SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1983

Prognoza vremenskih serija korištenjem radnog okvira Apache Spark

Dario Bošnjak

Zagreb, lipanj 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA DIPLOMSKI RAD PROFILA

Zagreb, 8. ožujka 2019.

DIPLOMSKI ZADATAK br. 1983

Pristupnik: **Dario Bošnjak (0036487554)**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Prognoza vremenskih serija korištenjem radnog okvira Apache Spark**

Opis zadatka:

Apache Spark je radni okvir otvorenog koda za raspodijeljenu obradu podataka. Vaš zadatak je detaljno opisati i usporediti podržane metode strojnog učenja za prognoziranje vremenskih serija u radnom okviru Apache Spark. Na odabranom studijskom slučaju stvarnih podataka predložite i implementirajte model, eksperimentalno ga evaluirajte te komentirajte dobivene rezultate i predložite moguća poboljšanja.
Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Zadatak uručen pristupniku: 15. ožujka 2019.
Rok za predaju rada: 28. lipnja 2019.

Mentor:

Izv. prof. dr. sc. Krešimir Pripužić

Djelovođa:

Izv. prof. dr. sc. Tomislav Hrkać

Predsjednik odbora za diplomski rad profila:

Doc. dr. sc. Marko Čupić

Zahvaljujem mentoru izv. prof. dr. sc. Krešimiru Pripužiću na podršci i pomoći tijekom studija. Zahvaljujem kolegi Ivanu Fabijaniću koji je savjetima pomogao u ostvarenju ovog rada. Također zahvaljujem svojoj obitelji i prijateljima na podršci i motivaciji koju su mi pružili tijekom studija.

SADRŽAJ

1. Uvod

Prognoziranje vremenskih serija povezuje se uz područja ekonometrije, meteorologije, hidrometrije itd. U navedenim, ali i drugim područjima primjene prognoziranje budućnosti smanjuje neizvjesnost što omogućuje donošenje boljih odluka koje se očituju u povećanju prihoda, većoj razini sigurnosti ili sprječavanju katastrofe. Kako bi se dobila kvalitetna prognoza najprije je potrebno analizirati vremensku seriju od interesa i opisati njezina statistička svojstva čime se dobiva dublji uvid u promatrani proces.

Za prognoziranje vremenske serije moguće je koristiti uvriježene klasične statističke metode (autoregresijski model, model pomičnih prosjeka, integrirani autoregresijski model pomičnih prosjeka, eksponencijalno zaglađivanje i model vektorske autoregresije), ali i metode strojnog učenja (linearna regresija, slučajne šume i gradijentno ojačana stabla). Rad opisuje i uspoređuje tipične predstavnike oba pristupa.

Radni okvir Apache Spark namijenjen je za raspodijeljenu obradu podataka te nudi knjižnicu za strojno učenje. Cilj rada je opisati i primijeniti knjižnicu te ocijeniti prikladnost radnog okvira za prognoziranje vremenskih serija od koraka pripreme podataka do samog prognoziranja.

Nastavak rada organiziran je u sljedeće cjeline. Poglavlje 2 bavi se opisom i analizom vremenskih serija. Prognoziranje pomoću klasičnih statističkih metoda i metoda strojnog učenja objašnjeno je poglavljem 3. Radni okvir Apache Spark i njegova knjižnica za strojno učenje objašnjeni su poglavljem 4, dok poglavlje 5 opisuje ostvarenje rješenja problema prognoziranja vremenskih serija u radnom okviru Apache Spark na primjeru podataka o proizvodnji mlijeka. Poglavlje također uspoređuje performanse klasičnih statističkih metoda s metodama strojnog učenja.

2. Vremenska serija

2.1. Osnovne definicije

Skup slijedno generiranih opažanja naziva se vremenskom serijom (Box et al., 2015). Ovisno o tome jesu li opažanja u vremenskoj seriji kontinuirana ili diskretna razlikuju se kontinuirana i diskretna vremenska serija (Box et al., 2015), no opažanja ne moraju biti isključivo brojčana, nego i simboličke vrijednosti (slova i riječi) kao što je i pokazano u (Lin et al., 2003). Vremenska serija skalarnih opažanja (npr. temperatura) naziva se univarijatna vremenska serija, dok se vremenska serija s više vrsta opažanja (npr. temperatura i vlaga) naziva multivarijatna vremenska serija. Ovisno vremenskom razmaku slijednih opažanja razlikuju se pravilna od nepravilne vremenske serije.

Primjeri vremenskih serija su senzorska očitanja temperature zraka, seizmogram, brzina otkucaja srca, cijene dionica itd. Vremenski razmak između opažanja je proizvoljan i ovisi o procesu koji generira vremensku seriju pa tako vremenska serija temperature zraka u većini primjena ne zahtijeva jednaku učestalost opažanja kao vremenska serije brzine otkucaja srca. Kod vremenskih serija s brojčanim vrijednostima vremenski razmak između opažanja moguće je povećati agregacijom – npr. izračun dnevnih prosjeka iz satnih podataka.

Iz navedenog je vidljivo da su vremenske serije prisutne u raznim područjima primjene zbog čega su do danas razvijeni brojni postupci analize i prognoziranja vremenskih serija kako bi shvatili uzorke u njima te ih iskoristili za predviđanje budućih vrijednosti.

2.2. Analiza vremenske serije

Inherentna značajka vremenskih serija jest zavisnost susjednih opažanja. Analiza vremenskih serija obuhvaća tehnike za analizu te ovisnosti. (Box et al., 2015)

2.2.1. Dekompozicija vremenske serije

Dekompozicija

Vremenska serija dekomponira se u četiri komponente: trend (T_t), sezonalnost (S_t), cikličnost (C_t) i nasumičnost (N_t), sve gledano u trenutku t. Moguća je aditivna (jednadžba 2.1) i multiplikativna dekompozicija (jednadžba 2.2):

$$y_t = T_t + S_t + C_t + N_t \tag{2.1}$$

$$y_t = T_t \cdot S_t \cdot C_t \cdot N_t. \tag{2.2}$$

Aditivna dekompozicija koristi se kada komponente trenda, sezonalnosti i cikličnosti ne mijenjaju vrijednost ovisno o srednjoj vrijednosti vremenske serije. U slučaju proporcionalne promjene navedenih komponenti u ovisnosti o srednjoj vrijednosti koristi se multiplikativna dekompozicija. (Hyndman i Athanasopoulos, 2018)

Komponente vremenske serije

Komponenta trenda odnosi se na dugoročno kretanje vrijednosti vremenske serije, a može biti rastuća ili padajuća. Trend se može odrediti pomoću pomičnog prosjeka.

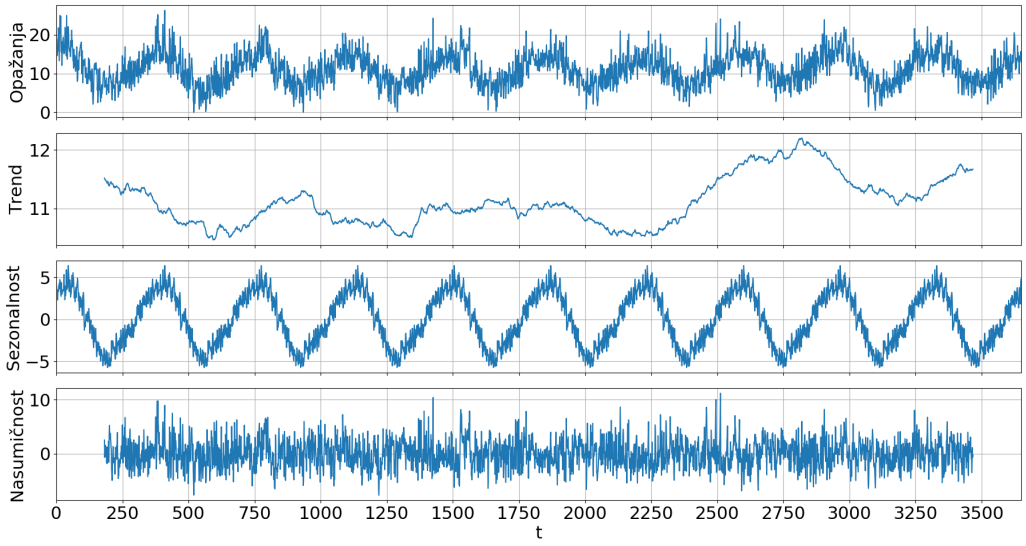
Komponenta sezonalnosti odnosi se na promjene unutar vremenske serije koje se događaju u pravilnim razmacima. Primjer sezonalnosti može biti dnevna sezonalnost u vremenskoj seriji koja bilježi broj automobila koji je prošao cestom u proteklom satu radnoga dana. Vrlo je vjerojatna pojava gužve u jutarnjim i popodnevnim satima svakoga radnog dana. U toj vremenskoj seriji pojavili bi se pravilni dvadesetčetverosatni razmaci između maksimalnih jutarnjih, kao i popodnevnih vrijednosti. U podacima može postojati i višestruka sezonalnost (npr. satna i dnevna sezonalnost).

Komponenta cikličnosti slična je sezonalnoj komponenti, ali za razliku od sezonalnosti, razmaci između ekstrema vremenske serije ne moraju biti pravilni.

Komponentu nasumičnosti čine nepredvidive promjene koje unose šum u vremensku seriju. Komponenta nasumičnosti može se izračunati oduzimanjem, odnosno dijeljenjem vremenske serije s ostalim komponentama prema izrazima 2.1 i 2.2

Primjer 2.2.1. Na primjeru podataka (Australian Bureau of Meteorology, 2014) o minimalnim dnevnim temperaturama u gradu Melbourneu očitavanih od $1981.$ do $1991.$ godine prikazane su komponente trenda, sezonalnosti i nasumičnosti. Kako se radi o dnevnim očitanjima sezonalnost je postavljena na 365 dana. Na apscisi su prikazani

dani protekli nakon 1.1.1981., a ordinatne osi prikazuju minimalne temperature. Vidljivi su jasni uzorci sezonalnosti te rastući trend minimalne temperature u razdoblju 2300. do 2800. dana.



Slika 2.1: Dekompozicija vremenske serije naivnom metodom

2.2.2. Stacionarnost vremenske serije

Stacionarnost vremenske serije može se objasniti pomoću stohastičnih procesa jer je vremenska serija realizacija promatranog stohastičnog procesa. Stohastički proces je kolekcija slučajnih varijabli $\{X(t)\}_{t\in T}$ (Gabbiani i Cox, 2017). Statistička svojstva stohastičkog procesa definiraju se pomoću statističkih svojstava slučajnih varijabli koje ga čine. Tako se definiraju srednja vrijednost stohastičkog procesa $M(t)$, varijanca stohastičkog procesa $Var(t)$ i kovarijanca stohastičkog procesa, u literaturi poznata kao autokovarijanca $Cov(t_1, t_2)$.

$$M(t) = E[X(t)] \tag{2.3}$$

$$Var(t) = Var(X(t)) \tag{2.4}$$

$$Var(X(t)) = E[(X(t) - M(t))^2] = E[X(t)^2] - M(t)^2 \tag{2.5}$$

$$Cov(t_1, t_2) = Cov(X(t_1), X(t_2)) \tag{2.6}$$

$$Cov(X(t_1), X(t_2)) = E[(X(t_1) - M(t_1)) \cdot (X(t_2) - M(t_2))] \tag{2.7}$$

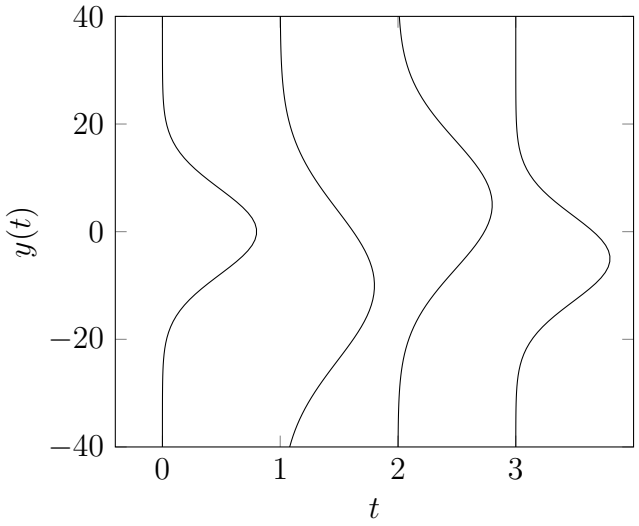
$$= E[X(t_1) \cdot X(t_2)] - E[X(t_1)] \cdot E[X(t_2)] \tag{2.8}$$

Strogo stacionarni stohastični proces je proces čija su statistička svojstva (srednja vrijednost, varijanca, kovarijanca između y_t i y_{t+k} i dr.) nepromjenjiva u vremenu

(Verbeek, 2008). Zahtjev stroge stacionarnosti u praksi je vrlo teško ispuniti, čak i raznim transformacijama vremenske serije. Zato se uvodi pojam slabe stacionarnosti koja zahtijeva konstantnu srednju vrijednost i varijancu te kovarijacijsku funkciju koja ovisi samo o razlici vremena (engl. *lag*) između dva opažanja (Verbeek, 2008).

Stacionarnost vremenske serije važna je jer se kod nestacionarne vremenske serije statistička svojstva mijenjaju kroz vrijeme pa modeli prognoziranja nauče distribuciju skupa za učenje koja se razlikuje od distribucije budućih podataka. Nestacionarnost negativno utječe na performanse statističkih modela i modela strojnog učenja. U praksi je nestacionarnost podataka čest problem koji se može ublažiti ponovnim učenjem modela za svaki novi podatak. Jedan primjer nestacionarne vremenske serije je slučajno gibanje (engl. *random walk*) u kojem je iduću vrijednost najtočnije predvidjeti s prethodnom poznatom vrijednosti.

Slika 2.2 prikazuje nestacionarni stohastički proces u četiri vremenska trenutka (četiri slučajne varijable). Realizacija prikazanog stohastičkog procesa jest vremenska serija koja se dobiva uzorkovanjem vrijednosti prema razdiobi u trenutku t. Proces je nestacionaran jer se srednja vrijednost i varijanca stohastičkog procesa mijenjaju kroz vrijeme.



Slika 2.2: Prikaz statističkih svojstava stohastičkog procesa čija srednja vrijednost i varijanca ovise o vremenu

Stacionarnost vremenske serije ispituje se statističkim testovima, a u nastavku su objašnjena dva komplementarna pristupa tom problemu.

2.2.3. Prošireni Dickey-Fullerov test jediničnog korijena

Dickey-Fullerov test jediničnog korijena testira vremensku seriju na postojanje jediničnog korijena. Tri su osnovne vrste testa: test jediničnog korijena, test jediničnog korijena uz deterministički trend te test jediničnog korijena uz deterministički vremenski trend.

Prošireni Dickey-Fullerov (engl. *Augmented Dickey-Fuller test*, ADF) test provodi se na jednak način kao i osnovni Dickey-Fullerov test, ali u obzir uzima autoregresijski model višeg reda pa su u nastavku objašnjene dvije vrste Dickey-Fullerovog testa prema (Verbeek, 2008).

Test jediničnog korijena uz deterministički trend

Test jediničnog korijena uz deterministički trend kreće od autoregresijskog modela prvog reda:

$$y_t = \delta + \theta \cdot y_{t-1} + \epsilon_t, \tag{2.9}$$

gdje je δ konstantni član regresijske jednadžbe, θ koeficijent, a ϵ_t pogrešku u trenutku t. Član jednadžbe θ procjenjuje se ($\hat{\theta}$) metodom najmanjih kvadrata (OLS). Testna statistika dana je s:

$$DF = \frac{\hat{\theta} - 1}{se(\hat{\theta})}, \tag{2.10}$$

gdje $se(\hat{\theta})$ označava standardnu pogrešku postupka najmanjih kvadrata.

Nulta hipoteza testa tvrdi da u vremenskoj seriji postoji jedinični korijen, tj. da je serija nestacionarna, a alternativna hipoteza tvrdi da je vremenska serija stacionarna. Iako nulta hipoteza testa implicira da je $\theta = 1$ i $\delta = 0$, u praksi se testira samo $\theta = 1$.

Testiranje se provodi usporedbom testne statistike i kritične vrijednosti. Ako je testna statistika manja od kritične vrijednosti nulta hipoteza se odbacuje. Za testiranje nulte hipoteze koriste se kritične vrijednosti zadane u (Fuller, 2009).

Testiranje se može provesti i nad diferenciranom vremenskom serijom. Oduzimanjem y_{t-1} s obje strane u 2.9 dobiva se sljedeći izraz:

$$y_t - y_{t-1} = \delta + \theta \cdot y_{t-1} + \epsilon_t - y_{t-1} \tag{2.11}$$

$$\Delta y_t = \delta + (\theta - 1) \cdot y_{t-1} + \epsilon_t, \tag{2.12}$$

gdje Δy_t označava promjenu vrijednosti vremenske serije u trenutku t. Ako nulta hipoteza vrijedi, tj. $\theta - 1 = 1 - 1 = 0$, dobivamo izraz slučajnog kretanja s trendom

čije je očekivanje $E[\Delta y_t] = \delta$:

$$\Delta y_t = \delta + \epsilon_t. \tag{2.13}$$

U slučaju stacionarnosti vremenske serije, član δ u 2.12 predstavlja srednju vrijednost vremenske serije. U slučaju postojanja jediničnog korijena član δ predstavlja deterministički trend u y_t jer se svaka sljedeća vrijednost od prethodne razlikuje za $\delta + \epsilon_t$. Problem nestacionarnosti vremenske serije zbog determinističkog trenda može se riješiti diferenciranjem vremenske serije čime se dobiva stacionarna vremenska serija s konstantnom srednjom vrijednosti i varijancom. Vremenska serija koja je nakon diferenciranja stacionarna naziva se diferencijski-stacionarna.

Test jediničnog korijena uz deterministički vremenski trend

Osim postojanja jediničnog korijena, nestacionarnost vremenske serije može uzrokovati i postojanje determinističkog vremenskog trenda.

Autoregresijski model korišten u ovom testu dobije se zbrajanjem vremenskog trenda ($\gamma \cdot t$) na izraz 2.9 i glasi:

$$y_t = \delta + \theta \cdot y_{t-1} + \gamma \cdot t + \epsilon_t, \tag{2.14}$$

uz $|\theta| < 1$ i $\gamma \neq 0$.

Oduzimanjem kao u 2.12 dolazi se do:

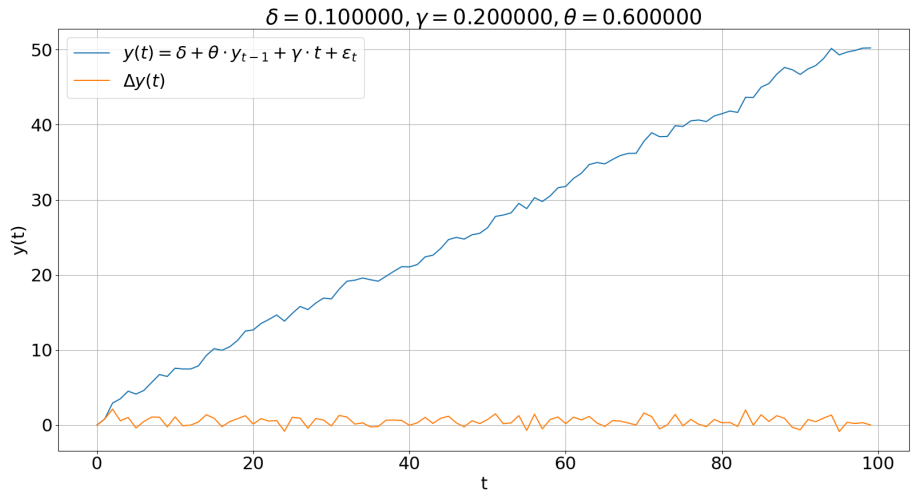
$$\Delta y_t = \delta + (\theta - 1) \cdot y_{t-1} + \gamma \cdot t + \epsilon_t. \tag{2.15}$$

Nulta hipoteza tvrdi da je $\delta = \gamma = \theta - 1 = 0$. Prilikom testiranja uzimaju se kritične vrijednosti za vremensku seriju s komponentom trenda jer je, za razliku od prethodnog testa, korišten drugačiji autoregresijski model.

Problem nestacionarnosti zbog postojanja determinističkog vremenskog trenda može se riješiti uvođenjem varijable t kao opažane ili transformacijom vremenske serije u reziduale dobivene oduzimanja vremenske serije i linearne regresije provedene nad vrijednostima vremenske serije. Vremenska serija koja se navedenim postupkom može dovesti u stacionarnu seriju naziva se trendno-stacionarna.

Primjer 2.2.2. U ovom primjeru generirana je vremenska serija s determinističkim vremenskim trendom prema izrazu 2.14. Test ADF proveden je nad generiranom vremenskom serijom te nad diferenciranom vremenskom serijom. Korišten je test iz Pythonovog paketa `statsmodels.tsa.stattools.adfuller` i to uz parametar

`regression` koji se postavlja na vrijednost `c` pri čemu se za testiranje koristi autoregresijski izraz s konstantnim članom δ ili `ct` pri čemu se koristi autoregresijski izraz s konstantnim članom δ te članom vremenskog trenda γ.



Slika 2.3: Generirana vremenska serija i diferencirana vremenska serija

Rezultati su prikazani tablicom 2.1. U slučaju izvorne vremenske serije test ADF nije odbacio nultu hipotezu u slučaju korištenja parametra `c` što je i očekivano jer se u tom slučaju koristi autoregresijski model s konstantnim članom. U slučaju izvorne vremenske serije i parametra `ct` test je odbacio nultu hipotezu jer je u autoregresijskom modelu imao član vremenskog trenda γ. Diferencirana vremenska serija stacionarna je po testu neovisno o korištenim parametrima jer ne sadrži komponentu trenda.

Tablica 2.1: Testne statistike

Osjenčane ćelije označuju odbacivanje nulte hipoteze.

Korak dif. / **Test-parametar**	/	1
ADF-c	-0.6205	-7.5912
ADF-ct	-5.1019	-7.5557

2.2.4. Kwiatkowski-Phillips-Schmidt-Shinov test stacionarnosti

Problem Dickey-Fullerovog, ali i ostalih testova na jedinični korijen koji kao nultu hipotezu imaju postojanje jediničnog korijena, je slaba snaga takvih testova koja se

očituje rijetkim odbacivanjem nulte hipoteze. Zato je u (Kwiatkowski et al., 1992) predložen drugačiji pristup testiranju na jedinični korijen. Ista referenca je poslužila za izradu ovog poglavlja.

Kod Kwiatkowski-Phillips-Schmidt-Shinovog (KPSS) testa nulta hipoteza tvrdi da je vremenska serija trendno-stacionarna, a alternativna hipoteza da je vremenska serija sadrži jedinični korijen. Osnovna ideja testa je dekompozicija vremenske serije na sumu deterministčkog vremenskog trenda ($\xi \cdot t$), slučajnog gibanja (r_t) i stacionarne pogreške (ϵ_t):

$$y_t = \xi \cdot t + r_t + \epsilon_t \tag{2.16}$$

$$r_t = r_{t-1} + u_t, \tag{2.17}$$

gdje su u_t nezavisne slučajne varijable s identičnom razdiobom. Po nultoj hipotezi je $\sigma_u^2 = 0$, a uz pretpostavku da je ϵ_t stacionaran je i varijanca slučajnog gibanja nula, tj. y_t je trendno-stacionarna.

Za izračun testne statistike najprije se izračuna pomoćna regresija vremenske serije nad pomakom po osi y i trendom te se pohrane OLS reziduali e_t. Iz reziduala se za svaki t izračuna parcijalna suma $S = \sum_{s=1}^{t} e_s$. Testna statistika dana je s:

$$KPSS = \sum_{t=1}^{T} \frac{S_t^2}{\hat{\sigma}^2}\,, \tag{2.18}$$

gdje $\hat{\sigma}^2$ predstavlja procjenu rezidualne varijance.

Ako je testna statistika veća od odabrane kritične vrijednosti nulta hipoteza se odbacuje. Kritične vrijednosti dane su u (Kwiatkowski et al., 1992).

2.2.5. Kombinacija testova KPSS i ADF

Pravilo statističkog testiranja kaže da zaključak testa može biti odbacivanje nulte i prihvaćanje alternativne hipoteze ili nema zaključka. Nulta hipoteza ADF testa tvrdi postojanje jediničnog korijena, dok za test KPSS tvrdi da jedinični korijen ne postoji. Testiranje se provodi na sljedeći način. Vremenska serija najprije se testira pomoću testa KPSS. Ako se nulta hipoteza testa KPSS odbaci vremenska serija sadrži jedinični korijen. Ako se nulta hipoteza testa KPSS ne može odbaciti nema zaključka te se nastavlja s testom ADF. Ako test ADF odbaci nultu hipotezu zaključak je da u vremenskoj seriji nije prisutan jedinični korijen. Ako test ADF ne odbaci nultu hipotezu nema zaključka i treba pokušati s transformacijama za uklanjanjem sezonalnosti i trenda.

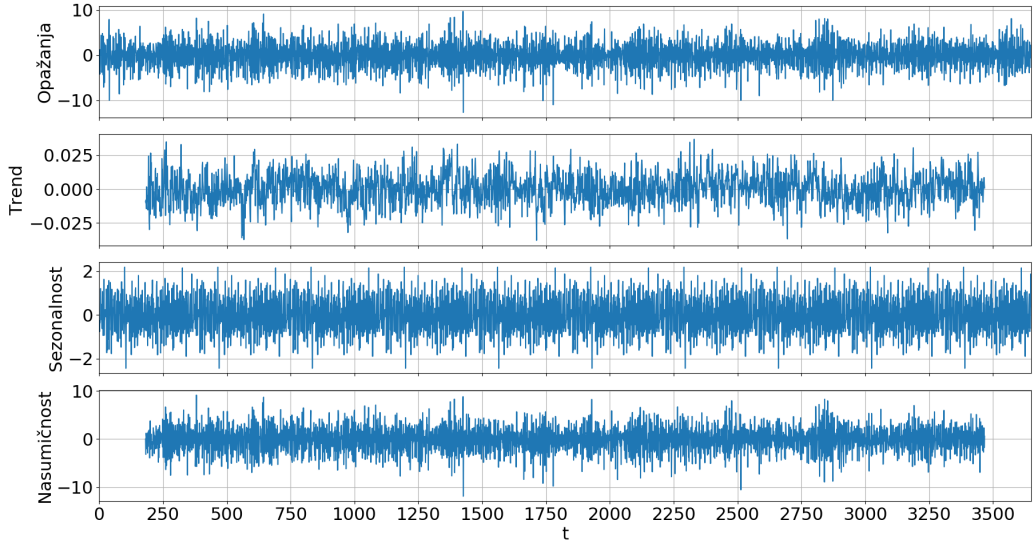
Primjer 2.2.3. U ovom primjeru je na podacima iz primjera 2.2.1 provedeno testiranje stacionarnosti uz potrebne transformacije vremenske serije pri čemu su korištene kritične vrijednosti za $5\ \%$ intervale pouzdanosti (KPSS: 0.463, ADF: -2.862). Za testiranje su korišteni testovi `kpss` i `adfuller` iz Pythonovog paketa `statsmodels.tsa.stattools`. Za parametar vrste regresije odabrana je vrijednost `c` jer vremenska serija iz primjera ne sadrži komponentu deterministčkog vremenskog trenda.

Rezultati testiranja prikazani su u tablici 2.2. Niti jedan test KPSS nije odbacio nultu hipotezu o trend-stacionarnosti, dok su sva tri proširena Dickey-Fullerova testa odbacila nultu hipotezu o postojanju jediničnog korijena. Zaključak testiranja jest stacionarnost vremenske serije. Iako je prošireni Dickey-Fullerov test već na izvornim podacima potvrdio stacionarnost, diferenciranje vremenske serije za korake 1 i 365 rezultiralo je većom pouzdanosti testa jer je testna statistika negativnija. Tomu je tako jer se iz vremenske serije uklonio trend (slika 2.4) i sezonalnost (slika 2.5).

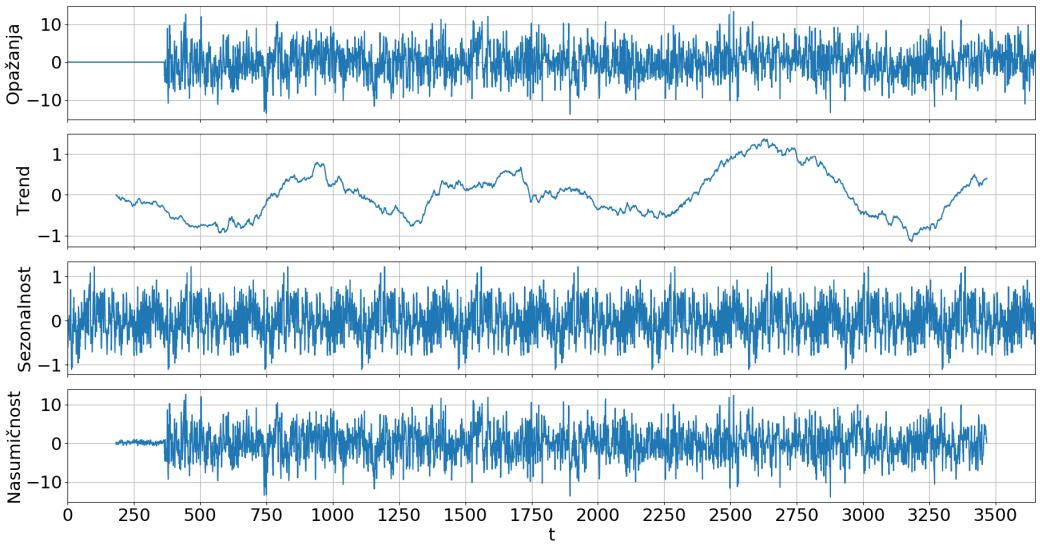
Tablica 2.2: Testne statistike

Osjenčane ćelije označuju odbacivanje nulte hipoteze.

Test-parametar \ **Korak dif.**	/	1	365
KPSS-c	0.0651	0.0269	0.272
ADF-c	-4.445	-18.0384	-18.508



Slika 2.4: Dekompozicija vremenske serije diferencirane za korak 1



Slika 2.5: Dekompozicija vremenske serije diferencirane za korak 365

2.2.6. Funkcija (parcijalne) autokorelacije

Funkcija autokorelacije (ACF) i parcijalne autokorelacije (PACF) su funkcije koje pokazuju koliko su opažanja vremenske serije y_t i y_{t+k} međusobno povezana. Kod stacionarnih vremenskih serija vrijednosti ACF i PACF jednake su za svaki odabrani vremenski trenutak t i ovise samo o vremenskom pomaku k.

Razlika funkcije parcijalne autokorelacije i funkcije autokorelacije je u tome što funkcija parcijalne autokorelacije uklanja ovisnost dvije varijable nastalu zbog drugih varijabli. U slučaju parcijalne autokorelacije između y_t i y_{t+k} to su vrijednosti $y_{t+1}, \ldots, y_{t+k-1}$.

Za dvije slučajne varijable x i y sa srednjim vrijednostima μ_x i μ_y te standardnim devijacijama σ_x i σ_y korelacija se računa prema:

$$Corr(x, y) = \frac{Cov(x, y)}{\sigma_x \cdot \sigma_y} = \frac{E[(x - \mu_x) \cdot (y - \mu_y)]}{\sqrt{Var(x)} \cdot \sqrt{Var(y)}}. \tag{2.19}$$

U praksi su srednja vrijednost i standardna devijacija procesa nepoznate pa se ACF i PACF procjenjuju iz konačnog skupa N vrijednosti vremenske serije. Kao najbolji procjenitelj za autokorelaciju s vremenskim pomakom k uzet je:

$$Corr(y_t, y_{t+k}) = \hat{\rho}_k = \frac{c_k}{c_0} \tag{2.20}$$

$$c_k = \frac{1}{N} \cdot \sum_{t=1}^{N-k} (y_t - \hat{\mu}_y) \cdot (y_{t+k} - \hat{\mu}_y), \; k = 0, 1, \ldots, K, \tag{2.21}$$

gdje se za K uzima vrijednost ne veća od $\frac{N}{4}$ (Box et al., 2015).

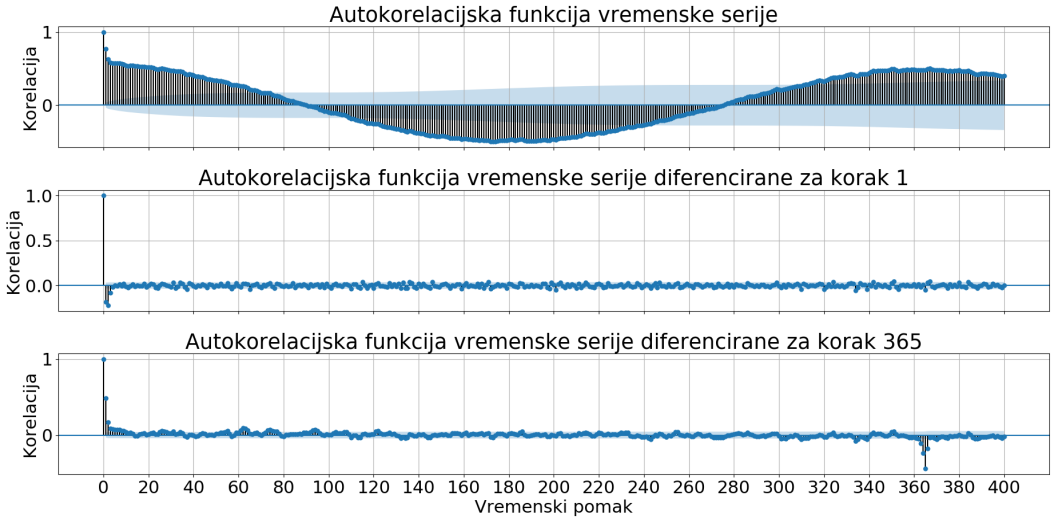
Primjer 2.2.4. U primjeru 2.2.1 dekomponirana je vremenska serija minimalnih temperatura te je ustanovljena sezonalnost od 365 dana.

Na grafovima u slici 2.6 prikazane su autokorelacijske funkcije izvorne vremenske serije (gornji graf), vremenske serije diferencirane za korak 1 (srednji graf) te vremenske serije diferencirane za korak 365 (donji graf). Plavo područje označava interval pouzdanosti.

Na grafu autokorelacijske funkcije izvorne vremenske serije vidljiva je jaka veza između početne vrijednosti i vrijednosti na svim vremenskim pomacima. Tako se na primjer može uočiti da je najveća negativna korelacija nastupila oko vremenskog pomaka 175 što je blizu vrijednosti polovine dana u godini. Taj rezultat ima smisla jer se negativna vrijednost autokorelacijske funkcije odnosi na suprotnu promjenu od početne vrijednosti. Primjerice ako je početna vrijednost bila u ljeto, vrijednost nakon 175 dana bit će u zimu. Najveća pozitivna korelacija javlja se nakon približno 365 dana zbog sezonalnosti.

Sezonalnost se može uočiti i iz donjeg grafa gdje je nakon uklanjanja komponente sezonalnosti negativna korelacija velika u vremenskom pomaku 365 što se može objasniti kao izmjena toplije i hladnije godine i obratno.

Iz grafova je također vidljivo kako diferenciranje vremenske serije uvelike smanjuje autokorelaciju.



Slika 2.6: Funkcije autokorelacije za podatke minimalnih temperatura

3. Prognoziranje vremenskih serija

Prognoziranje vremenskih serija odnosi se na predviđanje budućih vrijednosti vremenske serije pomoću trenutnih i prošlih vrijednosti. U najjednostavnijem slučaju prognozira se univarijatna vremenska serija pomoću svojih prošlih vrijednosti, a u složenijem se za prognoziranje univarijatne vremenske serije mogu koristiti druge vremenske serije koje utječu na vremensku seriju koju prognoziramo ili se prognozira multivarijatna vremenska serija.

Problem prognoziranja može se rješavati na klasični način modeliranjem procesa koji generira vremensku seriju ili kao problem nadziranog učenja.

3.1. Klasične statističke metode prognoziranja

Kako klasične statističke metode nisu tema ovog rada u nastavku su navedene samo najpoznatije klasične statističke metode prognoziranja vremenskih serija. Neke od tih metoda koristit će se kao osnovni modeli za usporedbu modela strojnog učenja izgrađenog u okviru praktičnog dijela rada.

Prednost klasičnih statističkih metoda prognoziranja nad metodama strojnog učenja je njihova dugotrajna primjena u praksi te potreba za malo domenskog znanja o procesu od interesa.

3.1.1. Integrirani autoregresijski model pomičnih prosjeka

Integrirani (I) autoregresijski (AR) model pomičnih prosjeka (MA) (engl. *autoregressive-integrated-moving average model*) je mješavina osnovnih modela koji se koristi za prognozu univarijatnih vremenskih serija. U nastavku su opisane komponente modela za prognoziranje nesezonalnih vremenskih serija, a sve jednadžbe su uz izmijenjenu notaciju preuzete iz (Hyndman i Athanasopoulos, 2018). U slučaju sezonalne vremenske serije model sadrži dodatne parametre, istovjetne navedenima, koje se odnose na sezonalnu komponentu.

Autoregresijski model

U poglavlju 2.2.3 naveden je autoregresijski model prvog reda (jednadžba 2.9). Poopćenjem se dolazi do autoregresijskog modela reda p koji buduću vrijednost vremenske serije predviđa temeljem linearne kombinacije p prošlih vrijednosti:

$$y_t = \delta + \theta_1 \cdot y_{t-1} + \ldots + \theta_p \cdot y_{t-p} + \epsilon_t \tag{3.1}$$

$$y_t = \delta + \sum_{i=1}^{p} \theta_i \cdot y_{t-i} + \epsilon_t \,, \tag{3.2}$$

gdje je δ konstantni član regresijske jednadžbe, θ_i parametri modela, a ϵ_t bijeli šum. Parametri modela mogu se procijeniti metodom najmanjih kvadrata ili drugim metodama (Yule-Walkerova metoda, metoda najveće izglednosti i dr.)

Model pomičnih prosjeka

Za razliku od autoregresijskog modela, model pomičnog prosjeka reda q za predviđanje buduće vrijednosti koristi q prošlih prognostičkih pogrešaka (ϵ):

$$y_t = c + \epsilon_t + \alpha_1 \cdot \epsilon_{t-1} + \ldots + \alpha_q \cdot \epsilon_{t-q} \tag{3.3}$$

$$y_t = c + \epsilon_t + \sum_{i=1}^{q} \alpha_i \cdot \epsilon_{t-i} \,. \tag{3.4}$$

Model nije regresija u uobičajenom smislu jer se varijable ϵ_t ne opažaju pa je prilagodba modela podacima kompliciranija nego kod autoregresijskog modela (Adhikari i Agrawal, 2013). Model pomičnog prosjeka za prognoziranje vremenske serije ne treba zamijeniti sa zaglađivanjem pomičnim prosjekom koje služi procjeni trenda i ciklusa (Hyndman i Athanasopoulos, 2018).

Kombinacija modela AR i MA

Kombinacijom autoregresijskog modela reda p i modela pomičnog prosjeka reda q dobije se model ARMA:

$$y_t = \delta + \sum_{i=1}^{p} \theta_i \cdot y_{t-i} + \epsilon_t + c + \epsilon_t + \sum_{i=1}^{q} \alpha_i \cdot \epsilon_{t-i} \,. \tag{3.5}$$

Ako se vrijednosti vremenske serije y_t zamijene s vrijednostima razlike vremenske serije $\Delta y_t = y_t - y_{t-1}$ dobiva se integrirani autoregresijski model pomičnih prosjeka. Diferenciranje može biti višestruko pa se s d označava broj uzastopnih diferenciranja prvog reda. Time je dobiven model ARIMA(p, d, q).

Kao i kod svih drugih modela, odabir parametara modela p, d i q ključan je za dobre performanse modela. Hyndman i Athanasopoulos (2018) predlažu korištenje grafova autokorelacijske i parcijalne autokorelacijske funkcije. U praksi se ovaj pristup nije pokazao dobrim jer su autokorelacijske i parcijalne autokorelacijske funkcije izračunate nad uzorkom podataka te nisu jednake teorijskim funkcijama.

3.1.2. Eksponencijalno zaglađivanje

Metode eksponencijalnog zaglađivanja buduću vrijednost vremenske serije predviđaju pomoću težinske sume prošlih vrijednosti, gdje težine eksponencijalno opadaju kako se ide u prošlost (Hyndman i Athanasopoulos, 2018). Time novije vrijednosti vremenske serije imaju veći utjecaj na prognozu budućih vrijednosti.

Najjednostavnija metoda eksponencijalnog zaglađivanja je jednostavno eksponencijalno zaglađivanje korištenjem $k+1$ prošlih vrijednosti:

$$\hat{y}_{t+1} = \alpha \cdot y_t + \alpha \cdot (1-\alpha) \cdot y_{t-1} + \ldots + \alpha \cdot (1-\alpha)^k \cdot y_{t-k}, \tag{3.6}$$

gdje je α parametar zaglađivanja iz intervala $[0, 1]$ (Hyndman i Athanasopoulos, 2018). Što je parametar zaglađivanja bliži 1 to će težine uz starija opažanja biti manje.

Holt-Wintersov model prognoziranja nadopunjuje jednostavni model kako bi se mogao koristiti za prognoziranje vremenskih serija s trendom i sezonalnosti.

3.1.3. Model vektorske autoregresije

Prethodni modeli koriste se za prognozu univarijatnih vremenskih serija. U slučaju povezanosti više varijabli koje međusobno utječu jedna na drugu koristi se vektorska autoregresija (VAR). Time se u proces prognoziranja uvode informacije o varijablama koje donose novo znanje o procesu koji se prognozira. Model pretpostavlja jednaku obostranu ovisnost varijabli, tj. da svaka varijabla na svaku drugu utječe jednako (Hyndman i Athanasopoulos, 2018).

Model vektorske autoregresije sastoji se od sustava jednadžbi koji je složen od po jedne regresije za svaku promatranu varijablu. U slučaju K međusobno zavisnih varijabli i modela reda p dobiva se:

$$\hat{y}_{k,t} = c_k + \sum_{i=1}^{K} \sum_{j=1}^{p} \phi_{ki,j} \cdot y_{i,t-j}\,, k = 1 \ldots K, \tag{3.7}$$

gdje je c_k konstantni član regresije, $\phi_{ki,j}$ parametar koji definira utjecaj varijable y_i vremenski pomaknute za j s varijablom y_k. Takav model ima K jednadžbi, a u svakoj po

$1+K\cdot p$ parametara što daje $K+K^2\cdot p$ parametara za procjenu. Porastom broja parametara povećava se i ukupna pogreška njihove procjene, a time i prognostička pogreška. Zato se u praksi bira manji broj koreliranih varijabli, a za određivanje optimalnog reda p koriste se informacijski kriteriji. (Hyndman i Athanasopoulos, 2018)

3.2. Metode strojnog učenja

Prognoziranje vremenske serije može se svesti na problem nadziranog učenja. Nadzirano učenje radi s označenim primjerima za učenje $(\vec{x}, y)$, odnosno za rad algoritama potrebne su ulazne vrijednosti $(\vec{x})$ i njihove oznake (y). Algoritmi koriste označene primjere kako bi naučili promjenu izlaza s obzirom na promjenu ulaza. Primjer $\vec{x}$ općenito je zapisan vektorom značajki, ali u specifičnom slučaju može biti skalar, vektor, matrica ili tenzor. U prognoziranju vremenske serije česte značajke su prošle vrijednosti vremenske serije koja se predviđa te vremenskih serija koje utječu na predviđanu vremensku seriju.

Prednosti metoda strojnog učenja s obzirom na klasične statističke metode je bolje podnošenje većeg broja značajki, nebitnih značajki, šuma u značajkama te učenje složenih veza između varijabli (Jason Brownlee, 2018). Dodatno, modeli strojnog učenja nisu ograničeni samo na numeričke značajke, nego mogu raditi i s kategoričkim značajkama.

Nadzirano učenje obuhvaća probleme klasifikacije i regresije. Klasifikacija je problem pridjeljivanja klase primjeru, pri čemu je klasa iz poznatog skupa klasa. Npr. klasifikacija slika pasa i mačaka bi slici $\vec{x}$ pridijelila oznaku $y \in \{\text{pas}, \text{mačka}\}$. Regresija je problem pridjeljivanja kontinuirane brojčane vrijednosti ulaznom primjeru. Primjer regresije može biti predviđanje plaće zaposlenika temeljem značajki: broj godina staža, stručna sprema i dr. Kada su u pitanju vremenske serije češće su primjene regresije nego klasifikacije pa se praktični dio rada bavi regresijom vremenskih serija.

Prognoziranje vremenske serije je specifično po tome što postoji vremenska uređenost. Prilikom učenja modela podaci moraju biti razvrstani prema vremenu. Kod određivanja značajki valja paziti da se u promatranom trenutku t kao značajke koriste samo tada poznate vrijednosti. U suprotnom bi se prilikom učenja modela kao značajke koristile vrijednosti koje u trenutku predviđanja nisu poznate čime bi se dobili lažno dobri rezultati. To je najočitiji primjer curenja podataka (engl. *data leakage*) kod predviđanja vremenskih serija.

Osmišljavanje i odabir značajki koje čine primjere nužan je, ali vremenski skup korak u prognoziranju vremenskih serija. Osim toga postupak zahtijeva domensko

znanje o problemu koji se rješava što je nedostatak prognoziranja pomoću strojnog učenja.

U nastavku su detaljno obrađeni odabrani algoritmi strojnog učenja korišteni u praktičnom dijelu rada.

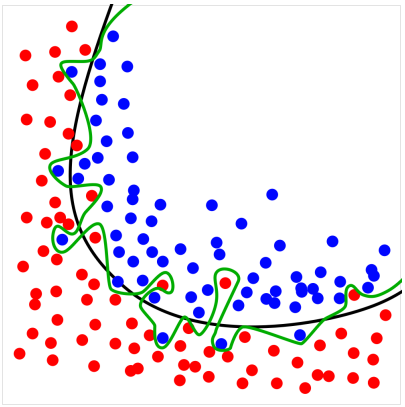
3.2.1. Osnovni pojmovi i notacija

Modeli nadziranog strojnog učenja sadrže parametre pomoću kojih se rade predviđanja. Vrijednosti parametara uče se pomoću primjera iz skupa za učenje $D_{train} = \{(\vec{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$. Svaki primjer za učenje sastavljen je od značajki $\vec{x}^{(i)}$ koje ga opisuju te oznake, odnosno vrijednosti koja se predviđa $y^{(i)}$. Performanse modela ispituju se na neviđenom skupu primjera za ispitivanje koji se u niti kojem smislu ne smije koristiti prilikom učenja ili odabira modela.

Hipoteza $h(\vec{x}, \vec{\theta})$ je funkcija koja primjere $\vec{x}$ i parametre modela $\vec{\theta}$ preslikava u izlaz modela. Skup svih hipoteza određenih parametrima $\vec{\theta}$ naziva se model. Cilj učenja modela je pronalazak parametara za koje je pogreška na ispitnom skupu najmanja.

Određivanje optimalnih parametara vođeno je odabranom funkcijom gubitka, odnosno pogreške. Funkcija gubitka odnosi se na pogrešku koju model čini na nekom primjeru $\vec{x}$ i u radu se označava s $L(y, h(\vec{x}, \vec{\theta}))$. Funkcija pogreške je zbroj funkcija gubitka na svim primjerima za učenje i u radu se označava s $E(\vec{\theta}|D_{train})$.

Prilikom učenja modela može doći do prenaučenosti. Prenaučenost je slučaj u kojem se parametri modela previše prilagode skupu za učenje pa je pogreška na tom skupu mala, ali je pogreška na ispitnom skupu velika. Za takav model se kaže da ne generalizira dobro jer šum i nasumičnost u podacima smatra kao bitne komponente. Za smanjenje prenaučenosti koristi se regularizacija. Regularizacija je svaka promjena koja se uvodi s ciljem bolje generalizacije, a ne poboljšava točnost na skupu za učenje. Primjer prenaučenosti dan je slikom 3.1 koja prikazuje dvije klase (crvena i plava). Zelenom linijom prikazana je decizijska granica prenaučenog klasifikacijskog modela, a crnom linijom decizijska granica regulariziranog modela.



Slika 3.1: Prikaz prenaučenog klasifikacijskog modela
(Commons, 2008)

3.2.2. Linearna regresija

Model

Linearna regresija hipotezu računa kao linearnu kombinaciju značajki:

$$h(\vec{x}, \vec{\theta}) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n \tag{3.8}$$

$$h(\vec{x}, \vec{\theta}) = \theta_0 + \sum_{i=1}^{n} \theta_i \cdot x_i \,, \tag{3.9}$$

gdje su x_i značajke, a θ_i parametri modela. Veći broj značajki povlači i veći broj parametara što rezultira složenijim modelom. U općenitom slučaju $n+1$ značajke hipoteza predstavlja hiperravninu u n-dimenzionalnom prostoru. Model je skup hipoteza indeksiran parametrima $\{h(\vec{x}, \vec{\theta})\}_{\vec{\theta}}$, odnosno skup svih hiperravnina u n-dimenzionalnom prostoru.

Funkcija gubitka i pogreške

Funkcija gubitka je kvadratna:

$$L(y, h(\vec{x}, \vec{\theta})) = (h(\vec{x}, \vec{\theta}) - y)^2. \tag{3.10}$$

Funkcija pogreške je zbroj funkcije gubitka nad svim primjerima iz skupa za učenje:

$$E(\vec{\theta} | D_{train}) = \sum_{i=1}^{N} L(y^{(i)}, h(\vec{x}^{\,(i)}, \vec{\theta})) \tag{3.11}$$

Optimizacijski postupak

Za potrebe izvoda jednadžbe optimalnih parametara korištena je (Šnajder i Dalbelo Bašić, 2014).

Parametri modela uče se postupkom najmanjih kvadrata pomoću skupa za učenje. Radi jednostavnijeg zapisa potrebno je vektorizirati hipotezu danu jednadžbom 3.9:

$$h(\vec{x}, \vec{\theta}) = \vec{\theta}^\intercal \cdot \vec{x}. \tag{3.12}$$

Funkcija pogreške dana s jednadžbom 3.11, uz vektoriziranu hipotezu danu s jednadžbom 3.12, može se zapisati pomoću operacija nad sljedećim matricama i vektorima:

$$\mathbf{X} = \begin{bmatrix} \vec{x}^{\,(1)\intercal} \\ \vec{x}^{\,(2)\intercal} \\ \vdots \\ \vec{x}^{\,(N)\intercal} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & \dots & x_n^{(N)} \end{bmatrix} \qquad \vec{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

Matrica primjera $\mathbf{X}$, vektor parametara $\vec{\theta}$ i vektor stvarnih vrijednosti (oznaka) $\vec{y}$.

Matrica primjera $\mathbf{X}$ sadrži N primjera za učenje, gdje svaki primjer sadrži n značajki i pomoćnu značajku koja je postavljena na 1. Ta značajka omogućuje matrično množenje matrice primjera i vektora parametara.

Uvrštavanjem jednadžbe 3.12 u jednadžbu 3.11 dolazi se do funkcije pogreške 3.15. Dodatno se funkcija pogreške radi matematičke jednostavnosti množi s $\frac{1}{2}$.

$$E(\vec{\theta}|D_{train}) = \frac{1}{2} \cdot \sum_{i=1}^{N} (\vec{\theta}^\intercal \cdot \vec{x}^{\,(i)} - y^{(i)})^2 \tag{3.13}$$

$$E(\vec{\theta}|D_{train}) = \frac{1}{2} \cdot (\mathbf{X} \cdot \vec{\theta} - \vec{y})^\intercal \cdot (\mathbf{X} \cdot \vec{\theta} - \vec{y}) \tag{3.14}$$

$$E(\vec{\theta}|D_{train}) = \frac{1}{2} \cdot (\vec{\theta}^\intercal \cdot \mathbf{X}^\intercal \cdot \mathbf{X} \cdot \vec{\theta} - \vec{\theta}^\intercal \cdot \mathbf{X}^\intercal \cdot \vec{y} - \vec{y}^\intercal \cdot \mathbf{X} \cdot \vec{\theta} + \vec{y}^\intercal \cdot \vec{y}). \tag{3.15}$$

U jednadžbi 3.15 članovi $\vec{\theta}^\intercal \cdot \mathbf{X}^\intercal \cdot \vec{y}$ i $\vec{y}^\intercal \cdot \mathbf{X} \cdot \vec{\theta}$ su skalari pa se jedan član smije transponirati i zbrojiti s drugim. Deriviranjem jednadžbe 3.15 po parametrima i izjednačavanjem s 0 dobivaju se parametri $\vec{\theta}^*$ za koje je pogreška na skupu za učenje najmanja:

$$\vec{\theta}^* = (\mathbf{X}^\intercal \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^\intercal \cdot \vec{y} \tag{3.16}$$

3.2.3. Linearni model regresije

Linearna regresija modelira linearnu ovisnost izlaza o ulaznim vrijednostima. Postavlja se pitanje što napraviti kada se želi modelirati nelinearna ovisnost izlaza o ulaznim vrijednostima. Jedan način je napraviti hipotezu koja modelira takvu ovisnost, npr. za kvadratnu ovisnost i dvije značajke:

$$h(\vec{x}, \vec{\theta}) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_1^2 + \theta_3 \cdot x_2 + \theta_4 \cdot x_2^2 + \theta_5 \cdot x_1 \cdot x_2. \tag{3.17}$$

U tom slučaju potrebno je provesti postupak najmanjih kvadrata za svaki takav model kako bi se dobili optimalni parametri što nije praktično.

Zato se umjesto promjene modela mijenjaju podaci. Matrica primjera **X** zamjenjuje se matricom dobivenom preslikavanjem podataka iz n-dimenzionalnog u m-dimenzionalni prostor. Preslikavanje se radi za svaki primjer $\vec{x}$ vektorskom funkcijom preslikavanja koja je sačinjena od baznih funkcija $\vec{\Phi}(\vec{x}) = [\phi_0(\vec{x}) \quad \phi_1(\vec{x}) \quad \dots \quad \phi_m(\vec{x})]$. Bazna funkcija preslikava vektor značajki u skalar množenjem i potenciranjem značajki. Time je dobiven linearni model regresije koji je linearan po svojim parametrima.

Za model dan izrazom 3.17 funkcija preslikavanja i model bi izgledali ovako:

$$\vec{\Phi}(\vec{x}) = [1 \quad x_1 \quad x_1^2 \quad x_2 \quad x_2^2 \quad x_1 \cdot x_2] \tag{3.18}$$

$$h(\vec{x}, \vec{\theta}) = \vec{\theta}^\intercal \cdot \vec{\Phi}(\vec{x}) \tag{3.19}$$

3.2.4. Regularizirana regresija

Odabirom nebitnih, koreliranih ili prevelikog broja značajki može doći do prenaučenosti. Ideja je početi sa složenijim modelom, koji općenito imaju veće vrijednosti parametara, ali omogućiti mehanizam koji bi iz njega isključio ili barem smanjio takve značajke. Kod linearnog modela regresije regularizacija se postiže promjenom funkcije pogreške u koju se dodaje regularizacijski izraz:

$$E_{Reg}(\vec{\theta}|D_{train}) = E(\vec{\theta}|D_{train}) + \lambda \cdot \Omega(\vec{\theta}), \tag{3.20}$$

gdje je λ faktor regularizacije koji određuje jakost regularizacije, a $\Omega(\vec{\theta})$ funkcija koja za veće vrijednosti parametara daje veće vrijednosti i tako povećava pogrešku za složenije modele. Za $\Omega(\vec{\theta})$ se uzima p-norma vektora parametara.

U tablici 3.1 dane su norme za $p = 0, 1, 2$.

Tablica 3.1: p-norme

p	Norma	Opis
0	$\|\|\vec{\theta}\|\|_0 = \sum_{j=1}^{n} [\![\theta_j \neq 0]\!]$	L_0 norma daje broj značajki koje nisu pritegnute na 0
1	$\|\|\vec{\theta}\|\| = \sum_{j=1}^{n} \|\theta_j\|$	L_1 norma linearno kažnjava porast parametara
2	$\|\|\vec{\theta}\|\|_2 = \sum_{j=1}^{n} \sqrt{\vec{\theta}^\intercal \cdot \vec{\theta}}$	L_2 norma daje Euklidovu normu vektora parametara

Oznaka $[\![]\!]$ označava Iversonovu zagradu koja poprima vrijednost 1 ako je izraz unutar zagrade istinit. Važno je primijetiti kako sume po parametrima kreću od $j = 1$. Kada bi sume kretale od $j = 0$ onda bi se parametar pomaka po y-osi također regularizirao, a kako taj parametar ne utječe na složenost modela nije ga potrebno regularizirati.

L_0 regularizacija radi odabir značajki i daje rijetke modele. Za $n + 1$ značajku, od kojih se regularizira n značajki ova regularizacija mora ispitati 2^n mogućnosti jer svaka značajka može biti uključena u model ili isključena iz modela. L_0 regularizacija nema rješenje u zatvorenoj formi. L_1 regularizacija također daje rijetke modele, odnosno može pritegnuti parametre blizu 0. Nedostatak je nepostojanje rješenja u zatvorenoj formi. Prednost L_2 regularizacije je u tome što ima rješenje u zatvorenoj formi, a nedostatak što ne daje rijetke modele.

U praksi se koristi kombinacija L_1 i L_2 regularizacije koja se zove *elastic net*.

Izraz optimalnih parametara L_2 regularizirane regresije dobiva se sličnim optimizacijskim postupkom kao kod neregularizirane linearne regresije uz izmjenu funkcije pogreške na pogrešku danu jednadžbom 3.20:

$$\vec{\theta}^* = (\mathbf{X}^\intercal \cdot \mathbf{X} + \lambda \cdot \boldsymbol{I}')^{-1} \cdot \mathbf{X}^\intercal \cdot \vec{y}, \tag{3.21}$$

gdje se $\mathbf{X}$ može zamijeniti s matricom preslikanih primjera, a $\boldsymbol{I}'$ označava jediničnu matricu uz iznimku da na prvom mjestu dijagonale ima upisanu nulu umjesto jedinice jer se θ_0 ne regularizira.

Važno je napomenuti kako regularizacija nije vezana isključivo uz linearne modele regresije, nego i ostale algoritme strojnog učenja. Podešavanje jačine regularizacije postiže se hiperparametrima algoritma povezanim sa složenošću, ali i povećanjem skupa za učenje.

3.2.5. Generalizirani linearni model regresije

Jedna od pretpostavki linearnog modela regresije je ravnanje izlaza po normalnoj razdiobi. Kod generaliziranih linearnih modela izlaz se ravna po nekoj od razdioba iz

eksponencijalne porodice razdioba.

Eksponencijalna porodica razdioba

Eksponencijalna porodica razdioba obuhvaća razdiobe koje se mogu zapisati kao:

$$p(\vec{x}|\vec{\eta}) = h(\vec{x}) \cdot g(\vec{\eta}) \cdot e^{\vec{\eta} \cdot \vec{u}(\vec{x})}, \tag{3.22}$$

gdje je $\vec{\eta}$ vektor parametara razdiobe (Bishop, 2006). U eksponencijalnu porodicu pripadaju mnoge poznate razdiobe: normalna, eksponencijalna, Bernoullijeva, Poissonova, geometrijska itd.

Model

Linearni model s funkcijom preslikavanja iz jednadžbe 3.19 općeniti je oblik linearnog modela regresije. Generalizirani linearni modeli poopćuju taj model s aktivacijskom funkcijom koja je određena odabranom razdiobom:

$$h(\vec{x}, \vec{\theta}) = f(\vec{\theta}^\intercal \cdot \vec{\Phi}(\vec{x})). \tag{3.23}$$

Za normalnu razdiobu uzima se funkcija identiteta, a generalizirani linearni model prelazi u linearni model regresije. Za Bernoullijevu razdiobu uzima se sigmoidalna funkcija te se dobiva logistička regresija koja, unatoč svojem nazivu, služi za binarnu klasifikaciju. U slučaju brojanja događaja u nekom intervalu koristi se Poissonova razdioba i eksponencijalna funkcija.

U literaturi se još spominju i funkcije povezivanja (engl. *link function*) koje su inverzi aktivacijske funkcije (Bishop, 2006). Veza aktivacijske funkcije i funkcije povezivanja objašnjena je u (Crosbie i Hinch, 1985), gdje se uspoređuje odnos aktivacijske funkcije koja djeluje na linearni model regresije $h = f(\vec{\theta}^\intercal \cdot \vec{\Phi}(\vec{x}))$ i funkcije povezivanja koja djeluje na predikciju $f^{-1}(h) = \vec{\theta}^\intercal \cdot \vec{\Phi}(\vec{x})$. Funkcija povezivanja f^{-1} je funkcija koja bi se primijenila za transformaciju nelinearnih podataka u slučaju korištenja linearnog modela regresije s aktivacijskom funkcijom identitetom i normalnom razdiobom izlaza.

3.2.6. Ansambli

Ansambl je meta-algoritam koji kombinira više osnovnih klasifikacijskih ili regresijskih algoritama s ciljem boljeg predviđanja. Meta-algoritmi slučajne šume i gradijentno ojačana stabla (engl. *gradient boosted trees*) tipični su predstavnici ansambala.

Motivacija za ansamble je činjenica da za sve podatke i probleme ne postoji najbolji algoritam koji bi ih podijelio po klasama ili predvidio kontinuiranu vrijednost. Zanimljiva ilustracija tri razloga zašto je ansambl osnovnih algoritama jači od pojedinog algoritma koji ga čini može se pronaći u (Dietterich, 2000). Prvi razlog odnosi se na slučaj skupa podataka koji je nedovoljno velik za veličinu prostora pretrage. U tom slučaju može postojati više modela koji postižu jednake performanse te se njihovim usrednjavanjem izbjegava rizik odabira pogrešnog modela. Drugi razlog jest pojava lokalnih optimuma u koje algoritmi upadaju prilikom pretrage prostora. Treći razlog je reprezentacija nepoznate funkcije koju se pokušava naučiti. Pojedine hipoteze ne moraju moći predstaviti traženu funkciju, dok se usrednjavanjem hipoteza postiže proširenje mogućih reprezentacija.

Da bi kombinacija osnovnih algoritama bila jača od osnovnog algoritma potrebno je diverzificirati osnovne algoritme te raditi s osnovnim algoritmima koji su bolji od slučajnog pogađanja. Diverzificirani algoritmi su oni koji na neviđenim podacima griješe na različit način (npr. jedan precjenjuje, drugi podcjenjuje). (Dietterich, 2000) Diverzifikacija se može postići manipulacijom skupa za učenje, manipulacijom značajki te manipulacijom algoritma. Kod manipulacije skupa za učenje osnovi algoritmi uče se na različitim podacima. Dodatno se osnovni algoritmi mogu razlikovati u značajkama pa će se algoritmu pridijeliti podskup značajki. Algoritam slučajnih šuma koristi ove dvije manipulacije. Manipulacija algoritma odnosi se na različite hiperparametre algoritama.

U slučaju regresije konačna predikcija ansambla od L osnovnih algoritama dobiva se težinskim usrednjavanjem:

$$h(\vec{x}, \vec{\theta}) = \frac{1}{L} \sum_{j=1}^{L} w_j \cdot h_j(\vec{x}, \vec{\theta}_j), \tag{3.24}$$

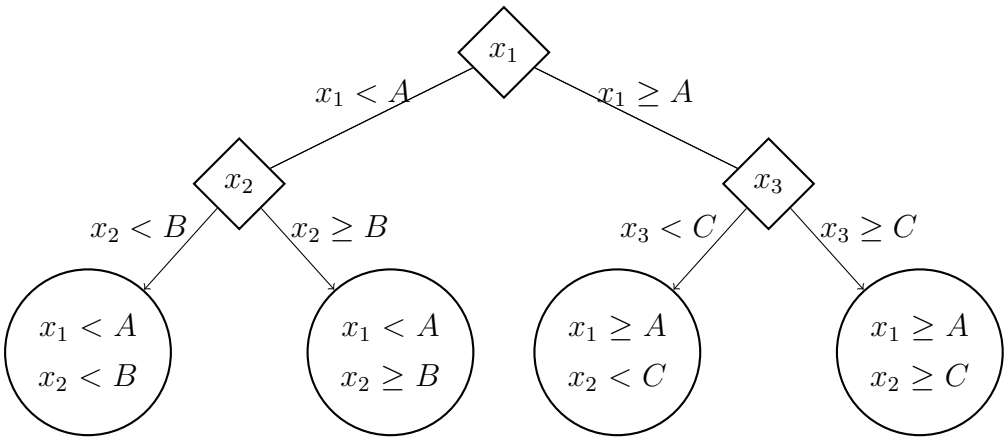
gdje w_j označava pouzdanost j-tog algoritma, odnosno koliko se j-tom algoritmu vjeruje.

3.2.7. Stablo odluke

Stablo odluke je algoritam koji se može koristiti za klasifikaciju i regresiju, a dio je algoritama slučajnih šuma i gradijentno ojačanih stabala. Algoritam kreće od korijena stabla u kojem se podaci granaju po vrijednosti odabrane značajke. Postupak se ponavlja sve do listova stabla. U slučaju regresije, predikcije se mogu dobiti usrednjavanjem ili linearnom regresijom primjera za učenje koji su završili u istim listovima. Za kla-

sifikaciju se može koristiti predikcija najčešćom klasom primjera za učenje u istom listu.

Na slici 3.2 prikazan je primjer stabla odluke pri čemu čvorovi prikazani rombom predstavljaju čvorove grananja, a čvorovi prikazani kružnicama listove stabla.



Slika 3.2: Primjer stabla odluke

Prvo pitanje koje se postavlja je kojim redoslijedom odabirati značajke i vrijednosti po kojima se podaci granaju. Drugo pitanje tiče se odabira dubine stabla. U nastavku su dani odgovori na ova pitanja nastali uz pomoć (Bishop, 2006).

Na prvo pitanje odgovara postupak izgradnje stabla odluke. Iscrpna pretraga svih mogućih kombinacija grananja nije moguća zbog velikog broja značajki i njihovih mogućih vrijednosti. Zato se problemu pristupa pohlepnim algoritmom u kojem se značajka i vrijednost po kojoj se grana odabiru s obzirom na prethodna grananja. Drugim riječima, ne traži se globalno najbolja strategija odabira značajki i vrijednosti grananja, nego se u trenutnim listovima stabla odabire ona kombinacija koja je po nekom kriteriju najbolja. Nakon grananja nastaju novi listovi za koje se odabir značajki i vrijednosti grananja ponavlja. Kriteriji odabira u slučaju regresije može biti informacijska dobit, srednja kvadratna pogreška ili standardna devijacija primjera za učenje koji su po prethodnim grananjima dospjeli do trenutno promatranog čvora grananja. Postupak izgradnje stabla odgovara učenju, a odabrane vrijednosti i redoslijed grananja parametrima modela.

Ako bi se grananja provodila dovoljno dugo, naposljetku bi svi primjeri iz skupa za učenje završili u zasebnim listovima. Time bi pogreška na skupu za učenje bila 0 jer bi svaki list sadržavao samo jedan primjer čija bi vrijednost ujedno i bila predikcija za taj list. No, u tom slučaju pogreška na skupu za ispitivanje bila bi velika što je znak prenaučenosti, kojoj su stabla odluke posebno sklona. Zato je odabir optimalne dubine stabla, odnosno broja grananja ključan. Najjednostavnije rješenje je postaviti najveću

dopuštenu dubinu stabla. Drugi pristup je prestati s grananjima kada promjena srednje kvadratne pogreške na skupu za učenje postane manja od neke vrijednosti ϵ ili kada se pogreška na ispitnom skupu počne povećavati. Treći pristup je definirati najmanji broj primjera za učenje koji se nalaze u istom listu. Uz navedene pristupe postoje i mnogi drugi.

Bitno je naglasiti razliku odabira optimalnog redoslijeda i vrijednosti grananja te odabira dubine stabla. Redoslijed i vrijednosti grananja uče se algoritmom iz skupa za učenje, a dubina stabla je hiperparametar algoritma koji se ne uči iz podataka.

3.2.8. Slučajne šume

Slučajne šume jedan su od najpoznatijih predstavnika ansambala, prvenstveno zbog široke uporabe u praksi i dobrih rezultata koje donose.

Naziv slučajne šume dolazi izravno iz pseudokoda algoritma. Na slučajan se način za svaki osnovni algoritam bira podskup značajki te se algoritam uči na podskupu skupa za učenje dobivenom uzorkovanjem s ponavljanjem. Šuma u nazivu algoritma označava da su za osnovni algoritam uzeta stabla odluke. Stabla odluke uče se nezavisno pa se implementacija može paralelizirati čime se postižu bolje vremenske performanse.

Algoritam 1 Pseudokod algoritma slučajnih šuma (Šnajder, 2017)

Veličina skupa za učenje označena je s N, a broj značajki s n. Za l-ti osnovni algoritam definira se veličina skupa za učenje D_l i odabrani podskup značajki F_l.

1: *šuma* $\leftarrow \emptyset$
2: **for** $l = 1 \ldots L$ **do**
3: $\quad D_l \leftarrow$ uzorkuj s ponavljanjem N' primjera iz skupa D, $N' \leq N$
4: $\quad F_l \leftarrow$ odabir n' značajki na slučajan način, $n' \leq n$
5: $\quad h \leftarrow$ nauči stablo odluke na primjerima iz D_l sa značajkama F_l
6: $\quad$ *šuma* $\leftarrow$ *šuma* $\cup \{h\}$

Hiperparametri

Osnovni hiperparametri algoritma su broj primjera za učenje N', broj značajki n', najmanji dozvoljeni broj primjera za učenje u listovima, broj stabala odluke i najveća dubina stabla odluke (hiperparametar stabla odluke).

U nastavku je prema Probst et al. (2019) objašnjen utjecaj hiperparametara na točnost algoritma i diverzifikaciju osnovnih algoritama (vidi poglavlje 3.2.6).

Smanjivanjem broja primjera za učenje N' osnovni klasifikatori su više diverzificirani jer je vjerojatnost odabira istih primjera za učenje manja. No, smanjivanjem broja primjera za učenje osnovnih algoritama njihova točnost opada jer se uče na manjem broju primjera. Bitno je naglasiti da se D_l dobiva uzorkovanjem s ponavljanjem pa i u slučaju $N' = N$ skup D_l ne mora sadržavati sve primjere iz D, nego se neki primjeri mogu pojaviti više puta.

Slično je i s hiperparametrom n' čije smanjivanje također donosi različitija, ali slabija stabla odluke. U literaturi je za regresiju predložena, a u praksi prihvaćena (programski jezik R i radni okvir Apache Spark) pretpostavljena vrijednost $n' = \frac{n}{3}$. Na hiperparametre N' i n' se može gledati kao kompromis između stabilnosti ansambla i točnosti osnovnih klasifikatora. Smanjivanjem oba hiperparametra N' i n' učenje algoritam se ubrzava.

Veza prenaučenosti i broja primjera u listovima spomenuta je u poglavlju 3.2.7. Smanjivanje ovog hiperparametra dovodi do dubljeg stabla, a time i prenaučenosti. Povećanje hiperparametra dovodi do eksponencijalnog smanjenja vremena izvođenja

Broj stabla odluke je hiperparametar koji se postavlja na što veću vrijednost pazeći na vrijeme izvođenja algoritma.

3.2.9. Gradijentno ojačana stabla

Jačanje (engl. *boosting*) je meta-algoritam koji slijedno generira komplementarne algoritme tako da svaki sljedeći algoritam ima veću točnost od prethodnog, tj. algoritmi svakom iteracijom jačaju. Takvo ponašanje postiže se učenjem algoritma na pogreškama prethodnih algoritama. Jačanje nije samo jedan meta-algoritam, nego postoji više inačica algoritma. Jedan od poznatijih je i meta-algoritam *AdaBoost* koji dijeli sličnosti s meta-algoritmom gradijentno ojačanih stabala. Meta-algoritam gradijentno ojačanih stabala osmislio je Friedman (1999), a u nastavku je dano objašnjenje algoritma.

Za osnovni algoritam odabrana su stabla odluke koja se po (Friedman, 1999) grade iterativno zbog čega paralelizacija učenja i predviđanja nije moguća. Inicijalno stablo sadrži samo jedan čvor, ujedno i list u stablu (osnovna predikcija F_0). Buduća stabla grade se temeljem izlaza prethodnih stabala. Prilikom izgradnje stabla odabir lista za grananje radi se izračunom smanjenja kvadratne pogreške nakon grananja, a za grananje se uzima list koji donosi najveće smanjenje kvadratne pogreške izračunate nad primjerima za učenje.

U slučaju M osnovnih algoritama definiraju se njihovi izlazi $F_m,\ m = 0, \ldots M$.

Indeks m kreće od 0 jer algoritam započinje s početnom predikcijom koja se dobiva minimizacijom funkcije pogreške nad skupom za učenje. Friedman (1999) kao funkciju gubitka uzima kvadratni gubitak pa je početna predikcija F_0 dobivena deriviranjem kvadratne pogreške i izjednačavanjem s 0 čime se dobiva aritmetička sredina oznaka primjera za učenje:

$$F_0 = \arg\min_{\gamma} \sum_{i=1}^{N} L(y^{(i)}, \gamma) \tag{3.25}$$

$$F_0 = \arg\min_{\gamma} \frac{1}{2} \cdot \sum_{i=1}^{N} (y^{(i)} - \gamma)^2 \tag{3.26}$$

$$\frac{\partial}{\partial \gamma} \left(\frac{1}{2} \cdot \sum_{i=1}^{N} (y^{(i)} - \gamma)^2 \right) = - \sum_{i=1}^{N} (y^{(i)} - \gamma) = 0 \tag{3.27}$$

$$F_0 = \frac{1}{N} \cdot \sum_{i=1}^{N} y^{(i)}, \tag{3.28}$$

gdje γ predstavlja predviđenu vrijednost. Razlika izlaza modela F_m i predviđanja γ objašnjena je u nastavku.

Zatim se iterativno gradi M osnovnih algoritama. Na početku m-te iteracije izračunaju se pseudo-reziduali nad primjerima za učenje. Pseudo-reziduali su negativne derivacije funkcije gubitka s obzirom na predikciju prethodnog algoritma:

$$\tilde{y}_m^{(i)} = - \left[\frac{\partial L(y^{(i)}, F(\vec{x}^{\,(i)}))}{\partial F(\vec{x}^{\,(i)})} \right]_{F(\vec{x}^{\,(i)}) = F_{m-1}(\vec{x}^{\,(i)})}, \; i = 1, \ldots N. \tag{3.29}$$

U slučaju kvadratnog gubitka pseudo-reziduali prelaze u obične reziduale:

$$\tilde{y}_m^{(i)} = - \frac{\partial}{\partial F_{m-1}(\vec{x}^{\,(i)})} \left(\frac{1}{2} \cdot (y^{(i)} - F_{m-1}(\vec{x}^{\,(i)}))^2 \right) \tag{3.30}$$

$$\tilde{y}_m^{(i)} = y^{(i)} - F_{m-1}(\vec{x}^{\,(i)}). \tag{3.31}$$

Pseudo-reziduale se može tumačiti slično kao gradijentni spust. Oni predstavljaju promjenu predikcije u smjeru negativne derivacije funkcije gubitka po izlazu prethodnog modela s ciljem smanjenja funkcije gubitka. Gradijent u nazivu meta-algoritma dolazi upravo iz ovog koraka. Izračunati pseudo-reziduali služe kao oznake pri izgradnji m-tog stabla odluke. Takvo stablo pomoću značajki $\vec{x}$ predviđa pogrešku prethodnog algoritma. Prilikom izgradnje (učenja) stabla odluke primjeri za učenje puštaju se iz korijena prema listovima. Na kraju u pojedinim listovima izgrađenog stabla odluke završi barem jedan podatak, odnosno njegov pseudo-rezidual (oznaka). Postavlja se pitanje s kojom vrijednosti predvidjeti vrijednosti koje završe u istom listu. S γ_{lm} označava se predikcija za l-ti list, $l = 1, \ldots L$. Predikcija l-tog lista m-tog algoritma

dobiva se minimiziranjem odabrane funkcije pogreške uzevši u obzir prethodnu predikciju $F_{m-1}(\vec{x}^{\,(i)})$:

$$\gamma_{lm} = \arg\min_{\gamma} \sum_{\vec{x}^{\,(i)} \in R_{lm}} L(y^{(i)}, F_{m-1}(\vec{x}^{\,(i)}) + \gamma), \tag{3.32}$$

gdje $\vec{x}^{\,(i)} \in R_{lm}$ označava primjere koji su grananjem završili u l-ti list m-tog stabla. U izrazu 3.32 član γ se može tumačiti kao ono što je potrebno dodati izlazu prethodnog algoritma za primjere koji su završili u promatranom listu kako bi se funkcija pogreške minimizirala. U slučaju da prethodni algoritam precjenjuje stvarnu vrijednost $y^{(i)}$ potrebno je umanjiti njegov izlaz pomoću negativne γ, a u suprotnom povećati s pozitivnim članom. U slučaju kvadratnog gubitka, predikcija za l-ti list je srednja vrijednost reziduala primjera koji su završili u tom listu:

$$\frac{\partial}{\partial \gamma} \left(\frac{1}{2} \cdot \sum_{\vec{x}^{\,(i)} \in R_{lm}} (y^{(i)} - (F_{m-1}(\vec{x}^{\,(i)}) + \gamma))^2 \right) = 0 \tag{3.33}$$

$$\gamma_{lm} = \frac{1}{|R_{lm}|} \cdot \sum_{\vec{x}^{\,(i)} \in R_{lm}} \tilde{y}_m^{(i)}. \tag{3.34}$$

Nakon izračuna predikcije za pojedini list stabla može se izračunati i konačni izlaz m-tog algoritma kao kombinacija predikcije prethodnog i trenutnog algoritma:

$$F_m(\vec{x}) = F_{m-1}(\vec{x}) + \eta \cdot \sum_{j=1}^{L} \gamma_{jm} \cdot [\![\vec{x} \in R_{jm}]\!], \tag{3.35}$$

gdje je η stopa učenja, a $[\![]\!]$ Iversonova zagrada koja poprima vrijednost 1 ako je tvrdnja unutar zagrade istinita, a inače 0.

Razlika predikcije γ_{lm} i F_m je u tome što se γ_{lm} odnosi na predikciju pojedinog lista u m-tom stablu, a F_m na predikciju m-tog algoritma.

Svakom sljedećom iteracijom smanjuje se pogreška ansambla na skupu za učenje jer se zbrajanjem srednjih vrijednosti pseudo-reziduala na prethodnu predikciju predikcije pomiču prema stvarnim vrijednostima. Stabla se nastavljaju izgrađivati istim postupkom sve dok se ne izgradi unaprijed zadani broj stabala ili smanjenje pogreške padne ispod male konstante ϵ. Moguće je i nastaviti izgradnju stabala sve dok se pogreška na ispitnom skupu ne počne povećavati.

Stohastička inačica algoritma

Friedman (1999) predlaže i stohastičku inačicu algoritma koja bi uvođenjem slučajnosti trebala poboljšati mogućnost generalizacije ansambla. Jedina razlika s obzirom

na izvorni algoritam jest učenje stabala odluke na podskupu skupa za učenje dobivenog slučajnim uzorkovanjem bez ponavljanja na početku svake iteracije. Veličina podskupa označena je s $\tilde{N}$, a u slučaju $\tilde{N} = N$ algoritam prelazi u izvornu inačicu jer se uzorkuje bez ponavljanja.

Hiperparametri

Stopa učenja je hiperparametar koji sprječava prenaučenost ansambla. Zbrajanjem predikcije pseudo-reziduala γ na izlaz prethodnog algoritma u jednadžbi 3.35 znači da se predikcija prethodnog algoritma prilagođava skupu za učenje. Stopa učenja osigurava male pomake u pravom smjeru između izlaza slijednih algoritama, bez koje bi se modeli previše prilagodili podacima za učenje, tj. došlo bi do prenaučenosti. U (Friedman, 1999) empirijski je pokazano da stope učenja manje od 0.1 poboljšavaju generalizaciju modela. Stopa učenja utječe na hiperparametar broja iteracija, odnosno broja stabala odluke M. Manja stopa učenja zahtjeva veći broj iteracija što uzrokuje veću vremensku složenost učenja ansambla, ali i predviđanja jer je za izračun izlaza ansambla potrebno izračunati izlaze $F_m, m = 0, \ldots M$.

Ostali hiperparametri vezani su uz stabla odluke i slični su hiperparametrima slučajnih šuma kao što su najmanji broj primjera u listu i najveća dubina stabla. U stohastičnoj inačici algoritma uvodi se hiperparametar veličine uzorkovanog podskupa za učenje $\tilde{N}$.

4. Radni okvir Apache Spark

Apache Spark je radni okvir za izračunavanje na računalnom grozdu s podrškom za programske jezike Java, Scala, Python i R. U praksi se uglavnom koristi uz Scalu i Python jer je za te programske jezike ponuđeno najviše funkcionalnosti. Apache Spark nudi komponente za obradu strukturiranih i polustrukturiranih podataka, obradu strujećih podataka, knjižnice za strojno učenje te komponentu za obradu grafova. U praktičnom dijelu rada intenzivno su korištene komponente za obradu podataka i strojno učenje.

Inačice Apache Sparka prije inačice 1.6 oslanjale su se na podatkovnu apstrakciju *resilient distributed dataset* ili skraćeno RDD. U novijim inačicama RDD-ove je zamijenila učinkovitija podatkovna apstrakcija *data frame* koja podsjeća na tablicu relacijske baze podataka, ali je optimizirana za raspodijeljenu obradu.

Ugrađene funkcije za dobivanje zaostalih vrijednosti, funkcije nad vremenskim prozorom i mogućnost pisanja vlastitih funkcija koje djeluju nad stupcima *data framea* čine Apache Spark pogodnim za učinkovitu obradu podataka i vremenskih serija.

4.1. Inačica Apache Sparka

U praktičnom dijelu rada korištena je inačica Apache Spark $2.4.0$ s podrškom za programski jezik Scala.

4.2. Knjižnica za strojno učenje

Apache Spark nudi dvije knjižnice za strojno učenje. Starija knjižnica bazira se na podatkovnoj strukturi RDD-ova te se u praksi sve češće zamjenjuje novijom knjižnicom koja se bazira na *data frameovima*. U praktičnom dijelu rada korištena je novija knjižnica. U nastavku poglavlja opisane su klase paketa `org.apache.spark.ml.regression` koje implementiraju algoritme strojnog učenja opisane u prethodnim poglavljima te korištene u praktičnom dijelu rada.

4.2.1. Linearna regresija

Klasa `LinearRegression` predstavlja linearnu regresiju i podržava kvadratni gubitak te Huberov gubitak koji je manje osjetljiv na stršeće vrijednosti. Klasa podržava L_1 i L_2 regularizacije, kao i njihovu kombinaciju (engl. *elastic net*) prilikom čega se hiperparametrom `elasticNetParam` definira u kojem postotku se koristi L_1, a u kojem L_2 regularizacija. Jačinu regularizacije podešava se hiperparametrom `regParam`.

Hiperparametar `fitIntercept` odlučuje hoće li se pomak po y-osi učiti iz podataka ili će se postaviti na 0 (npr. za prethodno centrirane podatke).

Standardizacija skupa za učenje prije učenja algoritma podešava se postavljanjem hiperparametra `standardization` na istinitu vrijednost.

Algoritam linearne regresije uči se pomoću jednog od rješavača, a ponuđeni su modificirani Newtonov postupak, analitičko rješavanje te automatska odluka prilikom čega se analitičko rješavanje koristi kada je moguće. Pri odabiru Newtonovog postupka moguće je postaviti najveći broj iteracija algoritma. Važno je napomenuti kako je uz Hubertov gubitak moguća jedino optimizacija modificiranim Newtonovim postupkom.

4.2.2. Generalizirana linearna regresija

Klasa `GeneralizedLinearRegression` implementira generaliziranu linearnu regresiju. Za razdiobu izlaza moguće je odabrati Gaussovu, binomnu, Poissonovu, gamma i Tweediejevu razdiobu. Za svaku od navedenih razdioba moguće je odabrati nekoliko funkcija povezivanja.

Klasa podržava samo rješavanje pomoću iterativne metode ponderiranih najmanjih kvadrata (engl. *iteratively reweighted least squares*), a hiperparametri najvećeg dozvoljenog broja iteracija i jačine regularizacije imaju istu interpretaciju kao i kod linearne regresije.

4.2.3. Stablo odluke

Algoritam stabla odluke za regresiju implementiran je klasom `DecisionTreeRegressor`.

Strategija grananja

U poglavlju 3.2.7 raspravlja se o strategijama odabira redoslijeda i vrijednosti značajki prilikom grananja. Trenutno je implementirana samo strategija odabira redoslijeda i vrijednosti odabira značajki za grananje prema kriteriju informacijske dobiti (engl.

information gain, IG). Informacijska dobit za regresiju se računa kao razlika varijance u čvoru koji se grana (D) i težinskog zbroja varijanci čvorova dobivenih grananjem (D_L i D_R):

$$IG = Var(D) - \left(\frac{N_L}{N} \cdot Var(D_L) + \frac{N_R}{N} \cdot Var(D_R) \right). \tag{4.1}$$

Za grananje se uzima značajka i vrijednost za koju je informacijska dobit najveća.

Vrijednosti značajke grananja

U poglavlju 3.2.7 također je spomenut problem složenosti odabira vrijednosti značajki po kojima se grana. Za kontinuirane značajke implementacija rješava problem odabirom kandidata vrijednosti za grananje. Tako se ne pretražuje po svim mogućim vrijednostima značajke, nego se podaci razvrstaju te se izračunaju kvantili. U ovom slučaju kvantili predstavljaju podjelu razvrstanih podataka u pretince tako da svaki pretinac sadrži jednak broj podataka. Radi jeftinije vremenske složenosti u slučaju raspodijeljenih podataka kvantili se računaju nad uzorkom podataka, umjesto nad svim podacima. Pomoću hiperparametra najvećeg dopuštenog broja pretinaca `maxBins` definira se granulacija podjele podataka. Finija granulacija znači da algoritam provjerava više mogućih kandidata, a time povećava vremensku složenost.

Kod kategoričkih značajki, u slučaju da je broj vrijednosti koje značajka poprima manji od najvećeg dopuštenog broja pretinaca pretraga se vrši po svim vrijednostima značajke. U suprotnom se koristi heuristička metoda odabira vrijednosti značajki čime se broj mogućih podjela za značajku s M mogućih vrijednosti smanjuje s $2^{M-1} - 1$ na $M - 1$ kandidata. Heuristički postupak za svaku vrijednost značajke računa varijancu primjera za koje značajka poprima tu vrijednost. Nakon razvrstavanja vrijednosti značajke kandidati se dobiju pomicanjem granice između razvrstanih vrijednosti značajke. Objašnjenje heurističke metode uz primjer dano je u primjeru 4.2.1.

Primjer 4.2.1. Neka je `maxBins` $= 2$ i neka je Z kategorička značajka koja poprima četiri vrijednosti: A, B, C i D. Neka je uzlazno razvrstavanje prema varijanci dalo sljedeći redoslijed: A, C, D, B. Tada su ovom heurističkom metodom definirane tri podjele: [(A), (C, D, B)], [(A, C), (D, B)], [(A, C, D), (B)].

Kriteriji zaustavljanja grananja

Implementacija zaustavlja grananja kada je ispunjen jedan od tri uvjeta. Prvi uvjet odnosi se na najveću dozvoljenu dubinu stabla koja se definira hiperparametrom

maxDepth. Drugi uvjet odnosi se na hiperparametar minInfoGain koji definira najmanju promjenu informacijske dobiti. Kada je informacijska dobit manja od postavljenog hiperparametra grananje se zaustavlja. Posljednji uvjet povezan je s najmanjim brojem primjera u čvoru nakon grananja koji se postavlja hiperparametrom minInstancesPerNode.

4.2.4. Slučajne šume

Meta-algoritam slučajnih šuma namijenjen regresiji implementiran je u klasi RandomForestRegressor. Hiperparametar strategije odabira podskupa značajki postavlja se pomoću featureSubsetStrategy, a ponuđene strategije su sve značajke, trećina značajki, korijen broja značajki, logaritam broja značajki te željeni postotak broja značajki. Moguće je odabrati automatski odabir prilikom kojeg se za regresiju odabire trećina značajki.

Prilikom instanciranja klase potrebno je definirati hiperparametre stabla odluke kao što su najveći dopušteni broj pretinaca maxBins, najveća dubina stabla maxDepth, minimalna informacijska dobit za nastavak grananja minInfoGain, najmanji broj primjera u listu za nastavak grananja minInstancesPerNode, kriteriji za izračun informacijske dobiti impurity (podržana samo varijanca) te postotak primjera za uzorkovanje s ponavljanjem iz skupa za učenje subsamplingRate. Dodatno se specificira broj stabala odluke numTrees.

4.2.5. Gradijentno ojačana stabla

Implementacija gradijentno ojačanih stabala temelji se na (Friedman, 1999), a nalazi se u klasi GBTRegressor. Hiperparametri strategije odabira podskupa značajki, najveći dopušteni broj pretinaca, najveća dubina stabla, minimalna informacijska dobit za nastavak grananja, najmanji broj primjera u listu za nastavak grananja, kriterij izračuna informacijske dobiti i postotak primjera za uzorkovanje s ponavljanjem iz skupa za učenje jednaki su kao kod slučajnih šuma.

Novi hiperparametri su funkcija gubitka lossType za koju su podržane kvadratna i apsolutna funkcija gubitka, najveći broj iteracija meta-algoritma maxIter i stopa učenja stepSize.

Prilikom učenja je moguće definirati koji primjeri su za učenje, a koji za validaciju te tako ostvariti rano zaustavljanje algoritma. Rano zaustavljanje algoritma događa se u dva slučaja. Kada je pogreška na validacijskom skupu veća od 0.01 promatra se razlika validacijske pogreške između iteracija. Kada je ta razlika manja od umnoška

validacijske tolerancije (hiperparametar `validationTol`) i validacijske pogreške aktivira se rano zaustavljanje. Ako je pak validacijska pogreška manja ili jednaka 0.01, rano zaustavljanje se aktivira u slučaju da je razlika validacijske pogreške između iteracija veća od umnoška validacijske tolerancije i konstante 0.01.

5. Studijski slučaj prognoziranja dnevne proizvodnje mlijeka

Ovo poglavlje obuhvaća opis praktičnog dijela rada u kojem je provedena obrada podataka, vizualizacija podataka, statistička testiranja, izgradnja značajki, odabir modela te na kraju prognoziranje vremenske serije i analiza rezultata. Podaci korišteni u praktičnom dijelu rada odnose se na mljekarsku industriju, a krajnji cilj je prognoziranje vremenske serije mase dnevne proizvodnje mlijeka na razini krda mliječnih krava. Motivacija za rješavanje ovog problema je planiranje buduće zarade, dogovor bolje otkupne cijene s obzirom na predviđenu proizvedenu količinu, planiranje odvoza proizvedenog mlijeka te predviđanje stanja krda mliječnih krava.

5.1. Opis podataka

Korišteni podaci predstavljeni su sa šest vrsta datoteka formata CSV prikazanih tablicom 5.1 i opisanih u nastavku. Po definiciji vremenske serije opažanja moraju biti jedinstveno identificirana vremenskom oznakom podatka. Sirovi podaci sami za sebe ne predstavljaju vremensku seriju jer je vremenska oznaka samo dio ključa pojedinih tablica pa ih je potrebno obraditi i svesti na vremensku seriju.

Podaci o dnevnim prinosima mlijeka sadrže zapise o tri dnevne mužnje za pojedinu životinju određenu jedinstvenim identifikatorom. Mase mlijeka izražene su u lb. Analizom podataka uočena je mogućnost da je neka od tri dnevne mužnje postavljena na 0 lb što znači da se ta mužnja nije dogodila.

Podaci o profilu životinje sadrže datume dolaska u krdo i datum odlaska iz krda. Datum odlaska iz krda omogućuje ispravnost podataka izračunatih u poglavlju 5.2.1.

Laktacijski podaci sadrže informacije o laktacijama. Laktacija predstavlja period u kojem muzne krave proizvode mlijeko, a započinje događajem teljenja. Nulta laktacija odnosi se na period prije prvog teljenja u kojoj životinja ne proizvodi mlijeko. Suhostaj označava kraj laktacije, a to je period u kojem mliječne krave ne proizvode

ili proizvode male količine mlijeka. Suhostaj uobičajeno započinje 40 do 60 dana prije teljenja i završava događajem teljenja koji označava prelazak u novu laktaciju.

Datoteka o ostalim događajima sadrži podatke o korisnički definiranim događajima. Vrijednosti stupca naziva događaja su „SOLD“, „DIED“, „MOVE“, „ABORT“ i „DRY“. Prve tri vrijednosti odnose se na odlazak životinje iz krda. Događaj „ABORT“ označava prekid trudnoće, a „DRY“ ulazak u suhostaj. Ovi podaci mogu biti redundantni s laktacijskim podacima (ulazak u suhostaj) i podacima o profilu (datum odlaska iz krda), ali koriste u slučaju nedostajućih zapisa u nekoj od ove dvije datoteke.

Podaci o atmosferskim uvjetima sadrže iznose najmanjih i najvećih temperatura, relativne vlažnosti i indeksa topline. Indeks topline računa se pomoću temperature i relativne vlažnosti što će se odraziti na odabir značajki. Navedene značajke imaju utjecaj na proizvodnju mlijeka muznih krava i njihovo fizičko stanje (West et al., 2003).

Tablica 5.1: Opis sirovih podataka

Podaci	Stupci	Opis
Dnevni prinos mlijeka po životinji	UniqueIdentifier	Jedinstveni identifikator životinje
	Date	Datum mjerenja
	Milk1 [lb]	Masa mlijeka dobivena prvom mužnjom
	Milk2 [lb]	Masa mlijeka dobivena drugom mužnjom
	Milk3 [lb]	Masa mlijeka dobivena trećom mužnjom
Podaci o profilu životinje	UniqueIdentifier	Jedinstveni identifikator životinje
	EDAT	Datum dolaska u krdo (rođenje, kupnja, premještaj)
	ARDAT	Datum odlaska iz krda (smrt, prodaja, premještaj)
Laktacijski podaci	UniqueIdentifier	Jedinstveni identifikator životinje
	DDAT	Datum ulaska u suhostaj
	Group	Redni broj laktacije
Podaci o teljenjima	UniqueIdentifier	Jedinstveni identifikator životinje
	Date	Datum teljenja
	Calf1Code	Kod ishoda teljenja
	Calf2Code	Kod ishoda teljenja
Ostali događaji	UniqueIdentifier	Jedinstveni identifikator životinje
	Date	Datum događaja
	Name	Naziv događaja
	Remark	Komentar događaja
Atmosferski uvjeti	Day	Datum mjerenja
	MinTemp	Najmanja temperatura
	MaxTemp	Najveća temperatura
	MinRh	Najmanja relativna vlaga
	MaxRh	Najveća relativna vlaga
	MinHeatIndex	Najmanji indeks topline
	MaxHeatIndex	Najveći indeks topline

5.2. Obrada podataka

Iz sirovih podataka izračunata je vremenska serija koja će se prognozirati te vremenske serije koje će poslužiti u njezinom predviđanju.

5.2.1. Izgradnja vremenskih serija

Vremenska serija dnevne proizvodnje mlijeka

Vremenska serija koju je cilj prognozirati odnosi se na dnevnu proizvodnju mlijeka na razini krda mliječnih krava. Podaci o dnevnim prinosima predstavljaju dnevne prinose po životinji pa ih je potrebno agregirati. Najprije su pomoću podataka o tri dnevne mužnje izračunate vrijednosti ukupne dnevne proizvodnje mlijeka po životinji, a zatim su podaci grupirani prema datumu i sumirani kako bi se dobila dnevna proizvodnja mlijeka cijelog krda. Tim postupkom dobiveni podaci pretvoreni su u vremensku seriju jer su jedinstveno identificirani pomoću vremenske oznake, odnosno datuma.

Vremenske serije broja dnevnih mužnji i muznih krava

Iz podataka o dnevnim prinosima mlijeka po životinji izračunata je vremenska serija broja dnevnih mužnji. Vremenska serija dobiva se grupiranjem po datumu i prebrajanjem mužnji za koje vrijedi da je masa mlijeka veća od 0 lb. Uvjet mase veće od 0 lb nužan je jer su neki od podataka koji predstavljaju tri mužnje postavljeni na masu 0 lb sa značenjem da se ta mužnja nije dogodila.

Iz podataka o dnevnim prinosima mlijeka po životinji izračunata je vremenska serija broja muznih krava. Vremenska serija izračunata je grupiranjem po datumu i prebrajanjem jedinstvenih identifikatora, uz uvjet da je suma tri dnevne mužnje veća od 0, odnosno da je krava na pojedini datum dala mlijeko.

Motivacija za određivanje ovih vremenskih serija je mogućnost da model nauči povezanost broja dnevnih mužnji i muznih krava s proizvodnjom mlijeka.

Vremenska serija broja krava u suhostaju

Laktacijski podaci i podaci o ostalim događajima služe za određivanje datuma ulaska u suhostaj. Iz podataka o događajima teljenja određuju se datumi izlaska iz suhostaja. Moguće je da životinja iz nekog razloga (bolest, smrt, prodaja itd.) napusti krdo prije događaja teljenja pri čemu se za datum izlaska iz suhostaja koristi datum odlaska iz krda uzet iz podataka o profilu životinje. Motivacija za izračun vremenske serije

broja krava u suhostaju je povezanost porasta broja takvih krava s padom proizvodnje mlijeka.

Vremenska serija broja krava u vrhuncu proizvodnje

Vrhunac proizvodnje mlijeka kod mliječnih krava nastaje oko 5 do 10 tjedana nakon događaja teljenja. Iz podataka o teljenjima za svaki datum iz vremenske serije o dnevnoj proizvodnji mlijeka izračunava se broj krava koje su se telile prije 5 do 10 tjedana uz uvjet da su na datum određivanja životinje prisutne u krdu. Uvjet se izračunava pomoću datuma odlaska iz krda iz podataka o profilu životinje. Motivacija izračuna vremenske serije broja krava u vrhuncu proizvodnje je povezanost porasta broja takvih krava i porasta proizvodnje.

Vremenska serija broja teljenja

Nakon događaja teljenja krava ulazi u novu laktaciju i započinje proizvoditi mlijeko. Grupiranjem događaja teljenja prema datumu i prebrajanjem jedinstvenih identifikatora dobiva se vremenska serija dnevnog broja teljenja. Motivacija za izračun ove vremenske serije je povezanost ulaska životinja u novu laktaciju i ponovne proizvodnje mlijeka s porastom proizvodnje mlijeka.

Vremenske serije atmosferskih uvjeta

Podaci o atmosferskim uvjetima već su zapisani u obliku vremenskih serija minimalnih i maksimalnih temperatura, relativne vlažnosti i indeksa topline pa ih nije potrebno transformirati.

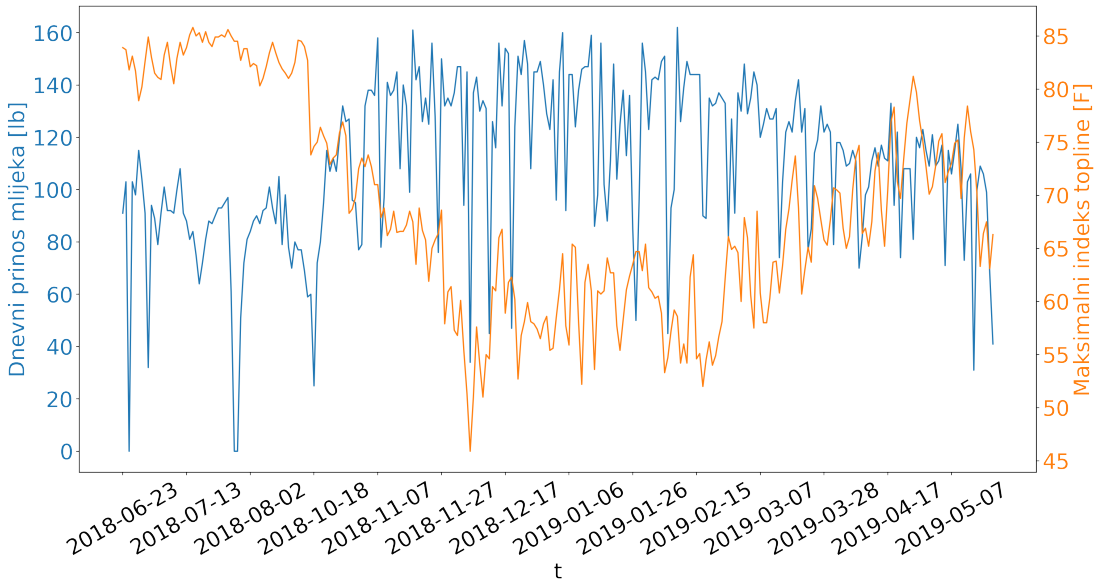
5.3. Prikaz podataka

Podaci o dnevnim prinosima mlijeka mogu se vizualizirati na razini dnevnog prinosa pojedine životinje, na razini dnevnih prinosa svih životinja i zbrojno na razini zbroja dnevnih prinosa mlijeka cijelog krda.

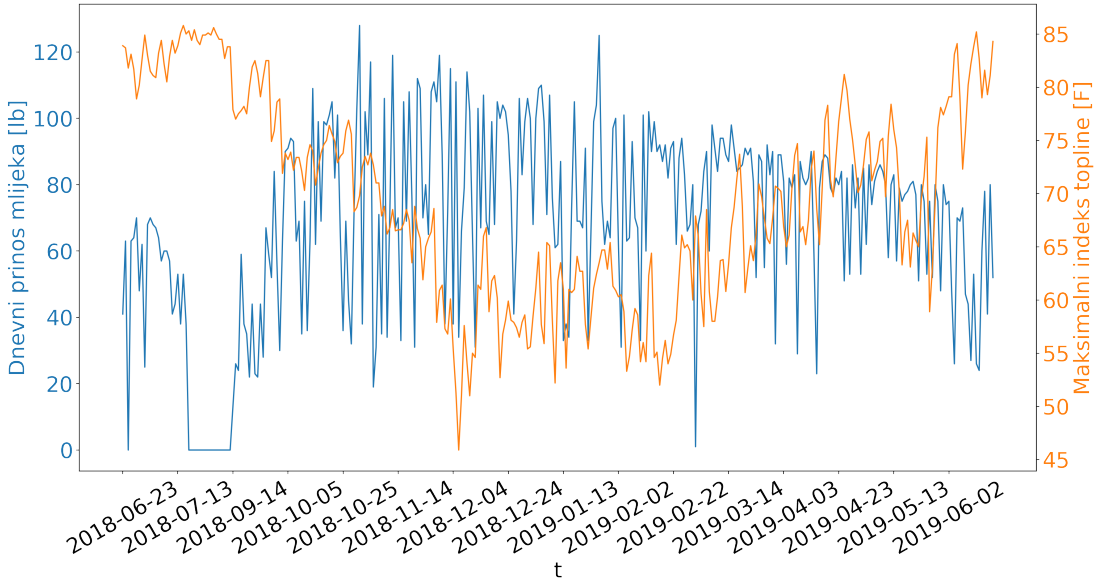
5.3.1. Prikaz na razini dnevnog prinosa pojedine životinje

Slika 5.1 prikazuje vremensku seriju dnevne proizvodnje mlijeka nasumično odabrane životinje i najvećeg indeksa topline. Na slici se uočava negativni utjecaj vremenskih uvjeta na prinos mlijeka jer je prinos veći u razdoblju s manjim indeksom topline. Na

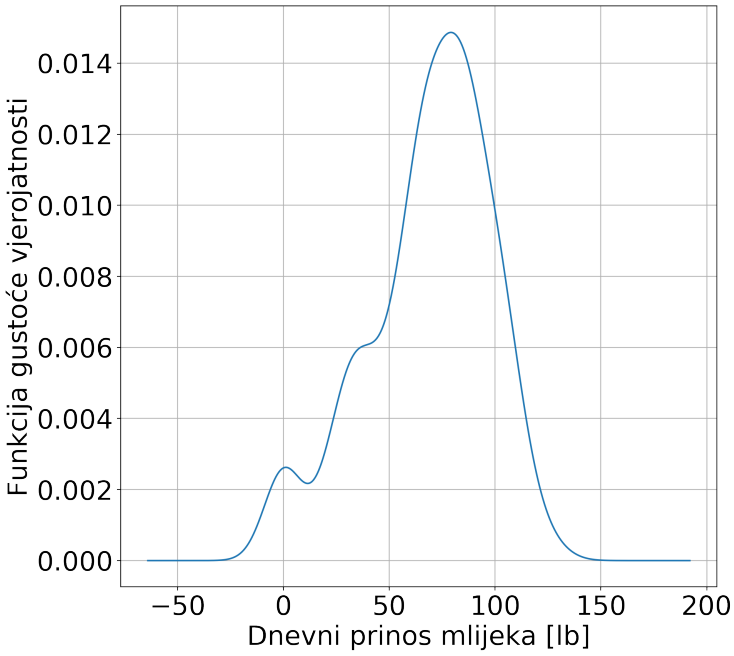
slici 5.2 se za drugu nasumično odabranu životinju prikazuje suhostaj u okvirnom razdoblju od 13.7.2018. do 14.9.2018., nakon kojeg kreće razdoblje povećanja i vrhunca proizvodnje. Nakon razdoblja vrhunca proizvodnje proizvodnja počinje opadati. Slika 5.3 prikazuje funkciju gustoće vjerojatnosti za dnevne prinose nasumično odabrane životinje. Funkcija poprima očekivani oblik jer se dnevni prinosi mliječnih krava najčešće kreću od 60 do 100 lb. Veća gustoća oko vrijednosti 0 lb javlja se zbog suhostaja.



Slika 5.1: Prikaz indeksa topline i dnevnog prinosa mlijeka nasumično odabrane životinje



Slika 5.2: Prikaz indeksa topline i dnevnog prinosa mlijeka nasumično odabrane životinje

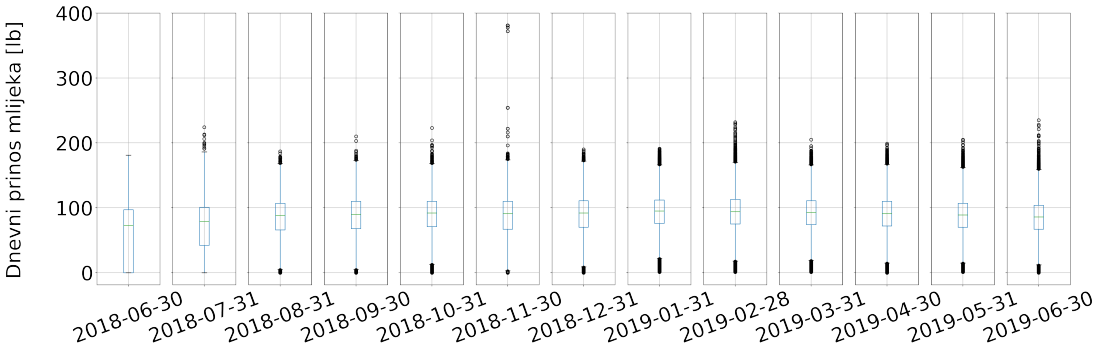


Slika 5.3: Prikaz funkcije gustoće vjerojatnosti dnevnih prinosa mlijeka nasumično odabrane životinje

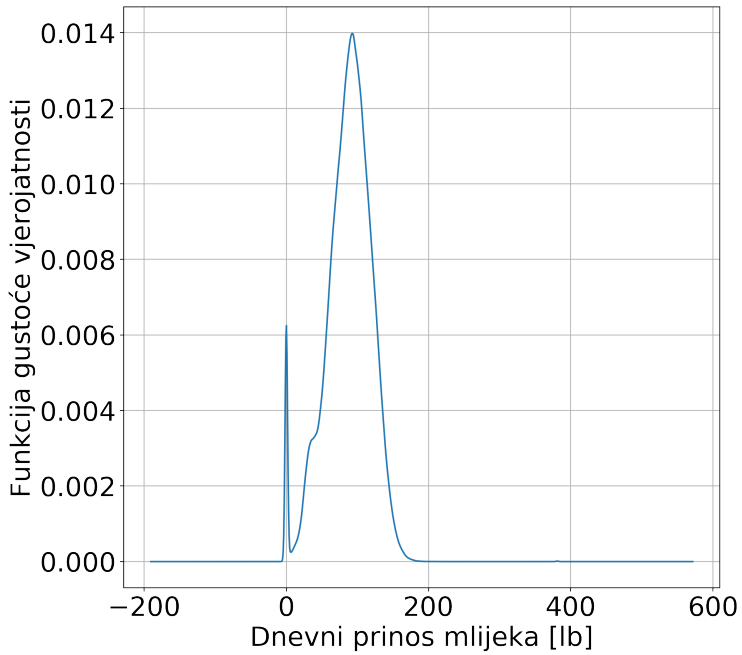
5.3.2. Prikaz na razini dnevnog prinosa svih životinja

Slika 5.4 prikazuje kutijaste dijagrame proizvodnje mlijeka na razini svih životinja gdje su dnevni prinosi grupirani po mjesecima. Kutijasti dijagrami sugeriraju postojanje stršećih vrijednosti u podacima (engl. *outlier*), pogotovo dijagram koji se odnosi na studeni u kojemu se javljaju tri vrijednosti veće od 350 lb mlijeka na dan. Ostali dijagrami također pokazuju postojanje stršećih vrijednosti, ali su one po iznosu manje. Jedna od mogućnosti javljanja stršećih vrijednosti je upisivanje dvostrukih vrijednosti u slučaju da mužnja prethodnog dana nije bila zabilježena.

Slika 5.5 prikazuje funkciju gustoće vjerojatnosti dnevnog prinosa mlijeka. Funkcija gustoće vjerojatnosti iscrtana nad domenom dnevnih prinosa otkriva da je veći broj prinosa oko vrijednosti 0 lb te oko 100 lb. Za vrijednosti oko 0 postoji više mogućih objašnjenja. Prva mogućnost je da te životinje stvarno nisu dale mlijeko jer su iz nekog razloga bile izuzete iz mužnje (suhostaj, bolest, veterinarski pregled i sl.), a druga da se radi o pogreškama u podacima. Vrijednosti oko 100 lb su očekivani dnevni prinosi za muzne krave.



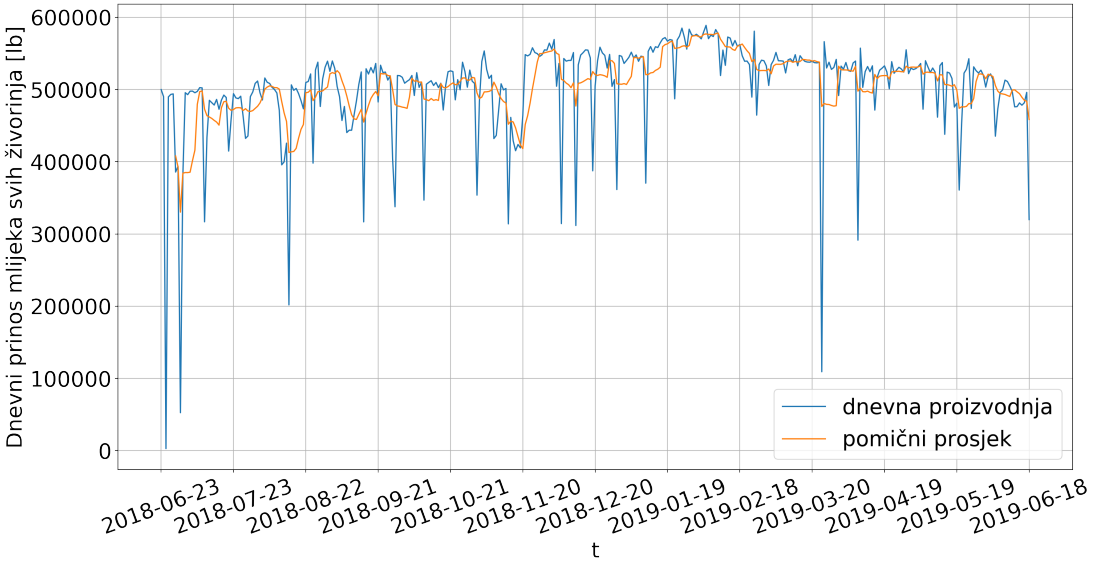
Slika 5.4: Prikaz kutijastog dijagrama dnevnih prinosa mlijeka na mjesečnoj razini svih životinja



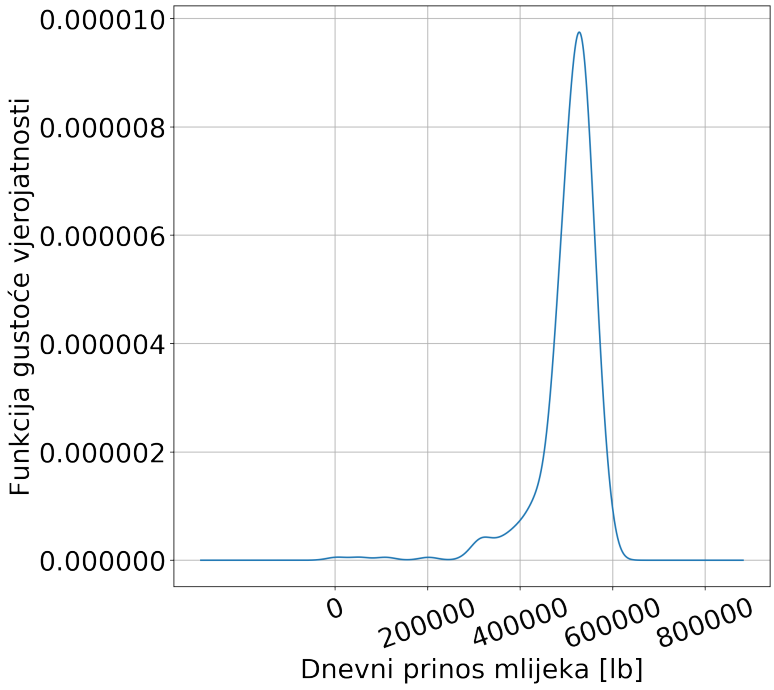
Slika 5.5: Prikaz funkcije gustoće vjerojatnosti dnevnih prinosa mlijeka svih životinja

5.3.3. Prikaz na razini krda

Slika 5.6 prikazuje vremensku seriju dnevne proizvodnje mlijeka na razini krda koja će se predviđati. U prikazu su vidljivi padovi u proizvodnji uzrokovani neuobičajeno malim brojem dnevnih mužnji. Ovi padovi mogu uzrokovati probleme za one klasične statističke metode predviđanja koje uzimaju u obzir samo povijesne vrijednosti vremenske serije, dok algoritmi strojnog učenja iz smanjenih vrijednosti značajke broja dnevnih mužnji mogu zaključiti uzrok pada proizvodnje. Slika 5.7 prikazuje funkciju gustoće vjerojatnosti dnevne proizvodnje mlijeka na razini krda.



Slika 5.6: Prikaz vremenske serije dnevne proizvodnje mlijeka na razini krda



Slika 5.7: Prikaz funkcije gustoće vjerojatnosti dnevne proizvodnje mlijeka na razini krda

5.4. Analiza vremenske serije

5.4.1. Dekompozicija vremenske serije

Nad vremenskom serijom dnevne proizvodnje mlijeka na razini krda i nad diferenciranom vremenskom serijom dnevne proizvodnje mlijeka na razini krda provedena je dekompozicija vremenske serije. Kao što je i bilo za pretpostaviti, ove vremenske serije ne sadrže komponentu sezonalnosti jer predstavljaju proizvodnju i promjenu proizvodnje cijelog krda. Životinje unutar krda se nalaze u različitim životnim ciklusima, na primjer neke životinje su tek počele proizvoditi mlijeko, druge se u vrhuncu proizvodnje, dok su treće u suhostaju. Kada bi se analizirala vremenska serija proizvodnje na razini pojedine životinje tada bi ta vremenska serija sadržavala godišnju sezonalnost.

5.4.2. Testovi stacionarnosti

Nad vremenskom serijom dnevne proizvodnje mlijeka na razini krda i nad diferenciranom vremenskom serijom dnevne proizvodnje mlijeka na razini krda provedeno je statističko testiranje na stacionarnost postupkom opisanim u poglavlju 2.2.5. Korišteni su autoregresijski modeli s konstantnim članom. Testovi KPSS provedeni su uz $95\ \%$ interval pouzdanosti, a testovi ADF uz kritičnu vrijednosti -2.89. Red autoregresijskog modela u testu ADF (engl. *maximum lag*) postavljen je prema (Schwert, 2002). Tablica 5.2 prikazuje rezultate statističkih testova. Oba testa odbacila su međusobno kontradiktorne nulte hipoteze za izvornu vremensku seriju iz čega se ne može zaključiti stacionarnost. Za diferenciranu vremensku seriju test KPSS nije odbacio nultu hipotezu o stacionarnosti, a test ADF odbacio je nultu hipotezu postojanja jediničnog korijena. Zaključak testiranja jest stacionarnost diferencirane vremenske serije zbog čega će se prognozirati ta vremenska serija.

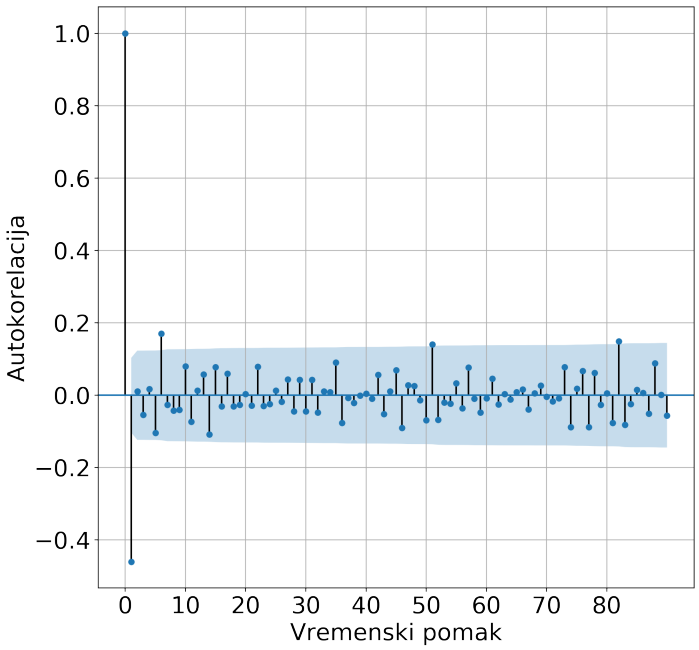
Tablica 5.2: Testne statistike

Osjenčane ćelije označuju odbacivanje nulte hipoteze.

Korak dif. / **Test-parametar**	/	1
KPSS-c	1.9200	0.0327
ADF-c	-4.0034	-7.6985

5.4.3. Autokorelacijska funkcija

Na slici 5.8 prikazana je autokorelacijska funkcija za diferenciranu vremensku seriju dnevne proizvodnje mlijeka na razini krda. Najveća vrijednost vremenske razlike odabrana je prema (Box et al., 2015). Graf sa slike 5.8 pokazuje jaku negativnu korelaciju za vremensku razliku 1 korak i izraženiju pozitivnu korelaciju za vremensku razliku 6 koraka. Ovi rezultati koristit će se pri izgradnji značajki iz zaostale vremenske serije.



Slika 5.8: Prikaz autokorelacijske funkcije za diferenciranu vremensku seriju dnevne proizvodnje mlijeka na razini krda

5.5. Osnovna prognoza

Kako bi se performanse modela strojnog učenja imala s čime usporediti neophodno je napraviti osnovno prognoziranje u koju je svrhu odabrana naivna prognoza te integrirani autoregresijski model pomičnih prosjeka. Vektorska autoregresija je isprobana, ali nije dala zadovoljavajuće rezultate. Razlog leži u pretpostavci modela o jednakoj obostranoj povezanosti parova varijabli (opažanja vremenskih serija). Npr. vremenske serije dnevne proizvodnje mlijeka na razini krda i atmosferskih uvjeta nemaju jednaku ovisnost, jer iako proizvodnja na razini životinje ovisi o vremenskim uvjetima, proizvodnja na razini krda dodatno ovisi o broju muznih krava i dnevnih mužnji. Tako se može dogoditi slučaj u kojem su vremenski uvjeti nepovoljni za proizvodnju mlijeka, a proizvodnja ipak raste zbog povećanja broja muznih krava u krdu.

5.5.1. Naivna prognoza

Naivna prognoza buduću vrijednost prognozira sa zadnjom poznatom vrijednosti. Za k koraka unaprijed u trenutku t prognoza se dobiva izrazom 5.1.

$$\hat{y}(t+i) = y(t),\ i = 1 \dots k \tag{5.1}$$

5.5.2. Integrirani autoregresijski model pomičnih prosjeka

Za odabir optimalnih hiperparametara modela napisana je Jupyter bilježnica koja koristi Pythonov paket `pmdarima` koji implementira učenje i prognoziranje modela ARIMA. Najprije su generirani modeli s različitim hiperparametrima p, d i q nakon čega je provedena inačica unakrsne provjere za vremenske serije nad validacijskim skupom podataka kako je opisano u (Hyndman i Athanasopoulos, 2018). Za svaki korak prognoze odabran je model koji je postigao najmanju mjeru korijena srednje kvadratne pogreške (*engl. Root Mean Squared Error*, RMSE) na validacijskom skupu, koji je ponovno naučen nad unijom skupa za učenje i validaciju te ispitan nad, neviđenim, ispitnim skupom. Generirani hiperparametri p i q su zbog oblika autokorelacijske funkcije postavljeni u rasponu od 0 do 6, a hiperparametar d je zbog rezultata testova stacionarnosti postavljen na vrijednost 1.

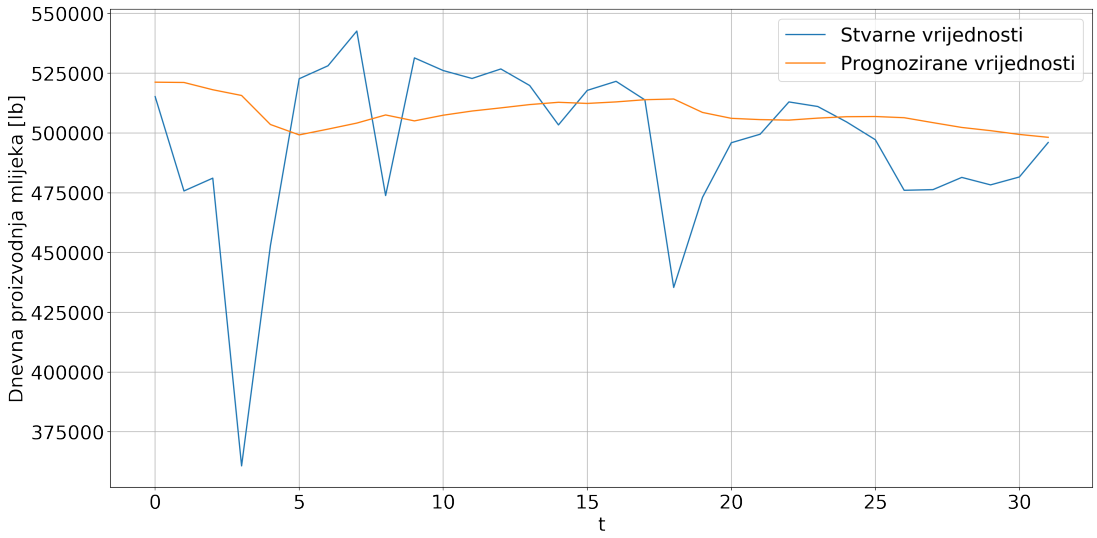
5.5.3. Rezultati osnovnih modela

Tablica 5.3 prikazuje performanse osnovnih modela na ispitnom skupu sačinjenom od posljednjeg 31 opažanja. Iz tablice je vidljivo kako performanse oba modela opadaju kako se k povećava te da performanse brže opadaju kod naivnog modela, nego kod modela ARIMA. Model ARIMA postigao je bolje rezultate u mjerama RMSE i koeficijenta determinacije (R^2) od naivnog modela za sve korake, a za mjeru srednje kvadratne pogreške (*engl. Mean Absolute Error*, MAE) u svim koracima, osim za korak $k = 1$. Mjere R^2 su negativne što govori da osnovni modeli ne objašnjavaju dobro promjenu budućih vrijednosti. Ćelije obojene u zeleno označavaju bolje vrijednosti s obzirom na modele.

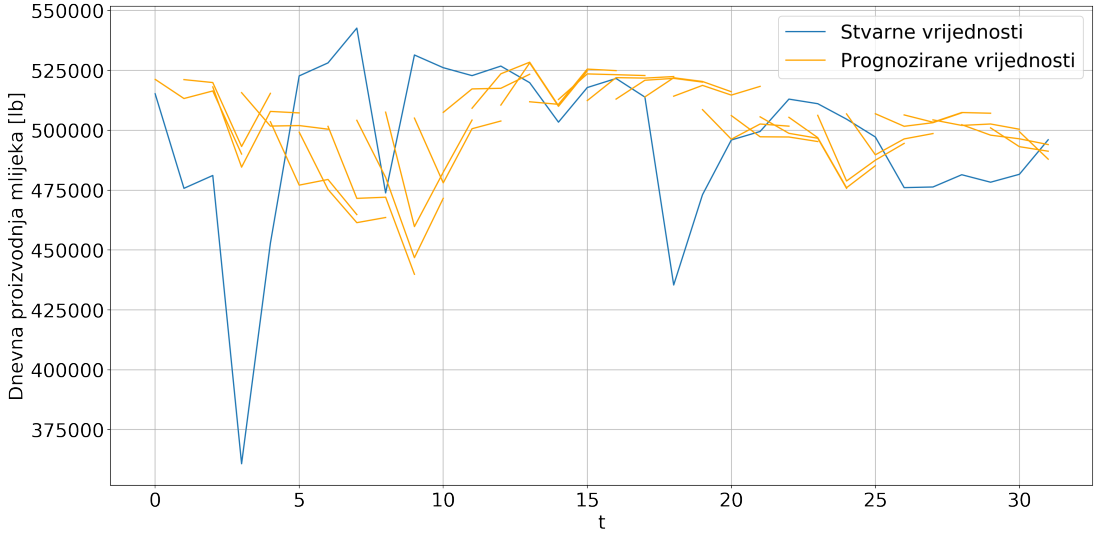
Tablica 5.3: Performanse osnovnih modela na ispitnom skupu

k	Model	MAE	RMSE	R^2
1	Naivni model	23813	38527	-0.258
	ARIMA$(0,1,1)$	24983	38084	-0.229
2	Naivni model	32212	48254	-0.932
	ARIMA$(5,1,0)$	28743	40706	-0.375
3	Naivni model	35897	54092	-1.373
	ARIMA$(5,1,0)$	30267	42866	-0.4905
4	Naivni model	37014	56006	-1.474
	ARIMA$(5,1,0)$	30943	43469	-0.490

Slike 5.9 i 5.10 prikazuju grafove stvarnih i prognoziranih vrijednosti za prognoze jedan i do četiri koraka unaprijed dobivene pomoću odabranih osnovnih modela. Na slici 5.9 je vidljivo kako osnovni model pravovremeno ne predviđa promjene vremenske serije.



Slika 5.9: Prikaz osnovne prognoze modelom ARIMA za jedan korak unaprijed



Slika 5.10: Prikaz osnovne prognoze dobivene pomoću četiri modela ARIMA za četiri koraka unaprijed

5.6. Prognoziranje metodama strojnog učenja

5.6.1. Korišteni modeli

Značajke

Potencijalne značajke modela dobivene su kao zaostale vrijednosti vremenskih serija iz poglavlja 5.2.1 uz iznimku vremenskih serija atmosferskih uvjeta koje su pokazale visoku međusobnu korelaciju (slika 5.11). Visoko korelirane značajke ne pridonose performansama modela pa su korištene vremenske serije najmanjeg i najvećeg indeksa topline jer one objedinjuju informaciju o temperaturi i relativnoj vlažnosti. Broj koraka zaostajanja definira se za svaku vremensku seriju zasebno prilikom pokretanja programa.

Uz zaostale vrijednosti korištena je značajka rednog broja dana u godini koja je zbog cikličnosti pretvorena u Fourierove članove prema izrazima:

$$sin\left(\frac{t \cdot 2 \cdot k \cdot \pi}{T}\right) \tag{5.2}$$

$$cos\left(\frac{t \cdot 2 \cdot k \cdot \pi}{T}\right), \tag{5.3}$$

gdje je k redni broj Fourierovog člana, t ciklička varijabla (dan u godini), a T period (365).

	min_temp	max_temp	min_rh	max_rh	min_heat_index	max_heat_index
min_temp	1	0.775888	-0.472123	-0.638714	0.998031	0.795572
max_temp	0.775888	1	-0.821147	-0.412706	0.760356	0.993566
min_rh	-0.472123	-0.821147	1	0.357084	-0.460225	-0.770806
max_rh	-0.638714	-0.412706	0.357084	1	-0.66762	-0.404349
min_heat_index	0.998031	0.760356	-0.460225	-0.66762	1	0.780205
max_heat_index	0.795572	0.993566	-0.770806	-0.404349	0.780205	1

Slika 5.11: Korelacija vremenskih serija atmosferskih uvjeta
Intenzivnija crvena boja označava jaču pozitivnu, a intenzivnija plava jaču negativnu korelaciju

Prilikom učenja i ispitivanja modela značajke su zbog različitih skala standardizirane na srednju vrijednost nula i jediničnu varijancu. Za procjenu srednje vrijednosti i standardne devijacije pojedinih značajki korišten je isključivo skup za učenje jer bi korištenjem skupa za učenje i ispitivanje došlo do curenja podataka.

Radni okvir Apache Spark ne podržava automatski odabir kontinuiranih značajki, zbog čega je odabir značajki napravljen zajedno s odabirom hiperparametara modela pretraživanjem po rešetci. Nedostatak automatskog odabira kontinuiranih značajki rezultira većom vremenskom složenosti što je pokazano primjerom 5.6.1.

Algoritmi

Podržani algoritmi su linearna regresija, slučajne šume i gradijentno ojačana stabala, čiji se rasponi hiperparametara zadaju pri pokretanju programa. Detaljan opis zadavanja koraka zaostajanja za dobivanje značajki te raspona hiperparametara algoritama nalazi se u poglavlju 5.7.

Modeli kandidati

Kandidati za odabir dobiveni su kombinacijom značajki, algoritama i hiperparametara čime su dobiveni modeli različite složenosti nad kojima je provedena pretraga optimalnog modela po rešetci. To je pretraga u prostoru s $n \cdot m$ dimenzija, gdje je n broj kombinacija modela dobivenih kombiniranjem algoritama i hiperparametara, a m broj kombinacija značajki. Broj kombinacija modela n raste eksponencijalno s obzirom na broj zadanih hiperparametara. Npr. za algoritam koji sadrži N hiperparametara, a svaki hiperparametar poprima $M_i,\ i = 1 \ldots N$ mogućih vrijednosti dobiva se $n = \prod_{i=1}^{N} M_i$ kombinacija hiperparametara iz kojih treba odabrati optimalne hiperparametre.

Isti problem javlja se kod kombinacija zaostalih vrijednosti vremenskih serija jer se za svaku vremensku seriju definira raspon najvećih zaostajanja koja se koriste kao potencijalne značajke. Tako dobiveni prostor ima previše dimenzija za optimizaciju pretraživanjem po rešetci. Zato su vremenske serije grupirane u logičke cjeline. Vremenske serije dnevne proizvodnje mlijeka, broja dnevnih mužnji i muznih krava grupirane su u logičku cjelinu koja se odnosi na informacije o produktivnosti krda. Vremenske serije broja krava u suhostaju, u vrhuncu proizvodnje te broja teljenja grupirane su u cjelinu koja se odnosi na informacije o stanju krda, dok vremenske serije atmosferskih uvjeta čine treću grupu značajki.

Primjer 5.6.1. Problem visoke dimenzionalnosti prostora pretrage.

Neka su za pretragu odabrani algoritmi, hiperparametri i značajke prikazane tablicom 5.4. U tom slučaju broj kombinacija algoritama i hiperparametara je $2 + 2 \cdot 2 \cdot 2 \cdot 1 \cdot 2 = 2 + 2^4 = 18$. Broj kombinacija značajki je $3 \cdot 3 = 9$. Time se dobiva $18 \cdot 9 = 162$ modela. Ovim primjerom pokazano je kako i za manji broj kombinacija hiperparametara i značajki dolazi do velikog broja modela za provjeru.

Tablica 5.4: Hiperparametri i značajke za pretragu

Naziv	**Zadane vrijednosti**
Linearna regresija - `regParam`	0.0, 1.5
Slučajne šume - `maxBins`	32, 64
Slučajne šume - `maxDepth`	5, 10
Slučajne šume - `minInstancesPerNode`	2, 5
Slučajne šume - `numTrees`	100
Slučajne šume - `subsamplingRate`	0.6, 0.8
Najveći korak zaostajanja vremenskih serija produktivnosti	3, 4, 5
Najveći korak zaostajanja vremenskih serija atmosferskih uvjeta	1, 2, 3

5.6.2. Odabir modela

Cilj je odabrati algoritam s onim hiperparametrima i značajkama za koje se postiže najbolja generalizacija.

Postupak odabira i evaluacije modela

Ostvarene su dvije vrste odabira modela. Prva se temelji na tzv. ispitivanju modela s fiksnom točkom izvorišta, a druga na tzv. unakrsnoj provjeri za vremenske serije (ispitivanje s pomičnom točkom izvorišta). Obje metode opisane su u (Hyndman i Athanasopoulos, 2018). Prvi postupak manje je pouzdan od drugoga koji oponaša produkcijske uvjete, ali je vremenski višestruko jeftiniji jer se u drugom model uči i ispituje za svaki ispitni primjer.

Podaci su podijeljeni u tri disjunktna skupa (učenje, validacija, ispitivanje), vodeći računa o vremenskom uređaju podataka. Skup za učenje i validaciju služio je za odabir modela, a skup za ispitivanje za pravednu evaluaciju modela. Kada bi se model evaluirao na validacijskom skupu to ne bi bilo pravedno jer je odabran onaj model koji je najbolji baš na tim podacima. Prilikom pretrage najboljeg modela korišteno je ispitivanje modela s fiksnom točkom uz skup za učenje i skup za validaciju. Zatim se najbolji model na validacijskom skupu ponovno naučio nad unijom skupa za učenje i validaciju te je primijenjeno ispitivanje s pomičnom točkom izvorišta na ispitnom skupu čime su dobivene performanse modela najsličnije stvarnim uvjetima produkcije. Implementacija podržava postavljanje kriterija (MAE, RMSE i R^2) po kojem se bira najbolji model na skupu za validaciju, a korišten je korijen srednje kvadratne pogreške.

5.6.3. Prognoza više koraka unaprijed

Implementacija podržava prognozu više koraka unaprijed pri čemu se za svaki korak prognoziranje radi zasebnim modelom. Time se postiže specijalizacija modela za predviđanje k koraka unaprijed. Kao i kod prognoziranja jednog dana unaprijed, za k koraka je provedeno statističko testiranje stacionarnosti vremenskih serija. Testovi su provedeni nad vremenskim serijama diferenciranim za korak k, gdje diferencirana vrijednost predstavlja razliku trenutno promatranog dana i vrijednosti udaljene za k dana u budućnost. Testovi potvrđuju stacionarnost diferenciranih vremenskih serija dnevne proizvodnje mlijeka do $k = 7$ što je dovoljno jer porastom broja koraka prognoze postaju nepouzdanije pa nema smisla prognozirati daleko u budućnost. Kako bi se spriječilo curenje podataka značajke dobivene zaostajanjem vremenskih serija kreću od koraka zaostajanja k.

5.6.4. Rezultati modela strojnog učenja

Performanse modela

Tablica 5.5 prikazuje performanse odabranih modela za prognozu k koraka unaprijed na ispitnom skupu sačinjenom od posljednjeg 31 opažanja. Za svaki model navedeni su korišteni hiperparametri i značajke istim redom kako su opisani u nastavku. Linearna regresija skraćeno je zapisana kao LR uz hiperparametre korištenja pomaka po y-osi, faktora regularizacije, udjela L_1 i L_2 regularizacije (engl. *elastic net*) te korištenja standardizacije značajki. Algoritam stabla odluke sadrži parametre broja pretinaca, najveće dubine stabla i najmanjeg broja primjera u listovima. Algoritam slučajnih šuma skraćeno je zapisan kao RF uz hiperparametre stabla odluke, broja stabala, strategije odabira podskupa značajki i postotka primjera za uzorkovanje s ponavljanjem iz skupa za učenje. Algoritam gradijentno ojačanih stabala skraćeno je zapisan kao GBT uz hiperparametre stabla odluke, strategije odabira podskupa značajki, postotka primjera za uzorkovanje bez ponavljanja iz skupa za učenje, stope učenja te najvećeg broja iteracija. Binarne značajke predstavljene su simbolima T i F sa značenjem da se koriste, odnosno ne koriste. Značajke su navedene redoslijedom: najveći broj koraka zaostajanja za grupe vremenskih serija produktivnosti krda, stanja krda, atmosferskih uvjeta te broj Fourierovih članova za dan u godini.

Ćelije u stupcima mjera MAE, RMSE i R^2 obojene u zelenu boju označuju da je model postigao bolji rezultat od osnovnog modela odabranog po zadanoj mjeri (zelena ćelija u stupcu zadane mjere u tablici 5.3), a crvena označava suprotno. Ćelije u stupcu modela obojene u zeleno označavaju odabrani model strojnog učenja. Rezultati pokazuju kako je prognoziranje metodama strojnog učenja dalo bolje rezultate po mjeri RMSE za prva tri koraka, dok su u četvrtom koraku klasične metode bile bolje. U svim koracima prognoze predstavnici ansambala daju bolje rezultate od linearne regresije. Iz korištenih značajki je vidljivo kako su preferirani jednostavniji modeli s manjim brojem značajki. Značajka transformiranog dana u godini nije korištena u niti jednom modelu te se pokazala nebitnom. U tablici se uočava da su modeli strojnog učenja češće bolji po mjerama RMSE i R^2, nego po mjeri MAE. Zanimljiv je slučaj prognoze za korak $k = 3$ u kojem su mjere MAE bolje nego za prva dva koraka prognoziranja. Odabrani modeli za svaki korak prognoziranja uz najmanju mjeru RMSE imaju i najveću mjeru R^2 što potvrđuje da mogu najbolje objasniti promjene u podacima.

Modeli linearne regresije za sve korake prognoze koriste L_2 regularizaciju. Modeli stabla odluke preferiraju stabla različitih dubina, ali se primjećuje kako su za korake $k = 1, 2, 3$ odabrani modeli s većim omjerom hiperparametara najmanjeg broja pri-

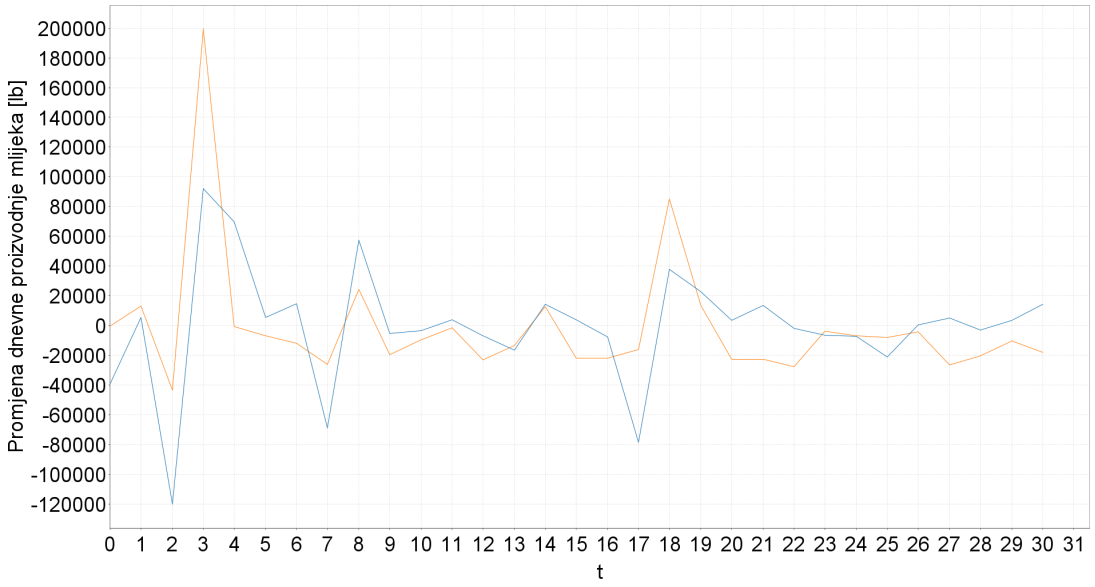
mjera u listovima i dubine stabla u usporedbi s korakom $k = 4$. Kod modela gradijentno ojačanih stabala taj slučaj se javlja za korake $k = 1, 3, 4$ u kojima je omjer navedenih hiperparametara veći od omjera za korak $k = 2$. Gledajući modele gradijentno ojačanih stabala mjera RMSE je za korak $k = 2$ najveća, a mjera R^2 najmanja što potvrđuje da taj model slabije generalizira od ostalih zbog veće složenosti.

Tablica 5.5: Performanse odabranih modela na ispitnom skupu

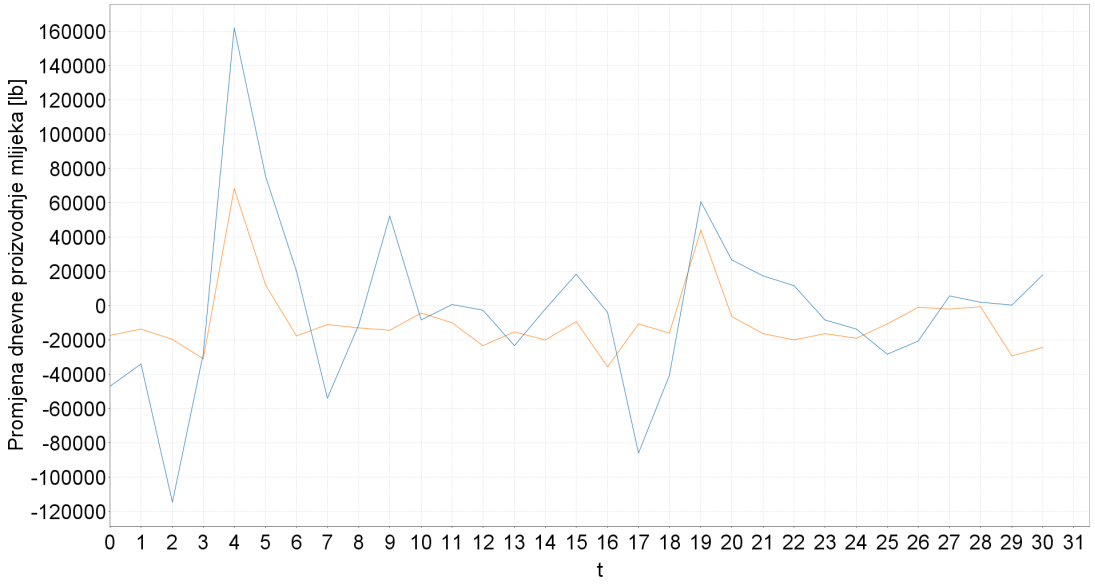
k	Model	Značajke	MAE	RMSE	R^2
1	LR(T, 2, 1, T)	2, 1, 1, 0	35097	44507	−0.295
	RF(64, 7, 7, 50, auto, 0.85)	1, 0, 0, 0	29617	37142	0.098
	GBT(64, 5, 7, auto, 0.8, 0.05, 20)	2, 0, 0, 0	26644	36281	0.140
2	LR(T, 4, 1, T)	3, 2, 2, 0	50232	59903	−0.544
	RF(32, 7, 5, 50, auto, 0.75)	2, 0, 0, 0	29797	38913	0.349
	GBT (64, 7, 3, auto, 0.85, 0.1, 25)	2, 0, 2, 0	36817	48828	−0.026
3	LR(T, 4, 1, T)	3, 3, 0, 0	31058	40102	0.445
	RF(32, 5, 9, 50, auto, 0.85)	3, 4, 0, 0	26232	39172	0.470
	GBT(32, 5, 9, auto, 0.8, 0.1, 25)	4, 4, 4, 0	29628	39850	0.452
4	LR(T, 3.5, 1, T)	4, 5, 4, 0	42226	50276	0.167
	RF(64, 9, 5, 50, auto, 0.85)	4, 4, 4, 0	31844	43673	0.372
	GBT(64, 5, 7, auto, 0.8, 0.05, 25)	4, 0, 4, 0	36654	47664	0.252

Grafovi prognoziranih vrijednosti

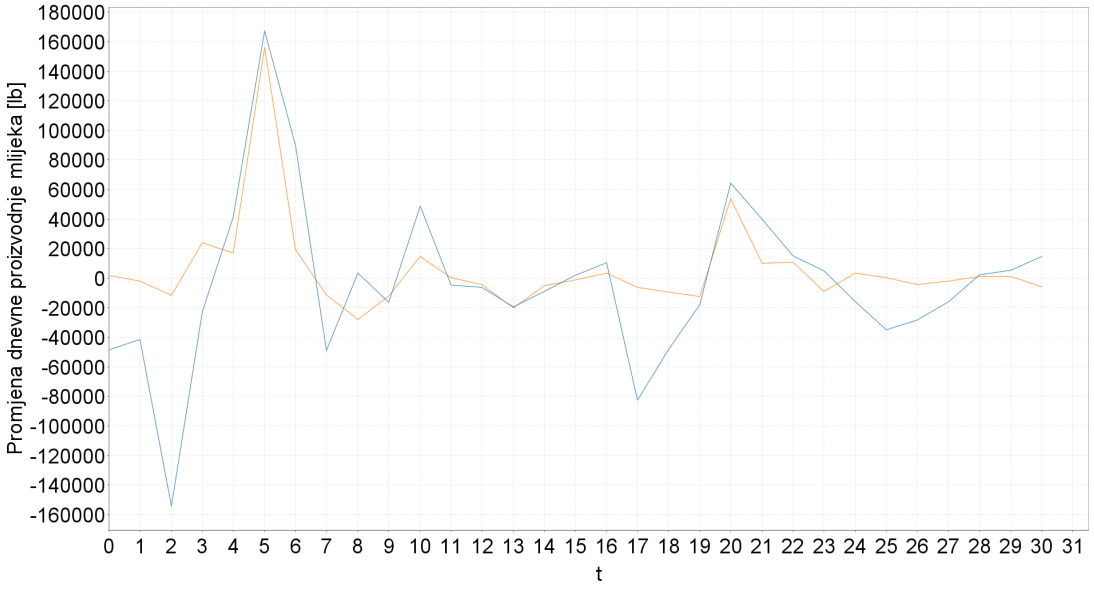
Slike 5.12,5.13, 5.14 i 5.15 prikazuju grafove prognoziranih vrijednosti promjene dnevne proizvodnje mlijeka do četiri koraka unaprijed. Plavom linijom iscrtane su stvarne, a narančastom prognozirane vrijednosti. Iz slika je vidljivo da su prognoze za prva tri koraka točnije od prognoze za četvrti korak te da za prva tri koraka modeli pravovremeno predviđaju promjene vrijednosti. Pravovremenost predviđanja promjene za prva tri koraka čini odabrane modele bolje od osnovnih modela. Kod prognoziranja vremenskih serija često se javlja problem kopiranja vrijednosti kada se model zbog prenaučenosti ponaša slično naivnom modelu. Takvo ponašanje nije prisutno u prognozama za prva tri koraka.



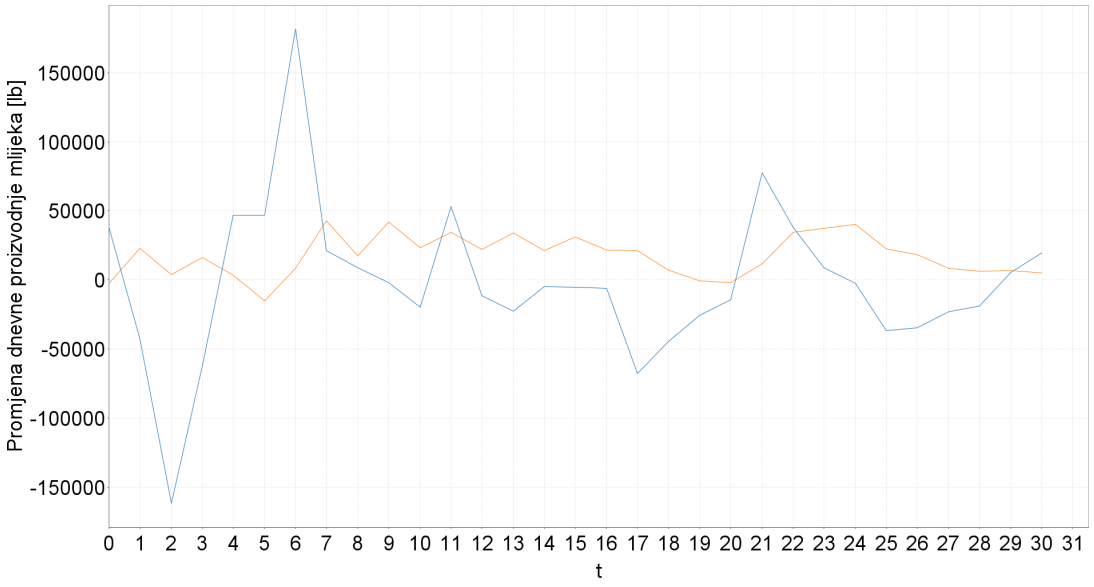
Slika 5.12: Prognoza promjene dnevne količine mlijeka za jedan korak unaprijed



Slika 5.13: Prognoza promjene dnevne količine mlijeka za dva koraka unaprijed



Slika 5.14: Prognoza promjene dnevne količine mlijeka za tri koraka unaprijed

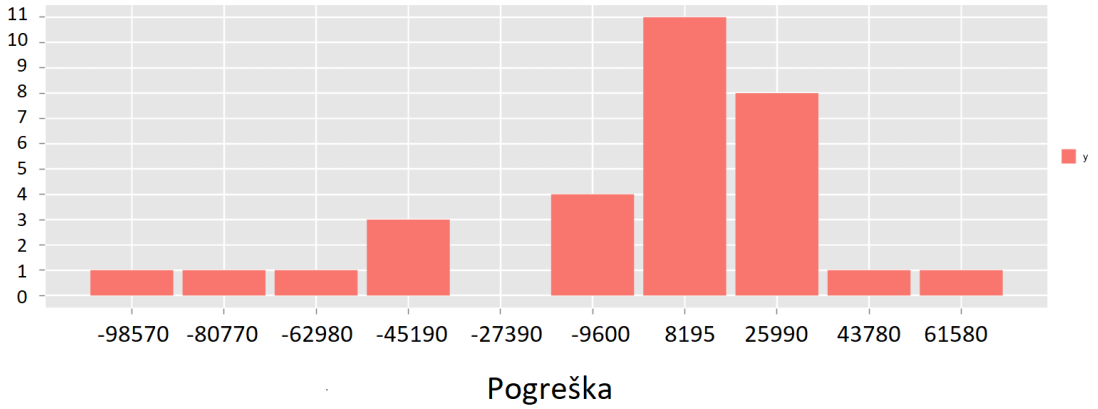


Slika 5.15: Prognoza promjene dnevne količine mlijeka za četiri koraka unaprijed

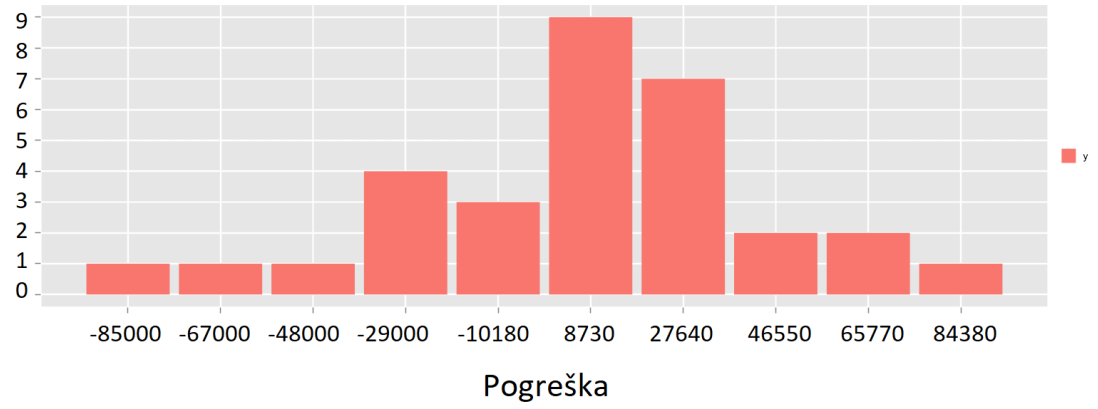
Analiza pogreške

Uz zahtjev minimizacije prognostičke pogreške poželjno je da se pogreške modela ravnaju po normalnoj razdiobi oko srednje vrijednosti 0 i sa što manjom varijancom. Analiza je provedena iscrtavanjem histograma pogrešaka odabranog modela nad ispitnim skupom. Pogreška je izračunata kao razlika stvarne i prognozirane vrijednosti. Za prvi i drugi korak prognoze histogram je centriran oko pozitivne vrijednosti s većim frekvencijama na pozitivnoj strani osi x što znači da model češće podcjenjuje. Odabrani

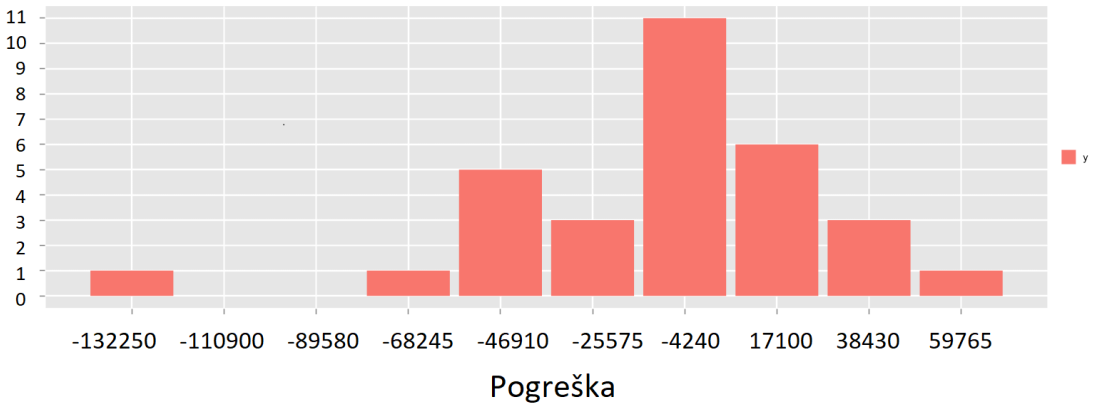
model za treći korak prognoze centriran je oko negativne vrijednosti, a na histogramu se uočava velika pogreška precjenjivanja. Kod modela za četvrti korak prognoze javlja se jače precjenjivanje, ali i dvije velike pogreške na suprotnim krajevima histograma.



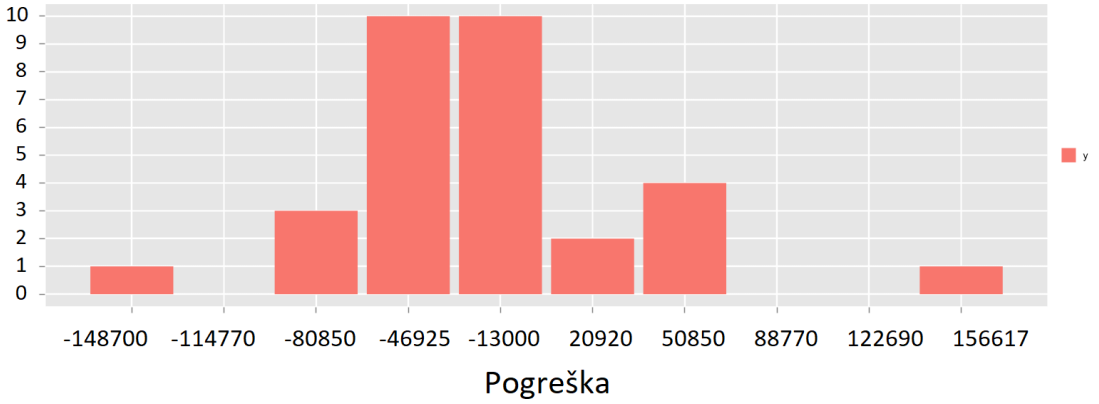
Slika 5.16: Histogram pogreške za prognozu jedan korak unaprijed



Slika 5.17: Histogram pogreške za prognozu dva koraka unaprijed

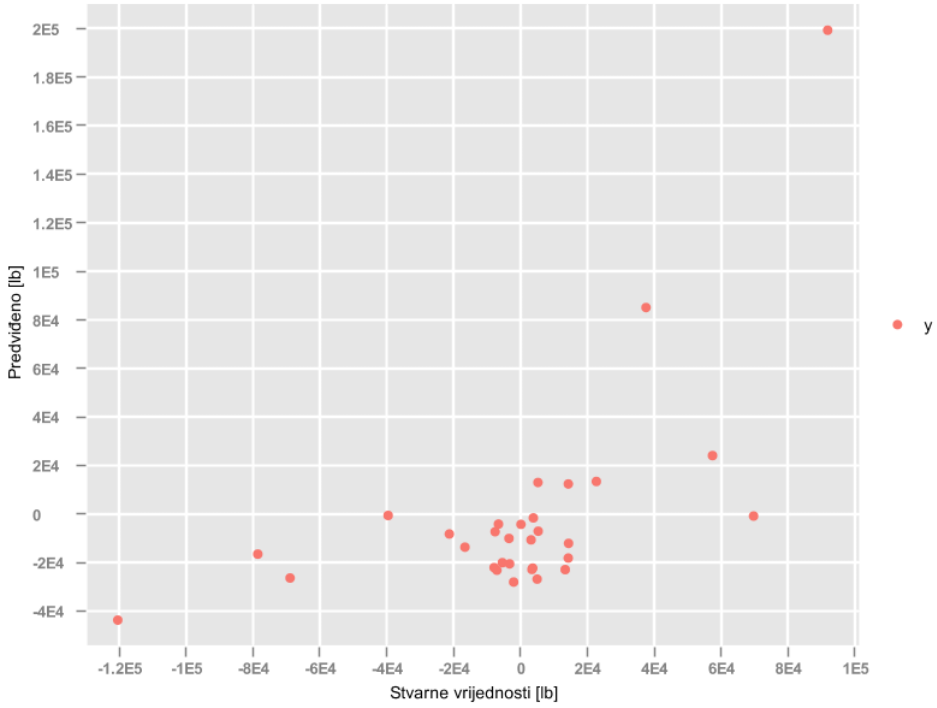


Slika 5.18: Histogram pogreške za prognozu tri koraka unaprijed

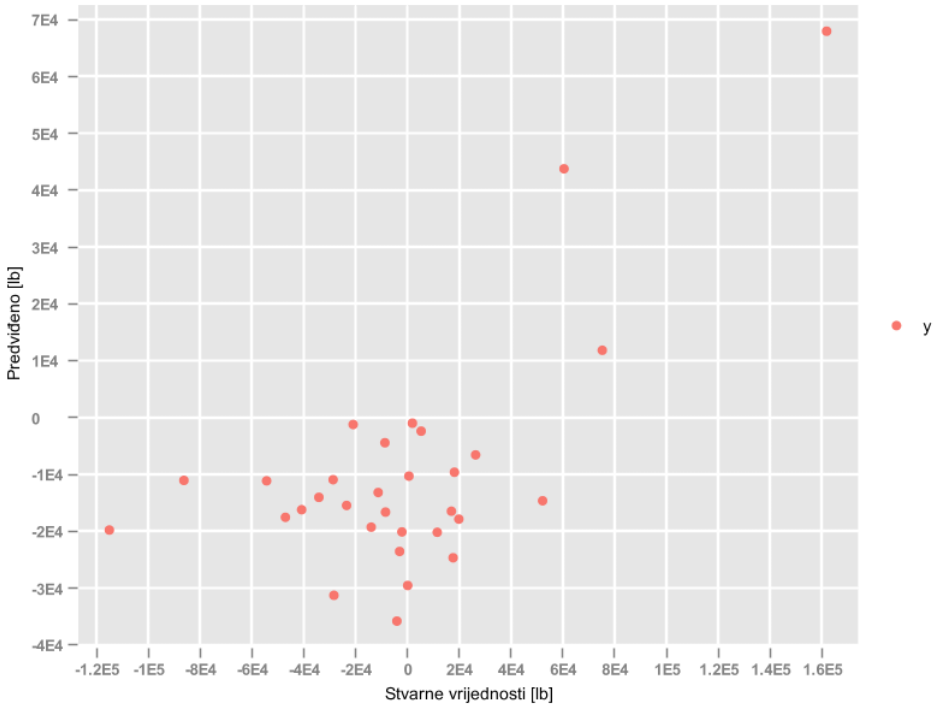


Slika 5.19: Histogram pogreške za prognozu četiri koraka unaprijed

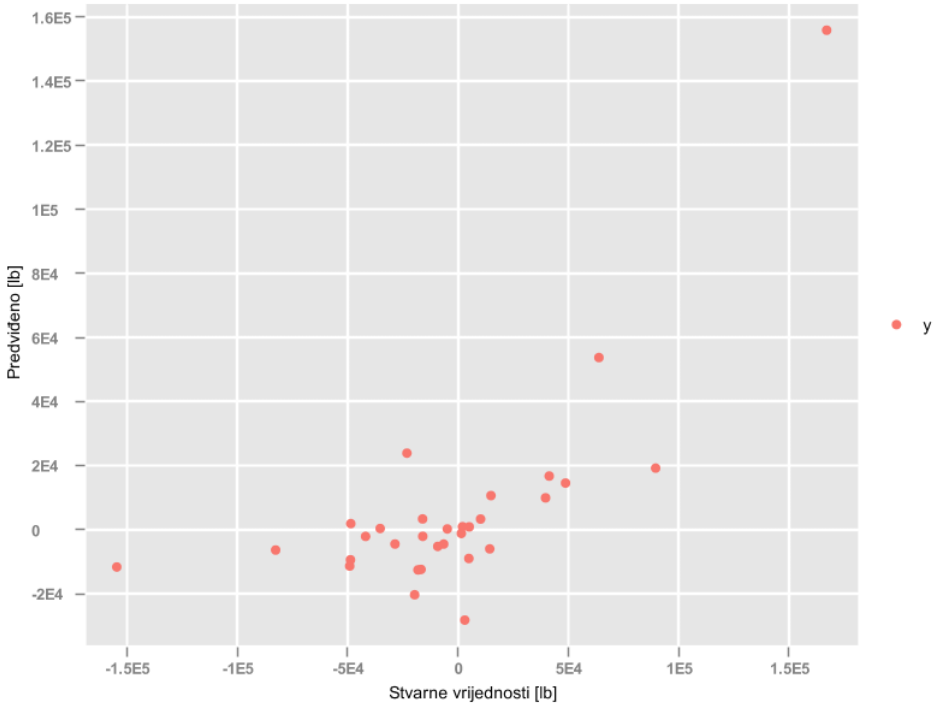
Dodatno su iscrtane točke na grafu stvarnih i predviđenih vrijednosti. U idealnom slučaju točke leže na pravcu $y = x$ ili su slučajno raspoređene oko tog pravca sa što manjim odstupanjem (normalna razdioba pogreške). Vrijednosti koje se nalaze iznad pravca $y = x$ odgovaraju precjenjivanju što je posebno vidljivo na slici 5.23, dok vrijednosti ispod osi odgovaraju podcjenjivanju.



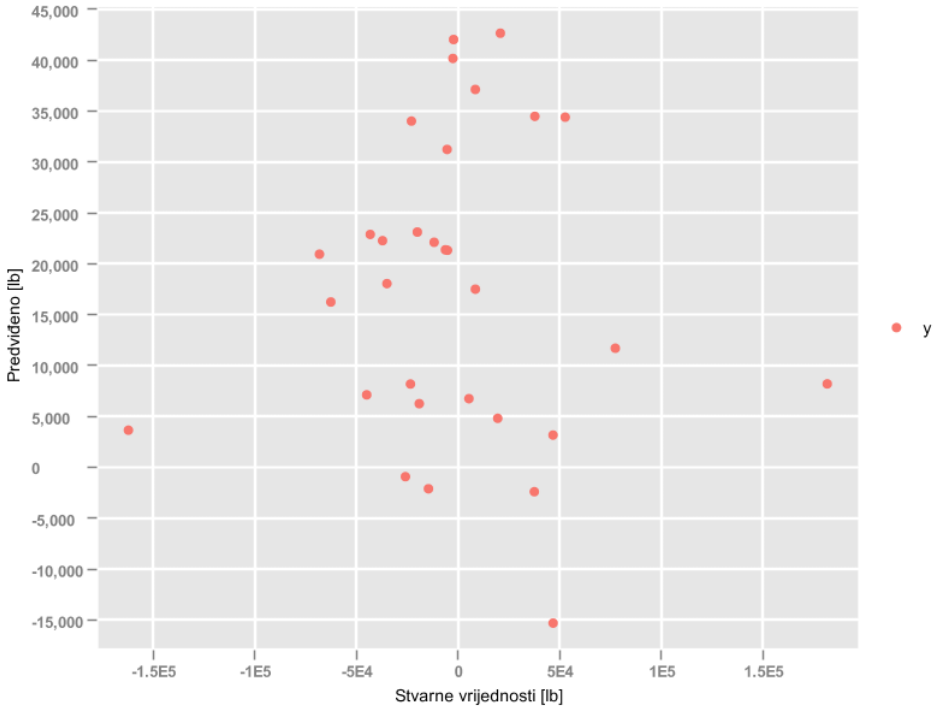
Slika 5.20: Točkasti prikaz stvarnih i prognoziranih vrijednosti za jedan korak unaprijed



Slika 5.21: Točkasti prikaz stvarnih i prognoziranih vrijednosti za dva koraka unaprijed



Slika 5.22: Točkasti prikaz stvarnih i prognoziranih vrijednosti za tri koraka unaprijed



Slika 5.23: Točkasti prikaz stvarnih i prognoziranih vrijednosti za četiri koraka unaprijed

Moguća poboljšanja

Iz rezultata, grafova prognoze i analize pogreške vidljivo je kako ima prostora za poboljšanja. Mjera R^2 pokazuje kako modeli ne mogu objasniti sve promjene vremenske serije. Poboljšanja mjere R^2 i ostalih mjera ostvariva su izgradnjom bitnih značajki koje bi bolje objasnile promjene u podacima. Za izgradnju takvih značajki potrebno je dublje domensko znanje o procesu proizvodnje mlijeka te dodatni podaci o životinjama.

5.7. Upute za pokretanje

Konzolna aplikacija ostvaruje pretragu za modelom koji na validacijskom skupu najbolje generalizira te evaluaciju tog modela na ispitnom skupu. Tijekom rada aplikacije na standardnom izlazu može se pratiti napredovanje pretrage. Programsko rješenje nudi mogućnost korisniku da s odabranim modelom napravi vlastite prognoze k koraka unaprijed.

Za uspješno pokretanje najprije je potrebno postaviti datoteku formata JAR na datotečni sustav računala s kojeg se aplikacija pokreće te ulazne datoteke na raspodijeljeni datotečni sustav instaliran na računalnom grozdu. Ulazne datoteke postav-

ljaju se u direktoriji `korisnik/data/raw`, gdje `korisnik` označava direktoriji korisnika računalnog grozda koji pokreće konzolnu aplikaciju. Nazivi ulaznih datoteka su: `daily_milk.csv`, `fresh_events.csv`, `lactation_data.csv`, `profile_data.csv`, `user_defined_events.csv` i `weather.csv`.

Zatim se konzolna aplikacija pokreće naredbom `spark-submit` pri čemu se koristi pretpostavljeni argument `deploy-mode client`. Argument `driver-memory` i `executor-memory` odnose se na naredbu `spark-submit` i služe postavljanju alocirane radne memorije u grozdu. Argumenti nakon datoteke formata JAR odnose se na argumente konzolne aplikacije, a njihovi nazivi navedeni su u primjeru pokretanja aplikacije na kraju poglavlja.

Argument `nForecasts` postavlja se na cjelobrojnu vrijednost sa značenjem koliko dana unaprijed se prognozira. Argument `diffSeries` navodi se u slučaju diferenciranja vremenske serije. Argument `fourierTermsMaxCount` označava koliko Fourierovih članova se koristi za transformaciju značajke dana u godini. Ako je postavljen na 0 značajka se ne koristi.

Argumenti vezani uz hiperparametre algoritama imaju slične nazive kao hiperparametri opisani u poglavlju 4.2. Svaki brojčani hiperparametar algoritma `x` predstavljen je s trojkom (`xLow`, `xHigh`, `xStep`) pomoću koje se definira raspon tog hiperparametra. Ako su donja i gornja vrijednost iste, vrijednost koraka se ignorira. Hiperparametre koji poprimaju binarne vrijednosti (`lrStandardize` i `lrFitIntercept`) potrebno je ili navesti sa značenjem da se koriste ili ispustiti sa značenjem da se ne koriste.

Argumenti `x` vezani za kombinacije značajki definirani su parovima (`xMaxLagLow`, `xMaxLagHigh`) koji služe generiranju značajki zaostalih vremenskih serija. Vremenske serije zaostaju se počevši od koraka prognoziranja $k, k = 1 \ldots$ `nForecasts` (zbog razloga opisanog u poglavlju 5.6.3) sve do koraka *xMaxLag*, *xMaxLag* = `xMaxLagLow` $\ldots$ `xMaxLagHigh`. Npr. za $k = 2$ i (`xMaxLagLow` $= 5$, `xMaxLagHigh` $= 6$) generiraju se kombinacije zaostajanja neke vremenske serije za korake: $\{2, 3, 4, 5\}$ i $\{2, 3, 4, 5, 6\}$. Korištenjem para vrijednosti (`xMaxLagLow`, `xMaxLagHigh`) postiže se ispitivanje manjeg broja kombinacija jer se neke kombinacije preskaču (za gornji primjer zaostajanje za $\{2\}$, $\{2, 3\}$ i $\{2, 3, 4\}$ koraka).

Isječak koda 5.1: Primjer pokretanja konzolne aplikacije za pretragu najboljeg modela

```
spark-submit
--driver-memory 5G
--executor-memory 5G
```

```
DiplomskiRad-assembly-0.1.jar
--nForecasts 4
--diffSeries
--lrRegLow 0.0
--lrRegHigh 2.0
--lrRegStep 0.5
--lrElNetLow 0.0
--lrElNetHigh 1.0
--lrElNetStep 0.5
--lrStandardize
--lrFitIntercept
--decTreeMaxBinsLow 64
--decTreeMaxBinsHigh 64
--decTreeMaxBinsStep 32
--decTreeMaxDepthLow 5
--decTreeMaxDepthHigh 10
--decTreeMaxDepthStep 5
--decTreeMinInstancesPerNodeLow 5
--decTreeMinInstancesPerNodeHigh 10
--decTreeMinInstancesPerNodeStep 5
--rfNumTreesLow 50
--rfNumTreesHigh 70
--rfNumTreesStep 20
--rfSubsamplingRateLow 0.6
--rfSubsamplingRateHigh 0.8
--rfSubsamplingRateStep 0.2
--rfFeatureSubsetStrategy auto
--gbtSubsamplingRateLow 0.6
--gbtSubsamplingRateHigh 0.8
--gbtSubsamplingRateStep 0.1
--gbtStepSizeLow 0.05
--gbtStepSizeHigh 0.15
--gbtStepSizeStep 0.05
--gbtMaxIterLow 20
--gbtMaxIterHigh 20
--gbtMaxIterStep 1
```

```
--gbtFeatureSubsetStrategy auto
--milkMaxLagLow 1
--milkMaxLagHigh 3
--herdMaxLagLow 0
--herdMaxLagHigh 3
--weatherMaxLagLow 1
--weatherMaxLagHigh 3
--fourierTermsMaxCount 2
```

6. Zaključak

Rad opisuje postupke analize i prognoziranja vremenskih serija. Opisani postupci analize vremenske serije nude uvid u statistička svojstva podataka (dekompozicija, stacionarnost i autokorelacija). U radu su opisani postupci prognoziranja vremenske serije pomoću klasičnih statističkih metoda i metoda strojnog učenja. Klasične statističke metode zahtijevaju malo domenskog znanja o problemu što je njihova prednost spram metoda strojnog učenja. To je posebno izraženo prilikom izgradnje značajki iz sirovih podataka. Međutim, korištenje izračunatih značajki za prilagodbu modela je prednost metoda strojnog učenja nad klasičnim metodama. Iako model vektorske autoregresije nudi tu mogućnost ograničen je pretpostavkom o obostranoj simetričnoj ovisnosti vremenske serije koja se predviđa s vremenskim serijama koje su značajke. Algoritmi strojnog učenja dodatno sadrže metode za smanjenje utjecaja šuma što je prednost kada se radi s podacima koji ga sadrže.

Radni okvir Apache Spark pokazao se pogodnim za manipulaciju podacima vremenskih serija zbog bogatog izbora ugrađenih funkcija i mogućnosti pisanja vlastitih koje djeluju nad strukturama podataka radnog okvira. Knjižnica za strojno učenje nudi standardne algoritme strojnog učenja s mogućnosti finog podešavanja pomoću hiperparametara. Nedostatak knjižnice je nepostojanje ugrađenog odabira kontinuiranih značajki.

Studijski slučaj prognoziranja dnevne proizvodnje mlijeka pokazao je da prognoziranje metodama strojnog učenja daje bolje rezultate od klasičnih metoda za prva tri koraka prognoze. Pri tome valja uzeti u obzir potrebu za obradom podataka, izgradnjom značajki te složenijom pretragom hiperparametara nego kod klasičnih metoda. Odabrani modeli dokazuju da algoritmi iz skupine ansambala bolje generaliziraju od linearne regresije. Nedostatak algoritma gradijentno ojačanih stabala u usporedbi sa slučajnim šumama je sporije učenje i predviđanje zbog iterativnog načina rada algoritma koji onemogućuje paralelizaciju.

LITERATURA

Ratnadip Adhikari i Ramesh K Agrawal. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*, 2013.

Australian Bureau of Meteorology. Daily minimum temperatures in Melbourne, Australia, 1981-1990, 2014. URL `https://datamarket.com/data/set/2324/daily-minimum-temperatures-in-melbourne-australia-1981-1990#!ds=2324&display=line`. Datum nastanka: 1.2.2014., Datum pristupa: 31.3.2019.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, i Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 5 izdanju, 2015.

Wikimedia Commons. Overfitting, 2008. URL `https://commons.wikimedia.org/wiki/File:Overfitting.svg`. File: `Overfitting.svg`, Datum nastanka: 24.2.2008., Datum pristupa: 6.6.2019.

SF Crosbie i GN Hinch. An intuitive explanation of generalised linear models. *New Zealand journal of agricultural research*, 28(1):19–29, 1985.

Thomas G Dietterich. Ensemble methods in machine learning. U *International workshop on multiple classifier systems*, stranice 1–15. Springer, 2000.

Jerome H Friedman. Stochastic gradient boosting. *mh (x; am)*, 1000:0, 1999.

Wayne A Fuller. *Introduction to statistical time series*, svezak 428. John Wiley & Sons, 2 izdanju, 2009.

Fabrizio Gabbiani i Steven James Cox. *Mathematics for neuroscientists*. Academic Press, 2017.

Rob J Hyndman i George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Jason Brownlee. How to Develop Machine Learning Models for Multivariate Multi-Step Air Pollution Time Series Forecasting, 2018. URL `https://machinelearningmastery.com/how-to-develop-machine-learning-models-for-multivariate-multi-step-air-pollution-time-series-forecasting`. Datum nastanka: 19.10.2018., Datum pristupa: 15.4.2019.

Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, i Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1-3): 159–178, 1992.

Jessica Lin, Eamonn Keogh, Stefano Lonardi, i Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. U *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, stranice 2–11. ACM, 2003.

Jan Šnajder. Predavanje iz predmeta Strojno učenje, 12. Ansambli, 2017.

Jan Šnajder i Bojana Dalbelo Bašić. *Strojno učenje*. Fakultet elektrotehnike i računarstva, 2014.

Philipp Probst, Marvin N Wright, i Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, stranica e1301, 2019.

G William Schwert. Tests for unit roots: A monte carlo investigation. *Journal of Business & Economic Statistics*, 20(1):5–17, 2002.

Marno Verbeek. *A guide to modern econometrics*. John Wiley & Sons, 2008.

JW West, BG Mullinix, i JK Bernard. Effects of hot, humid weather on milk temperature, dry matter intake, and milk yield of lactating dairy cows. *Journal of Dairy Science*, 86(1):232–242, 2003.

Prognoza vremenskih serija korištenjem radnog okvira Apache Spark

Sažetak

Prognoziranje vremenske serije omogućuje uvid u budućnost promatranog procesa. Za uspješno prognoziranje najprije je potrebno provesti analizu vremenske serije koja obuhvaća dekompoziciju, statistička testiranja na stacionarnost te prikaz funkcija (parcijalne) autokorelacije. Prognoziranje je moguće raditi pomoću klasičnih statističkih metoda kao što su modeli AR, MA, ARIMA, eksponencijalno zaglađivanje i vektorska autoregresija. Uz klasične statističke metode moguće je koristiti i metode strojnog učenja, preciznije regresijske algoritme kao što su linearna regresija, model linearne regresije, slučajne šume i gradijentno ojačana stabla. Radni okvir Apache Spark za raspodijeljenu obradu i izračunavanja posjeduje funkcionalnosti pogodne za rad s vremenskim serijama te kroz knjižnicu za strojno učenje nudi navedene algoritme strojnog učenja s mogućnosti finog podešavanja postavljanjem hiperparametara. Studijski slučaj prognoziranja dnevne proizvodnje mlijeka pokazao je da su metode strojnog učenja ostvarene pomoću radnog okvira Apache Spark dale bolje rezultate od klasičnih statističkih metoda za prva tri koraka prognoziranja. Također se pokazalo da su meta-algoritmi ansambala dali bolje rezultate od linearne regresije.

Ključne riječi: vremenska serija, analiza vremenske serije, prognoziranje vremenske serije, Apache Spark, strojno učenje

Time Series Forecasting with Apache Spark Framework

Abstract

Time series forecasting gives insights into the future behavior of the observed process. In order to make accurate forecasts, it is necessary to utilize time series decomposition, stationarity tests and to plot the (partial) autocorrelation function. Classical time series forecasting methods like AR, MA, ARIMA, exponential smoothing, and vector autoregression are widely used for time series forecasting. Besides classical methods, it is possible to use machine learning methods like linear regression, random forest, and gradient tree boosting. Apache Spark cluster-computing framework offers functionalities suitable for processing time series data and rich machine learning library. The library contains implementations of mentioned machine learning algorithms with the possibility of fine-tuning hyperparameters. Daily milk yield forecasting case study showed that machine learning methods implemented in Apache Spark produced better results than classical methods for the three-day-ahead forecast. The case study also showed that ensemble methods produced better results than linear regression algorithm.

Keywords: Time series, Time series analysis, Time series forecasting, Apache Spark, Machine learning