projekt-uupz

January 26, 2024

1 UVOD U OBRADU PODATKOVNE ZNANOSTI

1.1 Projekt

1.2 Analiza plaća u području podatkovne znanosti

Studentice: Laura Boras i Sunčica Prgomet

#1. Učitavanje podataka

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[ ]:
```
#Pristupanje mapi u kojoj su nam spremljeni podaci
%cd /content/drive/MyDrive/UUPZ/Projekt
```

```
/content/drive/MyDrive/UUPZ/Projekt
```

[ ]:
```
#Učitavanje potrebnih paketa:

import pandas as pd                 #služi za rad s tablicama
import matplotlib.pyplot as plt     #služi za vizualizaciju podataka
import numpy as np                  #služi za rad s višedimenzionalnim nizovima␣
 ↪i matricama
import seaborn as sns               #služi za vizualizaciju
```

[ ]:
```
podaci=pd.read_csv("ds_salaries.csv")    #učitavamo podatke u varijablu "podaci"
```

[ ]:
```
podaci.shape                             #ispisujemo dimenzije tablice
```

[ ]:
```
(3755, 11)
```

[ ]:
```
podaci.head()              #ispisujemo prvih 5 redova tablice
```

[ ]:
```
   work_year experience_level employment_type                job_title  \
0       2023               SE              FT  Principal Data Scientist
```

```
1       2023               MI              CT               ML Engineer
2       2023               MI              CT               ML Engineer
3       2023               SE              FT            Data Scientist
4       2023               SE              FT            Data Scientist

   salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0   80000             EUR          85847                 ES           100
1   30000             USD          30000                 US           100
2   25500             USD          25500                 US           100
3  175000             USD         175000                 CA           100
4  120000             USD         120000                 CA           100

  company_location company_size
0               ES            L
1               US            S
2               US            S
3               CA            M
4               CA            M
```

[ ]:
```
podaci.tail()             #ispisujemo posljednjih 5 redova tablice
```

[ ]:
```
      work_year experience_level employment_type                job_title  \
3750       2020               SE              FT            Data Scientist
3751       2021               MI              FT  Principal Data Scientist
3752       2020               EN              FT            Data Scientist
3753       2020               EN              CT     Business Data Analyst
3754       2021               SE              FT      Data Science Manager

       salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
3750   412000             USD         412000                 US           100
3751   151000             USD         151000                 US           100
3752   105000             USD         105000                 US           100
3753   100000             USD         100000                 US           100
3754  7000000             INR          94665                 IN            50

     company_location company_size
3750               US            L
3751               US            L
3752               US            S
3753               US            L
3754               IN            L
```

#2. Eksplorativna analiza podataka

[ ]:
```
podaci.info()      #ispis informacija o stupcima
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3755 entries, 0 to 3754
```

```
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   work_year           3755 non-null   int64
 1   experience_level    3755 non-null   object
 2   employment_type     3755 non-null   object
 3   job_title           3755 non-null   object
 4   salary              3755 non-null   int64
 5   salary_currency     3755 non-null   object
 6   salary_in_usd       3755 non-null   int64
 7   employee_residence  3755 non-null   object
 8   remote_ratio        3755 non-null   int64
 9   company_location    3755 non-null   object
 10  company_size        3755 non-null   object
dtypes: int64(4), object(7)
memory usage: 322.8+ KB
```

[ ]:
```
podaci.describe()        #ispis statističkih informacija po numeričkim stupcima
```

[ ]:

	work_year	salary	salary_in_usd	remote_ratio
count	3755.000000	3.755000e+03	3755.000000	3755.000000
mean	2022.373635	1.906956e+05	137570.389880	46.271638
std	0.691448	6.716765e+05	63055.625278	48.589050
min	2020.000000	6.000000e+03	5132.000000	0.000000
25%	2022.000000	1.000000e+05	95000.000000	0.000000
50%	2022.000000	1.380000e+05	135000.000000	0.000000
75%	2023.000000	1.800000e+05	175000.000000	100.000000
max	2023.000000	3.040000e+07	450000.000000	100.000000

[ ]:
```
podaci.isna().sum()       #ispis količine Nan vrijednosti po stupcima
```

[ ]:
```
work_year             0
experience_level      0
employment_type       0
job_title             0
salary                0
salary_currency       0
salary_in_usd         0
employee_residence    0
remote_ratio          0
company_location      0
company_size          0
dtype: int64
```

[ ]:
```
#Provjera dupliciranih podataka
podaci.duplicated().sum()
```

[ ]:
```
1171
```

[ ]:
```
#Izbacivanje duplikata
podaci.drop_duplicates(inplace=True)
```

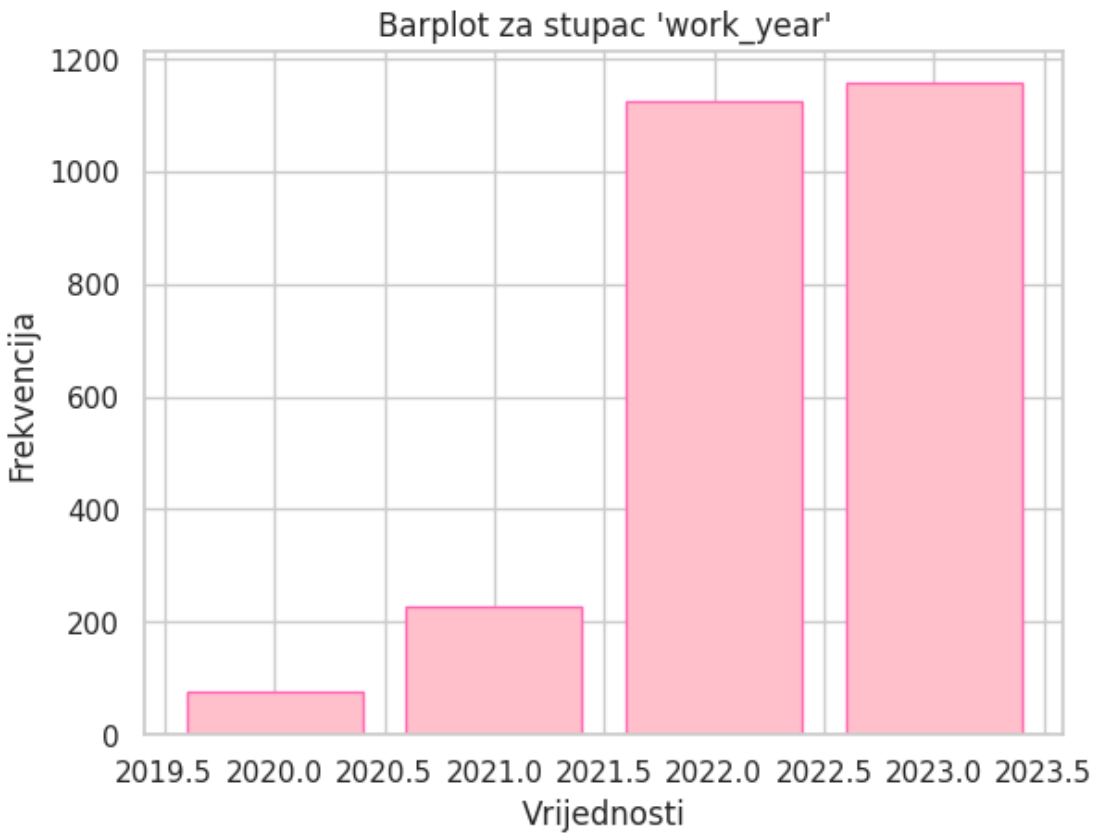
##Distribucije varijabli

Stupac work_year

[ ]:
```
#Ispis jedinstvenih vrijednosti u stupcu work_year te njihovih frekvencija
podaci.work_year.value_counts()
```

[ ]:
```
2023    1156
2022    1125
2021     228
2020      75
Name: work_year, dtype: int64
```

[ ]:
```
#Barplot za stupac "work_year"

# Stvaranje barplota
plt.bar([2023, 2022, 2021, 2020], [1156, 1125, 228, 75], color='pink',␣
 ↪edgecolor="hotpink")

# Dodatne postavke grafikona
plt.title("Barplot za stupac 'work_year'")
plt.xlabel('Vrijednosti')
plt.ylabel('Frekvencija')

# Prikazivanje barplota
plt.show()
```



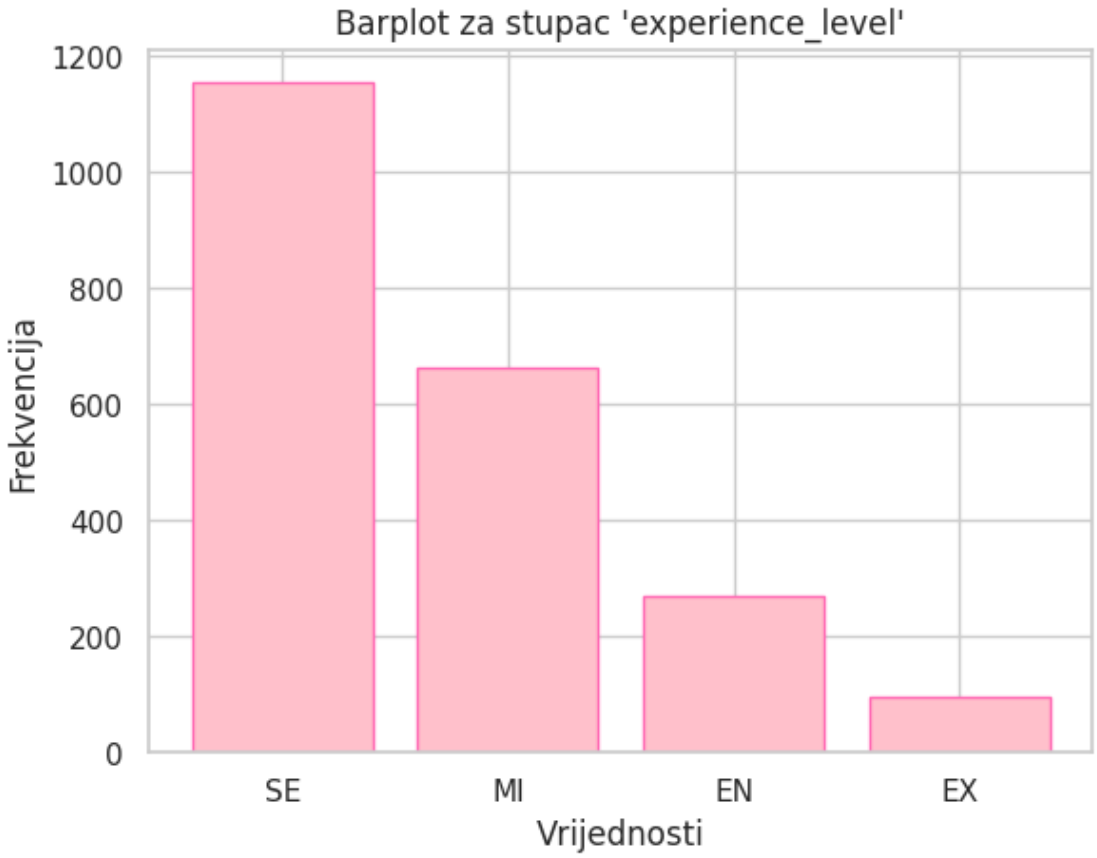
Stupac experience_level

[ ]:
```
#Ispis jedinstvenih vrijednosti u stupcu experience_level te njihovih␣
 ↪frekvencija
podaci.experience_level.value_counts()
```

[ ]:
```
SE    1554
MI     664
EN     270
EX      96
Name: experience_level, dtype: int64
```

[ ]:
```
#Barplot za stupac "experience_level"

# Stvaranje barplota
plt.bar(["SE","MI","EN","EX"], [1154, 664, 270, 96], color='pink',␣
 ↪edgecolor="hotpink")

# Dodatne postavke grafikona
plt.title("Barplot za stupac 'experience_level'")
```

```
plt.xlabel('Vrijednosti')
plt.ylabel('Frekvencija')

# Prikazivanje barplota
plt.show()
```



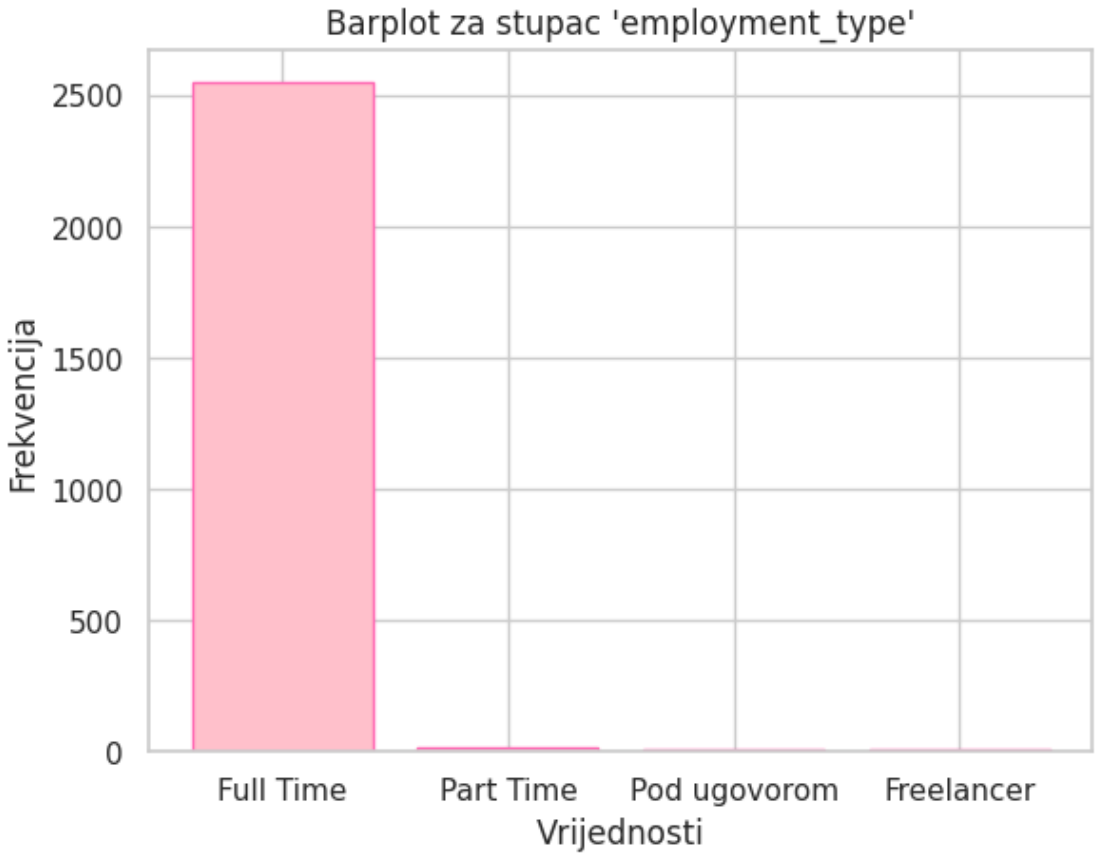
Stupac employment_type

[ ]:
```
#Ispis jedinstvenih vrijednosti u stupcu employement_type te njihovih␣
↪frekvencija
podaci.employment_type.value_counts()
```

[ ]:
```
FT    2547
PT      17
CT      10
FL      10
Name: employment_type, dtype: int64
```

[ ]:
```
#Barplot za stupac "employment_type"
```

```
# Stvaranje barplota
plt.bar(["Full Time","Part Time","Pod ugovorom","Freelancer"], [2547, 17, 10,␣
↪10], color='pink', edgecolor="hotpink")

# Dodatne postavke grafikona
plt.title("Barplot za stupac 'employment_type'")
plt.xlabel('Vrijednosti')
plt.ylabel('Frekvencija')

# Prikazivanje barplota
plt.show()
```



Stupac job_title

[ ]:
```
#Ispis jedinstvenih vrijednosti u stupcu job_title te njihovih frekvencija
podaci.job_title.value_counts()
```

[ ]:
```
Data Engineer              598
Data Scientist             538
Data Analyst               396
```

```
Machine Learning Engineer    206
Analytics Engineer            91
                             ...
Compliance Data Analyst        1
Deep Learning Researcher       1
Staff Data Analyst             1
Data DevOps Engineer           1
Finance Data Analyst           1
Name: job_title, Length: 93, dtype: int64
```

Stupac salary

[ ]:
```
#Ispis jedinstvenih vrijednosti u stupcu salary te njihovih frekvencija
podaci.salary.value_counts()
```

[ ]:
```
100000    70
120000    59
150000    58
200000    48
80000     46
          ..
126277     1
153088     1
323300     1
200160     1
412000     1
Name: salary, Length: 815, dtype: int64
```

Stupac salary_currency

[ ]:
```
#Ispis jedinstvenih vrijednosti u stupcu salary_currency te njihovih frekvencija
podaci.salary_currency.value_counts()
```

[ ]:
```
USD    2107
EUR     200
GBP     144
INR      59
CAD      25
AUD       9
SGD       6
BRL       6
PLN       5
CHF       4
HUF       3
DKK       3
JPY       3
TRY       3
THB       2
```

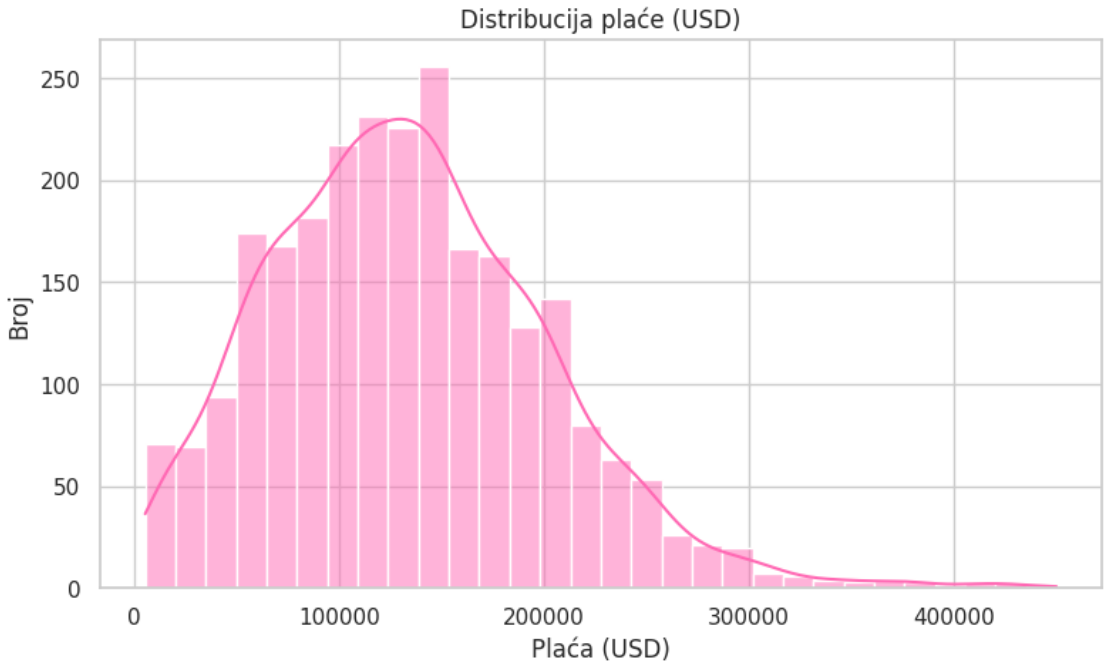
```
ILS       1
HKD       1
CZK       1
MXN       1
CLP       1
Name: salary_currency, dtype: int64
```

Stupac salary_in_usd

[ ]:
```
#Ispis jedinstvenih vrijednosti u stupcu salary_in_usd te njihovih frekvencija
podaci.salary_in_usd.value_counts()
```

[ ]:
```
100000    58
150000    56
120000    51
200000    47
130000    39
          ..
314100     1
195800     1
262500     1
209450     1
94665      1
Name: salary_in_usd, Length: 1035, dtype: int64
```

[ ]:
```
#Postavljanje stila
sns.set_style("whitegrid")

#Postavljanje dimenzije
plt.figure(figsize=(20, 5))

#Crtanje distribucije od salary_in_usd
plt.subplot(1, 2, 1)
sns.histplot(podaci['salary_in_usd'], kde=True, bins=30, color='hotpink')
plt.title('Distribucija plaće (USD)')
plt.xlabel('Plaća (USD)')
plt.ylabel('Broj')

plt.show()
```



Većina plaća grupirana je oko nižeg kraja, što ukazuje da većina stručnjaka za znanost podataka u ovom skupu podataka zarađuje manje od 200.000 dolara. Postoji nekoliko izuzetaka na višem kraju, što bi moglo predstavljati seniorske uloge ili visoko specijalizirane uloge u području znanosti podataka.

Stupac employee_residence

```
[ ]: #Ispis jedinstvenih vrijednosti u stupcu employee_residence te njihovih␣
      ↪frekvencija
     podaci.employee_residence.value_counts()
```

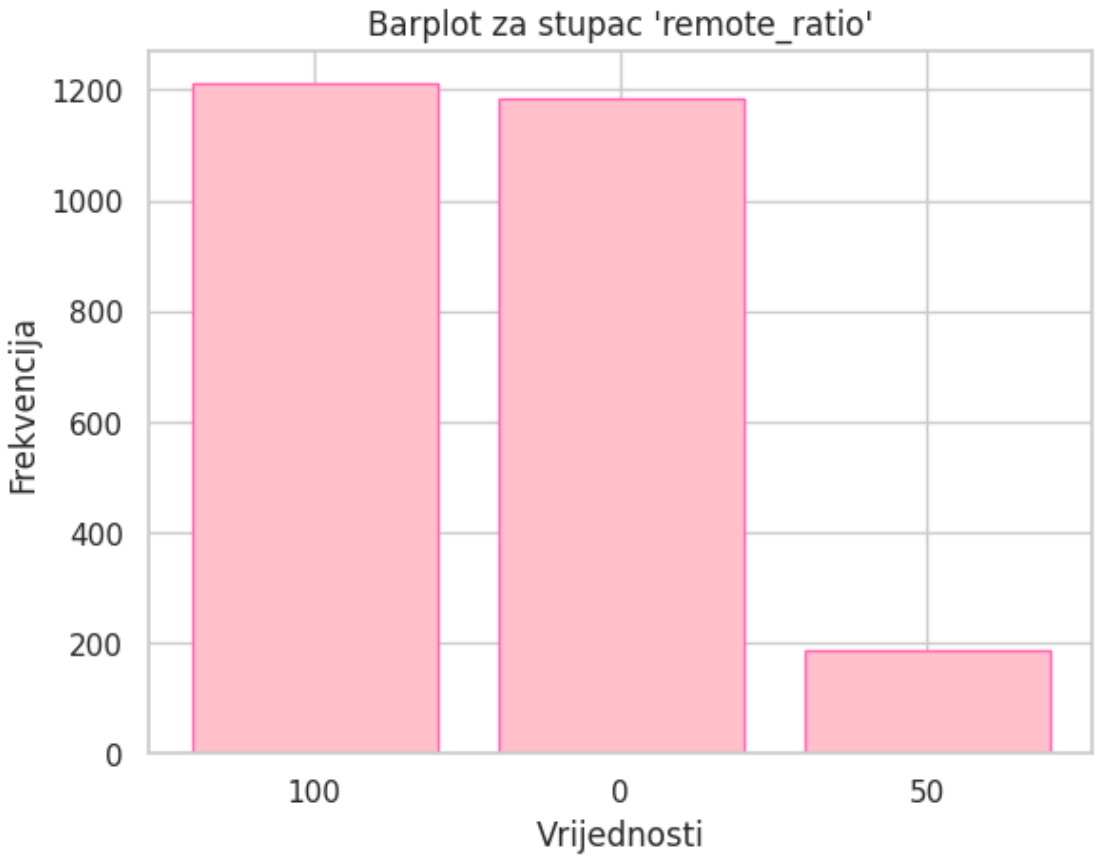
```
[ ]: US    1893
     GB     150
     CA      81
     IN      70
     ES      47
           …
     BA       1
     AM       1
     CY       1
     KW       1
     MT       1
     Name: employee_residence, Length: 78, dtype: int64
```

Stupac remote_ratio

[ ]:
```
#Ispis jedinstvenih vrijednosti u stupcu remote_ratio te njihovih frekvencija
podaci.remote_ratio.value_counts()
```

[ ]:
```
100    1211
0      1186
50      187
Name: remote_ratio, dtype: int64
```

[ ]:
```
#Barplot za stupac "remote_ratio"

# Stvaranje barplota
plt.bar(["100","0","50"], [1211, 1186, 187], color='pink', edgecolor="hotpink")

# Dodatne postavke grafikona
plt.title("Barplot za stupac 'remote_ratio'")
plt.xlabel('Vrijednosti')
plt.ylabel('Frekvencija')

# Prikazivanje barplota
plt.show()
```



Stupac company_location

```
[ ]: #Ispis jedinstvenih vrijednosti u stupcu company_location te njihovih␣
      ↪frekvencija
     podaci.company_location.value_counts()
```

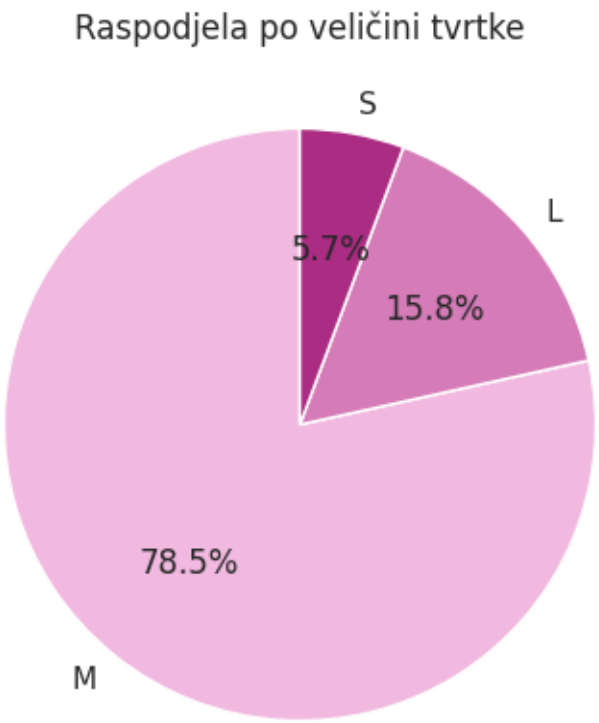
```
[ ]: US    1929
     GB     155
     CA      83
     IN      57
     DE      55
           …
     MK       1
     BS       1
     IR       1
     CR       1
     MT       1
     Name: company_location, Length: 72, dtype: int64
```

Stupac company_size

```
[ ]: #Ispis jedinstvenih vrijednosti u stupcu company_size te njihovih frekvencija
     counts=podaci.company_size.value_counts()
     counts
```

```
[ ]: M    2028
     L     409
     S     147
     Name: company_size, dtype: int64
```

```
[ ]: # Stvaranje pie dijagrama
     plt.figure(figsize=(5, 5))
     plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=90,␣
      ↪colors=['#f1b9df','#d47bb8','#ab2c83'])

     # Dodatne postavke dijagrama
     plt.title('Raspodjela po veličini tvrtke')
     plt.show()
```



##Pročišćavanje podataka

[ ]:
```
podaci
```

[ ]:
```
      work_year experience_level employment_type                 job_title  \
0          2023               SE              FT  Principal Data Scientist
1          2023               MI              CT               ML Engineer
2          2023               MI              CT               ML Engineer
3          2023               SE              FT            Data Scientist
4          2023               SE              FT            Data Scientist
…           …                …               …                         …
3750       2020               SE              FT            Data Scientist
3751       2021               MI              FT  Principal Data Scientist
3752       2020               EN              FT            Data Scientist
3753       2020               EN              CT     Business Data Analyst
3754       2021               SE              FT      Data Science Manager

       salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0       80000             EUR          85847                 ES           100
1       30000             USD          30000                 US           100
2       25500             USD          25500                 US           100
3      175000             USD         175000                 CA           100
```

```
4       120000             USD          120000                  CA            100
…         …                …              …                     …             …
3750    412000             USD          412000                  US            100
3751    151000             USD          151000                  US            100
3752    105000             USD          105000                  US            100
3753    100000             USD          100000                  US            100
3754   7000000             INR           94665                  IN             50

     company_location company_size
0                  ES            L
1                  US            S
2                  US            S
3                  CA            M
4                  CA            M
…                 …             …
3750               US            L
3751               US            L
3752               US            S
3753               US            L
3754               IN            L

[2584 rows x 11 columns]
```

Budući da imamo sve plaće izražene u USD u stupcu salary_in_usd izbacujemo stupce salary i salary_currency.

[ ]:
```
#Izbacujemo stupac "salary"
podaci=podaci.drop('salary',axis=1)
```

[ ]:
```
#Izbacujemo stupac "salary_currency"
podaci=podaci.drop('salary_currency',axis=1)
```

[ ]:
```
podaci.head()     #ispis prvih 5 podataka nakon izbacivanja stupaca "salary" i␣
↪"salary_currency"
```

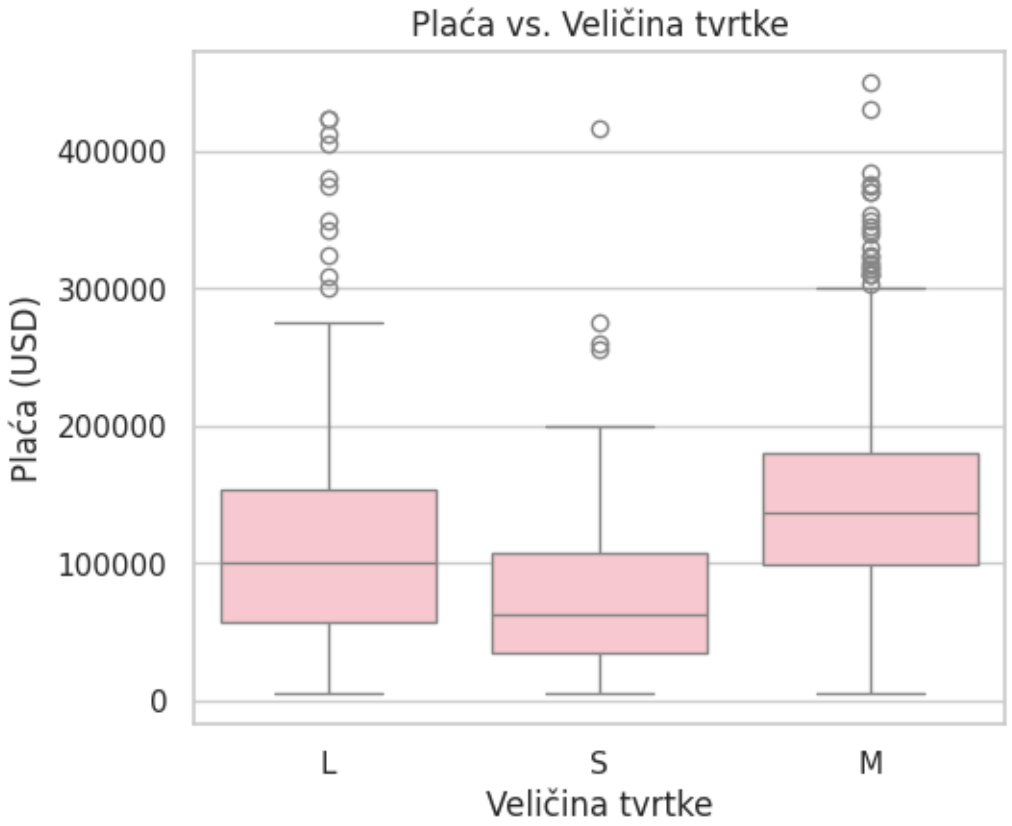
[ ]:
```
   work_year experience_level employment_type                 job_title  \
0       2023               SE              FT  Principal Data Scientist
1       2023               MI              CT             ML Engineer
2       2023               MI              CT             ML Engineer
3       2023               SE              FT          Data Scientist
4       2023               SE              FT          Data Scientist

   salary_in_usd employee_residence  remote_ratio company_location  \
0          85847                 ES           100               ES
1          30000                 US           100               US
2          25500                 US           100               US
3         175000                 CA           100               CA
```

```
4          120000                  CA            100                CA

  company_size
0            L
1            S
2            S
3            M
4            M
```

#3. Pronađimo sada odgovore na neka zanimljiva pitanja

###**Pitanje:** *Plaćaju li veće tvrtke više te kako prosječna plaća ovisi o veličini tvrtke?*

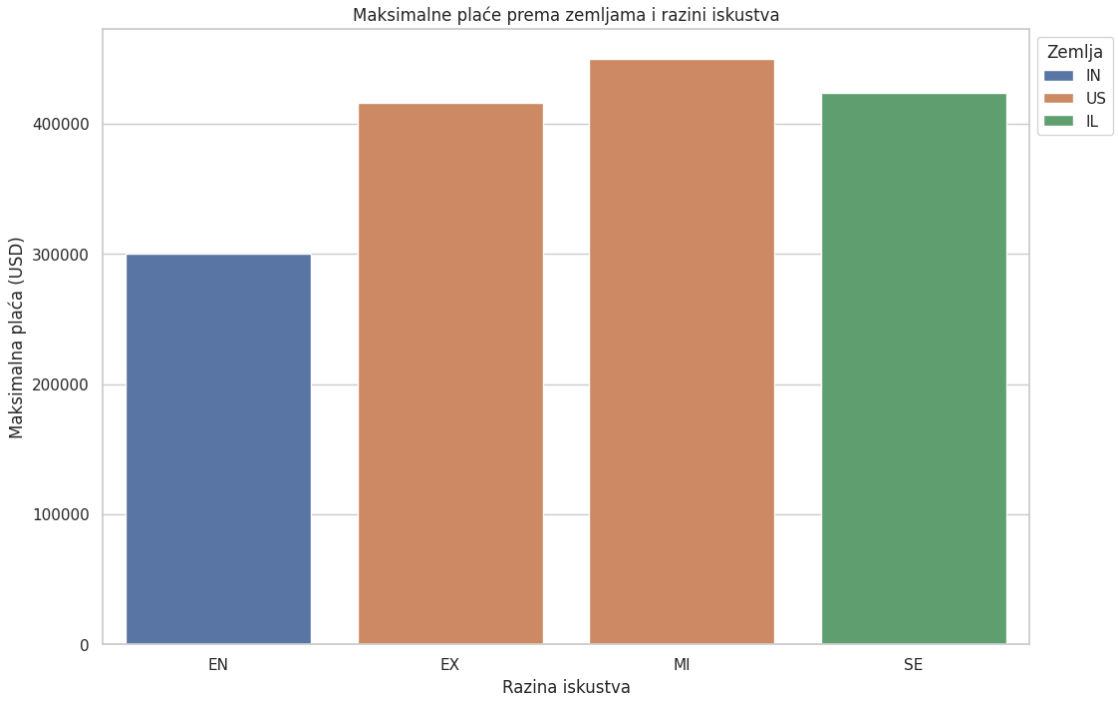
[ ]:
```
# Salary vs. Company Size
plt.figure(figsize=(12, 10))
plt.subplot(2, 2, 3)
sns.boxplot(x='company_size', y='salary_in_usd', data=podaci, color='pink')
plt.title('Plaća vs. Veličina tvrtke')
plt.xlabel('Veličina tvrtke')
plt.ylabel('Plaća (USD)')

plt.show()
```



Veće tvrtke obično nude više srednje plaće u usporedbi s srednjim i malim tvrtkama. To može biti zato što veće tvrtke imaju više resursa i obično privlače iskusniji kadar.Veće i srednje tvrtke imaju i više outliera za plaće koje su iznad prosjeka.

###**Pitanje:** *U kojoj zemlji tražiti posao s ciljem maksimizacije plaće ovisno o razini iskustva zaposlenika?*

[ ]:
```
# Sortiranje po stupcu iskustva
df_sorted = podaci.sort_values(by='experience_level')

# Grupiranje po zemljama i iskustvu te izračunavanje maksimalnih plaća
max_salaries_by_country = df_sorted.groupby(['company_location',␣
 ↪'experience_level'])['salary_in_usd'].max().reset_index()

# Prikaz zemalja s najvećim plaćama za svaku razinu iskustva
max_salaries_by_experience = max_salaries_by_country.
 ↪loc[max_salaries_by_country.groupby('experience_level')['salary_in_usd'].
 ↪idxmax()]

# Postavljanje stila grafa
sns.set(style="whitegrid")

# Stvaranje trakastog dijagrama
plt.figure(figsize=(12, 8))
sns.barplot(x='experience_level', y='salary_in_usd', hue='company_location',␣
 ↪data=max_salaries_by_experience)

# Dodavanje naslova i oznaka osi
plt.title('Maksimalne plaće prema zemljama i razini iskustva')
plt.xlabel('Razina iskustva')
plt.ylabel('Maksimalna plaća (USD)')

# Prikazivanje legende
plt.legend(title='Zemlja', bbox_to_anchor=(1, 1))

# Prikaz grafa
plt.show()
```



Iz grafa možemo zaključiti da je za razinu iskustva 'EN' (Entry) najveća prosječna plaća u Indiji, za razinu iskustva 'EX'(Executive) i 'MI'(Medium) u US, a za razinu iskustva 'SE'(Senior) u Izraelu.

1.2.1 Pitanje: *U kojoj su zemlji plaće iz područja podatkovne znanosti najviše?*

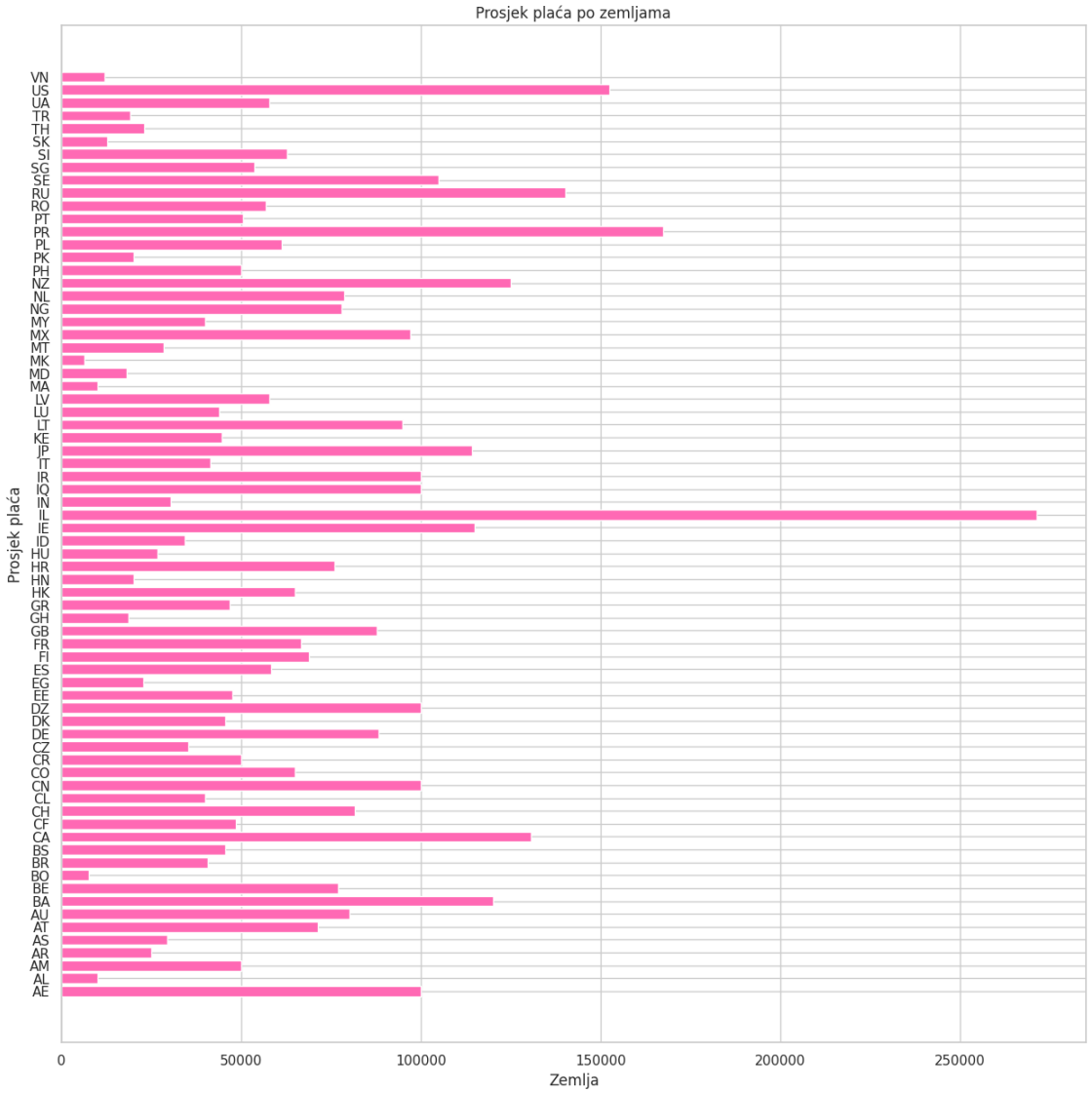
[ ]:
```
# Grupe po zemljama i računanje prosječnih plaća po zemlji
prosjeci_po_zemljama = podaci.groupby('company_location')['salary_in_usd'].
  ↪mean()

# Pronalaženje zemlje s najvišom prosječnom plaćom
zemlja_s_max_placom = prosjeci_po_zemljama.idxmax()
maksimalna_placa = prosjeci_po_zemljama.max()

# Ispisivanje rezultata
print(f"Najviša prosječna plaća ({maksimalna_placa}) je u zemlji:␣
  ↪{zemlja_s_max_placom}")   #IL - Izrael
```

```
Najviša prosječna plaća (271446.5) je u zemlji: IL
```

Prosječna plaća ovisno o lokaciji tvrtke:

[ ]:
```
# Postavljanje dimenzija slike
plt.figure(figsize=(15, 15))

# Stvaranje horizontalnog barplot-a
```

```
plt.barh(prosjeci_po_zemljama.index, prosjeci_po_zemljama, color='hotpink')

# Dodatne postavke grafikona
plt.title('Prosjek plaća po zemljama')
plt.xlabel('Zemlja')
plt.ylabel('Prosjek plaća')

# Prikazivanje histograma
plt.show()
```



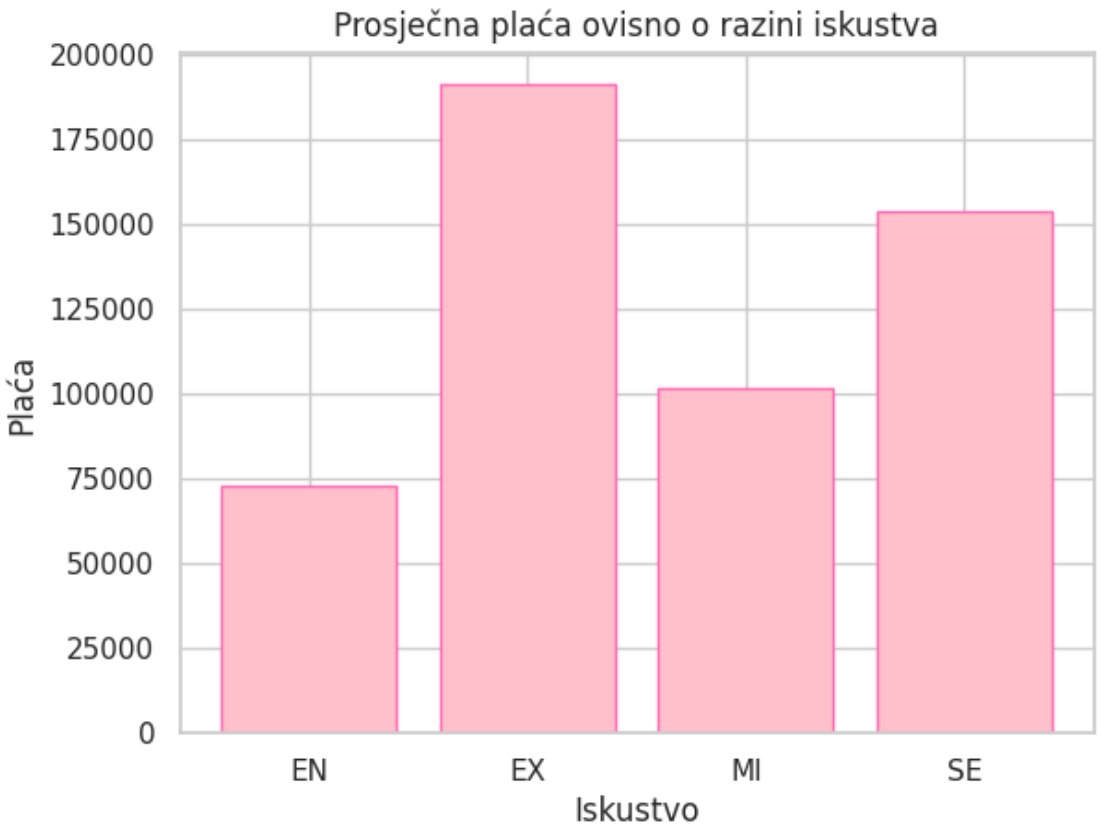
1.2.2 Pitanje: *U kojoj se zemlji većinski radi puno radno vrijeme?*

[ ]:
```
# Filtriranje redaka s "Puno radno vrijeme"
puno_radno_vrijeme = podaci[podaci['employment_type'] == 'FT']

# Grupe po zemljama i brojanje radnika puno radno vrijeme po zemlji
broj_puno_radno_vrijeme_po_zemlji = puno_radno_vrijeme.
 ↪groupby('company_location').size()

# Pronalaženje zemlje s najvećim brojem radnika puno radno vrijeme
zemlja_s_max_puno_radno_vrijeme = broj_puno_radno_vrijeme_po_zemlji.idxmax()
maksimalni_broj_puno_radno_vrijeme = broj_puno_radno_vrijeme_po_zemlji.max()

# Ispisivanje rezultata
print(f"U zemlji {zemlja_s_max_puno_radno_vrijeme} većinski se radi puno radno␣
 ↪vrijeme s brojem radnika: {maksimalni_broj_puno_radno_vrijeme}")
```

```
U zemlji US većinski se radi puno radno vrijeme s brojem radnika: 1911
```

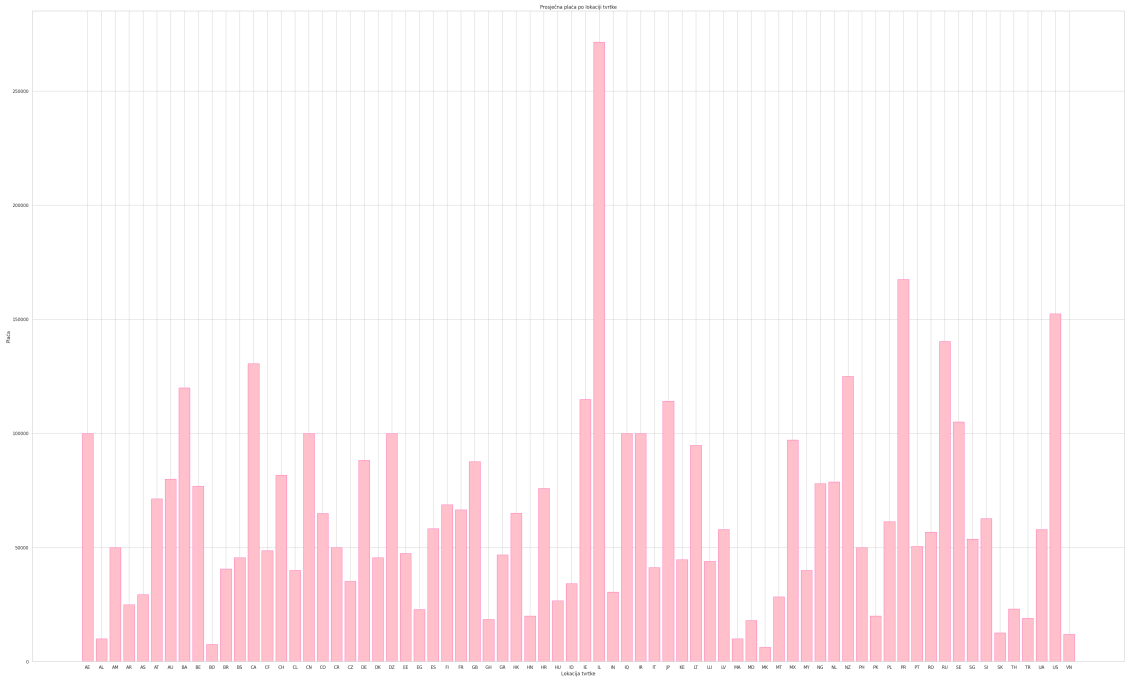
1.2.3 Pitanje: *Kolike su prosječne plaće ovisno o iskustvu?*

[ ]:
```
# Grupe po iskustvima i računanje prosječnih plaća po iskustvu
prosjeci_po_iskustvu = podaci.groupby('experience_level')['salary_in_usd'].
 ↪mean()

# Stvaranje barplota
plt.bar(prosjeci_po_iskustvu.index,prosjeci_po_iskustvu , color='pink',␣
 ↪edgecolor="hotpink")

# Dodatne postavke grafikona
plt.title("Prosječna plaća ovisno o razini iskustva")
plt.xlabel('Iskustvo')
plt.ylabel('Plaća')

# Prikazivanje barplota
plt.show()
```



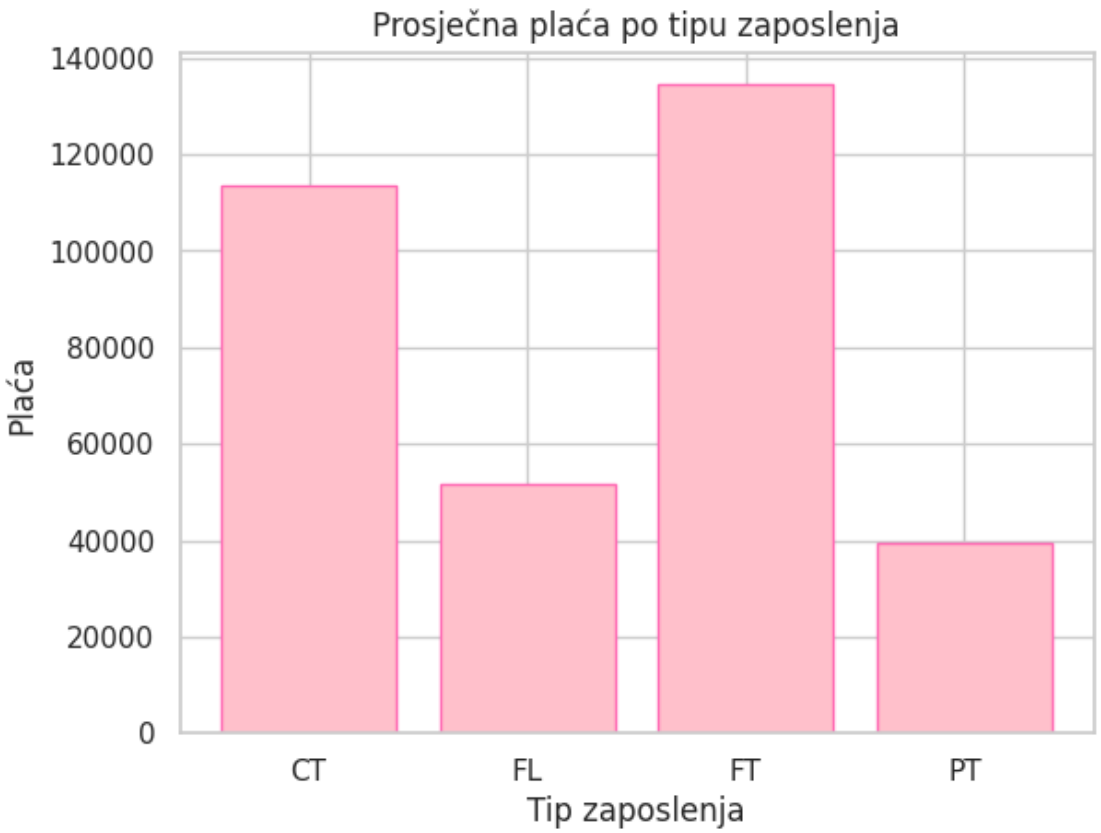
1.2.4 Pitanje: *Kolike su prosječne plaće ovisno o lokaciji tvrtke?*

[ ]:
```
plt.figure(figsize=(50, 30))

# Grupe po lokaciji tvrtke i računanje prosječnih plaća po lokaciji tvrtke
prosjeci_po_lokaciji_tvrtke = podaci.
 ↪groupby('company_location')['salary_in_usd'].mean()

# Stvaranje barplota
plt.bar(prosjeci_po_lokaciji_tvrtke.index,prosjeci_po_lokaciji_tvrtke ,␣
 ↪color='pink', edgecolor="hotpink")

# Dodatne postavke grafikona
plt.title("Prosječna plaća po lokaciji tvrtke")
plt.xlabel('Lokacija tvrtke')
plt.ylabel('Plaća')

# Prikazivanje barplota
plt.show()
```



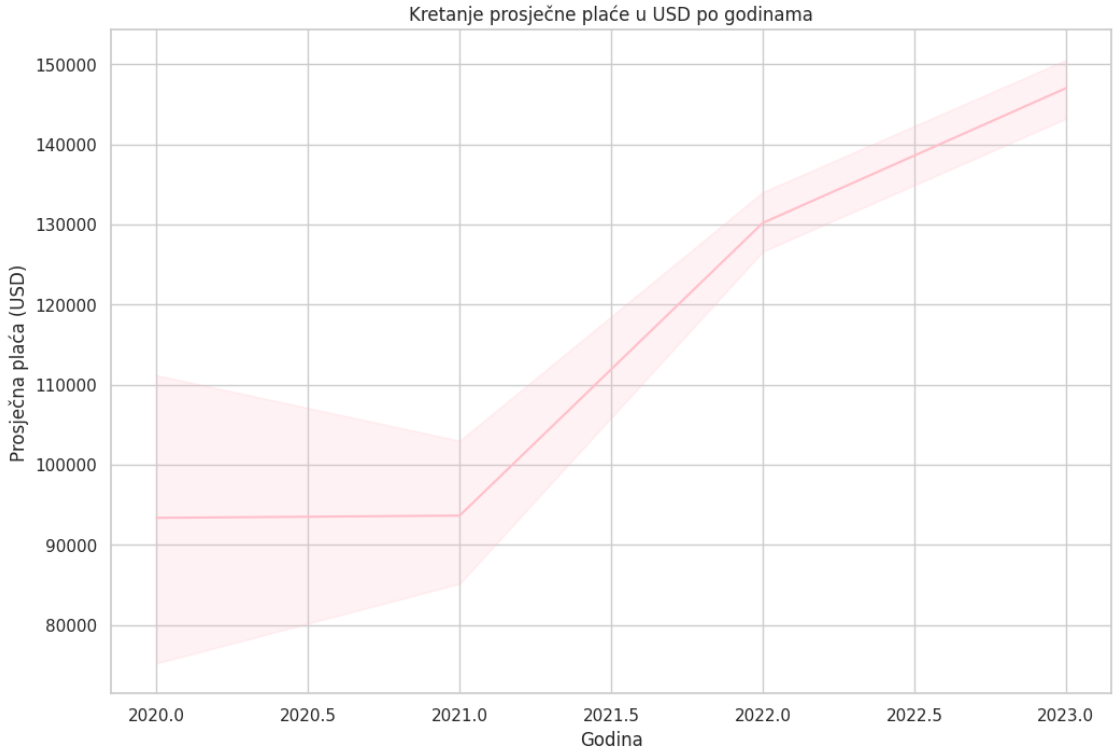
1.2.5 Pitanje: *Kolike su prosječne plaće ovisno o tipu zaposlenja?*

[ ]:
```
# Grupe po lokaciji tvrtke i računanje prosječnih plaća po tipu zaposlenja
prosjeci_po_tipu_zaposlenja = podaci.
 ↪groupby('employment_type')['salary_in_usd'].mean()

# Stvaranje barplota
plt.bar(prosjeci_po_tipu_zaposlenja.index,prosjeci_po_tipu_zaposlenja ,␣
 ↪color='pink', edgecolor="hotpink")

# Dodatne postavke grafikona
plt.title("Prosječna plaća po tipu zaposlenja")
plt.xlabel('Tip zaposlenja')
plt.ylabel('Plaća')

# Prikazivanje barplota
plt.show()
```



###Kretanje prosječne plaće po godinama

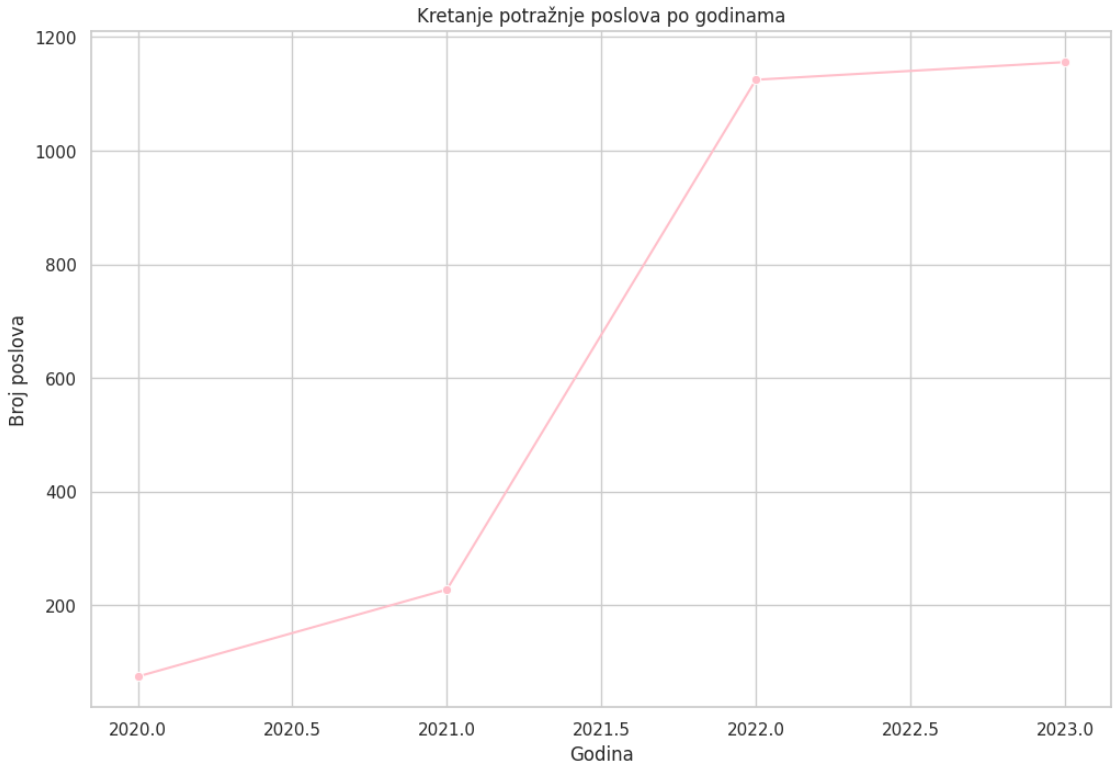
[ ]:
```
# Postavljanje stila grafa
sns.set(style="whitegrid")

# Stvaranje linijastog dijagrama
plt.figure(figsize=(12, 8))
sns.lineplot(x='work_year', y='salary_in_usd', data=podaci, color='pink')

# Dodavanje naslova i oznaka osi
plt.title('Kretanje prosječne plaće u USD po godinama')
plt.xlabel('Godina')
plt.ylabel('Prosječna plaća (USD)')

# Prikaz grafa
plt.show()
```



Iz grafa zaključujemo kako prosječna plaća raste te uočavamo nagli skok 2021. godine.

###Kretanje potražnje posla po godinama

[ ]:
```
# Grupiranje po godinama i računanje broja poslova
broj_poslova_po_godinama = podaci.groupby('work_year')['job_title'].count().
 ↪reset_index()

# Postavljanje stila grafa
sns.set(style="whitegrid")

# Stvaranje linijastog dijagrama
plt.figure(figsize=(12, 8))
sns.lineplot(x='work_year', y='job_title', data=broj_poslova_po_godinama,␣
 ↪marker='o', color='pink')

# Dodavanje naslova i oznaka osi
plt.title('Kretanje potražnje poslova po godinama')
plt.xlabel('Godina')
plt.ylabel('Broj poslova')

# Prikaz grafa
plt.show()
```

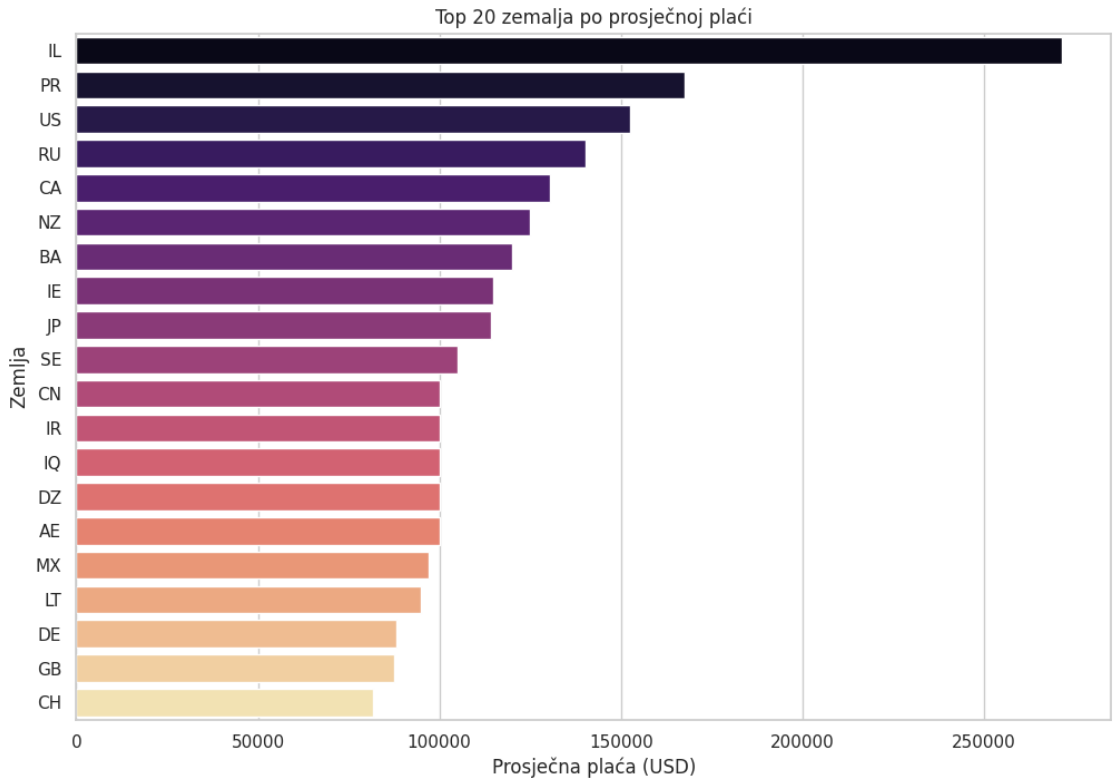


Iz grafa zaključujemo da potražnja posla raste te se nagli skok dogodio u 2021. godini.

###Top 20 zemlja po prosječnoj plaći

[ ]:
```
# Grupiranje po zemljama i računanje prosječne plaće
prosjecna_placa_po_zemljama = podaci.
 ↪groupby('company_location')['salary_in_usd'].mean().reset_index()

# Sortiranje po prosječnoj plaći
top_20_zemalja_placa = prosjecna_placa_po_zemljama.
 ↪sort_values(by='salary_in_usd', ascending=False).head(20)

# Prikaz top 20 zemalja po prosječnoj plaći
plt.figure(figsize=(12, 8))
sns.barplot(x='salary_in_usd', y='company_location', data=top_20_zemalja_placa,␣
 ↪palette='magma')

plt.title('Top 20 zemalja po prosječnoj plaći')
plt.xlabel('Prosječna plaća (USD)')
plt.ylabel('Zemlja')

plt.show()
```

```
<ipython-input-1208-5f5a6b61f166>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='salary_in_usd', y='company_location',
data=top_20_zemalja_placa, palette='magma')
```

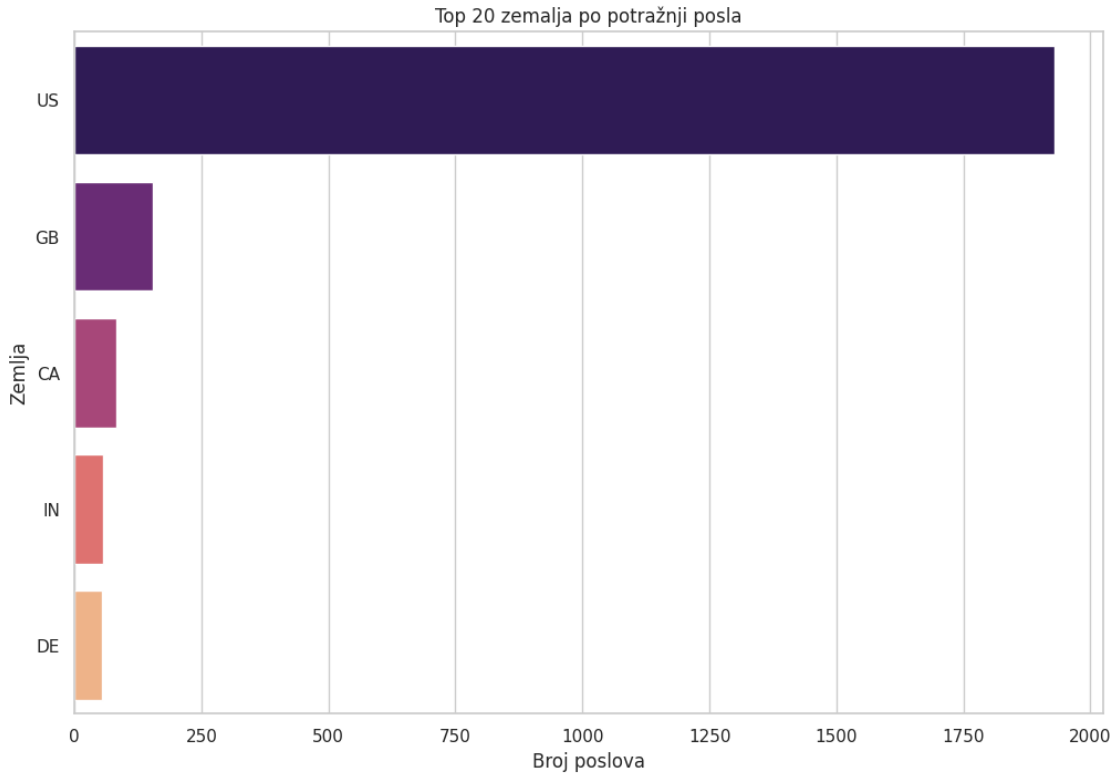


###Top 5 zemlja po potražnji posla

[ ]:
```
# Grupiranje po zemljama i računanje broja poslova
broj_poslova_po_zemljama = podaci.groupby('company_location')['job_title'].
  ↪count().reset_index()
broj_poslova_po_zemljama.columns = ['company_location', 'broj_poslova']

# Sortiranje po broju poslova
top_5_zemalja_poslova = broj_poslova_po_zemljama.sort_values(by='broj_poslova',␣
  ↪ascending=False).head(5)

# Prikaz top 20 zemalja po broju poslova
plt.figure(figsize=(12, 8))
```

```
sns.barplot(x='broj_poslova', y='company_location', data=top_5_zemalja_poslova,␣
 ↪palette='magma')

plt.title('Top 20 zemalja po potražnji posla')
plt.xlabel('Broj poslova')
plt.ylabel('Zemlja')

plt.show()
```

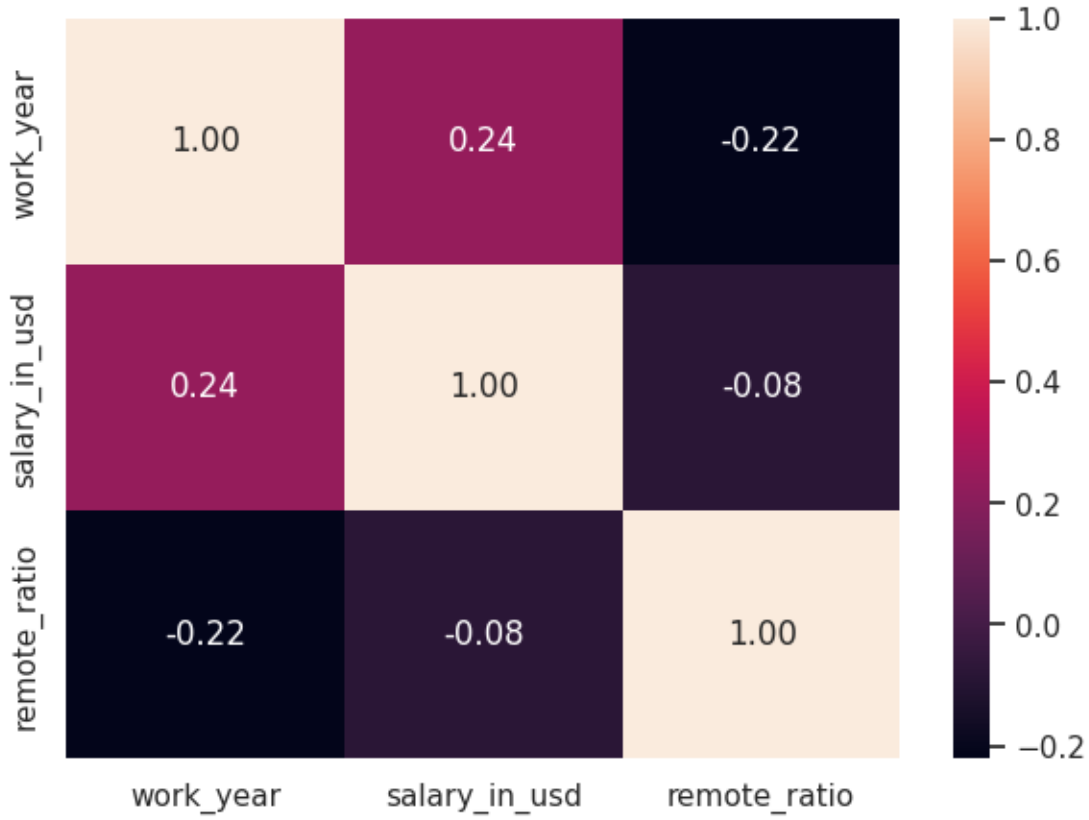
```
<ipython-input-1209-073dcc3a1cbe>:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x='broj_poslova', y='company_location',
data=top_5_zemalja_poslova, palette='magma')
```



#4. Veze između varijabli

##Matrica korelacije

[ ]:
```
plt.figure(figsize=(7,5))
sns.heatmap(podaci.corr(numeric_only=True, method="pearson"), annot=True,␣
↪cmap="rocket", fmt='.2f')
plt.show()
```



Iz matrice korelacije zaključujemo da je korelacija između 'work_year' i 'salary_in_usd' relativno slaba dok je između ostalih neznatna.

##Hipoteze

[ ]:
```
import scipy.stats as stats
```

###1.Je li prosječna plaća onih zaposlenika koji imaju iskustvo "SE" statistički različita od zaposlenika koji imaju iskustvo "EX"?

H0 : prosječna plaća onih zaposlenika koji imaju iskustvo "SE" jednaka je prosječnoj plaći zaposlenika koji imaju iskustvo "EX".

H1: prosječna plaća onih zaposlenika koji imaju iskustvo "SE" različita je prosječnoj plaći zaposlenika koji imaju iskustvo "EX".

Razina značajnosti testa: $= 0.05$

Prvo ćemo provjeriti jesu li naši podaci normalno distribuirani - i to je jedna vrsta testiranja hipoteza.

Hipoteze:

H0 : podaci su normalno distribuirani H1 : podaci nisu normalno distribuirani Uzmimo razinu značajnosti =0.05 .

[ ]:
```
SE = podaci[podaci['experience_level']=="SE"]        #u varijablu SE spremamo␣
 ↪sve podatke kojima je razina iskustva "SE"
EX = podaci[podaci['experience_level']=="EX"]        #u varijablu EX spremamo␣
 ↪sve podatke kojima je razina iskustva "EX"
```

[ ]:
```
SE['salary_in_usd'].mean()          #računamo srednju vrijednost plaća za␣
 ↪razinu iskustva "SE"
```

[ ]:
```
153897.43564993565
```

[ ]:
```
EX['salary_in_usd'].mean()          #računamo srednju vrijednost plaća za razinu␣
 ↪iskustva "EX"
```

[ ]:
```
191078.20833333334
```

[ ]:
```
SE_place=SE['salary_in_usd']      #u varijablu SE_place spremamo plaće za␣
 ↪razinu iskustva "SE"
```

[ ]:
```
EX_place=EX['salary_in_usd']      #u varijablu EX_place spremamo plaće za␣
 ↪razinu iskustva "EX"
```

[ ]:
```
statistic, pvalue = stats.normaltest(SE_place)
print(f"pvalue: {pvalue}")
print(f"pvalue < 0.05: {pvalue < 0.05}")
```

```
pvalue: 8.320636941656847e-25
pvalue < 0.05: True
```

p-vrijednost je manja od 0.05

Odbacujemo H0 tj. podaci nisu normalno distribuirani

[ ]:
```
statistic, pvalue = stats.normaltest(EX_place)
print(f"pvalue: {pvalue}")
print(f"pvalue < 0.05: {pvalue < 0.05}")
```

```
pvalue: 0.12418209782490677
pvalue < 0.05: False
```

S obzirom da nam podaci ne prate normalnu distribuciju, moramo odustati od parametarskog testa te odabrati neparametarski test. Neparametarski ekvivalent studentovom t-testu je Mann-Whitney U test.

[ ]:
```
statistic, pvalue = stats.mannwhitneyu(SE_place, EX_place)
print(f"pvalue: {pvalue}")
print(f"pvalue < 0.05: {pvalue < 0.05}")
```

```
pvalue: 2.234061088352505e-07
pvalue < 0.05: True
```

p-vrijednost $<$: Odbacujemo H0 tj ona je neistinita i prihvacamo H1 kao istinitu.

###2.Postoji li statistički značajna razlika između plaće i tipa zaposlenja?

H0 : tip zaposlenja i plaća su nezavisni

H1: tip zaposlenja i plaća su zavisni.

razina značajnosti testa: $= 0.05$

[ ]:
```
# Stvorimo tablicu kontingencije
cross_table = pd.crosstab(podaci['employment_type'], podaci['salary_in_usd'])

# Ispis tablice kontingencije
print(cross_table)
```

```
salary_in_usd    5132    5409    5679    5707    5723    5882    6072    \
employment_type
CT                  0       0       0       0       0       0       0
FL                  0       0       0       0       0       0       0
FT                  1       1       1       1       1       1       2
PT                  0       1       0       0       0       0       0

salary_in_usd    6270    6304    6359    …  376080  380000  385000  405000  \
employment_type                          …
CT                  0       0       0  …       0       0       0       0
FL                  0       0       0  …       0       0       0       0
FT                  1       1       1  …       1       1       1       1
PT                  0       0       0  …       0       0       0       0

salary_in_usd    412000  416000  423000  423834  430967  450000
employment_type
CT                  0       1       0       0       0       0
FL                  0       0       0       0       0       0
FT                  1       0       1       1       1       1
PT                  0       0       0       0       0       0

[4 rows x 1035 columns]
```

[ ]:
```
from scipy.stats import chi2_contingency
# Chi-kvadrat test nezavisnosti
chi2, p, _, _ = chi2_contingency(cross_table)
```

```
# Ispis rezultata testa
print(f"\nChi-kvadrat vrijednost: {chi2}")
print(f"P-vrijednost: {p}")

# Provjera značajnosti na razini   = 0.05
alpha = 0.05
if p < alpha:
    print("\nOdbacujemo H0: Postoji statistički značajna veza između plaće i␣
 ↪tipa zapolenja.")
else:
    print("\nPrihvaćamo H0: Nema dovoljno dokaza o statistički značajnoj vezi␣
 ↪između plaće i tipa zaposlenja.")
```

```
Chi-kvadrat vrijednost: 3770.6295214369184
P-vrijednost: 9.024076152710984e-16

Odbacujemo H0: Postoji statistički značajna veza između plaće i tipa zapolenja.
```

#5. Grupiranje podataka

##K-means algoritam

K-means algoritam je algoritam nenadziranog učenja koji dani skup podataka grupira u unaprijed zadan broj grupa k.

[ ]:
```
from sklearn.cluster import KMeans
```

###Kodiranje kategoričkih značajki

Da bismo mogli koristiti kategoričke varijable u algoritmu strojnog učenja, moramo ih kodirati u numeričke vrijednosti.

[ ]:
```
podaci
```

[ ]:
```
      work_year experience_level employment_type                 job_title  \
0          2023               SE              FT  Principal Data Scientist
1          2023               MI              CT               ML Engineer
2          2023               MI              CT               ML Engineer
3          2023               SE              FT            Data Scientist
4          2023               SE              FT            Data Scientist
…           …                …               …                         …
3750       2020               SE              FT            Data Scientist
3751       2021               MI              FT  Principal Data Scientist
3752       2020               EN              FT            Data Scientist
3753       2020               EN              CT     Business Data Analyst
3754       2021               SE              FT      Data Science Manager

      salary_in_usd employee_residence  remote_ratio company_location  \
```

```
0             85847                  ES           100                ES
1             30000                  US           100                US
2             25500                  US           100                US
3            175000                  CA           100                CA
4            120000                  CA           100                CA
…               …                    …             …                 …
3750         412000                  US           100                US
3751         151000                  US           100                US
3752         105000                  US           100                US
3753         100000                  US           100                US
3754          94665                  IN            50                IN

     company_size
0               L
1               S
2               S
3               M
4               M
…             …
3750            L
3751            L
3752            S
3753            L
3754            L

[2584 rows x 9 columns]
```

[ ]:
```
work_year_mapping={2020:0, 2021:1, 2022:2, 2023:3}
experience_level_mapping={'SE':0, 'MI':1, 'EN':2,'EX':3}
employment_type_mapping={'FT':0, 'PT':1,'CT':2,'FL':3}
company_size_mapping={'L':0,'M':1,'S':2}
remote_ratio_mapping={0:0,50:1, 100:2}

podaci['work_year'] = podaci['work_year'].map(work_year_mapping)
podaci['experience_level'] = podaci['experience_level'].
 ↪map(experience_level_mapping)
podaci['employment_type'] = podaci['employment_type'].
 ↪map(employment_type_mapping)
podaci['company_size'] = podaci['company_size'].map(company_size_mapping)
podaci['remote_ratio'] = podaci['remote_ratio'].map(remote_ratio_mapping)
```

[ ]:
```
podaci.
 ↪drop(['job_title','employee_residence','company_location'],axis=1,inplace=True)
```

[ ]:
```
podaci
```

[ ]:

	work_year	experience_level	employment_type	salary_in_usd	\
0	3	0	0	85847	
1	3	1	2	30000	
2	3	1	2	25500	
3	3	0	0	175000	
4	3	0	0	120000	
…	…	…	…	…	
3750	0	0	0	412000	
3751	1	1	0	151000	
3752	0	2	0	105000	
3753	0	2	2	100000	
3754	1	0	0	94665	

	remote_ratio	company_size
0	2	0
1	2	2
2	2	2
3	2	1
4	2	1
…	…	…
3750	2	0
3751	2	0
3752	2	2
3753	2	0
3754	1	0

[2584 rows x 6 columns]

Skalirajmo prvo numeričke značajke koje ćemo u ovom slučaju koristit za grupiranje

[ ]:
```
from sklearn.preprocessing import StandardScaler
```

[ ]:
```
scaler = StandardScaler()
```

[ ]:
```
podaci['salary_in_usd'] = scaler.fit_transform(podaci[['salary_in_usd']])
```

[ ]:
```
podaci
```

[ ]:

	work_year	experience_level	employment_type	salary_in_usd	\
0	3	0	0	-0.708575	
1	3	1	2	-1.540574	
2	3	1	2	-1.607615	
3	3	0	0	0.619612	
4	3	0	0	-0.199769	
…	…	…	…	…	
3750	0	0	0	4.150399	
3751	1	1	0	0.262064	

```
3752          0                  2                  0       -0.423237
3753          0                  2                  2       -0.497726
3754          1                  0                  0       -0.577206

      remote_ratio  company_size
0                2             0
1                2             2
2                2             2
3                2             1
4                2             1
…              …             …
3750             2             0
3751             2             0
3752             2             2
3753             2             0
3754             1             0

[2584 rows x 6 columns]
```

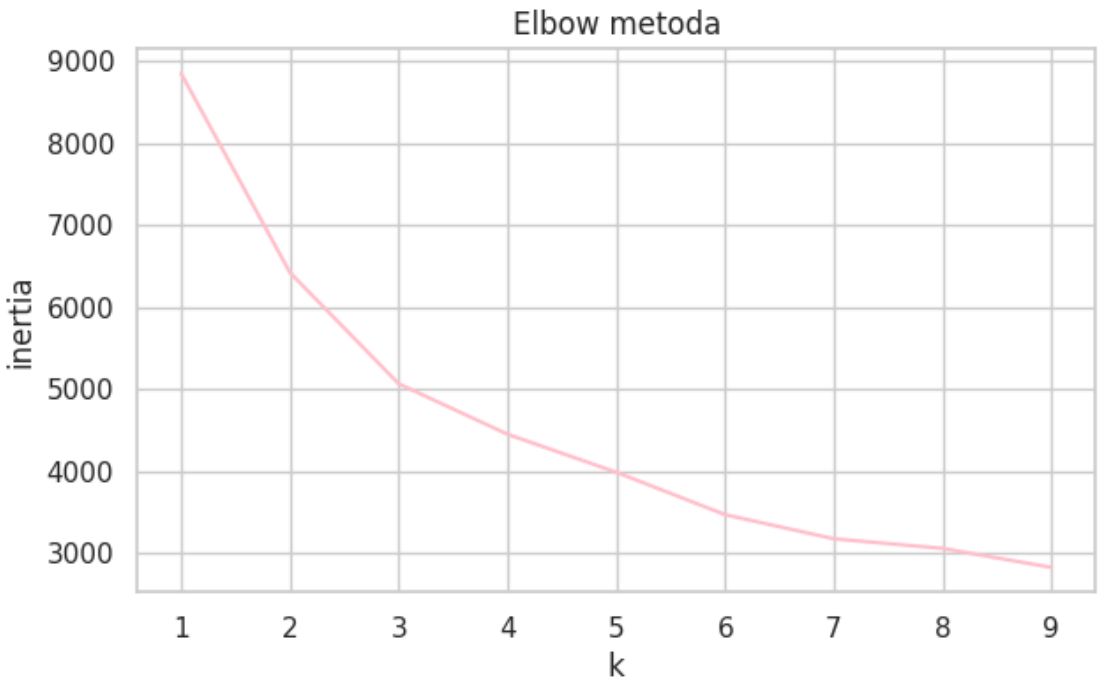
###Elbow metoda za najbolji broj grupa

Promatramo kako dodavanje nove grupe smanjuje sumu kvadrata udaljenosti (ssd) podataka od centara.Odabiremo k takav da se njegovim povećanjem ssd neznatno smanji,a njegovim smanjivanjem pogreška se znatno poveća.

[ ]:
```
ssd=[]
ks=range(1,10)
for k in ks:
  k_model=KMeans(n_clusters=k, n_init='auto')     #n_init="auto" - koliko puta
 ↪će se KMeans algoritam izvršiti s različitim (slučajnim) inicijalizacijama
 ↪početnih centara (default = 10)
  k_model.fit(podaci)
  ssd.append(k_model.inertia_)
print(ssd)
```

```
[8851.732972136226, 6423.479299666243, 5065.882452965847, 4450.061862519583,
3986.0597980214607, 3468.615271398648, 3173.7956667686312, 3056.6806861074015,
2824.691603160981]
```

[ ]:
```
plt.figure(figsize=(7,4))
plt.plot(ks,ssd, color='pink')
plt.xlabel('k')
plt.ylabel('inertia')
plt.title('Elbow metoda')
plt.show()
```



Iz grafa zaključujemo da je najbolje broj grupa postaviti na 3.

[ ]:
```
kmeans=KMeans(n_clusters=3, n_init='auto')
kmeans.fit(podaci)
```

[ ]:
```
KMeans(n_clusters=3, n_init='auto')
```

[ ]:
```
centers=kmeans.cluster_centers_
print(f'Centri:\n{centers}')
```

```
Centri:
[[ 2.35154394  0.16983373  0.00950119  0.39074163  1.98337292  0.90498812]
 [ 2.62009132  0.39634703  0.00547945  0.26128805  0.01004566  0.94063927]
 [ 1.69551777  1.41421947  0.08191654 -0.9507185   1.43431221  0.81916538]]
```

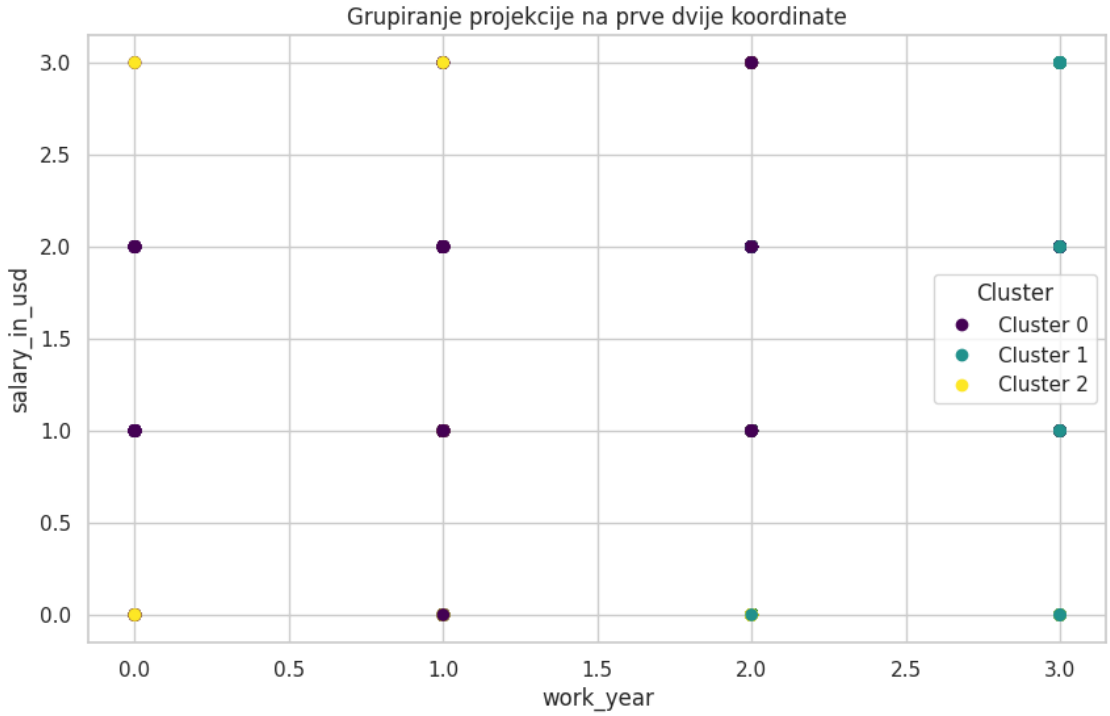
[ ]:
```
y_kmeans=kmeans.predict(podaci)        #za svaku točku odredi najbližu grupu␣
  ↪kojoj pripada
print(y_kmeans[:15])
```

```
[0 2 2 0 0 1 1 1 1 1 1 0 0 1 1]
```

[ ]:
```
podaci['cluster'] = kmeans.fit_predict(podaci)     #u stupcu "cluster" spremamo␣
  ↪predviđenu grupu za podatke
```

Prikaz grupiranja projekcije na prve 2 koordinate

```
[ ]: # Prikazivanje grupiranja projekcije na prve dvije koordinate
plt.figure(figsize=(10, 6))
scatter = plt.scatter(podaci.iloc[:, 0], podaci.iloc[:, 1],␣
 ↪c=podaci['cluster'], cmap='viridis')

# Dodatne postavke grafikona
plt.title('Grupiranje projekcije na prve dvije koordinate')
plt.xlabel('work_year')
plt.ylabel('salary_in_usd')

# Dodavanje legende
legend_labels = ['Cluster 0', 'Cluster 1','Cluster 2']
plt.legend(handles=scatter.legend_elements()[0], labels=legend_labels,␣
 ↪title='Cluster')

# Prikazivanje grafikona
plt.show()
```



Kvaliteta grupiranja (kreće se [-1, 1], a 1 je najbolja)

```
[ ]: from sklearn.metrics import silhouette_score
silhouette_avg=silhouette_score(podaci, y_kmeans)
print("Silhouette Score:", silhouette_avg)
```

```
Silhouette Score: 0.3947491590539465
```

##DBSCAN algoritam

DBSCAN algoritam je algoritam nenadziranog učenja koji grupira podatke na temelju gustoće pri čemu sam odredi broj grupa.

[ ]:
```
from sklearn.cluster import DBSCAN
```

[ ]:
```
# Primjena DBSCAN algoritma
dbscan = DBSCAN(eps=1, min_samples=3)
podaci['cluster_DBSCAN'] = dbscan.fit_predict(podaci)
```

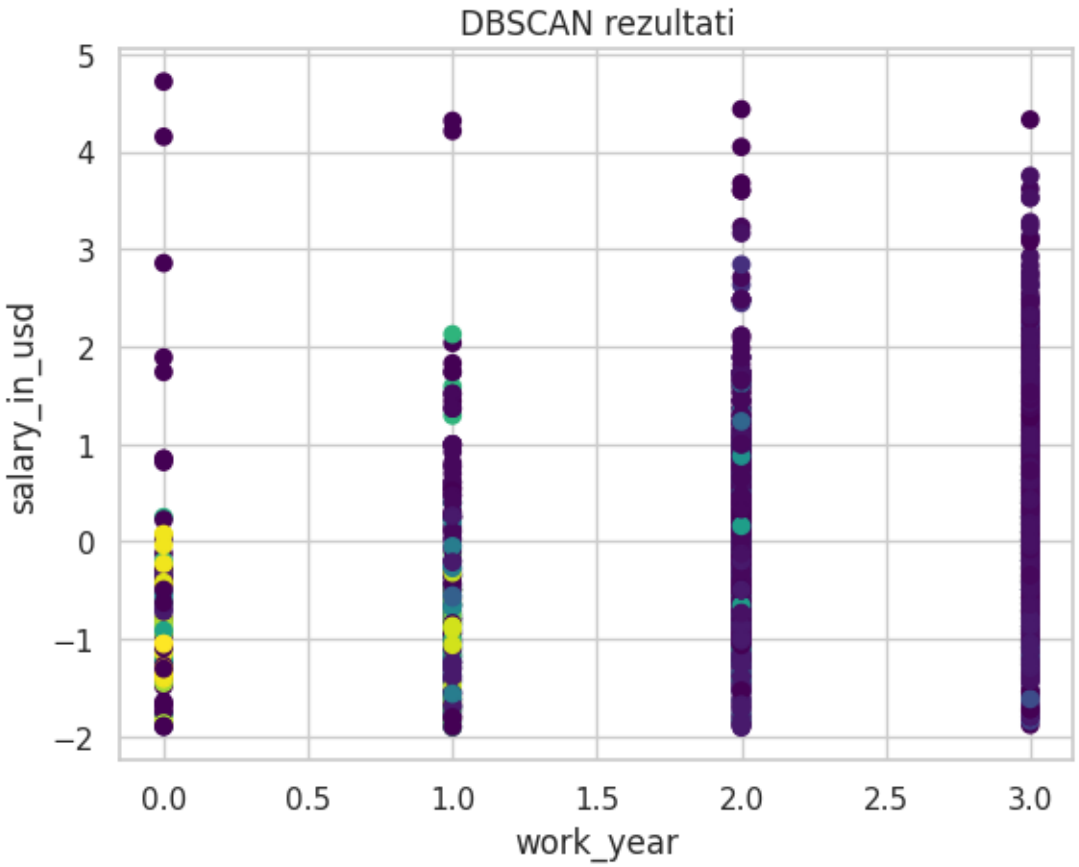
[ ]:
```
# Ispisivanje rezultata
podaci          #s -1 su označeni outlieri
```

[ ]:
```
      work_year  experience_level  employment_type  salary_in_usd  \
0             3                 0                0      -0.708575
1             3                 1                2      -1.540574
2             3                 1                2      -1.607615
3             3                 0                0       0.619612
4             3                 0                0      -0.199769
…           …                 …                …              …
3750          0                 0                0       4.150399
3751          1                 1                0       0.262064
3752          0                 2                0      -0.423237
3753          0                 2                2      -0.497726
3754          1                 0                0      -0.577206

      remote_ratio  company_size  cluster  cluster_DBSCAN
0                2             0        2               0
1                2             2        0              -1
2                2             2        0              -1
3                2             1        2               0
4                2             1        2               0
…              …             …      …              …
3750             2             0        2              -1
3751             2             0        0               2
3752             2             2        0              40
3753             2             0        0              -1
3754             1             0        0              12

[2584 rows x 8 columns]
```

Prikaz grupiranja projekcije na prve 2 koordinate

[ ]:
```
# Vizualizacija rezultata
plt.scatter(podaci['work_year'], podaci['salary_in_usd'],␣
 ↪c=podaci['cluster_DBSCAN'], cmap='viridis')
```

```
plt.title('DBSCAN rezultati')
plt.xlabel('work_year')
plt.ylabel('salary_in_usd')
plt.show()
```



[ ]:
```
# Broj grupa
num_clusters = len(set(podaci['cluster_DBSCAN'])) - (1 if -1 in␣
↪podaci['cluster_DBSCAN'] else 0)
print(f"Broj grupa (klastera): {num_clusters}")
```

```
Broj grupa (klastera): 44
```

#6.Linearna regresija

Linearna regresija je model nadrziranog učenja za predviđanje kontinuiranih varijabli. U našem slučaju predviđamo visinu plaća u USD.

[ ]:
```
from sklearn.model_selection import train_test_split        # podjela podataka␣
↪na skup za treniranje i testiranje
```

```
from sklearn.linear_model import LinearRegression          # model linearne␣
 ↪regresije
from sklearn import metrics
```

[ ]:
```
#Podjela podataka na skup za treniranje i testiranje
X_train, X_test = train_test_split(podaci, train_size=0.8, random_state=7)
y_train = X_train['salary_in_usd']
y_test = X_test['salary_in_usd']

X_train.drop(['salary_in_usd'], axis=1, inplace=True)
X_test.drop(['salary_in_usd'], axis=1, inplace=True)

print("Skup za treniranje:", X_train.shape)
print("Skup za testiranje:", X_test.shape)
```

```
Skup za treniranje: (2067, 7)
Skup za testiranje: (517, 7)
```

[ ]:
```
y_train= np.array(y_train).reshape(-1,1)
y_test= np.array(y_test).reshape(-1,1)
```

[ ]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

[ ]:
```
LinearRegression()
```

[ ]:
```
print(model.intercept_)
print(model.coef_)
```

```
[-0.58719523]
[[ 0.03700562  0.07547241 -0.21127998 -0.25495752 -0.08141211  0.73716218
  -0.00331718]]
```

Dakle, jednadžba dobivenog regresijskog pravca je:

$y = 0.82194477 + 0.166028959x_1 - 0.17799343x_2 - 0.39430873x_3 - 0.47312354x_4 - 0.00669593x_5 - 0.50763315x_6 - 0.023$

[ ]:
```
def evaluate_regression(y, y_pred):
  print('MAE: {:.3f}'.format(metrics.mean_absolute_error(y, y_pred)))
  print('MSE: {:.3f}'.format(metrics.mean_squared_error(y, y_pred)))
  print('RMSE: {:.3f}'.format(np.sqrt(metrics.mean_squared_error(y, y_pred))))
  print('Koeficijent determinacije: {:.3f}'.format(metrics.r2_score(y, y_pred)))
```

[ ]:
```
# Evaluacija na skupu za treniranje
y_pred_train = model.predict(X_train)
evaluate_regression(y_train, y_pred_train)
```

```
MAE: 0.637
MSE: 0.678
RMSE: 0.823
Koeficijent determinacije: 0.295
```

[ ]:
```
# Evaluacija na skupu za testiranje
y_pred_test = model.predict(X_test)
evaluate_regression(y_test, y_pred_test)
```

```
MAE: 0.696
MSE: 0.848
RMSE: 0.921
Koeficijent determinacije: 0.266
```

#7.Polinomijalna regresija

Polinomijalna regresija je model nadrziranog učenja za predviđanje kontinuiranih varijabli. To je poopćenje linearne regresije gdje umjesto linearne funkcije koja se koristi za predviđanje zavisne varijable y sada koristimo polinom. U našem slučaju predviđamo visinu plaća u USD.

[ ]:
```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import Pipeline
```

[ ]:
```
model_poly = Pipeline([('poly', PolynomialFeatures(degree=2)),
                       ('linear', LinearRegression(fit_intercept=False))])

model_poly.fit(X_train, y_train)
```

[ ]:
```
Pipeline(steps=[('poly', PolynomialFeatures()),
                ('linear', LinearRegression(fit_intercept=False))])
```

[ ]:
```
y_pred_test_poly = model_poly.predict(X_test)          # plaće koje␣
↪polinomijalni model predvidi
```

[ ]:
```
evaluate_regression(y_test, y_pred_test_poly)
```

```
MAE: 0.632
MSE: 0.716
RMSE: 0.846
Koeficijent determinacije: 0.380
```

Provedbom linearne i polinomijalne regresije na našem skupu podataka zaključujemo da polinomijalna daje malo bolje rezultate.