

3. Upoznavanje s Pandas bibliotekom. Predobradom podataka

3.1 Cilj Vježbe

Upoznati se s načinom korištenja Pandas biblioteke za programski jezik Python. Upoznati se s predobradom podataka i eksplorativnom analizom podataka pomoću grafičkog prikaza podataka.

3.2 Teorijska pozadina

U vježbi se studenti upoznaju s Pandas bibliotekom za programski jezik Python. Ova biblioteka omogućava relativno laganu manipulaciju tabularnim podacima te, zajedno s grafičkom bibliotekom, omogućuje dobivanje uvida u karakteristike raspoloživih podataka (distribucije, srednje vrijednosti i sl.). Ovo je obično i prvi korak u problemima strojnog učenja, a poznat je i pod nazivom eksplorativna analiza podataka.

3.2.1 Tabularni prikaz podataka

Dostupni podaci (podatkovni skup, engl. *dataset*) često se pohranjuju u tabularnom obliku pri čemu stupci predstavljaju veličine, a svaki redak predstavlja jedno mjerjenje svih veličina. Veličine mogu biti:

- numeričkog tipa (cjelobrojne ili realne vrijednosti)
- kategoričkog tipa

nominalne veličine – ne postoji odnos između mogućih vrijednosti (npr. boja očiju osobe – plave, smeđe...)

ordinalne veličine – postoji odnos između mogućih vrijednosti te je moguće ovakve vrijednosti poredati odnosno sortirati (npr. stupanj obrazovanja osobe).

U tablici 3.1 dan je isječak podatkovnog skupa koji sadrži mjerjenja provedena na različitim vozilima u Kanadi [1] za modele iz 2020., 2021. i 2022. godine. Primjerice, gradska potrošnja izražena u litrama na 100 prijeđenih kilometara je numerička veličina dok je vrsta goriva koje vozilo koristi kategorička veličina. Opis pojedine varijable dan je u dodatku vježbi, a podatkovni skup dostupan je u datoteci naziva `data_C02_emissions.csv`.

Poglavlje 3. Upoznavanje s Pandas bibliotekom. Predobrada podataka i eksplorativna analiza podataka

Proizvođač	Model	Klasa	Veličina motora	Broj cilindara	Vrsta prijenosa	Vrsta goriva	Gradska potrošnja (L/100km)	Izvengradska potrošnja (L/100km)	Kombinirana potrošnja (L/100km)	Kombinirana potrošnja (mpg)	Emisija CO2 (g/km)
Audi	A3	Subcompact	2.0	4	AM7	X	8.8	6.5	7.8	36	182
Hyundai	Kona AWD	SUV:Small	1.6	4	AM7	X	8.8	7.4	8.2	33	193
Infiniti	QX60 AWD	SUV:Standard	3.5	6	AS9	Z	11.9	9.5	10.8	26	253
...

Tablica 3.1: Primjer tabularnih zapisa u podatkovnom skupu

3.2.2 Pandas biblioteka

Pandas¹ je *open source* Python biblioteka koja značajno olakšava učitavanje i analizu tabularnih podataka u Pythonu. Osnovna struktura podataka u Pandas biblioteci su DataFrame i Series objekti zasnovani na Numpy koji omogućuje brzu i efikasnu manipulaciju pohranjenih podataka. U Pandas biblioteci su dostupni alati za učitavanje datoteka u kojima su pohranjeni podaci kao na primjer CSV i tekstualne datoteke, Excel, SQL baza i HDF5 datoteke. Učitani podaci spremaju se u DataFrame, a omogućen je ispis DataFrame u datoteke. Podržane su različite operacije nad DataFrameovima, kao izdvajanje, grupiranje i slično. Na taj način je moguće brzo dobiti uvid u karakteristike raspoloživih podataka.

Pandas Series

Series je jednodimenzionalni objekt sličan polju pri čemu je svakom elementu polja pridružen indeks (vrijednosti indeksa su 0 do N gdje je N duljina polja). Element polja može biti bilo koji tip podatka (cjelobrojne vrijednosti, realni brojevi, znakovni nizovi, Python objekti, itd.) kao što prikazuje primjer 3.1.

■ Primjer 3.1

```
import pandas as pd
import numpy as np

s1 = pd.Series(['crvenkapica', 'baka', 'majka', 'lovac', 'vuk'])
print(s1)

s2 = pd.Series(5., index=['a', 'b', 'c', 'd', 'e'], name = 'ime_objekta')
print(s2)
print(s2['b'])

s3 = pd.Series(np.random.randn(5))
print(s3)
print(s3[3])
```

Pandas DataFrame

Struktura DataFrame je dvodimenzionalna označena struktura nalik tablici gdje su u stupcima pohranjeni podaci istog tipa (npr. liste, rječnici, numpy polja i sl.). DataFrame struktura može se shvatiti i kao grupa više Series struktura koje imaju isti indeks. DataFrame se najčešće definira pomoću rječnika koji sadrži liste. Pri tome ključ definira naziv stupca u DataFrameu (vidi primjer 3.2).

¹<http://pandas.pydata.org/pandas-docs/stable/>

■ **Primjer 3.2**

```
import pandas as pd
import numpy as np

data = {'country': ['Italy', 'Spain', 'Greece', 'France', 'Portugal'],
        'population': [59, 47, 11, 68, 10],
        'code': [39, 34, 30, 33, 351]}

countries = pd.DataFrame(data, columns=['country', 'population', 'code'])
print(countries)
```

Učitavanje podataka iz CSV datoteke i osnovne funkcije Dataframea

Vrlo čest problem je učitavanje podataka iz nekog vanjskog izvora. Ako je skup podataka zapisan u CSV datoteci (engl. *comma separated values*), podaci se mogu učitati u DataFrame pomoću naredbe `.read_csv`. Nakon učitavanja će numeričke varijable biti predstavljene ili cjelobrojnim vrijednostima (int64) ili decimalnim vrijednostima (float64). Vrijednosti kategoričkih varijabli obično su predstavljene tekstrom u CSV datotekama pa će nakon učitavanja biti predstavljene tipom object kao što prikazuje primjer 3.3. za primjer podatkovnog skupa `data_C02_emissions.csv`. Kategoričke veličine potrebno je eksplisitno konvertirati u kategorički tip kao što je prikazano u primjeru 3.4.

■ **Primjer 3.3**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2212 entries, 0 to 2211
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   Make             2212 non-null   object  
 1   Model            2212 non-null   object  
 2   Vehicle Class   2212 non-null   object  
 3   Engine Size (L) 2212 non-null   float64 
 4   Cylinders       2212 non-null   int64  
 5   Transmission    2212 non-null   object  
 6   Fuel Type       2212 non-null   object  
 7   Fuel Consumption City (L/100km) 2212 non-null   float64 
 8   Fuel Consumption Hwy (L/100km)  2212 non-null   float64 
 9   Fuel Consumption Comb (L/100km) 2212 non-null   float64 
 10  Fuel Consumption Comb (mpg)     2212 non-null   int64  
 11  CO2 Emissions (g/km)          2212 non-null   int64  
dtypes: float64(4), int64(3), object(5)
memory usage: 207.5+ KB
```

Jednom kada su podaci učitani, na raspolaganju su različite metode DataFramea s kojima je moguće dobiti informacije o podacima kako prikazuje primjer 3.4.:

- `.info()` – daje osnovne informacije o DataFrameu,
- `.head(n)` – vraća prvih n zapisa u DataFrameu,
- `.tail(n)` - vraća zadnjih n zapisa u DataFrameu,

- `.describe()` – vraća statistiku za svaku veličinu u DataFrameu.

Postoje gotove matematičke funkcije u obliku metoda:

- `.mean()` – srednja vrijednost svakog stupca (veličine),
- `.median()` – medijan vrijednost svakog stupca (veličine),
- `.max()` - maksimalna vrijednost svakog stupca (veličine),
- `.min()` – minimalna vrijednost svakog stupca (veličine)
- `.sort_values(by=['col'])` – sortiranje DataFrame prema željenom stupcu col

■ **Primjer 3.4**

```
import pandas as pd

data = pd.read_csv('data_CO2_emission.csv')

#konvertiranje kategorickih velicina u tip category

print(len(data))
print(data)

print(data.head(5))
print(data.tail(3))
print(data.info())
print(data.describe())

print(data.max())
print(data.min())
```

■ **Indeksiranje DataFramea**

Izdvajanje pojedinog stupca iz DataFramea moguće je izvesti pomoću zagrada [] ili točke na način da se navede naziv stupca kako je prikazano u primjeru 3.5. Ako se želi izdvojiti više stupaca, potrebno je unutar zagrada predati listu koja sadrži nazine stupaca. Metoda `.iloc` izdvaja redove s određenim indeksima. Moguće je postaviti logičke uvjete na pojedine stupce – rezultat je DataFrame koji ima vrijednosti True ili False te se na taj način mogu izdvojiti samo redovi koji zadovoljavaju postavljeni uvjet.

■ **Primjer 3.5**

```
import pandas as pd
import numpy as np

data = pd.read_csv('data_CO2_emission.csv')

#izdvajanje pojedinog stupca
print(data['Cylinders'])
print(data.Cylinders)

#izdvajanje vise stupaca
print(data[['Model', 'Cylinders']])

#izdvajanje redaka koristenjem iloc metode
print(data.iloc[2:6, 2:7])
print(data.iloc[:, 2:5])
print(data.iloc[:, [0,4,7]])
```

```
#logicki uvjeti na pojedine stupce
print(data.Cylinders > 6)
print(data[data.Cylinders > 6])
print((data['Cylinders'] == 4) & (data['Engine Size (L)'] > 2.4).Model
      )

#dodavanje novih stupaca
data['jedinice'] = np.ones(len(data))
data['large'] = (data['Cylinders'] > 10)
```

Grupiranje DataFramea

Grupiranje podataka u DataFrameu je brz način dobivanja karakterističnih vrijednosti raspoloživih podataka po grupama. Primjer grupiranja podataka dan je u primjeru 3.6.

■ Primjer 3.6

```
import pandas as pd
import numpy as np

data = pd.read_csv('data_CO2_emission.csv')

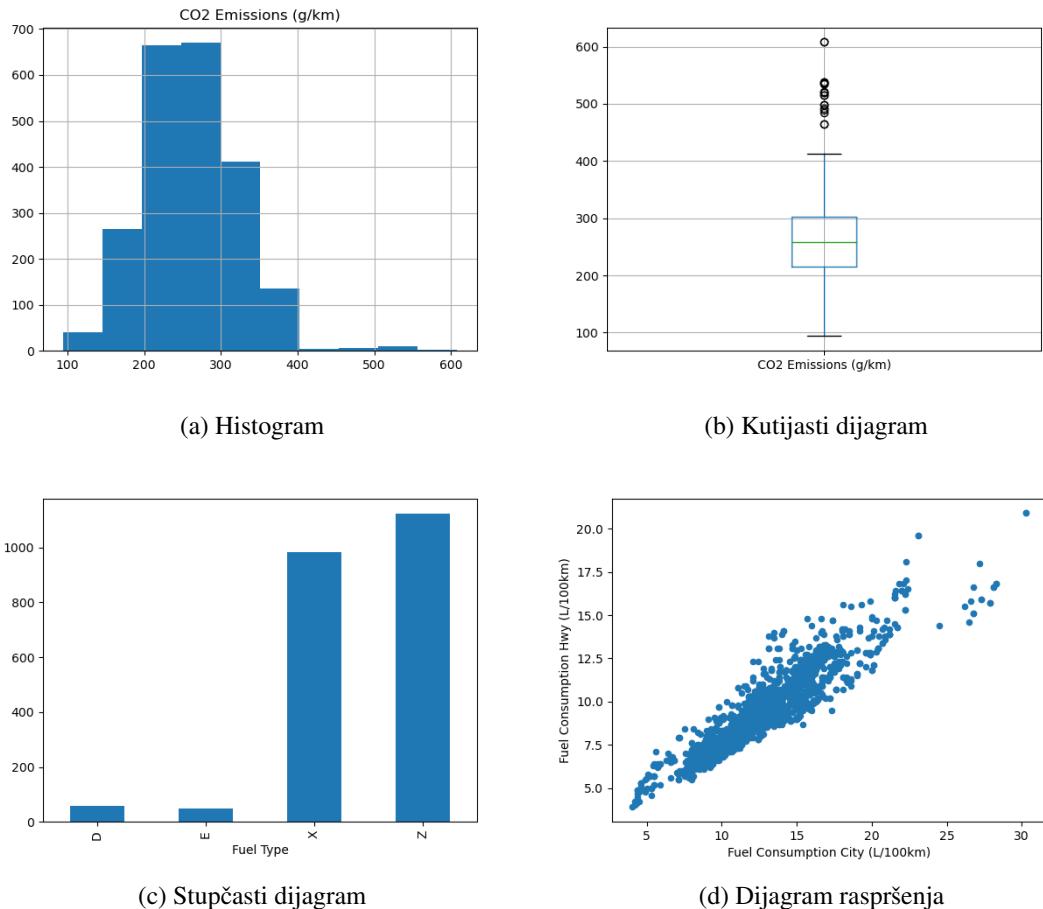
new_data = data.groupby('Cylinders')
print(new_data.count())
print(new_data.size())
print(new_data.sum())
print(new_data.mean())
```

3.2.3 Predobrada podataka i eksplorativna analiza podataka

Osnovna predobrada podataka uključuje čišćenje podatkovog skupa budući da su u skupu moguće izostale vrijednosti, duplicitne vrijednosti, ali i grube mjerne pogreške. Nakon osnovne predobrade podataka moguće je pristupiti analizi podataka. Eksplorativna analiza podataka je pristup analizi podataka na način da se sumiraju određene karakteristike podataka te se često prikazuju grafički. Ovim postupkom se dobiva uvid u karakteristike podataka, u odnose među veličinama, eventualne probleme vezane za prikupljanje podataka i sl. Eksplorativna analiza podataka je važan korak prilikom rješavanja problema strojnog učenja jer na temelju nje se može usmjeriti postupak učenja te odlučiti koje hipoteze bi bilo smisleno istraživati. Grafički prikazi koji se mogu koristiti su različiti, ali najčešće se koriste: histogram, kutijasti dijagram (engl. *boxplot*), stupčasti dijagram (engl. *barplot*), dijagram raspršenja (engl. *scatterplot*). Primjeri dijagrama dani su na slici 3.1.

Izostale i duplicitne vrijednosti

Često se u podacima pojavljuju izostale vrijednosti zbog pogreške u prikupljanju podataka, prilikom upisa podataka i sl. Ovakve vrijednosti mogu se pronaći pomoću metode `.isnull()` za svaki stupac u DataFrameu. U najjednostavnijem slučaju se takva mjerena mogu obrisati iz DataFramea pomoću metode `.dropna()` kako je ilustrirano u primjeru 3.7. Duplicirane vrijednosti su redovi u DataFrameu koji imaju potpuno jednake sve vrijednosti. Njih je moguće obrisati pomoću metode `.drop_duplicates()`.



Slika 3.1: Primjeri dijagonala

■ Primjer 3.7

```
import pandas as pd

data = pd.read_csv('data_CO2_emission.csv')

#provjera koliko je izostalih vrijednosti po svakom stupcu DataFramea
print(data.isnull().sum())

#brisanje redova gdje barem vrijednost jedne velicine nedostaje
data.dropna(axis=0)

#brisanje stupaca gdje barem jedna vrijednost nedostaje
data.dropna(axis=1)

#brisanje duplicitiranih redova
data.drop_duplicates()

#kada se obrisu pojedini redovi potrebno je resetirati indekse retka
data = data.reset_index(drop=True)
```

Analiza i vizualizacija podataka

Najjednostavniji način analize podataka je promatranje svake veličine zasebno. Pri tome se najčešće podaci o varijabli prikazuju u obliku histograma ili kutijastog dijagrama kao što je prikazano u primjeru 3.8. pri čemu se dobiva uvid u distribuciju podataka. Npr. na ovakvim prikazima je lako uočiti stršeće vrijednosti (engl. *outliers*) odnosno vrijednosti koje značajno odudaraju od ostatka podataka.

■ Primjer 3.8

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('data_CO2_emission.csv')

plt.figure()
data['Fuel Consumption City (L/100km)'].plot(kind='hist', bins = 20)
plt.figure()
data['Fuel Consumption City (L/100km)'].plot(kind='box')
plt.show()
```

Prikaz distribucije neke veličine (npr. pomoću kutijastog dijagrama) s obzirom na neku kategoričku veličinu moguće je postići korištenjem metode groupby ili korištenjem argumenta by prilikom crtanja.

■ Primjer 3.9

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('data_CO2_emission.csv')

grouped = data.groupby('Cylinders')
grouped.boxplot(column=['CO2 Emissions (g/km)'])

data.boxplot(column=['CO2 Emissions (g/km)', 'by='Cylinders'])
plt.show()
```

Za međusobni odnos dviju veličine moguće je koristiti dijagram raspršenja. Ako se želi uključiti i treća veličina, onda je to moguće preko boje ili veličine pojedine točke u dijagramu raspršenja kao u primjeru 3.10.

■ Primjer 3.10

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('data_CO2_emission.csv')

data.plot.scatter(x='Fuel Consumption City (L/100km)',
```

Poglavlje 3. Upoznavanje s Pandas bibliotekom. Predobrada podataka i eksplorativna analiza podataka

```
y='Fuel Consumption Hwy (L/100km)',  
c='Engine Size (L)', cmap="hot", s=50)  
plt.show()
```

Korelacija se koristi kako bi se sumirala jačinu i smjer linearnog odnosa između dvije numeričke veličine. DataFrame metoda `.corr()` računa korelaciju između numeričkih veličina DataFramea (vidi primjer 3.11).

■ Primjer 3.11

```
import pandas as pd  
  
data = pd.read_csv('data_CO2_emission.csv')  
  
print(data.corr(numeric_only=True))
```

3.3 Priprema za vježbu

1. Proučite poglavlje 3.2.

3.4 Rad na vježbi

1. Isprobajte Python primjere iz poglavlja 3.2 u Visual Studio Code IDE. Razmislite o svakoj liniji programskega koda i što je njen rezultat.
2. Riješite dane zadatke.

Zadatak 3.4.1 Skripta `zadatak_1.py` učitava podatkovni skup iz `data_CO2_emission.csv`. Dodajte programski kod u skriptu pomoću kojeg možete odgovoriti na sljedeća pitanja:

- a) Koliko mjerenja sadrži DataFrame? Kojeg je tipa svaka veličina? Postoje li izostale ili duplicitane vrijednosti? Obrišite ih ako postoje. Kategoričke veličine konvertirajte u tip `category`.
- b) Koja tri automobila ima najveću odnosno najmanju gradsku potrošnju? Ispišite u terminal: ime proizvođača, model vozila i kolika je gradska potrošnja.
- c) Koliko vozila ima veličinu motora između 2.5 i 3.5 L? Kolika je prosječna C02 emisija plinova za ova vozila?
- d) Koliko mjerenja se odnosi na vozila proizvođača Audi? Kolika je prosječna emisija C02 plinova automobila proizvođača Audi koji imaju 4 cilindara?
- e) Koliko je vozila s 4,6,8... cilindara? Kolika je prosječna emisija C02 plinova s obzirom na broj cilindara?
- f) Kolika je prosječna gradska potrošnja u slučaju vozila koja koriste dizel, a kolika za vozila koja koriste regularni benzin? Koliko iznose medijalne vrijednosti?
- g) Koje vozilo s 4 cilindra koje koristi dizelski motor ima najveću gradsku potrošnju goriva?
- h) Koliko ima vozila ima ručni tip mjenjača (bez obzira na broj brzina)?
- i) Izračunajte korelaciju između numeričkih veličina. Komentirajte dobiveni rezultat.

Zadatak 3.4.2 Napišite programski kod koji će prikazati sljedeće vizualizacije:

- a) Pomoću histograma prikažite emisiju CO₂ plinova. Komentirajte dobiveni prikaz.
- b) Pomoću dijagrama raspršenja prikažite odnos između gradske potrošnje goriva i emisije CO₂ plinova. Komentirajte dobiveni prikaz. Kako biste bolje razumjeli odnose između veličina, obojite točkice na dijagramu raspršenja s obzirom na tip goriva.
- c) Pomoću kutijastog dijagrama prikažite razdiobu izvengradske potrošnje s obzirom na tip goriva. Primjećujete li grubu mjernu pogrešku u podacima?
- d) Pomoću stupčastog dijagrama prikažite broj vozila po tipu goriva. Koristite metodu groupby.
- e) Pomoću stupčastog grafa prikažite na istoj slici prosječnu CO₂ emisiju vozila s obzirom na broj cilindara.

3.5 Izvještaj s vježbe

Kao izvještaj s vježbe prihvaća se web link na repozitorij pod nazivom OSU_LV.

Literatura

[1] <https://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64>

Dodatak

Opis veličina podatkovnog skupa koji je pohranjen u data_CO₂_emission.csv.

Naziv veličine	Opis
Make	Proizvođač vozila (Audi, BMW, Ford...)
Model	Model vozila (A5 quattro, M5 Sedan, ...)
Vehicle Class	Klasa vozila (SUV: Small, Two-seater, ...)
Engine Size (L)	Zapremnina motora u litrama
Cylinders	Broj cilindara (4,5,6,...)
Transmission	Tip mjenjača
Fuel Type	Tip goriva
Fuel Consumption City (L/100km)	Gradska potrošnja u litrama na 100 kilometara
Fuel Consumption Hwy (L/100km)	Izvengradska potrošnja u litrama na 100 kilometara
Fuel Consumption Comb (L/100km)	Kombinirana potrošnja (55% gradska vožnja, 45% izvengradska vožnja) u litrama na 100 kilometara
Fuel Consumption Comb (mpg)	Kombinirana potrošnja (55% gradska vožnja, 45% izvengradska vožnja) u miljama po galonu
CO ₂ Emissions (g/km)	Emisija CO ₂ plinova u gramima po kilometru za kombiniranu vožnju

Tip mjenjača	Opis
A	Automatic
AM	Automated manual
AS	Automatic with select shift
AV	Continuously variable
M	Manual
3 - 10	Number of gears

Tip goriva	Opis
X	Regular gasoline
Z	Premium gasoline
D	Diesel
E	Ethanol (E85)
N	Natural gas