SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1984

Prognoza vremenskih serija korištenjem programske knjižnice Scikit-learn

Renato Bošnjak

Zagreb, lipanj 2019.

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

Zagreb, 8. ožujka 2019.

DIPLOMSKI ZADATAK br. 1984

Pristupnik: **Renato Bošnjak (0036485149)**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Prognoza vremenskih serija korištenjem programske knjižnice Scikit-learn**

Opis zadatka:

Scikit-learn je programska knjižnica otvorenog koda za dubinsku analizu i obradu podataka te strojno učenje. Vaš zadatak je detaljno opisati i usporediti podržane metode strojnog učenja za prognoziranje vremenskih serija u programskoj knjižnici Scikit-learn. 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 usmjeravanju i savjetima tijekom studija te obitelji i prijateljima na pruženoj podršci.

SADRŽAJ

1. Uvod **1**

2. Vremenske serije **2**

2.1. Komponente vremenske serije 2

2.2. Svojstvo stacionarnosti vremenske serije 4

2.2.1. Definicija stacionarnosti 5

2.2.2. Nestacionarnost uzrokovana trendom 6

2.2.3. Nestacionarnost uzrokovana promjenjivom varijancom 9

2.3. Ispitivanje svojstva stacionarnosti statističkim testovima 11

2.3.1. Ispitivanje jediničnog korijena 11

2.3.2. Ispitivanje nestacionarnosti 12

2.3.3. Primjena statističkih testova stacionarnosti 13

3. Prognoziranje vremenskih serija stohastičkim modelima **14**

3.1. Stohastički modeli 14

3.2. Modeli ARIMA 14

3.2.1. Ograničenja ARIMA modela 15

3.2.2. Komponente modela 15

3.2.3. Određivanje hiperparametara Box-Jenkinsovom metodom . . 16

3.2.4. Sezonski model ARIMA 19

3.3. Model vektorske autoregresije 20

3.3.1. Definicija 21

3.3.2. Hiperparametri algoritma 22

4. Prognoziranje vremenskih serija modelima strojnog učenja **23**

4.1. Oznake 23

4.2. Linearni model regresije 24

4.2.1. Linearna regresija 24

4.2.2. Nelinearna regresija . . . 25
4.2.3. Regularizirana regresija . . . 26
4.3. Regresija algoritmom *k*-najbližih susjeda . . . 27
4.3.1. Ulazni podatci . . . 27
4.3.2. Opis rada algoritma . . . 28
4.3.3. Odabir hiperparametara algoritma . . . 29
4.3.4. Svojstva algoritma . . . 30
4.3.5. Metode odabira najbližih susjeda . . . 30
4.4. Regresija potpornih vektora . . . 32
4.4.1. Opis algoritma . . . 33
4.4.2. Primarna formulacija . . . 34
4.4.3. Dualna formulacija . . . 37
4.4.4. Odabir hiperparametara algoritma . . . 44

5. Studijski slučaj prognoziranja vodostaja Kupe **46**
5.1. Dostupni podatci . . . 46
5.1.1. Hidrološki podatci . . . 46
5.1.2. Podatci padalina . . . 48
5.1.3. Prikaz podataka . . . 48
5.2. Analiza podataka . . . 49
5.2.1. Dekompozicija vremenskih serija . . . 49
5.2.2. Ispitivanje svojstva stacionarnosti . . . 52
5.2.3. Analiza korelacija vremenskih serija . . . 53
5.2.4. Odabir mjernih postaja . . . 53
5.3. Kriterij uspješnosti prognoze . . . 54
5.4. Prognoziranje stohastičkim modelima . . . 56
5.4.1. Prognoziranje modelom ARIMA . . . 57
5.4.2. Prognoziranje modelom vektorske autoregresije . . . 57
5.5. Prognoziranje modelima strojnog učenja . . . 58
5.5.1. Izgradnja označenog skupa podataka . . . 58
5.5.2. Odabrani modeli . . . 60
5.6. Rezultati . . . 62

6. Zaključak **68**

Literatura **69**

1. Uvod

Podatci vremenskih serija prisutni su u mnogim područjima djelovanja u kojima oni nastaju bilježenjem stanja promatranog procesa kroz vrijeme. Njihovom se analizom omogućuje razumijevanje prošlog ponašanja te donosi uvid u moguće buduće ponašanje promatranog procesa.

U tu svrhu se već desetljećima primjenjuju klasične statističke metode analize i prognoze vremenskih serija stohastičkim modelima (autoregresijski modeli, modeli pomičnih prosjeka te njihove kombinacije, itd.). S druge strane, algoritmi strojnog učenja pokazali su korisnost u rješavanju specifičnih zadataka temeljem podataka koji općenito ne moraju biti vezani uz vremenske serije, a zbog sve veće količine dostupnih podataka te povećanja računalnih resursa, posljednjih godina dobivaju sve više pažnje.

Cilj ovog rada jest usporedba metoda i rezultata dobivenih klasičnim statističkim tehnikama prognoze vremenskih serija s algoritmima strojnog učenja na studijskom slučaju prognoziranja vodostaja rijeke Kupe temeljem podataka Državnog hidrometeorološkog zavoda Republike Hrvatske. Prognoziranje stohastičkim modelima ostvareno je programskom knjižnicom otvorenog koda StatsModels (Seabold i Perktold, 2010), dok je za implementaciju postupaka strojnog učenja korištena programska knjižnica otvorenog koda Scikit-learn (Pedregosa et al., 2011) koja nudi funkcionalnosti dubinske analize i obrade podataka te strojno učenje.

Nastavak rada organiziran je na sljedeći način. Poglavlje 2 opisuje vremenske serije te uvjete koji moraju biti zadovoljeni kako bi primjena klasičnih metoda nad njima bila moguća. Poglavlja 3 i 4 daju pregled klasičnih metoda te algoritama strojnog učenja korištenih u praktičnom dijelu rada. Zatim u poglavlju 5 slijedi opis studijskog slučaja nakon čega je u poglavlju 6 dan zaključak rada.

2. Vremenske serije

Vremenske serije definiraju se kao skupovi podataka nastali slijednim uzorkovanjem vrijednosti od interesa kroz vrijeme, gdje se uzorkovanja mogu događati u pravilnim ili nepravilnim vremenskim razmacima. Primjeri podataka vremenskih serija uključuju razne meteorološke i hidrološke podatke, podatke prikupljene senzorima, demografske i mnoge druge podatke.

Otkrivanje uzoraka i pravilnosti unutar podataka vremenske serije omogućeno je postojanjem njihovog vremenskog uređenja pomoću kojeg se definira međuovisnost slijednih vrijednosti. Međutim, zbog vremenskog uređenja kojim se uvodi nova dimenzija podataka primjene metoda povezanih sa strojnim učenjem, poput podjele skupa podataka, metode unakrsne provjere i drugih, postaju složenije.

Nastavak ovog poglavlja donosi opis komponenti koje čine svaku vremensku seriju te metode kojima se one mogu prikazati u svrhu analize. Zatim slijedi opis svojstva stacionarnosti, jednog od temeljnih svojstava vremenske serije kojim se omogućuje analiza i interpretacija rezultata, te pregled statističkih testova kojima se to svojstvo ispituje.

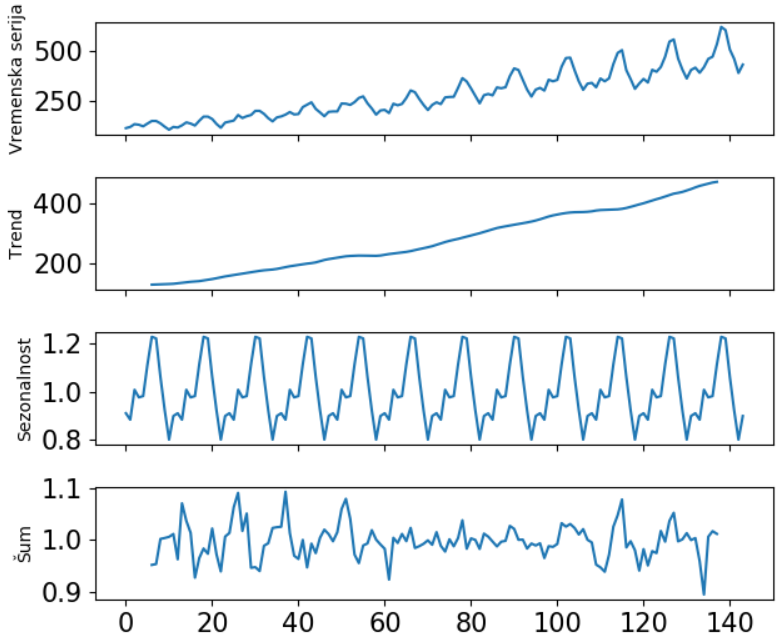
2.1. Komponente vremenske serije

Vremensku seriju moguće je rastaviti na komponente trenda, sezonalnosti, ciklusa te šuma koje su opisane u nastavku prema (Hyndman i Athanasopoulos, 2018).

Dugoročno ponašanje vremenske serije određeno je trendom koji prema iznosu može biti padajući ili rastući. Ovisno o tome predstavlja li ga pravac ili krivulja trend se dijeli na linearni i nelinearni, dok prema uzroku može biti deterministički i stohastički što je objašnjeno u odjeljku 2.2.2. Komponenta sezonalnosti se u vremenskim serijama javlja zbog društvenih, klimatoloških te drugih utjecaja koji izazivaju periodičko ponavljanje ponašanja vremenske serije. Za razliku od sezonalnosti, ciklus uzrokuje ponavljanje ponašanja serije u nepravilnim razmacima koji su često dulji od sezonskih. Nakon uklanjanja navedenih komponenti iz vremenske serije preostaje šum

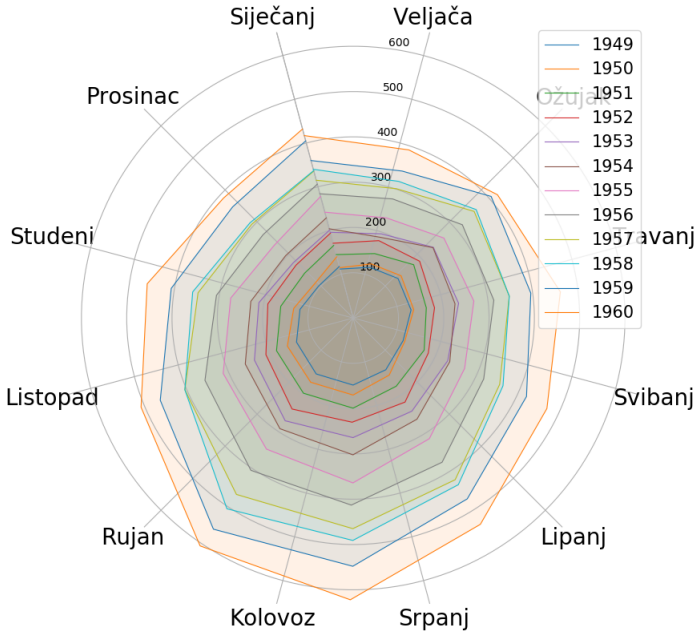
koji se zbog čestih i promjenjivih kolebanja ne može predvidjeti pa se naziva iregularnom komponentom.

Prilikom dekompozicije, komponente trenda i ciklusa često se objedinjuju te prikazuju kao trend, kao što je prikazano slikom 2.1 koja je nastala dekomponiranjem vremenske serije mjesečnog broja putnika zračnog prijevoza (u tisućama) tijekom razdoblja od početka 1949. do kraja 1960. godine (Box et al., 2015).



Slika 2.1: Komponente vremenske serije

Sezonalnost i trend najlakše se uočavaju prikazom podataka, a jedan od načina kojim se vremenska serija može sažeto prikazati po njenim sezonama je polarni graf, prikazan na slici 2.2. Na njoj se jasno uočava pravilnost tijekom svake sezone. Na primjer, vidljivo je kako se tijekom ljetnih mjeseci, osobito sredinom srpnja, preveze najviše putnika, dok je zimi njihov broj manji. Kako su grafovi za svaku godinu gotovo jednakog oblika, zaključuje se jaka sezonalna komponenta ove vremenske serije. Osim za uočavanje sezonalnosti, polarni graf može otkriti postojanje trenda. Tako se na istoj slici uočava sve veća udaljenost grafova od središta što označava postojanje rastućeg trenda.



Slika 2.2: Prikaz podataka polarnim grafom

2.2. Svojstvo stacionarnosti vremenske serije

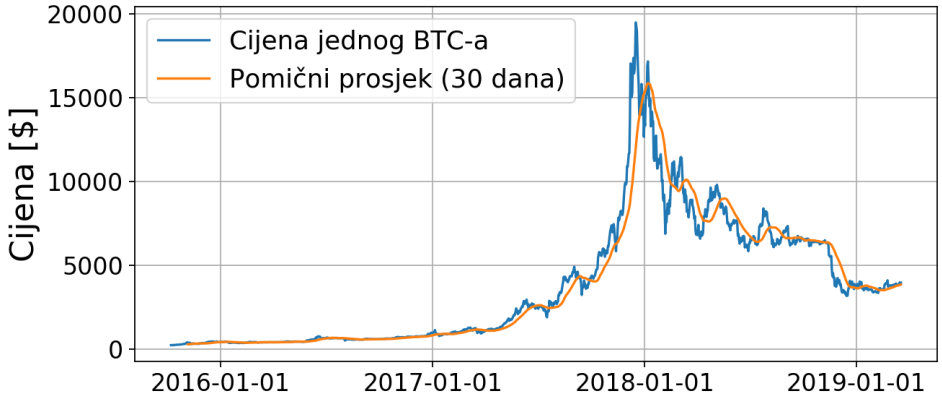
Svojstvo stacionarnosti vremenske serije jedno je od temeljnih pretpostavki mnogih metoda za analizu i predviđanje vremenskih serija. Zato u nastavku slijedi opis i formalna definicija stacionarnosti te opis razloga nastanka nestacionarnosti kao i pregled statističkih testova i metoda za utvrđivanje i ostvarivanje svojstva stacionarnosti podataka.

Stacionarnost podataka poželjno je svojstvo vremenske serije kojim se označava nepromjenjivost njenih statističkih svojstava kroz vrijeme (npr. srednja vrijednost i varijanca). Zbog nepromjenjivosti statističkih svojstava stacionarne procese je značajno lakše modelirati te interpretirati. Ovo svojstvo posebno je važno kod primjene stohastičkih modela koji buduće ponašanje predviđaju temeljem statističkih svojstava koja su aproksimirali iz dostupnih podataka. Tada se u slučaju stacionarnosti ta svojstva ne mijenjaju čime se modeliranje značajno olakšava.

U slučaju nestacionarnosti vremenske serije statistička svojstva mijenjaju se kroz vrijeme (npr. porast srednje vrijednosti, tj. postojanje trenda) pa primjena modela kojima se ona aproksimiraju nije smislena jer se buduće ponašanje, zbog promjene svojstava, ne može ispravno predvidjeti. Primjere takvih procesa čine cijene dionica i

kriptovaluta koji su zanimljivi zbog koristi koju bi donijeli, ako bi se zaključci o budućem ponašanju pokazali ispravnima. Problem navedenih slučaja uzrokuje velik broj varijabli koje utječu na formiranje cijene, a nisu opažane te se zato realiziraju kao šum.

Razlika stacionarnih i nestacionarnih podataka lako se uočava kod pojave šokova, odnosno iznenadnih događaja koji mijenjaju ponašanje vremenske serije. Kod stacionarnih podataka vremenska serija se nakon pojave šoka zbog nepromjenjivosti statističkih svojstava stabilizira i vraća u početno stanje. Kod nestacionarnih podataka šokovi utječu na promjenu distribucije te mijenjanju srednju vrijednost i varijancu vremenske serije čime se onemogućuje modeliranje vremenske serije jer se parametri distribucije mijenjaju kroz vrijeme. Takvo ponašanje prikazano je na slici 2.3 gdje je uz cijenu Bitcoina (CryptoDataDownload) prikazan i pomični prosjek izračunat nad vremenskim prozorom od 30 dana.



Slika 2.3: Primjer nestacionarne vremenske serije

2.2.1. Definicija stacionarnosti

Vremenska serija se, osim kao skup slijednih podataka, može promatrati i kao realizacija događaja stohastičkog procesa koji generira podatke te vremenske serije po određenoj distribuciji. To znači da je vremenska serija $\{x_{t_1}, x_{t_2}, \ldots, x_{t_N}\}$ nastala realizacijom N slučajnih varijabli $\{X_{t_1}, X_{t_2}, \ldots, X_{t_N}\}$, od kojih je svaka određena statističkim svojstvima srednje vrijednosti i varijance. Ako se statistička svojstva tih varijabli ne mijenjaju ovisno o vremenu, stohastički proces je strogo stacionaran čime se označava i stroga stacionarnost vremenske serije kao skupa podataka. U suprotnom, ako se statistička svojstva mijenjaju ovisno o vremenu, stohastički proces i vremenska serija kao skup podataka su nestacionarni. (Box et al., 2015)

Zbog strogih zahtjeva prethodne definicije, u primjeni je teško postići strogu stacionarnost vremenske serije pa se osim nje definira i pojam slabe stacionarnosti. Tada je proces slabo ili kovarijančno stacionaran, ako su srednja vrijednost i varijanca neovisne o vremenu, a kovarijanca ovisi samo o razlici vremenskih trenutaka u kojima su se slučajne varijable realizirale (Box et al., 2015). Slaba stacionarnost formalno je definirana izrazima 2.1.

$$\begin{aligned} \mathbb{E}(X_t) &= \mu, \quad t = 1, 2, \dots \\ \mathrm{Var}(X_t) &= \sigma^2, \quad t = 1, 2, \dots \\ \mathrm{Cov}(X_t, X_{t+\Delta_t}) &= f(\Delta_t), \quad t = 1, 2, \dots \end{aligned} \tag{2.1}$$

2.2.2. Nestacionarnost uzrokovana trendom

Jedan od uzroka nestacionarnosti podataka je promjenjiva srednja vrijednost vremenske serije, odnosno pojava trenda. Trend, ovisno o uzroku, može biti deterministički ili stohastički.

Deterministički trend

Deterministički trend uzrokovan je eksplicitnom ovisnošću podataka o poznatoj varijabli, najčešće vremenu. U tom se slučaju vrijednost vremenske serije u trenutku t definira prema jednadžbi 2.2 gdje je a konstanta koja definira pomak, b konstanta nagiba čijim se predznakom definira pozitivan ili negativan trend, a ϵ_t iznos šuma u trenutku t. Kako se šum ravna prema distribuciji $\mathcal{N}(0, \sigma_\epsilon^2)$ očekivanje varijable $X(t)$, definirano jednadžbom 2.3, ne ovisi o njemu.

$$X_t = f(t) = a + b \cdot t + \epsilon_t, \tag{2.2}$$

$$\mathbb{E}(X_t) = a + b \cdot t \tag{2.3}$$

Očekivanje u ovom slučaju izravno ovisi o vremenskom trenutku t čime se ostvaruje nestacionarnost procesa. Tada je u svrhu postizanja stacionarnosti trend potrebno ukloniti, a jedan od načina uklanjanja determinističkog trenda je metoda diferenciranja prvog reda kojom se vremenska serija transformira u razliku susjednih vrijednosti, kao što je prikazano izrazom 2.6. Time je postignuto očekivanje neovisno o vremenu

čime su podatci postali stacionarni. Izvorni podatci se u daljnjoj analizi i predviđanju mijenjaju diferenciranim koje je zbog svojstva invertibilnosti diferenciranja na kraju moguće vratiti na izvorni raspon vrijednosti.

$$X_t = a + b \cdot t + \epsilon_t \tag{2.4}$$

$$X_{t-1} = a + b \cdot (t-1) + \epsilon_{t-1} \tag{2.5}$$

$$\begin{aligned} \Delta_{X_t} &= X_{t-1} - X_t \\ &= b \cdot (t-1) - b \cdot t + \epsilon_{t-1} - \epsilon_t \\ &= b + (\epsilon_{t-1} - \epsilon_t) \\ \mathbb{E}(\Delta_{X_t}) &= \mathbb{E}(b) + \mathbb{E}(\epsilon_{t-1} - \epsilon_t) \\ &= \mathbb{E}(b) = b \end{aligned} \tag{2.6, 2.7}$$

Osim diferenciranja, trend je moguće ukloniti primjenom postupaka strojnog učenja. Tada se on oblikuje modelom strojnog učenja nakon čega slijedi postupak uklanjanja trenda koji je ostvaren oduzimanjem vrijednosti vremenske serije te modeliranog trenda. Time se dobivaju rezidualne vrijednosti koje se koriste u daljnjoj analizi nakon čega je povratak u izvornu skalu omogućen dodavanjem modeliranog trenda.

Stohastički trend

Stohastički trend složenija je pojava jer ne pretpostavlja eksplicitnu ovisnost trenda o poznatoj varijabli čime se u proces unosi nesigurnost. Tada je vrijednost vremenske serije u trenutku t određena prema jednadžbi 2.8 gdje su a i b konstante, a ϵ_t iznos šuma u trenutku t. Ovakav model vremenske serije naziva se autoregresijskim modelom u kojem buduće vrijednosti vremenske serije linearno ovise o prethodnim vrijednostima vlastite serije.

$$X_t = a + b \cdot X_{t-1} + \epsilon_t, \tag{2.8}$$

Tako definirana jednadžba je rekurzivne prirode pa se može raspisati sve do prvog člana vremenske serije, odnosno X_0 čime se dobiva izraz 2.9.

$$\begin{aligned} X_t =& a + b \cdot X_{t-1} + \epsilon_t \\ =& a \cdot \sum_{n=0}^{t-1} b^n + b^t \cdot X_0 + \sum_{n=0}^{t-1} (b^n \cdot \epsilon_{t-n}) \end{aligned} \tag{2.9}$$

Uvjet stacionarnosti je konvergencija izraza 2.9, a ona ovisi o konstanti b. Za sumu $\sum_{n=0}^{t-1} b^n$, prema d'Alembertovom kriteriju, vrijedi :

$$\lim_{n\to\infty}\left|\frac{b^{n+1}}{b^n}\right| = |b| = \begin{cases} < 1, & \text{suma konvergira,} \\ = 1, & \text{konvergenciju nije moguće odrediti,} \\ > 1, & \text{suma divergira.} \end{cases} \tag{2.10}$$

Istim postupkom dolazimo do uvjeta konvergencije za sumu $\sum_{n=0}^{t-1}(b^n \cdot \epsilon_{t-n})$:

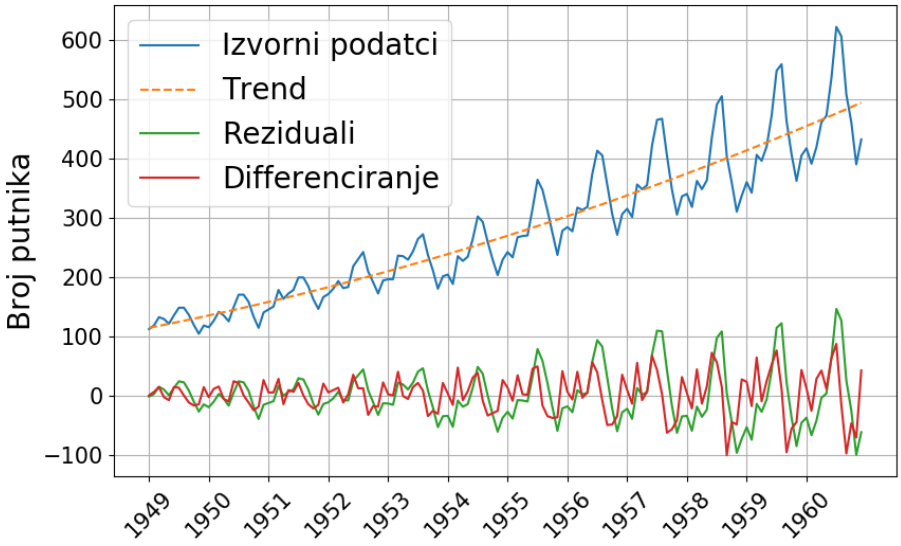
$$\lim_{n\to\infty}\left|\frac{b^{n+1}\cdot\epsilon_{t-(n+1)}}{b^n\cdot\epsilon_{t-n}}\right| = \lim_{n\to\infty}\left|\frac{b^{n+1}}{b^n}\right| \cdot \left|\frac{\epsilon_{t-(n+1)}}{\epsilon_{t-n}}\right|$$

Zbog $\mathcal{E} \sim iid(0,\, \sigma_{\mathcal{E}}^2)$ šum ne ovisi o n te slijedi

$$\lim_{n\to\infty}\left|\frac{b^{n+1}}{b^n}\right| = |b| = \begin{cases} < 1, & \text{suma konvergira,} \\ = 1, & \text{konvergenciju nije moguće odrediti,} \\ > 1, & \text{suma divergira.} \end{cases} \tag{2.11}$$

Iz jednadžbi 2.10 i 2.11 proizlazi da će serija biti stacionarna za vrijednost parametra $|b| < 1$. Za uklanjanje stohastičkog trenda koriste se iste metode kao i kod deterministčkog. Ovisno o podatcima, diferenciranje prvog reda nije uvijek dovoljno za potpuno uklanjanje trenda. Tada se može koristiti diferenciranje viših redova, odnosno diferenciranje diferenciranih vremenskih serija, sve dok se iz podataka ne ukloni trend.

Slika 2.4 prikazuje uklanjanje trenda vremenske serije mjesečnog broja putnika zračnog prijevoza (Box et al., 2015) primjenom diferenciranja prvog reda te modeliranja polinomijalnom regresijom drugog stupnja.



Slika 2.4: Uklanjanje trenda

2.2.3. Nestacionarnost uzrokovana promjenjivom varijancom

Osim trenda, nestacionarnost je uzrokovana i promjenom varijance ovisno o vremenu. Primjer takvog procesa je model slučajnog kretanja (engl. *random walk*), definiran jednadžbom 2.12, koji je dobiven uvrštavanjem parametara $a = 0$ i $b = 1$ u jednadžbu 2.8.

$$X_t = X_{t-1} + \epsilon_t \tag{2.12}$$

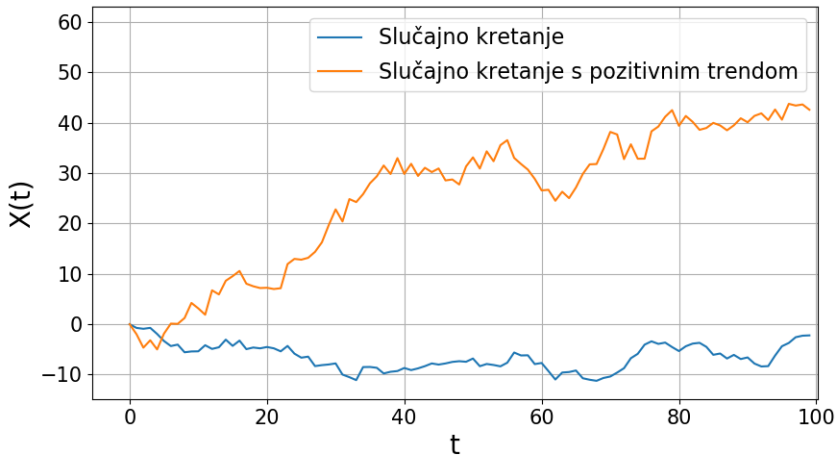
Zbog očekivanja šuma jednakog 0 srednja vrijednost procesa slučajnog kretanja je konstantna te iznosi X_0. Međutim, nestacionarnost slučajnog kretanja uzrokovana je promjenjivom varijancom koja nastaje gomilanjem šumova tijekom vremena, što je prikazano jednadžbom 2.14. Kako su šumovi međusobno nezavisni, ukupna varijanca procesa jednaka je zbroju varijanci svih šumova do trenutka t čime se postiže izravna ovisnost varijance o vremenu što dovodi do divergencije kada t teži beskonačnosti.

$$
\begin{aligned}
X_t &= X_{t-1} + \epsilon_t \\
&= X_{t-2} + \epsilon_{t-1} + \epsilon_t \\
&\vdots \\
&= X_0 + \sum_{i=1}^{t} \epsilon_i
\end{aligned} \tag{2.13}
$$

$$
Var(X_t) = Var(\sum_{i=1}^{t} \epsilon_i) = t \cdot \sigma_\epsilon^2 \tag{2.14}
$$

Susjedne vrijednosti vremenske serije nastale ovim procesom povezuje šum pa buduće ponašanje ovakvog procesa nije moguće predvidjeti. Kako je srednja vrijednost šuma jednaka 0, najbolji pokušaj predviđanja vrijednosti x_{t+1} jednak je njenoj prethodnoj vrijednosti x_t.

Uvrštavanjem parametara $a \neq 0$ i $b = 1$ u jednadžbu 2.8 nastaje proces slučajnog kretanja s trendom koji je, ovisno o predznaku parametra a pozitivan ili negativan. Primjeri vremenskih serija generiranih procesom slučajnog kretanja s i bez trenda prikazani su slikom 2.5.



Slika 2.5: Slučajno kretanje

2.3. Ispitivanje svojstva stacionarnosti statističkim testovima

Jedan od načina ispitivanja svojstva stacionarnosti je primjena statističkih testova koji se, ovisno o nultoj hipotezi te uvjetu njenog odbacivanja, dijele na testove jediničnog korijena i testove nestacionarnosti. Kako je cilj statističkog testiranja odbacivanje nulte hipoteze, čime bi se s određenom sigurnosti potvrdila alternativna, testovi iz ove dvije skupine su zbog postavljanja suprotnih hipoteza komplementarni.

2.3.1. Ispitivanje jediničnog korijena

Nulta hipoteza testova jediničnog korijena tvrdi postojanje jediničnog korijena u procesu čime se povlači njegova nestacionarnost, dok se svojstvo stacionarnosti tvrdi alternativnom hipotezom. Tako odbacivanjem nulte hipoteze testovi ove skupine, s određenom razinom sigurnosti, mogu potvrditi stacionarnost vremenske serije.

Zbog njihovih pretpostavki i ograničenja, testove jediničnog korijena treba pažljivo koristiti te oprezno interpretirati. Ovi testovi posebno su skloni pogreškama tipa II što dovodi do nemogućnosti dokazivanja stacionarnosti (Ng i Perron, 2001). Primjer statističkog testa ove skupine je prošireni Dickey-Fullerov test jediničnog korijena (Dickey i Fuller, 1979), kraće zapisan kao ADF test, koji je opisan u nastavku.

Prošireni Dickey-Fullerov test

Proširena verzija testa temelji se na osnovnom Dickey-Fullerovom testu jediničnog korijena koji ispituje njegovo postojanje u autoregresijskom modelu iz jednadžbe 2.8. Nultom hipotezom se tvrdi postojanje jediničnog korijena što znači da je vrijednost parametra b jednaka 1 čime serija postaje nestacionarna. Testiranje hipoteze ne provodi se nad izvornim podatcima, nego vrijednostima nastalim diferenciranjem prvog reda koje je prikazano izrazima iz 2.15.

$$\begin{aligned} X_t &= a + b \cdot X_{t-1} + \epsilon_t \\ \Delta_{X_t} &= X_t - X_{t-1} \\ &= a + (b-1) \cdot X_{t-1} + \epsilon_t \\ &= a + \delta \cdot X_{t-1} + \epsilon_t \end{aligned} \tag{2.15}$$

Time se nulta hipoteza mijenja i jedinični korijen postoji, ako je parametar δ jednak 0. Nakon provođenja t-testa kojim se ispituje je li parametar δ jednak 0, dobivena

vrijednost statistike uspoređuje se s kritičnim vrijednostima iz tablice (Dickey, 1976), specifičnim za Dickey-Fullerovo testiranje. Ovisno o tome postoji li trend ili pomak (srednja vrijednost vremenske serije nije jednaka 0) u podatcima nad kojima se provodi testiranje, potrebno je izabrati različiti tip testiranja kojim se definiraju zasebne kritične vrijednosti. Neopravdano uključivanje ili isključivanje komponenti trenda ili pomaka može dovesti do pogrešnih interpretacija rezultata zbog povećanja vjerojatnosti pogreška tipa I i tipa II.

Prošireni Dickey-Fullerov test omogućuje ispitivanje jediničnog korijena nad autoregresijskim procesima višeg reda, odnosno kada buduća vrijednost vremenske serije linearno ovisi o p prethodnih vrijednosti iste serije, gdje je p prirodni broj. Općenita definicija takvog modela prikazana je jednadžbom 2.16.

$$X_t = a + b_1 \cdot X_{t-1} + b_2 \cdot X_{t-2} + \cdots + b_p \cdot X_{t-p} + \epsilon_t \tag{2.16}$$

Prošireni test se također izvodi nad diferenciranim podatcima provođenjem t-testa kojim se ispituje je li dobiveni parametar δ jednak 0. Zatim se izračunata statistika uspoređuje s kritičnim vrijednostima definiranim u Dickey-Fullerovoj tablici. Ako je ona manja od definiranih kritičnih vrijednosti, nulta hipoteza se odbacuje te se s određenom razinom sigurnosti zaključuje stacionarnost serije.

Parametar p naziva se vremensko zaostajanje (engl. *lag*) i određuje broj prethodnih vrijednosti koje će utjecati na trenutnu te se mora odrediti prije provođenja testa. Kako vjerojatnosti pogrešaka tipa I i II značajno ovise o odabiru ovog parametra, za ispravne zaključke potrebno je obratiti pažnju na njegov odabir (Ng i Perron, 2001). Obično se u tu svrhu koriste informacijski kriteriji poput Akaikeovog ili Bayesovog dok je za preciznije procjene ovog parametra potrebno koristiti napredne metode opisane u (Ng i Perron, 2001).

2.3.2. Ispitivanje nestacionarnosti

Kako bi riješili problem male snage statističkih testova jediničnog korijena, testovi za ispitivanje nestacionarnosti postavljaju obrnute hipoteze. Tako se nultom hipotezom tvrdi stacionarnost procesa, dok se alternativnom pretpostavlja postojanje jediničnog korijena. Time odbacivanjem nulte hipoteze testovi ove skupine s određenom razinom sigurnosti mogu potvrditi nestacionarnost serije.

Kwiatkowski–Phillips–Schmidt–Shinov test

Prvi značajan doprinos ovoj skupini testova, objavljen u radu (Kwiatkowski et al., 1992), predstavlja Kwiatkowski–Phillips–Schmidt–Shinov test, kraće pisan kao KPSS test. Testiranje hipoteze provodi se nad modelom iz 2.17 kojeg su autori prikazali dekompozicijom vremenske serije na deterministički trend, slučajno kretanje r_t te šum ϵ_t koji je nastao stacionarnim procesom. Nulta hipoteza testa suprotna je ADF testu te se njome tvrdi stacionarnost vremenske serije oko srednje vrijednosti ili trenda.

Kako se deterministički trend može otkloniti diferenciranjem, a šum po pretpostavci nastaje stacionarnim procesom, nulta hipoteza testa pretvara u pretpostavku o varijanci procesa slučajnog kretanja. Tada se s njom tvrdi kako σ_u^2, prema jednadžbi 2.14, mora biti jednak 0 čime bi cjelokupni proces ostao stacionaran, dok se alternativnom hipotezom tvrdi kako je σ_u^2 veća od nule što povlači nestacionarnost procesa.

Rezultati statističkog testa uspoređuju se s kritičnim vrijednostima iz (Kwiatkowski et al., 1992). Ako je iznos dobivene statistike veći od definiranih kritičnih vrijednosti, nulta hipoteza se odbacuje te se s određenom razinom sigurnosti tvrdi kako je serija nestacionarna.

$$\begin{aligned} X_t &= b \cdot t + r_t + \epsilon_t \\ r_t &= r_{t-1} + u_t, \text{ gdje je } u_t \text{ iz } \mathcal{N}(0, \sigma_u^2) \end{aligned} \tag{2.17}$$

2.3.3. Primjena statističkih testova stacionarnosti

Zbog oprečnih hipoteza, testiranje stacionarnosti provodi se kombinacijom testova iz navedenih skupina. Proces testiranja moguće je provesti kako je opisano u nastavku, gdje ishodi odbacivanja nulte hipoteze podrazumijevaju statističku značajnost te ispravan odabir parametara testova.

Ispitivanje stacionarnosti počinje provođenjem testova nestacionarnosti (npr. KPSS test). Odbacivanjem njihove nulte hipoteze zaključuje se nestacionarnost procesa te postupak završava. U suprotnom, zaključak o stacionarnosti izostaje te se proces nastavlja ispitivanjem postojanja jediničnog korijena (npr. ADF test). Tada se odbacivanjem njegove nulte hipoteze zaključuje stacionarnost, dok u slučaju nemogućnosti njenog odbacivanja zaključci u oba testa izostaju što dovodi do nemogućnosti određivanja svojstva stacionarnosti. Kod takvih slučaja vremensku seriju je potrebno diferencirati te ponoviti opisano testiranje.

3. Prognoziranje vremenskih serija stohastičkim modelima

Nakon analize vremenske serije te utvrđivanja svojstva stacionarnosti slijedi korak prognoze. Iako se rad bavi prognoziranjem pomoću metoda strojnog učenja, u ovom poglavlju bit će opisane klasične tehnike prognoziranja korištenjem stohastičkih modela koje će se također koristiti u praktičnom dijelu rada. Tako će se rezultati dobiveni metodama strojnog učenja moći usporediti s onima dobivenim klasičnim metodama čime se omogućuje procjena uspješnosti te donošenje zaključka o korisnosti primjene metoda strojnog učenja na problem iz praktičnog dijela rada.

3.1. Stohastički modeli

Vremenska serija može se promatrati kao realizacija događaja nekog stohastičkog procesa, kako je i objašnjeno u poglavlju 2.2.1. Tada stohastički modeli imaju za cilj modelirati te procese te tako procijeniti njihove vjerojatnosne distribucije čime se omogućuje prognoza budućih vrijednosti. Ovisno o procesu kojeg predstavljaju te načinu modeliranja definiraju se različiti stohastički modeli.

3.2. Modeli ARIMA

Jedna od često korištenih porodica stohastičkih modela je integrirani autoregresijski model i model pomičnih prosjeka, kraće pisan kao ARIMA (engl. *autoregressive integrated moving average*). Model se temelji na kombinaciji dvaju jednostavnijih stohastičkih modela te postupku diferenciranja kojim se ostvaruje stacionarnost podataka. Kako su oba modela te postupak diferenciranja određeni svojim hiperparametrima koje je potrebno unaprijed odrediti, u ovom poglavlju se uz opis modela nalazi i pregled metoda odabira hiperparametara čime se povećava kvaliteta modela.

3.2.1. Ograničenja ARIMA modela

ARIMA modele moguće je primijeniti samo na univarijatne vremenske serije koje ne sadrže sezonalnu komponentu. Problem sezonalnosti u podatcima moguće je izbjeći obradom podataka ili korištenjem modificiranog modela SARIMA (engl. *seasonal autoregressive integrated moving average*) koji omogućuje izravan rad s takvim podatcima.

3.2.2. Komponente modela

Model ARIMA definiran je hiperparametrima p, d i q koji definiraju red autoregresijskog modela, red diferenciranja te red modela pomičnih prosjeka od kojih je model sastavljen.

Autoregresijski model

Autoregresijski model korišten je prilikom modeliranja stohastičkog trenda u poglavlju 2.2.2, gdje je i dana definicija modela prvog reda. Općeniti model reda p, definiran jednadžbom 3.1, opisuje stohastički proces u kojem trenutna vrijednost linearno ovisi o prethodnim vrijednostima te nepredvidljivoj pogrešci koja je uzorkovana iz normalne distribucije.

Oznake a i b_i u jednadžbi definiraju parametre modela koje je potrebno procijeniti, npr. korištenjem metode najmanjih kvadrata ili maksimizacijom očekivanja. Broj prethodnih vrijednosti koje će se uzeti u obzir te broj parametara koje je potrebno procijeniti ovisi o redu autoregresijskog modela definiranog hiperparametrom p. Tako je za model reda p potrebno procijeniti $p + 1$ parametar, tj. parametre a te $b_1, b_2, \ldots b_p$.

$$X_t = a + \sum_{i=1}^{p} b_i \cdot X_{t-i} + \epsilon_t \tag{3.1}$$

Diferenciranje

Model ARIMA podržava rad s nestacionarnim podatcima izazvanih postojanjem trenda pa se hiperparametrom d definira red diferenciranja kojim se postiže stacionarnost podataka. Ako je vremenska serija već stacionarna, diferenciranje je potrebno isključiti postavljanjem vrijednosti hiperparametra na nulu čime se izbjegava nepotrebno diferenciranje koje može narušiti točnost prognoze.

Model pomičnih prosjeka

Model pomičnih prosjeka, definiran jednadžbom 3.2, trenutnu vrijednost predstavlja kao linearnu kombinaciju prognostičkih pogrešaka nad trenutnom i prošlim vrijednostima, gdje je broj prošlih prognostičkih pogrešaka koje ulaze u model zadan hiperparametrom q.

U jednadžbi je srednja vrijednost vremenske serije predstavljena parametrom a, dok su prognostičke pogreške i njihovi koeficijenti koji čine parametre modela u koraku i označeni s ϵ_i te b_i. Kako prognostičke pogreške za buduće trenutke t nisu opažane, procjena parametara se provodi nelinearnim postupcima, dok će njihov broj ovisiti o redu modela. Tako će za model reda q biti potrebno procijeniti $q + 1$ parametara koji uključuju a te $b_1, b_2, \ldots b_q$.

$$X_t = a + \epsilon_t + \sum_{i=1}^{q} b_i \cdot \epsilon_{t-i} \tag{3.2}$$

3.2.3. Određivanje hiperparametara Box-Jenkinsovom metodom

Odabir hiperparametara ARIMA modela znatno utječe na kvalitetu prognoze, stoga ih je potrebno ispravno odrediti. Jedan od načina njihovog određivanja je Box-Jenkinsova metoda (Box et al., 2015) koja će biti opisana u nastavku. Prije opisa metode potrebno je uvesti pojmove autokorelacijske funkcije (engl. *autocorrelation function*, ACF) te funkcije parcijalne autokorelacije (engl. *partial autocorrelation function*, PACF) na kojima se ova metoda temelji.

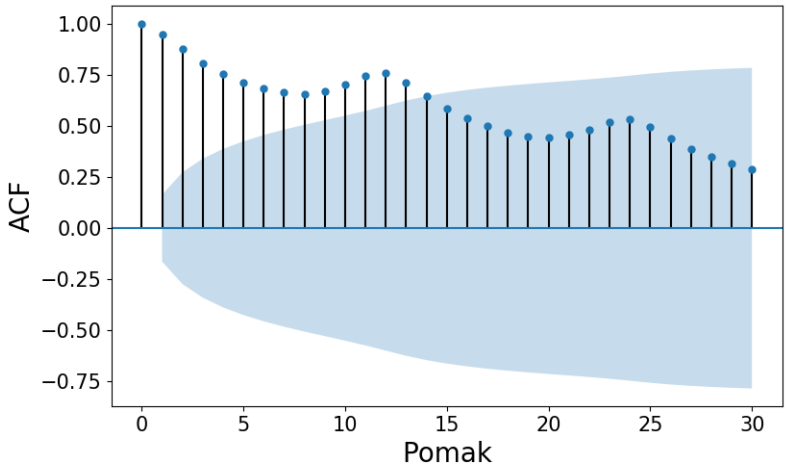
Autokorelacijska funkcija

Autokorelacijska funkcija mjeri linearnu ovisnost zadane vremenske serije i vremenske serije koja je dobivena pomakom zadane za određeni broj koraka. To znači da će za zadani vremenski pomak *l* autokorelacijska funkcija računati korelaciju između x_t i x_{t-l} za svaki trenutak *t*. Prema definiciji, iznos funkcije za nulti pomak uvijek će biti jednak 1 jer svaka vrijednost vremenske serije savršeno korelira sama sa sobom, dok je porastom pomaka očekivan pad vrijednosti autokorelacije koji se javlja zbog sve manje ovisnosti udaljenijih vrijednosti vremenske serije.

Slika 3.1 prikazuje iznose autokorelacijske funkcije izračunate nad skupom podataka o mjesečnom broju putnika zračnog prijevoza te pomake iz intervala $\{0, \ldots, 30\}$. Označene točke definiraju iznos autokorelacijske funkcije za vremenski pomak *l*, a

plava površina označava područje ispod zadanog intervala pouzdanosti (npr. 95-postotni interval pouzdanosti) što znači da su sve vrijednosti izvan njega statistički značajne.

Na njoj se, prema očekivanju, povećanjem pomaka primjećuje pad vrijednosti autokorelacije. Međutim, nakon osmog pomaka slijedi ponovni rast vrijednosti koji maksimalnu vrijednost postiže za vrijednost pomaka 12. Ova pojava se javlja zbog postojanja izražene komponente sezonalnosti. Kako se radi o mjesečnoj frekvenciji uzorkovanja, pomak od 12 vrijednosti označava jednogodišnji pomak što uzrokuje izračun korelacije nad vrijednostima istih mjeseci u susjednim godinama zbog čega je korelacija u tim trenutcima veća. Isti efekt javlja se i za višegodišnje pomake kada je iznos korelacije nešto manji zbog veće udaljenosti vrijednosti serije nad kojima se ona računa.



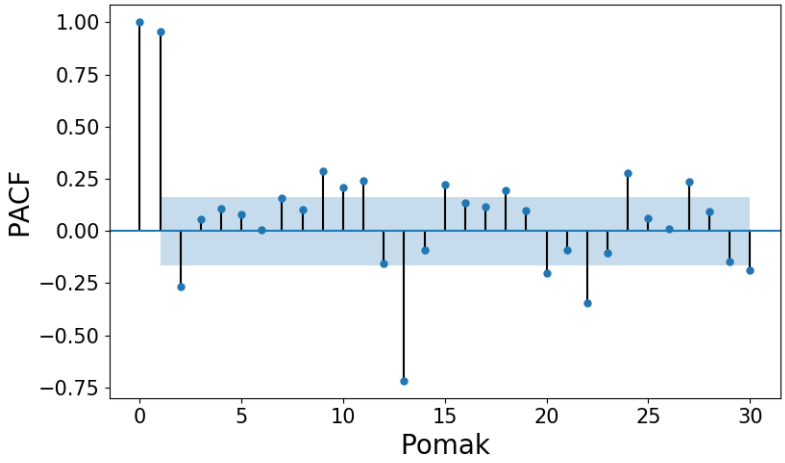
Slika 3.1: Prikaz autokorelacijske funkcije

Funkcija parcijalne autokorelacije

Zbog ulančavanja ovisnosti susjednih vrijednosti u seriji na izračunatu vrijednost autokorelacije za pomak l, osim izravne korelacije x_t i x_{t-l}, utječu i neizravne korelacije svih prethodnih pomaka manjih od l. Primjerice, kod podataka dobivenih dnevnim uzorkovanjem današnja vrijednost utjecat će na sutrašnju, a sutrašnja na preksutrašnju. Time će se korelacije izračunate za jednodnevne pomake propagirati na izračun dvodnevnih i ostalih višednevnih, a dvodnevne na tri i višednevnih, itd.

Zato se za izračun izravne korelacije koristi funkcija parcijalne autokorelacije koja izbacuje propagirani utjecaj pomaka manjih od l. Korelogram parcijalne autokorelacijske funkcije za iste podatke prikazan je na slici 3.2. Vidljivo je kako su iznosi zbog uklanjanja neizravnih korelacija značajno manji u odnosu na prethodni slučaj. Kako je

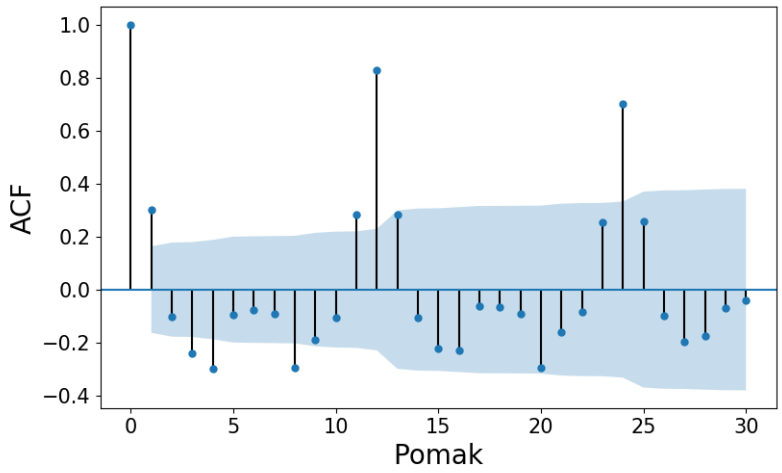
korelacija za nulti pomak uvijek jednaka 1, a za prvi pomak ne postoji manji pomak koji bi neizravno utjecao na izračun korelacije, iznosi na tim mjestima ostaju nepromijenjeni.



Slika 3.2: Prikaz funkcije parcijalne autokorelacije

Određivanje hiperparametara

Određivanje stupnja diferenciranja ARIMA modela temelji se na analizi korelograma ACF. Tako će naglo odumiranje vrijednosti na korelogramu označavati stacionarnost serije, dok se u suprotnom provodi postupno diferenciranje, počevši od reda 1 nadalje. Nakon svakog povećanja reda diferenciranja korelogram ACF se ponovno prikazuje i analizira. Time se izbjegava pretjerano diferenciranje koje model čini složenijim, a u podatke uvodi dodatnu autokorelaciju koja narušava točnost prognoze. Korelogram ACF sa slike 3.1, osim sezonalnosti, upućuje na postojanje trenda jer vrijednosti korelacija ne opadaju naglo, nego postupno. Nakon diferenciranja prvog reda te izračuna autokorelacijske funkcije dobiva se korelogram ACF sa slike 3.3. Većina korelacija je iščeznula, a preostale jake korelacije nalaze se na pomacima koji odgovaraju višekratnicima dužine sezone što sugerira primjenu SARIMA modela ili uklanjanja sezonalnosti provođenjem transformacije čime bi se omogućila primjena modela ARIMA.



Slika 3.3: Prikaz funkcije parcijalne autokorelacije nad diferenciranom serijom

Odabir hiperparametra p i q kod autoregresijskog modela te modela pomičnih prosjeka temelji se na obje funkcije korelacije. Kod autoregresijskih procesa korelogram za PACF će nakon pomaka definiranog idealnim redom modela p pasti ispod razine statističke pouzdanosti, dok će ACF imati oblik eksponencijalnog odumiranja. Za model pomičnih prosjeka vrijedi obratna situacija pa će kod njega vrijednosti na korelogramu za ACF nakon pomaka definiranog idealnim redom modela q pasti ispod razine statističke pouzdanosti, dok će PACF imati oblik eksponencijalnog odumiranja.

Složeniji slučaj dogodit će se kada vremenska serija nastane mješavinom ova dva modela. U tom slučaju moguće je kombinirati zaključke dobivene iz ACF i PACF grafova, ali oni neće uvijek biti jednoznačni. Tada treba koristiti dodatne alate za određivanje ova dva hiperparametra. Primjeri takvih su Akaikeov informacijski kriterij (AIC) te Bayesov informacijski kriterij (BIC) koji za cilj imaju odabrati one hiperparametre p i q koji će rezultirati modelom dobrih performansi i primjerene složenosti. Takav odabir postižu maksimizacijom izglednosti hiperparametara te kažnjavanjem pretjerane složenosti modela koja je izazvana odabirom njihovih prevelikih vrijednosti.

3.2.4. Sezonski model ARIMA

Modeli ARIMA ne podržavaju izravno modeliranje podataka koji sadrže sezonsku komponentu. U tom slučaju podatke je potrebno obraditi sezonalnim diferenciranjem što znači da se diferencirane vrijednosti ne dobivaju razlikom susjednih vrijednosti serije kao kod običnog diferenciranja, nego razlikom vrijednosti udaljenih za duljinu jedne sezone.

Korak uklanjanja sezonalnosti moguće je ugraditi u model ARIMA čime nastaje sezonski model ARIMA, odnosno SARIMA. Osim opisanih hiperparametara p, d i q, SARIMA uvodi četiri dodatna koji su vezani uz komponentu sezonalnosti. Tako se hiperparametrom m definira broj očitanja koja čine jednu sezonu (npr. za mjesečne podatke i jednogodišnju sezonalnost m će iznositi 12), a hiperparametrima P, D i Q sezonski redovi autoregresijskog modela, diferenciranja te modela pomičnih prosjeka.

Interpretacija sezonskih hiperparametara istovjetna je onima u modelu ARIMA, uz razliku da se susjedstvo definira po sezonama, a ne pojedinim vrijednostima serije. To znači da će trenutna vrijednost ovisiti o zadanom broju vrijednosti u trenutcima koji su od nje udaljeni točno za višekratnike duljine sezone, a sezonsko diferenciranje ponašati prema opisu s početka ovog potpoglavlja, uz mogućnost poopćenja na proizvoljni odabir reda.

Odabir sezonskih hiperparametara također je moguće provesti analizom korelograma ACF i PACF za pomake koji odgovaraju višekratnicima duljina sezone. Međutim, zbog velikog broja hiperparametara koje je potrebno istovremeno uskladiti zadatak njihovog odabira često postaje problematičan, a odabir optimalnog modela koji bi doveo do nekoreliranih pogrešaka ponekad nemoguć (Hyndman i Athanasopoulos, 2018). Djelomično rješenje ovog problema moguće je ostvariti automatizacijom procesa pretrage hiperparametara korištenjem pretraživanja po rešetci (engl. *grid search*) prema određenom kriteriju (npr. ukupna prognostička pogreška, AIC ili BIC). Kako bi se izbjegla prevelika složenost modela pretragu je potrebno provoditi nad razumnim rasponom vrijednosti koje najčešće ostaju u području jednoznamenkastih prirodnih brojeva.

3.3. Model vektorske autoregresije

Kako je modele ARIMA moguće koristiti samo nad univarijatnim vremenskim serijama, odnosno u slučaju kada se predviđanje vremenske serije obavlja samo temeljem prošlih vrijednosti te serije, potrebno je odgovoriti na problem predviđanja multivarijatnih vremenskih serija u kojima se kroz vrijeme mijenjaju iznosi više od jedne varijable.

Jednostavniji pristup rješavanja ovog problema uključivao bi korištenje zasebnih ARIMA modela nad svakom serijom. Iako se njegovom primjenom omogućava predviđanje multivarijatnih serija, ovaj pristup ne uzima u obzir moguću međusobnu povezanost opažanih varijabli sustava koja se može iskoristiti s ciljem povećanja točnosti predikcija. Na primjer, na vodostaj rijeke vjerojatno utječu podatci o vodostajima nje-

nih pritoka te padalinama mjerenih na okolnom području čijim bi se uključivanjem u proces mogla povećati točnosti predviđanja vodostaja.

Bolji način, koji omogućuje modeliranje ovisnosti promatranih serija kroz vrijeme zahtijeva generalizaciju dosad opisanih modela. Tako Sims (1980) uvodi poopćenje autoregresijskog modela za primjenu nad multivarijatnim vremenskim serijama nazvanog modelom vektorske autoregresije (engl. *vector autoregression*, VAR). Kako ne zahtijeva prethodno razumijevanje međusobnih utjecaja varijabli sustava, ovaj se model često koristi kao osnovni model (engl. *baseline*) prilikom rješavanja problema predviđanja multivarijatnih vremenskih serija.

3.3.1. Definicija

Matematička definicija VAR-a reda p kojim se modelira multivarijatna vremenska serija koju čini K varijabli prikazana je jednadžbom 3.3. Ona je gotovo jednaka definiciji modela AR iz jednadžbe 3.1, uz razliku što su skalarne vrijednosti postale vektori i matrice. Tako će K-dimenzionalni vektor $\vec{X}_j$ predstavljati vrijednosti svih varijabli sustava u trenutku j, dok će parametri modela biti određeni K-dimenzionalni vektorom pomaka $\vec{a}$ te $K \times K$ dimenzionalnom matricom utjecaja varijabli $\boldsymbol{W_i}$. Procjena navedenih parametara provodi se minimizacijom pogrešaka određenih K-dimenzionalnim vektorom $\vec{\epsilon}_t$.

$$\underset{K\times 1}{\vec{X}_t} = \underset{K\times 1}{\vec{a}} + \sum_{i=1}^{p} \underset{K\times K}{\boldsymbol{W_i}} \cdot \underset{K\times 1}{\vec{X}_{t-i}} + \underset{K\times 1}{\vec{\epsilon}_t} \tag{3.3}$$

Raspisivanjem vektora i matrica za model reda p i K varijabli iz vektorske jednadžbe 3.4 lako se uočava međuovisnost serija. U njoj X_t^k označava vrijednost k-te varijable vremenske serije u trenutku t, dok parametri a^k predstavljaju njezin pomak. Parametri $W_{k',l}^k$ utječu na izračun vrijednosti k-te vremenske serije, a označavaju utjecaj te serije s k'-tom vremenskom serijom u vremenskom pomaku l. Tako će vrijednost i-te varijable vremenske serije u trenutku t, označene s X_t^i, ovisiti o i-tom retku matrica $\boldsymbol{W_i}$ te svim komponentama vektora $\vec{X}_{t-i}$ za svaki i od 1 do p. Zato se, umjesto vektorskog zapisa, može koristiti i zapis pomoću K jednadžbi u kojem svaka jednadžba definira izračun jedne od varijabli.

$$\begin{bmatrix} X_t^1 \\ X_t^2 \\ \vdots \\ X_t^K \end{bmatrix} = \begin{bmatrix} a^1 \\ a^2 \\ \vdots \\ a^K \end{bmatrix} + \overbrace{\begin{bmatrix} W_{1,1}^1 & W_{2,1}^1 & \dots & W_{K,1}^1 \\ W_{1,1}^2 & W_{2,1}^2 & \dots & W_{K,1}^2 \\ \vdots & \ddots & & \\ W_{1,1}^K & W_{2,1}^K & \dots & W_{K,1}^K \end{bmatrix}}^{\text{Pomak } l=1} \cdot \begin{bmatrix} X_{t-1}^1 \\ X_{t-1}^2 \\ \vdots \\ X_{t-1}^K \end{bmatrix} + \overbrace{\dots}^{\text{Pomaci } l=2,3,\dots,p-1} + \\ + \underbrace{\begin{bmatrix} W_{1,p}^1 & W_{2,p}^1 & \dots & W_{K,p}^1 \\ W_{1,p}^2 & W_{2,p}^2 & \dots & W_{K,p}^2 \\ \vdots & \ddots & & \\ W_{1,p}^K & W_{2,p}^K & \dots & W_{K,p}^K \end{bmatrix}}_{\text{Pomak } l=p} \cdot \begin{bmatrix} X_{t-p}^1 \\ X_{t-p}^2 \\ \vdots \\ X_{t-p}^K \end{bmatrix} + \begin{bmatrix} \epsilon_t^1 \\ \epsilon_t^2 \\ \vdots \\ \epsilon_t^K \end{bmatrix} \tag{3.4}$$

3.3.2. Hiperparametri algoritma

Prilikom definiranja modela potrebno je odrediti K vremenskih serija, odnosno varijabli koje će biti uključene u sustav te broj vremenskih pomaka p kojim se određuje broj prošlih vrijednosti koje će utjecati na izračun trenutne. Broj parametara, u odnosu na univarijatnu verziju modela, zbog vektorizacije značajno raste. Kako je potrebno procijeniti vektor parametara $\vec{a}$ te p različitih matrica $\boldsymbol{W_i}$, ukupan broj parametara modela iznosi $K + pK^2$.

Zbog kvadratne ovisnosti broja parametara o broju odabranih varijabli u sustavu postoji opasnost od pojavljivanja prevelike složenosti modela uzrokovane prevelikim brojem parametara, posebno ako se on približi broju raspoloživih podataka. Smanjenje složenosti ostvarivo je selekcijom varijabli temeljenoj na izračunu korelacije te primjerenim odabirom reda modela pomoću informacijskih kriterija. U slučaju VAR-a, zbog brzorastuće složenosti u ovisnosti o broju parametara, literatura (Hyndman i Athanasopoulos, 2018) preporučuje korištenje BIC-a koji uvijek rezultira jednako ili manje složenim modelima u odnosu na AIC.

4. Prognoziranje vremenskih serija modelima strojnog učenja

Metode strojnog učenja dokazale su korisnost primjene na mnogim problemima u različitim područjima djelovanja. Među njima se nalaze i problemi predviđanja vremenskih serija za koje tipični postupci vezani uz korake pripreme podataka, odabira modela te njegovog vrednovanja, zbog nastanka vremenske dimenzije podataka, postaju složeniji u odnosu na klasične skupove podataka.

Nastavak poglavlja donosi opis korištenih oznaka nakon čega slijedi detaljniji pregled modela i njihovih svojstava korištenih u praktičnom dijelu, dok će postupci pretvorbe vremenske serije u problem nadziranog učenja te detalji vezani uz odabir i vrednovanje modela biti opisani u poglavlju 5.

4.1. Oznake

Prije opisa algoritama potrebno je uvesti oznake i simbole korištene u nastavku gdje svaki vektor predstavlja vektor-stupac. Kako se rad bavi prognozom vremenskih serija pomoću regresije, koja pripada porodici metoda nadziranog strojnog učenja, skup podataka sastoji se od označenih primjera. Tako će svaki primjer iz skupa za učenje biti definiran vektorom značajki te svojom oznakom, odnosno labelom. Izrazom 4.1 definiran je skup označenih podataka kojeg čini N označenih primjera kojima je redni broj određen unutar zagrade u eksponentu.

Svaka od komponenti n-dimenzionalnog vektora značajki $\vec{x}$ određena je pojedinim opažanim svojstvom podatka. Prostor svih primjera definiranih vektorom značajki naziva se ulaznim ili prostorom primjera, a označen je s $\mathcal{X}$. Željene izlazne vrijednosti svakog podatka označene su s y te u slučaju regresije pripadaju skupu realnih brojeva.

$$D = \{(\vec{x}^{(i)}, y^{(i)})\}_{i=1}^{N},\ y^{(i)} \in \mathbb{R} \tag{4.1}$$

$$\vec{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}) \tag{4.2}$$

$$h \colon \mathcal{X} \to \mathbb{R} \tag{4.3}$$

$$\mathcal{H} = \{h(\vec{x}^{(i)}; \vec{w})\}_{\vec{w}} \tag{4.4}$$

Hipoteza, označena s $h(\vec{x}^{(i)}; \vec{w})$, je funkcija koja definira preslikavanje primjera iz ulaznog prostora u skup realnih brojeva te je prikazana izrazom 4.3. Model je tada predstavljen skupom svih hipoteza koje čine prostor hipoteza $\mathcal{H}$, kako je i prikazano izrazom 4.4. Kako je svaka hipoteza definirana do na parametar $\vec{w}$ učenje modela svodi se na pretragu prostora hipoteza, odnosno pretragu parametara koji će definirati najtočniju hipotezu.

4.2. Linearni model regresije

Linearni modeli regresije tipični su predstavnici algoritama nadziranog učenja korištenih za rješavanje regresijskih zadaća. Nastavak poglavlja donosi pregled vrsta regresije te modele korištene u praktičnom dijelu rada.

Tip regresije ovisi o brojnosti ulaznih te izlaznih varijabli. Tako se u slučaju modeliranja izlazne varijable pomoću samo jedne značajke regresija naziva jednostavnom, dok je u suprotnom višestruka. Prema broju izlaznih varijabli regresija je, u slučaju jedne izlazne varijable univarijatna, dok je u suprotnom multivarijatna.

4.2.1. Linearna regresija

Linearni model regresije, u kojem je izlazna varijabla predstavljena linearnom kombinacijom značajki, naziva se linearnom regresijom. Vektorski zapis modela višestruke univarijatne linearne regresije prikazan je jednadžbom 4.5. Učenje modela temelji se na minimizaciji funkcije pogreške iz jednadžbe 4.6, koja je zadana sumom kvadratnih odstupanja labela i predviđenih vrijednosti nad svim primjerima iz skupa za učenje. Time se učenje svodi na rješavanje optimizacijskog problema najmanjih kvadrata (engl. *ordinary least squares*, OLS).

$$h(\vec{x}^{(i)}) = \vec{w}^{\intercal} \cdot \vec{x}^{(i)} + w_0 \tag{4.5}$$

$$E(\vec{w}|D) = \sum_{i=1}^{N} (y^{(i)} - h(\vec{x}^{(i)}))^2 \tag{4.6}$$

4.2.2. Nelinearna regresija

Mnoge se pojave u praksi ne mogu dobro predstaviti linearnom ovisnosti izlazne varijable o značajkama pa je potrebno uvesti nelinearnost. Umjesto promjene modela, radi očuvanja optimizacijskog postupka linearne regresije za kojeg postoje učinkovite metode rješavanja, nelinearnost se ostvaruje transformacijom podataka. Ideja se temelji na preslikavanju nelinearnih podataka ulaznog prostora u prostor više dimenzije korištenjem skupa baznih funkcija (engl. *basis function*). Tako nastaju nove značajke kojima kodira nelinearnost što dozvoljava zadržavanje linearnosti modela po parametrima $\vec{w}$.

Model nelinearne regresije prikazan je jednadžbom 4.7 iz koje je vidljivo kako se parametri nisu mijenjali u odnosu na linearnu regresiju čime je model ostao linearan po parametrima. Preslikavanje podataka u pojedinu dimenziju novog prostora ostvaruje se baznim funkcijama ϕ_j, definiranih izrazom 4.10, dok se potpuno preslikavanje podataka u prostor više dimenzije ostvaruje funkcijom preslikavanja iz izraza 4.9 i 4.8. Ona je definirana skupom od m baznih funkcija čime se ostvaruje preslikavanje iz n-dimenzionalnog ulaznog prostora u prostor dimenzije m, koja će u slučaju ostvarivanja nelinearnosti biti veća od n.

$$h(\vec{x}^{(i)}) = \vec{w}^{\intercal} \cdot \vec{\phi}(\vec{x}^{(i)}) + w_0 \tag{4.7}$$

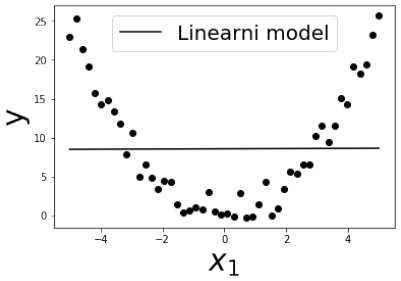
$$\vec{\phi} = (\phi_1(\vec{x}^{(i)}), \ldots, \phi_m(\vec{x}^{(i)})) \tag{4.8}$$

$$\vec{\phi} \colon \mathbb{R}^n \to \mathbb{R}^m \tag{4.9}$$

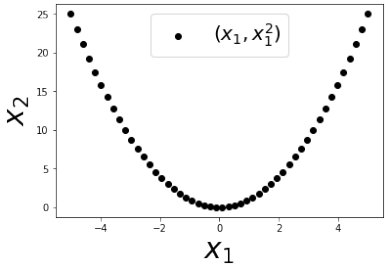
$$\phi_j \colon \mathbb{R}^n \to \mathbb{R} \tag{4.10}$$

Primjer preslikavanja prikazan je slikama 4.1. One prikazuju rezultate učenja linearnih modela nad nelinearnim funkcijama u slučaju nekorištenja baznih funkcija (slika 4.1a) te nakon njihovog korištenja (slika 4.1c). Na slici 4.1b je prikazano preslikavanje iz jednodimenzionalnog u dvodimenzionalni prostor korištenjem polinoma drugog stupnja. Tako je nova značajka x_2 dobivena kvadriranjem izvorne značajke x_1.

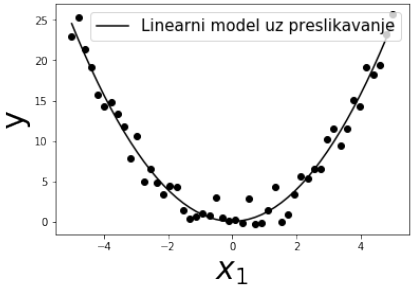
Uvođenjem funkcije preslikavanja nelinearna regresija uvodi hiperparametar dimenzije novog prostora koja definira stupanj polinoma preslikavanja. Odabir višeg



(a) Linearni model prije preslikavanja podataka



(b) Preslikavanje podataka iz 1D u 2D prostor



(c) Linearni model nakon preslikavanja podataka

Slika 4.1: Modeliranje nelinearnosti preslikavanjem ulaznih podataka u višu dimenziju

stupnja polinoma rezultira složenijim modelom što može uzrokovati problem prenaučenosti koji je moguće suzbiti uvođenjem regularizacije.

4.2.3. Regularizirana regresija

Regularizacija se uvodi modifikacijom funkcije pogreške iz jednadžbe 4.6 u koju se dodaje regularizacijski izraz. Nova funkcija pogreške prikazana je jednadžbom 4.11 u kojoj $\lambda\Omega(\vec{w})$ predstavlja regularizacijski izraz. Snaga regularizacije određena je definicijom funkcije Ω te faktorom regularizacije λ koji postaje novi hiperparametar algoritma te se određuje metodom unakrsne provjere.

Kako se regularizacijom želi postići smanjenje složenosti modela uzrokovane rastom magnitude parametara, za regularizacijsku funkciju se odabiru norme njihovih vektora. U praksi se najčešće koriste prva i druga norma, a njihov se izbor temelji na metodi unakrsne provjere. Tako se tijekom optimizacijskog postupka preferiraju one hipoteze koje su definirane parametrima manje magnitude čime se postiže sklonost odabira jednostavnijih modela.

$$E_R(\vec{w}|D) = E(\vec{w}|D) + \lambda\Omega(\vec{w}) \tag{4.11}$$

4.3. Regresija algoritmom *k*-najbližih susjeda

Algoritam *k*-najbližih susjeda, objavljen u (Cover et al., 1967), jedan je od najjednostavnijih algoritama koji se mogu koristiti u svrhu klasifikacije i regresije. Pripada porodici neparametarskih modela regresije koji se nazivaju i modelima zaglađivanja, čija će svojstva biti opisana unutar ovog poglavlja.

Područje primjene algoritma je široko, a činjenica da ono obuhvaća hidrologiju i klimatologiju (Al-Qahtani i Crone, 2013) dodatno je utjecala na odluku uključivanja ovog algoritma u rad. Osnovna ideja algoritma proizlazi iz pretpostavke lokalnosti neparametarskih metoda prema kojoj međusobno blisku podatci imaju slične labele. Tako se određivanje labele ulaznog primjera provodi temeljem labela podataka iz skupa za učenje koji su najsličniji ulaznom primjeru. U nastavku slijedi opis algoritma i njegovih svojstava te pregled metoda koje smanjuju njegovu računalnu složenost.

4.3.1. Ulazni podatci

Ulazni podatci algoritma predstavljeni su vektorima d-dimenzijskog prostora $\mathcal{X}$ koji je razapet osima koje predstavljaju značajke. Ovisno o odabiru metrike udaljenosti, značajke mogu biti numeričke ili kategoričke. Tipično se za numeričke podatke koristi euklidska udaljenost koja se može poopćiti na Minkowskijevu udaljenost, dok se za kategoričke podatke mogu koristiti Hammingova ili Jaccardova udaljenost.

Algoritam zahtijeva potpune ulazne podatke pa je ulazne vektore koji sadrže nepotpune vrijednosti potrebno ukloniti ili nadopuniti. U slučaju nadopune nedostajućih vrijednosti potrebno je odabrati onu metodu koja neće značajno utjecati na odnos udaljenosti prema različitim ulaznim primjerima iz skupa za učenje čime se izbjegava problem pristranosti prilikom odabira određenih primjerima kao najbližih susjeda. Jednostavnije metode uključuju nadopunjavanje nedostajućih značajki fiksnim vrijednostima (npr. nulama), prosjekom ili medijanom koji se računaju nad značajkom svih primjera iz skupa za učenje za koje je ona postavljena.

Osim navedenih metoda, za nadopunu nedostajućih vrijednosti moguće je koristiti i algoritam *k*-najbližih susjeda. Usporedba tog pristupa s navedenim metodama te metodom nadopune zasnovanoj na dekompoziciji na singularne vrijednosti (engl. *singular value decomposition*, SVD) prikazana je u (Troyanskaya et al., 2001). Prema zaključku autora, metoda nadopune vrijednosti algoritmom *k*-najbližih susjeda pokazala se superiornom u odnosu na druge, osobito ako se radi o zašumljenim podatcima vremenskih serija ili podatcima bez vremenskog uređenja, dok se jedino u slučaju po-

datka vremenskih serija koji nisu sadržavali šum boljim pokazala metoda zasnovana na SVD-u.

4.3.2. Opis rada algoritma

Rad algoritma započinje fazom učenja koja se sastoji od pohrane vektora značajki i njihovih labela koji zajedno čine skup za učenje.

Prilikom predikcije, algoritam za ulazni primjer kojem određuje labelu pretražuje prostor primjera za učenje te, ovisno o zadanom hiperparametru *k* i metrici udaljenosti, pronalazi *k* primjera skupa za učenje koji se u ulaznom prostoru nalaze najbliže zadanom primjeru. Ovaj korak ujedno je i najsloženiji dio algoritma jer iziskuje izračun udaljenosti od zadanog primjera do svih primjera iz skupa za učenje. Ta složenost ovisi o veličini skupa za učenje te dimenzionalnosti ulaznih vektora, odnosno broju značajki temeljem kojih se udaljenost računa. Tako će skup za učenje kojeg čini *n* *d*-dimenzionalnih primjera rezultirati složenosti $\mathcal{O}(nd)$.

Nakon određivanja najsličnijih primjera slijedi korak predikcije nepoznate labele u kojem se labele *k* najbližih primjera iz skupa za učenje dovode na ulaz odabranoj agregatnoj funkciji (npr. uprosječivanje) koja daje predikciju. Jednadžba 4.12 prikazuje izračun labele za primjer $\vec{x}$ uprosječivanjem labela *k* najbližih susjeda koji su označeni skupom $NN_k(\vec{x})$.

$$h(\vec{x}) = \frac{1}{k} \sum_{(\vec{x}^{(i)}, y^{(i)}) \in NN_k(\vec{x})} y^{(i)} \tag{4.12}$$

Poboljšanje rezultata moguće je ostvariti odabirom složenije agregatne funkcije koja bi prilikom uprosječivanja labela odabranih susjeda u obzir uzela i njihovu relevantnost čime se prednost daje onim labelama za koje su vektori značajki bliži ispitnom primjeru. Općenito, umjesto udaljenosti je moguće koristiti proizvoljnu jezgrenu funkciju (engl. *kernel function*), definiranu u 4.13, koja računa sličnost između dva primjera na željeni način. Tada jednadžba 4.12 prelazi u jednadžbu 4.13 gdje κ predstavlja jezgrenu funkciju kojom se računa sličnost između ispitnog primjera i svih primjera iz skupa za učenje.

$$h(\vec{x}) = \frac{\sum_{i=1}^{N} \kappa(\vec{x}, \vec{x}^{(i)}) \cdot y^{(i)}}{\sum_{i=1}^{N} \kappa(\vec{x}, \vec{x}^{(i)})} \tag{4.13}$$

$$\kappa \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{4.14}$$

4.3.3. Odabir hiperparametara algoritma

Odabir metrike udaljenosti

Odabir metrike udaljenosti temelji se na vrsti i svojstvima podataka. Zbog intuitivne geometrijske interpretacije i jednostavnog izračuna euklidska udaljenost jedna je od najčešće korištenih metrika u slučaju kontinuiranih podataka. Glavni nedostatci ove metrike su velika osjetljivost na različite magnitude značajki te neprepoznavanje visokokoreliranih i nebitnih značajki.

Problem osjetljivosti na magnitudu značajki izaziva nepoželjno pridavanje veće važnosti značajkama većih magnituda, a rješava se skaliranjem značajki. Drugi nedostatak euklidske udaljenosti javlja se u slučaju visokokoreliranih i nebitnih značajki koje neopravdano utječu na izračun udaljenosti. Zato je za uklanjanje ovog problema potrebno provesti postupak odabira značajki kojim se takve značajke uklanjaju.

Navedene probleme moguće je riješiti i zamjenom euklidske s Mahalanobisovom udaljenosti koja je definirana jednadžbom 4.15. Ona u izračun udaljenosti ugrađuje matricu kovarijacije značajki $\mathbf{\Sigma}$, čime se svaka značajka skalira u postupku izračuna udaljenosti, ovisno o iznosima korelacije s drugim značajkama. Kako vremenska i memorijska složenost izračuna kovarijacijske matrice rastu kvadratno s obzirom na broj značajki, Mahalanobisova udaljenost nije primjenjiva kod visokodimenzionalnih podataka.

$$d_M(\vec{x_1}, \vec{x_2}) = \sqrt{(\vec{x_1} - \vec{x_2})^T \mathbf{\Sigma}^{-1} (\vec{x_1} - \vec{x_2})} \tag{4.15}$$

Odabir hiperparametra *k*

Uz metriku udaljenosti, algoritam je definiran i hiperparametrom *k* koji znatno utječe na kvalitetu rezultata. Hiperparametar poprima vrijednosti iz cjelobrojnog intervala $\{1, \ldots, N\}$, gdje je *N* veličina skupa za učenje.

Optimalna vrijednost, označena s k^*, određena je podatcima te se utvrđuje metodom unakrsne provjere. Tada je suboptimalnost hiperparametra izazvana odabirom vrijednosti koja je manja ili veća od k^*. U prvom slučaju, za određivanje nepoznatih labela koristi se premalo primjera iz skupa za učenje koji su slični ulaznom primjeru. Tada se zbog visoke vjerojatnosti pojavljivanja stršećih vrijednosti te šuma u skupu najbližih primjera javljaju problemi karakteristični prenaučenosti koji smanjuju robusnost algoritma što dovodi do smanjenja kvalitete algoritma.

Porastom vrijednosti hiperparametra opada vjerojatnost pojavljivanja stršećih vrijednosti u skupu najbližih susjeda te se smanjuje utjecaj šuma što dovodi do povećanja robusnosti algoritma. Povećanje vrijednosti hiperparametra iznad optimalnog k^* vodi prema gubitku glavne pretpostavke algoritma, odnosno svojstva lokalnosti. Tako se u ekstremnom slučaju, kada je *k* jednak veličini skupa za učenje, predikcija svodi na uprosječivanje labela skupa za učenje.

4.3.4. Svojstva algoritma

Prethodna potpoglavlja istaknula su prednosti poput jednostavnosti i interpretabilnosti te nedostatke koji uključuju osjetljivost na šum i stršeće vrijednosti te redundantne i nebitne značajke kao i probleme sa složenosti.

Dodatno ograničenje algoritma, vezano uz sve modele zaglađivanja, tiče se predviđanja vrijednosti labela za primjere koji se nalaze izvan prostora primjera iz skupa za učenje. U tom slučaju, algoritam će odluku o labeli primjera donijeti temeljem najbližih primjera iz skupa za učenje koji nužno nemaju veze s njim, ali su se zbog odsutnosti relevantnih podataka našli najbliže ulaznom primjeru.

Rješenje ovog problema jest odabir dovoljno velikog reprezentativnog skupa za učenje koji će dobro pokrivati prostor primjera, što je posebno teško ostvariti u slučaju visoko-dimenzionalnih podataka. Problem pokrivenosti visokodimenzionalnog ulaznog prostora primjerima za učenje opisan je pojmom prokletstva dimenzionalnosti (engl. *curse of dimensionality*) koje se javlja zbog eksponencijalnog opadanja gustoće primjera u ulaznom prostoru s porastom brojem dimenzija podataka.

4.3.5. Metode odabira najbližih susjeda

Linearna ovisnost vremena izvođenja algoritma ovisno o broju primjera iz skupa za učenja, uzrokovana korakom određivanja najbližih susjeda, narušava performanse u slučaju velikih skupove za učenje. Problem je moguće riješiti primjenom egzaktnih i aproksimativnih metoda čiji odabir ovisi o brojnosti i dimenzionalnosti podataka, dok su u nastavku opisane metode korištene u praktičnom dijelu rada.

K-d stabla

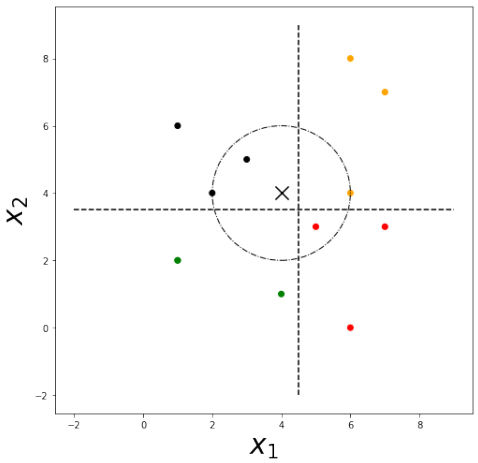
U slučaju niskodimenzionalnih podataka, odnosno kada vrijedi $d << n$, smanjenje složenosti ostvarivo je korištenjem strukture podataka *k*-dimenzionalnog stabla (*k*-d

stabla) (Bentley, 1975). Ovdje k označava dimenzionalnost prostora podataka i ne treba biti miješan s hiperparametrom algoritma najbližih susjeda.

Metoda se temelji na rekurzivnoj podjeli ulaznog prostora na particije određene skupom za učenje. Time se iscrpna pretraga cijelog ulaznog prostora svodi na pretragu određene particije kojoj ulazni primjer, čiju je labelu potrebno odrediti, pripada.

Postupak izgradnje stabla kreće odabirom značajke, odnosno dimenzije podataka koja će činiti korijen stabla te dijeliti ulazni prostor na dvije particije. Podjelu prostora moguće je ostvariti na više načina, međutim jedino odabir medijana izabrane dimenzije podataka iz skupa za učenje garantira izgradnju balansiranog stabla. Time se osigurava ravnomjerna podjela podataka u svakom čvoru stabla čime se jamči logaritamska vremenska složenost određivanja nepoznate labele. Nakon inicijalne podjele podataka na dvije particije postupak se rekurzivno ponavlja nad dobivenim particijama sve dok se ne zadovolji željeni uvjet zaustavljanja.

Reduciranje vremenske složenosti algoritma na logaritamsku dolazi s cijenom smanjenja točnosti jer ovakav algoritam ne jamči ispravan odabir najbližih susjeda. Taj problem prikazan je na slici 4.2. Ona prikazuje primjer dvodimenzionalnih podataka u kojima je particioniranje vršeno jednom po svakoj od dimenzija čime je ulazni prostor podijeljen na ukupno četiri particije. Particije, čije su granice prikazane crtkanom linijom, određene su temeljem medijana, a primjeri koji pripadaju jednoj particiji su obojeni istom bojom. Tako za ispitni primjer, označen crnim križićem, više nije potrebno pretražiti sve primjere iz skupa za učenje, označene točkama, nego samo one koje pripadaju njegovoj particiji. Međutim, particioniranjem prostora su iz susjedstva ispitnog primjera uklonjena dva bliska primjera susjednih particija, prikazana unutar kružnice, što rezultira manjom točnosti algoritma.



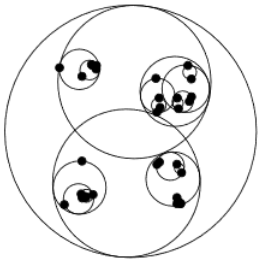
Slika 4.2: Particioniranje prostora k-d stablom

Stablo lopti

Smanjenje složenosti kod viskodimenzionalnih podataka ostvarivo je uvođenjem indeksne strukture stabla lopti (Omohundro, 1989) koja se temelji na sličnoj ideji poput k-d stabla. Autor je *lopte* definirao kao hipersfere d-dimenzionalnog ulaznog prostora, određene s koordinatama središta te radijusom. Tada je stablo lopti definirano kao potpuno binarno stablo u kojem svaki čvor predstavlja loptu, takvu da je ona najmanja lopta koja sadrži sve lopte predstavljene djecom tog čvora.

Stablo lopti slično je strukturi k-d stabla, uz razliku da se prostor više ne dijeli na međusobno isključive potprostore, nego na lopte za koje su međusobni presjeci dozvoljeni, a potpuna podjela prostora neobavezna. Zbog dozvoljenih presjeka između lopti podatak može biti obuhvaćen s nekoliko lopti, međutim pripadat će samo onoj čijem je središtu najbliži.

Postupak izgradnje stabla ovisi o podatcima te nije jednoznačno definiran (Omohundro, 1989). U radu je predloženo nekoliko pristupa od kojih neki rezultiraju sličnim podjelama prostora kao kod k-d stabala, dok drugi omogućavaju rješavanje problema prokletstva dimenzionalnosti. Takvi pristupi u obzir uzimaju raspored podataka u prostoru te prema tome dijele prostor samo na dijelovima u kojima su podatci prisutni. Podjela takvim pristupom prikazana je slikom 4.3 koja je preuzeta iz originalnog rada.

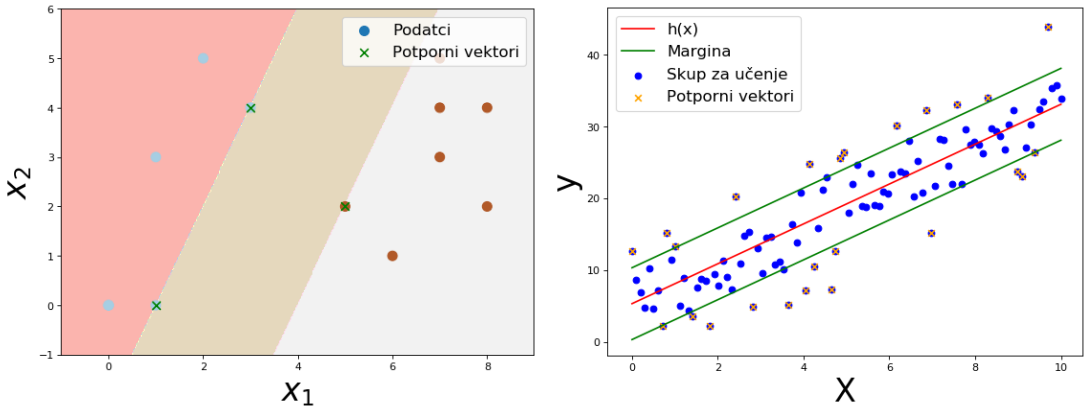


Slika 4.3: Podjela prostora stablom lopti (Omohundro, 1989)

4.4. Regresija potpornih vektora

U ovom poglavlju opisan je algoritam regresije zasnovan na stroju potpornih vektora (engl. *support vector machines*, SVM). Iako je SVM izvorno definiran kao algoritam linearne klasifikacije, kasnije se proširio i na slučajeve nelinearne klasifikacije te regresiju (Drucker et al., 1997) i grupiranje.

Motivacija za odabir algoritma proizlazi iz rezultata dosadašnjih istraživanja (Müller et al., 1997; Sapankevych i Sankar, 2009) koja upućuju na visoku primjenjivost



(a) Linearni SVM **(b)** Linearni SVR

Slika 4.4: Usporedba margina modela SVM i SVR

algoritma u području predviđanja vremenskih serija. U nastavku poglavlja slijedi opis algoritma te komentar odabira hiperparametara s kojima je definiran.

4.4.1. Opis algoritma

Temeljna ideja algoritma regresije stroja potpornih vektora, u nastavku SVR, zasnovana je na klasifikacijskom algoritmu potpornih vektora, čije se poznavanje u ovom poglavlju pretpostavlja. Njihova osnovna razlika, koja ujedno omogućava regresiju, odnosi se na korak učenja, odnosno na način određivanja margine.

Kod klasifikacijskog algoritma margina je određena maksimizacijom udaljenosti decizijske granice i najbližih primjera za učenje koji dolaze iz različitih klasa što rezultira marginom koja ne sadrži niti jedan primjer za učenje (pod pretpostavkom tvrde margine). Primjer određivanja margine linearnim SVM-om i skupom linearno odvojivih podataka prikazan je na slici 4.4a, gdje su podatci i dijelovi prostora koji pripadaju pojedinoj klasi obojeni jednakim bojama, dok je margina prikazana svijetlosmeđim pojasom. Potporni vektori nalaze se na samom rubu margine te su označeni križićima.

Za razliku od SVM-a, SVR će marginu odrediti obrnutim pristupom, tako da se unutar nje nađe što više primjera za učenje. Promjena pristupa određivanja margine povlači i promjenu definicije potpornih vektora. Tako su kod SVR-a potporni vektori određeni primjerima iz skupa za učenje koji se nalaze na rubu margine ili izvan nje. Primjer linearne regresije provedene modelom SVR prikazan je na 4.4b, gdje $h(x)$ predstavlja naučenu linearnu funkciju koja najbolje opisuje prikazane podatke. Detaljniji opis metode određivanja margine bit će opisan u nastavku poglavlja.

Kao i kod SVM-a model je moguće opisati i korištenjem principa dualnosti u ko-

jem se optimizacijski postupak iz primarnog problema, zbog jednostavnijeg rješavanja, prebacuje u dualni. Tako će model u slučaju odabira primarne formulacije pripadati parametarskim modelima, dok će dualna formulacija rezultirati neparametarskim modelom. Opis obje formulacije te motivacija uvođenja dualne dani su u nastavku poglavlja.

4.4.2. Primarna formulacija

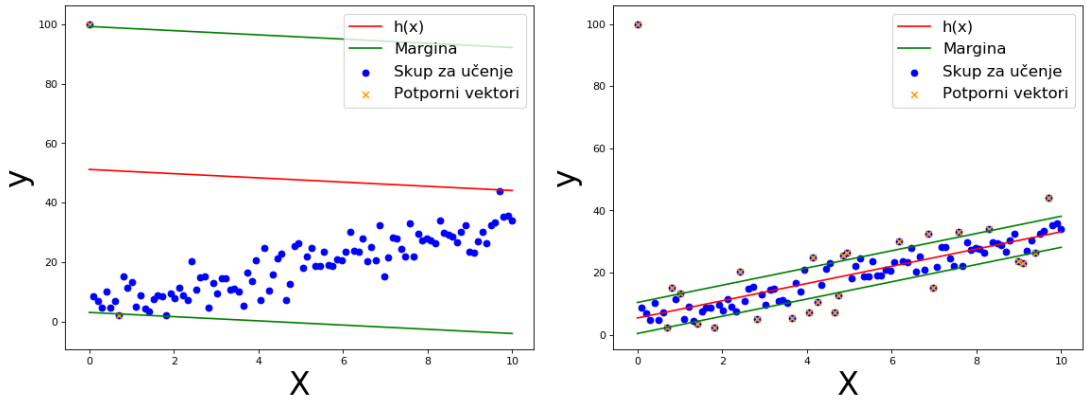
Model je u primarnoj formulaciji predstavljen običnim linearnim modelom iz jednadžbe 4.16 prema kojoj se predikcija donosi računanjem skalarnog umnoška vektora parametara te ulaznog primjera uz dodavanje pomaka.

$$h(\vec{x}^{(i)}) = \vec{w}^{\intercal} \cdot \vec{x}^{(i)} + w_0 \tag{4.16}$$

Specifičnost prema kojoj se SVR razlikuje od linearne regresije, predstavljene istim modelom, nalazi se u definiciji funkcije pogreške i koraku optimizacije. Tako će se učenje SVR-a svesti na određivanje najjednostavnije hipoteze $h(\vec{x})$ za koju vrijedi da pogreška predikcije svakog primjera iz skupa za učenje nije veća od zadane pozitivne konstante ε, zbog čega se model naziva i ε-SVR.

Tada se margina SVR-a definira pomoću udaljenosti ε od hipoteze $h(\vec{x})$ čime se definira dio ravnine ulaznog prostora unutar kojeg se nalaze svi primjeri za učenje. Zbog specifičnog izgleda taj se dio ravnine naziva i ε-cijev (engl. ε-*tube*), a potporne vektore čine oni primjeri koji leže na njenom rubu. Prikaz navedenog vidljiv je na slici 4.5a.

Ovakva formulacija može se zapisati kao problem konveksne optimizacije funkcije pogreške u ovisnosti o parametrima modela uz ograničenje. Navedeni optimizacijski problem matematički je definiran jednadžbama iz 4.17. Prema njoj se određivanje optimalnog vektora parametara modela svodi na minimizaciju njegove norme uz zadovoljenje ograničenja kojim se uvjetuje najveće dopušteno odstupanje od hipoteze. Motivacija određivanja navedene funkcije cilja proizlazi iz činjenica kako povećanje vrijednosti komponenti vektora parametara $\vec{w}$ vodi prema složenijim modelima. Tako će minimizacija kvadratne norme vektora parametara uz ograničenja dovesti do pronalaska najjednostavnijeg modela za kojeg navedena ograničenja vrijede. Zato se član unutar funkcije pogreške naziva regularizacijskim članom.



(a) Prije uvođenja varijabli labavosti **(b)** Nakon uvođenja varijabli labavosti

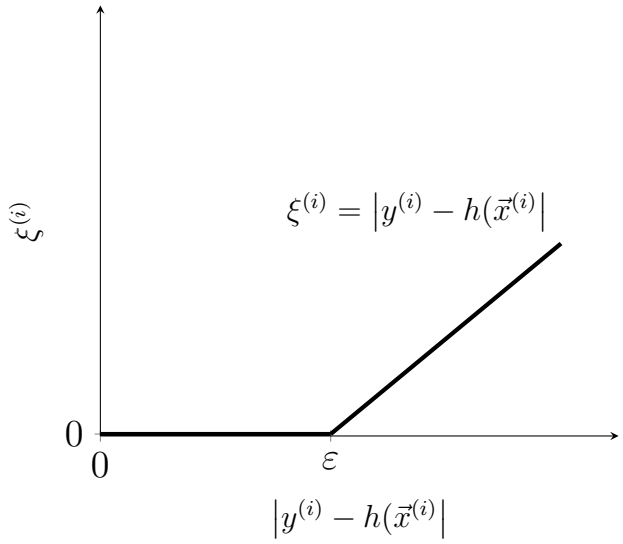
Slika 4.5: Utjecaj stršećih vrijednosti na model ε-SVR

$$\begin{aligned} \min_{\vec{w}} \quad & E(\vec{w}|D) = \frac{1}{2} \cdot \|w\|^2 \\ \text{uz ograničenje} \quad & \left|y^{(i)} - h(\vec{x}^{(i)})\right| \leq \varepsilon, \ i = 1, \ldots, N. \end{aligned} \tag{4.17}$$

Najveća dopuštena predikcijska pogreška na pojedinom primjeru iz skupa za učenje zadana je hiperparametrom ε koji, uz vektora parametara, utječe na složenost modela. Smanjenje iznosa hiperparametra dovodi do sve manjih pogrešaka tijekom učenja, ali ujedno postrožava uvjet iz ograničenja čime se prostor mogućih rješenja smanjuje što na kraju može prouzročiti isključivanje svih mogućih rješenja te nemogućnost rješavanja problema iz 4.17.

Dodatni problem u zadanom optimizacijskom problemu uzrokuju stršeće vrijednosti u unutar skupa za učenje koje, zbog unošenja značajnih predikcijskih pogrešaka, postojanje rješenja uvjetuju proširivanjem margine na iznos njihove predikcijske pogreške. Takav slučaj prikazan je slikom 4.5a gdje je vidljivo kako stršeća vrijednost odvlači hipotezu, od većine podataka, prema sebi što dovodi do loše naučenog modela.

Neovisnost širine margine o stršećim vrijednostima, uz garanciju postojanja rješenja optimizacijskog problema 4.17, moguće je ostvariti modifikacijom funkcije pogreške u koju se uvode varijable labavosti (engl. *slack variable*). One su označene N-dimenzijskim vektorom $\vec{\xi}$ kojem je svaka komponenta povezana s odgovarajućim primjerom za učenje. Njime se rješenja koja prema ograničenjima iz optimizacijskog problema 4.17 nisu bila dozvoljena, sada dozvoljavaju uz dodatno kažnjavanje onih primjera iz skupa za učenje za koje je predikcijska pogreška veća od zadanog ε. Time se dozvoljava izlazak primjera iz skupa za učenje izvan margine što vodi prema robusnijim modelima koji se više ne prilagođavaju šumu i stršećim vrijednostima. Rezultat



Slika 4.6: Gubitak zglobnice

uvođenja varijabli labavosti prikazan je na slici 4.5b, gdje je vidljiv nestanak utjecaja stršećih vrijednosti na izgled hipoteze.

Novi optimizacijski problem koji nastaje uvođenjem varijabli labavosti u 4.17 definiran je jednadžbama 4.18. Komponente vektora $\xi^{(i)}$ određuju se prema jednadžbi 4.19, iz čega je vidljivo kako se one računaju samo za potporne vektore, dok su za ostale postavljene na nulu. Tada je iznos komponenti koje odgovaraju potpornim vektorima jednak razlici predikcijskih pogrešaka pojedinog potpornog vektora i zadanog ε. Pojedini gubitci uzrokovani varijablama labavosti prikazani su na slici 4.6 te se zbog karakterističnog izgleda nazivaju gubitkom zglobnice (engl. *hinge loss*).

$$\begin{aligned} \min_{\vec{w}} \quad & E(\vec{w}|D) = \frac{1}{2} \cdot \|w\|^2 + C \cdot \sum_{i=0}^{N} \xi^{(i)},\ C \geq 0 \\ \text{uz ograničenja} \quad & \left|y^{(i)} - h(\vec{x}^{(i)})\right| \leq \varepsilon + \xi^{(i)},\ i = 1, \ldots, N. \\ & \xi^{(i)} \geq 0,\ i = 1, \ldots, N. \end{aligned} \tag{4.18}$$

$$\xi^{(i)} = \max(0, \left|y^{(i)} - h(\vec{x}^{(i)})\right| - \varepsilon) \tag{4.19}$$

Osim varijabli labavosti, optimizacijski problem 4.18 uvodi i hiperparametar njihove kazne, označen slovom *C*. Njime se definira kompromis između jednostavnosti modela i iznosa najveće dopuštene predikcijske pogreške. Tako će, za male pozitivne vrijednosti (oko nule), u funkciji pogreške dominirati regularizacijski član, dok će suma predikcijskih pogrešaka većih od ε manje utjecati na ukupni iznos pogreške. Time se u optimizacijskom postupku preferiraju hipoteze definirane jednostavnijim pa-

rametrima. Vizualno, naučeni model imat će užu marginu koja se, zbog zanemarivanja predikcijskih pogrešaka većih od ε, nije prilagodila šumu i stršećim vrijednostima.

Povećanje vrijednosti *C* uzrokuje pridavanje sve većeg značaja varijablama labavosti u odnosu na regularizacijski član funkcije pogreške. Zato će se prilikom minimizacije funkcije pogreške preferirati one margine koje obuhvaćaju što više primjera skupa za učenje što vodi prema širim marginama, odnosno složenijim modelima.

Kako ε utječe na odabir hiperparametra kazne, a oni zajedno na složenost modela, njihovo određivanje značajno utječe na performanse modela te nije jednostavno. Detaljniji opis postupaka njihovog određivanja iznesen je u poglavlju 4.4.4.

Nakon definiranja modela i funkcije pogreške, za potpuno rješavanje problema u primarnoj formulaciji preostaje riješiti optimizacijski problem iz 4.18. U tu svrhu moguće je koristiti metode unutarnje točke, metode kazni te gradijenti spust. Međutim, zbog niza prednosti navedenih u nastavku, umjesto primarne formulacije problema, bolje je koristiti dualnu.

4.4.3. Dualna formulacija

Dualna formulacija temelji se ideji Lagrangeovih multiplikatora, kojom se ograničenja iz optimizacijskog problema izravno ugrađuju u funkciju cilja. Time optimizacijski problem s ograničenjima postaje problem bez ograničenja.

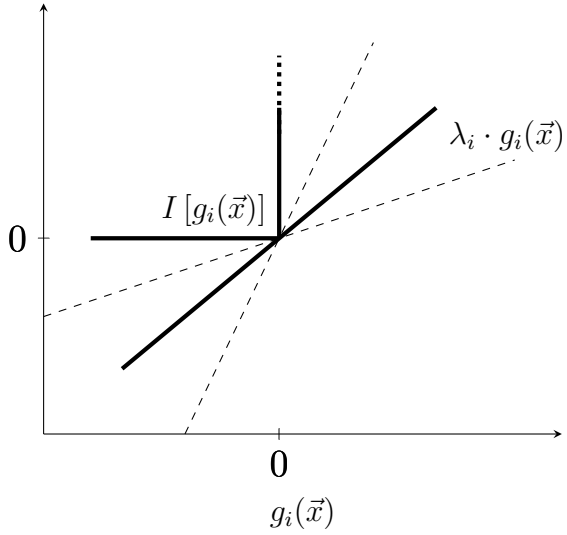
Lagrangeova funkcija

Primarni problem postavljen s 4.18 moguće je poopćiti uvođenjem proizvoljnog broja ograničenja nejednakosti. Poopćenje je, pod pretpostavkom konveksnosti funkcije cilja i svih ograničenja, dano izrazom 4.20.

$$\begin{aligned} &\min_{\vec{x}} && f(\vec{x}) \\ &\text{uz ograničenja} && g_i(\vec{x}) \leq 0,\ i = 1, \ldots, p. \end{aligned} \tag{4.20}$$

Ugrađivanje ograničenja nejednakosti u funkciju cilja ostvarivo je promjenom funkcije cilja tako da ona, u slučaju zadovoljenja svih ograničenja nejednakosti ostaje nepromijenjena, dok se narušavanjem bilo kojeg ograničenja $g_i(\vec{x})$ ona postavlja na beskonačnu vrijednost. Time se u postupku optimizacije rješenja iz neostvarivog područja sigurno eliminiraju.

Formalno, nova funkcija cilja definirana je jednadžbom 4.21, u kojoj $I[g_i(\vec{x})]$ predstavlja beskonačnu *step*-funkciju definiranu jednadžbom 4.22.



Slika 4.7: Prikaz beskonačne *step*-funkcije i njene donje granice

$$J(\vec{x}) = f(\vec{x}) + \sum_{i=1}^{p} I[g_i(\vec{x})] \tag{4.21}$$

$$I[u] = \begin{cases} 0, & \text{ako } u \leq 0 \\ \infty, & \text{inače} \end{cases} \tag{4.22}$$

Iako ovaj pristup rješava problem ugrađivanja ograničenja u funkciju cilja, dobivena funkcija cilja, zbog beskonačne *step*-funkcije, ne posjeduje svojstva derivabilnosti i neprekidnosti što ju čini nepogodnom za optimizaciju. Zato se za funkciju cilja uzima njena aproksimacija, odnosno donja ograda, koja će biti derivabilna i neprekidna. Funkcija s navedenim svojstvima kojom se aproksimira beskonačna *step*-funkcija je pravac pozitivnog nagiba, postavljen u ishodište. Beskonačna *step*-funkcija i njena aproksimacija pravcem prikazani su slikom 4.7.

Tada se za svako ograničenje $g_i(\vec{x})$ definira odgovarajući pravac $\lambda_i g_i(\vec{x})$ kojim se aproksimira beskonačna *step*-funkcija. Kako je koeficijent nagiba pravca λ_i uvijek pozitivan ili jednak nuli, u slučaju zadovoljenja ograničenja $g_i(\vec{x})$, vrijednosti definirane pravcem $\lambda_i g_i(\vec{x})$ bit će manje ili jednake nuli, dok će narušavanje ograničenja rezultirati pozitivnim vrijednostima.

Aproksimacijom beskonačne *step*-funkcije, iz jednadžbe 4.21, pomoću pravca dobiva se Lagrangeova funkcija definirana jednadžbom 4.23. Koeficijent nagiba λ_i iz iste jednadžbe naziva se Lagrangeovim multiplikatorom. Time je nastala derivabilna, neprekidna funkcija cilja u kojoj su kodirana sva ograničenja nejednakosti iz primarnog problema.

$$L(\vec{x}, \vec{\lambda}) = f(\vec{x}) + \sum_{i=1}^{p} \lambda_i \cdot g_i(\vec{x}),\ \lambda_i \geq 0, \forall i \in \{1, \ldots, p\} \tag{4.23}$$

Rješavanjem problema kodiranja ograničenja u funkciju cilja uvedeni su Lagrangeovi multiplikatori predstavljeni varijablom $\vec{\lambda}$ koju je, uz $\vec{x}$, također potrebno odrediti. Multiplikatori se nazivaju i dualnim varijablama, dok se varijabla $\vec{x}$ naziva primarnom varijablom. Njihovo određivanje uvjetovano je što boljom aproksimacijom funkcije cilja $J(\vec{x})$ koja je ovisno o zadovoljenju ograničenja bila jednaka $f(\vec{x})$ ili beskonačnosti.

Promatrajući sliku 4.7, u slučaju zadovoljenja ograničenja, $g_i(\vec{x})$ poprima vrijednosti iz prostora negativnih vrijednosti određenih s osi apscisa. Tada je za najbolju aproksimaciju *step*-funkcije pravac potrebno položiti, tako da bude paralelan s osi apscisa što se može postići postavljanjem multiplikatora λ_i na nulu. Suprotno, ako ograničenje nije zadovoljeno, $g_i(\vec{x})$ poprima vrijednosti iz prostora pozitivnih vrijednosti na osi apscisa pa je najbolja aproksimacija *step*-funkcije pravac okomit na os apscisa koji se postiže odabirom multiplikatora koji teži prema beskonačnosti. Prilagođavanje pravca funkciji $J(\vec{x})$, ovisno o zadovoljenju ograničenja, prikazano je crtkanim pravcima na slici 4.7.

Takvo ponašanje moguće je formalizirati jednadžbom 4.24 u kojoj se maksimizacijom Lagrangeove funkcije po svim Lagrangeovim multiplikatorima postiže najbolja aproksimacija funkcije $J(x)$.

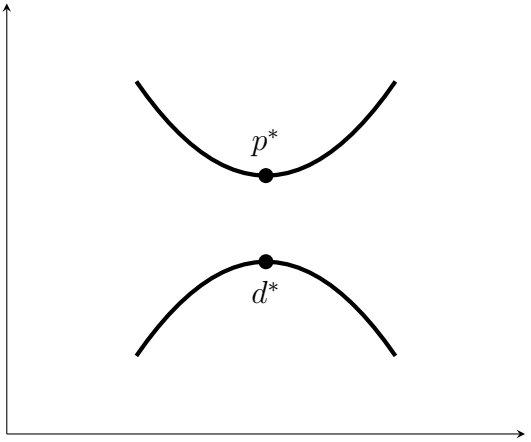
$$J(\vec{x}) = \max_{\vec{\lambda}} L(\vec{x}, \vec{\lambda}) \tag{4.24}$$

Za rješavanje izvornog problema potrebno je minimizirati $f(\vec{x})$, odnosno $J(\vec{x})$ što povlači minimizaciju izraza 4.24 po primarnoj varijabli. Time se dobiva izraz 4.25 koji predstavlja primarni problem optimizacije bez ograničenja, gdje p^* predstavlja vrijednost optimalnog rješenja tog problema.

$$p^* = \min_{\vec{x}} \max_{\vec{\lambda}} L(\vec{x}, \vec{\lambda}) \tag{4.25}$$

Lagrangeova dualnost

Prijelaz u dualnu formulaciju temelji se na načelu dualnosti prema kojem se optimizacijski problem može zapisati u primarnom te odgovarajućem dualnom obliku. Zbog aproksimacije funkcije $J(\vec{x})$ njenom donjom granicom rješenje dualnog problema, označeno s d^*, će uvijek biti donja granica primarnog. Navedeno svojstvo prikazano je



Slika 4.8: Skica dualnog procjepa

slikom 4.8 te izrazom 4.26. Zbog moguće nejednakosti dualnog i primarnog rješenja tim se izrazom definira svojstvo slabe dualnosti koja vrijedi i za nekonveksne primarne probleme (Boyd i Vandenberghe, 2004). Razlika pronađenih rješenja se tada naziva dualnim procjepom (engl. *duality gap*) koji je skiciran na slici 4.8.

$$d^* \leq p^* \tag{4.26}$$

Jednakost rješenja vrijedit će samo u slučaju konveksnosti optimizacijskog problema iz 4.16 te zadovoljenja slabog Slaterovog uvjeta, objašnjenog u nastavku. Tada se svojstvo jednakosti rješenja primarne i dualne formulacije naziva jakom dualnosti koja je opisana izrazom 4.27. Navedeni slabi Slaterov uvjet definira se za problem iz 4.16, a proizlazi iz strože formulacije uvjeta kojom se zahtijeva poštivanje svih ograničenja nejednakosti $g_i(\vec{x})$. Oslabljivanjem uvjeta dozvoljava se kršenje onih ograničenja čije su funkcije ograničenja afine funkcije (Boyd i Vandenberghe, 2004).

$$d^* = p^* \tag{4.27}$$

Kako je dualni problem donja ograda primarnog tada se minimizacija primarnog problema svodi na maksimizaciju dualnog problema, što je vidljivo iz slike 4.8. Tada se funkcija cilja dualnog problema naziva dualnom Lagrangeovom funkcijom koja je definirana u jednadžbi 4.28.

$$\widetilde{L}(\vec{\lambda}) = \min_{\vec{x}} L(\vec{x}, \vec{\lambda}) \tag{4.28}$$

Njenom se maksimizacijom postavlja dualni problem određen izrazom 4.29 čijim se rješavanjem dobiva optimalno rješenje dualnog problema d^* iz jednadžbe 4.30. Ovaj kvadratni program se od primarnog razlikuje prema broju varijabli funkcije cilja. Tako prelaskom iz primarnog u dualni problem broj varijabli više ne ovisi o dimenzionalnosti vektora $\vec{x}$, nego o njihovoj brojnosti unutar skupa za učenje. Koristi ovakvog prijelaza očituju se primjenom posebnih algoritama rješavanja ovog kvadratnog programa, temeljenih na algoritmu slijedne minimalne optimizacije (engl. *sequential minimal optimization*, SMO). Njegova složenost kvadratno ovisi o broju primjera, dok drugi algoritmi za rješavanje kvadratnih programa primarnog problema imaju kubnu složenost u ovisnosti o dimenzionalnosti ulaznog vektora.

$$\begin{aligned} &\max_{\vec{\lambda}} && \widetilde{L}(\vec{\lambda}) \\ &\text{uz ograničenje} && \lambda_i \geq 0, \forall i \in \{1, \ldots, p\} \end{aligned} \tag{4.29}$$

$$d^* = \max_{\vec{\lambda}} \widetilde{L}(\vec{\lambda}) = \max_{\vec{\lambda}} \min_{\vec{x}} L(\vec{x}, \vec{\lambda}) \tag{4.30}$$

Karush-Kuhn-Tuckerovi uvjeti

U slučaju optimizacijskog problema bez ograničenja, globalni optimum određen je točkom za koju je gradijent jednak nuli. Prilikom uvođenja ograničenja u optimizacijski problem taj uvjet više ne mora vrijediti jer se točka globalnog optimuma nužno ne nalazi unutar dozvoljenog područja. Tada se za probleme s ograničenjima definiraju Karush-Kuhn-Tuckerovi uvjeti (Karush, 1939; Kuhn i Tucker, 1951), kraće pisani kao KKT uvjeti, koji su opisani prema (Boyd i Vandenberghe, 2004).

Prvi KKT uvjet definiran je u slučaju jake dualnosti. Tada primarna varijabla minimizira $L(\vec{x}, \vec{\lambda}^*)$, u kojoj dualna varijabla označena zvjezdicom predstavlja optimalnu vrijednost. Kako su sva ograničenja kodirana Lagrangeovom funkcijom, minimizacija po primarnoj varijabli odgovara optimizaciji bez ograničenja prema kojoj gradijent u točki optimuma mora biti jednak nuli. Uvjet je prikazan jednadžbom 4.31 prema kojoj se može i interpretirati. Vidljivo je kako za točku optimuma gradijent funkcije cilja primarnog problema mora imati jednak smjer, ali suprotnu orijentaciju gradijentu kodiranih ograničenja.

$$\nabla_x L(\vec{x}^*, \vec{\lambda}^*) = \nabla_x f(\vec{x}^*) + \sum_{i=1}^{p} \lambda_i^* \cdot \nabla_x g_i(\vec{x}^*) = 0 \tag{4.31}$$

Drugi KKT uvjet proizlazi iz izraza 4.32 - 4.35 koji su objašnjeni u nastavku. U slučaju jake dualnosti vrijedi jednakost primarnog i dualnog rješenja što povlači jednakost vrijednosti početne funkcije cilja primarnog problema za optimalnu primarnu varijablu i dualne Lagrangeove funkcije za optimalnu dualnu varijablu. Kako je dualna funkcija dobivena minimizacijom Lagrangeove funkcije po primarnoj varijabli ona se može zapisati preko jednakosti 4.33. Nejednakost 4.34 vrijedi jer se minimizacija zapravo odnosi na pronalaženje infimuma, odnosno najveća donja granice Lagrangeove funkcije koja je prema definiciji uvijek manja ili jednaka vrijednosti funkcije za optimalnu varijablu. Tada, zbog nenegativnosti dualnih varijabli i vrijednosti funkcije ograničenja koja su manja ili jednaka 0, suma njihovih umnožaka također mora biti manja ili jednaka 0 što dovodi do nejednakosti iz 4.35.

$$f(\vec{x}^*) = \widetilde{L}(\vec{\lambda}^*) \tag{4.32}$$

$$= \min_{\vec{x}} \left(f(\vec{x}) + \sum_{i=1}^{p} \lambda_i^* \cdot g_i(\vec{x})\right) \tag{4.33}$$

$$\leq f(\vec{x}^*) + \sum_{i=1}^{p} \lambda_i^* \cdot g_i(\vec{x}^*) \tag{4.34}$$

$$\leq f(\vec{x}^*) \tag{4.35}$$

Ulančavanjem izraza vidljivo je kako su oni zadovoljeni jedino u slučaju prelaska nejednakosti u jednakost čime se zahtijeva da suma iz 4.34 bude jednaka 0. Kako su multiplikatori nenegativni, a vrijednosti ograničenja manja ili jednaka 0, suma može rezultirati nulom samo u slučaju kada je svaki od umnožaka jednak 0.

Time se određuje drugi KKT uvjet definiran u jednadžbi 4.36 iz koje proizlazi njegova interpretacija. Prema njoj se globalni optimum može naći unutar područja dozvoljenog pojedinim ograničenjem $g_i(\vec{x})$ čime ono ne dodaje nove informacije u postupak optimizacije te tako postaje neaktivno. Takva se ograničenja gase postavljanjem odgovarajućih multiplikatora na nulu. U suprotnom, kada ograničenje nije zadovoljeno, multiplikator poprima pozitivne vrijednosti te uvjet vrijedi samo ako je vrijednost ograničenja jednaka 0. Objašnjenje proizlazi iz činjenice kako se kod funkcija kojima globalni optimum nije dosezljiv zbog ograničenja ekstremi uvijek nalaze na rubu ograničenja u kojima je vrijednost ograničenja uvijek jednaka 0.

$$\lambda_i^* \cdot g_i(\vec{x}^*) = 0, \ \forall i \in \{1, \ldots, p\} \tag{4.36}$$

Ostale KKT uvjete čine ograničenja primarnog problema te dualnih varijabli koje moraju biti nenegativne.

Dualni model SVR-a

Općenito izvedene jednadžbe iz prethodnih odjeljaka moguće je primijeniti na primarni optimizacijski problem SVR-a postavljen izrazom 4.17. Tako se izvod s rješenjem dualnog problema za problem SVR-a nalazi u (Smola i Schölkopf, 2004), dok je u jednadžbi 4.37 prikazan konačni rezultat kojim se definira dualni model SVR-a. Prema njoj dualni model SVR-a više ne ovisi o vektoru težina $\vec{w}$, kao u primarnom modelu, nego samo o primjerima iz skupa za učenje $\vec{x}^{(i)}$ te dualnim varijablama čiji je broj definiran veličinom skupa za učenje. Time je model iz parametarskog postao neparametarski.

$$h(\vec{x}, \vec{\lambda}) = \sum_{i}^{N} \lambda_i \cdot \vec{x}^{(i)} \cdot \vec{x} + w_0 \tag{4.37}$$

Prema gornjoj jednadžbi, dualni model određen je skalarnim umnoškom vektora primjera iz skupa za učenje te vektora ulaznog primjera za kojeg se nepoznata labela određuje. Važno je primijetiti kako prema drugom KKT uvjetu svi primjeri skupa za učenje $\vec{x}^{(i)}$ koji se nalaze u područje gdje je ograničenje neaktivno, odnosno koji se nalaze unutar ε-cijevi, imaju pripadni multiplikator λ_i postavljen na nulu. Time se oni isključuju iz modela te preostaju samo primjeri $\vec{x}^{(i)}$ koji leže izvan ε-cijevi, odnosno potporni vektori. Ovisnost o podskupu primjera za učenje rezultira rijetkim modelom kojeg je teže prenaučiti, a lakše izračunati te interpretirati.

Druga bitna prednost dualnog modela nad primarnim je lakoća prelaska u nelinearnost. Modeliranje nelinearnosti primarnim linearnim modelom ostvarivo je preslikavanjem ulaznih primjera u prostor više dimenzije korištenjem baznih funkcija, kao i kod nelinearne regresije u poglavlju 4.2.2. Problem ovog pristupa predstavlja odabir dimenzije preslikavanja koja uvijek ovisi o podatcima što ga u primjeni čini značajno složenijim od demonstracijskog primjera sa slika 4.1. Pogrešan odabir vodi do loših modela koji će u slučaju odabira previsoke dimenzije postati prenaučeni, dok će odabir preniske dimenzije onemogućiti modeliranje nelinearnosti te rezultirati podnaučenim modelom.

Dualni model navedene probleme rješava modeliranjem nelinearnosti pomoću jezgrenog trika (engl. *kernel trick*). Skalarni produkt iz 4.37 prema geometrijskoj interpretaciji predstavlja sličnost ulaznog vektora $\vec{x}$ i svakog primjera skupa za učenje te

se kao takav može zamijeniti jezgrenom funkcijom $\kappa(\vec{x}^{(i)}, \vec{x})$. Uvjeti pod kojima se jezgrena funkcija ponaša kao skalarni produkt prostora određene dimenzije zadani su Mercerovim teoremom (Mercer, 1909). Neke od često korištenih jezgrenih funkcija uključenih u programske knjižnice uključuju linearne i polinomijalne jezgre te radijalne bazne funkcije.

Prednosti korištenja jezgrenih funkcija nad preslikavanjem podataka očituju se u lakšem odabiru funkcije u odnosu na dimenziju preslikavanja te smanjenu računalnu složenost izračuna jezgrene funkcije u odnosu na skalarni produkt dobivenih visokodimenzionalnih vektora.

Osim primjene jezgrenog trika, dualna formulacija rezultira optimiranjem po dualnim varijablama $\vec{\lambda}$, umjesto po vektoru značajki $\vec{w}$ što dozvoljava primjenu algoritma SMO. Kako je broj dualnih varijabli ograničen brojem potpornih vektora takva optimizacija često je pogodnija u odnosu na primarni problem s potencijalno visokodimenzionalnim vektorom značajki.

4.4.4. Odabir hiperparametara algoritma

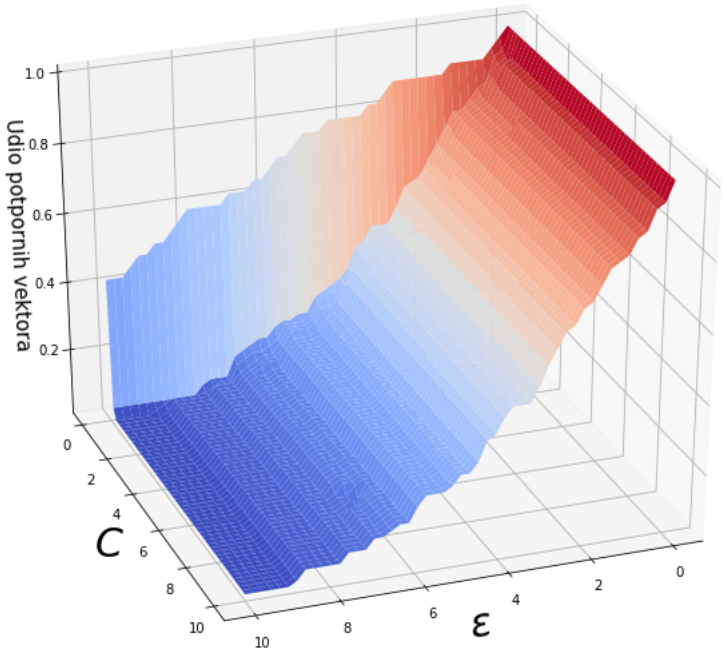
Kako složenost modela istovremeno ovisi o hiperparametrima C i ε potrebno je definirati metodu odabira koja dovodi do primjerene složenosti modela. Hiperparametar ε utječe na širinu ε-cijevi te tako na broj potpornih vektora definiranih primjerima iz skupa za učenje koji se nalaze izvan nje. Utjecaj ovog hiperparametra na složenost očituje se u dualnoj formulaciji gdje složenost modela i procesa optimizacije izravno ovisi o broju potpornih vektora. Tako će modeli određeni manjim hiperparametrom imati užu ε-cijev što povlači veću točnost na skupu za učenje, ali rezultira većim brojem potpornih vektora. Kako su parametri modela predstavljeni potpornim vektorima njihovo povećanje dovodi do složenijih modela čime se povećava vjerojatnost pojave prenaučenosti. Jednostavniji modeli imat će veću vrijednost hiperparametra te će ε-cijev biti šira što dovodi do manjeg broja potpornih vektora čime se proces optimizacije te model pojednostavljuju.

Složenost modela ovisi i o hiperparametru C kojim se regulira kažnjavanje primjera koji se nalaze izvan margine. Za vrlo visoke vrijednosti hiperparametra funkcija pogreške iz 4.18 će brzo rasti, čak i za mala odstupanja od margine definirana s $\xi^{(i)}$ pa će regularizacijski član funkcije pogreške biti zanemariv u odnosu na iznos sume produkata kazne i varijabli labavosti. Zato će model, u svrhu minimizacije funkcije pogreške, biti prisiljen smanjivati sumu varijabli labavosti na štetu regularizacijskog člana što rezultira povećanjem složenosti te u konačnici prenaučenosti.

Smanjenje vrijednosti hiperparametra C dovodi do slabljenja utjecaja sumi varijabli labavosti na funkciju pogreške iz 4.18 u korist regularizacijskog člana. Tako će za odabir premalih vrijednosti ovog hiperparametra u iznosu funkcije pogreške dominirati regularizacijski član čime će model u procesu optimizacije biti prisiljen smanjivati svoju složenost kako bi minimizirao njezin iznos. Ovakav scenarij rezultira podnaučenim modelom koji postaje prejednostavan te ne može dobro opisati podatke.

Slika 4.9 prikazuje utjecaj hiperparametara C i ε na udio potpornih vektora u odnosu na veličinu skupa za učenje. Za generiranje slike korišten je linearni model SVR-a te umjetni skup podataka koji predstavlja zašumljeni pravac. Vidljivo je kako na udio potpornih vektora dominantno utječe hiperparametar ε. Zato se u literaturi (Schölkopf et al., 1998) predlaže zamjena ε s hiperparametrom ν kojim bi se omogućilo izravno određivanje njihovog udjela.

Literatura ne daje jednoznačan odgovor na pitanje odabira navedenih hiperparametara pa se u praksi koriste razne metode od kojih je najpoznatija metoda unakrsne provjere koja će biti korištena u praktičnom dijelu rada. Pregled postojećih i prijedlog naprednije metode odabira C i ε dostupan je u (Cherkassky i Ma, 2004).



Slika 4.9: Utjecaj hiperparametara C i ε na broj potpornih vektora

5. Studijski slučaj prognoziranja vodostaja Kupe

U ovom poglavlju opisan je studijski slučaj predviđanja vodostaja rijeke Kupe kod mjesta Farkašić kojim bi se omogućio uvid kretanja vodostaja u bliskoj budućnosti. Iako je prognoza izvedena za mjernu postaju Farkašić, opisani postupci su, uz manje izmjene, primjenjivi i za ostale mjerne postaje, dok su glavni koraci, poput analize te izgradnje označenog skupa podataka primjenjivi i kod drugih vremenskih serija.

5.1. Dostupni podatci

Sve korištene podatke praktičnog dijela rada ustupio je Državni hidrometeorološki zavod Republike Hrvatske. Oni uključuju podatke dobivene mjerenjima na brojnim hidrološkim i kišomjernim postajama porječja rijeke Kupe kroz razne periode te su prikazani tablicom 5.1. Podatci hidroloških postaja nastali su mjerenjima dnevnih vodostaja i protoka rijeke Kupe, Dobre, Mrežnice, Korane i Gline, dok su dnevne količine oborina dobivene mjerenjima padalina u mjestima Parg, Delnice, Lokve, Bosiljevo, Ogulin, Plaški, Slunj, Karlovac i Pisarovina. Sve mjerne postaje prikazane su slikom 5.1 na kojoj su hidrološke, ovisno o rijeci na kojoj se nalaze, označene različitim bojama, dok su kišomjerne predstavljene plavim oznakama. Na slici su postaje rijeke Kupe obilježene crvenim oznakama, Dobre žutim, Mrežnice smeđe-zelenim, Korane narančastim, a Gline zelenim oznakama.

5.1.1. Hidrološki podatci

Podatci dobiveni mjerenjima hidroloških postaja uključuju dnevne vodostaje izražene u centimetrima te protoke izražene u metrima kubnim po sekundi. Iznimke su mjerne postaje Karlovac, Glina i Pribanjci za koje su dostupni samo podatci vodostaja.

U određenim periodima podatci na nekoliko postaja sadrže nedostajuće vrijednosti

Tablica 5.1: Skup podataka

Vremenska serija	Tip mjerne postaje	Mjerna jedinica	Tip podataka
Vodostaj	Hidrološka	cm	cijeli broj
Protok	Hidrološka	m^3/s	broj s pomičnim zarezom
Padaline	Kišomjerna	mm/m^2	broj s pomičnim zarezom

Slika 5.1: Lokacije mjernih postaja

koje su označene labelom -777, dok stršećih vrijednosti koje bi nastale pogreškama prilikom mjerenja ili zapisivanja nema. Prema tipu podatci pripadaju cijelim brojevima koji jedino u slučaju vodostaja mogu biti negativni. Takvi zapisi, osim onih koji definiraju nedostajuće vrijednosti, nisu rezultat pogreške, nego prirodnog procesa produbljivanja korita rijeke uzrokovanog protjecanjem vode. Time se korito spušta ispod razine referentne točke mjerenja za koju se uzima očitanje vodostaja jednako nuli.

Na vodostaj određene postaje utječu vodostaji njenih uzvodnih postaja pa je, ovisno o njihovim lokacijama, potrebno definirati njihove odnose. Tako su one, krećući se nizvodno, za pojedinu rijeku navedene u tablici 5.2. U njoj su također navedena ušća svake rijeke uz oznaku prve mjerne postaje glavnog riječnog toka nakon ušća.

Tablica 5.2: Hidrološke postaje

Rijeka	Hidrološke postaje	Ušće [Hidrološka postaja]
Kupa	Hrvatsko, Kupa, Zapeć, Pribanjci, Ladešići, Kamanje, Brodarci, Karlovac, Rečica, Jamnička Kiselica, Šišinec, Farkašić	-
Dobra	Turkovići Ogulinski, Lešće Toplice, Stative Donje	Kupa [Brodarci]
Mrežnica	Juzbašići, Mrzlo Polje	Korana [Karlovac]
Korana	Sastavci, Slunj, Veljun, Velemerić, Karlovac	Kupa [Karlovac]
Glina	Široka Rijeka, Vranovina, Glina	Kupa [Farkašić]

5.1.2. Podatci padalina

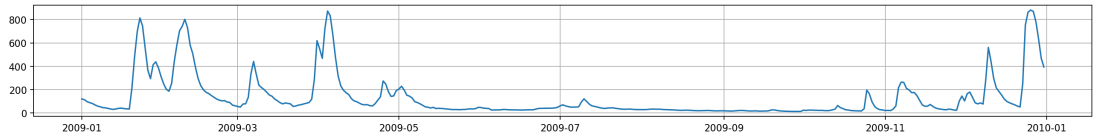
Padaline mjerene na kišomjernim postajama porječja Kupe predstavljene su pozitivnim brojevima s pomičnim zarezom kojima se izražava njihova količina u milimetrima po četvornom metru. Podatci padalina dostupni su za razdoblje 2009. do kraja 2014. godine te ne sadrže nedostajuće ili stršeće vrijednosti.

5.1.3. Prikaz podataka

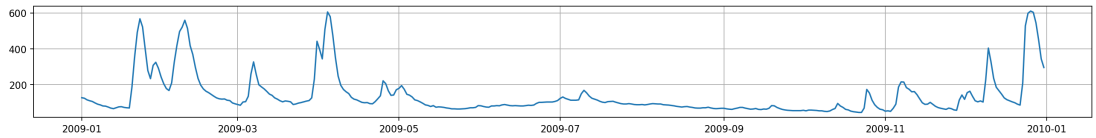
Vremenske serije protoka, vodostaja i padalina na području Pisarovine koje uključuje hidrološku postaju Jamnička Kiselica prikazane su slikom 5.2. Na njoj su, zbog sažetosti prikaza, vremenske serije prikazane za 2009. godinu, dok su ostala razdoblja i mjerne postaje prikazane unutar interaktivne Jupyter bilježnice.

Prema slikama 5.2a i 5.2b je vidljivo kako se vremenske serije protoka i vodostaja ponašaju vrlo slično, dok se vremenska serija padalina, prikazana slikom 5.2c, uglavnom sastoji od impulsnih događaja koji se odražavaju na iznose vodostaja i protoka.

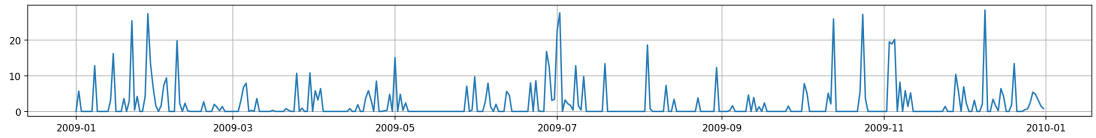
Sažeti prikaz svih podataka jedne hidrološke postaje i njoj najbliže kišomjerne postaje ostvaren je kutijastim dijagramom (engl. *box plot*) sa slike 5.3 koja prikazuje njihove podatke grupirane po mjesecima. Iz njih se jasno može uočiti ponašanje srednje vrijednosti protoka i vodostaja kroz mjesece kao i relativno velik broj stršećih vrijednosti. Također, zbog ovisnosti iznosa vremenskih serija o vremenu tijekom analize podataka treba razmotriti utjecaj komponente sezonalnosti.



(a) Protok - Jamnička Kiselica



(b) Vodostaj - Jamnička Kiselica



(c) Padaline - Pisarovina

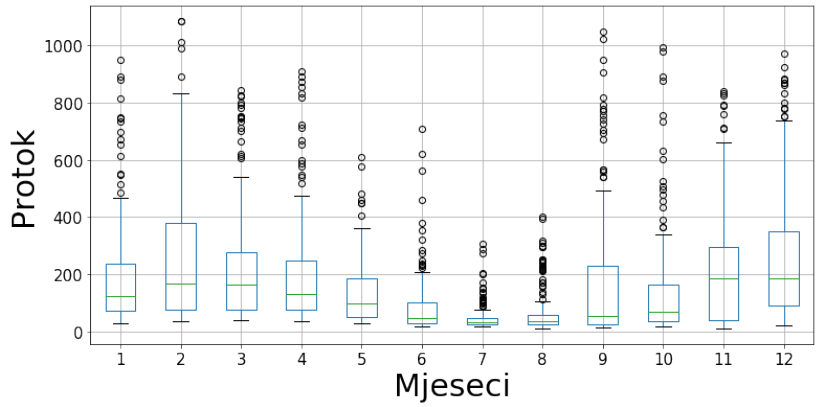
Slika 5.2: Protok, vodostaj i padaline na području Pisarovine za 2009. godinu

5.2. Analiza podataka

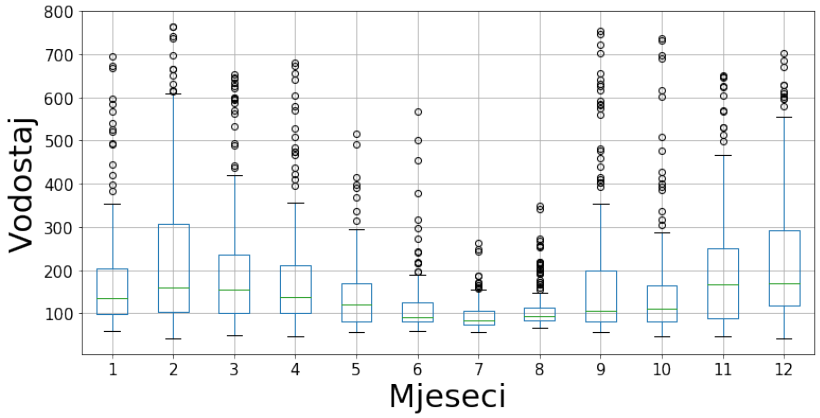
Prije implementacije modela potrebno je napraviti inicijalnu analizu podataka koja uključuje ispitivanje kvalitete podataka, vizualizacije, testiranje svojstva stacionarnosti te analizu povezanosti vremenskih serija. Time će se omogućiti donošenje odluke o uključivanju pojedine postaje u skup podataka. Rezultati cjelokupne analize podataka, koja uključuje sve dostupne mjerne postaje uz njihovu vizualizaciju iz prethodnog poglavlja, su pohranjeni u interaktivnu Jupyter bilježnicu.

5.2.1. Dekompozicija vremenskih serija

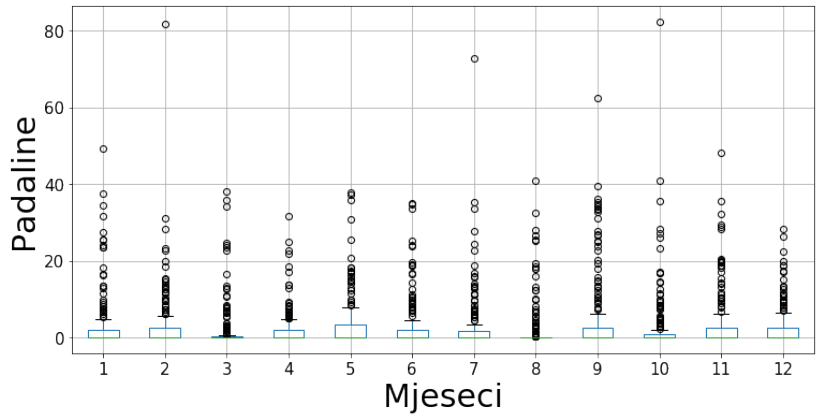
Dekompozicija vremenskih serija izvedena je za cjelokupni period koji uključuje šest godina dnevnih podataka te je prikazana slikom 5.4. Budući da se radi o hidrometeorološkim podatcima ne čudi kako grafovi pokazuju izraženu sezonsku komponentu. Iako grafovi također upućuju na postojanje rastućeg trenda, zbog prirode podataka nije opravdano zaključiti njegovo dugoročno postojanje koje bi se nastavilo u godinama za koje podatci nisu prisutni.



(a) Protok - Jamnička Kiselica

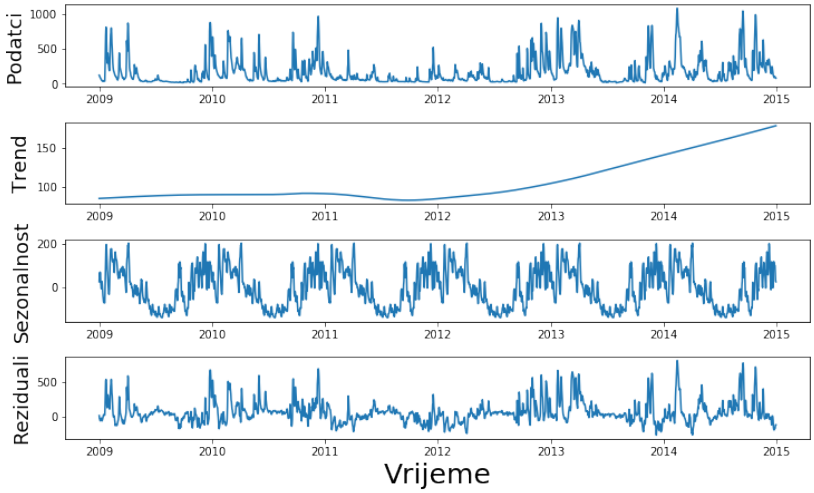


(b) Vodostaj - Jamnička Kiselica

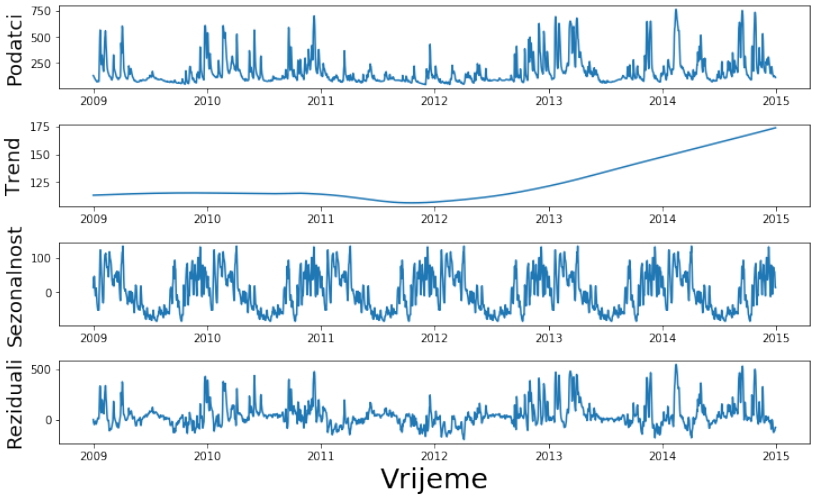


(c) Padaline - Pisarovina

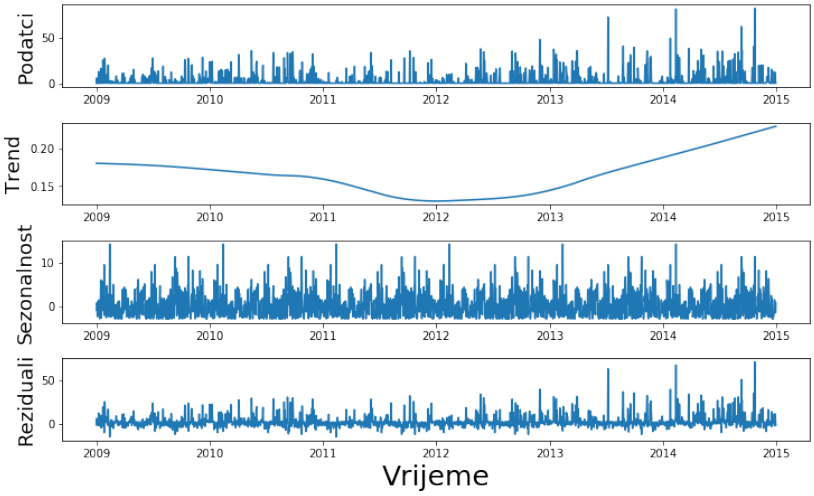
Slika 5.3: Kutijasti dijagrami svih podataka protoka, vodostaja i padalina na području Pisarovine



(a) Dekompozicija protoka - Jamnička Kiselica



(b) Dekompozicija vodostaja - Jamnička Kiselica



(c) Dekompozicija padalina - Pisarovina

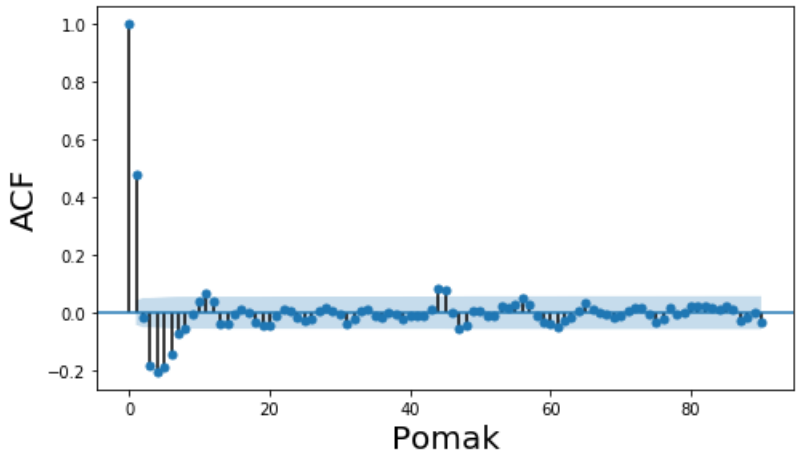
Slika 5.4: Dekompozicija vremenskih serija na području Pisarovine

5.2.2. Ispitivanje svojstva stacionarnosti

Radi potrebe modeliranja vremenske serije stohastičkim modelima nužno je utvrditi njezinu stacionarnost metodama opisanim u poglavlju 2.3. Za provođenje oba testa stacionarnosti korištena je programska knjižnica StatsModels, a testovi su, zbog velikog broja vremenskih serija kojima se stacionarnost ispitivala, provođeni automatizirano. Tako se stacionarnost svake vremenske serije ispitala za zadani interval pouzdanosti sa svim ponuđenim opcijama vremenskog zaostajanja te komponentama trenda i pomaka.

Rezultati s 95-postotnim intervalom pouzdanosti pokazuju kako većina vremenskih serija nije stacionarna za sve ponuđene konfiguracije parametara testa. Zato su testovi ponovljeni nad diferenciranim podatcima, nakon čega su svi postali stacionarni s jednakim intervalom pouzdanosti. Time se zaključuje potreba postupka diferenciranja koje će biti primijenjeno na sve vremenske serije.

Osim statističkim testovima, tvrdnju stacionarnosti podataka moguće je dodatno ojačati analizom grafova autokorelacije prema Box-Jenkinsovoj metodi opisanoj u poglavlju 3.2.3. Tako će graf autokorelacijske funkcije za stacionarne vremenske serije imati oblik eksponencijalnog odumiranja, dok će kod nestacionarnih procesa pad biti sporiji. Slikom 5.5 je prikazan graf autokorelacijske funkcije za podatke vodostaja mjerne postaje Jamnička Kiselica. Grafovi vodostaja i protoka mjerenih na drugim postajama slični su prikazanom, dok autokorelacijska funkcija padalina svih mjernih postaja iščezne već nakon prvih par pomaka. Prikazani rezultati vode prema zaključku o stacionarnosti svih vremenskih serija.



Slika 5.5: Korelogram ACF vodostaja - Jamnička Kiselica

5.2.3. Analiza korelacija vremenskih serija

Analiza korelacije provedena je s ciljem identifikacije vremenskih serija povezanih s vremenskom serijom koju se prognozira. Tako će one vremenske serije koje su značajno korelirane s labelama činiti kandidate značajki prilikom izgradnje skupa za učenje. Dodatno, analizom korelacija kandidata značajki otkrivaju se redundantne značajke koje se izbacuju. Time se dimenzionalnost skupa podataka smanjuje,a postupak učenja ubrzava.

Slikom 5.6 prikazan je korelogram svih vremenskih serija, dok se korelogrami uz navedene iznose koeficijenata korelacija te dijagrami raspršenja svih parova vremenskih serija nalaze u Jupyter bilježnici. Na korelogramu sa slike svaka je serija označena lokacijom mjerne postaje te sufiksom *h*, *q* i *r*, ovisno o tome radi li se o vodostaju, protoku ili padalinama.

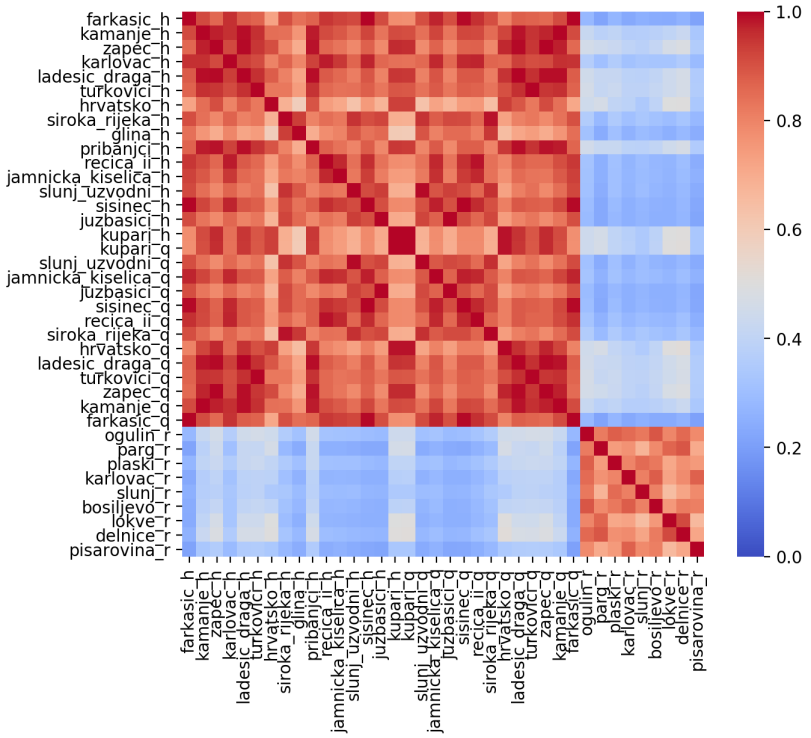
Nad parovima vremenskih serija izračunati su Pearsonov i Spearmanov koeficijent korelacije kojima se mjeri linearna, odnosno nelinearna korelacija. Nakon analize dijagrama raspršenja koji pokazuju postojanje nelinearnih ovisnosti protoka i vodostaja za relevantnu mjeru odabran je Spearmanov koeficijent korelacije. Rezultati očekivano pokazuju najveću korelaciju vremenskih serija protoka i vodostaja mjerenih na istim postajama. Nakon njih slijede korelacije istovrsnih tipova podataka koje, uz nekoliko iznimki, variraju u ovisnosti o udaljenosti mjernih postaja za koje se korelacija računa. Najmanji iznosi korelacija dobiveni su za parove hidroloških podataka i padalina.

Korelacija nužno ne povlači kauzalnost pa je za donošenje odluka o korisnosti uključivanja podataka jedne vremenske serije u proces predviđanja druge, osim jake korelacije, potrebno provjeriti i smislenost njihovog međudjelovanja. Time se izbjegava uključivanje značajki koje imaju visok stupanj korelacije s labelama, ali između njih i labele ne postoji uzročno-posljedična veza čime bi takve značajke ometale model.

5.2.4. Odabir mjernih postaja

Odluka o uključivanju mjernih postaja u skup podataka donesena je temeljem rezultata provedene analize podataka. Time se iz svih dostupnih podataka odabiru relevantni što dovodi do uklanjanja redundantnih i nebitnih podataka koji bi povećali skup za učenje bez uvođenja novih informacija.

Prvi kriterij odabira postaje odnosi se na razdoblje mjerenja. Budući da su hidrološki podatci dobiveni mjerenjima kroz različite periode, a podatci o padalinama tijekom razdoblja 2009. do 2014. godine, odlučeno je kako svaka odabrana mjerna postaja



Slika 5.6: Spearmanovi koeficijenti korelacije

mora sadržavati mjerenja iz tog perioda. Zatim su postaje filtrirane prema lokacijama i kvaliteti podataka tako da se od postaja koje su lokacijski jedna blizu druge odabere ona s boljom kvalitetom podataka, odnosno s manje nedostajućih vrijednosti. Posljednji kriterij vezan je uz udaljenost mjerne postaje od ušća rijeke na kojoj se nalazi čime prednost odabira imaju postaje udaljenije od ušća. Motivacija za uvođenje ovog kriterija temelji se na činjenici da su informacije hidroloških postaja pritoka koje se nalaze neposredno prije ušća sadržane u podatcima postaje koje se nalazi na glavnom riječnom toku nakon ušća.

Odabrane hidrološke postaje podcrtane su u tablici 5.2, dok su od kišomjernih odabrane sve osim postaje Lokve koja se nalazi blizu postaje Delnice.

5.3. Kriterij uspješnosti prognoze

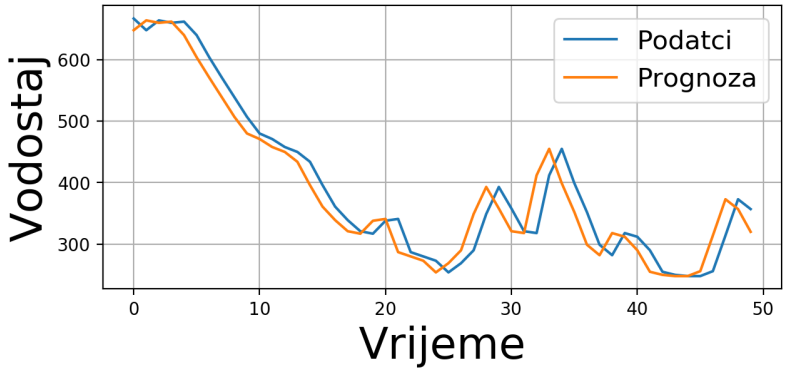
Za procjenu uspješnosti prognoze potrebno je definirati kriterije uspjeha pojedinih metoda te metrike temeljem kojih se oni vrednuju. Tako su kriteriji uspješnosti za prognoze modela strojnog učenja zadani rezultatima stohastičkih modela ARIMA i VAR, opisanih u poglavlju 3, dok je za njih uspješnost definirana osnovnim modelom prognoze kojim se ona ostvaruje kopiranjem prethodne vrijednosti iz serije. Odabrane

metrike vrednovanja uključuju prosječno kvadratno odstupanje (engl. *mean squared error*, MSE) te prosječno apsolutno odstupanje (engl. *mean absolute error*, MAE), od kojih je za odabir modela korištena metrika MSE. Formule za njihovo izračunavanje dane su izrazima 5.1 i 5.2.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - h(\vec{x}^{(i)}))^2 \tag{5.1}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y^{(i)} - h(\vec{x}^{(i)})\right| \tag{5.2}$$

Slika 5.7 prikazuje primjer prognoze osnovnog modela. Prognostičke pogreške po ravnim dijelovima vremenske serije, gdje nema velikih odstupanja susjednih vrijednosti su male, dok se kod naglih promjena susjednih vrijednosti one znatno povećavaju. Ovaj model je u primjeni beskoristan jer ne daje informacije o promjeni vrijednosti sve dok se one ne dogode. Zato će njegova ukupna prognostička pogreška nad ispitnim skupom poslužiti kao osnovni kriterij uspjeha kojeg ostali modeli moraju zadovoljiti kako bi se proglasili uspješnima.



Slika 5.7: Primjer prognoze osnovnim modelom

Procjena uspješnosti stohastičkih modela te modela strojnog učenja provedena je nad ispitnim skupom kojeg čine podatci posljednje godine, dok se odabir hiperparametara temeljio na rezultatima dobivenih na validacijskom skupu podataka sastavljenog od podataka iz 2013. godine. U nastavku slijedi opis postupka unakrsne provjere vremenskih serija kojim se procjenjuje kvaliteta modela te odabire skup hiperparametara stohastičkih modela, dok je postupak procjene kvalitete modela strojnog učenja te odabir njihovih hiperparametara objašnjen naknadno.

Procjena konačne kvalitete i odabir hiperparametara stohastičkih modela temelje se na iterativnom postupku kojim se oponaša proces prognoziranja nakon postavljanja

modela u produkciju. Ulazni parametri postupka uključuju veličinu početnog skupa za učenje, broj ispitnih ili validacijskih primjera te željeni korak prognoze. Postupak započinje učenjem modela na početnom skupu za učenje nakon čega slijedi prognoziranje željenog koraka. Dobivena prognozirana vrijednost uspoređuje se s labelom na odgovarajućoj lokaciji u validacijskom ili ispitnom skupu koja je određena korakom prognoze. Npr. za jednodnevne prognoze validacijski ili ispitni primjer činit će prva iduća vrijednost nakon kraja skupa za učenje, dok će za sedmodnevne prognoze to biti sedma vrijednost od posljednje vrijednosti iz skupa za učenje. Time prvi korak postupka završava nakon čega slijedi proširivanje skupa za učenje jednim primjerom iz ispitnog ili validacijskog skupa te ponavljanje opisanog postupka. Završetkom postupka se temeljem izračunatih prognostičkih pogrešaka računaju metrike MSE i MAE čime se dobiva ocjena modela.

Odabirom primjerene početne veličine skupa za učenje izbjegava se provjeravanje modela treniranih nad nereprezentativnim skupovima podataka za učenje u početnim iteracijama postupka uzrokovanih njihovom premalom veličinom. Takvi skupovi za učenje bi uzrokovali značajno veće prognostičke pogreške u odnosu na one dobivene kasnijim iteracijama kada bi, zbog povećanja brojnosti primjera koji ga čine, skup postao reprezentativan. Radi izbjegavanja tog problema početni skup za učenje je u slučaju validacije postavljen na prve četiri godine podataka, dok se prilikom testiranja postavlja na prvih pet godina podataka čime se izbjegava preklapanje validacijskog i ispitnog skupa.

Mana opisanog postupka je njegova dugotrajnost koja proizlazi iz potrebe za ponovnim učenjem modela nakon svake iteracije. Kako su odabrani modeli strojnog učenja određeni brojnim hiperparametrima, od kojih su neki u eksperimentima postavljeni na veliki raspon vrijednosti, učenje modela nakon svake iteracije za svaku od konfiguracija zadanih hiperparametara nije moguća. Zato je odabir modela strojnog učenja proveden jednostavnijim postupkom u kojem se model uči na fiksnom skupu za učenje koji se sastoji od prve četiri godine podataka nakon čega se hiperparametri biraju na validacijskom skupu koji sadrži podatke naredne godine. Istim postupkom obavljeno je ispitivanje kvalitete modela strojnog učenja.

5.4. Prognoziranje stohastičkim modelima

Prognoze stohastičkim modelima ARIMA i VAR ostvarene su programskom knjižnicom StatsModels. U nastavku slijedi opis prognoza te komentar dobivenih rezultata.

5.4.1. Prognoziranje modelom ARIMA

Modeli ARIMA omogućuju prognozu univarijatnih vremenskih serija pa ulazne podatke čini vremenska serija vodostaja hidrološke postaje Farkašić. Budući da modeli ARIMA imaju ugrađeno diferenciranje zadanog reda, ulazne podatke nije potrebno diferencirati.

Okvirni odabir hiperparametara proveden je Box-Jenkinsovom metodom, opisanom u poglavlju 3.2.3. Ona sugerira odabir drugog reda autoregresijskog modela i modela pomičnih prosjeka te prvi red diferenciranja. Zbog mogućih problema u određivanju hiperparametara modela kod vremenskih serija koje su nastale mješavinom autoregresijskog i modela pomičnih prosjeka te radi povećanja točnosti modela provedena je pretraga po rešetci nad susjedstvom okvirno odabranih hiperparametara. Tako je korištenjem postupka vrednovanja opisanog u poglavlju 5.3 ustanovljeno kako model ARIMA$(3, 1, 2)$ daje najbolje rezultate na validacijskom skupu nakon čega su performanse odabranog modela provjerene na ispitnom skupu. Rezultati provjere na ispitnom skupu zapisani su tablicom 5.5. Prema njima, model ARIMA zadovoljava kriterij uspješnosti za sve korake prognoze.

Kako je model ARIMA određen trima hiperparametrima, nedostatak njegove primjene veže se uz dugotrajnost pretrage optimalnih hiperparametara. Druga mana vezana je uz loše rezultate višednevnih prognoza koje proizlaze iz svojstva univarijatnosti modela zbog čega se informacije drugih vremenskih serija, koje utječu na promatranu, ne mogu iskoristiti. S druge strane, agnostičnost prema području primjene vezanog uz podatke, jednostavnost te ugrađeno diferenciranje predstavljaju glavne prednosti ovog modela.

5.4.2. Prognoziranje modelom vektorske autoregresije

Model VAR ne sadrži ugrađenu komponentu diferenciranja pa je podatke postaja odabranih prema poglavlju 5.2.4 prije predaje algoritmu potrebno diferencirati. Prema zaključcima iz poglavlja 5.2.2 odabran je prvi red diferenciranja kojim su podatci postali stacionarni.

Raspon hiperparametra najvećeg dozvoljenog pomaka postavljen je na prvih sedam prirodnih brojeva, a optimalne vrijednosti birane su temeljem ponuđenih informacijskih kriterija. Uz tako odabrane pomake u eksperiment je uključena najveća vrijednost pomaka postavljena na 7 te vrijednost pomaka kojom se minimizira pogreška skupa za učenje. Kako je postupak ispitivanja kvalitete modela iterativan te zahtijeva učenje modela u svakom koraku, čime se omogućuje odabir različitih pomaka kroz iteracije,

izvješće kvalitete modela uključuje medijan odabranih pomaka za svaki od modela definiranih informacijskim kriterijem.

Najjednostavniji model odabran je BIC-om te sadrži samo jedan vremenski pomak, dok su AIC te kriterij minimizacije pogreške skupa za učenje rezultirali najsloženijim modelom s pet vremenskih pomaka. Ostali načini odabira rezultirali su modelima koji su prema odabranom pomaku između njih. Najbolji rezultati postignuti su primjenom Hannan-Quinnovog informacijskog kriterija koji je odabrao jednostavan model s dva vremenska pomaka čiji su rezultati provjere na ispitnom skupu dani tablicom 5.5.

Prema njoj su iznosi pogrešaka značajno manji u odnosu na osnovni model te model ARIMA što je uzrokovano uključivanjem ostalih vremenskih serija koje utječu na vodostaj rijeke Kupe kod mjesta Farkašić. Upravo svojstvo multivarijatnosti modela VAR predstavlja glavnu prednost u odnosu na model ARIMA. Osim toga, model je definiran jednim hiperparametrom što rezultira značajno bržim postupkom njegovog odabira u odnosu na model ARIMA. Odabir vremenskih serija koje utječu na prognoziranu zahtijeva određeno domensko znanje pa je korištenje ovog modela složenije u odnosu na model ARIMA.

5.5. Prognoziranje modelima strojnog učenja

Iduća potpoglavlja opisuju postupak izgradnje označenog skupa podataka te daju popis algoritama i korištenih hiperparametara nad kojima je provedena pretraga za optimalnim modelom.

5.5.1. Izgradnja označenog skupa podataka

Modeli regresije pripadaju nadziranoj vrsti strojnog učenja pa je za njihovo učenje potrebno konstruirati označeni skup podataka. U tu svrhu korištena je knjižnica otvorenog koda Pandas (McKinney, 2010) koja nudi funkcionalnosti za učinkovitu i jednostavnu manipulaciju podatcima te stvaranje i obradu podataka vremenskih serija.

Izgradnja označenog skupa podataka vođena je idejom generiranja parova značajki i labela za svaki datum vremenske serije, ovisno o zadanom koraku prognoze. Tako će svaki označeni primjer za pojedini datum sadržavati skup značajki koji predstavlja sve informacije poznate do tog datuma, dok će odgovarajuća labela, ovisno o koraku prognoze, sadržavati stvarnu vrijednost prognozirane varijable. Za skup značajki odabrane su prošle vrijednosti prognozirane te drugih vremenskih serija koje na nju utječu, dok je labela predstavljena stvarnim vrijednostima prognozirane vremenske serije. Do-

Tablica 5.3: Primjer označenog skupa podataka

Datum	farkasic_h(t)	farkasic_h(t-2)	farkasic_h(t-1)	parg_r(t-1)
2009-01-01	248	NaN	NaN	NaN
2009-01-02	246	NaN	248.0	1.4
2009-01-03	235	248.0	246.0	5.3
2009-01-04	227	246.0	235.0	0.2
2009-01-05	218	235.0	227.0	0.3
2009-01-06	211	227.0	218.0	0.0
2009-01-07	200	218.0	211.0	0.0
2009-01-08	194	211.0	200.0	0.0
2009-01-09	187	200.0	194.0	8.4
2009-01-10	182	194.0	187.0	0.0

datno, u skup značajki je moguće ugraditi druge podatke koji nisu predstavljeni vremenskim serijama, a kojima bi model dobio dodatne informacije čime bi se povećala točnost prognoze. Kako njihovo identificiranje zahtijeva domensko znanje, značajke tog tipa vezane uz hidrometeorološke podatke u radu nisu korištene.

Primjer označenog skupa podataka za jednokoračnu prognozu vodostaja mjerne postaje Farkašić prikazan je tablicom 5.3. U njoj kolona *farkasic_h(t)* predstavlja labelu kojom se, na određeni datum, definira vodostaj, dok ostale kolone, osim datuma, predstavljaju značajke. Prvi i drugi redak tablice sadrže nedefinirane vrijednosti značajki zaostalih vremenskih serija, označenih s NaN, jer na te datume one zbog nepostojanja prethodnih podataka nisu poznate. Takvi su zapisi, prije predaje algoritmu strojnog učenja, uklonjeni iz skupa označenih primjera.

Postupak izgradnje označenog skupa podataka parametriziran je vremenskim serijama značajki, njihovim vremenskim zaostajanjima, vremenskom serijom koja čini labelu, korakom prognoze te odlukom o diferenciranju labele. Parametri vremenske serije koja određuje labelu te korak prognoze unaprijed su zadani, dok su ostali parametri podložni optimizaciji na validacijskom skupu. Vremenske serije značajki, kao i kod modela VAR, odabrane su prema poglavlju 5.2.4, dok su vremenska zaostajanja te odluka o diferenciranju labele dobivene optimizacijom na validacijskom skupu podataka.

S obzirom na to da su podatci dobiveni mjerenjima hidroloških i meteoroloških postaja, radi smanjenja složenosti pretrage hiperparametri zaostajanja vremenskih serija grupirani su prema njenom tipu po kojem su dobili imena *HydroLag* i *MeteoLag*.

Njima se zadaje najveći dozvoljeni vremenski pomak čime se svi pomaci manji od njega implicitno uključuju u skup značajki. Time će sve značajke vodostaja i protoka dijeliti iste korake vremenskog zaostajanja, dok će se za značajke padalina definirati zasebni. Prikaz navedenog dan je u tablici 5.3, gdje su postavljanjem vrijednosti *HydroLag* na 2 odabrani prvi i drugi korak vremenskog zaostajanja vodostaja, dok je za padaline odabran prvi korak definiran hiperparametrom *MeteoLag* jednakim 1.

Glavni izazov prilikom implementacije ove funkcionalnosti povezuje se s vremenskom komponentom podataka vremenskih serija zbog koje se javlja opasnost od nenamjernog uključivanja nedozvoljenih informacija koje ne bi bile poznate u trenutku donošenja prognoze (engl. *data leakage*). Problem je osobito izražen prilikom izgradnje skupa podataka za višednevna prognoziranja kada je potrebno obratiti dodatnu pažnju te pregledati skup dobivenih značajki.

Dobiveni skup sadrži značajke različitih skala pa je prije provođenja postupka učenja provedena njihova transformacija oduzimanjem srednje vrijednosti te skaliranjem na jediničnu varijancu. Navedeni postupak dostupan je unutar programske knjižnice Scikit-learn u razredu `sklearn.preprocessing.StandardScaler`. Izračun srednje vrijednosti i varijance obavljen je samo nad skupom primjera za učenje čime se izbjegava otkrivanje podataka skupa za validaciju te ispitivanje.

5.5.2. Odabrani modeli

U eksperimentima su korišteni algoritmi opisani poglavljem 4 koji uključuju linearni model regresije, regresiju algoritmom najbližih susjeda te regresiju potpornih vektora. Za navedene algoritme hiperparametri su određeni postupkom pretrage po rešetci nad validacijskim skupom podataka u koju je ugrađen postupak pronalaska optimalnih hiperparametara *HydroLag*, *MeteoLag* te odluka o diferenciranju labela. Parametri *HydroLag* i *MeteoLag* postavljeni su na raspon prvih sedam prirodnih brojeva čime se ostvaruje najviše jednotjedno zaostajanje vremenskih serije značajki.

Tablicom 5.4 su definirani pronađeni optimalni hiperparametri pojedinog algoritma, dok je u nastavku dan popis svih hiperparametara te njihovih vrijednosti nad kojima je pretraga po rešetci obavljena.

Linearni model regresije

Programska knjižnica Scikit-learn nudi više razreda kojima je moguće ostvariti različite tipove linearne regresije. Eksperimenti uključuju običnu te regulariziranu linearnu regresiju, predstavljene razredima `LinearRegression` i `Ridge` unutar mo-

dula `sklearn.linear_model`, dok je preslikavanje podataka u višu dimenziju ostvareno razredom `sklearn.preprocessing.PolynomialFeatures`.

Pretragom su obuhvaćeni hiperparametri regularizacijskog faktora te stupnja preslikavanja koji su u tablici 5.4 označeni s λ te *d*, a u postupku pretrage postavljeni na raspon vrijednosti $(0.01, 0.1, 1, 2, 5)$ i $(1, 2, 3)$. Prema dobivenim rezultatima, najbolje svojstvo generalizacije ostvareno je modelom linearne regresije, dok odluka o regularizaciji ovisi o koraku prognoze.

Regresija algoritmom *k*-najbližih susjeda

Algoritam regresije najbližim susjedima je u knjižnici Scikit-learn predstavljen razredom `sklearn.neighbors.KNeighborsRegressor` u kojem je omogućeno podešavanje hiperparametara broja susjeda, algoritam njihove pretrage te odabir metrike udaljenosti i njihovih težina kojima se regulira utjecaj pojedinog susjeda.

Metrika udaljenosti, definirana oznakom *dist* u tablici 5.4, postavljena je na Minkowskijevu, dok je hiperparametar njenog stupnja biran nad vrijednostima iz skupa $(1, 2)$ kojim se definiraju Manhattan i euklidska udaljenost. Doprinos svakog susjeda definiran je njegovom težinom koja ovisi o udaljenosti od promatranog primjera. Osim ponuđene jednolike težine svih primjera te težine definirane inverzom udaljenosti, razred nudi mogućnost definiranja vlastite metode kojom bi se one računale. Za potrebe ovog rada korištene su ponuđene težine od kojih je optimalni odabir predstavljen oznakom *w* u tablici 5.4.

Pretraga broja susjeda, označenog s *k*, provedena je nad susjedstvom korijena broja primjera za učenje koji se u praksi pokazao dobrim početnim izborom, dok su mogući algoritmi njihovog odabira, opisani u poglavlju 4.3.5, postavljeni na *k*-d stabla te stablo lopti od kojih je optimalni odabir naveden u tablici 5.4 pod oznakom *alg*.

Odabrani hiperparametri nešto su ujednačeniji u odnosu na linearnu regresiju, međutim ponovno ne daju jednoznačan odgovor o najboljim hiperparametrima neovisnima o koraku prognoze.

Regresija potpornih vektora

Algoritam regresije potpornih vektora predstavljen je razredima `LinearSVR` te `SVR` unutar modula `sklearn.svm`. Razlog implementacije algoritma linearne regresije potpornih vektora posebnim razredom opravdan je učinkovitom implementacijom koja je ostvariva u slučaju linearnosti te dolazi do izražaja prilikom učenja nad velikim skupovima podataka koji su prema službenoj dokumentaciji definirani s nekoliko desetaka

tisuća primjera za učenje.

Optimirani hiperparametri uključuju širinu margine te parametar kazne varijabli labavosti koji su u tablici 5.4 označeni ε i C. Pretraga širine margine provedena je nad vrijednostima iz skupa $(0.1, 1.0, 10)$, dok je parametar kazne određen pretragom skupa $(0.1, 1.0)$. Osim njih, razred `SVR`, kojim se ostvaruje prelazak u nelinearnost, nudi odabir jezgrene funkcije. Tako su jezgrene funkcije postavljene na radijalnu baznu funkciju te polinomijalnu jezgru kojoj su mogući stupnjevi polinoma postavljeni na drugi i treći red. Odabrana jezgrena funkcija i stupanj polinoma unutar tablice 5.4 označeni su s *kernel* i *d*.

Rezultati, kao i u dosadašnjim slučajevima, ovise o koraku prognoze što dovodi do zaključka o učenju posebnog modela za pojedini korak prognoze.

5.6. Rezultati

Konačni rezultati svakog algoritma po koraku prognoze, dobiveni nad ispitnim skupom podataka, prikazani su tablicom 5.5, gdje su algoritmi regresije potpornih vektora, ovisno o tome radi li se o linearnom modelu, predstavljeni s LSVR i SVR, dok je algoritam regresije najbližih susjeda označen s *k*-NN.

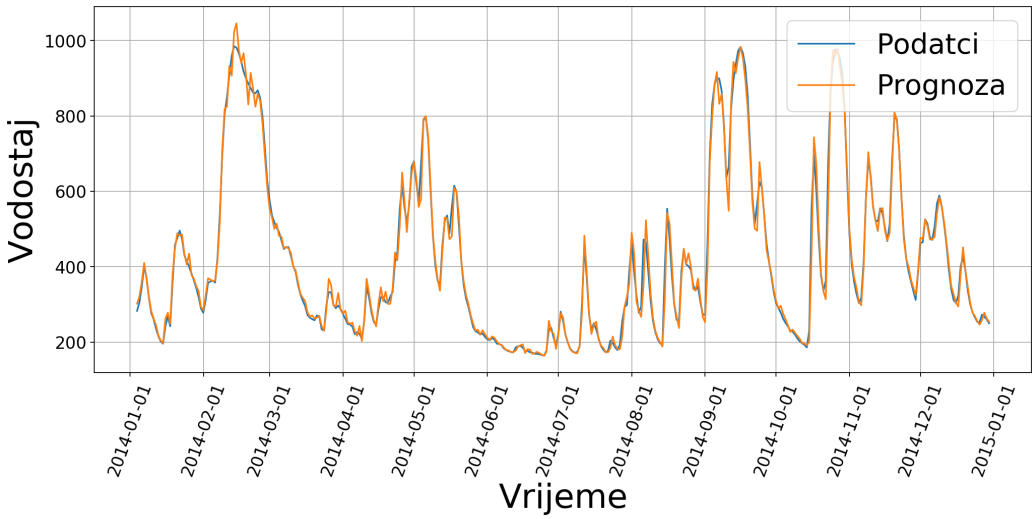
Prema rezultatima se za sve ispitane modele zaključuje zadovoljenje osnovnog kriterija uspješnosti definiranog osnovnim modelom. Drugi kriterij uspješnosti, vezan uz algoritme strojnog učenja, zadovoljen je u slučaju jednokoračne prognoze za model linearne regresije, dok je za ostale korake model VAR dao bolje rezultate. Zaključak uspješnosti linearnih modela potvrđuje tezu boljeg svojstva generalizacije jednostavnijih modela u odnosu na složenije.

U tablici su, uz metrike, prikazani i odabrani parametri skupa podataka *HydroLag* i *MeteoLag* kojima se definiraju vremenska zaostajanja hidroloških i meteoroloških značajki. Česta izostavljanja meteoroloških značajki, osobito za korake i modele koji su rezultirali najboljim prognozama, pokazuju manju važnost tih značajki u odnosu na hidrološke. Također, odabir relativno malih vremenskih pomaka hidroloških značajki kod uspješnih modela strojnog učenja pokazuje kako uključivanje viših pomaka ne donosi nove informacije.

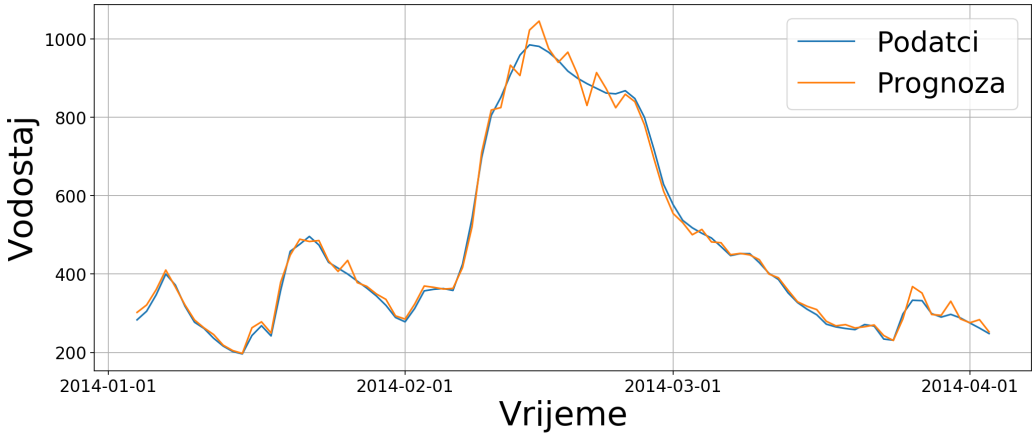
Grafovi jednodnevnih i višednevnih prognoza odabranim modelom linearne regresije, koja je od svih algoritama strojnog učenja dala najbolje rezultate, prikazani su slikama 5.9 i 5.8, dok su slikama 5.10 prikazani histogrami njegovih prognostičkih pogrešaka. Bitno svojstvo ostvarenih prognoza odnosi se na pravovremenost promjene trenda serije, kojeg osnovni model nije imao. Navedeno svojstvo prikazano je na slici

Tablica 5.4: Hiperparametri algoritama strojnog učenja

Korak prognoze	**Algoritam**			
	Linearni model regresije	LSVR	SVR	k-NN
1	λ=1.0 d=1	C=1.0 ε=10.0	C=1.0 ε=0.1 kernel='poly' d=2	alg='ball_tree' k=28 dist='euclidean' w='distance'
3	λ=0 d=1	C=1.0 ε=10.0	C=1.0 ε=1.0 kernel='poly' d=2	alg='kd_tree' k=28 dist='euclidean' w='distance'
5	λ=1.0 d=1	C=0.1 ε=10.0	C=0.1 ε=1.0 kernel='poly' d=2	alg='ball_tree' k=28 dist='euclidean' w='distance'
7	λ=0 d=1	C=0.1 ε=1.0	C=1.0 ε=0.1 kernel='poly' d=2	alg='kd_tree' k=47 dist='manhattan' w='uniform'



(a) Cijela godina



(b) Prva tri mjeseca

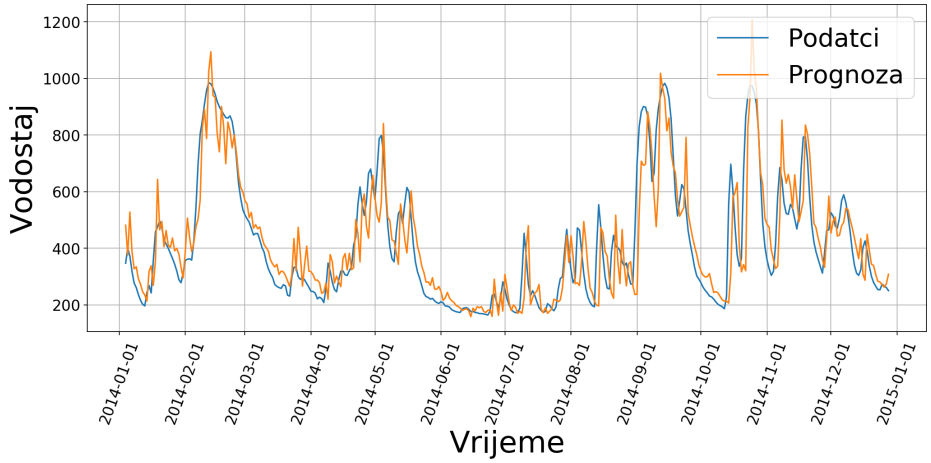
Slika 5.8: Jednokoračna prognoza vodostaja modelom linearne regresije na ispitnom skupu podataka

5.8b koja pokazuje prognozu i podatke iz prva tri mjeseca ispitnog skupa. One pokazuju kako prognostičke pogreške zadovoljavaju poželjno svojstvo ravnanja prema normalnoj razdiobi centriranoj oko 0. Povećanjem koraka prognoze, odnosno otežavanjem regresijskog zadatka, varijanca pogrešaka očekivano raste.

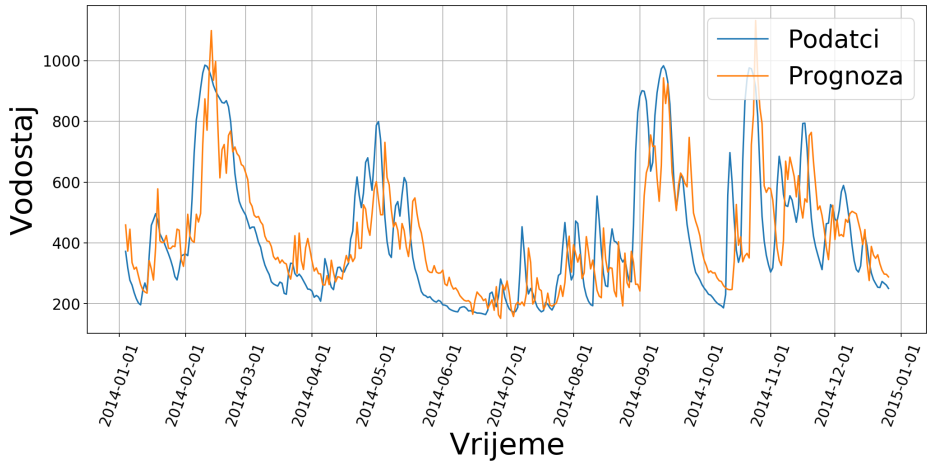
Moguća poboljšanja uključuju razdvajanja hiperparametara vremenskih zaostajanja pojedinih značajki te uklanjanja implicitnog uključivanja svih manjih pomaka. Time bi se omogućilo preciznije ugađanje navedenih hiperparametara, ali bi složenost njihove pretrage značajno porasla. Također, poboljšanje uključuje dodavanje novih značajki iz hidrometeorološke domene koje bi povećale preciznost prognoze.

Tablica 5.5: Usporedba rezultata na ispitnom skupu

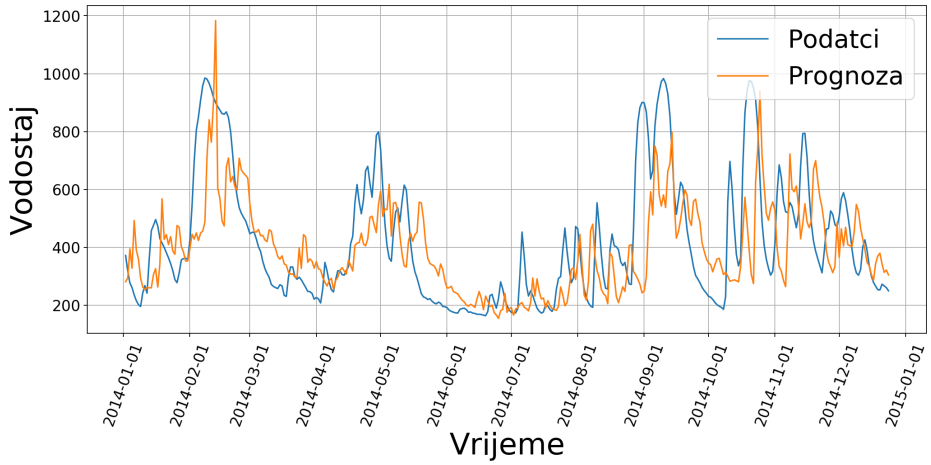
Model	**Metrika**	**Korak prognoze**			
		1	3	5	7
Osnovni model	MSE	3910.43	21104.45	34636.41	44092.08
	MAE	41.67	101.08	132.22	153.23
ARIMA (3,1,2)	MSE	2143.80	18078.88	31035.65	39138.00
	MAE	30.01	102.01	134.82	152.39
VAR (2)	MSE	553.30	3747.47	3879.42	3980.59
	MAE	15.22	41.35	43.09	42.49
Linearni model regresije	MSE	537.77	11552.54	24608.77	34208.81
	MAE	14.33	76.48	117.35	136.38
	Hydro Lag	3	2	3	1
	Meteo Lag	0	4	4	2
LSVR	MSE	667.59	12725.51	28155.65	36568.15
	MAE	15.19	70.29	113.73	133.82
	Hydro Lag	2	5	3	1
	Meteo Lag	0	0	0	0
SVR	MSE	2571.61	16731.72	30982.72	39505.25
	MAE	34.59	91.22	127.15	147.25
	Hydro Lag	1	1	4	2
	Meteo Lag	0	0	0	6
k-NN	MSE	1331.70	13582.86	29699.20	37914.44
	MAE	22.85	77.21	124.60	140.88
	Hydro Lag	2	2	1	5
	Meteo Lag	0	0	0	4



(a) Prognoza tri koraka unaprijed

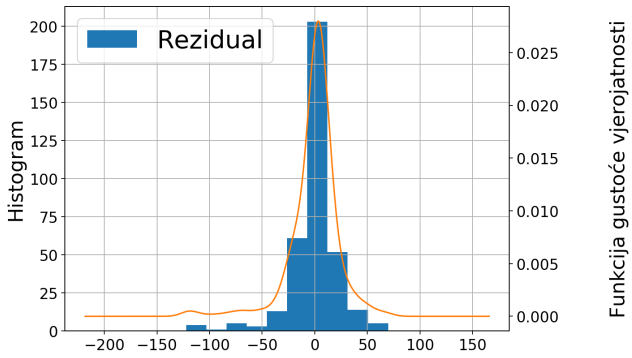


(b) Prognoza pet koraka unaprijed

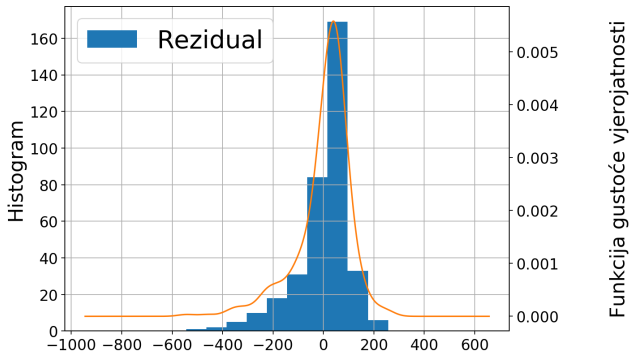


(c) Prognoza sedam koraka unaprijed

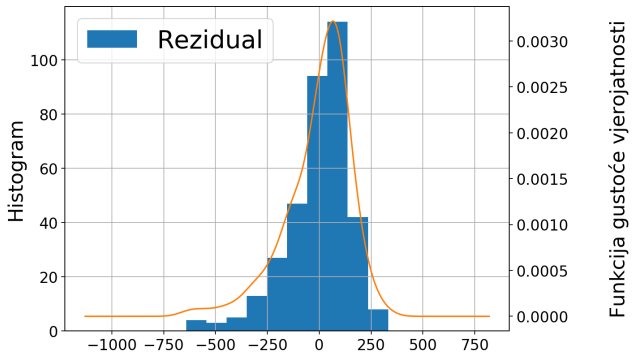
Slika 5.9: Višekoračne prognoza vodostaja modelom linearne regresije na ispitnom skupu podataka



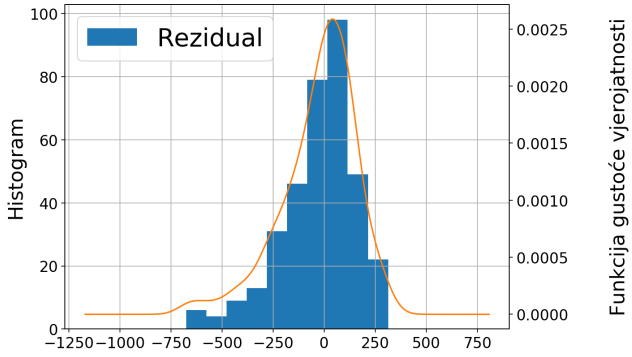
(a) Jednokoračna prognoza



(b) Trokoračna prognoza



(c) Peterokoračna prognoza



(d) Sedmerokoračna prognoza

Slika 5.10: Histogram prognostičkih pogrešaka po koraku prognoze

6. Zaključak

U radu je dan teoretski opis metoda analize i prognoziranja vremenskih serija korištenih za rješavanje problema studijskog slučaja prognoziranja vodostaja rijeke Kupe. Modeli prognoziranja uključivali su stohastičke modele ARIMA i VAR, dok su predstavnike modela strojnog učenja činili linearni model regresije, regresija najbližih susjeda te regresija potpornih vektora.

Stohastički modeli, od kojih je VAR rezultirao boljim prognozama, zadovoljili su osnovni kriterij uspješnosti. Agnostičnost prema podatcima te ugrađeni postupak diferenciranja kojim se ostvaruje stacionarnost predstavljaju temeljne prednosti modela ARIMA, dok multivarijatnost, manji broj hiperparametara i bolji rezultati čine prednosti modela VAR.

Algoritmi strojnog učenja su također zadovoljili osnovni kriterij uspješnosti, dok je uspješnost zadana stohastičkim modelima ovisila o koraku prognoze. Tako je ona zadovoljena samo za jednodnevna predviđanja vodostaja, dok je za ostale korake prognoze model VAR bio uspješniji. Glavnu prednost primjene algoritama strojnog učenja predstavlja prilagodljivost problemu koja se očituje definiranjem vlastitog skupa značajki koji može sadržavati podatke bez vremenske komponente. Međutim, ta prednost predstavlja i njihovu manu jer postupak izgradnje označenog skupa podataka nije jednostavan te unosi dodatne hiperparametre.

Korištena programska knjižnica Scikit-learn nudi jednostavan i učinkovit rad s algoritmima strojnog učenja te ostale pomoćne funkcionalnosti poput skaliranja i pretrage hiperparametara. Dodatno, određeni algoritmi omogućuju paralelizaciju postupka učenja čime se ono značajno ubrzava. Također, funkcionalnosti učenja algoritama nad podatcima vremenskih serija ne zahtijevaju značajan trud čime se zaključuje primjenjivost knjižnice nad problemima prognoziranja vremenskih serija. Obrada i manipulacija podataka ostvarena je programskom knjižnicom Pandas koja nudi funkcionalnosti usko povezane s vremenskim serijama poput diferenciranja te njihovog pomaka kao i manipulaciju nad stupcem vremenskih oznaka koja definira redoslijed podataka vremenske serije.

LITERATURA

Fahad H Al-Qahtani i Sven F Crone. Multivariate k-nearest neighbour regression for time series data—a novel algorithm for forecasting uk electricity demand. U *The 2013 international joint conference on neural networks (IJCNN)*, stranice 1–8. IEEE, 2013.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

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

Stephen Boyd i Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Vladimir Cherkassky i Yunqian Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, 17(1):113–126, 2004.

Thomas M Cover, Peter E Hart, et al. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

CryptoDataDownload. Free CryptoCurrency Time Series Data. `https://www.cryptodatadownload.com/cdd/Gemini_BTCUSD_d.csv`. Datum nastanka: 3.2.2019., datum pristupa: 1.4.2019.

David A Dickey i Wayne A Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74 (366a):427–431, 1979.

David Alan Dickey. Estimation and hypothesis testing in nonstationary time series. 1976.

Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, i Vladimir Vapnik. Support vector regression machines. U *Advances in neural information processing systems*, stranice 155–161, 1997.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

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

W. Karush. Minima of functions of several variables with inequalities as side constraints. Magistarski rad, Dept. of Mathematics, Univ. of Chicago, Chicago, Illinois, 1939.

H. W. Kuhn i A. W. Tucker. Nonlinear programming. U *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, stranice 481–492, Berkeley, Calif., 1951. University of California Press. URL `https://projecteuclid.org/euclid.bsmsp/1200500249`.

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.

Lavan Mahadeva i Paul Robinson. *Unit root testing to help model building*. Centre for Central Banking Studies, Bank of England, 2004.

Wes McKinney. Data structures for statistical computing in python. U Stéfan van der Walt i Jarrod Millman, urednici, *Proceedings of the 9th Python in Science Conference*, stranice 51 – 56, 2010.

James Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446, 1909.

K-R Müller, Alexander J Smola, Gunnar Rätsch, Bernhard Schölkopf, Jens Kohlmorgen, i Vladimir Vapnik. Predicting time series with support vector machines. U *International Conference on Artificial Neural Networks*, stranice 999–1004. Springer, 1997.

Serena Ng i Pierre Perron. Lag length selection and the construction of unit root tests with good size and power. *Econometrica*, 69(6):1519–1554, 2001.

Stephen M Omohundro. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, i E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Nicholas I Sapankevych i Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, 4(2):24–38, 2009.

B Schölkopf, P Bartlett, A Smola, i R Williamson. Support vector regression with automatic accuracy control. U *International Conference on Artificial Neural Networks*, stranice 111–116. Springer, 1998.

Skipper Seabold i Josef Perktold. Statsmodels: Econometric and statistical modeling with python. U *9th Python in Science Conference*, 2010.

Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, stranice 1–48, 1980.

Alex J Smola i Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, i Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Prognoza vremenskih serija korištenjem programske knjižnice Scikit-learn

Sažetak

Buduće ponašanje vremenske serije moguće je predvidjeti prognoziranjem. Prije prognoziranja potrebno je provesti analizu vremenske serije koja uključuje njenu dekompoziciju te provjeru svojstva stacionarnosti koje se ispituje statističkim testovima. Prognoziranje je ostvarivo klasičnim statističkim metodama, kojima pripadaju stohastički modeli prognoziranja vremenskih serija temeljeni na autoregresijskim modelima te modelima pomičnih prosjeka ili nekim od regresijskih algoritama strojnog učenja, čija primjenjivost nije ograničena na područje vremenskih serija. Njima pripadaju linearni modeli regresije, regresija algoritmom najbližih susjeda te regresija potpornih vektora koji su opisani u radu. Efikasne implementacije algoritama strojnog učenja te često korištenih pomoćnih funkcionalnosti dostupne su u programskoj knjižnici otvorenog koda Scikit-learn. Na studijskom slučaju prognoziranja vodostaja rijeke Kupe pokazana je primjenjivost algoritama strojnog učenja za jednodnevne prognoze, dok su stohastički modeli rezultirali boljim višednevnim prognozama.

Ključne riječi: vremenska serija, prognoziranje, stohastički modeli, strojno učenje, Scikit-learn.

Time Series Forecasting with Scikit-learn Programming Library

Abstract

Time series forecasting reduces uncertainty from the future behavior of the underlying process. Before making forecasts, it is necessary to perform time series analysis, which implies decomposition and stationarity analysis. Forecasting can be done with stochastic time series models, including autoregressive models, moving-average models and their combinations, as well as with machine learning regression algorithms, such as linear regression, nearest neighbor regression or support vector regression. The latter allows usage of Scikit-learn, an open source efficient machine learning programming library for data mining and data analysis which implements machine learning algorithms and commonly used machine learning techniques. The case study in water level forecasting showed the applicability of machine learning techniques for one-step forecasts, while stochastic models performed better on multi-step forecasts.

Keywords: Time series, Forecasting, Stochastic time series models, Machine learning, Scikit-learn.