xlabel('k')
plt.ylabel('Točnost')
plt.scatter(k_range, scores)
plt.xticks([2,4,6,8,10,12,14,16,18,20]);

```



Slika 31. Rezultati KNN – klasifikatora

Izvor: Autor

Zatim se kreira klasifikator logističkom regresijom, te ga se trenira na podacima za treniranje.

Na slici 32. vidljive su dobivene predikcije na temelju skupa za testiranje i točnost od 0.889 što također predstavlja vrlo zadovoljavajući rezultat.

```

from sklearn.linear_model import LogisticRegression

log_reg = LogisticRegression()

log_reg.fit(X_train1, y_train1)

print(log_reg.predict(X_test1))

print('Točnost: {:.3f}'.format(log_reg.score(X_test1, y_test1)))

[0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0
 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0
 0 1 1 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 1 1 1 0 0 0
 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 1 0 0 1
 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0
 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1
 0 1 1 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0 1
 0 0 1 0 0 1 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1 0 0 1
 0 0 1 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1
 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 1 0
 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
 1 0 1 0 1 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0
 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0 1 0 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 1 1
 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 0 0 1 1 0 0 0
 0 1 0 0 1 0 0 1 0]
Točnost: 0.889

```

Slika 32. Klasifikator logističke regresije i točnost na skupu za testiranje

Izvor: autor

Na slici 33. su prikazani rezultati za koeficijente ulaznih varijabli i intercept za ciljanu varijablu. Na temelju dobivenih rezultata može se zaključiti da jedinična ulazna varijabla ima iznimno mali utjecaj na predviđanje modela, razlog tome može biti velika količina podataka u korištenoj bazi podataka.

```

print(log_reg.coef_)
print(log_reg.intercept_)

[[-1.04903633e-10  8.05450602e-07 -5.10556278e-13  3.52043712e-12
  5.65107254e-11 -8.52632004e-07  7.58418926e-12  2.90556127e-12]]
[-8.83628503e-13]

```

Slika 33. Težinski koeficijenti varijabli

Izvor: autor

Naposljetku, kreira se klasifikator neuronskom mrežom koji ima 7 skrivenih slojeva sa 100 neurona u svakom. Najprije ga se trenira na podacima za treniranje, a zatim dobivamo predikcije na temelju skupa za testiranje.

Dobivena točnost na skupu za testiranje klasifikatora neuronskom mrežom iznosi 0.80.

```
from sklearn.neural_network import MLPClassifier

nn_class = MLPClassifier(hidden_layer_sizes = [100,100,100,100,100,100,100], random_state = 0, solver='lbfgs')

nn_class.fit(X_train1, y_train1)

print(nn_class.predict(X_test1))

print('Točnost na skupu za testiranje: {:.2f}'.format(nn_class.score(X_test1, y_test1)))

print(X_train1.shape, X_test1.shape)

[[0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0
0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1
0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1
0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0 0]
```

Točnost na skupu za testiranje: 0.80
(1801, 8) (601, 8)

Slika 34. Klasifikator - neuronska mreža

Izvor: autor

Slika 35. prikazuje matricu konfuzije na kojoj je vidljiv broj točno klasificiranih varijabli.

```

matrica_konfuzije1 = pd.crosstab(knn_class.predict(X_test1), y_test1.values)
print(matrica_konfuzije1)

matrica_konfuzije2 = pd.crosstab(log_reg.predict(X_test1), y_test1.values)
print(matrica_konfuzije2)

matrica_konfuzije3 = pd.crosstab(nn_class.predict(X_test1), y_test1.values)
print(matrica_konfuzije3)

```

col_0	0	1
row_0	374	42
1	25	160

col_0	0	1
row_0	371	39
1	28	163

col_0	0	1
row_0	394	116
1	5	86

Slika 35. Matrica konfuzije

Izvor: autor

Na slici 36. vidljive su stope točnosti za sva 3 klasifikatora koje pokazuju visoke rezultate točnosti, ali s obzirom da se radi o modelu s neuravnoteženom podjelom podataka dobiveni podaci se moraju promatrati sa zadržkom. Iz tog razloga su osmišljenje alternativne mjere vrednovanja. Tako se pored točnosti uobičajeno koriste mjere kao što su odziv i preciznost.

```

print(f'Točnost za KNN: {accuracy_knn:.4f}')
accuracy_log_reg = calculate_accuracy(matrica_konfuzije2.values)
print(f'Točnost za Logističku Regresiju: {accuracy_log_reg:.4f}')
accuracy_nn = calculate_accuracy(matrica_konfuzije3.values)
print(f'Točnost za Neuronsku mrežu: {accuracy_nn:.4f}')

```

Točnost za KNN: 0.8885
Točnost za Logističku Regresiju: 0.8885
Točnost za Neuronsku mrežu: 0.7987

Slika 36. Izračun točnosti za sva 3 modela

Izvor: autor

S obzirom da se radi o modelu s prevladavajućim vrijednostima klase 0 potrebno je izračunati stope odziva i preciznosti kako bi se dobio bolji uvid u rezultate modela.

Prethodno je već navedeno kako stopa točnosti nije uvijek prikladan pokazatelj iz razloga što je ona intuitivna mjera vrednovanja. Problem s točnošću je taj što je skup podataka neuravnotežen, što znači da u ovom slučaju ima mnogo više negativnih primjera od pozitivnih, samim time će točnost uvijek biti relativno visoka. (Šnajder, J., 2022.)

Na slici 37. vidljivi su rezultati stopa odziva i preciznosti za sva tri klasifikatora. Prije nego se interpretiraju dobivene vrijednosti, valjalo bi ukratko pojasniti spomenute mjere vrednovanja kako bi se bolje razumjelo dobivene rezultate.

Stopa odziva se definira kao udio stvarno pozitivnih primjera (TP) u skupu svih pozitivnih primjera (TP + FN). Ova se mjera naziva “odziv” jer nam govori koliko se pozitivnih primjera “odazvalo” klasifikatoru. Stopa preciznosti se definira kao udio stvarno pozitivnih primjera (TP) u skupu svih primjera koje je klasifikator označio pozitivno (TP + FP). (Šnajder, J., 2022.)

```
Stopa odziva za KNN: 0.79
Stopa odziva za Logističku Regresiju: 0.81
Stopa odziva za Neuronsku mrežu: 0.43
-----
Preciznost za KNN: 0.86
Preciznost za Logističku Regresiju: 0.85
Preciznost za Neuronsku mrežu: 0.95
```

Slika 37. Stopa odziva i preciznosti za sva tri modela

Izvor: autor

Dakle, prema dobivenim rezultatima prikazanim na slici iznad vidljive su stope odziva za KNN, logističku regresiju i neuronsku mrežu, a one iznose 0.79, 0.81 i 0.43 respektivno. Rezultati prva dva klasifikatora su relativno dobra, dok se onaj za neuronsku mrežu pokazao mnogo lošijim. Stopa odziva od 0.43 za klasifikator neuronske mreže ukazuje na to da je on predvidio više lažno negativnih (FN = 116) od stvarno pozitivnih vrijednosti (TP = 86). Drugim riječima, kod dobivenih vrijednosti za lažno negativni ishod, predikcija klasifikatora je bila 0, dok je zapravo stvarna oznaka bila 1. Kod ostala dva klasifikatora omjer stvarno pozitivnih predikcija je puno veći u odnosu na lažno negativne predikcije, što je pokazatelj da su klasifikatori imali vrlo dobar postotak

predviđanja pozitivnih, odnosno negativnih klasa.

Naposljetku, dobiveni su i rezultati za preciznost tri klasifikatora. Stope preciznosti za KNN, logističku regresiju i neuronsku mrežu su poprilično dobre s vrijednostima od 0.86, 0.85 i 0.95 respektivno. Navedeni rezultati ukazuju na to da su sva tri klasifikatora bila poprilično točna u označavanju pozitivnih primjera. Drugim riječima, klasifikatori su predvidjeli mnogo više vrijednosti klase 1 kojima je i stvarna oznaka 1, u odnosu na one kojima je stvarna oznaka 0.

5. Rasprava

Izrađene su tri metode strojnog učenja, KNN klasifikator, logistička regresija i neuronska mreža, nastojao se izgraditi model za predviđanje cijena dionica kako bi se koristio u procesu donošenja odluka prilikom trgovanja vrijednosnim papira. Sve zajedno rezultati triju modela su pokazali zadovoljavajuće rezultate i ogroman potencijal kao sredstvo kojim bi se moglo koristiti kao pomoć pri donošenju odluka u procesu trgovanja dionicama. Alati u Python okruženju su od iznimne koristi, te njihova moć oblikovanja, sortiranja, analiziranja i procesiranja podataka je gotovo neograničena, te kao takva predstavlja vrlo moćan alat u donošenju odluka.

Iako su se modeli bazirali na iscrpnoj bazi podataka, postoji još dosta mjesta za napredak, rezultati na skupovima za treniranje i testiranje kod sva 3 modela su prihvatljivi, dok su oni kod stope odziva pokazali nešto lošije vrijednosti za neuronske mreže. Niska stopa odziva je pokazatelj da je model predvidio puno manje pozitivnih predviđanja, nego što ih zapravo ima, ali isto tako s visokom stopom preciznosti ta predviđanja su uglavnom točna, odnosno ima manje lažno pozitivnih, međutim model je propustio prepoznati veliki broj stvarnih pozitivnih predviđanja.

Neki od razloga za takav rezultat mogu biti loša priprema podataka, nedostatak ključnih varijabli ili krivo formiranje izlazne varijable. No, isto tako valja napomenuti da su se preostala 2 modela pokazala kao vrlo obećavajuća i potencijalno vrlo korisna.

Prednost ovog programskog rješenja potječe iz vrlo jednostavnog korištenja, odnosno implementacije i primjene. Programski jezik Python je moguće vrlo lako koristiti u programu Anaconda, čiji glavni aduti su jednostavnost, ali i dostupnost s obzirom da je program besplatan. Također, u istom okruženju mogu se dobiti dobri rezultati točnosti, ali za to je potrebna kvalitetna i iscrpna baza podataka, te dobro utrenirani model koji je pomno izabran za primjenu kojoj najbolje odgovara.

Glavni nedostaci ovog programskog rješenja su greške prilikom predviđanja. Modeli ne uspijevaju postići željenu preciznost, te je ona nerijetko ispod 90%, kako bi se to poboljšalo potrebno je povećati broj ulaznih varijabli. Mora se znati da postoje mnogi različiti modeli kojima se može pristupiti treniranju podataka i dobivanju rezultata, a da bi se to na pravi način iskoristilo potrebno je mnogo znanja i iskustva.

6. Zaključak

Umjetna inteligencija svakim danom napreduje sve više, implementirala se u moderno društvo do te mjere da se koristi u svakom aspektu života. Nepobitna je činjenica da je umjetna inteligencija sadašnjost i budućnost razvitka tehnologije.

Upravo iz tog razloga se u ovom radu najprije spominje rana prošlost umjetne inteligencije, njezin začetak i rani razvoj, te naposljetku njezina konačna transformacija u modernu tehnologiju. Također su se nabrojali prednosti i nedostaci umjetne inteligencije u stvarnom životu, te se dotaklo pitanja da li će ona u skoroj budućnosti zamijeniti čovjeka.

U drugom dijelu rada, objašnjene su neke od metoda i algoritmi koji se uvelike primjenjuju u modernoj tehnologiji. Tako su se objasnile i metode strojnog učenja i algoritmi kao što su logistička regresija, kNN algoritam i neuronske mreže. Nastavno na modele u ovom radu dotaklo se programskog rješenja Python koji se koristio za kreiranje i testiranje prethodno navedenih modela.

U posljednjem dijelu rada provedeno je istraživanje na temelju kretanja cijena dionica kompanije Tesla u razdoblju od 10 godina. Najprije su se objasnili parametri koji su dostupni u korištenoj bazi podataka, zatim se krenulo na daljnju obradu podataka. Dobiveni podaci su najprije sortirani i klasificirani kako bi se lakše njima upravljalo, te se nakon toga pristupilo statističkim analizama skupova podataka. Svi uneseni kodovi i dobiveni rezultati su detaljno objašnjeni i grafički prikazani u radu pomoću grafikona, slika i tablica.

Na samome kraju dobiveni rezultati su pokazali zadovoljavajuću točnost, ali i ostavili veliki prostor za unapređenje istih. Od tri odabrana modela najbolje rezultate su pokazali modeli KNN klasifikatora, te model logističke regresije. Da bi se došlo do boljih rezultata potrebno je više rada i znanja u izradi kvalitetnijeg modela, prikupljanju i analiziranju kvalitetnijih podataka, te povrh svega toga, potrebno je iscrpno znanje i iskustvo u području trgovanja dionicama.

7. Literatura

1. Airon, P., 2022. *Kratki vodič za razumijevanje KNN algoritma* [Mrežno]
Dostupno na: <https://www.unite.ai/hr/a-quick-guide-to-knn-algorithm/>
2. Anon., 2020. *What is Python?* [Mrežno]
Dostupno na: <https://www.python.org/doc/essays/blurb/>
3. Anon., 2021. *What is Supervise learning?* [Mrežno]
Dostupno na: <https://www.ibm.com/topics/supervised-learning>
4. Anon., 2021. *What is Unsupervised learning?* [Mrežno]
Dostupno na: <https://www.ibm.com/topics/unsupervised-learning>
5. Anon., 2022. *What is Semi-supervised learning?* [Mrežno]
Dostupno na: <https://www.altexsoft.com/blog/semi-supervised-learning/>
6. Anon., 2022. *What is the history of Artificial Intelligence (AI).* [Mrežno]
Dostupno na: <https://www.tableau.com/data-insights/ai/history>
7. Anon., 2023. *What is supervised learning?* [Mrežno]
Dostupno na: <https://www.ibm.com/topics/supervised-learning>
8. Anon., 2024. *Koja je razlika između hiperparametara i parametara modela?* [Mrežno]
Dostupno na: <https://hr.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/introduction/what-is-machine-learning/what-is-the-difference-between-hyperparameters-and-model-parameters/>
9. Anon., 2024. *Logistic Regression in Machine Learning* [Mrežno]
Dostupno na: <https://www.geeksforgeeks.org/understanding-logistic-regression/>

10. Anon., 2024. *Machine Learning Tutorial* [Mrežno]
Dostupno na: <https://www.geeksforgeeks.org/machine-learning/>
11. Bobbitt, Z., 2021. *The 3 Types of Logistic Regression* [Mrežno]
Dostupno na: <https://www.statology.org/types-of-logistic-regression/>
12. Bolf, N., 2021. *Strojno učenje* [Mrežno]
Dostupno na: <https://hrcak.srce.hr/file/382926>
13. Bozsolik, T., 2020. *Tesla Stock Data from 2010 to 2020* [Mrežno]
Baza podataka korištena za izradu modela strojnog učenja
Dostupno na: <https://www.kaggle.com/datasets/timoboz/tesla-stock-data-from-2010-to-2020>
14. Brown, S., 2021. *Machine Learning, explained* [Mrežno]
Dostupno na: <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>
15. Chai, W., 2020. *A timeline of machine learning history*. [Mrežno]
Dostupno na: <https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History>
16. Coursera, 2024. *How do Neural Networks work?* [Mrežno]
Dostupno na: <https://www.coursera.org/articles/how-do-neural-networks-work>
17. Copeland, B.J., 2023. *Artificial Intelligence* [Mrežno]
Dostupno na: <https://www.britannica.com/technology/artificial-intelligence>

18. Crabtree, M., 2023. *What is machine learning? Definition, types, tools & more* [Mrežno]
Dostupno na: <https://www.datacamp.com/blog/what-is-machine-learning>
19. Fedorychak, V., 2024. *Key Benefits of Using Artificial Intelligence for Business* [Mrežno]
Dostupno na: <https://smarttek.solutions/blog/benefits-of-using-ai-for-business/>
20. Glover, E., 2024. *What is Sentient AI?* [Mrežno]
Dostupno na: <https://builtin.com/artificial-intelligence/sentient-ai>
21. Grieve, P., 2023. *Deep learning vs. Machine learning: what's the difference?* [Mrežno]
Dostupno na: <https://www.zendesk.com/blog/machine-learning-and-deep-learning/>
22. Hussain, A., 2020. *K-Nearest Neighbors (KNN) and its Applications* [Mrežno]
Dostupno na: https://medium.com/@arman_hussain786/k-nearest-neighbors-knn-and-its-applications-7891a4a916c6
23. LaViale, T., 2023, *Deep dive on KNN: Understanding and Implementing the K-Nearest Neighbors Algorithm* [Mrežno]
Dostupno na: <https://arize.com/blog-course/knn-algorithm-k-nearest-neighbor/>
24. Marr, B., 2021. *Understanding the 4 types of Artificial Intelligence* [Mrežno]
Dostupno na: <https://bernardmarr.com/understanding-the-4-types-of-artificial-intelligence/>
25. Nisha, A., 2024. *Theory of mind AI in Artificial Intelligence* [Mrežno]
Dostupno na: <https://www.ejable.com/tech-corner/ai-machine-learning-and-deep-learning/theory-of-mind-ai-in-artificial-intelligence/>

26. Patrick, 2023., *Unsupervised Learning: Clearly Explained* [Mrežno]
Dostupno na: <https://www.alexanderthamm.com/en/blog/this-is-how-unsupervised-machine-learning-works/>
27. Pillai, A., 2024. *Tesla stock price with indicators (10 years)* [Mrežno]
Dostupno na: <https://www.kaggle.com/datasets/aspillai/tesla-stock-price-with-indicators-10-years/discussion/483377>
28. Potrimba, P., 2022. *What is Semi-supervised learning?* [Mrežno]
Dostupno na: <https://blog.roboflow.com/what-is-semi-supervised-learning/>
29. Pranjić, A., 2020. *Primjena tehnika strojnog učenja na otvorenim podacima* [Mrežno]
Dostupno na:
<https://repositorij.foi.unizg.hr/islandora/object/foi%3A6443/datastream/PDF/view>
30. Selman, H., 2022. *The history of Machine Learning- dates back to the 17th century* [Mrežno]
Dostupno na: <https://dataconomy.com/2022/04/27/the-history-of-machine-learning/>
31. Stryker, C., Kavlakoglu E., *What is AI?* [Mrežno]
Dostupno na: <https://www.ibm.com/topics/artificial-intelligence>
32. Šnajder, J., 2022., *Strojno učenje 1, 21. Vrednovanje modela* [Mrežno]
Dostupno na: https://www.fer.unizg.hr/_download/repository/SU1-2022-P21-VrednovanjeModela.pdf
33. Tuama, D., 2022. *What are libraries in Python?* [Mrežno]
Dostupno na: <https://codeinstitute.net/global/blog/what-are-libraries-in-python/>

34. Yasar, K., 2024. *What is Neural Network?* [Mrežno]

Dostupno na: <https://www.techtarget.com/searchenterpriseai/definition/neural-network>

Repozitorij s dostupnim kodom korištenim za izradu modela prikazanih u diplomskom radu nalaze se na poveznici:

<https://github.com/LukaIljic/diplomski-rad>

POPIS SLIKA I TABLICA

Slika 1. Povijest umjetne inteligencije.....	4
Slika 2. Podjela umjetne inteligencije na 2 tipa.....	8
Slika 3. Podjela strojnog učenja na kategorije.....	11
Slika 4. Nadzirano učenje vs polunadzirano učenje vs nenadzirano učenje.....	22
Slika 5. Odnos između AI-a, strojnog učenja, neuronskih mreža i dubokog učenja.....	24
Slika 6. Prikaz slojeva dubokog učenja.....	25
Slika 7. Naredba za deskriptivnu statistiku.....	28
Slika 8. Graf kretanja cijene dionica.....	30
Slika 9. Prikaz prvih 5 redaka baze podataka.....	30
Slika 10. Naredba za usporedbu koliko se redova dviju varijabli podudara.....	31
Slika 11. Naredba za brisanje stupca „Adj Close“ iz tablice.....	31
Slika 12. Naredba za kreiranje dijagrama distribucije.....	32
Slika 13. Dijagrami distribucije za varijable Open, High i Low.....	32
Slika 14. Dijagrami distribucije za varijable Close i Volume.....	32
Slika 15. Naredba kojom se uz pomoć varijable „Date“ kreiraju 3 dodatna stupca „day“, „month“, „year“.....	33
Slika 16. Rezultat naredbe split.....	33
Slika 17. Naredba za kreiranje stupca „kraj_kvartala“.....	34
Slika 18. Rezultati unosa stupca „kraj_kvartala“.....	34
Slika 19. Naredba za prikaz prosječnih vrijednosti svih varijabli u usporedbi s varijablom „kraj_kvartala“ Izvor: autor.....	34
Slika 20. Rezultati naredbe sa slike 19.....	35
Slika 21. Naredba za kreiranje grafičkog prikaza kretanja prosječne cijene dionice po svakoj godini.....	35
Slika 22. Godišnja prosječna cijena dionice za razdoblje od 10 godina.....	36
Slika 23. Naredbe pomoću kojih se dobivaju postotne promjene i prosječne vrijednosti dionica.....	36
Slika 24. Naredba za izračun oscilacija cijena dionica i procjenu kada treba kupovati.....	37
Slika 25. Rezultati naredbe sa slike 24.....	37
Slika 26. Raspodjela podataka za varijablu "Kupiti".....	38
Slika 27. Naredbe za brisanje stupaca i redova.....	39
Slika 28. Konačna baza podataka prije kreiranja modela.....	39
Slika 29. Odvajanje izlazne varijable i ulaznih varijabli.....	40
Slika 30. Razdvajanje podataka na skupove za treniranje i testiranje.....	41
Slika 31. Rezultati KNN – klasifikatora.....	42
Slika 32. Klasifikator logističke regresije i točnost na skupu za testiranje.....	43
Slika 33. Težinski koeficijenti varijabli.....	43
Slika 34. Klasifikator - neuronska mreža.....	44
Slika 35. Matrica konfuzije.....	45
Slika 36. Izračun točnosti za sva 3 modela.....	45
Slika 37. Stopa odziva i preciznosti za sva tri modela.....	46
Tablica 1. Opis svih varijabli iz baze podataka.....	28
Tablica 2. Deskriptivna statistika baze podataka.....	29