

Zadatak - Predikcija neto nacionalnog prihoda po glavi stanovnika korišćenjem linearne regresije

Uvoz biblioteka

```
from re import X
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

Učitavanje skupa *net_national_incom.csv* preuzet sa *World Bank Open Data*. Okvir podataka ima 266 vrsta i 53 kolone. U prvoj koloni date su države, a u ostalim kolonama njihovi prihodi od 1970. do 2021. godine. Za neke države nisu poznati svi podaci.

```
#Učitavanje skupa
data_frame = pd.read_csv("/content/net_national_incom.csv")
print(data_frame.shape)
```

Definisanje novog okvira podatka - pošto originalni skup nije u željenom formatu. Novi skup se definiše po odabranom nazivu zemlje i on predstavlja okvir njenih podataka. Za razliku od glavnog skupa, novi skup ima dve kolone; prvu za period, a drugu za godišnji dohodak.

Metoda *iterrows()* omogućava ponavljanje svakog reda u skupu. Prilikom svake iteracije proverava se da li je vrednost tražene ćelije jednaka zadatom imenu. Ako jeste - uzimaju se vrednosti tog reda počevši od 1 do kraja. Nulta pozicija se preskače jer sadrži ime države.

```
def Select(name):
    #Prolazimo kroz sve redove glavnog skupa
    for index, row in data_frame.iterrows():
        #print(index, row)
        if row['Country Name'] == name:
            dohodak = row.iloc[1:].values
```

Potom se vrši inicijalizacija praznog niza kao i njegovo popunjavanje unutar *While* petlje vrednostima od 1970 do 2022.

```
#Inicijalizacija prazanog niza
godina = []
i = 1970
while i < 2022:
    godina.append(i) # Dodavanje vrednosti od 1970 do 2022
    i += 1
```

Nakon definisana niza godina i dohodka njihove vrednosti se dodeljuju u kolone novog skupa.

```
#Inicijalizacija praznog skupa podataka
```

```
new_df = pd.DataFrame()
new_df['Godina'] = godina
new_df['Dohodak'] = dohodak
new_df = new_df.dropna()
```

Na kraju se vrši čuvanje novokreiranog skupa bez indexa pod nazivom *ime_drzave_net_national_incom.csv*.

```
#Cuvanje novog skupa
```

```
new_df.to_csv(name.lower() + '_net_national_incom.csv' , index = False)
return(new_df)
```

Kreiramo skup kome dodeljujemo okvir podataka za Srbiju. Skup ima 15 vrsta i dve kolone za godišnji dohodak u intervalu od 2006. do 2020. godine.

```
df = Select('Serbia')
print(df.shape)
print(df)
```

Godina	Dohodak
2006	3644.792633
2007	4708.043974
2008	5751.963631
2009	5069.820372
2010	4662.550758
2011	5413.318382
2012	4494.986443
2013	4992.095299
2014	4752.379044
2015	4585.51385
2016	4680.4751
2017	5026.04952
2018	5868.352728
2019	5939.755291
2020	6361.959439

Ponovo učitavanje skupa - jer je potreban neindeksiran skup za prikaz odnosa dohodka i godine. Zapažena koleracija je nisko pozitivna.

#Ponovono učitavamo skup

```
c = pd.read_csv("/content/serbia_net_national_incom.csv")
```

```
print(c.corr()) # koeficijent korelacije
```

```
print('\n')
```

	Godina	Dohodak
Godina	1.000000	0.583793
Dohodak	0.583793	1.000000

```
x= df.iloc[:,0].values # skup features-a
```

```
y= df.iloc[:,1].values # skup target-a
```

Podela originalnog skupa na obučavajući i testirajući

#Podela glavnog skupa na obucavajuci i testirajuci

```
X1_obucavajuci, X1_testirajuci, Y1_obucavajuci, Y1_testirajuci = train_  
test_split(x, y, test_size=0.20, random_state=1)
```

```
reg = LinearRegression()
```

Vrši se preoblikovanje nizova zbog metoda *LinearRegression()* koja zahteva dvodimenzionalne skupove.

```
print(X1_obucavajuci.shape,X1_testirajuci.shape,Y1_obucavajuci.shape,Y1_  
_testirajuci.shape) #Provera dimenzija
```

#Preoblikovanje skupova u dvodimenzionalne skupove

```
X1_obucavajuci=np.reshape(X1_obucavajuci, (-1,1))
```

```
Y1_obucavajuci=np.reshape(Y1_obucavajuci, (-1,1))
```

```
X1_testirajuci=np.reshape(X1_testirajuci, (-1,1))
```

```
Y1_testirajuci=np.reshape(Y1_testirajuci, (-1,1))
```

```
print(X1_obucavajuci.shape,X1_testirajuci.shape,Y1_obucavajuci.shape,Y1_  
_testirajuci.shape ) #Ponovna provera dimenzija
```

```
Ispis: (12,)      (3,)      (12,)      (3,)
```

```
Ispis: (12, 1)    (3, 1)      (12, 1)    (3, 1)
```

Obučavanje modela

#Obucavanje modela

```
reg.fit(X1_obucavajuci, Y1_obucavajuci)
```

```
print(reg.coef_, reg.intercept_) # y = reg.coef * x reg.intercept
```

```
#Is crtavanje podataka
```

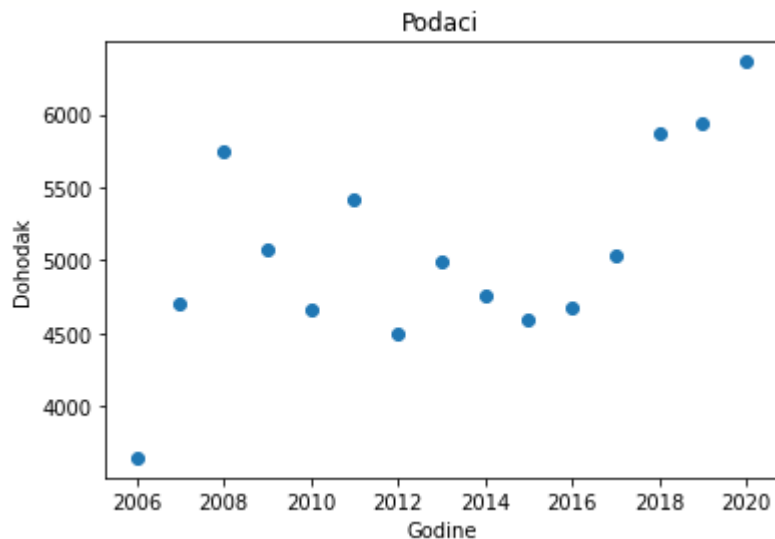
```
plt.scatter(x,y)
```

```
plt.xlabel('Godine')
```

```
plt.ylabel('Dohodak')
```

```
plt.title('Podaci')
```

```
plt.show()
```



```
#Is crtavanje regresione prave nad obucavajucim podacima
```

```
plt.scatter(X1_obucavajuci,Y1_obucavajuci , color = 'orangered')
```

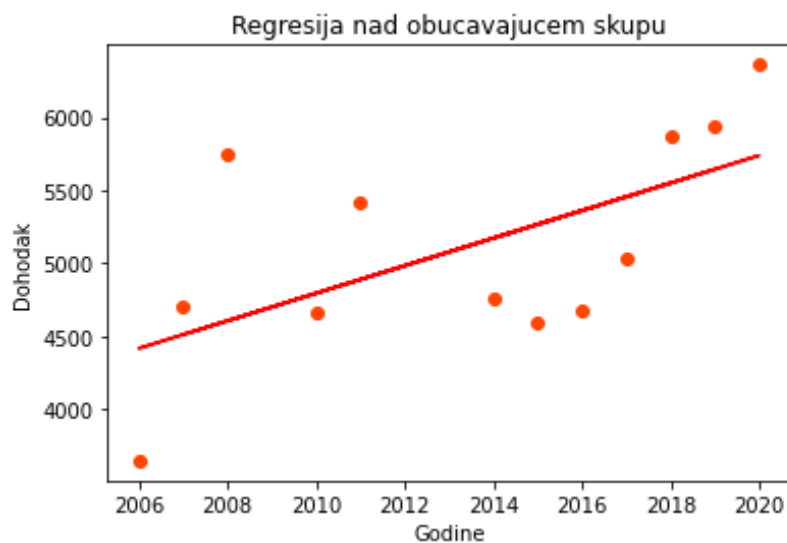
```
plt.plot(X1_obucavajuci, reg.predict(X1_obucavajuci),color='red')
```

```
plt.xlabel('Godine')
```

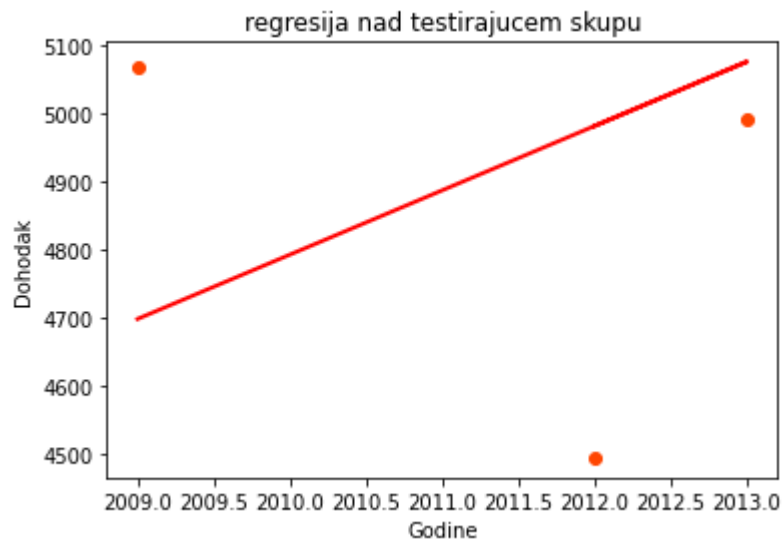
```
plt.ylabel('Dohodak')
```

```
plt.title('Regresija nad obucavajucem skupu')
```

```
plt.show()
```



```
#Iscrtavanje regresione prave nad testirajucim podacima
plt.scatter(X1_testirajuci,Y1_testirajuci,color='orangered')
plt.plot(X1_testirajuci, reg.predict(X1_testirajuci),linewidth=2, color
='red')
plt.xlabel('Godine')
plt.ylabel('Dohodak')
plt.title('regresija nad testirajucem skupu')
plt.show()
```



```
#Iscrtavanje grafikona neto nacionalnog dohodka po godinama
plt.plot(x,y, marker='.', mfc='black')
plt.xlabel('Godine')
plt.ylabel('Dohodak')
plt.title('neto nacionalni dohodak po glavi stanovnika')
plt.show()
```



Evaluacija modela, procena greške i koeficijenta poklapanja između pravih targeta i target vrednosti koju je model procenio na testirajućem skupu

```
Y_predikcija = reg.predict(X1_testirajuci) # Predikcija nad testirajuci
m podacima
print(metrics.mean_squared_error(Y1_testirajuci/max(Y1_testirajuci),Y_p
redikcija/max(Y_predikcija))) # Proracun error-a
print(metrics.r2_score(Y1_testirajuci/max(Y1_testirajuci),Y_predikcija/
max(Y_predikcija))) # Proracun koeficijenta poklapanja

0.004920910782673373
-0.950578720677961
```

Kalkulacija godišnjeg rasta

```
#Funkcija ispisuje procenu godisnjeg rasta neto nacionalnog dohodka po
glavi stanovnika
def Calc_Rate():
    lastY = y[len(y) - 1] # Uzima poslednji segment skupa
    growth = reg.coef_
    rate = growth / lastY * 100
    print('Prema obradjenim podacima godisnji rast neto nacionalnog dohod
aka je %.0fEUR %.1f%% \n' % (growth, rate))

Calc_Rate()
```

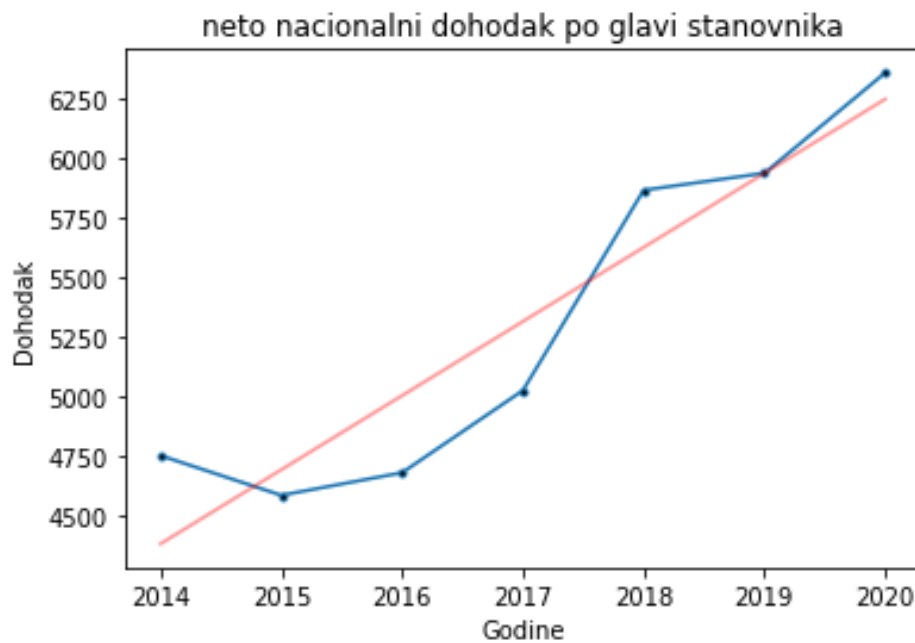
*Prema obradjenim podacima godisnji rast neto nacionalnog
dohodaka je 95EUR 1.5%*

Smanjeni skup se pokazao kao precizniji u predikciji dohodka.
Novi skup koristi podatke od 2014. do 2020. godine.

```
#Redukcija skupa
x= df.iloc[8:,0].values # skup features-a
y= df.iloc[8:,1].values # skup target-a

#Ponovno treniranje modela
reg = LinearRegression()
reg.fit(np.reshape(x, (-1,1)), np.reshape(y, (-1,1)))
```

```
#Is crtavanje novog grafikona neto nacionalnog dohodka sa regresijom
plt.plot(x,y, marker='.', mfc='black')
plt.plot(x, reg.predict(np.reshape(x, (-1,1))),color='red', alpha=0.4)
plt.xlabel('Godine')
plt.ylabel('Dohodak')
plt.title('neto nacionalni dohodak po glavi stanovnika')
plt.show()
```



	Godina	Dohodak
8	2014	4752.379044
9	2015	4585.51385
10	2016	4680.4751
11	2017	5026.04952
12	2018	5868.352728
13	2019	5939.755291
14	2020	6361.959439

Kreiranje novog skupa za **predvidane neto nacionalnog dohodka** po glavi stanovnika u narednih 8 godina. Skup se popunjava podacima za: godinu, dohodak, mesečni dohodak u dinarima i mesečni dohodak u evrima. Zatim se čuva u tekstualnoj *datoteci* *predction_net_national_incom.csv*.

```
#Predikcija neto nacionalnog dohodka u narednih 8 godina
godina = []
i = 2022
while i <= 2030:
    godina.append(i) # Dodavanje vrednosti u intervalu od 2022 do 2030
    i += 1

#Inicijalizacija praznoog skupa podataka
predction_df = pd.DataFrame()
predction_df['Godina'] = godina

godina = np.reshape(godina, (-1,1)) # Preoblikovanje u 2D skup
dohodak = reg.predict(godina) # Predikcija dohodka za godine[]
predction_df['Dohodak'] = dohodak # Definisanje kolone 'Dohodak'

plata_EUR = dohodak / 12
predction_df['Msecni Dohodak EUR'] = plata_EUR

plata_DIN = plata_EUR * 117.55
predction_df['Msecni Dohodak DIN'] = plata_DIN

#Pretvaranje okvira podataka u tekstualnu datoteku
predction_df.to_csv('predction_net_national_incom.csv' , index = False)

print(predction_df)
```

Godina	Dohodak	Msecni Dohodak EUR	Msecni Dohodak DIN
2022	6874.408870107145	572.8674058422621	67340.5635567579
2023	7186.019644928514	598.8349704107095	70393.0507717789
2024	7497.630419749999	624.8025349791666	73445.53798680104
2025	7809.241194571368	650.770099547614	76498.02520182202
2026	8120.851969392854	676.7376641160712	79550.51241684417
2027	8432.462744214223	702.7052286845186	82602.99963186515
2028	8744.073519035708	728.6727932529757	85655.48684688729
2029	9055.684293857077	754.6403578214231	88707.97406190829
2030	9367.295068678563	780.6079223898802	91760.46127693041

Kreiranje **funkcije za poredjenje država** koja ispisuje dohodak tangetirane države, kao i godinu kada odabrana država dostiže taj dohodak.

```
#Funkcija kao argument uzima ime drzave za poredjenje
def Compare(target):
    country_target_df = Select(target)
    #Uzima poslednji segment skupa
    target_dohodak = country_target_df.iloc[-1 , -1]

    god_pred = (np.reshape(target_dohodak, (-1,1)) -
reg.intercept_) / reg.coef_

    print(target + ' net nacional incom is %.0f EUR in 2020 year' % target_dohodak)
    print('Srbija dostize dohodak od %.0f EUR %.0f. godine \n' % (target_dohodak, god_pred))
```

```
Compare("Madagascar")
Compare("Japan")
Compare("Brazil")
Compare("Turkey")
Compare("Costa Rica")
Compare("Kazakhstan")
```

*Madagascar net nacional incom is 388 EUR in 2020 year
Srbija dostize dohodak od 388 EUR 2001. godine*

*Japan net nacional incom is 31279 EUR in 2020 year
Srbija dostize dohodak od 31279 EUR 2100. godine*

*Brazil net nacional incom is 5123 EUR in 2020 year
Srbija dostize dohodak od 5123 EUR 2016. godine*

*Turkey net nacional incom is 7137 EUR in 2020 year
Srbija dostize dohodak od 7137 EUR 2023. godine*

*Costa Rica net nacional incom is 10700 EUR in 2020 year
Srbija dostize dohodak od 10700 EUR 2034. godine*

*Kazakhstan net nacional incom is 6634 EUR in 2020 year
Srbija dostize dohodak od 6634 EUR 2021. Godine*

```
#Kalkulacija rasta nad novim podacima
Calc_Rate()
```

Prema obradjenim podacima godišnji rast neto nacionalnog dohodaka je 95EUR 1.5%

IZVORI

Podaci preuzeti sa:

<https://data.worldbank.org/indicator/NY.ADJ.NNTY.PC.CD?locations=RS>