METODE STROJNOG UČENJA ZA KLASIFIKACIJU CIJENA MOBILNIH UREĐAJA

Pejić, Ružica

Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Diplomski studij Poslovna informatika

Ružica Pejić

METODE STROJNOG UČENJA ZA KLASIFIKACIJU CIJENA MOBILNIH UREĐAJA

Diplomski rad

Osijek, 2021.

Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Diplomski studij Poslovna informatika

Ružica Pejić

METODE STROJNOG UČENJA ZA KLASIFIKACIJU CIJENA MOBILNIH UREĐAJA

Diplomski rad

Kolegij: Sustavi poslovne inteligencije

JMBAG: 0010217506

e-mail: rpejic@efos.hr

Mentor: doc.dr.sc. Slobodan Jelić

Osijek, 2021.

Josip Juraj Strossmayer University of Osijek

Faculty of Economics in Osijek

Graduate Study of Business Informatics

Ružica Pejić

MACHINE LEARNING METHODS FOR PRICE CLASSIFICATION OF MOBILE DEVICES

Graduate paper

Osijek, 2021.

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 rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu 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 institucijskom 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 znanstvenoj djelatnosti i visokom obrazovanju, NN br. 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).
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 studentice: Ružica Pejić

JMBAG: 0010217506

OIB: 09478234335

e-mail za kontakt: rrpejic.97@gmail.com

Naziv studija: Diplomski sveučilišni studij Poslovna informatika

Naslov rada: Metode strojnog učenja za klasifikaciju cijena mobilnih uređaja

Mentor/mentorica diplomskog rada: doc. dr. sc. Slobodan Jelić

U Osijeku, 24.06.2021. godine

Potpis Ružica Pejić

SAŽETAK

U radu je teoretizirana umjetna inteligencija kao i samo strojno učenje. Prikazane su osnovne podjele strojnog učenja te su jasnije objašnjene metode nadziranog učenja i sama klasifikacija. Također, dana je matematička podloga modela neuronske mreže, logističke regresije i algoritma k-najbližih susjeda, koji su korišteni u izgradnji modela strojnog učenja za potrebe ovoga diplomskog rada. Cilj rada je doći do najoptimalnijeg modela koji rješava konkretan problem predviđanja te ga evaluirati koristeći za to procijenjene parametre. Ubrzanim razvojem informacijsko komunikacijske tehnologije, dolazi do prikupljanja različitih podataka, kako o ljudima i pojavama, tako i o fizičkim proizvodima koji se prodaju putem različitih kanala. Zbog potrebe za što bržom obradom podataka i donošenjem što bržih i točnijih poslovnih odluka, u poslovne sustave se implementiraju sustavi poslovne inteligencije temeljeni na strojnom učenju. Problem optimizacije cijena u stvarnom vremenu je zapravo svakodnevna pojava te područje koje se svaki dan poboljšava.

Ključne riječi: strojno učenje, neuronske mreže, logistička regresija, KNN algoritam, evaluacija, predikcija

ABSTRACT

The paper theorizes artificial intelligence as well as machine learning itself. The basic divisions of machine learning are presented and the methods of supervised learning and the classification itself are more clearly explained. Also, the mathematical basis of neural network models, logistic regression, and algorithms of k-nearest neighbours, which were used in the construction of machine learning models for this thesis, are given. The aim of this paper is to arrive at the most optimal model that solves a specific problem by predicting the estimate using the predicted parameters. With the accelerated development of information and communication technology, various data are being collected, both about people and phenomena, and about physical products that are sold through various channels. Due to the need for faster data processing and making faster and more accurate business decisions, business intelligence systems based on machine learning are implemented in the business system. The problem of real-time price optimization is the daily occurrence of an area that is improving every day.

***Keywords**: machine learning, neural networks, logistic regression, KNN algorithm, evaluation, prediction*

SADRŽAJ

1. Uvod

Strojno učenje, kao grana umjetne inteligencije, je metoda koja se sve češće koristi u poslovanju, kako bi se iskoristio veliki broj dostupnih podataka, osobito onih koji se generiraju putem interneta. Shodno tome, može se reći kako je u modernom poslovanju, podatak ključni resurs, koji ukoliko se pravilno iskoristi stvara informaciju pogodnu za donošenje poslovnih odluka. S obzirom na sve ne izvjesniju i dinamičniju poslovnu okolinu, poslovne odluke je potrebno donositi u što kraćem roku kako bi se postigao uspjeh na tržištu te u konačnici osigurao opstanak poslovnog subjekta.

1.1. Identifikacija problema i istraživačka pitanja

Velik broj poslovnih subjekata koristi podatke u smislu deskriptivne i inferencijalne statistike, ali nerijetko i na svoja web sjedišta ugrađuju algoritme strojnog učenja kako bi automatizirali dio usluge koju putem weba pružaju korisniku. Najbolji primjer za to su Netflix i Amazon sa svojim svjetski poznatim sustavom predviđanja budućih izbora korisnika. U kategoriju usluga temeljenih na strojnom učenju pripadaju i automatski prevoditelji, bilo da je riječ aplikaciji kao što je Google prevoditelj ili automatskom prevođenju teksta na društvenim mrežama. Prema tome, može se zaključiti kako je umjetna inteligencija već danas svakodnevna pojava na kojoj se temelji uspješnost mnogih poslovnih subjekata.

Brojna istraživanja i statističke analize ukazuju na eksponencijalan rast internetski generiranih podataka što ima za posljedicu potrebu za njihovom analizom i obradom, što je opet posljedica ubrzanog razvoja tehnologije, globalizacije te povezivanja poslovnih subjekata na više razina (međusobno, prema klijentima/korisnicima ili državi, tj. vlasti). U nesagledivo velikim bazama strukturiranih i nestrukturiranih podataka, potrebno je pronaći podatke koji su uistinu reprezentativni te kao takvi mogu doprinijeti unapređenju poslovanja. Odgovor na tu problematiku daje podatkovna znanost (eng. *Data science*) koja temeljem matematičke metodologije, a posredstvom informacijsko komunikacijske tehnologije pruža mnoge mogućnosti kada su podatci u pitanju. Dakako, najefikasniji način obade podataka leži upravo u algoritmima strojnog učenja koji su najpoznatija vrsta umjetne inteligencije.

Kada je u pitanju konkretan poslovni problem, primjerice određivanje cijena, što je i tema ovoga diplomskog rada, mnogi manji poslovni subjekti posežu za klasičnim metodama te se njihova prodajna cijena temelji na nabavnoj cijeni i previđenoj zaradi od prodaje određenog proizvoda ili usluge. Kako je strategija određivanja cijena nekoga proizvoda ili usluge jedno od najvažnijih

pitanja u poslovanju, ovaj rad se bavi istraživanjem optimizacije cjenovnog ranga nekog proizvoda, u ovome slučaju mobilnih uređaja, s obzirom na njegove fizičke karakteristike.

U tom pravcu razmišljanja, umjetna inteligencija na najbolji način može optimizirati cijenu određenog proizvoda ili usluge u danom trenutku kako bi se postiglo zadovoljstvo kupca, te u konačnici postigao profit. Ono što je velika prednost kod ovakvog sustava određivanja cijena je brzina cijelog postupka, koji je takoreći automatiziran. Svakodnevnim proširivanjem baza podataka, algoritmi strojnog učenja postaju sve točniji te su u mogućnosti dati preciznije odgovore na postavljena pitanja. S druge strane, potrebno je voditi računa o zastarijevanju podataka i trenutku kada određene opservacije više nisu relevantne za predviđanje, tj. za donošenje prave poslovne odluke. Naglasak na tome je puno veći danas nego prije, ponajviše zbog socio-ekonomskih promjena, uzrokovanih razvojem tehnologije, što opet rezultira promjenama u ponašanju potrošača.

Polazeći od metodologije koja stoji u pozadini strojnog učenja, u radu će se nastojati doći do odgovora na sljedeća istraživačka pitanja:

„U kojoj mjeri umjetna inteligencija doprinosi unapređenju poslovanja?“

„Koliko su predviđanja točna te kako ih tumačiti?“

„Koja metoda je najpogodnija za rješavanje danog problema?“

1.2. Cilj rada

Cilj rada je doći do što cjelovitijih odgovora na postavljena istraživačka pitanja. U tu svrhu će se na skupu strukturiranih podataka kreirati modeli strojnog učenja, a dobiveni rezultati će biti uspoređeni te analizirani. Temeljem rezultata modela strojnog učenja, bit će navedene prednosti i nedostatci istraživanja, te smjernice za daljnja istraživanja kao i osvrt na mogućnost implementacije modela u praksi. Treba napomenuti kako postoji veliki broj istraživanja na ovu temu koja će biti predstavljena u poglavlju Teorijska podloga i prethodna istraživanja.

2. Teorijska podloga i prethodna istraživanja

2.1. Povijest strojnog učenja i umjetne inteligencije

Kako je već poznato, sama podatkovna znanost je interdisciplinarno područje koje su utemeljili mnogi matematičari i statističari. Među najpoznatijim je definitivno Thomas Bayes i njegovo promišljanje o vjerojatnosti budućih događaja zasnovanoj na prethodnim događajima. Kasnije je o istom problemu promišljao i Pierre Simon Laplace te neovisno došao do istog zaključka. (Good, 1980, str. 489-519) Matematička formulacija, nazvana je Bayesov teorem koji se koristi u teoriji vjerojatnosti, te također u strojnom učenju.

Godine 1950. Arthur Samuel iz IBM-a razvio je računalni program za igranje dame, te kako memorija računala nije bila dostatna za generirane podatke, Samuel je kreirao funkciju koja je bodovanjem pokušavala izmjeriti šanse za pobjedu svake strane. Program odabire sljedeći potez koristeći MinMax strategiju koja je kasnije evoluirala u MinMax algoritam. (McCarthy & Feigenbaum, 1990, str. 10-10) Smatra se kako je upravo Samuel tvorac izraza „strojno učenje“.

Frank Rosenblatt bio je psiholog koji poznat po interesu za umjetnu inteligenciju te radu na njezinom razvoju. Godine 1957. kombinirao je Hebbov model interakcije moždanih stanica sa Samuelovom idejom strojnog učenja i stvorio *perceptron* koji je u početku bio zamišljen i planiran kao stroj, a ne kao program namijenjen prepoznavanju slika. S obzirom na tadašnju ograničenu tehnologiju patent nije ispunio očekivanja te se nešto ambiciozniji pothvati pojavljuju tek kasnije, 1990-ih godina. (Kanal, 2003, str. 1383-1384)

Godine 1967. kreiran je jedan od poznatijih algoritama strojnog učenja, algoritam k-najbližih susjeda (KNN algoritam). Zaslugu za kreiranje algoritma dobio je Marcello Pelillo, a više o samom algoritmu bit će rečeno u nastavku rada.

U svrhu pojašnjavanja razvoja područja strojnog učenja kao i umjetne inteligencije, potrebno je naglasiti kako i kada je došlo do razdvajanja ova dva pojma. Naime, do početka 1980-ih godina, strojno učenje se koristilo kao program obuke umjetne inteligencije, no u to vrijeme znanstvenici tehničkih usmjerenja, napustili su istraživanje neuronskih mreža. Ono što je bitno za ovo razdoblje je usredotočenost istraživača umjetne inteligencije na znanje, a ne na algoritme. Kasnije je koncept strojnog učenja prenamijenjen za rješavanje praktičnih problema i pružanje različitih usluga. (Foote, 2019)

U kasnijim godinama, napretkom tehnologije, došlo je i do značajnog razvoja strojnog učenja, primjerice kreirani su algoritmi za prepoznavanje slike ili govora koji su uistinu radili, a danas

su svakodnevica. Shodno tome, može se reći kako je upravo strojno učenje zaslužno za najveće tehnološke dosege 21. stoljeća što je imalo za posljedicu razvoj gospodarstva u globalu, olakšavanje svakodnevnog života ali i poslovnih procesa. Nekoliko je glavnih područja korištenja strojnog učenja danas (Schachter, 2018):

- Analiziranje podataka o prodaji (eng. *Analyzing sales data*)– transkacijske baze, analiza web sjedišta
- Mobilna personalizacija u stvarnom vremenu (eng. *Real-time mobile personalization*)– personalizirani pristup kupcima kroz web i mobilne aplikacije
- Detekcija internetskih prijevara (eng. *Fraud detection*)– zaštita podataka
- Preporuke proizvoda (eng. *Product recommendations*) – Netflixov algoritam predviđanja budućih izbora
- Sustavi upravljanja učenjem (eng. *Learning management systems*)– sustavi za tzv. e-učenje
- Dinamične cijene (eng. *Dynamic pricing*)– predviđanje i prilagodba cijena u stvarnom vremenu kao odgovor na potrebe tržišta
- Obrada govorenog jezika (eng. *Natural language processing*) – automatski odgovori tzv. *chatbot*

2.2. Dinamične cijene kao okosnica poslovanja

Ono na što se izravno veže na temu ovoga diplomskog rada je već spomenuti pojam dinamičnih cijena. Jedna od definicija kaže kako su „dinamične cijene strategija u kojoj se cijene proizvoda kontinuirano prilagođavaju, ponekad u nekoliko minuta, kao odgovor na ponudu i potražnju u stvarnom vremenu. Misao vodilja ovoga koncepta je fleksibilnost ponude koja je bazirana na podatcima generiranim u stvarnom vremenu. “ (Khan, 2021.)

Koncept dinamičnih cijena najčešće je implementiran u sustave e-trgovine, te daje mnoge prednosti (Khan, 2021.):

- **Daje veću kontrolu** kada je riječ od strategiji upravljanja cijenama, koja se ogleda u stalnoj dostupnosti podataka o transakcijama, kupcima i proizvodima.
- **Mogućnost fleksibilnosti** bez narušavanja ugleda brenda – kreiranje sezonskih cijena, popusta što je nešto teže izvedivo bez modela dinamičnih cijena.
- **Dugoročno smanjenje troškova**, s obzirom da su izračuni i kategorizacije izvedeni u stvarnom vremenu pomoću različitih programskih rješenja, štedi se vrijeme te pravovremeno odgovara na potražnju.

- **Efektivno upravljanje** kroz programsko rješenje koje automatizira cijeli proces te pruža točne podatke u cilju optimizacije cijena.
- Pravilno upravljanje strategijom **dugoročno povećava prodaju,** a samim time i profit.

Za dinamično određivanje cijena koriste se različiti matematički modeli koji uglavnom ovaj problem formuliraju kao problem optimizacije. U ovisnosti od vrste matematičkog alata korištenog u procesu modeli se mogu podijeliti na pet kategorija (Narahari, Raju, Ravikumar, & Shah, 2005., str. 237)

- Modeli zasnovani na zalihama kod kojih se odluke o cijenama prvenstveno temelje na razinama zaliha.
- Modeli koji se temelje na statističkoj analizi dostupnih podataka o preferencijama kupaca te obrascima ponašanja kako bi se došlo do izračuna optimalnih cijena.
- Modeli teorije igara koji se ogledaju u natjecanju za istu skupinu kupaca od strane različitih poslovnih subjekata što dovodi do dinamike u cijenama.
- Modeli strojnog učenja koji se zasnivanju na prikupljanju korisnih podataka o kupcima, njihovim preferencijama i kupnjama, kao i ponudi od strane poslovnog subjekta u vidu karakteristika proizvoda (primjerice, Dell prodaje proizvode prilagođene zahtjevima kupaca, na način da različite konfiguracije imaju različite cijene).
- Simulacijski modeli mogu koristiti bilo koji od prethodna četiri modela za oponašanje dinamike cjelokupnog sustava.

Stavljajući u isti kontekst pojmove kao što je proizvod, cijena i strojno učenje vidljivo je kako je određivanje cijena pomoću metoda strojnog učenja svakodnevna pojava koja pomaže generirati profit pogotovo kada je u pitanju e-trgovina.

2.3. Rezultati prethodnih istraživanja temeljenih na strojnom učenju

S obzirom na činjenicu kako je strojno učenje nezaobilazan dio poslovanja 21. stoljeća, publikacije i istraživanja vezana za različite tehnike, metode i modele su mnogobrojne. U svrhu teme oko određivanja cijena u nastavku će biti prikazana istraživanja koja se bave ovim područjem.

Jedno od istraživanja bavi se predviđanjem cijene kuća s obzirom na karakteristike nekretnina (cijena, godina gradnje, broj soba, površina, broj kupaona, broj parkirnih mjesta i slično). Izlazna varijabla je dakako cijena koja je kontinuirana varijabla, pa je u pitanju regresijski model u kome je korištena linearna regresija (eng. *Linear Regression*), model stabla odlučivanja

(eng. *Decision tree*), neuronska mreža (eng. *Neural network*), a za izgradnju modela je korišten R programski jezik. Iz rezultata je vidljivo kako „regresijsko stablo daje rezultat predviđanja jednako dobar kao i linearna regresija, dok polinomijalna regresija rezultira manjim pogreškama što je prihvatljivo. Nadalje, čini se da neuronska mreža ne djeluje učinkovito s ovim skupom podataka.“ (Phan, 2018., str. 39)

Što se tiče istraživanja koja se bave istim alatom i metodologijom, međutim društvenim područjem, valja istaknuti istraživanje vezano za predviđanje (ne)dolaska gostiju na društvene događaje. Riječ je o detekciji potencijalnih sudionika te bilježenja njihovog interesa, ali i praćenje aktivnosti kao što je otvaranje e-pošte. U istraživanju je također korišteno više metoda: stabla odlučivanja, metoda slučajne šume (eng. *Random forest*), indukcija pravila (eng. *Rule induction*) te metoda potpornih vektora (eng. *Support vector machine*). Zaključeno je kako bi „rezultati bili bolji kada bi se prikupljali dodatni atributi na temelju kojih bi se poboljšao skup za treniranje i skup za testiranje. Navedeno jest očekivana problematika obzirom da se radi o društvenom području jer bi promatranjem sustava u tehničkom području skup značajki za razne sustave bio veći.“ (Kraljević & Ognjen, 2020., str. 39)

Osim toga, metode strojnog učenja se koriste i u svrhu predviđanja drugih društvenih pojava, kao što je odluka o poduzetničkoj karijeri. Korištenjem metoda kao što je k-najbližih susjeda (eng. *K-nearest neighbours*) i metoda potpornih vektora te neuronske mreže za klasifikaciju, a u „svrhu ispitivanja generalizacijske sposobnosti modela, 10 različitih skupova podataka nasumično je generirano iz početnog skupa podataka postupkom deseterostruke unakrsne validacije (eng. *Cross-validation*) tako da se u svrhu provjere valjanosti koriste različita 44 slučaja podataka. Svaka od četiri metode klasifikacije provedena je na 10 skupova podataka generiranih u 10-strukom CV postupku.“ (Zekić-Sušac, Sanja, & Šarlija, 2014., str. 89) Zaključeno je kako je najefikasniji model neuronske mreže sa prosječnom točnošću od 0,7797.

Nadalje, nekoliko je publikacija vezanih uz metode strojnog učenja za cijene mobilnih uređaja gdje su korišteni različiti klasifikatori za postizanje veće točnosti modela. Za izgradnju modela korišteni su podatci preuzeti sa portala GSMArena.com, a varijable su slične varijablama koje se nalaze u bazi podataka koja je korištena za pisanje ovoga diplomskog rada. U ovome slučaju korištena su dva klasifikatora i to stablo odlučivanja i *Naive Bayes* klasifikator, a u oba slučaja postignuta je točnost veća od 70%. „Kako bi se postigla maksimalna točnost, i kako bi predikcija bila što točnija potrebno je proširiti bazu podataka novim instancama. Također, odabir prikladnijih značajki bi povećao točnost.“ (Asim & Zafar, 2018., str. 11) Evidentno je

kako je za veću točnost predikcije potreban dovoljno velik skup podataka, i prava metoda što u ovome primjeru nije slučaj.

Osim ovoga, drugi znanstvenici su pokušavali optimizirati predviđanje cijene mobilnih uređaja korištenjem različitih baza kako bi došli do što veće točnosti. Neki su „izvukli jednogodišnju cijenu za deset vrsta mobilnih telefona i upotrijebili izvorne podatke za predviđanje cijene na temelju tehnike nazvane *Adaptive sliding windows*. Prosječna točnost ovoga predviđanja postiže 99,4%, a istraživanja su bila korisna za potrošače i za tvrtke na tržištu mobilnih uređaja.“ (Yin, Jiajun, & Zhu, 2012., str. 247-248)

Druga istraživanja na ovu temu su jako slična, a razlikuju se po korištenju nešto drugačijih modela kao što je metoda potpornih vektora u kojoj je r^2 vrijednost[1] u odnosu na ostale metode bila najveća, odnosno iznosila je 0,93 na skupu za testiranje. (Chandrashekhara, Thungamani, Babu, & Manjunath, 2019., str. 370)

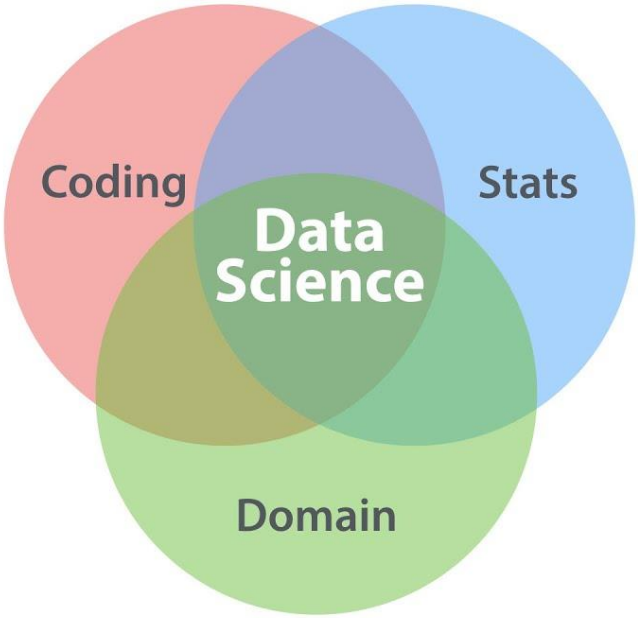
[1] Linearni korelacijski koeficijent označava se slovom r te poprima vrijednosti od -1 do +1 gdje 0 znači ne postojanje povezanosti između promatranih varijabli dok vrijednosti bliže -1, odnosno +1 označavaju jaču povezanosti. Kvadrirana r vrijednosti (r^2) se naziva koeficijent determinacije ili jačina povezanosti.

3. Metodologija rada

U ovome poglavlju bit će opisane metode strojnog učenja koje su korištene za potrebe ovoga diplomskog rada, ali i druge vrste strojnog učenja radi konteksta i boljeg definiranja samoga problema.

3.1. Strojno učenje

Kompleksnost podatkovne znanosti kao i područja njezine primjene rezultirala su brojnim granama koje su međusobno povezane. Ono što je zajedničko svim granama znanosti o podatcima je zahtijevanje širokog spektra znanja, kao što je prikazano na *Slika 1*. Područje preklapanja matematičkih, tehnoloških, i domenskih znanja odlikuje ovu disciplinu i daje joj posebnost te je kao takva široko primjenjiva u praksi.



Slika 1 Podatkovna znanost - Vennov dijagram
Izvor:https://i.ytimg.com/vi/r2I3IDKwyMw/maxresdefault.jpg

Bitno je uočiti kako se podatkovna analiza koja se i sama često naziva podatkovnom znanošću odnosi na izvlačenje korisnih informacija uvidom u velike količine podataka. S druge strane, „strojno učenje pomaže u preciznom predviđanju ili klasificiranju ishoda za nove instance pomoću obrazaca učenja iz povijesnih podataka.“ (Chatterjee, 2020.)

„Strojno učenje je brzorastuća grana računalnih algoritama koji su dizajnirani da oponašaju ljudsku inteligenciju učeći se iz okruženja. Smatraju se radnom snagom u novoj eri takozvanih velikih podataka (eng. *Big data*).“ (El Naqua & Murphy, 2015., str. 3) Za razliku od prirodnog procesa učenja strojno učenje odlikuje brzina i akumulacija znanja, tj. podataka zapisanih u

različitim bazama podataka. Prema (El Naqua & Murphy, 2015.), stupnjevi složenosti procesa mogu varirati i uključivati nekoliko faza sofisticirane interakcije čovjek-stroj te donošenje odluka što prirodno poziva na upotrebu algoritama strojnog učenja za optimizaciju i automatizaciju poslovnih procesa.

Algoritmi strojnog učenja mogu poprimiti različite matematičke formulacije, te se razlikuju i po tehnikama i metodologijama kako bi se izgradio model za rješavanje stvarnih problema na temelju podataka. Obično se metode strojnog učenja mogu klasificirati na više načina, a u nastavku su prikazana neka od glavnih područja ove discipline.

- Metode temeljene na količini ljudskog nadzora (eng. *supervision*) u procesu učenja:
    - Nadzirano učenje (eng. *Supervised learning*)
    - Nenadzirano učenje (eng. *Unsupervised learning*)
    - Polu-nadzirano učenje (eng. *Semi-supervised learning*)
    - Učenje podrškom (eng. *Reinforcement learning*)
- Metode koje se temelje na sposobnosti učenja iz inkrementalnih uzoraka podataka:
    - Skupno učenje (eng. *Batch learning*)
    - Mrežno učenje (eng. *Online learning*)
- Metode temeljene pristupu generalizaciji:
    - Učenje temeljeno na primjeru (eng. *Instance based learning*)
    - Učenje temeljeno na modelu (eng. *Model based learning*) (Sarkar, Bali, & Sharma, 2018., str. 35)

Modeli strojnog učenja korišteni za potrebe diplomskog rada pripadaju metodi tzv. nadziranog učenja koja će detaljnije biti objašnjena u nastavku rada.

3.2. Nadzirano učenje

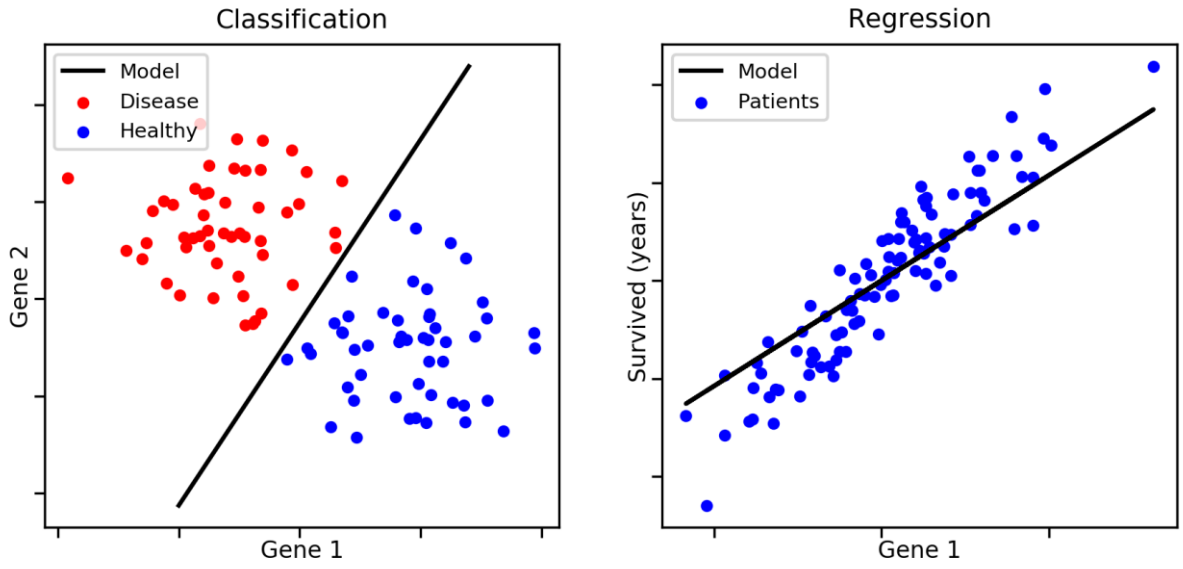
Može se reći kako je nadzirano učenje najvažnija metodologija strojnog učenja. „Ono podrazumijeva učenje mapiranja između skupa ulaznih varijabli X i izlazne varijable Y (varijable cilja) i primjenu ovog mapiranja za predviđanje rezultata za neviđene podatke.“ (Cunningham & Cord, 2008., str. 21) Riječ je dakle, o dijeljenju podataka na skupove za treniranje i skupove za testiranje na kojima će algoritam provjeriti naučene obrasce. Ono što je najvažnije značajka ove metodologije je činjenica kako je vrijednost izlazne varijable već poznata.

Dakle, varijabla cilja je varijabla čiju vrijednost algoritam strojnog učenja predviđa te u ovisnosti o njezinoj vrsti, metode strojnog učenja se također razlikuju. Sve varijable, pa i varijabla cilja mogu biti kvalitativne i kvantitativne.

Podatci u kvantitativnim varijablama se iskazuju brojevima, te se one dijele na diskontinuirane (podatci koji se izražavaju cijelim brojem) i kontinuirane (numerički podatci koji se osim cijelim brojem, mogu izraziti i decimalnim brojem).

Kad je riječ o kvalitativnim varijablama, podatci u njima se iskazuju riječima (tj. slovnim oznakama). Takvi podatci se ne mjere numeričkom ljestvicom već se razvrstavaju u kategorije varijabli prema određenoj karakteristici ili atributu. Ove varijable nazivaju se još i kategorijalne varijable. (Horvat & Mijoč, 2019., str. 38-39)

U ovisnosti o vrsti kojoj pripada varijabla cilja (kvalitativna ili kvantitativna), u strojnom učenju razlikuju se problemi klasifikacije i problemi regresije. Prema tome, ukoliko je predmet predviđanja konkretna cijena nekog proizvoda izražena u brojevima, tada je riječ o regresijskom modelu, međutim ukoliko je riječ o predviđanju cjenovnog ranga kao što je slučaj u ovome radu, u pitanju je klasifikacijski problem. Kako je prikazano na *Slika 2*, algoritam strojnog učenja namijenjen regresijskom problemu nastoji previdjeti konkretnu vrijednost što bližu postojećoj uz minimiziranje pogreške. S druge strane, algoritam strojnog učenja čiji je zadatak točno odrediti kategoriju kojoj pojedina opservacija pripada, ima za zadatak razgraničiti područje pripadanja pojedinoj kategoriji.



Slika 2 Klasifikacijski i regresijski problem
Izvor: https://aldro61.github.io/microbiome-summer-school-2017/figures/figure.classification.vs.regression.png

Statistički gledano, varijable se mogu podijeliti na zavisne i nezavisne, a stavljajući ovu podjelu u kontekst strojnog učenja, zavisna varijabla je varijabla cilja (Y), a nezavisne su sve ostale koje se koriste u svrhu predikcije zavisne varijable.

U ovisnosti o prirodi zavisne varijable, koriste se različite tehnike strojnog učenja namijenjenog klasifikacijskom problemu. Prema tome, problem klasifikacije se dijeli na (Bansal, 2021.):

Binarna klasifikacija (eng. *Binary classification*) je najjednostavniji i najčešće korišteni oblik klasifikacije. Ovdje ovisna varijabla sadrži isključivo dvije kategorije koje su označene s 1 i 0, zbog čega je i nazvana binarnom. Brojem jedan se često označava afirmacija promatrane pojave (koja može biti bilo što – trenutna dostupnost proizvoda na policama) dok je nula negacija.

Binomna klasifikacija (eng. *Binomial classification*) – Praktično gledano, ova vrsta klasifikacije se ne razlikuje od binarne. Također se sastoji od dvije kategorije koje su označene s 1 i 0, međutim u ovom slučaju one ne označavaju afirmaciju i negaciju nego se odnose na dvije nezavisne kategorije, kao što su primjerice, mobilni uređaj i računalo. Razlika je dakle, samo u interpretaciji i logičkom shvaćanju kategorija.

Višerazredna klasifikacija (eng. *Multi-class classification*) na sebi svojstven način generalizira binomnu klasifikaciju. U pitanju je napredniji oblike klasifikacije, tj. u ovome slučaju varijabla cilja se sastoji od više od dvije kategorije, a cilj predviđanja je odrediti kojoj kategoriji pojedina opservacija pripada s tim da svaka opservacija može pripadati samo jednoj kategoriji. Ovakve varijable cilja, često mogu poprimiti gradacijski izgled, primjerice niske, srednje i visoke cijene.

Klasifikacija s više oznaka (eng. *Multi-label classification*) je oblik klasifikacije sličan višerazrednoj klasifikaciji, međutim, u ovome slučaju varijabla cilja može pripadati više nego jednoj kategoriji, te algoritam ima zadaću prepoznati obrasce i razumjeti s kojim klasama se promatranje može povezati. Ovakva vrsta klasifikacije se najčešće koristi kada je u pitanju rad s podatcima tekstualnog formata kao što je novinski članak gdje opažanja mogu sadržati nekoliko svojstava kao što je, rubrika (primjerice tehnologija), osoba koja je tema članka, zemljopisno područje i drugi.

Osim toga, sami klasifikacijski algoritmi se mogu podijeliti prema funkcioniranju tj. redoslijedu izvršavanja zadataka u dvije grupe.

Eager learners – U pitanju je tipičan način učenja odnosa između ulaznih podataka i varijable cilja gdje se u jednokratnoj tehnici provjere valjanosti na skupu za treniranje otkrivaju uzorci i

uspostavlja kvantificirani odnos između zavisne i nezavisne varijable te se na taj način stvara klasifikacijski model. Kvantificirani odnos se zatim primjenjuje na skup podataka namijenjen testiranju, a glavno svojstvo ovih algoritama je da imaju nešto duži razvojni proces od samog procesa predviđanja. Neki od algoritama koji funkcioniraju na ovaj način su: logistička regresija (eng. *Logistic regression*), neuronska mreža, stabla odlučivanja itd.

Lazy learners – Za razliku od prethodno navedenih algoritama, proces predviđanja se sastoji od spremanja podataka u fazi treniranja, a uočavanje obrazaca između zavisne i nezavisne varijable se izvodi u testnoj fazi koja je u ovome slučaju nešto duža od prethodne faze. Tipičan algoritam koji funkcionira na ovaj način je algoritam k-najbližih susjeda. (Galvan, Valls, Garcia, & Isasi, 2011., str. 1-2)

3.2.1. Algoritam k-najbližih susjeda

„Klasifikator k-najbližih susjeda jednostavna je, ali učinkovita nadaleko poznata metoda u rudarenju podataka i strojnom učenju.“ (Maillo, Ramirez, Triguero, & Herrera, 2017., str. 3) Riječ je o intuitivnom, efektivnom i neparametarskom modelu koji se koristi kako za klasifikaciju tako i za regresiju. Princip po kome radi ovaj algoritam sastoji se u tome da se pripadnost određenoj kategoriji određuje obrascima učenja koji se ogledaju u udaljenosti k-najbližih testnih instanci. Obično se koristi Euklidska udaljenost kao metrika koja se računa prema *Jednadžba 1*.

$$d(\boldsymbol{p},\boldsymbol{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Jednadžba 1 Euklidska udaljenost
Izvor: https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d

gdje su p i q točke u višedimenzionalnom prostoru koje odgovaraju instancama iz skupa podataka. (Jose, 2018.)

U tom pravcu razmišljanja jako je bitno koliki je k koji određuje broj najbližih susjeda. On je najčešće proizvoljan, a težnja je odabrati k koji maksimizira točnost modela prikazanu *Jednadžba 6*. Jedan od savjeta iz struke odnosi se na parnost k parametra u smislu da je uvijek bolje da k bude neparan broj kako bi algoritam lakše odredio pripadnost određenoj klasi. Jednostavno rečeno, ukoliko je $k = 1$, algoritam će tražiti najbližu instancu te njezinu pripadnost pripisati testnoj instanci, međutim ukoliko je $k = 3$, tada će algoritam pri

određivanju kategorije u obzir uzeti tri najbliže točke te će prevagnuti ona kategorija kojoj pripada najviše susjeda (po čemu je ova metoda i dobila ime).

Pozitivne strane ove metode su što je jednostavna i lagana za interpretaciju, ne zasniva se na ni jednoj pretpostavci pa se može primijeniti i na nelinearne probleme, jednako dobro funkcionira na binarnoj i višerazrednoj klasifikaciji, te se može koristiti kako za klasifikaciju tako i za regresiju. Jedna od negativnih strana ove metode je činjenica da proces predikcije postane dosta spor kako se broj podataka povećava, budući da model bilježi pozicije svih instanci. Iz toga proizlazi memorijska neefikasnost, a valja napomenuti kako je model osjetljiv na stršeće vrijednosti (eng, *Outliers*). (Starzacher & Bernard, 2008., str. 3)

3.2.2. Logistička regresija

Logistička regresija je statistička metoda slična linearnoj regresiji, međutim za razliku od nje varijabla cilja može biti kategorijalna varijabla s dvije ili više kategorija. Izvorno, logistička regresija je metoda namijenjena binarnoj klasifikaciji, međutim, može se koristiti i u višerazrednoj klasifikaciji. Da bi predvidio pripadnost određenoj kategoriji algoritam koristi eng. *logartihm of odds ration* što se ponekad prevodi i kao logaritam šanse. (DiGangi & Moore, 2012., str. 139)

Logistička funkcija, koja se također zove i sigmoidna funkcija, razvijena je od strane statističara čija je nakana bila opisati svojstva porasta stanovništva te maksimizaciju nosivosti okoliša. To je krivulja u obliku slova S koja može uzeti bilo koji realni broj i preslikati ga u vrijednost između 0 i 1 ali nikad ne poprima njihove vrijednosti. (Kucharavy & DeGuio, 2011., str. 559-562) U nastavku slijedi *Jednadžba 2* logističke funkcije gdje je e tzv. Eulerov broj (baza prirodnog logaritma) čija vrijednost iznosi približno 2,718.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

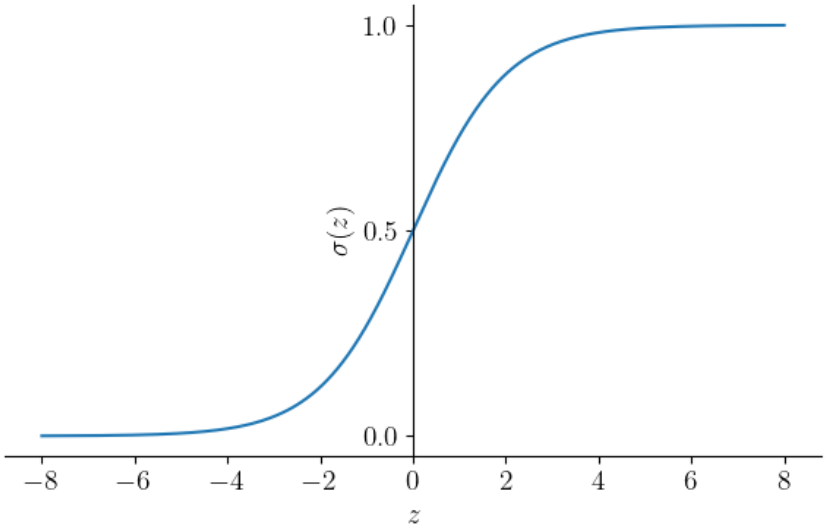
Jednadžba 2 Logistička funkcija
Izvor: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/

Logistička regresija je klasifikacijski algoritam koji se zasniva na pretpostavci kako se vjerojatnost pripadanja neke instance određenoj kategoriji može aproksimirati logističkom funkcijom koja je prikazana na *Slika 3*. Nadalje, potrebno je pronaći parametre θ koji maksimiziraju vjerojatnosti pripadanja instanci pojedinoj kategoriji, tj. koristi se eng. *Log likelihood estimation* (MLE) i to po principu koji je prikazan *Jednadžba 3*

$$L(\Theta) = \log \prod_{i=1}^{n} P\left(Y = y^{(i)} \middle| X = x^{(i)}\right)$$

Jednadžba 3 Log Likelihood Eastimation
Izvor: https://web.stanford.edu/class/archive/cs/cs109/cs109.1178/lectureHandouts/220-logistic-regression.pdf

gdje y i x označavaju zavisnu odnosno nezavisne varijable. S obzirom na to kako logistička regresija predviđa binarne vrijednosti, izlaz logističke funkcije bi trebao biti vjerojatnost da promatrana instanca pripada kategoriji 1. U tom pravcu razmišljanja, promatrana varijabla se tretira kao Bernoullijeva slučajna varijabla, $Y \sim \text{Ber}(p)$ gdje je $p = \sigma(\theta^T x)$. (Monroe, 2017.)



Slika 3Graf logističke funkcije
Izvor:https://scipython.com/static/media/uploads/blog/logistic_regression/sigmoid.png

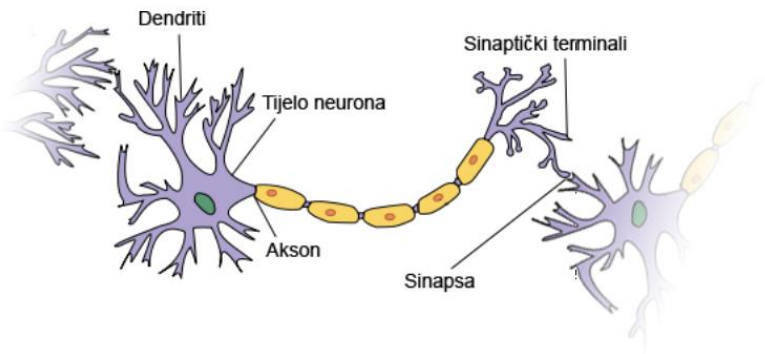
Pri izgradnji modela temeljenog na logističkoj regresiji potrebno je voditi računa o nekoliko stvari:

- Nelinearni problemi ne mogu se riješiti pomoću logističke regresije (pretpostavka samog modela je linearnost).
- Podložna je prenaučenosti (eng. *Overfitting*) ukoliko je velika baza podataka u pitanju. Prenaučenost se događa kada je model jako dobro istreniran te uočava detaljne veze u podatcima do te mjere da kada se isti model primjeni na neviđene podatke, točnost predikcije pada.
- Teško ukazuje na kompleksne veze među podatcima (Rout, 2020.)

3.2.3. Neuronska mreža

U svojoj srži, neuronske mreže su danas najnapredniji oblik strojnog učenja. Kroz repetitivni postupak koji se često naziva dubokim učenjem (eng. *Deep learning*), neuronske mreže osmišljene su i osposobljene za pronalaženje skrivenih obrazaca i temeljnih nelinearnih matematičkih odnosa u velikim skupovima podataka (poput podataka financijskog tržišta).

Ako se uzme u obzir činjenica kako je strojno učenje, a samim time i neuronske mreže, relativno novi pojam, treba se dotaknuti i utjecaja znanstvenika iz drugih područja koji su zaslužni za razvoj strojnog učenja kao koncepta. Jedan od njih je psiholog Donald Hebb, na čijem modelu interakcije stanica djelomično počiva koncept strojnog učenja. Ovaj model Hebb je kreirao 1949. godine u knjizi Organizacija ponašanja, u kojoj autor teoretizira komunikaciju između moždanih stanica. Prema Hebbu:“ Kad jedna stanica uzastopno pomaže u ispaljivanju druge, akson prve stanice razvija sinaptičke gumbe (ili ih povećava ako već postoje) u kontaktu s tijelom druge stanice.“ (Hebb, 1949, str. 63) Treba istaknuti kako se ljudski mozak sastoji od 10^{11} neurona, od koji svaki ima oko 10^{3} sinapsi, obrada informacija se izvršava serijski i paralelno, tolerantan je na pogreške, prima analogne signale te je svjestan tj. inteligentan. Dendriti primaju signale poslane sa drugih neurona, dok akson prenosi impulse koji se preko sinaptičkih terminala šalju na idući neuron (dendrite drugog neurona), a mjesto spajanja se naziva sinapsa (Bošnjak, 2011.) Izgled biološke neuronske stanice prikazan je *Slika 4.*



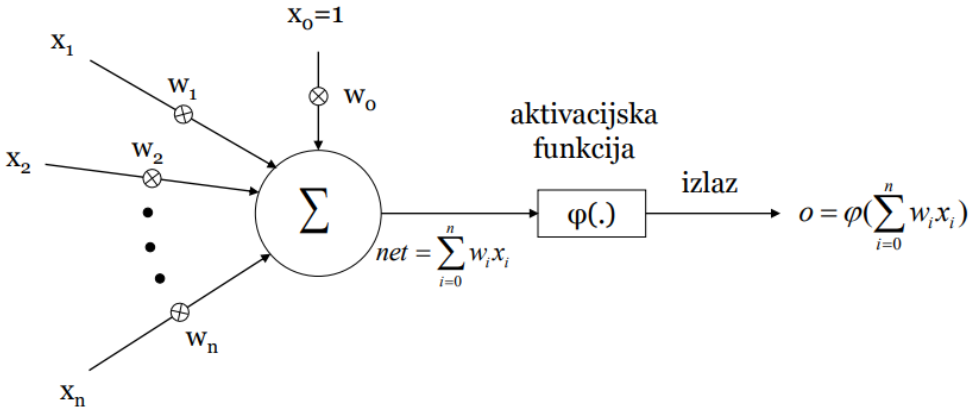
Slika 4 Prirodni neuron
Izvor: https://web.math.pmf.unizg.hr/nastava/su/index.php/download_file/-/view/109/

S obzirom na to kako se metoda neuronskih mreža zasniva na biološkim neuronima tj. moždanim stanicama, povlači se paralela između Hebbove teorije i umjetnih neuronskih mreža gdje takav koncept opisuje zapravo način odnosa između umjetnih neurona.

Prema tome, može se reći kako neuronske mreže imaju korijene u računalnoj neuroznanosti, a poveznica leži u radu Louisa Lapicquea čija je tema bila izgradnja modela interakcije moždanih stanica. Kasnija istraživanja kognitivnih psihologa i ranih računalnih znanstvenika, kao i statističara te matematičara bila su temeljena na ideji mislećih strojeva koji bi mogli rješavati diskretne probleme jednako dobro ili još i bolje nego što to čine ljudi. Ovo rano miješanje neuroznanosti i računalne znanosti je zapravo pokrenulo polje umjetne inteligencije te je zaslužno za ideju i koncept strojnog učenja.

Neuronski model u osnovi povezuje određene čvorove s drugim čvorovima, te u konačnici navedene veze rezultiraju generiranjem zaključaka koji se mogu primijeniti na slične okolnosti. Kada je riječ o prirodnom misaonom procesu, ljudsko promišljanje je protkano subjektivnošću, a sami podatci nisu uredno definirani te posloženi. Osim toga, znanstvenici još uvijek nisu pronašli konkretno objašnjenje nekih obrazaca ponašanja. Te neizvjesnosti su ono što je spriječilo umjetne neuronske mreže da dosegnu svoj konačni i idejni cilj, tj. da se ponašaju poput ljudskog mozga. Međutim, izuzme li se ova činjenica, ideja oko stvaranja stroja koji će razmišljati poput čovjeka, rezultirala je izumom algoritma koji s velikom preciznošću, na temelju kompleksnih odnosa među podatcima, može predvidjeti zadane parametre. (Israel, 2018.)

„Umjetna neuronska mreža koristi kolekciju povezanih čvorova koji se nazivaju umjetni neuroni (moždane stanice) koji su pojednostavljena imitacija bioloških moždanih stanica. Veze između njih su inačice sinapsi i svrha im je prenošenje signala između neurona. Umjetni neuron koji prima signal može ga obraditi, a zatim signalizirati druge umjetne neurone povezane s njim.“ (Foote K. D., 2021.) U tom pravcu razmišljanja, kreiran je model umjetnog neurona, prikazan *Slika 5*..

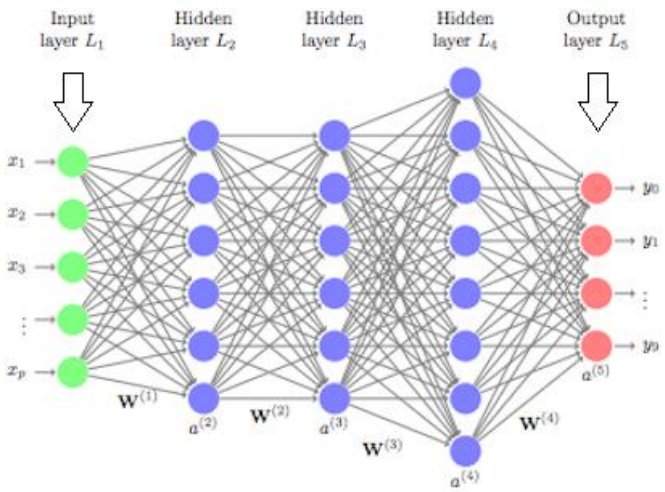


Slika 5 Umjetni neuron
Izvor: https://web.math.pmf.unizg.hr/nastava/su/index.php/download_file/-/view/109/

Analogija prirodnog i umjetnog neurona je sljedeća (Bošnjak, 2011.):

- X_n – signali (numeričke vrijednosti)
- W_n – jakost sinapse (težinski faktor)
- $\sum$ – tijelo neurona (zbrajanje)
- φ – akson (aktivacijska funkcija)

Za transformaciju podataka u neuronskoj mreži, najčešće se koristi izračun temeljen na zbrajanju tzv. težina na način da se vrijednosti čvorova iz skrivenog sloja izračunavaju pomoću ukupnog zbroja vrijednosti ulaznih čvorova pomnoženih za dodijeljenim težinama. Nakon toga, na svaki čvor u skrivenom sloju, primjenjuje se aktivacijska funkcija (ReLu, sigmoidna, linearna, tangens hiperbolna itd.) te se postupak ponavlja, ovaj puta koristeći kao uzlazne podatke, prethodno dobivene vrijednosti iz čvorova skrivenog sloja. (Brownlee, 2021.) Prema toma, neuronske mreže se sastoje od ulaznih podataka, jednog ili više skrivenih slojeva te izlaznih podataka. Potrebno je napomenuti kako se neuronska mreža koja se sastoji od više do jednog skrivenog sloja može smatrati algoritmom dubokog učenja, dok je neuronska mreža s jednim ili niti jednim skrivenim slojem osnovna neuronska mreža. Neuronska mreža koja se sastoji od više od jednog skrivenog sloja, prikazana je *Slika 6.*



Slika 6 Neuronska mreža sa više skrivenih slojeva
Izvor: https://healthitanalytics.com/images/site/features/_normal/deep_nn.png

Neuronske mreže se mogu podijeliti u nekoliko skupina (Joshi, 2017.):

Povratne neuronske mreže (eng. *Feedforward neural network*) – Najjednostavnija je od svih neuronskih mreža, a informacije u njoj se kreću samo u jednom smjeru, tj. pronalazeći skrivene čvorove podatci se kreću od ulaznih ka izlaznima.

Neuronska mreža za funkcijom radijalne osnove (eng. *Radial basis function neural network*) – Riječ je o vrlo intuitivnom tipu neuronske mreže koji se koristi u interpolaciji[2] u višedimenzionalnom prostoru. Svaki „neuron“ pohranjuje primjer instance iz skupa za treniranje kao prototip, a linearnost uključena u funkciju ove neuronske mreže nudi prednost jer ne pati od lokalnih minimuma.

Kohonen samo-organizirajuća neuronska mreža (eng. *Kohonen self-organizing neural network*) – Ova neuronska mreža je idealna za vizualizaciju niskodimenzionalnih[3] prikaza visokodimenzionalnih podataka, te se razlikuje od ostalih vrsta budući da primjenjuje kompetitivno učenje na ulaznim podatcima. Također, ova neuronska mreža je poznata po obavljanju funkcija na neobilježenim podatcima za opisivanje skrivenih struktura.

Rekurentne neuronska mreža (eng. *Recurrent neural network*) – Tip neuronske mreže koji za razliku od povratne, omogućuje dvosmjerni protok podataka te je sposobna koristiti svoju unutarnju memoriju za obradu proizvoljnog slijeda ulaza. Ova neuronska mreža je popularan izbor za zadatke kao što su prepoznavanje rukopisa ili govora.

Modularna neuronska mreža (eng. *Modular neural network*) – Ova neuronska mreža se sastoji od niza neovisnih neuronskih mreža koje moderira posrednik, na način da svaka od neovisnih neuronskih mreža radi s odvojenim ulazima izvršavajući podzadatke koji u konačnici čine cjeloviti zadatak koji modularna mreža mora riješiti. Posrednik, u ovome slučaju, prihvaća ulaze od neovisnih neuronskih mreža, obrađuje ih te u konačnici stvara izlaz za modularnu neuronsku mrežu. Bitno je napomenuti kako neovisne neuronske mreže, međusobno ne „komuniciraju“.

Fizička neuronska mreža (eng. *Physical neural network*) – Cilj ove neuronske mreže je naglasiti oslanjanje na fizički dio računala (hardver), kao razliku od programskog rješenja (softvera) prilikom simulacije neuronske mreže. Električno podesivi materijal, koristi se za oponašanje funkcije neuronske sinapse dok sam algoritam (softverski dio) oponaša cjelokupnu neuronsku mrežu.

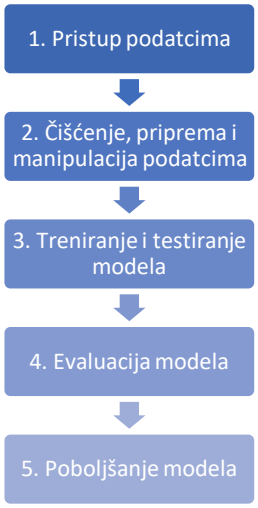
[2] Interpolacija se definira kao postupak kreiranja novih podatkovnih točaka unutar diskretnog skupa poznatih točaka podataka.

[3] Smanjenje dimenzionalnosti je transformacija podataka iz visokodimenzionalnog prostora u niskodimenzionalni prostor tako da niskodimenzionalni prikaz zadržava neka značajna svojstva izvornih podataka, idealno blizu njihovih unutarnja dimenzija. Rad u visokodimenzionalnim prostorima može biti nepoželjan iz mnogih razloga, a analiza podataka je obično računski nerazrješiva.

Neke od prednosti neuronskih mreža kao metode strojnog učenja su: sposobnost paralelne obrade podataka, pohranjivanje podataka na cijeloj mreži, a ne samo u bazi, sposobnost učenja nelinearnih i složenih odnosa, ne postoje ograničenja ulaznih varijabli (primjerice distribucija), sposobnost učenja skrivenih odnosa. Osim toga, neuronske mreže su tzv. crne kutije (eng. *Black boxes*), što znači da nije poznato u kojoj mjeri svaka neovisna varijabla utiče na ovisnu varijablu, tj. varijablu cilja. Korištenje neuronske mreže, računalno je intenzivno te je dugotrajno na standardnim računalnim procesorima. (Burns, 2021.)

3.3. Proces strojnog učenja i metrike klasifikacije

Proces strojnog učenja se prema nekim autorima može podijeliti na više načina, ono što je svima zajedničko je su koraci prikazani na *Slika 7.*



Slika 7 Proces izgradnje modela strojnog učenja
Izvor: autorski rad

Prikupljanje podataka se odnosi na kreiranje same baze podataka nad kojim će se izgraditi model strojnog učenja. Ovaj dio procesa ovisi o prirodi samoga problema i lokaciji te vrsti podataka. Nerijetko su potrebni podatci nestrukturirani, pa ovaj dio procesa traje nešto duže budući da pojedine metode strojnog učenja zahtijevaju strukturirane podatke (tabličnu strukturu). Ovaj korak je vrlo bitan jer će kvaliteta i količina podataka koji se prikupljaju izravno odrediti koliko prediktivni model može biti dobar.

3.3.1. Priprema podataka

Drugi korak u procesu strojnog učenja se odnosi na samu pripremu (eng. *Preprocessing*) podataka, dakle čišćenje nepotrebnih podataka, popunjavanje nedostajućih vrijednosti (imputacija) itd. U ovome koraku može se napraviti i odgovarajuća vizualizacija podataka kako bi se uočili različiti odnosi između pojedinih varijabli koji bi mogli utjecati na kasniju

predikciju. Upoznavajući se sa podatcima, u ovome koraku se izdvaja varijabla cilja, tj. podatci se dijele na ulazne i izlazne te također na podatke za treniranje i testiranje. Ponekad, prikupljeni podatci zahtijevaju daljnju obradu, tj. skaliranje (eng. *scaling*) kao što je standardizacija ili normalizacija podataka. Skaliranje je postupak koji pomaže strojnom učenju kako bi algoritmi brže i učinkovitije radili. (Vashisht, 2021.)

Normalizacija (Min-Max skaliranje) je tehnika skaliranja u kojoj se vrijednosti promatrane varijable za željenu opservaciju (u formuli označeno sa x) pomiču i skaliraju na način da je njihov interval kretanja između 0 i 1, a matematička formulacije je prikazana *Jednadžba 4*

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Jednadžba 4 Normalizacija
Izvor: https://www.atoti.io/when-to-perform-a-feature-scaling/

gdje je $min(x)$ minimum promatrane varijable, a $max(x)$ maksimum, a sam x' je normalizirana x vrijednost.

Standardizacija je druga tehnika skaliranja gdje su vrijednosti centrirane oko aritmetičke sredine podijeljene sa standardnom devijacijom. Ova vrijednost se također zove Z-vrijednost (eng. *Z-score*), a računa se prema formuli prikazanoj *Jednadžba 5*.

$$x' = \frac{x - \bar{x}}{\sigma}$$

Jednadžba 5 Standardizacija
Izvor: https://www.atoti.io/when-to-perform-a-feature-scaling/

gdje je σ standardna devijacija, a $\bar{x}$ aritmetička sredina promatrane varijable. Kao i u prethodnoj tehnici skaliranja podataka, x označava vrijednost varijable za pojedinu opservaciju dok je x' skalirana, odnosno standardizirana x vrijednost. (Vashisht, 2021.) U konačnici se postavlja pitanje, koja se metoda skaliranja podataka bolja, tj. koju metodu izabrati kako bi se postigao maksimalan učinak. U ovome slučaju ne postoji točan odgovor prema kojemu bi se pojedini slučajevi mogli generalizirati te točno odrediti kada koju metodu koristiti. Međutim, postoje neki savjeti kada koja metoda skaliranja bolje funkcionira. Postupak normalizacije se savjetuje kada raspodjela podataka ne slijedi Gaussovu[4] raspodjelu te se može koristiti u algoritmima

[4] Prema Horvat, Mijoč (2019), Gaussova krivulja ili normalna distribucija je teorijska distribucija kontinuiranih podataka, oblika zvona, a jedna od njezinih pretpostavki je je simetričnost prema kojoj će sve tri srednje Mijoč ((aritmetička sredina, medijan i mod) biti jednake te mjere zaobljenosti i simetrije biti jednake nuli. Centralni granični teorem govori kako će se distribucija podataka približavati normalnoj distribuciji sa porastom veličine uzorka bez obzira na oblik izvorne distribucije (populacije).

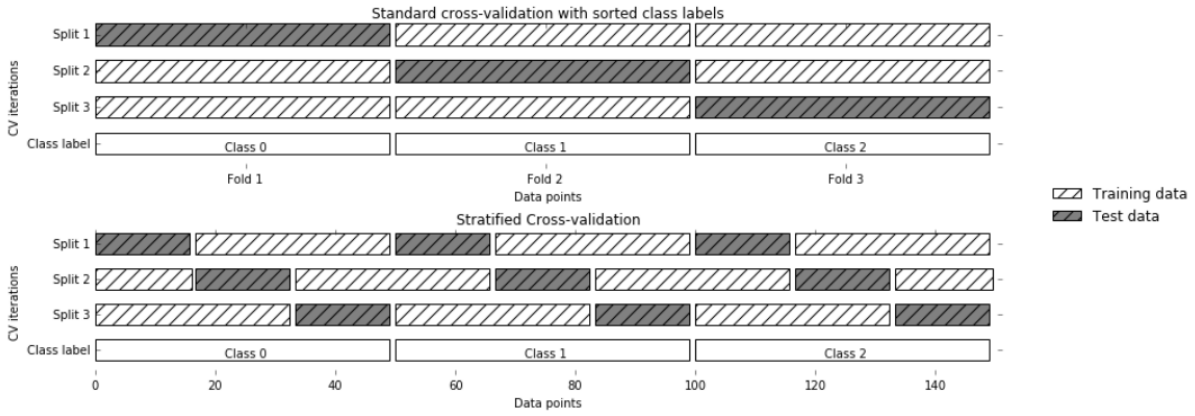
koji se ne baziraju na pretpostavkama distribucije, kao što je KNN algoritam. Može se reći kako bi ovakva vrsta skaliranja odgovarala modelu neuronske mreže, ali i modelima čiji su ulazni podatci pikseli čiji je intenzitet potrebno uklopiti u određeni raspon tj. od 0 do 255 za RGB raspon boja.

Kada je riječ o standardizaciji, ona je korisna u slučajevima kada podatci odgovaraju normalnoj distribuciji, iako ne nužno. Ovdje se treba dotaknuti i učinka stršećih vrijednosti, koji nisu značajni u standardizaciji koliko u normalizaciji. Preporuka je ipak, procijeniti točnost modela uzimajući u obzir obje metode skaliranja, a daljnjim poglavljima bit će prikazan postupak standardizacije nad bazom podataka. (Vashisht, 2021.)

3.3.2. Treniranje podataka

Najjednostavniji način podjele podataka je podjela na dva djela gdje veći dio (obično 75%) čine podatci za treniranje dok manji dio odnosno 25% čine podatci za testiranje. Nešto složeniji model, koji pruža bolji uvid u točnost samog modela te poboljšava performanse je takozvana unakrsna validacija „Unakrsna validacija je statistička metoda evaluacije generalizacijske performanse koja je stabilnija i temeljitija od korištenja podjele na skupove za treniranje i testiranje.“ (Muller & Guido, 2016., str. 254)

U ovoj provjeri, podatci se dijele i treniraju više puta. Najčešće korištena metoda unakrsne validacije je ona koja dijeli podatke u *k* dijelova (eng. *Folds*) gdje *k* određuje sam kreator modela strojnog učenja te on obično iznosi 5 ili 10. Nakon toga, treniranje *k* modela se odvija po sljedećem principu: prvi model koristi prvi dio kao skup podataka za testiranje, dok je ostalih *k-1* dijelova namijenjeno treniranju podataka, te se točnost modela procjenjuje na skupu za trestiranje. Ovakav proces ponovit će se *k* puta, te se za svako testiranje izražava točnost i u konačnici se može izračunati prosječna točnost modela. U slučaju da varijabla cilja ima zadan redoslijed na način da recimo prvih 20% opservacija čini jedna kategorija, odabir unakrsne provjere valjanosti ne bi davao željene rezultate. Umjesto toga, metoda kojom se na neki način mijenja unakrsna validacija je tzv. stratificiranje (eng. *Stratified cross-validation*), tj. metoda u kojoj zastupljenost kategorija u *foldovima* odgovara raspodjeli kategorija u cijeloj bazi. (Muller & Guido, 2016., str. 254-257) Obično je dobra ideja koristiti slojevitu (stratificiranu) unakrsnu provjeru valjanosti za procjenu klasifikatora zato što daje bolje rezultate. Vizualni prikaz ova dva načina unakrsne provjere prikazani su na *Slika 8*.



Slika 8 Unakrsna provjera valjanosti i stratificirana unakrsna provjera valjanosti
Izvor: https://miro.medium.com/max/1796/1*rW9I6BFbzub8-peMnZFFlQ.png

Prema (Muller & Guido, 2016.), nekoliko je prednosti korištenja unakrsne validacije u odnosu na standardnu podjelu na skupove za treniranje i testiranje. Prednost se prvenstveno ogleda u činjenici da su podatci podijeljeni u k-dijelova bolje raspoređeni, točnije u jednakoj mjeri zastupljeni u svim dijelovima. Nadalje, podijelivši neku bazu podataka u primjerice deset dijelova, modeli učenja se primjenjuju na 90% podataka i to deset puta, što ima za posljedicu bolje uočavanje obrazaca i veza među podatcima budući da modeli uče na većem broju podataka. Korištenje ove metode također pruža informacije koliko je model osjetljiv na odabir skupa podataka za treniranje. Isti autori kao glavni nedostatak ove metode navode sporost modela, tj. treniranje *k* modela umjesto jednog usporava cijeli proces *k* puta. Za potrebe ovoga rada bit će korištena metoda desetorostruke slojevite unakrsne provjere valjanosti ($k = 10$).

Treba još napomenuti kako unakrsna validacija optimizira točnost modela na način da se izbjegava prenaučenost i nenaučenost (eng. *Underfitting*). Oba slučaja daju jednako loše rezultate te ih je jako teško izbjeći korištenjem standarde podjele podataka na jedan skup za treniranje i jedan skup za testiranje. Za razliku od prenaučenosti, nenaučenost modela odnosi se na situaciju kada algoritam ne može dobro generalizirati i uočiti veze među podatcima koje su bitne za točnu predikciju. Ovaj problem se rješava povećavanjem kompleksnosti modela, varijabli, instanci ili produljenjem vremena treniranja. (Brownlee, 2019.)

3.3.3. Evaluacije metrike

Što se tiče evaluacije modela, nekoliko je parametara kojim se isti mogu procijeniti, a oni podrazumijevaju:

Točnost (eng. *Accuracy*) je mjera koja govori koliko je puta klasifikator napravio točnu predikciju. U pitanju je, dakle, omjer točno predviđenih kategorija i ukupnih predikcija (broj

podataka u skupu podataka za testiranje), a izražava se formulom koja je prikazana na *Jednadžba 6*.

$$Točnost = \frac{točne\ predikcije}{ukupan\ broj\ predikcija}$$

Jednadžba 6 Točnost modela
Izvor: Izvor: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Matrica konfuzije (eng. *Confusion Matrix*) je nešto kompleksnija metoda prikazivanja (ne)točno klasificiranih opservacija, budući da izračun same točnosti ne pokazuje razlike između klasa. U pitanju je tablični prikaz gdje redci odgovaraju izvornim podatcima, a stupci predviđenima. Bitno je napomenuti kako broj stupaca i redaka u matrici konfuzije ovisi o broju kategorija varijable cilja (ako je u pitanju varijabla sa četiri kategorije kao što je to u ovome radu, dimenzije matrice će biti 4x4). (Zheng, 2015., str. 9-10)

U tom pravcu razmišljanja, javljaju se novi pojmovi kao što su: (Kruger, 2018., str. 71):

- **Istinski pozitivni** (*eng. True positive* - TP) – Broj opservacija koje su ispravno klasificirane kao pozitivne
- **Lažno pozitivni** (*eng. False positive* - FP) – Broj opservacija koje su klasificirane kao pozitivne, međutim prava vrijednost im je negativna
- **Lažno negativni** (*eng. False negative* - FN) – Broj opservacija koje su klasificirane kao negativne, međutim prava vrijednost im je pozitivna
- **Istinski negativni** (*eng. True negative* - TN) Broj opservacija koje su ispravno klasificirane kao negativne

Treba napomenuti kako je tumačenje matrice konfuzije kod višerazredne klasifikacije nešto kompleksnije nego kada je u pitanju binarna klasifikacija. Nadalje, iz ovih vrijednosti moguće je izračunati pokazatelje kao što su preciznost, odaziv i F1 parametar.

Preciznost (eng. *Precision*) je, prema *Jednadžba 7*.,

$$Preciznost = \frac{TP}{TP + FP}$$

Jednadžba 7 Preciznost
Izvor: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

omjer točno klasificiranih i broja svih, takoreći, pozitivno klasificiranih opservacija, dok se odaziv (eng. *Recall*) računa prema formuli koja se prikazuje *Jednadžba 8*

$$Odaziv = \frac{TP}{TP + FN}$$

Jednadžba 8 Odaziv
Izvor: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

iz koje je vidljivo omjer točno klasificiranih pozitivnih opservacija i ukupnog broja stvarnih „pozitivnih“ opservacija. (Shung, 2018.)

F1 parametar je matematički izračun kojim se kombiniraju mjere preciznosti i odaziva putem njihove harmonične sredine i to prema formuli koja je prikazana *Jednadžba 9*.

$$F_1 = 2\frac{preciznost * odaziv}{preciznost + odaziv}$$

Jednadžba 9 F1 parametar
Izvor:https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Logaritamski gubitak (eng. *Log-loss*), koji se poistovjećuje sa unakrsnom entropijom (eng. *Cross-entropy*) je funkcija gubitka koja se koristi u logističkoj regresiji (kako za binarnu, tako i za višerazrednu klasifikaciju), a definirana je kao negativni logaritam vjerojatnosti logističkog modela te se izračunava prema formuli koja je prikazana *Jednadžba 10.*

$$LogLoss = -\frac{1}{N}\sum_{i}^{N} 1y_i \log p_i + (1 - y_i) \log 1 - p_i$$

Jednadžba 10 Logaritamski gubitak
Izvor: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

gdje je p_i vjerojatnost da će, prema klasifikatoru, i-ta instanca pripadati kategoriji 1, dok je y_i prava vrijednost promatrane instance koji iznosi 0 ili 1, ako je u pitanju binarna ili binomna klasifikacija. (Zheng, 2015., str. 9-10)

3.4. Alati za izgradnju modela strojnog učenja

Postoje brojni alati namijenjeni podatkovnoj analitici koji u sebi sadrže mogućnosti izgradnje modela strojnog učenja. Među najpoznatijim analitičkim softverima je dakako Microsoftov Power BI koji se koristi za vizualizaciju podataka i stvaranje dinamičkih panela (eng. *Dashboard*), kao i IBM-ov SPSS, te nadaleko poznati SAS i Statistica. Treba napomenuti kako su ovi alati, ipak općenitije namjene tj. u prvom redu služe podatkovnoj analitici. Osim toga, vrlo poznat alat namijenjen analizi velike količine podataka u vidu vizualizacija je dakako Tableau. Potrebno je još spomenuti i neizostavan dio MS Office paketa, a riječ je dakako o Excelu koji se temelji na proračunskim tablicama i brojnim funkcijama koje omogućavaju uvid u podatke.

Kada govorimo i programskim jezicima koji omogućuju podatkovnu analizu kao i izgradnju modela strojnog učenja, tu se zasigurno izdvajaju R i *Python*. Oba jezika su *Open Source*, što znači da su besplatno dostupni svima. *Python* je interpreterski programski jezik jednostavne sintakse koji pruža napredan analitički proces te je u usporedbi sa drugim programskim jezicima vrlo pristupačan, što ga ujedno i čini vrlo popularnim na području rudarenja podataka i strojnog učenja. (Kappagantula, 2021.) Upravo je programski jezik *Python* korišten za izradu modela strojnog učenja za potrebe ovoga diplomskog rada.

S obzirom na raširenost i upotrebu *Pythona*, (Sarkar, Bali, & Sharma, 2018.) izdvajaju neke prednosti:

Lakoća učenja – koja se ogleda u vrlo jednostavnoj sintaksi

Podrška za višestruke programske paradigme – *Python* je višenamjenski programski jezik koji podržava objektno orijentirano programiranje, jednako kao i strukturirano i funkcionalno, pa čak i aspektno orijentirano programiranje. Ova svestranost omogućuje korištenje *Pythona* za različite namjene od strana različitih programera.

Proširivost (eng. *Exstensibility*) – je jedna od najvažnijih karakteristika *Pythona*. Riječ je o velikom broju lako dostupnih modula koji se mogu koristiti u okviru ovoga programskog jezika. Bitno je napomenuti kako moduli pokrivaju sve aspekte programiranja (od pristupa podatcima do implementacije algoritama) što se ogleda u velikom broj tzv. biblioteka (eng. *Libraries*).

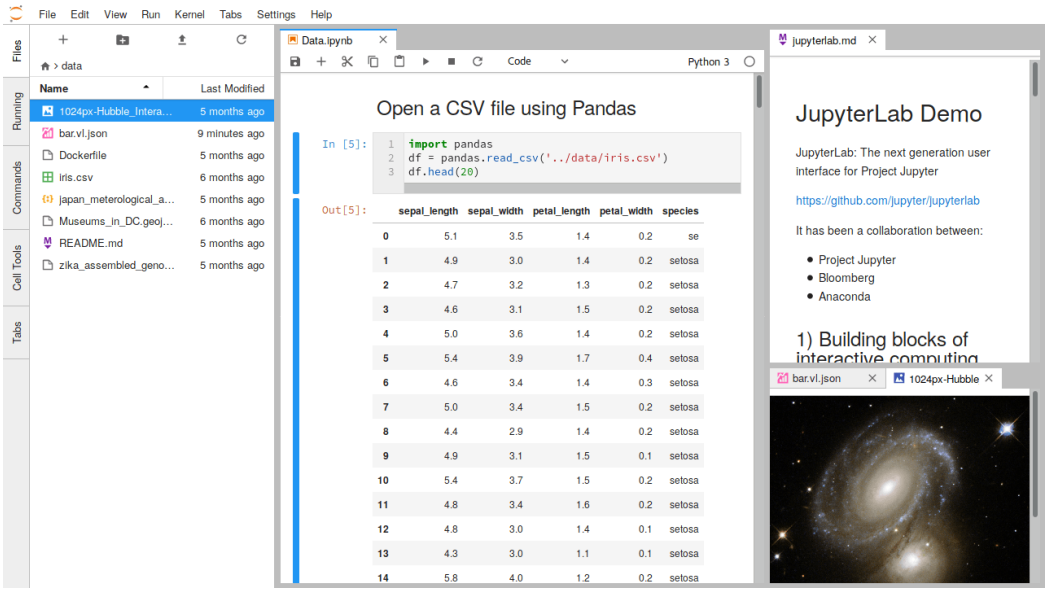
Aktivna zajednica – *Python* podržava velika zajednica programera što ga čini robusnim i prilagodljivim, te je na internetu lako pronaći mnoštvo savjeta vezanih za ovu problematiku.

Prva verzija *Pythona* datira iz 1991. godine, a tvorac jezika je nizozemski matematičar i informatičar Guido van Rossum. U idućim godinama jezik se razvio u široko korišteni programski jezik visoke razine, a bitno je napomenuti kako je njegova prvotna namjena bila objektno orijentirano programiranje. (Sarkar, Bali, & Sharma, 2018.) Trenutno postoje dvije glavne verzije *Pythona* – *Python* 2 i *Python* 3. *Python* 2 nije više aktivan te je zamijenjen novijom inačicom koja sadrži značajne promjene, tj. *Python* 3 koji je i korišten za potrebe ovoga diplomskog rada.

3.4.1. Programsko okruženje

Danas najpopularnije okruženje za korištenje *Pythona* je Jupyter Notebook, kao i JupyterLab. JupyterLab je interaktivno okruženje koje za pisanje i pokretanje koda koristi internetski pretraživač. Treba napomenuti kako osim *Pythona*, ovo okruženje podržava i druge programske

jezike, kao i lagano uključivanje teksta, koda i slika. Okruženje funkcionira po principu *drag and drop* kada je u pitanju preuređivanje ćelija, njihovo premještanje, dodavanje ili brisanje, a blokovi koda se izvode interaktivno iz tekstualnih datoteka (.py, .R, .md, .tex itd). JupyterLab omogućuje uređivanje popularnih formata datoteka kao što su CSV, Vega, JSON, Markdown i drugi. (Project Juypter, 2018.)Vrlo jednostavan izgled koji omogućuje lako i intuitivno snalaženje u okruženju, prikazan je *Slika 9.*



Slika 9 JupyterLab okruženje
Izvor: https://jupyterlab.readthedocs.io/en/stable/_images/interface_jupyterlab.png

3.4.2. Python biblioteke

Središnji pojam vezan za izgradnju modela strojnog učenja u okviru programskog jezika *Python* je dakako ***scikit-learn*** biblioteka, koja sadrži mnoštvo efikasnih funkcija za strojno učenje i statističko modeliranje uključujući klasifikaciju, regresiju klasteriranje itd. Ova biblioteka omogućuje kreiranje prediktivnog modela u svega par linija programskog koda, te naknadnu evaluaciju.

Biblioteka ***NumPy*** je jedna od temeljnih biblioteka kada je u pitanju znanstveno računanje i analitika podataka. Sadrži funkcionalnosti za višedimenzionalne nizove te matematičke funkcije na visokoj razini. U kontekstu *sickit-learna*, *NumPy* je temeljna struktura podataka u smislu da *scikit-learn* uzima podatke u obliku *NumPy* nizova. Treba napomenuti kako ova biblioteka zahtijeva da svi podatci u nizu budu istog tipa. (Muller & Guido, 2016., str. 7-11)

Vrlo bitna je i ***matplotlib*** biblioteka, koja pruža funkcije za izradu vizualizacija kao što su linijski grafikoni, histogrami, dijagrami rasipanja, tortni grafikoni, kutijasti dijagrami i drugi.

Vizualizacije podataka su bitne, budući da pružaju drugačiji uvid u same podatke pridonoseći kvaliteti same analize.

Pandas je biblioteka koja omogućuje podatkovnu manipulaciju i analizu kroz dva temeljna formata. Serije (eng. *Series*) su jednodimenzionalni nizovi podataka, dok podatkovni okviri (eng. *DataFrames*) pružaju tabličnu strukturu podataka slična Excel proračunskoj tablici. *Pandas*, za razliku od prethodno navedenog *NumPy*-a omogućuje različite tipove podataka po stupcima što ga čini pristupačnijim za obradu i analizu podataka. Još jedna od značajki *Pandasa* je sposobnost čitanja različitih datotečnih formata. (Muller & Guido, 2016., str. 7-11)

Za potrebe ovoga diplomskog modela, bit će korištene navedene *Python* biblioteke prema opisanom procesu uz korištenje navedenih metrika evaluacije. Sam postupak, *Python* sintaksa te rezultati istraživanja bit će prikazani u sljedećem poglavlju.

4. Opis istraživanja i rezultati istraživanja

Koristeći metode navedene u prethodnom poglavlju, provedeno je istraživanje baze podataka preuzete sa repozitorija *Kaggle*. Već samo upoznavanje s bazom svjedoči činjenici kako je riječ o višerazrednoj klasifikaciji. U tom pravcu razmišljanja, bit će izgrađena tri modela strojnog učenja dakle, logistička regresija, algoritam k-najbližih susjeda i neuronska mreža. U ovome poglavlju bit će prikazana deskripcija podataka, rezultati modela, te evaluacije metrike.

4.1. Opis podataka

Riječ je o bazi podataka čije su varijable zapravo fizičke karakteristike mobilnih uređaja. Baza se sastoji od 2000 opservacija i 21 varijable, a nazivi i opisi varijabli prikazani su *Tablica 1*.

Tablica 1 Varijable baze podataka
Izvor: https://www.kaggle.com/iabhishekofficial/mobile-price-classification?select=train.csv

Ime varijable	**Opis varijable**
battery_power	Jačina baterije mjerena u mAh
blue	Bluetooth
clock_speed	Brzina kojom mikroprocesor izvršava naredbe
dual_sim	Podrška za dvije kartice
fc	Sporedna kamera - MP
four_g	Podrška za 4G mrežu
int_memory	Interna memorija – GB
m_dep	Debljina uređaja – cm
mobile_wt	Težina uređaja
n_cores	Broj jezgri procesora
pc	Primarna kamera - MP
px_height	Rezolucija zaslona – visina (px)
px_width	Rezolucija zaslona – širina (px)
ram	Radna memorija uređaja (MB)
sc_h	Visina zaslona – cm
sc_w	Širina zaslona – cm
talk_time	Vrijeme trajanja jedne baterije tijekom poziva
three_g	Podrška za 3G
touch_screen	Zaslon osjetljiv na dodir
wifi	Podrška za bežični Internet
price_range	Cjenovni rang

Budući da je baza spremljena u .csv datotečni format, ista je učitana naredbom prikazanom *Slika 10*.

```
price = pd.read_csv("prices.csv" , sep=",")
```

Slika 10 Učitavanje baze podataka
Izvor: autorski rad

.

Nakon učitavanja baze u podatkovni okvir koji nudi *Pandas* biblioteka, moguće je pregledati sadržaj same baze koji je prikazan na *Slika 12*, a postiže se naredbom prikazanom *Slika 11*.

```
price.head()
```

Slika 11 Prikazivanje podataka iz baze
Izvor: autorski rad

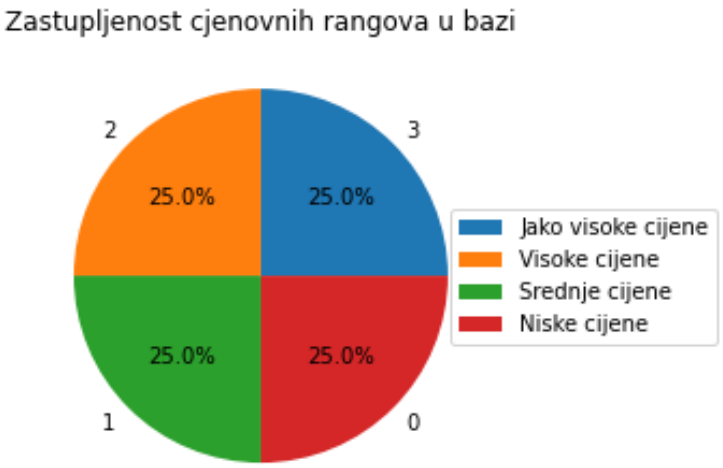
	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411

Slika 12 Djelomični sadržaj baze
Izvor: autorski rad

Baza podataka sadrži 7 kategorijalnih varijabli koje su zbog uštede memorije šifrirane sa 0 i 1, te je varijabla *cjenovni rang* (koja će ujedno i biti varijabla cilja, a sastoji se od četiri kategorije) šifrirana po principu:

- **0** - Uređaji niske cijene
- **1** - Uređaji srednje cijene
- **2** - Uređaji visoke cijene
- **3** - Uređaji jako visoke cijene

Zastupljenost pojedine kategorije u cijeloj bazi podataka prikazana je *Slika 13*. gdje je vidljivo kako su sve četiri kategorije zastupljene u jednakoj mjeri (svakoj kategoriji pripada po 25% opservacija) te se može zaključiti kako je riječ o balansiranoj bazi podataka, koja se već u ovoj fazi istraživanja čini kao dobra podloga za izgradnju prediktivnih modela.



Slika 13 Zastupljenost cjenovnih rangova
Izvor: autorski rad

Osim toga, u okviru biblioteke *Pandas*, moguće je jednom naredbom izvući deskriptivnu statistiku cijele baze ili samo željenih varijabli. S obzirom na činjenicu kako na cijenu uređaja najčešće utiču karakteristike hardverskih komponenti, *Tablica 2* je prikazana deskriptivna statistika (zaokružena na tri decimalna mjesta) upravo onih kontinuiranih varijabli u kojima su sadržani podatci o hardveru. Ovakva tablica rezultat je funkcije *describe()*, a postiže se kodom prikazanim *Slika 14*.

```
price[['battery_power','talk_time', 'clock_speed','n_cores', 'fc','pc','int_memory']].describe()
```

Slika 14 Računanje deskriptivne statistike odabranih kontinuiranih varijabli
Izvor: autorski rad

Tablica 2 Deskriptivna statistika varijabli hardverskih komponenti
Izvor: autorski rad

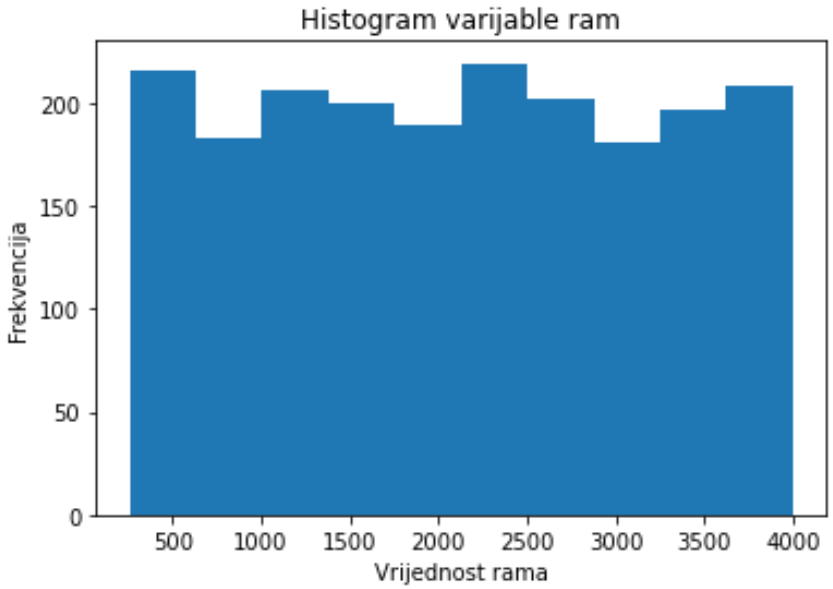
	battery_power	talk_time	clock_speed	n_cores	pc	fc	int_memory	ram
prosjek	1238,519	11,011	1,522	4,520	9,917	4,309	32,047	2124,213
st.dev.	439,418	5,464	0,816	2,288	6,064	4,341	18,146	1084,732
min	501	2	0,5	1	0	0	2	256
25%	851,75	6	0,7	3	5	1	16	1207,5
50%	1226	11	1,5	4	10	3	32	2146,5
75%	1615,25	16	2,2	7	15	7	48	3064,5
max	1998	20	3	8	20	19	64	3998

U većini varijabli, median (redak označen sa „50%“) i prosjek su približno jednaki, na temelju čega se može pretpostaviti normalnost varijabli, primjerice, u varijabli „ram“, prosjek iznosi 2124,213, dok je median 2146,5. U statističkom smislu moguće je postaviti hipoteze o

normalnosti distribucija ovih varijabli, međutim, s obzirom na temu ovoga rada, bit će dovoljno samo prikazati histograme iz kojih približno vidljiva distribucija podataka. Na primjeru varijable „ram“, a naredbom biblioteke *matplotlib, Slika 16* prikazan je histogram navedene varijable te *Slika 15* kreiranje navedenog histograma.

```
price ['ram'].plot.hist()
plt.xlabel('Vrijednost rama')
plt.ylabel('Frekvencija')
plt.title('Histogram varijable ram')
```

Slika 15 Kreiranje histograma
Izvor: autorski rad



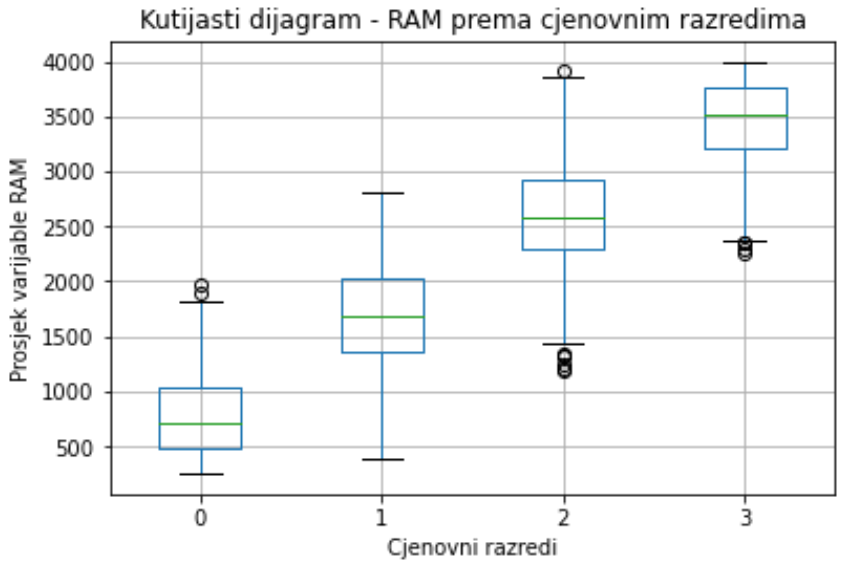
Slika 16 Histogram varijable ram
Izvor: autorski rad

Kako je navedeno, već iz samog histograma je vidljivo kako varijabla „ram“ nije normalno distribuirana te samim time, daljnja statistička analiza glede normalnosti ove varijable ne bi bila niti potrebna.

Ono što je bitno, a tiče se zahtjeva ovoga rada je utjecaj nezavisnih varijabli na zavisnu. U ovoj fazi istraživanja, moguće je grafički prikazati deskriptivnu statistiku neke od kontinuiranih ulaznih varijabli po kategorijama izlazne varijable (cjenovnog ranga) i to tzv. kutijastim dijagramom. Ovakav dijagram izrađen je također uz korištenje naredbi iz *matplotlib* biblioteke koje su prikazane *Slika 17*.

```
price.boxplot(column=['ram'], by=['price_range'])
plt.title('Kutijasti dijagram - RAM prema cjenovnim razredima')
plt.suptitle('')
plt.xlabel('Cjenovni razredi')
plt.ylabel('Prosjek varijable RAM')
```

Slika 17 Kreiranje kutijastog dijagrama
Izvor: autorski rad



Slika 18 Kutijasti dijagram varijabe RAM prema cjenovnim kategorijama
Izvor: autorski rad

Dijagram na *Slika 18* vjerno prikazuje razlike aritmetičkih sredina varijable ram prema kategorija izlazne varijable što doprinosi pretpostavci o razlikama među kategorijama koje doprinose točnosti prediktivnog modela. Osim, toga, vidljivi su svi glavni pokazatelji kada je riječ o deskriptivnoj statistici te stršeće vrijednosti koje se ne prikazuju funkcijom *descirbe()*. Što se tiče utjecaja kategorijalnih varijabli na varijablu cilja, razlike među kategorijama je moguće analizirati naredbom *crosstab()* gdje je jasno vidljivo koliko opservacija iz pojedine kategorije jedna varijable pripada kategorijama druge kategorijalne varijable. U nastavku je prikazan kod na *Slika 19* koji rezultira kontigencijskom tablicom (*Tablica 3* je prilagođenoga formata u odnosu na izvornu tablicu dobivenu naredbom).

```
pd.crosstab(price['price_range'], price['dual_sim'])
```

Slika 19 Kreiranje kontigencijske tablice za odabrane varijable
Izvor: autorski rad

Tablica 3 Kontigencijska tablica varijabli dual_sim i price_range
Izvor: autorski rad

dual_sim	**Ne**	**Da**
price_range		
Niske cijene	250	250
Srednje cijene	254	255
Visoke cijene	251	249
Jako visoke cijene	235	265

Osim što je zastupljenost svih cjenovnih rangova u bazi jednaka, iz *Tablica 3* je vidljivo kako je raspršenost drugih varijabli podjednaka. Osim što se u bazi nalazi približno jednak broj opservacija tj. uređaja koji podržavaju dvije SIM kartice, taj broj je u jednakoj mjeri raspoređen i u cjenovnim kategorijama. Iz *Tablica 3* je vidljivo kako raspršenost cjenovnih kategorija za uređaje koji podržavaju dvije SIM kartice od niskih do jako visokih cijena iznosi 250, 255, 249 i 265 opservacija.

S obzirom na činjenicu kako su za izgradnju modela strojnog učenja potrebni uglavnom potpuni podatci, potrebno je provjeriti postoje li nedostajuće vrijednosti, što se u kodu provodi naredbom prikazanom na *Slika 20.*

```
price.isnull().sum()
```

Slika 20 Provjera nedostajućih vrijednosti
Izvor: autorski rad

Rezultat ove naredbe je popis svih varijabli te broj njihovih nedostajućih vrijednosti što je u ovome slučaju nula. Nadalje, prije same izgradnje modela, potrebno je bazu podijeliti na ulazne varijable te jednu izlaznu, a osim toga baza se dijeli na skup podataka za treniranje i skup za testiranje. Prema tome, 20 je ulaznih varijabli i jedna izlazna, a sve opservacije bit će uključene u izgradnju modela budući da veći broj istih doprinosi točnosti modela. Slijedi podjela baze na ulazne i izlazne podatke te standardizacija ulaznih podataka (pred-procesuiranje), a prikazani su *Slika 21*.

```
X = price.iloc[:,:-1]
y = price.iloc[:,-1]
print(X.shape)
print(y.shape)

from sklearn import preprocessing
scaler=preprocessing.StandardScaler().fit(X)
X_scaled=scaler.transform(X)
```

Slika 21 Podjela podataka na ulazne i izlazne i pred-procesuiranje podataka
Izvor: autorski rad

4.2. Izgradnja i rezultati modela

Radi postizanja veće točnosti modela i izbjegavanja kako nenaučenosti, tako i prenaučenosti, umjesto standardne podjele podataka na skup za treniranje i skup za testiranje, korištena je slojevita (stratificirana) unakrsna validacija (u nastavku teksta CV), koja je objašnjena u metodologiji. Korištena je desetorostruka CV i to na sva tri modela, tako da je svaki *fold* sadržavao 200 opservacija. Stratificiranje dijelova se postiže kodom prikazanim na *Slika 22*.

```
from sklearn.model_selection import KFold
kfold = KFold(n_splits=10)
```

Slika 22 Stratificiranje *foldova*
Izvor: autorski rad

Osim toga, potrebno je i učitati potrebne klasifikatore iz *scikit-learn* biblioteke, i to na način prikazan *Slika 23*.

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
```

Slika 23 Učitavanje klasifikatora
Izvor: autorski rad

Nakon toga, u svega par linija koda, moguće je kreirati prediktivni model, definirajući u kodu kako je riječ o višerazrednoj klasifikaciji, te pozivajući ulazne i izlazne varijable i naravno provodeći prethodno definiranu desetorostruku CV. Najprije je na *Slika 24* prikazan dio koda koji se odnosi na kreiranje prediktivnog modela metodom logističke regresije, a koji rezultira točnošću po *foldovima*, te prosječnom točnošću.

```
from sklearn.model_selection import cross_val_score
logreg = LogisticRegression(multi_class='multinomial', solver='lbfgs')
scores = cross_val_score(logreg, X_scaled, y, cv=kfold)
print("Cross-validation scores: {}".format(scores))
avgsc_lr=format(scores.mean())
print("Prosječni score ", avgsc_lr)
```

Slika 24 Kreiranje modela strojnog učenja metodom logističke regresije
Izvor: autorski rad

Na sličan način je kreiran i model koji se temelji na metodi k-najbližih susjeda, jedino što je prethodno izvršena procjena najboljeg tj. optimalnog *k* (broja susjeda koji se uzimaju u obzir), što je prikazano *Slika 25*.

```
from sklearn.model_selection import GridSearchCV
#kreiranje novog modela
knn = KNeighborsClassifier()
#kreiranje rječnika svih vrijednosti koje se testiraju kao potencijalan broj susjeda
param_grid = {"n_neighbors": np.arange(1, 25)}
#testiranje svih vrijednosti
knn_gscv = GridSearchCV(knn, param_grid, cv=10)
#fit
knn_gscv.fit(X_scaled, y)
#optimalan broj susjeda
knn_gscv.best_params_
```

Slika 25 Procjena optimalnog broja susjeda za KNN metodu
Izvor: autorski rad

Koristeći skalirane podatke, vrijednost optimalnog *k* iznosila je 22 najbliža susjeda, dok je na ne skaliranim podatcima taj broj bio nešto manji, odnosno 12 najbližih susjeda. Prilikom evaluacije podataka bit će prikazana razlika između korištenja skaliranih i ne skaliranih podataka, no za sada će biti naglasak na skaliranim podatcima, budući da su se isti koristili u sva tri modela. Slijedi izgradnja KNN modela (prikazana na *Slika 26*) temeljenog na 22 najbliža susjeda koji rezultira procjenama točnosti modela.

```
knn = KNeighborsClassifier(n_neighbors = 22)
scores_n = cross_val_score(knn, X_scaled, y, cv=kfold)
print("Cross-validation scores: {}".format(scores_n))
avgsc_n=format(scores_n.mean())
print("Prosječni score ", avgsc_n)
```

Slika 26 Kreiranje modela strojnog učenja KNN metodom
Izvor: autorski rad

Nadalje, po istom principu izgrađena je neuronska mreža sa jednim skrivenim slojem od 5 elemenata, te je za ovu priliku korištena ReLu aktivacijska funkcija. Osim toga, zbog kompleksnosti neuronske mreže i veličine baze podataka, povećan je i maksimalan broj iteracija, a sve to je prikazano *Slika 27*.

```
nn = MLPClassifier(hidden_layer_sizes = [5],random_state = 0,
                   solver='lbfgs', max_iter=100000000000)
scores_nn = cross_val_score(nn, X_scaled, y, cv=kfold)
print("Cross-validation scores: {}".format(scores_nn))
avgsc_nn=format(scores_nn.mean())
print("Prosječni score ", avgsc_nn)
```

Slika 27 Kreiranje modela strojnog učenja metodom neuronske mreže
Izvor: autorski rad

Rezultati sva tri modela, odnosno točnosti po *foldovima*, kao i prosječne točnosti prikazane su *Tablica 4*.

Tablica 4 Točnosti modela prema metodama
Izvor: autorski rad

	Logistička regresija	**KNN - skalirano**	**KNN – ne skalirano**	**Neuronska mreža**
fold 1	0,955	0,635	0,92	0,96
fold 2	0,98	0,63	0,96	0,98
fold 3	0,955	0,66	0,94	0,965
fold 4	0,95	0,615	0,945	0,965
fold 5	0,97	0,595	0,93	0,965
fold 6	0,96	0,62	0,95	0,96
fold 7	0,96	0,68	0,94	0,975
fold 8	0,96	0,57	0,93	0,97
fold 9	0,97	0,565	0,91	0,97
fold 10	0,96	0,64	0,915	0,975
$\bar{x}$	**0,962**	**0,621**	**0,934**	**0,968**

U *Tablica 4* je vidljivo kako je CV metoda predviđanja na svim metodama rezultirala visokom točnošću osim kod KNN metode gdje su korišteni skalirani podatci. Za pretpostavit je da bi model eventualno imao bolju performansu kada bi umjesto standardizacije, podatci bili normalizirani. Međutim, ono što je očekivano s obzirom na linearnost modela je visoka točnost logističke regresije, kao i neuronske mreže koja je na koncu ipak imala najveću prosječnu točnost. Razlika točnosti među *foldovima* je neznatna, tako da se može reći kako su modeli dobro istrenirani te sa visokom točnošću uspijevaju kategorizirati neviđene podatke.

Ono što treba naglasiti je kompleksnost neuronske mreže. Naime, kako je riječ o linearnom modelu, povećavanjem kompleksnosti neuronske mreže, došlo je do smanjivanja točnosti modela, odnosno mreža je bila prenaučena u smislu da je uočavala obrasce među podatcima koji su bili svojstveni samo promatranom skupu te na taj način mreža ne može dobro

generalizirati preostale podatke koji se nalaze u skupu za testiranje. U konačnici, model sa jednim skrivenim slojem od pet čvorova je davao najbolji rezultat, koji je ujedno i prikazan u radu. Sam proces predviđanja, koji je prikazan *Slika 28*, odvija se po istom principu kod sva tri modela, pa će za potrebe ovoga rada biti prikazan samo metoda predviđanja logističke regresije, budući da se sam kod razlikuje jedino po imenu varijable.

```
from sklearn.model_selection import cross_val_predict
prediction_lr = cross_val_predict (logreg, X_scaled,y)
print ("Predviđene vrijednosti", prediction_lr)
```

Slika 28 Proces predviđanja logističke regresije unakrsnom validacijom
Izvor: autorski rad

Predviđene vrijednosti također neće biti prikazane u radu, nego će služiti za izračune evaluacijske metrike koje će zbog boljeg uvida u performanse modela biti prikazane u sljedećem potpoglavlju.

4.3. Evaluacija modela

Ono što je srž svake evaluacije klasifikacijskog modela je matrica konfuzije, gdje je točno vidljivo koju kategoriju je model točno ili netočno predvidio određen broj puta. Kako je u metodologiji objašnjeno, riječ je o tablici koja se sastoji od stvarnih i predviđenih vrijednosti, a *Tablica 5* je prikazana prilagođena matrica konfuzije prethodno kreirane logističke regresije, dok je naredba za dobivanje ovakve tablice prikazana *Slika 29*.

```
from sklearn.metrics import confusion_matrix
confusion_lr = confusion_matrix(y, prediction_lr)
print("Matrica konfuzije:\n{}".format(confusion_lr))
```

Slika 29 Kreiranje matrice konfuzije za podatke predviđene metodom logističke regresije
Izvor: autorski rad

Tablica 5 Matrica konfuzije logističke regresije
Izvor: autorski rad

	Predviđene vrijednosti				
Stvarne vrijednosti		0	1	2	3
	0	489	11	0	0
	1	14	474	12	0
	2	0	15	473	12
	3	0	0	11	489

Na dijagonali matice uvijek su brojevi točno predviđenih vrijednosti iz pojedine kategorije, što je i istaknuto u ovome slučaju. Vidljivo je kako je za svaku kategoriju to vrlo visok broj te je za kategorije niskih i vrlo visokih cijena taj broj nešto veći od ostale dvije kategorije i iznosi 489. To su dakle istinski pozitivne vrijednosti kojih u ovome slučaju imaju 4, tj. za svaku kategoriju se računaju TP, TN, FP i FN vrijednosti. Izračun preciznosti i odaziva, kao i prosječne točnosti po kategorijama može se dobiti naredbom prikazanom na *Slika 30,* koja rezultira prilagođenom *Tablica 6*.

```
from sklearn.metrics import classification_report
print(classification_report(y, prediction_lr))
confusion_lr.diagonal()/confusion_lr.sum(axis=1)
```

Slika 30 Kreiranje klasifikacijskog izvještaja za podatke predviđene metodom logističke regresije
Izvor: autorski rad

Tablica 6 Klasifikacijski izvještaj logističke regresije
Izvor: autorski rad

	Preciznost	**Odaziv**	**F1**	**Točnost**
0	0,97	0,98	0,98	0,978
1	0,95	0,95	0,95	0,948
2	0,95	0,95	0,95	0,946
3	0,98	0,98	0,98	0,978
Točnost			0,96	
Makro-prosjek	0,96	0,96	0,96	
Težinski prosjek	0,96	0,96	0,96	

Tablica 6 je prikazan tzv. klasifikacijski izvještaj logističke regresije uz dodanu točnost po kategoriji. Najprije je vidljivo kako su svi pokazatelji vrlo visoki što je dobro, te referirajući se na matricu konfuzije, kategorije niskih i jako visokih cijena su predviđene sa najvećom točnošću.

F1 metrika se najčešće koristi za neuravnotežene skupove podataka međutim, na dobar način može evaluirati i ovakav balansiran skup podataka. Ideja koja stoji iza višerazredne F vrijednosti je izračunati binarnu F vrijednost po kategoriji smatrajući promatranu kategoriju pozitivnom a ostale negativnima. Makro prosjek (eng. *Macro-averaged*) izračunava ne ponderirane prosjeke F vrijednosti, kao i mjere preciznosti i odaziva, što daje jednaku težinu svim klasama bez obzira na njihovu veličinu, dok težinski prosjek (eng. *Weighted-averaged*) uzima u obzir veličinu kategorije. (Khalusova, 2019.) U ovome slučaju makro prosjek preciznosti se izračunava po formuli prikazanoj *Jednadžba 11*

$$MakroProsjek = \frac{PR_0 + PR_1 + PR_2 + PR_3}{4}$$

Jednadžba 11 Makro-Prosjek preciznosti
Izvor: https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2

gdje PR označava preciznost. Težinski prosjek se računa prema formuli prikazanoj *Jednadžba 12*

$$Težinski\ prosjek = \frac{(PR_0 * N_0) + (PR_1 * N_1) + (PR_2 * N_2) + (PR_3 * N_3)}{Ukupan\ broj\ opservacija}$$

Jednadžba 12 Težinski prosjek preciznosti
Izvor: https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2

gdje je PR također preciznost pojedine kategorije, a N je broj pripadajućih opservacija. (Khalusova, 2019.)

S obzirom da je veličina kategorija u promatranom slučaju jednaka, težinski i makro prosjeci po kategoriji su također jednaki što je i vidljivo u *Tablica 6.* Nadalje, na isti način je evaluiran model temeljen na metodi k-najbližih susjeda, međutim ovaj puta s manjom točnošću, a sam kod je prikazan *Slika 31.*

```
from sklearn.metrics import confusion_matrix
confusion_n = confusion_matrix(y, prediction_knn)
print("Matrica konfuzije:\n{}".format(confusion_n))
from sklearn.metrics import classification_report
print(classification_report(y, prediction_knn))
confusion_n.diagonal()/confusion_n.sum(axis=1)
```

Slika 31 Kreiranje matrice konfuzije i klasifikacijsko izvještaja predviđenih vrijednosti KNN metodom
Izvor: autorski rad

Tablica 7 Matrica konfuzije KNN algoritma
Izvor: autorski rad

	Predviđene vrijednosti				
Stvarne vrijednosti		0	1	2	3
	0	388	98	14	0
	1	124	252	114	10
	2	18	151	249	82
	3	0	24	136	340

Metoda temeljena na algoritmu k-najbližih susjeda rezultirala je manjom točnošću tako da je je iz matrice konfuzije prikazane *Tablica 7* moguće vidjeti kako model nije griješio jedino pri

kategorizaciji niskih cijena kao visokih i obrnuto. U svim drugim slučajevima prisutne su greške.

Tablica 8 Klasifikacijski izvještaj KNN algoritma
Izvor:autorski rad

	Preciznost	**Odaziv**	**F1**	**Točnost**
0	0,73	0,78	0,75	0,776
1	0,48	0,50	0,49	0,504
2	0,49	0,50	0,49	0,498
3	0,79	0,68	0,73	0,68
Točnost			0,61	
Makro-prosjek	0,62	0,61	0,62	
Težinski prosjek	0,62	0,61	0,62	

Iz *Tablica 8* vidljivo je kako je točnost, očekivano, najveća kod kategorizacije niskih cijena, a najmanja kod kategorizacije visokih (kategorija 2). Također, preciznost, kao omjer točno klasificiranih i svih opservacija klasificiranih kao promatrana kategorija, je puno manja kod kategorije 1 i 2. Ista stvar je i kada je u pitanju mjera odaziva kao mjera točno klasificiranih i ukupnog broja stvarnih kategorija. Ostaje još procijeniti neuronsku mrežu kao model koji sa najvećom točnošću kategorizira neviđene podatke. Dakle, prosječna točnost modela temeljenog na neuronskoj mreži, iznosila je 0,968, a u nastavku je *Tablica 9* prikazana matrica konfuzije kao i klasifikacijski izvještaj (*Tablica 10)*. *Slika 32* prikazana je naredba za kreiranje dvije prethodno navedene tablice.

```
from sklearn.metrics import confusion_matrix
confusion_nn = confusion_matrix(y, prediction_nn)
print("Matrica konfuzije:\n{}".format(confusion_nn))
from sklearn.metrics import classification_report
print(classification_report(y, prediction_nn))
confusion_nn.diagonal()/confusion_nn.sum(axis=1)
```

Slika 32 Kreiranje matrice konfuzije i klasifikacijskog izvještaja za podatke predviđene metodom neuronske mreže
Izvor: autorski rad

Tablica 9 Matrica konfuzije neuronske mreže
Izvor: autorski rad

	Predviđene vrijednosti				
Stvarne vrijednosti		0	1	2	3
	0	492	8	0	0
	1	8	478	14	0

	2	0	7	478	15
	3	0	0	10	490

Iz *Tablica 9* može se vidjeti kako je najveći broj točno klasificiranih opservacija imala kategorija 0 koja se odnosi na niske cijene proizvoda, što i nije začuđujuće s obzirom na činjenicu kako se proizvodi nižeg cjenovnog ranga u praksi znatno razlikuju po svojim fizičkim karakteristikama od ostalih. Međutim, potrebno je naglasiti kako ni u ovome slučaju ni došlo do netočnih klasifikacija u vidu velikog odstupanja od kategorija u smislu da ni jedna opservacija s oznakom nižeg cjenovnog ranga nije klasificirana kao visoka odnosno vrlo visoka cijena i obrnuto.

Tablica 10 Klasifikacijski izvještaj neuronske mreže
Izvor: autorski rad

	Preciznost	**Odaziv**	**F1**	**Točnost**
0	0,98	0,98	0,98	0,984
1	0,97	0,96	0,96	0,956
2	0,95	0,96	0,95	0,956
3	0,97	0,98	0,98	0,98
Točnost			0,97	
Makro-prosjek	0,97	0,97	0,97	
Težinski prosjek	0,97	0,97	0,97	

Klasifikacijski izvještaj nije potrebno previše komentirati budući da sama točnost modela te točnost prema kategorijama dovoljno govori. Ono što je glavni nedostatak neuronske mreže kao metode je teška interpretabilnost parametara u smislu utjecaja ulaznih varijabli na izlaznu.

4.3.1. Koeficijenti modela

Govoreći o utjecaju ulaznih varijabli na izlaznu, regresijske analize pružaju detaljniji uvid u podatke. Naime, budući da se logistička regresija temelji na koeficijentima, u programerskom pogledu, na vrlo jednostavan način se može doći do tzv. koeficijenata modela, koji za svaku kategoriju govore o utjecaju promjene promatranih varijabli na samu predikciju.

Osim toga, druge regresijske analize govore o značajnost pojedinih varijabli glede same predikcije. Neovisno o činjenici kako je u ovome slučaju riječ o klasifikacijskom problemu treba imati na umu kako je logistička regresija ipak transformirana linearna regresija.

U tom pravcu razmišljanja „*Regresor Ridge*“ ima varijantu klasifikatora: „*RidgeClassifier“*. Ovaj klasifikator prvo pretvara binarne vrijednosti u varijabli cilja u niz {-1, 1}, a zatim problem

tretira kao regresijski zadatak, optimizirajući model. Predviđena klasa odgovara predznaku regresora, a višerazrednu klasifikaciju, problem se tretira kao tzv. *multi-output* regresija, te predviđena klasa odgovara rezultatu s najvećom vrijednošću.“ (ScikitLearn, n.d.)

Radi tumačenja podataka, bitno je definirati regularizaciju kao metodu izbjegavanja prekomjernih prilagođavanja tzv. kažnjavanjem prekomjerno visokih regresijskih koeficijenata. Regularizacija L1 dodaje penale jednake apsolutnoj vrijednosti veličine koeficijenata. Drugim riječima, ograničava se veličina koeficijenata. Lasso regresija koristi ovu metodu, a može se primijeniti i na logističku regresiju i to na način prikazan *Slika 33*.

```
logreg1 = LogisticRegression(C = 0.1, class_weight= 'balanced', penalty= 'l1',
                             solver= 'liblinear')
```

Slika 33 Kreiranje L1 regularizacije
Izvor: autorski rad

Regulacija L2 dodaje penale jednake kvadratu veličine koeficijenata, a *ElasticNet* kombinira L1 i L2 metode (prilikom izgradnje modela logističke regresije koristi se *ElasticNet Penalty* parametar). U tom pravcu razmišljanja, definiranje logističke regresije se provodi postupkom prikazanim na *Slika 34*.

```
logreg1 = LogisticRegression(multi_class='multinomial', penalty='elasticnet'
                             , l1_ratio=0.5, solver='saga')
```

Slika 34 ElasticNet regularizacija
Izvor: autorski rad

Usporedbe radi, u *Tablica 11* bit će prikazani koeficijenti koji se odnose na kategoriju jako visokih cijena, za tri prethodno navedene regresijske analize.

Tablica 11 Koeficijenti logističke regresije
Izvor: autorski rad

Ime varijable	**Logistička regresija**	**Lasso**	**Ridge**	**ElasticNet**
battery_power	2,994	1,477	$1{,}627*10^{-1}$	3,641
blue	$-2{,}705*10^{-2}$	0	$7{,}439*10^{-3}$	$-4{,}353*10^{-2}$
clock_speed	$-1{,}909*10^{-2}$	0	$-1{,}929*10^{-3}$	0
dual_sim	$-3{,}049*10^{-3}$	0	$1{,}597*10^{-3}$	0
fc	$2{,}750*10^{-2}$	0	$-1{,}6*10^{-2}$	$1{,}570*10^{-2}$
four_g	$8{,}096*10^{-2}$	2,789*10-^{3}	$2{,}877*10^{-2}$	$7{,}517*10^{-2}$
int_memory	$2{,}656*10^{-1}$	$1{,}093*10^{-1}$	$3{,}045*10^{-2}$	$3{,}220*10^{-1}$
m_dep	$-6{,}259*10^{-2}$	0	$-2{,}217*10^{-3}$	$-4{,}815*10^{-2}$

mobile_wt	-5,641 * 10 $^{-1}$	-3,291*10^{-1}	-5,128*10^{-2}	-6,707*10^{-1}
n_cores	1,156 * 10 $^{-1}$	0	-4,337*10^{-3}	1,277*10^{-1}
pc	3,515 * 10 $^{-2}$	0	9,9345*10^{-3}	3,005*10^{-2}
px_height	1,729	8,192*10^{-1}	6,906*10^{-2}	2,123
px_width	1,761	8,766*10^{-1}	1,003-10^{-1}	2,129
ram	1,178*10 $^{-1}$	5,906	6,106*10^{-1}	1,437*10 ^{1}
sc_h	1,176 *10 $^{-1}$	1,129*10^{-1}	2,979*10^{-2}	2,445*10^{-1}
sc_w	3,062 *10^{-2}	0	-3,964*10^{-4}	0
talk_time	6,908 *10^{-2}	0	-6,091*10^{-3}	6,004*10^{-2}
three_g	-3,429 *10^{-3}	0	-1,653*10^{-2}	0
touch_screen	-1,734 * 10^{-2}	0	1,247*10^{-2}	-3,639*10^{-3}
wifi	*1,419 * 10 $^{-1}$	0	-2,607*10^{-3}	-1,5*10^{-1}

Koeficijenti modela se mogu tumačiti prema formuli: Uz sve ostalo nepromijenjeno, logaritam šansi da je mobilni uređaj jako visoke cijene se u slučaju jediničnog povećanja varijable 1 povećava za 2.944. Drugim riječima, koeficijent logističke regresije iznosi 2,994, odnosno šansa da će neki uređaj pripadati višem cjenovnom rangu se povećava ukoliko se povećava prva navedena varijabla što je u ovome slučaju trajanje baterije. Vidljivo je ovakvo tumačenje koeficijenata sasvim logično s obzirom na prirodu određivanja cijena visokotehnoloških uređaja. Nadalje, u preostalim metodama izračuna koeficijenata, vidljiva je razlika koja leži u regularizaciji parametara. Osim toga, vidljivo je kako su koeficijenti nekih varijabli dosta mali, što je dakako posljedica same prirode varijable, budući da baza sadrži i kategorijalne varijable. Također, neke od metoda su rezultirale koeficijentima jednakim nuli što bi značilo da jedinično povećanje tih varijabli ne utiče na šansu pripadnosti pojedinoj kategoriji. Drugim riječima, Lasso pristup može se tretirati kao tzv. *Feature selection* koji ima za cilj smanjivanje dimenzionalnosti skupova podataka što može rezultirati poboljšanjem točnosti ili performanse ako je riječ o visokodimenzionalnim podatcima. Ovdje je, dakle za razliku od neuronske mreže, vidljivo kako pojedina varijabla može utjecati na izlaznu varijablu i samu predikciju. Osim toga, različitim metodama regularizacije parametara može se utjecati kako na točnost modela tako i na druge evaluacijske metrike.

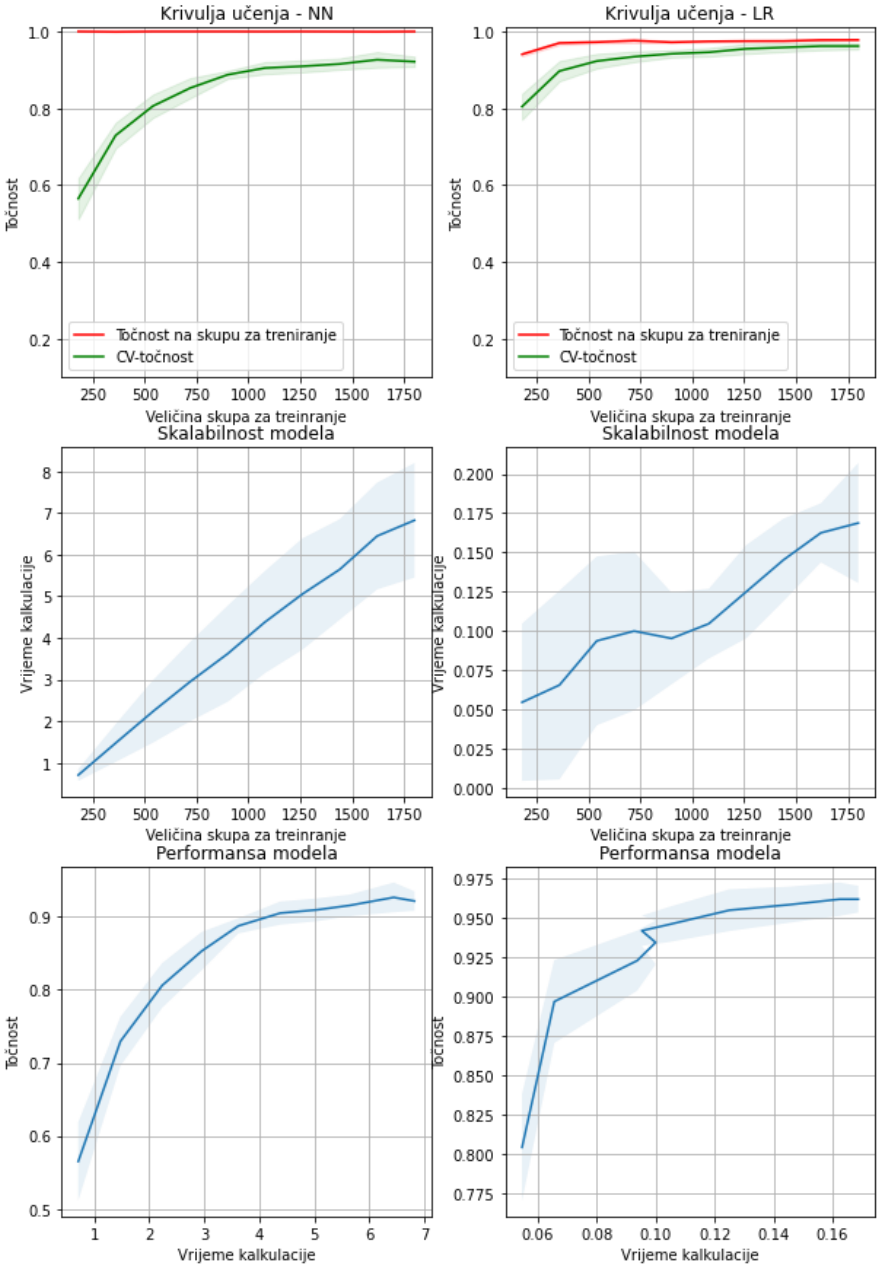
4.3.2. Krivulja učenja

Za potrebe evaluacije modela, izrađena je i tzv. krivulja učenja (eng. *Learning Curve*) kojom se dobiva bolji uvid u performanse modela. Usporedno je prikazana performansa neuronske mreže i logističke regresije koristeći modul *scikit-learna* koji je rezultirao višestrukim krivuljama prikazanima *Slika 36,* a primjena *Python* modula je vidljiva na *Slika 35*.

```
title1 = "Krivulja učenja - NN"
cv1 = kfold
estimator1 = MLPClassifier()
plot_learning_curve(estimator1, title1, X_scaled, y, axes=axes[:, 0], ylim=(0.1, 1.01),
                    cv=cv1, n_jobs=4)

title2 = r"Krivulja učenja - LR"
cv2 = kfold
estimator2 = LogisticRegression()
plot_learning_curve(estimator2, title2, X_scaled, y, axes=axes[:, 1], ylim=(0.1, 1.01),
                    cv=cv2, n_jobs=4)
```

Slika 35 Kreiranje krivulje učenja za metode neuronske mreže i logističke regresije
Izvor: autorski rad

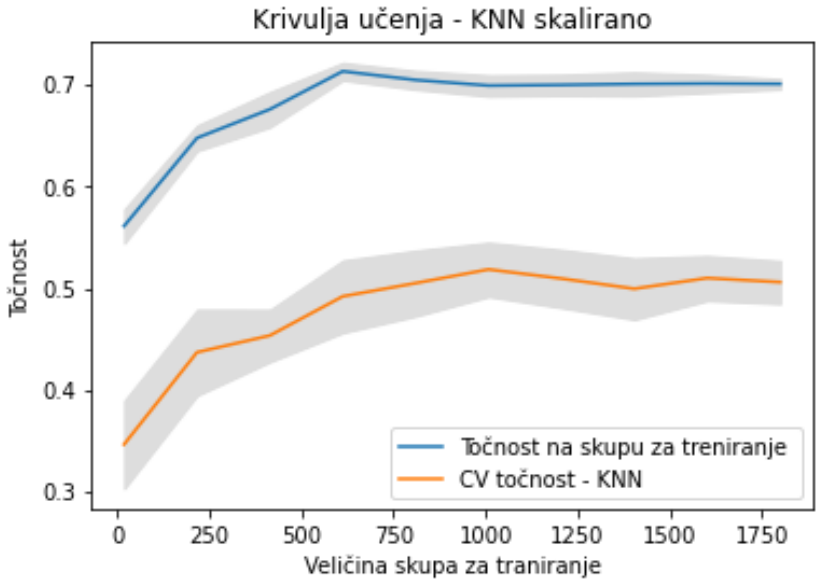


Slika 36 Krivulje učenja
Izvor: autorski rad

Krivulja učenja neuronske mreže pokazuje vrlo visoku točnost na svim veličinama skupova za treniranje. Vidljivo je kako povećanjem istih tih skupova CV točnost postupno raste te daleko manja šansa za prenaučenost modela. Ono što je sasvim logično, a vidljivo je na druge dvije krivulje koje se tiču neuronske mreže, je rast točnosti s vremenom tzv. kalkulacije (eng. *fit times*) te isto tako rast samog vremena kalkulacije sa porastom veličine skupova za treniranje, naravno zbog veličine podataka koje je potrebno obraditi. U ovom trenutku može se vidjeti priroda neuronske mreže u kontekstu tzv. *Eager learnera*.

Kada je logistička regresija u pitanju, situacija je jako slična međutim dinamika kretanja krivulja je nešto drugačija. Vidljivo je kako je manja veličina skupa za treniranje rezultirala nešto manjom točnošću modela te da je standardna devijacija, odnosno odstupanje u omjeru veličine skupa i vremena kalkulacije nešto veća, a sama krivulja manje strma u odnosu na neuronsku mrežu. Odnos vremena kalkulacije i same točnosti je jako sličan kao i kod neuronske mreže. tj. raste po opadajućoj stopi.

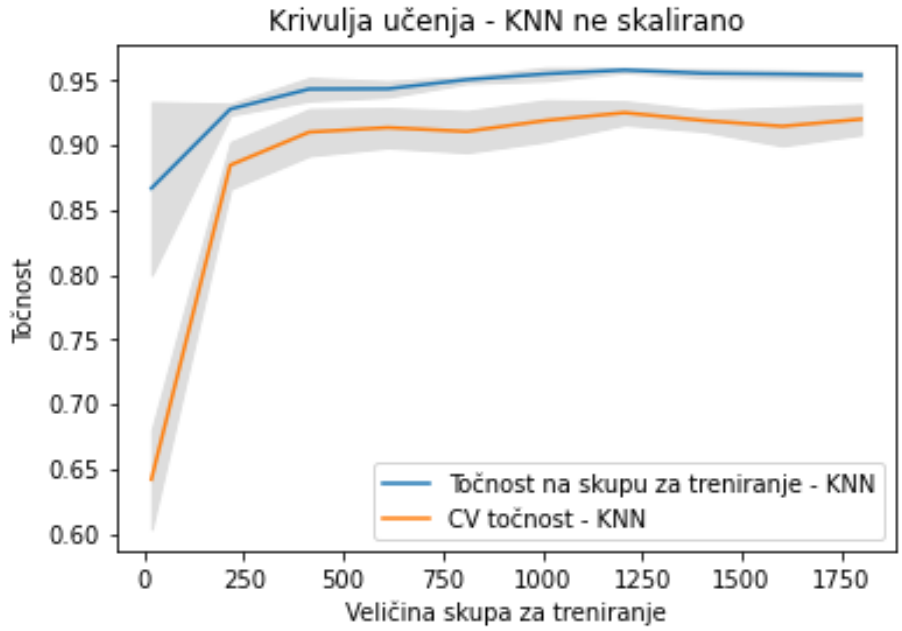
Nadalje, radi usporedbe performansi metoda k-najbližih susjeda na skaliranim i ne skaliranim podatcima, krivulja učenja ove dvije metode, prikazane su u nastavku.



Slika 37 Krivulja učenja KNN algoritma sa skaliranim podatcima
Izvor: autorski rad

Na *Slika 37* vidljivo je kako je točnost na skupu za treniranje najveća kada je veličina skupa iznosi oko 600 opservacija, međutim, točnost provedena CV metodom je najveća kada veličina skupa za treniranje iznosi oko 100 opservacija, te kasnije pada. Međutim, te točnosti su i dalje niske ukoliko se promatra mogućnost implementacije ovakvog modela. Ukoliko se model

temeljen na algoritmu k-najbližih susjeda želi poboljšati, potrebno je bolje istražiti veze među podatcima i međusobne utjecaje. *Slika 38* je prikazan model temeljen na istoj metodologiji, međutim riječ je o ne skaliranim podatcima gdje je performansa znatno bolja, tj. točnosti modela prelaze 0,9 sa povećanjem veličine skupa za treniranje.



Slika 38 Krivulja učenja KNN algoritma na ne skaliranim podatcima
Izvor: autorski rad

5. Rasprava

Na primjeru tri metode izrađeni su modeli strojnog učenja koji se bave klasifikacijom cjenovnog ranga mobilnih uređaja. Podatci su pred-procesuirani metodom standardizacije, te je zbog izbjegavanja prenaučenosti modela korištena unakrsna validacija modela. Neuronska mreža je, kako je i očekivano, pokazala najbolje performanse, međutim treba biti oprezan kada je kompleksnost modela u pitanju. Budući da se problem dovoljno točno može opisati linearnim modelom, model temeljen na neuronskoj mreži, sklon je prenaučenosti te samim time točnost modela opada. Dubljom analizom u ovu metodu zaključeno je kako je točnost vrlo visoka u sve četiri kategorije te kako ovako relativno jednostavan model neuronske mreže može s velikom točnošću kategorizirati cjenovne rangove s obzirom na fizičke karakteristike uređaja.

Osim toga, logistička regresija daje jako dobre rezultate, neznatno manje od neuronske mreže te je ova metoda vrlo korisna ukoliko se želi istražiti utjecaj pojedinih ulaznih varijabli na izlaznu varijablu, odnosno na šansu pripadanja opservacija pojedinoj kategoriji. Treba napomenuti kako se točnosti ovoga modela mijenjaju u odnosu na metodu regularizacije.

Ono što svakako treba prokomentirati je niska performansa modela temeljenog na metodi k-najbližih susjeda u odnosu na prethodne dvije metode. U pitanju je vrlo jednostavan model koji često nije u stanju uočiti sve obrasce i povezanosti među podatcima. Treba također napomenuti kako je KNN algoritam davao daleko bolje rezultate kada su za njegovu provedbu korišteni ne skalirani podatci. Ono što bi svakako bilo poželjno napraviti je skaliranje podataka metodom normalizacije te usporedba parametara za sve tri metode. Normalizacija podataka možda bi pozitivno utjecala na podložnost neuronske mreže prekomjernoj naučenosti u složenijim modelima. Također, postoji mnoštvo metoda namijenjenih višerazrednoj klasifikaciji te bi radi usporedbe bilo dobro izgraditi ovakve modele.

Kada je u pitanju implementacija ovakvog modela u praksi, može se reći kako je moguća, te bi očit izbor bio model neuronske mreže. Međutim, uzimajući u obzir kompleksnost podataka kao i poslovne okoline što ima za posljedicu kompleksnost donošenja poslovne odluke, ovakav klasifikacijski model bi bio samo dio nekog većeg sustava umjetne inteligencije koji bi u sebi sadržavao podatke o kupcima, transakcijama te općenito poslovnoj okolini kao što to sadrže modeli strojnog učenja namijenjeni dinamičnom određivanju cijena. Osim toga, budući da je u pitanju tehnološki dinamična industrija, koja svakim danom sve više napreduje i poboljšavana svoje proizvode, ovakvi podatci su podložni zastarijevanju te ih je potrebno redovito ažurirati.

6. Zaključak

Neosporno je kako umjetna inteligencija već odavno nije stvar budućnosti nego je neizostavni dio svakog tehnološki naprednog poslovnog sustava bilo da je u pitanju klasično poslovanje ili poslovanje čiji je glavni prodajni, ali i komunikacijski kanal Internet. Vođeni mišlju o kreiranju stroja koji će razmišljati poput ljudi, znanstvenici 20. stoljeća su postavili temelje za razvitak danas sve više primijenjene grane podatkovne znanosti – strojnog učenja.

Sam razvoj informacijsko komunikacijskih tehnologija je pridonio razvoju ovoga područja te se prosječni korisnik interneta gotovo svakodnevno susreće sa nekim oblikom strojnog učenja, a da toga možda nije ni svjestan. Oblasti primjene kao i metode se razlikuju u ovisnosti o vrsti podataka te problemu koji je potrebno riješiti. Sami podatci poprimaju različite oblike i formate koje je matematičkom metodologijom i digitalnom obradom potrebno pročitati i analizirati kako bi isti podatci bili od koristi poslovnom subjektu. Ručna obrada velike količine podataka je dakako nemoguća te iznimno spora, čak i kada je u pitanju korištenje računala kao alata za računanje. U tu svrhu razvijeni su i zapravo se svakodnevno razvijaju, modeli strojnog učenja koji u stvarnom vremenu mogu obraditi velike količine podataka, u čijoj podlozi su računalni algoritmi temeljeni na matematičkim zakonitostima.

Dotičući se teorijske podloge strojnog učenja te vrsta i problema koje iste rješavaju, objašnjen je problem klasifikacije te na praktičnom primjeru prikazana izgradnja modela namijenjenih višerazrednoj klasifikaciji. Osim toga, teoretiziran je sam proces izgradnje modela strojnog učenja, te objašnjen svaki korak počevši s upoznavanjem podataka, preko konkretne izgradnje modela pa do evaluacije i usporedbe. Na koncu je donesen zaključak kako je ipak najbolju performansu davao model neuronske mreže, no neznatno lošiji bio je model logističke regresije, te je donesena odluka o implementaciji modela.

LITERATURA

- Asim, M., & Zafar, K. (2018.). Mobile price class prediction using machine learning techniques. *International Journal of Computer Applications, 29*, str. 6-11. Preuzeto 16.. lipanj 2021. iz https://www.researchgate.net/profile/Muhammad_Asim41/publication/323994340_Mobile_Price_Class_prediction_using_Machine_Learning_Techniques/links/5b2b23b94585150c63446830/Mobile-Price-Class-prediction-using-Machine-Learning-Techniques.pdf
- Bansal, S. (2021.). *What is Classification Algorithm in Machine Learning? With Examples.* Preuzeto 17. lipanj 2021. iz Analytixlabs: https://www.analytixlabs.co.in/blog/classification-in-machine-learning/
- Bošnjak, M. (2011.). *Neurosnske mreže*. Preuzeto 25. lipanj 2021. iz PMF: https://web.math.pmf.unizg.hr/nastava/su/index.php/download_file/-/view/109/
- Brownlee, J. (2019.). *Overfitting and Underfitting With Machine Learning Algorithms*. Preuzeto 12. lipanj 2021. iz Machine Learning Mastery: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/
- Brownlee, J. (2021.). *How to Choose an Activation Function for Deep Learning*. Preuzeto 19. lipanj 2021. iz Machine Learning Mastery: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/
- Burns, E. (20. lipanj 2021.). *What is a neural network? Explanation and examples*. Dohvaćeno iz TechTarget: https://searchenterpriseai.techtarget.com/definition/neural-network
- Chandrashekhara, K. T., Thungamani, M., Babu, C. G., & Manjunath, T. N. (2019.). Smartphone price prediction in retail industry using machine learning techniques., (str. 363-373). Singapore. Preuzeto 16. lipanj 2021. iz https://link.springer.com/chapter/10.1007/978-981-13-5802-9_34#citeas
- Chatterjee, M. (2020.). *Data Science vs Machine Learning and Artificial Intelligence*. Preuzeto 17. lipanj 2021. iz GreatLearning: https://www.mygreatlearning.com/blog/difference-data-science-machine-learning-ai/#machinelearning
- Cunningham, P., & Cord, M. (2008.). *Machine learning techniques for multimedia: case studies on organization and retrieval.* Berlin, Springer. Preuzeto 13. lipanj 2021. iz https://www.google.com/books?hl=hr&lr=&id=uSvYmki2yg0C&oi=fnd&pg=PA2&dq=Machine+learning+techniques+for+multimedia&ots=136A3h3hWx&sig=aCj56GjoivavI-orHAni96UB4aI
- DiGangi, E. A., & Moore, M. K. (2012.). Research methods in human skeletal biology. Preuzeto 18. lipanj 2021. iz https://www.google.com/books?hl=hr&lr=&id=2S7oiT-YV7AC&oi=fnd&pg=PP1&dq=Ancestry+Estimation.+Research+Methods+in+Human+Skeletal+Biology,&ots=oauM2YyXF6&sig=gBDBCoPfxeYX0p79WC4SrNLDOV4

- El Naqua, I., & Murphy, M. J. (2015.). What is machine learning? U I. El Naqua, M. J. Murphy, & R. Li, *Machine learning in radiation oncology* (str. 3-11). Springer. Preuzeto 9. lipanj 2021. iz https://books.google.ba/books?hl=hr&lr=&id=1N7yCQAAQBAJ&oi=fnd&pg=PR5&dq=What+Is+Machine+Learning%3F+Machine+Learning+in+Radiation+Oncology&ots=n-3ofh5mH3&sig=uIlb2QGi-AOzqsifQnYonaCSTB0&redir_esc=y#v=onepage&q=What%20Is%20Machine%20Learning%3F%20Machine%
- Foote, K. (2019). *A Brief History of Machine Learning*. Preuzeto 10. lipanj 2021. iz Dataversity: https://www.dataversity.net/a-brief-history-of-machine-learning/
- Foote, K. D. (20. lipanj 2021.). *Artificial Neural Networks: An Overview*. Dohvaćeno iz Dataversity: https://www.dataversity.net/artificial-neural-networks-overview/
- Galvan, I., Valls, J. M., Garcia, M., & Isasi, P. (2011.). A lazy learning approach for building classification models. *nternational journal of intelligent systems, 26*(8), str. 773-786. Preuzeto 5. lipanj 2021. iz https://onlinelibrary.wiley.com/doi/abs/10.1002/int.20493
- Good, I. J. (1980). Some history of the hierarchical Bayesian methodology. *Trabajos de estadística y de investigación operativa, 31*(1), str. 489-519. Preuzeto 15. lipanj 2021. iz https://link.springer.com/article/10.1007/BF02888365
- Hebb, D. O. (1949). *The Organization.* New York: JOHN WILEY & SONS. Preuzeto 3. lipanj 2021. iz http://s-f-walker.org.uk/pubsebooks/pdfs/The_Organization_of_Behavior-Donald_O._Hebb.pdf
- Horvat, J., & Mijoč, J. (2019.). *Istaživački SPaSS.* Zagreb: Ljevak.
- Israel, S. (2018.). *What Are Neural Networks?* Preuzeto 20. lipanj 2021. iz Benzinga: https://www.benzinga.com/fintech/18/02/11245602/what-are-neural-networks
- Jose, I. (2018.). *KNN (K-Nearest Neighbors) #1*. Preuzeto 18. lipanj 2021. iz Towardsdatascience: https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d
- Joshi, N. (2017.). *Six types of neural networks*. Preuzeto 14. lipanj 2021. iz Allerin: https://www.allerin.com/blog/six-types-of-neural-networks
- Kanal, L. (2003). Perceptron. *Encyclopedia of Computer Science*, str. 1383-1385. Preuzeto 15. lipanj 2021. iz https://dl.acm.org/doi/abs/10.5555/1074100.1074686
- Kappagantula, S. (2021.). *Top 10 Data Analytics Tools You Need To Know In 2021*. Preuzeto 22. lipanj 2021. iz edureka!: https://www.edureka.co/blog/top-10-data-analytics-tools
- Khalusova, M. (2019.). *MACHINE LEARNING MODEL EVALUATION METRICS PART 2: MULTI-CLASS CLASSIFICATION*. Preuzeto 22. lipanj 2021. iz MariaKhalusova: https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2
- Khan, J. (2021.). *What Is Dynamic Pricing, and How Does It Affect E-commerce?* Preuzeto 14. lipanj 2021. iz Business.com: https://www.business.com/articles/what-is-dynamic-pricing-and-how-does-it-affect-ecommerce/

- Kraljević, S., & Ognjen, S. (2020.). Primjena algoritama dubinske analize podataka i strojnog učenja za klasifikaciju i predikciju u društvenom području. *Polytechnic and design, 8*(1), str. 38-46. Preuzeto 15. lipanj 2021. iz https://hrcak.srce.hr/242766
- Kruger, F. (2018.). Activity, context, and plan recognition with computational causal behavior models. Universiät Rostock. Fakultät für Informatik und Elektrotechnik. Preuzeto 19. lipanj 2021. iz http://rosdok.uni-rostock.de/resolve/urn/urn:nbn:de:gbv:28-diss2018-0009-0
- Kucharavy, D., & DeGuio, R. (2011.). Application of S-shaped curves., (str. 559-572). Strasbourg. Preuzeto 25. lipanj 2021. iz https://pdf.sciencedirectassets.com/278653/1-s2.0-S1877705811X00048/1-s2.0-S1877705811001597/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEGsaCXVzLWVhc3QtMSJIMEYCIQC6tJG85g8nxh38rsZIyN%2B4QQWsu2aldcddYxitgjfXagIhALQ1gtcifemh1GHSGIzGQi8pAohmvsODN%2FJZyrpg
- Maillo, J., Ramirez, S., Triguero, I., & Herrera, F. (2017.). kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. *Knowledge-Based Systems*, str. 3-15. Preuzeto 18. lipanj 2021. iz https://www.sciencedirect.com/science/article/pii/S0950705116301757
- McCarthy, J., & Feigenbaum, E. A. (1990). In memoriam: Arthur samuel: Pioneer in machine learning. *AI Magazine, 11*(3), str. 10-10. Preuzeto 15. lipanj 2021. iz https://ojs.aaai.org/index.php/aimagazine/article/view/840
- Monroe, W. (2017.). *Logistic Regression*. Preuzeto 25. lipanj 2021. iz web.stanford.edu: https://web.stanford.edu/class/archive/cs/cs109/cs109.1178/lectureHandouts/220-logistic-regression.pdf
- Muller, A. C., & Guido, S. (2016.). *Introduction to machine learning with Python: a guide for data scientists.* O'Reilly Media, Inc.
- Narahari, Y., Raju, C., Ravikumar, K., & Shah, S. (2005.). Dynamic pricing models for electronic business. *Sadhana, 30*(2), str. 231-256. Preuzeto 14. lipanj 2021. iz https://link.springer.com/article/10.1007%2FBF02706246
- Phan, T. D. (2018.). Housing price prediction using machine learning algorithms: The case of Melbourne city, Australia. (str. 39). Sydney: IEEE. Preuzeto 15. lipanj 2021. iz https://ieeexplore.ieee.org/abstract/document/8614000/authors#authors
- Project Juypter. (2018.). *JupyterLab is Ready for Users*. Preuzeto 23. lipanj 2021. iz Jupyter: https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906
- Rout, A. R. (2020.). *Advantages and Disadvantages of Logistic Regression*. Preuzeto 6. lipanj 2021. iz GeeksForGeeks: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/
- Sarkar, D., Bali, R., & Sharma, T. (2018.). *Practical machine learning with Python.* Preuzeto 12. lipanj. 2021. iz https://link.springer.com/content/pdf/10.1007/978-1-4842-3207-1.pdf
- Schachter, B. (2018). *7 Ways Your Business Should Be Using Machine Learning Today*. Preuzeto 10. lipanj 2021. iz Readwrite: https://readwrite.com/2018/04/06/7-ways-your-business-should-be-using-machine-learning-today/

- ScikitLearn. (n.d.). *RidgeClassifier*. Preuzeto 14. lipanj 2021. iz Scikit-Learn: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html?highlight=ridge%20classifier#sklearn.linear_model.RidgeClassifier
- Shung, K. P. (2018.). *Accuracy, Precision, Recall or F1?* Preuzeto 13. lipanj 2021. iz Towardsdatascience: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
- Starzacher, A., & Bernard, R. (2008.). Evaluating KNN, LDA and QDA classification for embedded online feature fusion. Sydney: IEEE. Preuzeto 26. lipanj 2021. iz https://ieeexplore.ieee.org/abstract/document/4761967
- Vashisht, R. (2021.). *When to perform a Feature Scaling?* Preuzeto 19. lipanj 2021. iz Atoti: https://www.atoti.io/when-to-perform-a-feature-scaling/
- Yin, Y., Jiajun, Z., & Zhu, Q. (2012.). Cell Phones Price Forecast Based on Adaptive Sliding Window. (str. 247-250). Guilin: IEEE. Preuzeto 16. lipanj 2021. iz https://ieeexplore.ieee.org/abstract/document/6385282
- Zekić-Sušac, M., Sanja, P., & Šarlija, N. (2014.). A comparison of machine learning methods in a high-dimensional classification problem. *Business Systems Research Journal, 5*(3), str. 82-96. Preuzeto 13.. lipanj 2021. iz https://hrcak.srce.hr/126919
- Zheng, A. (2015.). *Evaluating machine learning models: a beginner's guide to key concepts and pitfalls.* O'Reilly Media, Inc.

POPIS SLIKA

POPIS TABLICA

POPIS JEDNADŽBI