

Planiranje potražnje uz pomoć strojnog učenja

Matijević, Marija

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:119:335677>

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

Download date / Datum preuzimanja: **2024-04-18**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences -
Institutional Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Marija Matijević

**PROGNOZA I PLANIRANJE POTRAŽNJE UZ POMOĆ STROJNOG
UČENJA**

DIPLOMSKI RAD

Zagreb, rujan 2020.

Zagreb, 4. veljače 2020.

Zavod: **Zavod za transportnu logistiku**
Predmet: **Distribucijska logistika II**

DIPLOMSKI ZADATAK br. 5496

Pristupnik: **Marija Matijević (0135233771)**
Studij: **Inteligentni transportni sustavi i logistika**
Smjer: **Logistika**

Zadatak: **Planiranje potražnje uz pomoć strojnog učenja**

Opis zadatka:

Korištenje strojnog učenja omogućuje razvitak sofisticiranijih modela prognoze potražnje u odnosu na postojeće. U radu je potrebno istražiti mogućnosti primjene sustava strojnog učenja u segmentu prognoze potražnje te mogućnosti primjene u praksi.

Mentor:

Predsjednik povjerenstva za
diplomski ispit:

prof. dr. sc. Kristijan Rogić

Sveučilište u Zagrebu

Fakultet prometnih znanosti

DIPLOMSKI RAD

PROGNOZA I PLANIRANJE POTRAŽNJE UZ POMOĆ STROJNOG UČENJA

IMPROVING DEMAND FORECASTING WITH MACHINE LEARNING

Mentor : prof. dr. sc. Kristijan Rogić

Student: Marija Matijević

JMBAG: 0135233771

Zagreb, rujan 2020.

SAŽETAK: Visoka učinkovitost opskrbnog lanca može se postići primjenom odgovarajućih metoda za dugoročno predviđanje prognoze. Klasične metode prognoze jednostavne su u svome izračunu, ali njihova točnost zna biti na nezavidnoj razini. Stoga su metode strojnog učenja dobra prilika za poboljšanje ovoga polja. Ovaj rad donosi usporedbu prognoze potražnje klasičnom metodom sezonalnog autoregresivnog integriranog pomičnog prosjeka s vanjskim varijablama te metodama strojnog učenja, metoda slučajne šume i metoda umjetne neuronske mreže.

KLJUČNE RIJEČI: prognoza potražnje; strojno učenje; sezonalni autoregresivni integrirani pomični prosjek s vanjskim varijablama; slučajna šuma; umjetna neuronska mreža

SUMMARY: High efficiency of the supply chain can be attained by applying the best-performing methods for long term forecasting. Traditional forecasting methods are easy to implement and compute but their accuracy can be lackluster. Therefore, methods that implement machine learning are a great way to improve this field. This paper compares forecast results using three methods, one of them is a classic method of the autoregressive integrated moving average with external variables, the other two methods are machine learning methods, random forest regression, and artificial neural networks.

KEYWORDS: demand forecasting; machine learning; seasonal autoregressive integrated moving average with external variables; random forest regressor; artificial neural network

SADRŽAJ

1. UVOD.....	1
2. PROGNOZA POTRAŽNJE.....	3
2.1. Kratkoročno predviđanje potražnje.....	4
2.2. Srednjoročno predviđanje potražnje.....	6
2.3. Dugoročno predviđanje potražnje.....	6
3. KLASIČNE METODE PROGNOZE POTRAŽNJE.....	8
3.1. Kvalitativne metode prognoze potražnje.....	8
3.1.1. Delphi metoda.....	8
3.1.2. Grass roots metoda.....	9
3.1.3. Ekspertne skupine.....	9
3.1.4. Istraživanje tržišta.....	9
3.2. Metode analize vremenskih serija.....	9
3.2.1. Metoda pomičnih prosjeka.....	12
3.2.2. Metoda ponderiranih pomičnih prosjeka.....	14
3.2.3. Metoda eksponencijalnog izgladivanja.....	15
3.2.4. Autoregresivni integrirani pomični prosjek.....	16
3.2.5. Metode zasnovane na analizi trenda.....	17
3.2.6. Metode analize sezonskih podataka.....	18
3.3. Uzročne metode.....	19

3.4.	Simulacijske metode	20
4.	STROJNO UČENJE	22
4.1.	Nadzirano učenje	24
4.2.	Polu-nadzirano učenje	25
4.3.	Nenadzirano učenje	25
4.4.	Podržano učenje	25
5.	METODE PROGNOZE POTRAŽNJE STROJNOG UČENJA	27
5.1.	Umjetne neuronske mreže	27
5.2.	Šuma slučajnih odluka	30
6.	STUDIJA SLUČAJA	31
6.1.	Opis i obrada skupa podataka	31
6.2.	Primjena metoda	36
6.2.1.	Primjena metode autoregresivnog integriranog pomičnog prosjeka	36
6.2.2.	Primjena metode neuronske mreže	39
6.2.3.	Primjena metode slučajne šume	41
6.3.	Prikaz rezultata	41
7.	Zaključak	50
	Literatura	51
	Popis slika	54
	Popis tablica	55

Popis grafikona.....	55
----------------------	----

1. UVOD

Prognoza potražnje predstavlja ključni faktor u optimizaciji učinkovitog opskrbnog lanca i njegovih subjekata. Važnost prognoze potražnje najviše se očituje u subjektima dobavljača repromaterijala, distributera i maloprodaje. Kako bi se postigla učinkovita distribucija, ali i odredile učinkovite strategije maloprodaje bitno je prema ciljevima poduzeća odabrati prikladne metode prognoze potražnje.

Ukoliko postoji određeni skup podataka za predviđanje većih količina, od nekoliko godina, moguće je izvršiti kratkoročno do srednjoročno predviđanje uz pomoć metoda vremenskih serija. Kratkoročno predviđanje pogodno je pri prognozi potražnje u industriji i maloprodaji. U ovakvom predviđanju iznimno je bitna točnost prognoze jer o njoj ovisi dostupnost proizvoda koje vodi do postizanja veće konkurentnosti na tržištu. Srednjoročno predviđanje postaje dostupnije kako računalna snaga postaje jeftinija pa sofisticiranije metode prognoziranja dolaze do izražaja. Uz pad cijene računalnih komponenti raste i njihova dostupnost pa time i veća mogućnost implementacija metoda strojnog učenja.

Metode strojnog učenja razlikuju se od klasičnih metoda u nezavisnim varijablama, odnosno varijablama koje se koriste za predviđanje vrijednosti. Metode strojnog učenja imaju mogućnost primiti više nezavisnih varijabli kako bi predvidjele zavisnu varijablu. Ova karakteristika pruža mogućnost uspostavljanja korelacijskih veza između varijabli. Kompleksnost ovih metoda opravdana je zbog visoke točnosti što potvrđuju provedena istraživanja.

Autori Réal Carbonneau, Rustam Vahidov i Kevin Laframboise u svome radu navode kako upotreba strojnog učenja daje bolje rezultate u prognozi od klasičnih metoda poput prognoze zasnovanih na analizi trenda i metode pomičnog prosjeka.[1] Cilj ovoga rada je potvrditi postojeća istraživanja i dokazati kako implementacija metoda strojnog učenja zaista daje bolje rezultate.

Ovaj rad se sastoji od 7 poglavlja. Nakon uvodnog poglavlja teorijski se obrađuje tematika prognoze potražnje kroz prva dva poglavlja. Drugo poglavlje opisuje sam pojam prognoze potražnje i njenu podjelu na kratkoročno, srednjoročno i dugoročno predviđanje. Treće poglavlje prikazuje podjelu klasičnih metoda prognoze potražnje uz opis svake od navedenih

metoda. Sljedeća dva poglavlja, 4. i 5. poglavlje obrađuju tematiku strojnog učenja i metoda prognoze strojnog učenja korištenih u studiji slučaja. Poglavlje broj 6 opisuje studiju slučaja ovoga rada. U ovome poglavlju detaljnije je prikazan korišteni skup podataka, opisana je implementacija korištenih metoda te su prikazani rezultati studije. Zadnje poglavlje donosi analizu studije slučaja i zaključak cijeloga rada.

2. PROGNOZA POTRAŽNJE

Upravljanje potražnjom skup je aktivnosti od presudnog značenja za upravljanje opskrbnim lancem. Bez procjene budućnosti nije moguće planirati razinu aktivnosti koja se može očekivati, a samim time nije moguće procijeniti resurse koji su potrebni za planiranje i kontrolu razina aktivnosti. Dobro osmišljena primjena i izvršavanje procesa mogu pridonijeti značajnu korist poduzećima poput, na primjer, smanjenja razine zaliha te dostupnosti proizvoda. Nije dovoljno samo dobro prognozirati i imati dobar sustav planiranja izvršavanja operacija. Upravljanje potražnjom treba uključivati pronalaženje načina za smanjenje promjene potražnje i povećanje fleksibilnosti u radu, te primjenu dobrog sustava upravljanja u kriznim situacijama kako bi poduzeće bilo u mogućnosti brzo reagirati na neplanirane situacije.[2]

Zajednički cilj dobrih sustava prognoze potražnje je imati najmanju količinu zaliha koja zadovoljava zahtjeve kupaca, a istovremeno minimizirati troškove nabave i držanja zaliha. Poduzeća s prevelikom razinom zaliha imaju prekomjerne troškove uzrokovane skladištenjem i zastarijevanjem proizvoda. U suprotnom, ukoliko poduzeće nema dovoljan broj proizvoda na zalihama, rezultat će niskom razinom zadovoljstva kupca zbog nepronalaska željenog proizvoda. Pouzdane prognoze su ključne kako bi poduzeće moglo opstati i rasti.

U proizvodnji menadžment mora predvidjeti buduće zahtjeve za svoje proizvode kako bi osigurali materijal, radnu snagu i kapacitet za ispunjavanje ovih potreba. Resursi se planiraju mnogo prije no što se proizvodi zatraže na tržištu.

U skladištenju zaliha, distributivnim centrima i maloprodaji, prognoze potražnje su bitne za sustav kontrole zaliha. Lokacije poput maloprodajnih skladišta i distributivnih centara sadrže velik broj proizvoda i potrebno je unaprijed predvidjeti zahtjeve koji će se pojaviti za svakog od njih. Prognoza potražnje -potrebna je kako bi se pronašao odgovarajući inventar i sredstva za ispunjavanje zahtjeva kupca. U distribucijskim centrima vrijeme izvođenja je obično dva ili više mjeseci, a u maloprodaji dva ili više dana. Menadžment je zadužen za procjenu broja dostupnih proizvoda. Prognoza je katalizator koji određuje vrijeme i količinu nabave. U tipičnom sustavu, prognoze se revidiraju jednom mjesečno kako novi podaci o potražnji postaju dostupni. [3]

Jedan od načina kategorizacije prognoze potražnje je temeljen na vremenskoj jedinici podataka koji su analizirani što je prikazano tablicom 1.

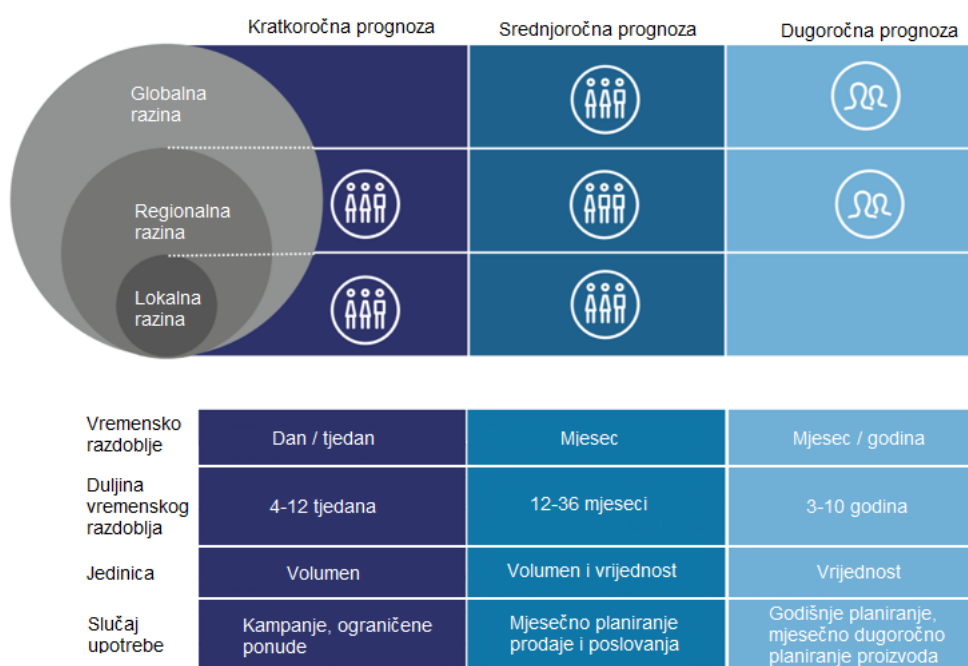
Tablica 1 Kategorizacija vrste prognoze na temelju vremenske jedinice podataka, Izvor: [3]

Kategorija vrste prognoze	Vremensko razdoblje analiziranih podataka	Primjer primjene potražnje	Korištene tehnike
Trenutno predviđanje	$\frac{1}{4}$ sata – 1 dan	Prognoza potražnje električne energije	Raznove rsne metode
Kratkoročno predviđanje	1 tjedan – 1 mjesec	Prognoza potražnje u industriji i maloprodaji	Eksp onenci jalni težinski prosjeci i derivacije
Srednjoročno predviđanje	1 mjesec – 1 godina	Maloprodaja i financijsko predviđanje	Regresija, aproksimiranje podataka krivuljom, prognoza vremenskih serija
Dugoročno predviđanje	1 godina – 1 desetljeće	Tehnološko predviđanje	DELPHI, trust mozgova, itd.

2.1. Kratkoročno predviđanje potražnje

S kraćim periodom predviđanja raste razina detalja, ali stupnjevi slobode postaju uži. Odnosno, postoji ograničenje zbog odluka koje su donesene na višim razinama kako je vidljivo na slici 1. Dakle, u kratkoročnom predviđanju nema mjesta optimizaciji kao što je to slučaj kod dugoročnog predviđanja jer su u većini slučajeva resursi već predodređeni za očekivani ishod. Kratkoročno predviđanje više se odnosi na postizanje jasnoće i preciznosti predviđanja

najvjerojatnijeg ishoda kako bi se dobio što profitabilniji rezultat. Također, iz perspektive Gartnerovog CORE okvira (konfiguriranje, optimiziranje, reagiranje i izvršavanje) za kategorizaciju procesa u lancu opskrbe, prepoznavanje potražnje postupak je koji je definiran kao „pomaganje u stvaranju inteligentnog odgovora na događaje izvršenja koji drže kratkoročne planove što je moguće bliže ciljevima poduzeća.“[4]



Slika 1 Vrste planiranja, Izvor:[5]

Kratkoročne prognoze u pravilu su točnije od dugoročnih prognoza, odnosno, kratkoročne prognoze imaju manju standardnu devijaciju pogreške u odnosu na srednji iznos od dugoročnih predviđanja potražnje. Japansko poduzeće Seven-Eleven iskoristilo je ovu ključnu karakteristiku kako bi unaprijedilo svoje performanse. Poduzeće je pokrenulo postupak nadopune koji omogućuje da odgovori na narudžbu u roku od nekoliko sati. Na primjer, ako je narudžba izdana do 10:00 sati ista se dostavlja isti dan u 19:00 sati. Stoga je potrebno predvidjeti što će se prodati te noći, samo 12 sati prije stvarne prodaje. Kratko vrijeme trajanja omogućuje uzimanje u obzir trenutne informacije koje mogu utjecati na prodaju proizvoda poput vremenskih prilika. Ova je prognoza vjerojatno točnija nego predviđanje potražnje tjedan dana unaprijed.[5]

Modeli prognoziranja temeljeni na eksponencijalno ponderiranom prosjeku pokazali su veću uspješnost u zadovoljavanju kriterija više od bilo koje druge grupe modela te postali osnova u mnogim situacijama za predviđanje kontrole zaliha zbog:[3]

- računalnih pogodnosti u vidu kraćeg vremena obrade prognoze
- robusnosti i sposobnosti praćenja procesa predviđanja tako da se mogu otkriti situacije izvan kontrole,
- jednostavnosti pokretanja kada se uključuju nove stavke bez prethodne povijesti potražnje,
- prilagodljivosti u smislu promjene osjetljivosti u skladu s karakteristikama podataka o potražnji i
- fleksibilnosti u pogledu sposobnosti funkcioniranja sa stagnacijom, rastom i sezonskim obrascima potražnje.

2.2. Srednjoročno predviđanje potražnje

Kako računalna snaga postaje jeftinija, sofisticiraniji modeli prognoziranja, koji zahtijevaju složenije izračune od izračuna poput eksponencijalno ponderiranog prosjeka, postaju financijski održivi kao metoda predviđanja za mnoge zalihe proizvoda. Metode predviđanja poput:

- regresije i aproksimiranja podataka krivuljom,
- Fourier-ove analize i
- Bayesian-ove prognoze

sada imaju neke od naprednijih, profesionalno razvijenih softvera za kontrolu zaliha.[3]

2.3. Dugoročno predviđanje potražnje

Dugoročno predviđanje u nekim industrijama pokriva period do deset godina. Predviđanja potražnje često se temelje na kvalitativnim unosima koje pružaju marketing, poslovno planiranje i financije. Ključnu ulogu u ovom procesu imaju dugoročne makroekonomske situacije, odjeli istraživanja i razvoja, sredstva za akviziciju te planovi proširenja proizvodnje. Predviđanje što će se dogoditi na tržištima na globalnoj razini oslanja se na podatke koji su obično grubozrnati što vodi planovima koji su na visokoj razini agregacije.

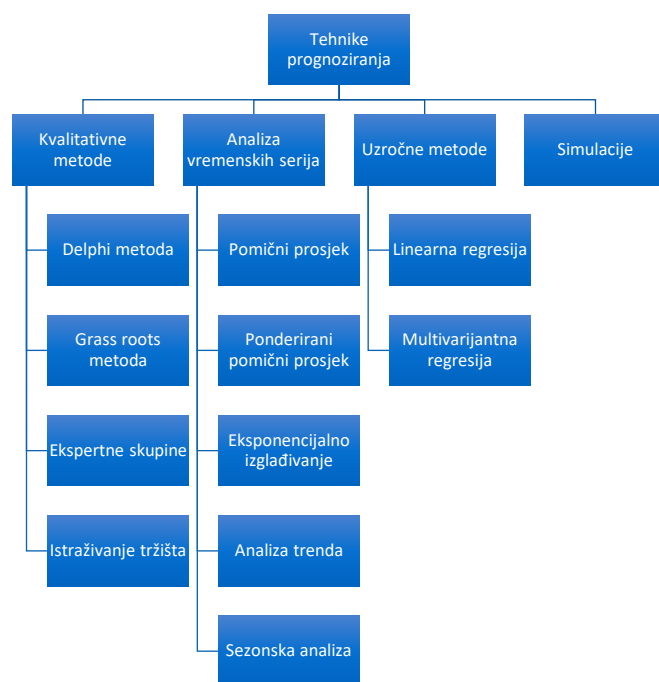
Tehnike statističkog predviđanja za dugoročno planiranje često se oslanjaju na modele koji se temelje na odabranim varijablama za određivanje utjecaja na dugoročna predviđanja. Primjerice, promjena bruto domaćeg prihoda po glavi stanovnika će povećati kupovnu moć ljudi u određenoj regiji ili zemlji.

Dugoročne odluke uključuju više stupnjeve slobode, odnosno nude širi prostor za donošenje odluka koje uključuju alternativne scenarije i kao takve zahtijevaju razmatranje niza faktora, vanjskih i unutarnjih, oni pozivaju na više razine sofisticiranosti što prikazuje slika 1. Budući da odluke na ovoj razini imaju snagu mijenjati dano (npr. nova tržišta i nove lokacije), pretpostavka kontinuiteta (prošlost je dobar indikator budućnosti) ponekad se raspada što zauzvrat zahtjeva više kvalitativne ili uzročne metode prognoze. S kraćim periodom predviđanja raste pritisak u odlučivanju te postoji potreba za bržim donošenjem odluka što jednostavnije vremenske metode čini korisnijima za odluke veće preciznosti i frekvencije.[4]

Zaključno, kombiniranje rezultata kratkoročnog prognoziranja sa srednjoročnim i dugoročnim predviđanjem može pružiti poboljšanje reaktivnosti te veću fleksibilnost opskrbnog lanca.

3. KLASIČNE METODE PROGNOZE POTRAŽNJE

Kako bi prognostička metoda dala optimalne rezultate vrlo je važno pronaći odgovarajuću metodu ili kombinaciju metoda. Većina autora navodi tri skupine metoda: kvalitativne, uzročne i metode vremenskih serija. Meindl i Chopra uz navedene dodaju i simulacije kao jednu od skupina prognostičkih metoda što je prikazano grafikonom 1.[6]



Grafikon 1 Kategorizacija tehnika prognoze potražnje, Izvor: [7]

3.1. Kvalitativne metode prognoze potražnje

Kvalitativne metode prognoze potražnje temelje se na stručnom mišljenju, odnosno intuiciji eksperata. Ove su metode korisne kada postoji nedostatak podataka te kada su prisutni subjektivni faktori poput reakcije tržišta na proizvod. Cilj metoda je kvalitativne informacije eksperata prevesti u kvantitativne procjene. Uslijed navedenih karakteristika kvalitativne metode su pogodnije za dugoročne prognoze.

3.1.1. Delphi metoda

Delphi metoda okuplja eksperte različitih područja djelatnosti poduzeća koji anonimno odgovaraju na postavljena pitanja. Odgovori se objedinjuju i šalju na daljnja razmatranja i

reviziju. Proces se ponavlja onoliko puta koliko je potrebno kako bi se postigao konsenzus. Ova je metoda pogodna za poduzeća s dovoljnim brojem eksperata. Ukoliko je teško doći do zajedničkog mišljenja vrijeme izvršavanja ove metode može biti dugo što je jedan od razloga zbog kojega se koristi za dugoročne prognoze potražnje.[8]

3.1.2. Grass roots metoda

Grass roots metoda zasnovana je na ideji da su osobe najbliže klijentu ili osobe uključene u krajnju upotrebu proizvoda sposobne predvidjeti poslovne trendove proizvoda ili usluga. Osobe najbliže klijentu sastavljaju zajedničku prognozu te je zatim šalju na više razine radi prilagođavanja. Nedostatak ove metode je dugo vrijeme izvršavanja.

3.1.3. Ekspertne skupine

Ova metoda temelji se na sastavljanju ekspertnih skupina kako bi se dobilo zajedničko mišljenje. Cilj ove metode je komunikacija subjekata te razmjena ideja i informacija. Kao kod Delphi metode, okupljaju se vanjski stručnjaci ili stručnjaci različitih funkcionalnih područja unutar poduzeća.[9]

3.1.4. Istraživanje tržišta

Istraživanje tržišta provodi se na dva načina, kroz testiranje određenog proizvoda ili grupe proizvoda od strane ciljane grupe potrošača te kroz anketne upitnike gdje se podaci dobivaju putem intervjua, pismenih ili telefonskih upitnika. Ova metoda najčešće se koristi pri razvoju prognoza novih proizvoda.[9]

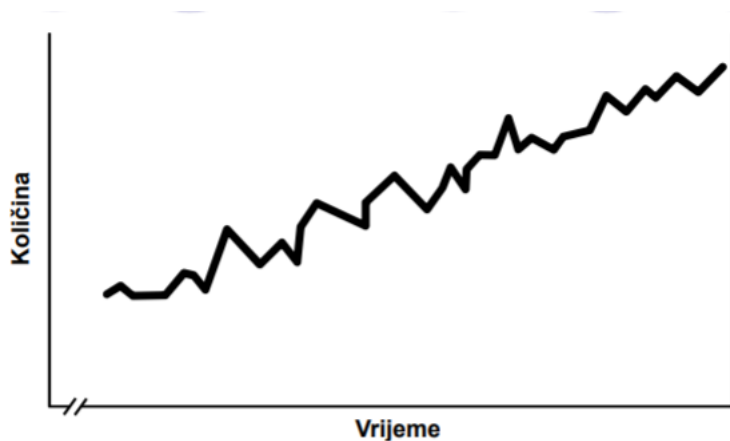
3.2. Metode analize vremenskih serija

Kvantitativne metode analize vremenskih serija za uspješnu prognozu koriste povijesne podatke potražnje. Ove su metode zasnovane na pretpostavci kako su povijesni podaci potražnje dobri pokazatelji buduće razine potražnje. Korištenje ovih metoda pogodno je kada uzorci potražnje ne variraju značajno od godine do godine.[5] U svom najjednostavnijem obliku pretpostavka je kako se povijest ponavlja u nekom obliku, ali je za precizniju prognozu moguće koristiti podatke poput ekonomskih pokazatelja ili vremenskih podataka. Prednosti ovih metoda su:

- preciznost u uvjetima stabilne okoline,
- pouzdanost za kratkoročne prognoze,
- prilagodba rezultata modela novim podacima,
- brzina i jednostavnost izvedbe.

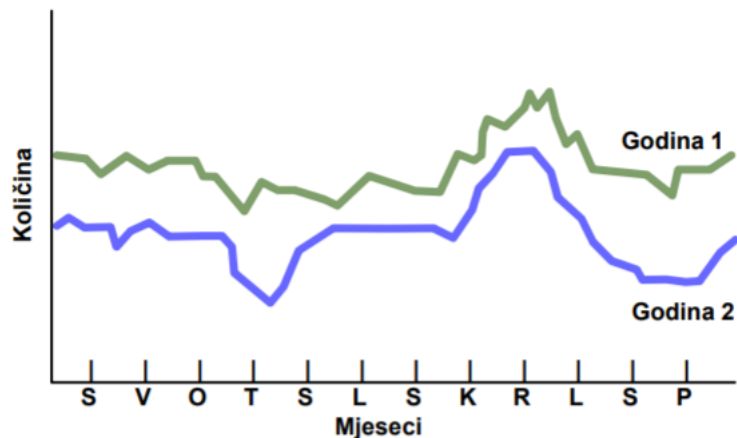
Za duge vremenske serije gdje postoje podaci za razdoblja duža od godine dana vrši se dekompozicija vremenskih serija, odnosno identificiraju se i analiziraju četiri komponente:

- Trend – pravilna promjena u podacima kroz vrijeme. Promjena može biti linearna kao što prikazuje slika 2, eksponencijalna ili geometrijska. Trend može biti bez promjena od perioda do perioda ili može biti uzlazni te silazni;



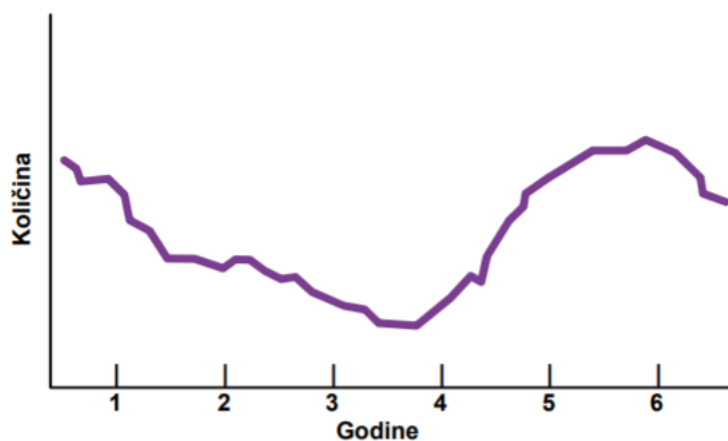
Slika 2 Pravilna linearna uzlazna promjena u podacima potražnje, [10]

- Sezonalnost – pravilno odstupanje od prosjeka unutar jediničnog razdoblja prikazano slikom 3. Jedinično razdoblje uglavnom je zasnovano na periodu jedne godine no može biti zasnovano i na tjednim ili dnevnim periodima;



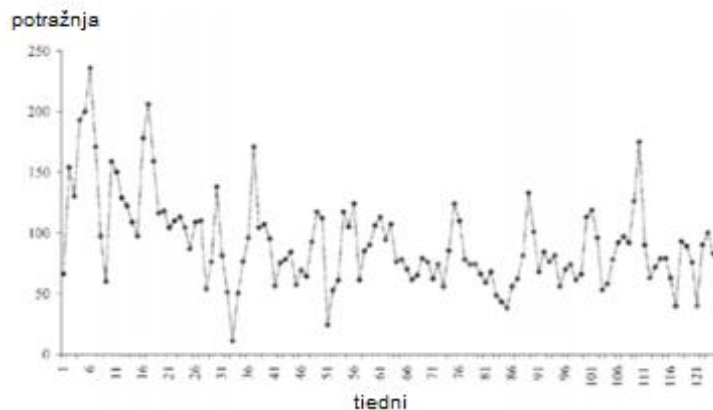
Slika 3 Sezonalna promjena podataka potražnje, [10]

- Ciklusi – slika 4 prikazuje pravilno odstupanje od prosjeka unutar jediničnog razdoblja. Jedinično razdoblje može biti nekoliko godina do nekoliko desetljeća;



Slika 4 Ciklička promjena podataka potražnje, [10]

- Slučajne promjene – slika 5 prikazuje nasumične promjene koje nastaju uslijed mnogih faktora koji utječu na potražnju kroz određeni period. Promjena može biti mala, a uzrokovati velike promjene u potražnji ili biti velika sa široko raspršenim točkama.[11]



Slika 5 Slučajne promjene podataka potražnje, [10]

Najčešće korištene metode vremenskih serija su:

- pomični prosjek,
- ponderirani pomični prosjek,
- eksponencijalno izgladivanje,
- autoregresivni integrirani pomični prosjek,
- analiza trenda,
- sezonska analiza.

3.2.1. Metoda pomičnih prosjeka

Jedan od jednostavnijih načina predviđanja je određivanje prosjeka potražnje. Jednostavni pomični prosjeci pomažu u neutralizaciji posljedica naglih kolebanja. Prednosti metode su ponajviše u jednostavnosti izvedbe i sposobnosti reduciranja posljedica cikličkih promjena u podacima dok su negativne strane potreba velikog broja podataka te spora reakcija na promjene na tržištu.

Metoda se koristi u uvjetima kada:[10]

- je potražnja stabilna,
- nema trendova u potražnji,
- nema periodičkih kolebanja.

Metoda je zasnovana na limitiranom broju perioda „ k “. U metodi se izbacuju najstariji podaci iz skupa nakon svakog pojavljivanja novih podataka. Ovo je načelo „pomicanja“ podataka jer za svako novo razdoblje potražnje najnoviji podaci uzrokuju uklanjanje najstarijih.

Prema sljedećoj formuli, prognoza potražnje za sukcesivno razdoblje $t + 1$ izračunava se na sljedeći način:

$$P_{t+1} = \frac{\sum_{i=1}^k X_{t-k+i}}{K} \quad 3.1$$

gdje je X_i potražnja tijekom perioda i .

Najbitnija odluka u metodi pomičnih prosjeka je određivanje broja perioda „ k “ koji se uzimaju u obzir. Savjetuje se uzimanje broja perioda „ k “ tako da minimiziraju vrijednost standardnog odstupanja.

$$S_k = \sqrt{\frac{\sum_{j=k+1}^t (P_j - X_j)^2}{t - (k + 1)}} \quad 3.2$$

gdje je P_j prognoza za period j i X_j efektivna potražnja perioda j .

Izračun se odvija prema sljedećem redoslijedu:

1. Izračun svih standardnih odstupanja povezanih s različitim „ k “ periodima;
2. Odabir „ k “ perioda koji minimiziraju S_k ;
3. Određivanje prognoze potražnje samo za „ k “ periode definirane u točki 2;
4. Određivanje intervala pouzdanosti formulom:

$$P_{t+1} \pm t_{\frac{\alpha}{2}, t-(k+1)} S_k \quad 3.3$$

Stupanj slobode razlika je broja opažanja unutar uzorka i „ k “ parametara koji zahtijevaju procjenu iz uzorka dok je vrijednost t spomenuta „kritična vrijednost“ ili „koeficijent pouzdanosti“ jer ovisi o odabranom stupnju pouzdanosti i o amplitudi uzorka.

Shodno tome, moguće je dodijeliti pravu vrijednost „ t “ ovisno o stupnju slobode i odabranom α , stupanj slobode koji odgovara odabranom α stupcu pokazat će ispravnu vrijednost „ t “.[4]

3.2.2. Metoda ponderiranih pomičnih prosjeka

Poboljšanje metode pomičnog prosjeka je metoda ponderiranih pomičnih prosjeka. Ova metoda omogućuje dodjeljivanje različite težine različitim podacima iz istog skupa podataka.

Prednost metode je u mogućnosti pomicanja i mijenjanja dodijeljenih težina podataka. Metoda omogućava prilagodljivost i fleksibilnost prema zahtjevima korisnika. S druge strane, ova metoda zahtijeva temeljito dodjeljivanje vrijednosti „ k “ i težina „ w “. Također, zahtijeva više vremena za izvršavanje, manje je precizna od metode eksponencijalnog izgladivanja te nije pogodna za kompleksnije skupove podataka sa značajkama trenda i sezonalnosti.

Formula za ponderirani prosjek je:

$$P_{t+1} = \sum_{i=1}^k X_{t-k+1} * W_i \quad 3.4$$

$$\sum_{i=1}^k W_i = 1 \quad 3.5$$

gdje je W_i težina efektivne potražnje za period i .

Kompleksnost ove metode leži u potrebi određivanja broja „ k “ (periodi koji minimiziraju standardno odstupanje pogreške prognoze) i periodične vrijednosti koja simbolizira „težinu“ povijesnih podataka (W_i). Pravilo koje se primjenjuje govori kako su najnoviji podaci najrelevantniji pokazatelj budućnosti, shodno tome, pridodjeljuje im se najveća težina.

Težina za određeni podatak se smanjuje s vremenom, relevantnost se prepušta novijim podacima. Automatski, kada je novi podatak dodan u skup, težina dodijeljena određenom periodu se pomiče nalijevo za jedan period te zadnji period izlazi iz skupa za proračun.[4]

3.2.3. Metoda eksponencijalnog izgladivanja

Nepotrebno je čuvati povijesne podatke kako bi izračunali pomični prosjek jer je prethodno izračunata prognoza već izračunata ovim povijesnim podacima. Iz ovog se razloga prognoza može temeljiti na staroj izračunatoj prognozi i novim podacima.[12]

Ova se metoda koristi pri izračunu prognoze s najnovijim podacima potražnje uz ponderiranje novije povijesti potražnje na relativan način.

Metoda se koristi:

- za kratke periode,
- kada nije zabilježen trend,
- te kada nije zabilježena sezonalnost.

Nedostatak metode su loši rezultati kada je potražnja za proizvodom niska stoga se koristi za proizvode velikog broja obrtaja.[11]

Pri izračunu potrebni su podaci:

- prognoza prethodnog razdoblja,
- potražnja u tom razdoblju,
- koeficijent izgladivanja α , $0 \leq \alpha \leq 1$.

Postavljeno je sljedeće:

Sustavna komponenta potražnje = razina

Početna procjena razine L_0 određuje se kao prosjek povijesnih podataka zbog pretpostavke da potražnja nema vidljivog trenda ili sezonalnosti. S obzirom na podatke o potražnji za razdoblja od 1 do n vrijedi sljedeće:

$$L_0 = \frac{1}{n} \sum_{i=1}^n D_i \quad 3.6$$

Trenutna prognoza za sve sljedeće periode jednaka je trenutnoj predviđenoj razini koja je dana izrazima:

$$F_{t+1} = L_t \text{ i } F_{t+n} = L_t \quad 3.7$$

Nakon promatranja potražnje, D_{t+1} , za period $t + 1$, izračunata je očekivana razina prema sljedećoj formuli:

$$L_{t+1} = \alpha D_{t+1} + (1 - \alpha) L_t \quad 3.8$$

gdje je α ($0 < \alpha < 1$) konstanta izgladivanja za određenu razinu. Izračunata vrijednost razine je ponderirani prosjek promatrane vrijednosti razine (D_{t+1}) za period $t + 1$ i staro očekivanje razine (L_t) za period t . Korištenjem formule 3.8 moguće je izraziti razinu danog perioda kroz funkciju trenutne potražnje i razine u prethodnom periodu. Navedeno daje novi zapis formule:

$$L_{t+1} = \sum_{n=0}^{t-1} \alpha (1 - \alpha)^n D_{t+1-n} + (1 - \alpha)^t D_1 \quad 3.9$$

Trenutno očekivanje razine je ponderirani prosjek svih prošlih zapažanja potražnje, s novijim zapažanjima većih pondera od starijih zapažanja. Viša vrijednost α odgovara prognozi koja više podliježe novijim opažanjima dok niža vrijednost α predstavlja stabilniju prognozu na koju manje utječe novije zapažanje. [5]

3.2.4. Autoregresivni integrirani pomični prosjek

Autoregresivni integrirani pomični prosjek (ARIMA, eng. *Autoregressive Integrated Moving Average*) predstavlja kombinaciju osnovnih modela za prognozu univarijantnih metoda. ARIMA modeli i modeli eksponencijalnog izgladivanja dva su najčešće korištena pristupa predviđanju vremenskih serija te pružaju komplementarne pristupe problemu. Iako su mogli eksponencijalnog izgladivanja temeljeni na opisu trenda i sezonalnosti u podacima, ARIMA modeli imaju za cilj opisati korelacije u podacima. Prije opisa ARIMA modela valja opisati i stacionarnost te diferencijaciju koje su ključne značajke modela.

Stacionarnost je svojstvo vremenskog niza koje ne ovisi o vremenu u kojemu se serija promatra. Kako trend i sezonalnost utječu na vrijednost vremenskih serija na različita vremena, vremenski nizovi s trendom ili sezonalnosti nisu stacionarni. Ukoliko postoje razlike između uzastopnih promatranja njihova razlika naziva se diferencijacija.[13]

ARIMA model sastoji se od tri dijela:

- Autoregresivni model (AR) pretpostavlja kako se vremenska serija ponaša poput linearnih funkcija svojih zaostalih vrijednosti;
- Integrirana komponenta (I) dio je modela koji radi diferencijaciju nad vremenskom serijom;
- Model pomičnog prosjeka (MA, eng. *Moving Average*) objašnjen u poglavlju 3.2.1.

3.2.5. Metode zasnovane na analizi trenda

Prethodne metode zasnovane su na pretpostavci da nema zabilježenih trendova niti sezonalnosti. Ukoliko je zabilježen trend koriste se metode regresijske analize i Holtova metoda eksponencijalnog zaglađivanja.

Holtova metoda koristi se kada je zabilježena razina trenda, ali ne i sezonalnost.

U ovom slučaju vrijedi sljedeće:

$$\text{Sustavna komponenta potražnje} = \text{razina} + \text{trend}$$

Početna procjena razine i trenda dobivena je izvođenjem linearne regresije između potražnje, D_t , i vremena, perioda t , sljedećom formulom:

$$D_t = at + b \quad 3.10$$

U ovom slučaju, provođenje linearne regresije između potražnje i vremena perioda je prihvatljivo zbog pretpostavke kako ne postoji sezonalnost već samo trend. Stoga je odnos potražnje i vremena linearan. Konstanta b mjeri procjenu potražnje u razdoblju $t = 0$ te je vrijednost za početnu razinu L_0 . Koeficijent smjera a označava razinu promjene u potražnji po periodu te je početna pretpostavka trenda L_0 .

U periodu t , dane pretpostavke razine L_t , i trenda T_t , prognoza za buduće periode je izražena formulama:

$$F_{t+1} = L_t + T_t \text{ i } F_{t+n} = L_t + nT \quad 3.11$$

Nakon promatranja potražnje za period t , računa se predviđanje za razinu i trend prema sljedećem:

$$L_{t+1} = \alpha D_{t+1} + (1 - \alpha)(L_t + T_t) \quad 3.12$$

$$T_{t+1} = \beta(L_{t+1} - L_t) + (1 - \beta)T_t \quad 3.13$$

gdje je α ($0 < \alpha < 1$) konstanta izgladivanja za razinu i β ($0 < \beta < 1$) je konstanta izgladivanja za trend. U oba ažuriranja revidirana je procjena (razine ili trenda) ponderirani prosjek promatrane vrijednosti i stara procjena.[5]

3.2.6. Metode analize sezonskih podataka

Velik broj proizvoda ima sezonalni ili periodični uzorak potražnje. Periodičnost je uglavnom određena godišnjim periodima, no može biti određena i mjesečnim, tjednim i dnevnim periodima.

Korisni pokazatelj stupnja sezonske varijacije za određeni proizvod je indeks sezonalnosti. Ovaj indeks pokazuje koliko će razina potražnje tijekom sezone biti iznad ili ispod generalnog prosjeka potražnje proizvoda. Formula kojom se računa indeks sezonalnosti jest:

$$\text{Indeks sezonalnosti} = \frac{\text{prosječna potražnja perioda}}{\text{prosječna potražnja za sve periode}} \quad 3.14$$

Prosječna potražnja za sve periode vrijednost je koja prosječno predstavlja sezonalnost i naziva se desezonirana potražnja. Shodno tome, formula 3.15 poprima novi zapis:

$$\text{Indeks sezonalnosti} = \frac{\text{prosječna potražnja perioda}}{\text{desezonirana potražnja}} \quad 3.15$$

Winterova metoda eksponencijalnog izgladivanja koristi se kada sustavna komponenta potražnje ima razinu, trend i sezonalni faktor. Za ovaj slučaj vrijedi sljedeće:

$$\text{Sustavna komponenta potražnje} = (\text{razina} + \text{trend}) \times \text{sezonalni faktor}$$

Pretpostavka je kako je periodičnost potražnje p . Procjene razine (L_0), trenda (T_0) i sezonskih faktora ($S_1, \dots, S_p$) dobivene su korištenjem postupaka statičkog predviđanja opisanog ranije u poglavlju.

U periodu t , dano je predviđanje razine, L_t , trenda, T_t , i faktora sezonalnosti, $S_1, \dots, S_p$, predviđanje za buduće periode računa se prema:

$$F_{t+1} = (L_t + T_t)S_{t+1} \text{ i } F_{t+l} = (L_t + lT_t)S_{t+l} \quad 3.16$$

Promatranoj potražnji za period $t + 1$ modificirana je razina, trend i sezonalni faktori prema sljedećem:

$$L_{t+1} = \alpha(D_{t+1}/S_{t+1}) + (1 - \alpha)(L_t + T_t) \quad 3.17$$

$$T_{t+1} = \beta(L_{t+1} - L_t) + (1 - \beta)T_t \quad 3.18$$

$$S_{t+p+1} = \gamma(D_{t+1}/L_{t+1}) + (1 - \gamma)S_{t+1} \quad 3.19$$

gdje je α ($0 < \alpha < 1$) konstanta izgladivanja razine, β ($0 < \beta < 1$) konstanta izgladivanja trenda i γ ($0 < \gamma < 1$) konstanta izgladivanja sezonalnih faktora. U svakom ažuriranju (razine, trenda ili sezonalnih faktora) revidirana je procjena ponderiranog prosjeka promatrane vrijednosti i stare procjene.[5]

3.3. Uzročne metode

Za razliku od metoda analize vremenskih serija, koje koriste povijesne podatke kao procijenjene varijable za prošlost i imaju za cilj ekstrapolirati ih kao varijable za budućnost, uzročne metode koriste neovisne varijable koje su u korelaciji s prognoziranim varijablom. Teorija uzročnih metoda je da je potražnja za grupom proizvoda izravno proporcionalna ili korelirana s aktivnostima na drugom polju.

Problem ovih metoda je pronaći indikator koji je u korelaciji s potražnjom i onaj koji vodi potražnji, odnosno onaj koji se javlja prije potražnje. Kada nije moguće pronaći vodeći pokazatelj moguće je koristiti pokazatelje koje vlada ili druga organizacija koristi za prognoziranje. Odnosno, temeljenje prognoze na prognozi.

Ova vrsta predviđanja najkorisnija je za predviđanje ukupne potražnje za proizvodima poduzeća. Stoga se najčešće koristi u planiranju poslovanja i proizvodnje, a ne za predviđanje izlaska proizvoda iz opskrbnog lanca.

Regresijske su metode najčešće među uzročnim metodama. Jednostavna regresija, koja je posebna klasa regresijskih metoda, koristi jednu neovisnu varijablu. Odnos između ovisnih i neovisnih varijabli u slučaju jednostavne regresije može se izraziti na sljedeći način:

$$Y_i = a + b * X_i + e \quad 3.20$$

gdje su Y_i i X_i uparene zavisne i nezavisne varijable, a e je pogreška. Varijabla X može predstavljati vrijeme u slučaju vremenske serijske regresije. [4]

3.4. Simulacijske metode

Simulacijske metode predviđanja oponašaju potrošačeve izbore koji potiču promjenu potražnje kojom se dolazi do predviđanja. Pomoću simulacije, poduzeće može kombinirati vremenske serije i uzročne metode kako bi odgovorilo na pitanja poput: Kakav će biti utjecaj promotivnih cijena? Kakav će biti utjecaj konkurencije koja otvara prodavaonicu u blizini? Zrakoplovne tvrtke simuliraju kupčevo ponašanje kako bi predviđale potražnju sjedala s višim cijenama kada za niže cijene nema mjesta. Poduzeće će možda teško odlučiti koja je metoda najprikladnija za predviđanje. Nekoliko studija je pokazalo da je upotreba više metoda predviđanja za stvaranje kombinirane prognoze efikasnija od upotrebe bilo koje jedine metode. S bilo kojom metodom predviđanja uvijek postoji slučajni element koji se ne može objasniti povijesnim obrascima potražnje. Stoga se svaki promatrani zahtjev može raščlaniti na sustavnu i slučajnu komponentu:

$$\text{Promatrana potražnja (O)} = \text{sustavna komponenta (S)} + \text{slučajna komponenta (R)}$$

Sustavna komponenta mjeri očekivanu vrijednost potražnje i sastoji se od razine, trenutne desezonirane potražnje; trenda, stopa rasta ili pada potražnje u sljedećem razdoblju; i sezonalnosti, predvidljiva sezonska kolebanja potražnje.

Slučajna komponenta je dio prognoze koji odstupa od sustavnog dijela. Poduzeće ne može (i ne bi smjelo) predvidjeti smjer slučajne komponente. Sve što tvrtka može predvidjeti je veličina

i varijabilnost nasumičnih komponenti koje osiguravaju mjeru pogreške. Cilj predviđanja je filtriranje slučajne komponente i procjena sustavne komponente. Pogreška prognoze mjeri razliku između prognoze i stvarne potražnje. U prosjeku, dobra metoda predviđanja ima grešku čija je veličina usporediva sa slučajnom komponentom potražnje. [5]

4. STROJNO UČENJE

Strojno učenje (*eng. Machine Learning*) je područje unutar računalnih znanosti. Vrsta je umjetne inteligencije (*eng. Artificial Intelligence*) koje programerima omogućava programerima pisanje programa na jednostavniji način. Fokus je na razvijanju programa koji uče računalo da se mijenja kada je izloženo novim informacijama te dalje razvija. Cilj je da program razumije i rješava problem pomoću raznih algoritama bez ikakve ljudske pomoći. Arthur Samuel je 1959. definirao strojno učenje kao „*Polje znanosti koje računalima daje mogućnost učenja bez da su eksplicitno programirana*“. Ovo je bila velika promjena u odnosu na to kako je većina računalnih znanstvenika gledala na računala. I danas je većina onoga što računala rade eksplicitno određeno zapisanim kôdom programera. Strojno učenje potpuna je suprotnost jer se ne kreiraju eksplicitni zadaci već se računalu daju podaci i alati potrebni za proučavanje i rješavanje problema. Ovim načinom računalu se daje mogućnost da pamti ono što nauči i na taj način prilagođava i razvija.

Princip stajnog učenja vrlo je jednostavan i odvija se kroz nekoliko koraka. Osoba pruža računalu podatke i algoritam koji određuje način na koji će računalo obraditi podatke. Algoritam je skup pravila za rješavanje problema u određenom broju koraka. Uz strojno učenje algoritmi uključuju pravila koja računalo koristi za oblikovanje modela, testiranje modela na primjerima, a zatim podešavanje modela kako bi ga računalo moglo koristiti za precizno tumačenje podataka koji mu se u budućnosti daju. Model koje računalo kreira može biti veoma složen te može ili ne mora biti nešto što čovjek u potpunosti razumije. Poput čovjeka, računalo može učiti iz iskustva (okruženje ili unos podataka), iz pokušaja (pogreške) i od korisnika (obuka i ispravljanje).

Strojno učenje uključuje pet osnovnih komponenti[14]:

- učenik – računalo;
- podaci – unos potreban za obuku i testiranje računala te za tumačenje nakon treninga;
- algoritam – matematička formula koja prima i analizira ulazne varijable kako bi predvidjelo rješenje unutar prihvatljivog raspona;
- hiperparametri – parametri koji se ne mogu naučiti iz podataka, određuje ih korisnik. Obično je to odabir pravog algoritma, podešavanje metrike (poput k u k -sredina klasteriranju) ili broj skrivenih slojeva u dubokoj neuronskoj mreži;

- model – algoritam s parametrima koji govore računalu kako obrađivati i interpretirati ulazne podatke.

Proces strojnog učenja odvija se kroz korake[14]:

- dodavanje hiperparametara odnosno, odlučivanje o algoritmu strojnog učenja;
- dodavanje podataka, uobičajeno parovi ulaznih i izlaznih varijabli za treniranje. Pri čemu su ulazne i izlazne varijable poput pitanja i odgovora. Tijekom treninga dodajemo obje varijable, i pitanja i odgovore.
- korištenjem algoritma, računalu vrši proračune na ulaznim varijablama, prilagođavajući parametre modela ako je potrebno kako bi proizveo izlazne varijable povezane s ulazima;
- dok obrađuje podatke treninga računalu kreira model koji se sastoji od algoritma i parametara koji najpreciznije izračunavaju izlazne varijable na temelju datih ulaznih varijabli;

kada računalu dobiva ulazne varijable s nepoznatim izlaznim varijablama može predvidjeti izlazne varijable.

Tablica 2 Tehnike strojnog učenja, Izvor: [15]

Tehnika strojnog učenja	
Učenje	Algoritam
Tip:	
Nadzirano (npr. predviđanje rasta, kreditno stanje)	Regresija grebena, neuronska mreža, stablo odlučivanja
Nenadzirano (npr. klaster analiza, otkrivanje anomalija)	k-sredina/hijerarhijsko klasteriranje, glavne komponente
Metoda:	
Treniranje i testiranje	Algoritam primijenjen na podskupu podataka (set za treniranje) i uspješnost ocjenjena na drugom podskupu (set za testiranje)
Validacija	Podešavanje parametara algoritma

Ovisno o vrsti podataka i razini nadziranja strojno učenje dijeli se u četiri kategorije: nadzirano učenje, polu-nadzirano učenje, nenadzirano učenje i podržano učenje. Tablica 2 pokazuje najčešće korištene algoritme za pojedinu vrstu strojnog učenja.

4.1. Nadzirano učenje

Nadzirano učenje (*eng. supervised learning*) koristi označene podatke (*eng. labeled*) što znači da su ulazi definirani i poznati. Na skupu podataka za trening (*eng. training*) algoritam uči relacije i povezanosti između ulaznih i izlaznih podataka. Nakon treninga algoritmu se daju nepoznati (testni) podaci za koje kreira rješenje problema.

Istaknuti algoritmi nadziranog učenja su:[16]

- k-najbliži susjed,

- linearna regresija,
- logistička regresija,
- metoda potpunih vektora (metoda jezgrenih funkcija),
- stablo odluke,
- neuronske mreže.

4.2. Polu-nadzirano učenje

Polu-nadzirano učenje (*eng. semi-supervised learning*) koristi djelomično označene podatke tijekom treniranja, uobičajeno je to veća količina neoznačenih podataka i manja količina označenih. Neoznačeni podaci, ako se koriste zajedno s malom količinom označenih podataka, mogu značajno poboljšati točnost učenja.

4.3. Nenadzirano učenje

U nenadziranom učenju (*eng. unsupervised learning*) koriste se samo ulazni neoznačeni podaci i ne postoji supervizor koji pruža ispravne vrijednosti ulaznih i izlaznih varijabli. Cilj nenadziranog učenja je pronaći regularnosti u ulaznim podacima. U ulaznim podacima postoje obrasci koji se pojavljuju češće od ostalih te je cilj pronaći uzorak ponavljanja. U statistici se ovo naziva procjena gustoće. Metoda za procjenu gustoće je klasteriranje gdje je cilj pronaći klustere ili grupe ulaznih podataka. Klasteriranje se vrši grupiranjem podataka prema atributima što omogućuje lako identificiranje i odbacivanje onih koji ne spadaju u određene klustere, odnosno grupiranje onih s istim atributima.

Neki od algoritama nenadziranog učenja su:[16]

- k-prosječne vrijednosti,
- hijerarhijska klaster analiza i
- maksimizacija očekivanja.

4.4. Podržano učenje

Podržano učenje (*eng. reinforcement learning*) veoma je drugačije od prethodno navedenih vrsta učenja. Podržano učenje temelji se na „nagrađivanju“ i „kažnjavanju“ algoritma. Ono se bavi problemom pronalaženja odgovarajućih radnji koje treba poduzeti u određenoj situaciji

kako bi se nagrada maksimalno povećala. Algoritam učenja koristi samo ulazne podatke, kao što je to u nadziranom učenju te sam otkriva optimalne rezultate postupkom pokušaja i pogreške. Postoji niz stanja i radnji u kojima algoritam učenja djeluje u interakciji s okolinom. U velikom broju slučajeva trenutna akcija ne samo da utječe na neposrednu nagradu već ima utjecaj i na nagradu u svim narednim vremenskim koracima. Ova vrsta učenja često se primjenjuje na robotima za učenje radnji poput hodanja.[17]

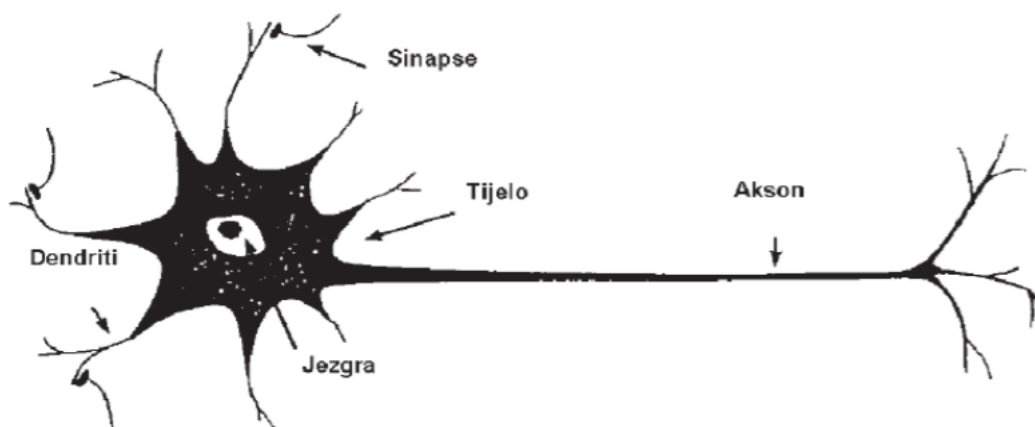
5. METODE PROGNOZE POTRAŽNJE STROJNOG UČENJA

Napredak u računalnoj tehnologiji pruža mogućnost pohrane i obrade velikih količina podataka kao i pristup s fizički udaljenih mjesta preko računalne mreže. Primjerice, lanac supermarketa ima stotine trgovina diljem zemlje u kojima prodaje velik broj proizvoda. Na prodajnim mjestima bilježe se podaci o svakoj transakciji poput datuma, identifikacijskog broja kupca, prodanog proizvoda, njegove cijene i slično. Ovi podaci obično postaju korisni tek kada se analiziraju i obrade u format koji možemo koristiti za predviđanja. Iako nisu poznati detalji procesa koji se temelje na generiranju podataka poput ponašanja potrošača poznato je kako uzorci nisu potpuno nasumični i kako postoje određeni obrasci. Iz prikupljenih podataka nije moguće potpuno identificirati postupak, ali je moguće konstruirati dobru i korisnu aproksimaciju. Niša strojnog učenja je uočavanje obrazaca ili pravilnosti. Pod pretpostavkom kako se budućnost ne razlikuje mnogo od prikupljenih obrazaca očekuju se točni rezultati predviđanja.[18]

Kako se povećava broj tehnika i parametara predviđanja, postaje sve teže odabrati odgovarajuću metodu u određenom kontekstu. Jedno od mogućih rješenja je osloniti se na klasu algoritama nazvanih „univerzalni aproksimatori“ koji su u mogućnosti približno odrediti neprekidnu funkciju za proizvoljnu točnost. Korištenjem ove klase algoritama može se naučiti svaka funkcija za podatke prošlosti i budućnosti. Neuronske mreže i metode potpornih vektora jedne su od tehnika strojnog učenja koje spadaju u klasu univerzalnih aproksimatora.[19]

5.1. Umjetne neuronske mreže

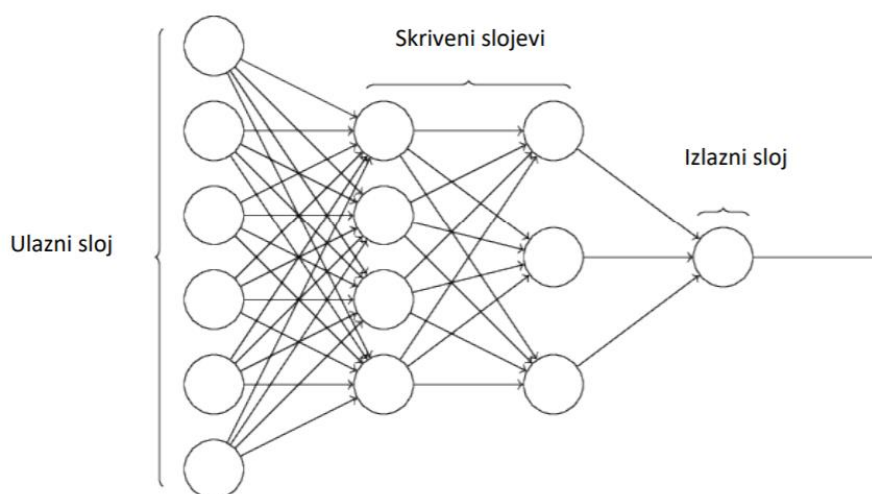
Biološke neuronske mreže građene su od bioloških neurona povezanih u periferni živčani sustav što prikazuje slika 6. Neuron, živčana stanica, je osnovna jedinica živčanog sustavakoj je građena od tijela same stanice i mnoštva aksona i dendrita. Dendriti prenose živčane impulse s osjetilnih organa do tijela stanice. Aksoni s jedne strane povezani s tijelom stanice, a s druge strane podijeljeni na niz grana. Sinapsa je razmak između završetka aksoma prethodnog neurona i dendrita sljedećeg neurona kojom signali prelaze s jedne na drugu stanicu. Umjetne neuronske mreže potječu upravo iz ovog procesa.



Slika 6 Biološki neuron Izvor: [Bolf, Jerbić (2006)]

Neuronska mreža je struktura umjetnih neurona koji su međusobno povezani i interaktivni kroz operacije obrade signala. Ističe se velikom prilagodljivosti jer s malim brojem primjera može naučiti procijeniti parametre neke populacije. Neuronske mreže izgrađene su od jednostavnih jedinica koje se nazivaju neuroni. Jedinice su organizirane u slojeve.

Neuronske mreže sastoje se od više slojeva što prikazuje slika 7. Prvi sloj nazvan je ulazni sloj, a posljednji izlazni. Međuslojeve, ukoliko ih ima, nazivamo skriveni slojevi. Informacije koje se analiziraju dovode se do ulaznog sloja, a zatim se šire u neurone drugog sloja na obradu. Rezultat obrade prenosi se u sljedeći sloj i tako sve do posljednjeg, izlaznog sloja.[20]

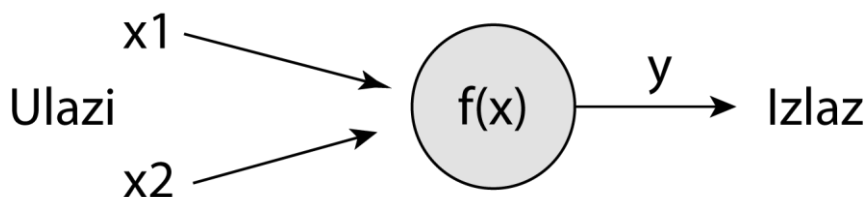


Slika 7 Osnovna arhitektura neuronske mreže, [21]

Svaka jedinica prima neke informacije od drugih jedinica koje zatim obrađuje i pretvara u izlazne jedinice. Neuron za određeni unos x_1, x_2 kroz funkciju $f(x)$ daje iznos y što prikazuje slika 8. Primjerice, ukoliko je $f(x)$ suma onda vrijedi:

$$y = x_1 + x_2.$$

5.1

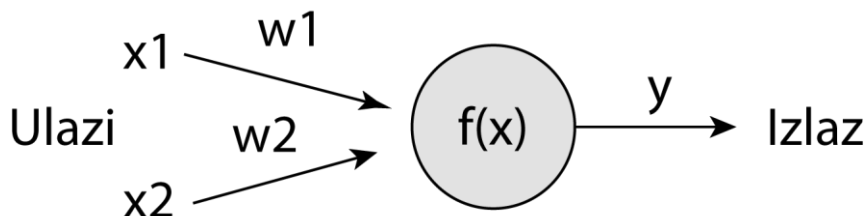


Slika 8 Prikaz neurona

Kako bi neuronima omogućili učenje treba omogućiti prilagodbu pojedinih parametara. Iz ovog razloga dodajemo težinske koeficijente w_n za pojedini unos x_n . Sada vrijedi:

$$y = x_1 * w_1 + x_2 * w_2.$$

5.2

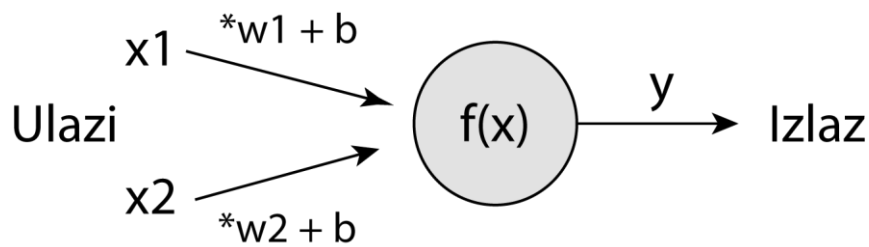


Slika 9 Prikaz neurona s težinama

Ovdje se pojavljuje problem ukoliko je $x = 0$, tada težinski koeficijent ne utječe na vrijednost y . Ovo je razlog dodavanja tzv. biasa (šuma), jedinice pristranosti (*eng. bias*) b . Na ovaj način uvijek se može utjecati na vrijednost y . Vrijednost umnoška ulazne vrijednosti x_n i težine w_n treba biti dovoljno velika kako bi prevladala vrijednost biasa b i tako utjecao na krajnji izlaz y .

$$y = \sum_{i=1}^n x_i w_i + b_i$$

5.3



Slika 10 Prikaz neurona s težinskim koeficijentima i koeficijentima pristranosti

5.2. Šuma slučajnih odluka

Slučajna šuma ili šuma slučajnih odluka algoritam je nadziranog učenja koji se koristi za raspoznavanje uzoraka i regresijsku analizu. Glavna odlika algoritma je kreiranje većeg broja klasifikatora u obliku stabla odluke koja sudjeluju u donošenju konačnog rezultata. Svako od stabala trenirano je na određenom broju uzoraka podatkovnog skupa za treniranje.

Algoritam slučajnih šuma koristi dva parametra:[22]

- n - broj stabala koja će se generirati i
- m – broj atributa iz kojeg će se svako stablo generirati.

Optimalne vrijednosti parametara ovise o broju atributa uzoraka i broju uzoraka za učenje. Za vrijednosti parametra n preporučeno je koristiti vrijednost veću od 100, a za parametar m vrijedi:

$$m = \log_2 M \quad 5.4$$

ili

$$m = \sqrt{M} \quad 5.5$$

gdje je M broj atributa uzoraka. Povećavanjem broja stabala n ne dolazi do zasićenja, odnosno pogoršanja performansi algoritma. Ukoliko je velik broj vrijednosti atributa zašumljen valja povećati broj atributa iz kojeg će svako stablo raditi m . [22]

6. STUDIJA SLUČAJA

U ovom poglavlju obrađen je skup podataka neimenovanoga poduzeća. Poduzeće se sastoji od petnaest trgovina u kojima postoji izbor od pedeset proizvoda. Programski zadatak ovoga rada je uz pomoć metoda strojnog učenja i klasičnih metoda prognozirati tjednu potražnju na razini poduzeća i pojedinog proizvoda za određeni period. Cilj ovog zadatka je usporediti rezultate korištenih metoda te performanse i isplativost korištenja.

6.1. Opis i obrada skupa podataka

Korišteni skup podataka prikazan je tablicom 3 te obuhvaća dnevne podatke u periodu od pet godina, odnosno od 2013. godine do kraja 2017. godine. Skup podataka vrlo je jednostavne strukture i sastoji se od sljedećih stupaca:

- datum (eng. *date*);
- trgovina (eng. *store*);
- proizvod (eng. *item*);
- prodaja (eng. *sales*).

Tablica 3 Početni skup podataka

date	store	item	sales
2013-01-01	1	1	13
2013-01-02	1	1	11
...	...	...	...
2017-12-31	15	50	82

Kako bi se skup podataka prilagodio zadatku ovoga rada potrebna je dodatna obrada. Dorada podataka, kao i ostatak programskog zadatka pisan je programskim jezikom Python uz korištenje knjižnica Pandas i NumPy. Python je programski jezik opće namjene i visoke razine. Njegova jezična konstrukcija i objektno orijentirani pristup omogućuju programerima pisanje jasnog kôda za male i velike projekte.[23] Pandas je programska knjižnica otvorenog kôda koja služi za obradu i analizu podataka. Glavne odlike ove knjižnice su DataFrame objekti, poravnavanje podataka, integrirano rukovanje podacima koji nedostaju, umetanje i brisanje

stupaca i sl.[24] NumPy je softver otvorenoga kôda pisan za Python programski jezik. Knjižnica NumPy pruža podršku za velika, višedimenzionalna polja i matrice te matematičke operacije visoke razine kojima se vrši obrada navedenih polja i matrica.[25] Skup za pojedini proizvod grupiran je po datumu, a zatim po tjednima s ponedjeljkom kao početkom tjedna što je prikazano u ispisu programskog kôda Programski kôd 1.

```
df = pd.read_csv('../data.csv')
df = df[df['item'] == item]
df['date'] = pd.to_datetime(df['date']) - pd.to_timedelta(7, unit='d')
df = df.filter(['date', 'sales']).groupby([pd.Grouper(key='date', freq='W-MON')]).sum().reset_index()
```

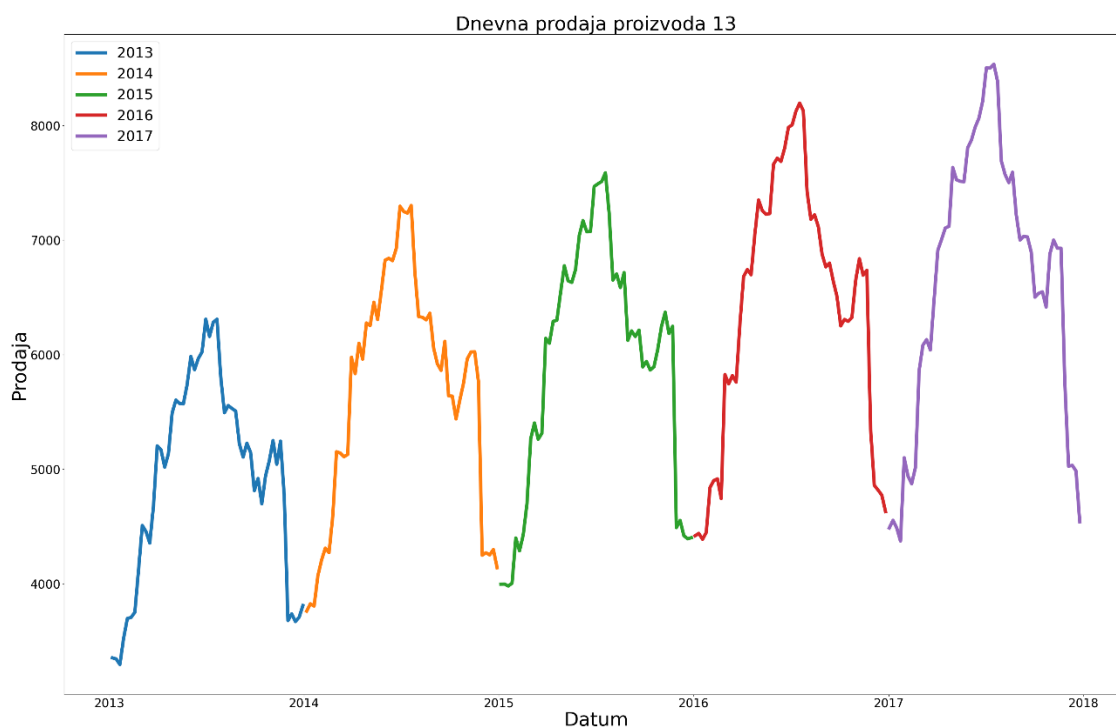
Programski kôd 1

Nasumičnim odabirom od 50 proizvoda za daljnju analizu i prikaz rezultata metoda odabran je proizvod 13. Tablica 4 prikazuje skup podataka nakon grupiranja.

Tablica 4 Skup podataka proizvoda 13 nakon grupiranja

date	sales
2012-12-31	3218
2013-01-07	3353
...	...
2017-12-25	4542

Slika 11 prikazuje tjednu prodaju proizvoda 13 kroz razdoblje od 5 godina. Pojedina boja označava pojedinu godinu potražnje. Sa slike 11 je jasno vidljivo kako postoji sezonalnost s četiri ciklusa. Osim sezonalnosti iz grafa se može iščitati kako postoji trend u potražnji.



Slika 11 Tjedna prodaja proizvoda

Kako bi metode mogle pronaći korelacije dodani su novi stupci:

- tjedan (eng. `week`);
- pomični tjedan s korakom veličine n (eng. `shift_week+n`) pri čemu je n -ti element iz skupa `[1, 12]`;
- mjesec (eng. `month`);
- pomični mjesec s korakom veličine n (eng. `shift_month+n`) pri čemu je n -ti element iz skupa `[1, 4]`;
- tjedna prodaja (eng. `week_sale`);
- pomična prodaja s korakom veličine n (eng. `shift_sale+n`) pri čemu je n -ti element iz skupa `[1, 4]`;
- prosjek prodaje perioda od dva tjedna (eng. `two_week_avg`);
- prosjek prodaje perioda od tri tjedna (eng. `three_week_avg`);
- prosjek prodaje perioda od četiri tjedna (eng. `four_week_avg`).

Parametar `week` označava grupiranu prodaju za period od jednog tjedna, a `shift_week+n` predstavlja tjedan pomaknut za veličinu vremenskog prozora n . Parametar `month` predstavlja

mjesec s oznakama od 1 do 12, a `shift_month+n` označava mjesec pomaknut za veličinu vremenskog prozora n . `week_sale` predstavlja prodaju u određenom tjednu, a `shift_sale+n` pomaknutu prodaju za veličinu vremenskog prozora n . Parametri `two_week_average`, `three_week_average` i `four_week_average` označavaju prosječnu vrijednost prodaje za periode od dva, tri i četiri tjedna. Dodavanje navedenih novih stupaca prikazano je u ispisu programskog kôda Programski kôd 2 prikazanog u nastavku.

```
def two_week_avg(df):
    sum = 0
    week_avg = []

    for i in range(1, -1, -1):
        for j in range(i):
            sum += df[j]
        if(i!=0):
            week_avg.append(sum/i)
        sum = 0
    week_avg.append(df[0])
    week_avg.reverse()

    for row in range(len(df) - 2):
        for row in range(row, row + 2):
            sum += df[row]
        week_avg.append(sum / 2)
        sum = 0
    return week_avg

def three_week_avg(df):
    sum = 0
    week_avg = []

    for i in range(2, -1, -1):
        for j in range(i):
            sum += df[j]
        if(i!=0):
            week_avg.append(sum/i)
        sum = 0
    week_avg.append(df[0])
    week_avg.reverse()

    for row in range(len(df) - 3):
        for row in range(row, row + 3):
            sum += df[row]
        week_avg.append(sum / 3)
        sum = 0
    return week_avg

def four_week_avg(df):
    sum = 0
    week_avg = []

    for i in range(3, -1, -1):
```

```

        for j in range(i):
            sum += df[j]
        if(i!=0):
            week_avg.append(sum/i)
        sum = 0
    week_avg.append(df[0])
    week_avg.reverse()

    for row in range(len(df) - 4):
        for row in range(row, row + 4):
            sum += df[row]
        week_avg.append(sum / 4)
        sum = 0
    return week_avg

def monthly_sales(df):
    sum = 0
    day_avg = []

    for i in range(29, -1, -1):
        for j in range(i):
            sum += df[j]
        if (i != 0):
            day_avg.append(sum / i)
        sum = 0
    day_avg.append(df[0])
    day_avg.reverse()

    for row in range(len(df) - 30):
        for row in range(row, row + 30):
            sum += df[row]
        day_avg.append(sum)
        sum = 0
    return day_avg

data['week'] = df.date.dt.week
for i in range(1, 13):
    data[f'shift_week+{i}'] = data['week'].shift(i)

data['month'] = df.date.dt.month
for i in range(1, 5):
    data[f'shift_month+{i}'] = data['month'].shift(i)

data['week_sale'] = df['sales']
for i in range(1, 5):
    data[f'shift_sale+{i}'] = data['week_sale'].shift(i)

data['two_week_avg'] = two_week_avg(data['week_sale'].tolist())
data['three_week_avg'] = three_week_avg(data['week_sale'].tolist())
data['four_week_avg'] = four_week_avg(data['week_sale'].tolist())

```

Programski kód 2

6.2. Primjena metoda

Prethodno pripremljeni skup podataka sadrži NaN (*eng. not a number*) vrijednosti koje označavaju „ne broj“, odnosno nevažeći broj. NaN vrijednosti u ovome slučaju nastaju zbog pomičnih prozora pa ih je potrebno odbaciti kako bi podaci bili cjeloviti. Uobičajeno u području strojnog učenja podaci se dijele na skup podataka za validaciju i skup podataka za testiranje modela. Podatkovni skup za treniranje modela čini skup perioda od 1. siječnja 2013. do 31. prosinca 2016. godine. Skup podataka za testiranje uspješnosti pojedinog modela sastoji se od podataka perioda od 1. siječnja do 31. prosinca 2017. godine.

Skup podataka za validaciju dodatno je podijeljen na skup podataka za treniranje i testiranje uz pomoć `train_test_split` razreda iz modula `model_selection` sklearn knjižnice. Ovaj razred podatke nasumično dijeli u omjeru 80% podataka za treniranje i 20% podataka za testiranje modela. U podjeli je korišteno nasumično stanje određenog broja kako bi se osigurala podjednaka podjela podataka pri svakom izvršavanju. Podjela podataka prikazana je ispisom programskog kôda Programski kôd 3 prikazanog u nastavku.

```
X = df.drop('week_sale', axis=1)
y = df['week_sale']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Programski kôd 3

Podaci su na prethodno opisani način podijeljeni za metode neuronske mreže i nasumične šume. Za metodu autoregresivnog integriranog pomičnog prosjeka podaci su podijeljeni na jednostavniji način zbog potrebe osiguravanja frekvencije. Podjela je za ovu metodu nešto jednostavnija i dijeli se na skup za treniranje kojemu pripadaju podaci perioda od 2013. do 2016. godine i skup za testiranje koji čine podaci za 2016. godinu.

6.2.1. Primjena metode autoregresivnog integriranog pomičnog prosjeka

Prognoziranje potražnje metodom autoregresivnog integriranog pomičnog prosjeka ostvareno je uz pomoć Statsmodels paketa. Statsmodels je Python paket koji korisnicima omogućuje istraživanje podataka, procjenu statističkih modela i provođenje statističkih testova.[26] Za prognoziranje ovom metodom bitno je napomenuti kako se radi o sezonalnim podacima koji imaju utjecaj vanjskih parametara što ovaj model čini sezonalnim integriranim autoregresivnim

pomičnim prosjekom s vanjskim varijablama (SARIMAX, eng. *Seasonal Autoregressive Integrated Moving Average with external variables*). Primjena modela prikazana je ispisom programskog kôda Programski kôd 4.

```
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)

sarimax = sm.tsa.statespace.SARIMAX(s1i1['week_sale'],
                                     exog=s1i1.drop('week_sale', axis=1),
                                     trend='t', freq='W-MON',
                                     enforce_invertibility=False,
                                     order=(0, 0, 1),
                                     seasonal_order=(0, 0, 1, 6))

results = sarimax.fit(maxiter=500)
fcst = results.predict(start='2017-01-01', end='2017-12-31',
                       exog=df_test.drop('week_sale', axis=1))

results = pd.DataFrame()
results['Predicted Values'] = fcst
results['True Values'] = df_test['week_sale']
results['week'] = df_test['week'].tolist()
results['month'] = df_test['month'].tolist()
```

Programski kôd 4

Sve navedeni parametri u prošlom poglavlju u ovoj metodi se koriste kao vanjske varijable, odnosno korištene su za parametar `exog`. Parametru `freq` pridodijeljena je vrijednost 'W-MON' što ukazuje na to da podaci imaju tjednu frekvenciju i to s ponedjeljkom kao početkom tjedna. Parametri `trend`, `order` i `seasonal_order` određeni su uz pomoć pretraživanja mreže parametara autora Jason Brownleea.[27] Pretraga mreže parametara prikazana je isječkom programskog kôda Programski kôd 5.

```
def sarima_forecast(history, config):
    order, sorder, trend = config
    model = SARIMAX(history, order=order, seasonal_order=sorder, trend=trend,
                    enforce_stationarity=False, enforce_invertibility=False)
    model_fit = model.fit(dispatch=False)
    yhat = model_fit.predict(len(history), len(history))
    return yhat[0]

def measure_rmse(actual, predicted):
    return sqrt(mean_squared_error(actual, predicted))

def train_test_split(data, n_test):
    return data[:-n_test], data[-n_test:]

def walk_forward_validation(data, n_test, cfg):
    predictions = list()
```

```

train, test = train_test_split(data, n_test)
history = [x for x in train]
for i in range(len(test)):
    yhat = sarima_forecast(history, cfg)
    predictions.append(yhat)
    history.append(test[i])
error = measure_rmse(test, predictions)
return error

def score_model(data, n_test, cfg, debug=False):
    result = None
    key = str(cfg)
    if debug:
        result = walk_forward_validation(data, n_test, cfg)
    else:
        try:
            with catch_warnings():
                filterwarnings("ignore")
                result = walk_forward_validation(data, n_test, cfg)
        except:
            error = None
    if result is not None:
        print(' > Model[%s] %.3f' % (key, result))
    return (key, result)

def grid_search(data, cfg_list, n_test, parallel=True):
    scores = None
    if parallel:
        executor = Parallel(n_jobs=cpu_count(), backend='multiprocessing')
        tasks = (delayed(score_model)(data, n_test, cfg) for cfg in cfg_list)
        scores = executor(tasks)
    else:
        scores = [score_model(data, n_test, cfg) for cfg in cfg_list]
    scores = [r for r in scores if r[2] != None]
    scores.sort(key=lambda tup: tup[2])
    return scores

def sarima_configs(seasonal=[0]):
    models = list()
    p_params = [0, 1, 2]
    d_params = [0, 1]
    q_params = [0, 1, 2]
    t_params = ['n', 'c', 't', 'ct']
    P_params = [0, 1, 2]
    D_params = [0, 1]
    Q_params = [0, 1, 2]
    m_params = seasonal
    for p in p_params:
        for d in d_params:
            for q in q_params:
                for t in t_params:
                    for P in P_params:
                        for D in D_params:
                            for Q in Q_params:
                                for m in m_params:
                                    cfg = [(p,d,q), (P,D,Q,m), t]
                                    models.append(cfg)

```

```

return models

if __name__ == '__main__':
    series = read_csv('../data/data.csv', header=0, index_col=0)
    data = series['week_sale'].values
    print(data.shape)
    n_test = 12
    cfg_list = sarima_configs(seasonal=[0,6,12])
    scores = grid_search(data, cfg_list, n_test)
    print('done')
    for cfg, error in scores[:3]:
        print(cfg, error)

```

Programski kôd 5, [27]

Ispis navedene pretrage mreže je `Model[[ (0, 0, 1), (0, 0, 1, 12), 'c' ]]`. Parametar 'c' predstavlja konstantni trend. Navedeni ispis pretrage može se pronaći i u zapisu (0, 0, 1), (0, 0, 1) 12, odnosno (p, d, q) (P, D, Q) m pri čemu je:

- p – broj autoregresivnosti;
- d – stupanj razlikovanja;
- q – broj termina pomičnih prosjeka
- (P, D, Q) – predstavlja (p, d, q) za sezonski dio vremenske serije;
- m – broj razdoblja u svakoj sezoni.

6.2.2. Primjena metode neuronske mreže

Primjena modela neuronskih mreža ostvarena je kroz korištenje Keras te TensorFlow knjižnica. TensorFlow je knjižnica otvorenog kôda za strojno i duboko učenje koje za računanje koristi multidimenzionalne matrice nazvane tenzori. TensorFlow jezgra pruža podršku za Keras knjižnicu. TensorFlow knjižnica sadrži unaprijed određene funkcije koje olakšavaju postavljanje mreže, pripremu podataka i izvršavanje operacija u pripremi učenja mreže. TensorFlow

U ispisu programskog kôda Programski kôd 6 prikazana je primjena modela neuronske mreže. Pri korištenju modela neuronske mreže prvotno je potrebno normalizirati raspone neovisnih varijabli uz pomoć `MinMaxScaler` klase. Zatim kreiramo mrežu koja se sastoji od jednog ulaznog, tri skrivena i jednog izlaznog sloja. Ulazni sloj dimenzije je 500, a skriveni dimenzija 250, 125 i 25 pri čemu dimenzija predstavlja broj neurona u pojedinom sloju mreže. Izlazni sloj veličine je 1 koje je ujedno i rješenje. Svi slojevi, ulazni, skriveni i izlazni međusobno su

povezani. Aktivacijska funkcija uvodi nelinearnost u mrežu. Iako linearne aktivacijske funkcije postoje one ne mogu biti korištene za bolje učenje mreže. Druge aktivacijske funkcije koje se koriste su sigmoidalna, step aktivacijska funkcija i ispravljenja linearna jedinica (*ReLU*, eng. *Rectified Linear Unit*). ReLu aktivacijska funkcija najčešće je korištena funkcija u modelu neuronskih mreža, a njena funkcija glasi:

$$f(x) = \max(0, x).$$
6.1

Ova se aktivacijska funkcija uvijek koristi u skrivenim slojevima mreže. Kada su ulazne vrijednosti negativne njen gradijent je 0. Negativne vrijednosti postaju 'mrtve' te se stoga neuron ne aktivira. Zahvaljujući gradijentu 0 samo su pozitivne vrijednosti one koje čine neuron aktivnim što olakšava izvršavanje algoritma.[28]

Parametar optimizator pruža mogućnost kontroliranja brzine učenja, a korišteni adam algoritam prilagođava brzinu učenja tijekom treninga. Za funkciju gubitka korištena je prosječna kvadratna razlika (*MSE*, eng. *Mean Squared Error*). MSE uzima prosječnu kvadratnu razliku između stvarnih i predviđenih vrijednosti. Što je rezultat MSE bliži 0, model je bolje izveden.

```
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
eval_X = scaler.transform(eval_X)

model = Sequential()
model.add(Dense(500, 'relu'))
model.add(Dense(250, 'relu'))
model.add(Dense(125, 'relu'))
model.add(Dense(25, 'relu'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

model.fit(x=X_train, y=y_train, epochs=1000)

loss_df = pd.DataFrame(model.history.history)
loss_df.plot()
test_predictions = model.predict(eval_X)
```

Programski kôd 6

6.2.3. Primjena metode slučajne šume

Metoda slučajne šume ostvarena je uz pomoć sklearn knjižnice modula `sklearn.ensemble` i razreda `RandomForestRegressor`. Primjena modela vrlo je jednostavna i prikazana ispisom programskog kôda Programski kôd 7.

```
rfr = RandomForestRegressor()
rfr.fit(X_train, y_train)
predictions = rfr.predict(X_test)
predictions = rfr.predict(test.drop('week_sale', axis=1))
```

Programski kôd 7

6.3. Prikaz rezultata

Evaluacija konačnih rezultata prethodno navedenih modela provedena je kroz verifikacijske metode opisane u nastavku.

EVS (*eng. Explained Variance Score*) metrika je za izračunavanje omjera varijance pogreške i varijance pravih vrijednosti, odnosno ovaj rezultat predstavlja koliko dobro model može objasniti varijacije skupa podataka. Njegove se vrijednosti kreću u intervalu $[0, 1]$, a visina koeficijenta govori o reprezentativnosti, model je reprezentativniji što je vrijednost bliža vrijednosti 1. Računa se prema formuli:

$$EVS = \frac{Var\{y - \hat{y}\}}{Var\{y\}}, \quad 6.2$$

gdje y označava stvarnu vrijednost, a $\hat{y}$ predviđenu vrijednost.[29]

MAD (*eng. Mean Absolute Deviation*) srednje je apsolutno odstupanje koje izražava prosječno apsolutno odstupanje pojedinačnih vrijednosti od srednje vrijednosti. Računa se prema formuli:

$$MAD = \frac{\sum_{i=1}^n |y_i - \bar{y}|}{n}, \quad 6.3$$

pri čemu je $\bar{y}$ srednja vrijednost podataka, y_i stvarna vrijednost i -tog podatka, a n broj promatranja u podacima.[30]

MAE (*eng. Mean Absolute Error*) srednja je apsolutna pogreška koja mjeri magnitudu pogrešaka, drugim riječima, predstavlja mjeru ukupne točnosti. Formula MAE glasi:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad 6.4$$

gdje je y_i stvarna vrijednost i -tog uzorka, a $\hat{y}_i$ predviđena vrijednost odgovarajućeg uzorka.[31]

ME (*eng. Maximum Error*) označava najgori scenarij pogreške između predviđenih i stvarnih vrijednosti.

MSE (*eng. Mean Squared Error*) srednja je kvadratna pogreška koja mjeri prosjek kvadrata pogrešaka. MSE se računa prema:

$$MSE = \frac{1}{n_{uzorak}} \sum_{i=0}^{n_{uzorak}-1} (y_i - \hat{y}_i)^2, \quad 6.5$$

gdje y_i predviđena vrijednost i -tog uzorka, a $\hat{y}_i$ odgovarajuća stvarna vrijednost istog uzorka.[29]

R^2 (*eng. R Squared Score*) je koeficijent determinacije pokazuje u kojoj mjeri regresijski model pokriva odstupanja. Vrijednosti metrike se kreću u intervalu $[0, 1]$, a visina koeficijenta govori o reprezentativnosti što prikazuje tablica 5. Model je reprezentativniji što je vrijednost bliža vrijednosti 1. Izračun ove metrike vrši se prema sljedećoj formuli:

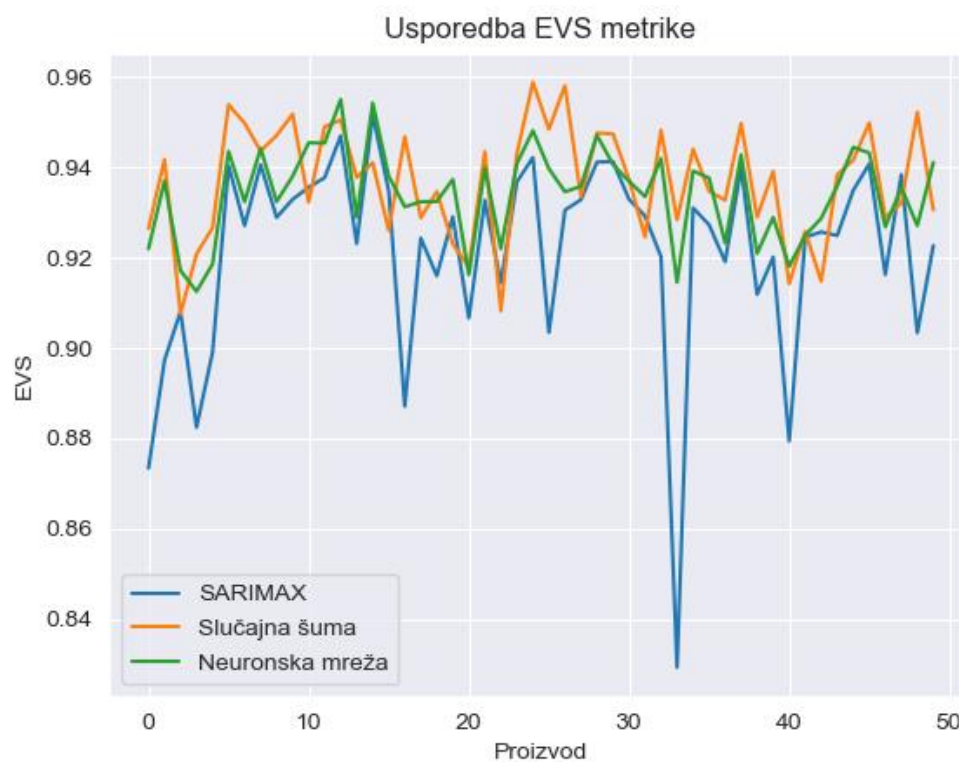
$$R^2 = \frac{\sum_{i=0}^n Y}{\sum_{i=0}^n \hat{y}}, \quad 6.6$$

pri čemu je Y koeficijent varijacije zavisne varijable, a $\hat{y}$ predviđena vrijednost.[32]

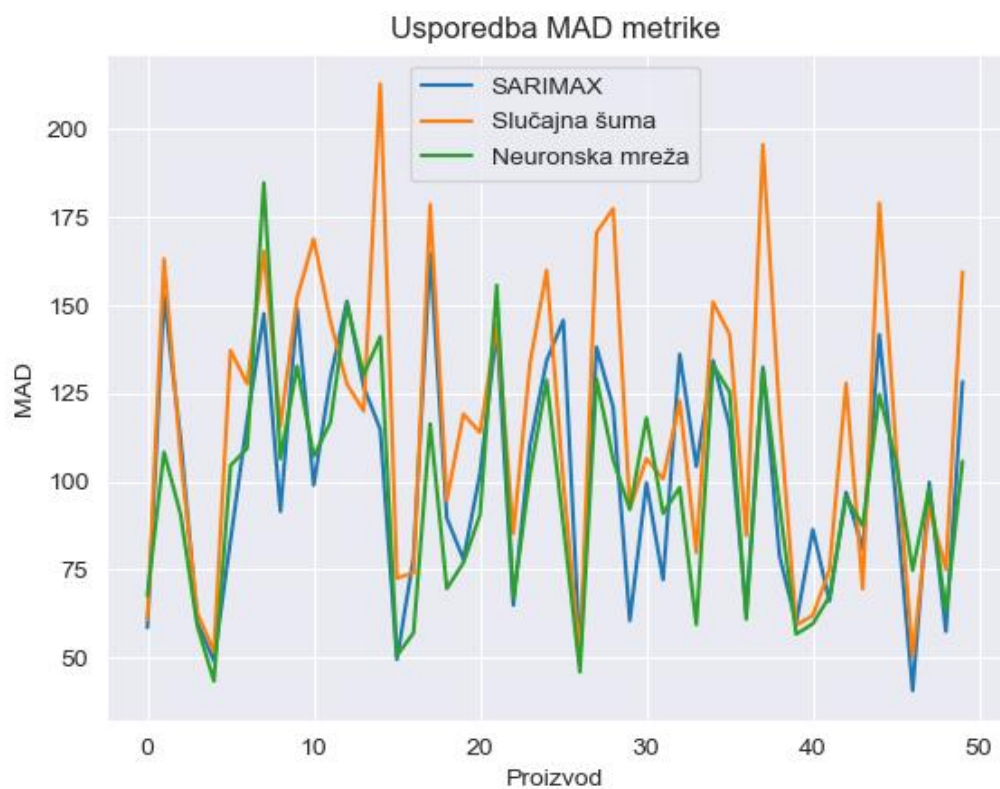
Tablica 5 Tumačenje jačine veze koeficijenta determinacije

R^2	Tumačenje
0	Odsutnost veze
0,00 – 0,25	Slaba veza
0,25 – 0,64	Veza srednje jakosti
0,64 – 1	Čvrsta veza
1	Potpuna veza

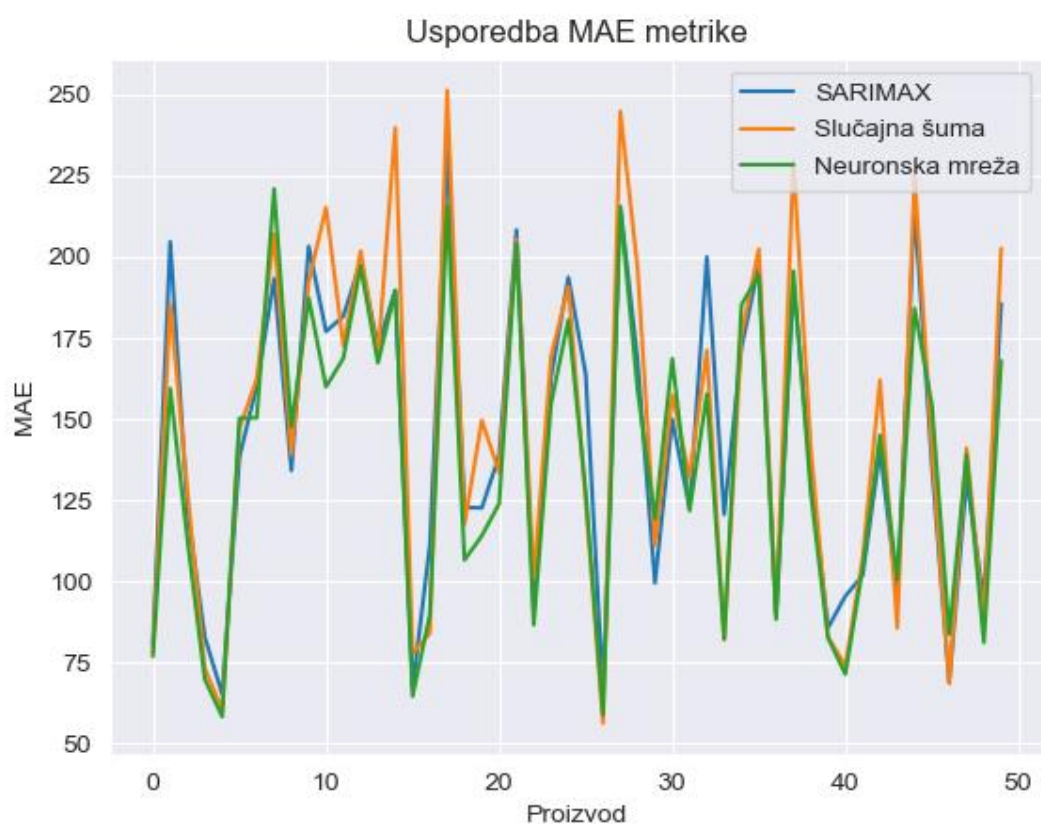
Slike 12 - 17 prikazuju usporedbe pojedinih korištenih metoda i prethodno navedenih metrika za svaki proizvod. Na vertikalnoj osi dane su vrijednosti metrike, a na horizontalnoj osi naznačen je proizvod.



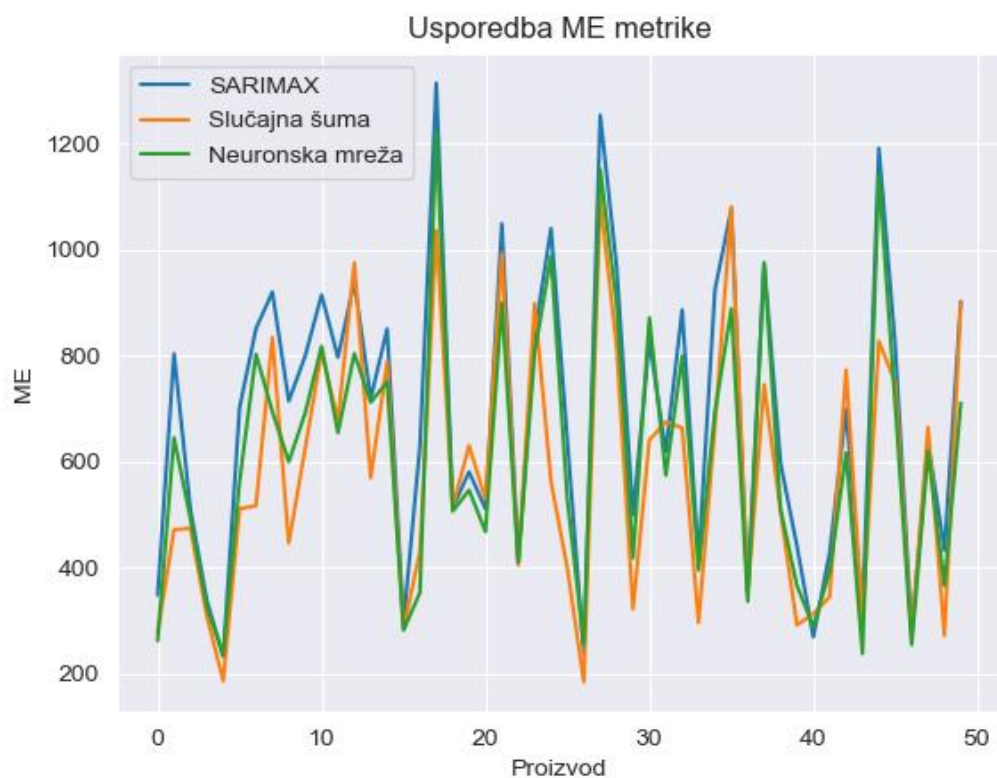
Slika 12 Usporedba EVS metrike



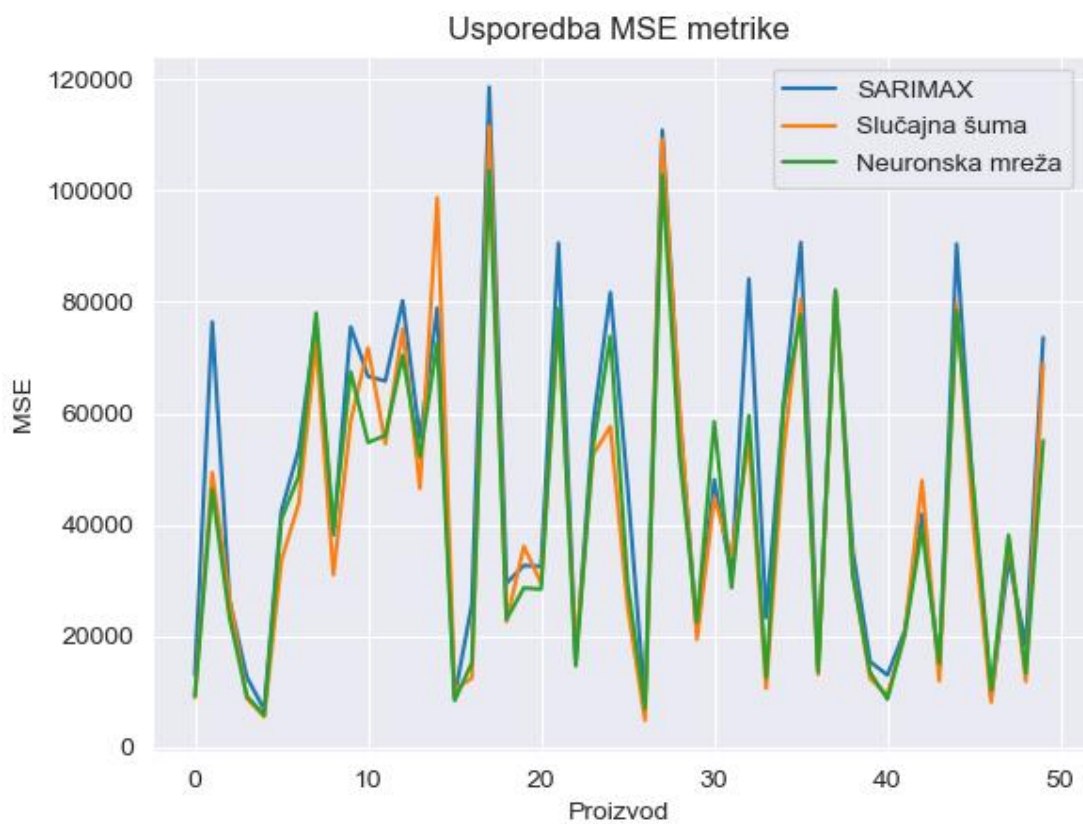
Slika 13 Usporedba MAD metrike



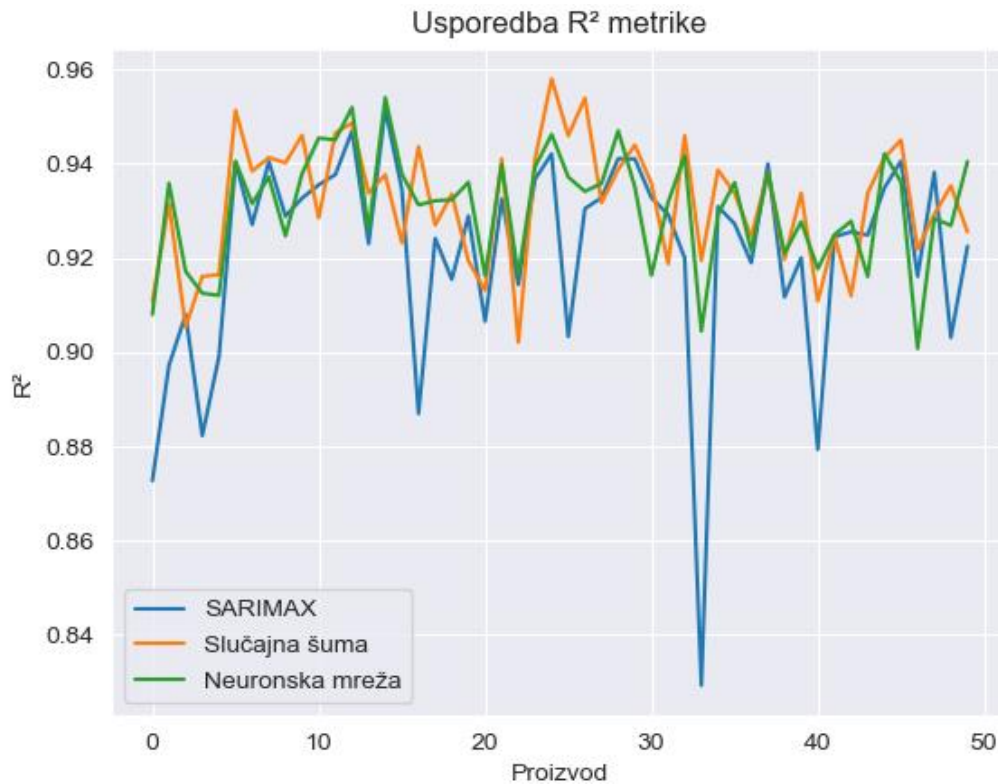
Slika 14 Usporedba MAE metrike



Slika 15 Usporedba ME metrike

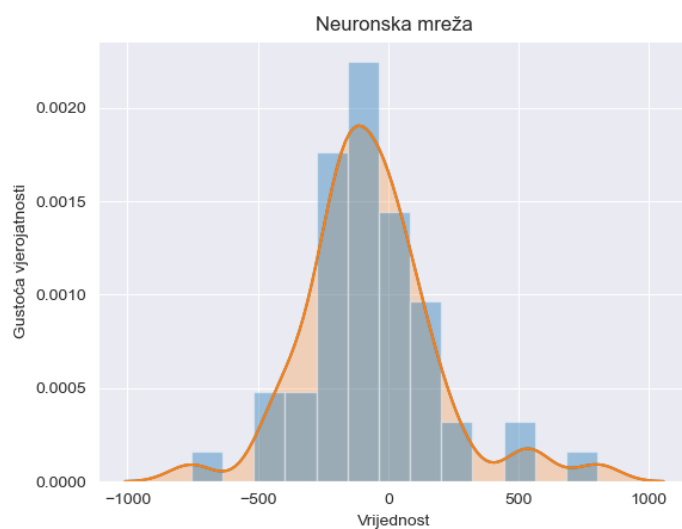
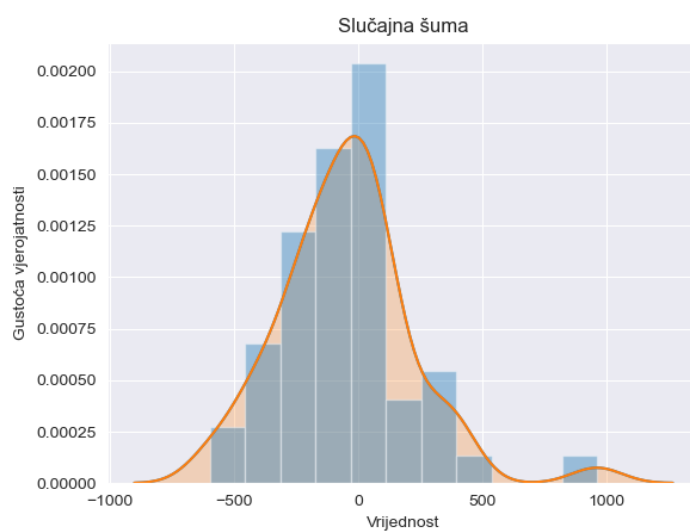
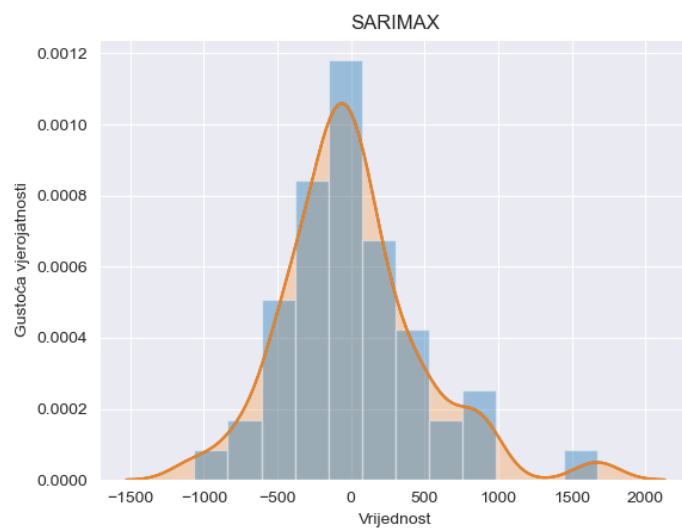


Slika 16 Usporedba MSE metrike



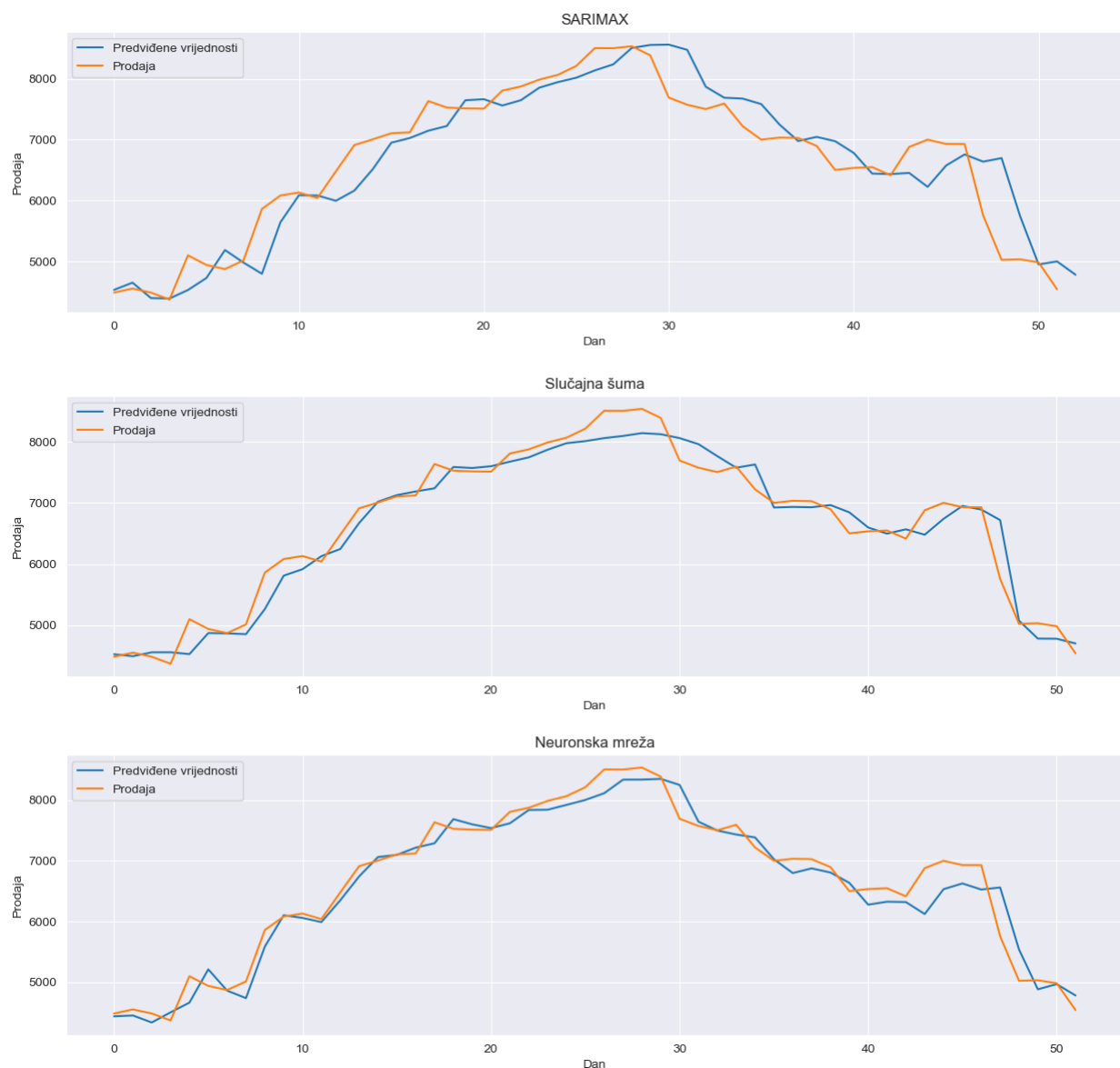
Slika 17 Usporedba R² metrike

Na grafu slike 18 prikazana je razdioba pogreške. Vrijednosti x osi predstavljaju vrijednost pogreške, a visina stupca na y osi prikazuje učestalost ponavljanja. Narančasti dio grafa pokazuje normalizirane vrijednosti. U modelu neuronske mreže vidljivo je kako model postiže pogreške u krajnostima, odnosno postiže vrlo velike pogreške, i negativne, ali i pozitivne što smanjuje generalnu točnost metode. U modelima slučajne šume i SARIMAX ove ekstremne pogreške postoje samo u pozitivnom dijelu grafa.



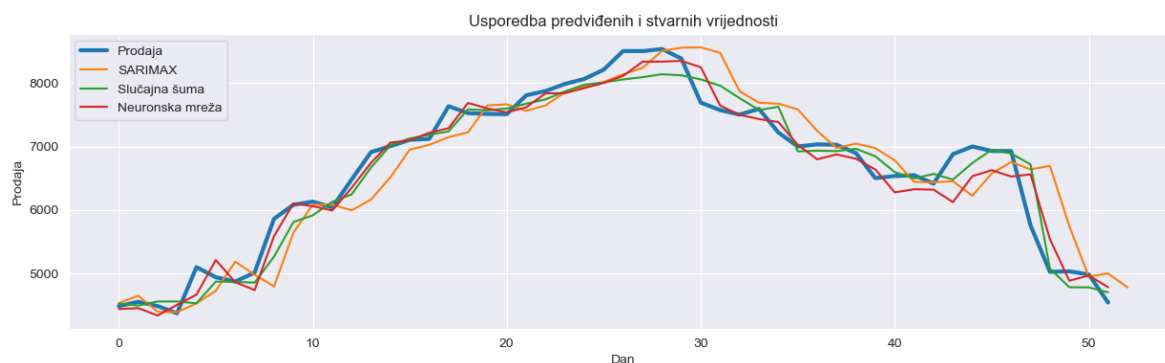
Slika 18 Raspodjela pogreške

Slika 19 u konačnici predstavlja usporedbe predviđenih i stvarnih vrijednosti potražnje za period od jedne godine. Iz grafa je vidljivo kako model SARIMAX oblikom izuzetno dobro prati krivulju, no zbog pomičnog vremenskog prozora koji koristi javlja se kašnjenje. Model slučajne šume generalno vrlo dobro prati krivulju osim u periodu od dvadesetog do trideset petoga tjedna gdje postoje povećana odstupanja. Model neuronske mreže u prvom dijelu predviđanja, do dvadesetoga tjedna iznimno dobro prati krivulju, ali nakon toga postiže predviđanja uz određeno kašnjenje.



Slika 19 Usporedba stvarnih i predviđenih vrijednosti

Slika 20 jasnija je reprezentacija slike 19 i predstavlja usporedbu stvarnih vrijednosti prodaje i predviđenih vrijednosti za metode SARIMAX, slučajna šuma i neuronska mreža.



Slika 20 Usporedba stvarnih i predviđenih vrijednosti svih metoda

Tablica 6 prikazuje vrijednosti pojedine metrike za svaku od korištenih metoda za zadani proizvod 13. U tablici 6 su označene najbolje vrijednosti pojedine metrike. Vidljivo je kako model neuronske mreže postiže najbolje rezultate u gotovo svih metrika: EVS, MAE, ME, MSE, R^2 . Model slučajne šume postiže najbolji rezultat za metriku MAD dok model SARIMAX ne postiže niti jedan najbolji rezultat.

Tablica 6 Vrijednosti metrika

SARIMAX					
EVS	MAD	MAE	ME	MSE	R^2
0.947	151.17	198.49	938.97	80235.68	0.947
Slučajna šuma					
EVS	MAD	MAE	ME	MSE	R^2
0.950	127.63	201.83	974.51	75194.25	0.949
Neuronska mreža					
EVS	MAD	MAE	ME	MSE	R^2
0.955	150.42	197.13	803.48	70390.27	0.952

7. Zaključak

Prognoza potražnje ključan je zadatak pri optimizaciji opskrbnog lanca. Dobro osmišljena prognoza potražnje može pogodovati svakome od subjekata opskrbnog lanca pa tako i prodaji. Uz dobro osmišljenu prognozu potražnje mogu se riješiti mnogi od postojećih problema u sektoru prodaje poput stvaranja nekurentnih zaliha, isteka roka trajanja proizvoda, nedostupnosti proizvoda i sl.

U ovom su radu predviđena je potražnja uz pomoć multivarijatnih sezonskih vremenskih serija. Za klasičnu metodu prognoze korištena je metoda sezonalnog autoregresivnog integriranog pomičnog prosjeka, a za metode strojnog učenja metoda slučajne šume i neuronska mreža. Rezultati predstavljenog istraživanja pokazuju kako metode strojnog učenja postižu bolje rezultate, odnosno metoda neuronske mreže postiže najbolje rezultate. Točnosti navedenih metoda gotovo su podjednake te stoga možemo zaključiti kako metode strojnog učenja, ali i klasične metode postižu iznimno dobre rezultate. Iako metoda neuronske mreže postiže najveću točnost u prognozi njena implementacija je nešto zahtjevnija od implementacije drugih metoda te zahtijeva veću potrebnu računalnu moć dok metoda slučajne šume zahtijeva vrlo malo truda u implementaciji, a postiže približno dobre rezultate. Metoda slučajne šume osim lake implementacije ističe se brzinom izvršavanja i manjom potrebnom računalnom moći. Metoda SARIMAX postiže najlošije rezultate, a ne ide joj u prilog niti podugo vrijeme izvršavanja, potrebna računalna moć te zahtjevnija implementacija uz dobro poznavanje materije u području prognoze potražnje. Kako se ovaj rad temelji na tjednoj prognozi potražnje postoji prostor za poboljšanjem u budućim radovima.

Potencijalna buduća istraživanja mogla bi prognozirati dnevnu umjesto tjedne potražnje koja je iznimno važna na razini pojedinog dućana. Uz ovakav način prognoziranja postoji mogućnost uvođenja strategija u poduzeće koje bi omogućilo zadovoljenje potražnje u što kraćem periodu. Više ulaznih parametara pogoduje uspostavljanju jačih korelacijskih veza pa je tako u dnevnoj prognozi potražnje moguće upotrijebiti dodatne parametre poput označavanja pojedinih dana (npr. Uskrs, Božić, međunarodni praznih rada i sl.), označavanja dana u tjednu zbog mogućnosti postojanja izražene tjedne sezonalnosti i sličnih parametara.

Literatura

- [1] Carbonneau, R., Vahidov, R., Laframboise, K., Application of machine learning techniques for supply chain demand forecasting, Elsevier, vol. 184(3), 1140-1154, 2008.
- [2] Croxton, K. L., Lambert, D. M., Garcia-Dastugue, S. J., Rogers, D.S., The Demand Management Process, u Lambert, D. M., Supply Chain Management: Processes, Partnerships, Performance, 3 rd Edition, Supply Chain Management Institute, Sarasota, Florida, 2008.
- [3] Thomopoulos, N. T., Demand Forecasting for Inventory Control, Springer International Publishing, Switzerland, 2015.
- [4] Sankaran, G., Sasso, F., Kepczynski, R., Chiaraviglio, A., Improving Forecasts with Integrated Business Planning: From Short-Term to Long-Term Demand Planning Enabled by SAP IBP, Cham Springer International Publishing, 2019.
- [5] Chopra, S., Supply chain management: Strategy, planning, and operation, Harlow: Pearson Education Limited, 2019.
- [6] Chopra, S. i Meindl, P., idem, 182-196.; Bozarth, C. C. i Handfield, R. B., idem, str. 247-272.; Hill, A. V., idem, str. 129.; Bowersox, D., Closs, D. i Cooper, B., idem, str. 147-150.
- [7] Dujak, D., Uloga maloprodaje u upravljanju opskrbnim lancem, Disertacija, Sveučilište Josipa Jurja Strossmayera u Osijeku, Ekonomski fakultet u Osijeku, 2012
- [8] Dugic, M., Zaulich, D., Forecasting System at IKEA Jönköping, Jönköping International Business School, Jönköping University, Jönköping, 2011.
- [9] Simchi-Levi, D, Kaminsky, P., Simchi-Levi, E., Designing & Managing the Supply Chain: Concepts, Strategies, and Case Studies, New York, McGraw-Hill/Irwin, 2003.
- [10] Fakultet prometnih znanosti, Distribucijska logistika II (materijali), Zagreb, 2019.
- [11] Chapman, S. N., Arnold, J. R. T., Gatewood, A. K., Clive, L. M., Introduction to Materials Management, Harlow, Essex: Pearson Education Limited, 2017.
- [12] Olhager, J., Produktionsekonomi, Lund : Studentlitteratur, 2000.
- [13] Hyndman, R.J., Athanasopoulos, G., Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia, 2018

- [14] Rose, D., Artificial Intelligence for Business, Financial Times Prentice Hall, 2020.
- [15] Jung, J.-K., Patnam, M., Ter-Martirosyan, A., An Algorithmic Crystal Ball: Forecasts-based on Machine Learning, IMF Working Papers, v18 n230, 2018
- [16] Geron, A., Hands-on Machine Learning with Scikit-Learn & TensorFlow : Concepts, Tools, and Techniques to build Intelligent Systems, O'Reilly Media, 2017.
- [17] Bishop, C. M., Pattern Recognition and Machine Learning, Springer-Verlag New York, 2016.
- [18] Alpaydin, E., Introduction to Machine Learning, MIT Press, 2020.
- [19] Carbonneau, R., Vahidov, R., Laframboise, K., Machine learning-Based Demand forecasting in supply chains, International Journal of Intelligent Information Technologies, v3 n4 (200710): 40-57
- [20] Abdi, H., Edelman, B., Valentin, D., Neural networks, Thousand Oaks, CA: Sage Publ, 1999.
- [21] Nielsen, M., Neural network and deep learning, Determination Press, 2015
- [22] Breiman, L., Cutler, A., Random Forest, http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#intro, pristupljeno 8.9.2020.
- [23] Kuhlman, D., A Python Book: Beginning Python, Advanced Python, and Python Exercises, Platypus Global Media, 2011
- [24] Chen, D. Y., Pandas for Everyone : Python Data Analysis. Boston: Addison-Wesley, 2017
- [25] McKinney, W., Python for Data Analysis : Data Wrangling with Pandas, NumPy, and Ipython, O'Reilly, 2017.
- [26] <https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html>, Pristupljeno: 8.9.2020.
- [27] Jason Brownlee, Deep Learning with Time Series Forecasting, Machine Learning Mastery, Available from <https://machinelearningmastery.com/machine-learning-with-python/> , Pristupljeno: 8.9.2020.
- [28] Iyer, B., Deshpande, P. S., Sharma, S. C., Shiurkar, U., Computing in Engineering and Technology, Proceedings of ICCET 2019
- [29] Garofalakis, M., Gehrke, J., Rastogi, R., Data Stream Management: Processing High-Speed Data Streams, Springer, 2016

- [30] CFA Institute, Quantitative Investment Analysis, John Wiley & Sons, 2020
- [31] Lu, Z., Zhou, S., Integration of Large Scale Wind Energy with Electrical Power Systems in China, John Wiley & Sons, 2018
- [32] Information Resources Management Association, Banking, Finance, and Accounting: Concepts, Methodologies, Tools, and Applications, IGI Global; 1st Edition, 2014

Popis slika

SLIKA 1 VRSTE PLANIRANJA, IZVOR:[5]	5
SLIKA 2 PRAVILNA LINEARNA UZLAZNA PROMJENA U PODACIMA POTRAŽNJE, [10]	10
SLIKA 3 SEZONALNA PROMJENA PODATAKA POTRAŽNJE, [10]	11
SLIKA 4 CIKLIČKA PROMJENA PODATAKA POTRAŽNJE, [10]	11
SLIKA 5 SLUČAJNE PROMJENE PODATAKA POTRAŽNJE, [10]	12
SLIKA 6 BIOLOŠKI NEURON IZVOR: [BOLF, JERBIĆ (2006)]	28
SLIKA 7 OSNOVNA ARHITEKTURA NEURONSKE MREŽE, [21]	28
SLIKA 8 PRIKAZ NEURONA	29
SLIKA 9 PRIKAZ NEURONA S TEŽINAMA	29
SLIKA 10 PRIKAZ NEURONA S TEŽINSKIM KOEFICIJENTIMA I KOEFICIJENTIMA PRISTRANOSTI	30
SLIKA 11 TJEDNA PRODAJA PROIZVODA	33
SLIKA 12 USPOREDBA EVS METRIKE	43
SLIKA 13 USPOREDBA MAD METRIKE	44
SLIKA 14 USPOREDBA MAE METRIKE	44
SLIKA 15 USPOREDBA ME METRIKE	45
SLIKA 16 USPOREDBA MSE METRIKE	45
SLIKA 17 USPOREDBA R^2 METRIKE	46
SLIKA 18 RASPODJELA POGREŠKE	47
SLIKA 19 USPOREDBA STVARNIH I PREDVIĐENIH VRIJEDNOSTI	48
SLIKA 20 USPOREDBA STVARNIH I PREDVIĐENIH VRIJEDNOSTI SVIH METODA	49

Popis tablica

TABLICA 1 KATEGORIZACIJA VRSTE PROGNOZE NA TEMELJU VREMENSKE JEDINICE PODATAKA, IZVOR: [3].....	4
TABLICA 2 TEHNIKE STROJNOG UČENJA, IZVOR: [15]	24
TABLICA 3 POČETNI SKUP PODATAKA	31
TABLICA 4 SKUP PODATAKA PROIZVODA 13 NAKON GRUPIRANJA	32
TABLICA 5 TUMAČENJE JAČINE VEZE KOEFICIJENTA DETERMINACIJE	43
TABLICA 6 VRIJEDNOSTI METRIKA	49

Popis grafikona

GRAFIKON 1 KATEGORIZACIJA TEHNIKA PROGNOZE POTRAŽNJE, IZVOR: [7].....	8
---	---



Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj _____ diplomski rad
isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na
objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz
necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj
visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu _____ diplomskog rada
pod naslovom **Proгноza i planiranje potražnje uz pomoć strojnog učenja**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom
repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, 10.9.2020

Marija Matijević