

**SVEUČILIŠTE U RIJECI
POMORSKI FAKULTET**

LINEARNA REGRESIJA I „SVM“ REGRESIJA

SEMINARSKI RAD

Kolegij: Sustavi za podršku odlučivanju

Mentor: Marko Valčić i Ivan Panić

Student: Marko Jelić

Studijski smjer: Elektroničke i informatičke tehnologije u pomorstvu

JMBAG: 0112069816

Rijeka, 2020.

SADRŽAJ

| | |
|--|-----------|
| 1. UVOD | 3 |
| 2. GUI APLIKACIJA | 4 |
| 2.1. LINEARNA REGRESIJA..... | 6 |
| 2.1.1. METODA NAJMANJIH KVADRATA | 7 |
| 2.2. METODA REGRESIJE POTPORNIH VEKTORA | 8 |
| 3. DOBIVENI REZULTATI..... | 9 |
| 4. ZAKLJUČAK | 20 |
| LITERATURA | 21 |
| POPIS SLIKA | 22 |

1. UVOD

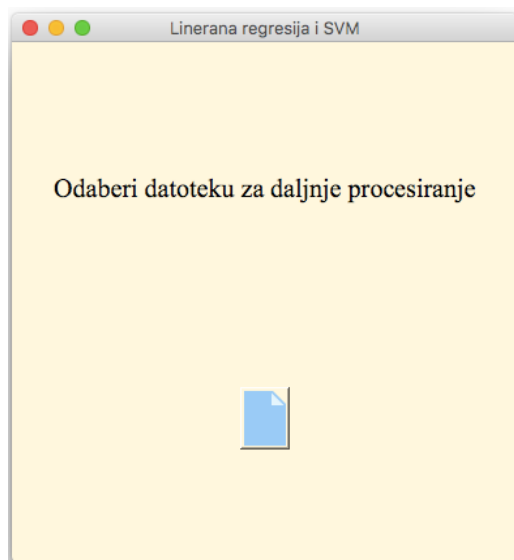
Tema projektnog zadatka bila je izrada sustava za podršku odlučivanju koristeći linearnu regresiju i metodu regresije potpornih vektora (eng. Support Vector Regression). Zadan set podataka se sastoji od broja biciklista na odgovarajućim biciklističkim stazama u Montrealu. Problem ovoga rada je stvoriti algoritam koji će predvidjeti koliko će biciklista biti na jednoj od staza s obzirom koliko ih je na drugim stazama. Najveći problem kod izrade sustava za podršku odlučivanju, na predhodno navedenom setu podataka, javlja se kod odabira podataka, tj. može se uočiti kako određene biciklističke staze dijele veću povezanost od ostalih te se odabirom ispravnih podataka postižu najbolji rezultati.

Projektni zadatak izrađen je u programskom jeziku Python, verzija 3.6.4. te su korištene sljedeće programske biblioteke (u zagradama se nalaze komande za instalaciju odgovarajućih programskih biblioteka):

- sys
- random
- PySide2 (pip3 install PySide2)
- pandas (pip3 install pandas)
- matplotlib (pip3 install matplotlib)
- numpy (pip3 install numpy)
- scipy (pip3 install scipy)
- sklearn (pip3 install sklearn)
- statsmodels (pip3 install statsmodels)

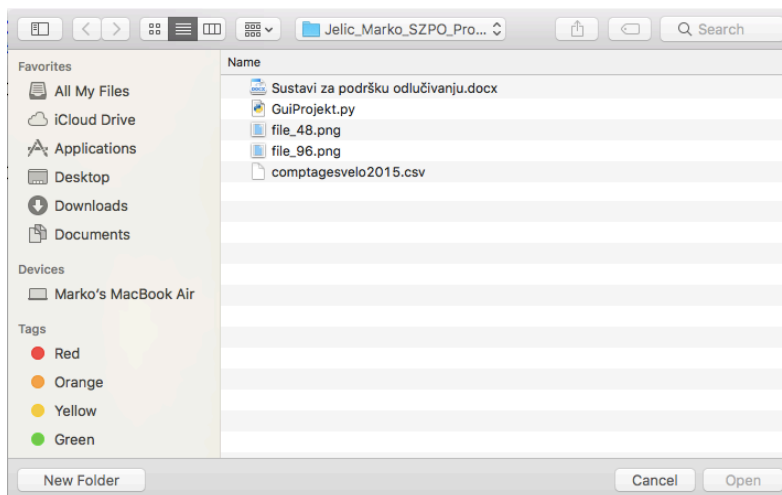
2. GUI APLIKACIJA

Pokretanjem aplikacije javlja se sljedeći prozor (slika 1.) na kojem se nalazi botun pomoću kojega se importiraju željeni podaci u sustav za podršku odlučivanju.



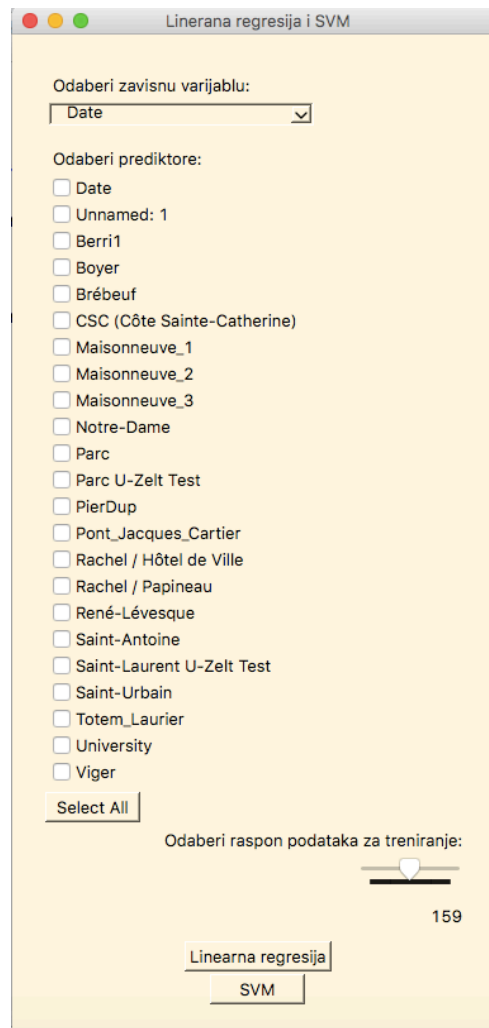
Slika 1 Pokretanje aplikacije

Klikom na predhodno spomenuti botun otvara se tzv. *File Picker* (slika 2) kojim se selektira *csv* datoteka za daljnju obradu.



Slika 2 File picker

Nakon odabira željenih podataka otvara se sljedeći prozor (slika 3) koji sadrži padajući izbornik za odabir zavisne varijable. Padajući izbornik sadrži sva imena stupaca iz csv datoteke koja u sebi ne sadrže NaN vrijednosti. Ispod padajućeg izbornika se nalaze potvrdni okviri sa pripadajućim imenima stupaca iz csv datoteke, no bitno je naznačiti kako je moguće selektirati samo one stupce koji ne sadrže NaN vrijednosti. Botun *Select All* služi za selektiranje svih stupaca koji ne sadrže NaN vrijednosti, ujedno sa ponovom aktivacijom i za deselektiranje svih selektiranih stupaca. Klizeći botun služi za odabir raspona podataka za treniranje. Klikom na botun *Linearna regresija* vrši se linearna regresija na odabranim podacima, dok se klikom na botun *SVM* vrši regresija potpornih vektora.



Slika 3 Odabir podataka za obradu

2.1. LINEARNA REGRESIJA

Za izradu modela koristila se metoda najmanjih kvadrata (eng. *Ordinary Least Squares*) iz programske biblioteke *statsmodel*, programski kod za izradu modela prikazan je na slici 3.

```
def proces_data(self):
    '''
    Procesiraju se selektirani podaci i stvara se model linearne regresije.
    '''
    self.prediktori = []
    self.zavisna_varijabla = self.dropdown_picker.currentText()
    self.training_range = range(0, self.slider.value())
    self.test_range = range(self.slider.value(), len(self.data))

    #Naslovi svih chekiranih stupaca koji ne sadrže NaN vrijednosti
    for i in self.lista_check_box:
        if i.isChecked() and ~self.data[i.text()].isnull().values.any():
            self.prediktori.append(i.text())

    self.test_data_prediktori = self.data[self.prediktori].iloc[self.test_range, :]
    self.test_data_prediktori = sm.add_constant(self.test_data_prediktori)
    self.test_data_zavisnaV = self.data[self.zavisna_varijabla].iloc[self.test_range].to_numpy()

    y = self.data[self.zavisna_varijabla].iloc[self.training_range]
    X = self.data[self.prediktori].iloc[self.training_range, :]
    X = sm.add_constant(X)

    model = sm.OLS(y, X)
    self.est = model.fit()

    self.estimacije = self.est.predict(self.test_data_prediktori).to_numpy()
    e = np.array(self.test_data_zavisnaV[:] - self.estimacije[:])
    self.S = np.array(e[:] * e[:]).sum()

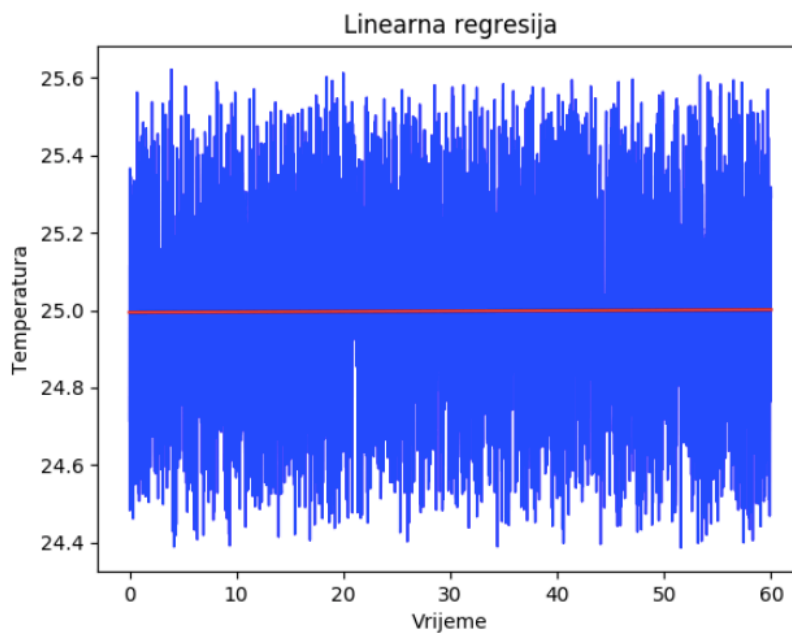
    if ~self.timer_linear_r.isActive():
        self.timer_linear_r.start()
    if ~self.timer_linear_tr.isActive():
        self.timer_linear_tr.start()
```

Slika 4 Linearna regresija

Prediktori, zavisna varijabla te raspon podataka za treniranje modela dohvaćaju se sa stranice za odabir podataka za obradu (slika 3). Nakon što se izradi model na rasponu podataka za treniranje, preostali dio podataka se koristi za estimaciju te se računa suma kvadrata svih rezidualnih odstupanja.

2.1.1. METODA NAJMANJIH KVADRATA

Zadaća linearne regresije je odrediti pravac koji “najbolje” slijedi neke mjerene podatke. Princip rada je najbolje objasniti pomoću jednostavnog primjera. Za primjer uzmimo izmjerene podatke temperature u rasponu jednog sata, gdje se temperatura mjerila svake sekunde. Temperatura će, ovisno o senzoru, varirati oko stvarne vrijednosti temperature te kako bi dobili stvarnu temperaturu možemo koristiti linearnu regresiju koja će povući pravac koji najbolje slijedi mjerene podatke (slika 5).



Slika 5 Primjer linearne regresije

Na slici 5 plavom bojom je prikazana izmjerena temperatura koja oscilira oko stvarne vrijednosti temperature (25°C) dok je crvenom bojom prikazan pravac dobiven linearnom regresijom koji je približan stvarnoj temperaturi. Od nekog pravca $y = ax + b$ izmjerena točka (x_i, y_i) odstupa za e_i te je onda stvarna vrijednost $y_i = ax_i + b + e_i$. Princip rada metode najmanjih kvadrata je odrediti varijablu a i b da budu takve da je suma kvadrata svih odstupanja minimalna.

2.2. METODA REGRESIJE POTPORNIH VEKTORA

Za izradu modela metodom regresije potpornih vektora koristi se programska biblioteka *sklearn* te je programski kod prikazan na slici 6.

```
def proces_svm(self):
    """
    Procesiraju se selektirani podaci i stvara se model sa SVR.
    """
    self.prediktori = []
    self.zavisna_varijabla = self.dropdown_picker.currentText()
    self.training_range = range(0, self.slider.value())
    self.test_range = range(self.slider.value(), len(self.data))

    # Naslovi svih chekiranih stupaca koji ne sadrže NaN vrijednosti
    for i in self.lista_check_box:
        if i.isChecked() and ~self.data[i.text()].isnull().values.any():
            self.prediktori.append(i.text())

    self.test_data_prediktori = self.data[self.prediktori].iloc[self.test_range, :]
    self.test_data_zavisnaV = self.data[self.zavisna_varijabla].iloc[self.test_range].to_numpy()

    y = self.data[self.zavisna_varijabla].iloc[self.training_range].to_numpy().ravel()
    X = self.data[self.prediktori].iloc[self.training_range, :].to_numpy()

    regr = svm.SVR(kernel='linear')
    model = regr.fit(X, y)
    self.R_squared = model.score(X, y)
    self.estimacije = model.predict(self.test_data_prediktori)
    self.model_score = model.score(X, y)
    self.Svm_koeficijetni = model.coef_
    self.Svm_konstanta = model.intercept_

    e = np.array(self.test_data_zavisnaV[:] - self.estimacije[:])
    self.S = np.array(e[:] * e[:]).sum()

    self.svm_text = ''
    for i in range(len(self.prediktori)):
        self.svm_text = self.svm_text + str(self.prediktori[i]) + ': ' + str(self.Svm_koeficijetni[0][i]) + '\n\n'

    self.svm_text = self.svm_text + 'Konstanta: ' + str(self.Svm_konstanta[0])

    if ~self.timer_svm.isActive():
        self.timer_svm.start()
```

Slika 6 Metoda regresije potpornih vektora

Prediktori, zavisna varijabla te raspon podataka za treniranje modela dohvaćaju se sa stranice za odabir podataka za obradu (slika 3). Nakon što se izradi model na rasponu podataka za treniranje, preostali dio podataka se koristi za estimaciju te se računa suma kvadrata svih rezidualnih odstupanja.

3. DOBIVENI REZULTATI

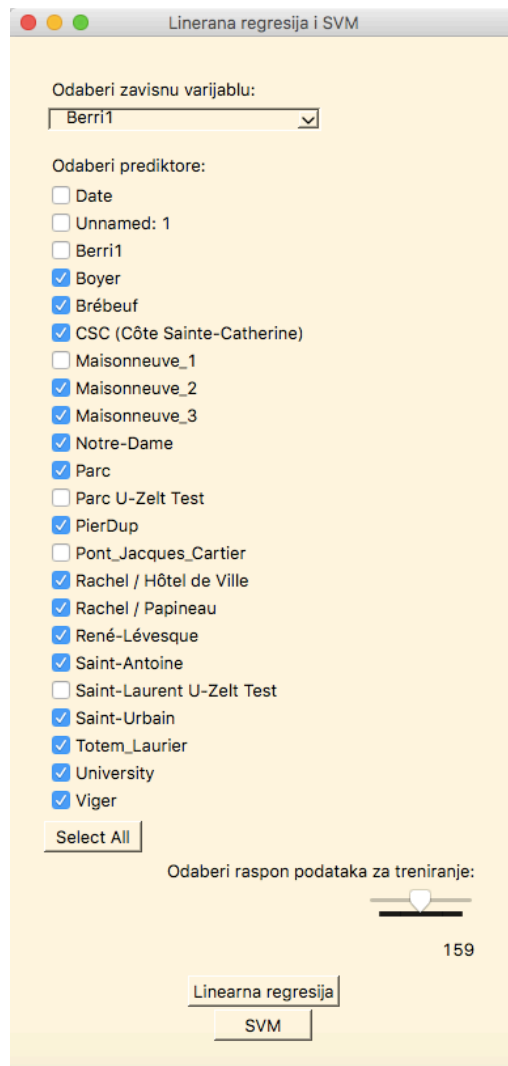
Analizom podataka ustanovilo se da je predprocesiranje podataka nepotrebno. Stupci koji sadrže NaN vrijednosti predstavljaju nevjerodostojne podatke zbog velikog obujma NaN vrijednosti (slika 7). Odnosno, interpolacijom ne bi stvorili vjerodostojne podatke.

```
NaN vrijednosti za 0 stupac: False 0
NaN vrijednosti za 1 stupac: False 0
NaN vrijednosti za 2 stupac: False 0
NaN vrijednosti za 3 stupac: False 0
NaN vrijednosti za 4 stupac: True 257
NaN vrijednosti za 5 stupac: False 0
NaN vrijednosti za 6 stupac: False 0
NaN vrijednosti za 7 stupac: False 0
NaN vrijednosti za 8 stupac: False 0
NaN vrijednosti za 9 stupac: True 267
NaN vrijednosti za 10 stupac: False 0
NaN vrijednosti za 11 stupac: True 110
NaN vrijednosti za 12 stupac: False 0
NaN vrijednosti za 13 stupac: False 0
NaN vrijednosti za 14 stupac: False 0
NaN vrijednosti za 15 stupac: False 0
NaN vrijednosti za 16 stupac: True 269
NaN vrijednosti za 17 stupac: False 0
NaN vrijednosti za 18 stupac: False 0
NaN vrijednosti za 19 stupac: False 0
NaN vrijednosti za 20 stupac: False 0
```

Slika 7 Ispitivanje NaN vrijednosti

Slika 7 prikazuje koliko stupac ima NaN vrijednosti od mogućih 319 podataka koje sadrži.

Kao zavisnu varijablu uzet je stupac pod imenom *Berry1* te kroz testiranja modela s različitim prediktorima, zavisna varijabla ostaje ista. Kao prvi primjer, uzmimo sve moguće prediktore kao što je prikazano na slici 8, a zatim stvorimo model linearnom regresijom.



Slika 8 Odabir prediktora u prvom primjeru

Klikom na botun *Linearna regresija* dobivamo model prikazan na slici 9. U modelu je moguće očitati razne parametre koji prikazuju razinu ispravnosti modela. U prvom je odlomku bitno je istaknuti vrijednosti *R-squared* i *Adj. R-squared*. *R-squared* i *Adj. R-squared* daju informaciju o tome koliko je jaka funkcijska veza između x i y . Što je vrijednost koeficijenta bliža 1, zavisnost je jača. U drugom se odlomku nalaze regresijski koeficijenti koji se mogu pronaći pod stupcem *coef* te je isto vrijedno napomenuti stupac *std err* koji označava standardnu devijaciju. Standardna devijacija može se definirati kao prosječno odstupanje vrijednosti numeričkog obilježja od aritmetičke sredine.

Natrag

OLS Regression Results

```

=====
Dep. Variable:      Berri1  R-squared:      0.999
Model:              OLS    Adj. R-squared:    0.999
Method:             Least Squares  F-statistic: 6798.
Date:               Sat, 27 Jun 2020  Prob (F-statistic): 3.98e-196
Time:               18:32:16  Log-Likelihood: -910.10
No. Observations:   159     AIC:              1854.
Df Residuals:       142     BIC:              1906.
Df Model:            16
Covariance Type:    nonrobust
=====

```

```

=====
              coef  std err      t  P>|t|  [0.025  0.975]
-----
const          18.3695  15.108    1.216  0.226  -11.496  48.235
Boyer          -0.2305   0.096   -2.410  0.017  -0.419  -0.041
Brébeuf         0.4454   0.095    4.697  0.000    0.258  0.633
CSC (Côte Sainte-Catherine)  0.1027   0.071    1.449  0.149  -0.037  0.243
Maisonneuve_2   0.4251   0.069    6.197  0.000    0.290  0.561
Maisonneuve_3  -0.5430   0.182   -2.977  0.003  -0.904  -0.182
Notre-Dame      0.5277   0.214    2.464  0.015    0.104  0.951
Parc           -0.3139   0.072   -4.336  0.000   -0.457  -0.171
PierDup         0.0396   0.026    1.526  0.129  -0.012  0.091
Rachel / Hôtel de Ville -0.0427   0.036   -1.185  0.238  -0.114  0.029
Rachel / Papineau  0.0829   0.049    1.700  0.091  -0.013  0.179
René-Lévesque   0.2227   0.149    1.497  0.137  -0.071  0.517
Saint-Antoine   0.3701   0.241    1.534  0.127  -0.107  0.847
Saint-Urbain   -0.0136   0.077   -0.177  0.860  -0.165  0.138
Totem_Laurier   0.1259   0.053    2.368  0.019    0.021  0.231
University      0.1485   0.115    1.291  0.199  -0.079  0.376
Viger          0.2798   0.248    1.128  0.261  -0.211  0.770
=====

```

```

=====
Omnibus:          15.683  Durbin-Watson:      1.403
Prob(Omnibus):    0.000  Jarque-Bera (JB):    53.636
Skew:             0.085  Prob(JB):             2.26e-12
Kurtosis:         5.840  Cond. No.              1.65e+04
=====

```

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

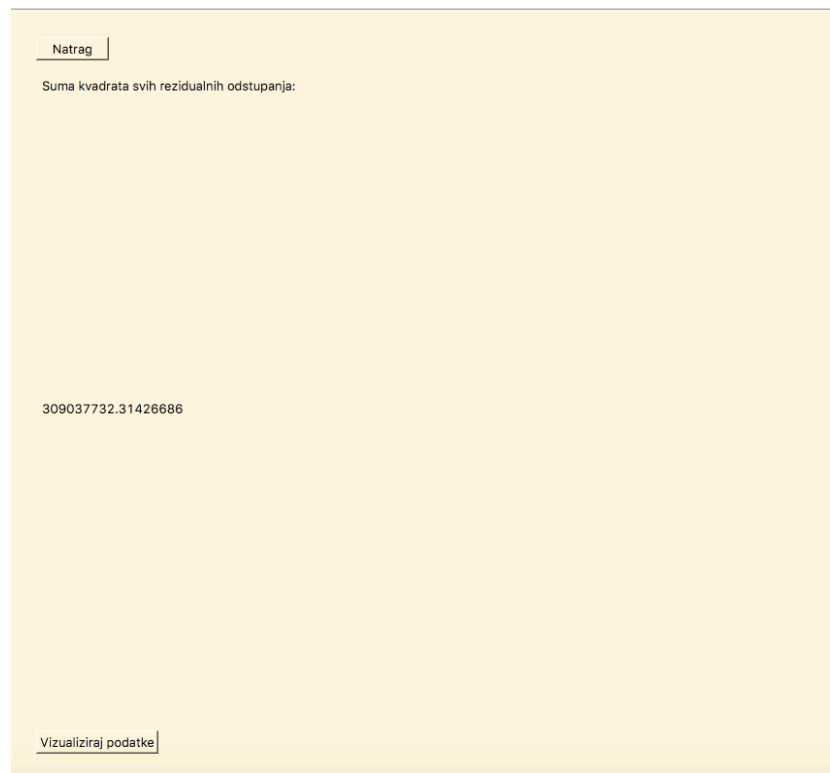
[2] The condition number is large, 1.65e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Treniraj model

Slika 9 Rezultati modela linearne regresije iz prvog primjera

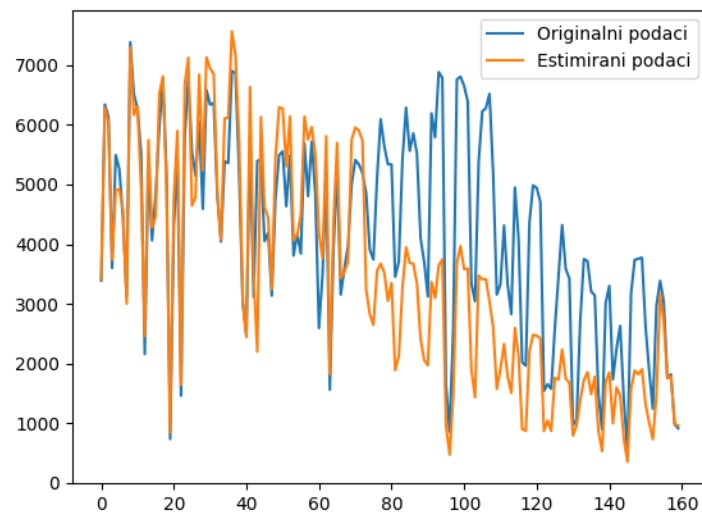
U zadnjem odlomku nalaze se rezultati modela linearne regresije koji su povezani sa podacima prikazanim u prva dva odlomka. *Omnibus* prikazuje iskrivljenost i kurtozu reziduala te želimo da ta vrijednost bude približna nuli. *Prob(Omnibus)* vrši statistički test te prikazuje vjerojatnost da su reziduali normalno distribuirani. Ovu vrijednost želimo da bude što bliža jedinici. *Skew* je mjera simetrije podataka te želimo da vrijednost bude približna nuli. *Kurtosis* je mjera zakrivljenosti podataka te želimo da vrijednost bude približna nuli (velika vrijednost bi značila da nam podaci imaju puno ekstremnih vrijednosti). *Durbin-Watson* testira podatke na homoskedastičnost te se očekuje vrijednost između 1 i 2. *Jarque-Bera(JB)* i *Prob(JB)* prikazuju isto što i *Omnibus* test te služe za provjeru. *Cond. No.* je test za multikolinearnost i poželjno je dobiti relativno malu vrijednost. Analizirajući podatke sa slike 9 možemo automatski ustanoviti kako nam model nije vrlo dobar. Klikom na botun *Treniraj model* prikazuje nam se suma

kvadrata svih rezidualnih odstupanja koja nam služi kao pokazatelj kvalitete estimiranih vrijednosti (slika 10).

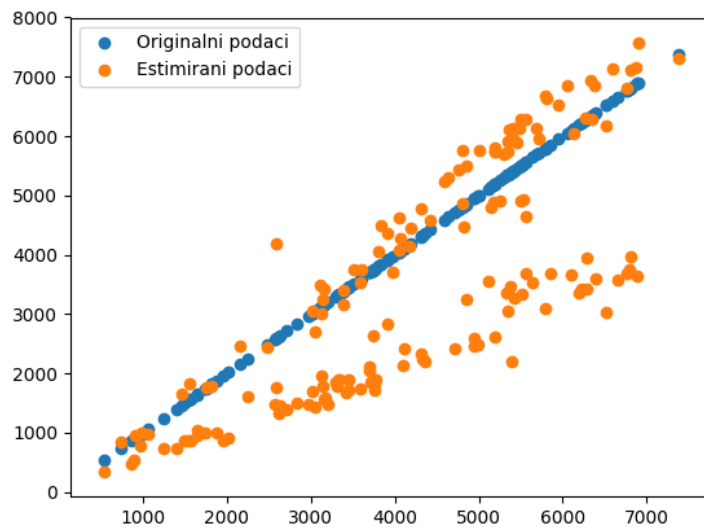


Slika 10 Suma kvadrata svih rezidualnih odstupanja iz prvog primjera

Klikom na botun *Vizualiziraj podatke* prikazuje se graf estimiranih vrijednosti i stvarnih vrijednosti (Slika 11, Slika 12).



Slika 11 Vizualizacija estimiranih i stvarnih vrijednosti prvog primjera



Slika 12 Vizualizacija estimiranih i stvarnih vrijednosti prvog primjera

Iz slike 11 i slike 12 vidljivo je da se radi o lošem modelu koji je dao loše estimacije. U drugom primjeru pažljivo izaberimo prediktore za koje smatramo da će nam pružati najbolje rezultate (slika 13).

Odaberi zavisnu varijablu:

Berri1

Odaberi prediktore:

☐ Date

☐ Unnamed: 1

☐ Berri1

☒ Boyer

☒ Brébeuf

☒ CSC (Côte Sainte-Catherine)

☐ Maisonneuve_1

☐ Maisonneuve_2

☐ Maisonneuve_3

☐ Notre-Dame

☐ Parc

☐ Parc U-Zelt Test

☐ PierDup

☐ Pont_Jacques_Cartier

☒ Rachel / Hôtel de Ville

☐ Rachel / Papineau

☐ René-Lévesque

☐ Saint-Antoine

☐ Saint-Laurent U-Zelt Test

☒ Saint-Urbain

☒ Totem_Laurier

☒ University

☒ Viger

Select All

Odaberi raspon podataka za treniranje:

159

Linearna regresija

SVM

Slika 13 Odabir prediktora u drugom primjeru

Zatim stvorimo model linearne regresije (slika 14).

Natrag

OLS Regression Results

| | | | |
|-------------------|------------------|---------------------|-----------|
| Dep. Variable: | Berri1 | R-squared: | 0.996 |
| Model: | OLS | Adj. R-squared: | 0.996 |
| Method: | Least Squares | F-statistic: | 5329. |
| Date: | Sat, 27 Jun 2020 | Prob (F-statistic): | 5.50e-180 |
| Time: | 21:11:07 | Log-Likelihood: | -988.74 |
| No. Observations: | 159 | AIC: | 1995. |
| Df Residuals: | 150 | BIC: | 2023. |
| Df Model: | 8 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P> t | [0.025 | 0.975] |
|-----------------------------|---------|-------------------|----------|-------|--------|---------|
| const | 92.1025 | 20.818 | 4.424 | 0.000 | 50.967 | 133.238 |
| Boyer | 0.4091 | 0.123 | 3.334 | 0.001 | 0.167 | 0.652 |
| Brébeuf | 0.2840 | 0.117 | 2.437 | 0.016 | 0.054 | 0.514 |
| CSC (Côte Sainte-Catherine) | -0.1432 | 0.087 | -1.655 | 0.100 | -0.314 | 0.028 |
| Rachel / Hôtel de Ville | -0.1696 | 0.033 | -5.218 | 0.000 | -0.234 | -0.105 |
| Saint-Urbain | 0.0466 | 0.111 | 0.420 | 0.675 | -0.173 | 0.266 |
| Totem_Laurier | 0.0681 | 0.063 | 1.087 | 0.279 | -0.056 | 0.192 |
| University | 0.4698 | 0.095 | 4.953 | 0.000 | 0.282 | 0.657 |
| Viger | 1.7961 | 0.355 | 5.063 | 0.000 | 1.095 | 2.497 |
| Omnibus: | 20.565 | Durbin-Watson: | 1.065 | | | |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 43.175 | | | |
| Skew: | 0.562 | Prob(JB): | 4.21e-10 | | | |
| Kurtosis: | 5.292 | Cond. No. | 1.05e+04 | | | |

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 1.05e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Treniraj model

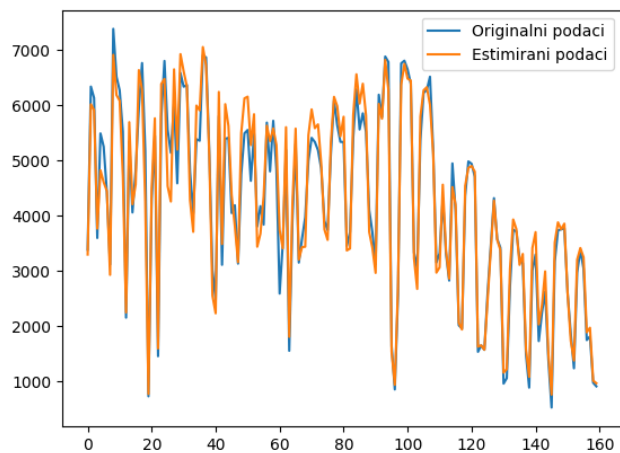
Slika 14 Rezultati modela linearne regresije iz drugog primjera

Analizom dobivenih podataka ne vide se neke značajne promjene no treniranjem modela vidljivo je kako je suma kvadrata svih rezidualnih odstupanja puno manja od prijašnje (slika 15).

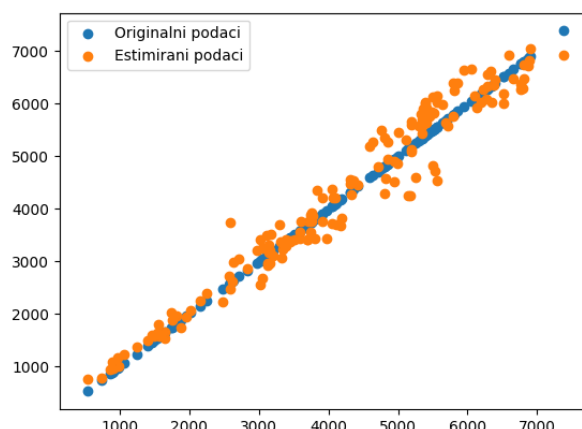


Slika 15 Suma kvadrata svih rezidualnih odstupanja iz drugog primjera

Vizualizacijom podataka je vidljivo kako je model iz drugog primjera puno bolji u odnosu na model iz prvog primjera (slika 16, slika 17).

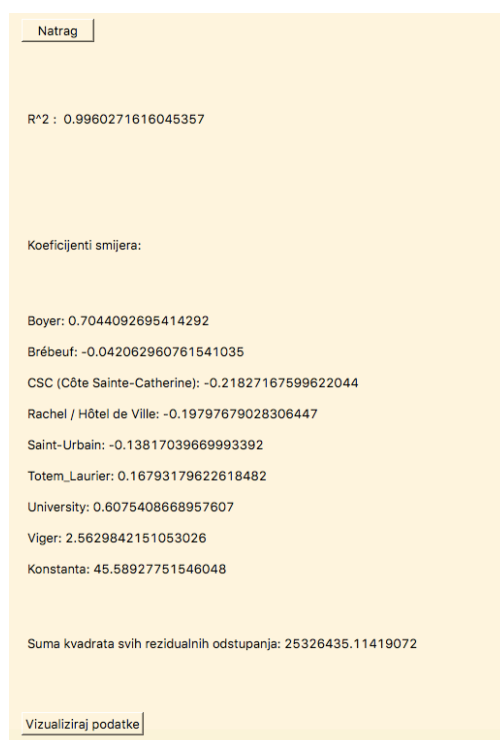


Slika 16 Vizualizacija estimiranih i stvarnih vrijednosti drugog primjera



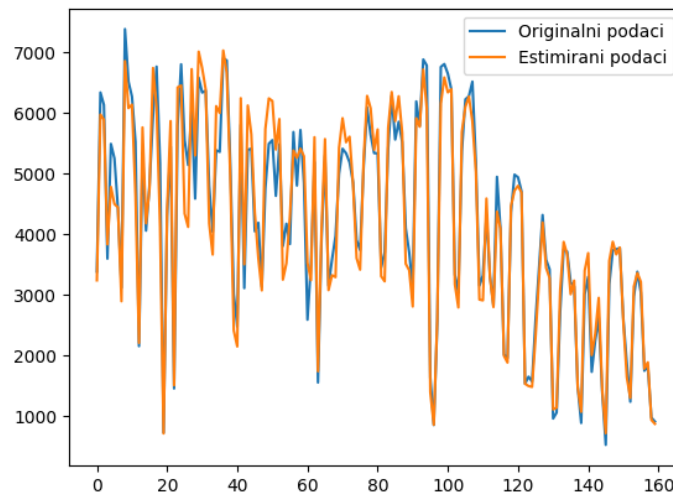
Slika 17 Vizualizacija estimiranih i stvarnih vrijednosti drugog primjera

Za iste prediktore koristit će se metoda potpornih vektora kako bi se ustanovilo koja je od dviju metoda bolja na zadanom setu podataka. Nakon što se selektiraju prediktori kao što je to prikazano na slici 13 klikne se na botun *SVM* te se prikaže sljedeća stranica (slika 18).



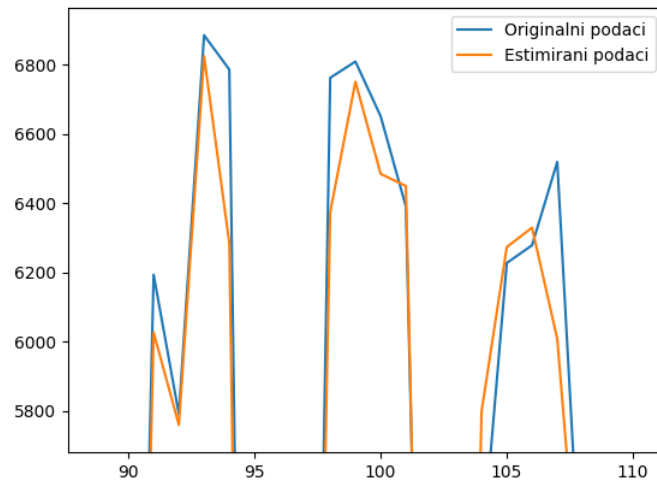
Slika 18 SVR model

Na stranici je prikazan podatak R^2 koji daje informaciju o tome koliko je jaka funkcijska veza između x i y . Što je vrijednost koeficijenta bliža 1, zavisnost je jača. Zatim su prikazani koeficijenti smjera te suma kvadrata svih rezidualnih odstupanja. Usporedbom sume kvadrata svih rezidualnih odstupanja dobivenih metodom linearne regresije i metodom potpornih vektora moguće je ustanoviti kako je model linearne regresije izvršio puno bolju estimaciju za dani set podataka. Ovu tvrdnju možemo i vizualno usporediti (Slika 19).

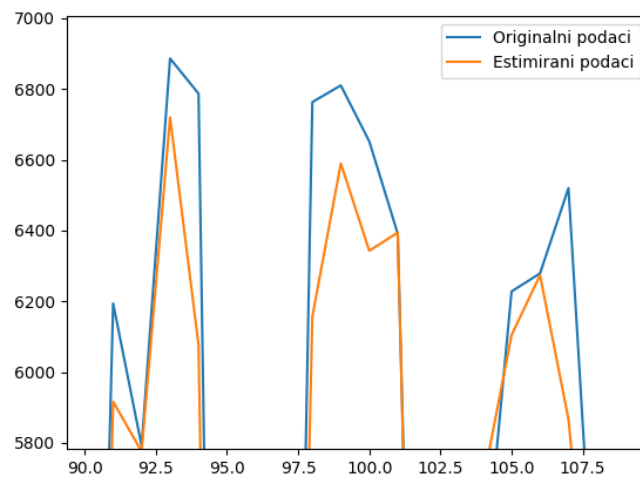


Slika 19 Vizualizacija estimiranih i stvarnih vrijednosti dobivenih koristeći SVR

Na prvi pogled slika 19 i slika 16 izgledaju isto no kada izoštrimo sliku uočavamo vidljivu razliku u estimiranim vrijednostima (Slika 20, slika 21).



Slika 20 Rezultati dobiveni linearnom regresijom



Slika 21 Rezultati dobiveni SVR

4. ZAKLJUČAK

Izradom sustava za podršku odlučivanju koristeći GUI aplikaciju daje korisniku slobodu vlastoručnog istraživanja no sama aplikacija sadrži mnogo nedostataka. Prvenstveno ne sadrži opcije predprocesiranja podataka. U slučaju kada se žele koristiti podaci kojima je potrebno predprocesiranje potrebno je najprvo izvršiti predprocesiranje u zasebnoj aplikaciji ili programu te unjeti “čiste” podatke. Zatim, aplikacija prihvaća samo *csv* datoteke jer su podaci zadani u zadatku bili u datoteci tipa *csv* te se nisu radile daljnje nadogradnje na aplikaciji u tom segmentu. Jedno od većih nedostataka je to što aplikacija unosi podatke tipa *date* kao *string* te ih je moguće selektirati kao prediktore no to će prouzročiti grešku u aplikaciji. U ovom kontekstu stupac koji je sadržavao podatke tipa *date* nije bio relevantan te se zbog toga ovaj problem ostavio u samoj aplikaciji. Uz istaknute nedostatke, sama aplikacija ima značajne prednosti. Jedna od prednosti aplikacije je intuitivna interakcija sa podacima te mjenjanje prediktora i gledanje rezultata u realnom vremenu. Omogućen je uz to i vizualan prikaz podataka što poboljšava razumjevanje.

Testirajući zadane podatke u aplikaciji moguće je zaključiti kako će najbolji model ispasti koristeći se precizno selektiranim podacima. Imati veliki obujam podataka ne znači da će model biti bolji. Puno je bolje izabrati manje podataka koji su povezani s onime što želimo postići. Isto je tako vidljivo kako je za zadani set podataka metoda linearne regresije bolja od metode potpornih vektora.

LITERATURA

1) Internet stranice

- <https://www.statsmodels.org/stable/index.html>
- <https://www.datarobot.com/blog/ordinary-least-squares-in-python/>
- https://link.springer.com/chapter/10.1007/978-1-4302-5990-9_4#:~:text=As%20in%20classification%2C%20support%20vector,in%20real%20Dvalue%20function%20estimation.
- https://hr.wikipedia.org/wiki/Linearna_regresija

POPIS SLIKA

| | |
|---|----|
| Slika 1 Pokretanje aplikacije | 4 |
| Slika 2 File picker | 4 |
| Slika 3 Odabir podataka za obradu | 5 |
| Slika 4 Linearna regresija | 6 |
| Slika 5 Primjer linearne regresije | 7 |
| Slika 6 Metoda regresije potpunih vektora | 8 |
| Slika 7 Ispitivanje NaN vrijednosti | 9 |
| Slika 8 Odabir prediktora u prvom primjeru | 10 |
| Slika 9 Rezultati modela linearne regresije iz prvog primjera | 11 |
| Slika 10 Suma kvadrata svih rezidualnih odstupanja iz prvog primjera | 12 |
| Slika 11 Vizualizacija estimiranih i stvarnih vrijednosti prvog primjera | 13 |
| Slika 12 Vizualizacija estimiranih i stvarnih vrijednosti prvog primjera | 13 |
| Slika 13 Odabir prediktora u drugom primjeru | 14 |
| Slika 14 Rezultati modela linearne regresije iz drugog primjera | 15 |
| Slika 15 Suma kvadrata svih rezidualnih odstupanja iz drugog primjera | 16 |
| Slika 16 Vizualizacija estimiranih i stvarnih vrijednosti drugog primjera | 16 |
| Slika 17 Vizualizacija estimiranih i stvarnih vrijednosti drugog primjera | 17 |
| Slika 18 SVR model | 17 |
| Slika 19 Vizualizacija estimiranih i stvarnih vrijednosti dobivenih koristeći SVR | 18 |
| Slika 20 Rezultati dobiveni linearnom regresijom | 19 |
| Slika 21 Rezultati dobiveni SVR | 19 |