VJEŽBA 3: RAD S PANDAS PYTHON BIBLIOTEKOM. EKSPLORATIVNA ANALIZA PODATAKA.

I. Cilj vježbe:

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

II. Opis vježbe:

U vježbi se studenti upoznaju s Pandas bibliotekom za programski jezik Python. Ova biblioteka omogućava relativno laganu manipulaciju 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.

II.1. Pandas biblioteka

Pandas je *open source* Python biblioteka koja značajno olakšava učitavanje i analizu 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.

Struktura Series

Series je jednodimenzionalni objekt sličan polju pri čemu je svakom elementu polja pridružen indeks (defaultne vrijednosti 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.)

Primjer 3.1.

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

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])
```

Struktura DataFrame

Struktura DataFrame je dvodimenzionalna označena struktura nalik tablici gdje su u kolonama pohranjeni podaci istog tipa (npr. liste, dictionary, numpy polja i sl.). DataFrame struktura može se shvatiti i kao grupa više Series struktura koje imaju isti indeks. DataFrame se najčešće pomoću dictionary objekta koji sadrži liste. Pri tome key definira naziv kolone u DataFrameu.

Primjer 3.2.

```
data = {'year': [2010, 2011, 2012, 2011, 2012, 2010, 2011, 2012],
    'team': ['Bears', 'Bears', 'Bears', 'Packers', 'Packers', 'Lions', 'Lions', 'Lions'],
    'wins': [11, 8, 10, 15, 11, 6, 10, 4],
```

```
    'losses': [5, 8, 6, 1, 5, 10, 6, 12]}

football = pd.DataFrame(data, columns=['year', 'team', 'wins', 'losses'])
print(football)
```

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:

Primjer 3.3.

```
mtcars = pd.read_csv('mtcars.csv')
print(len(mtcars))
print(mtcars)
```

Na raspolaganju su različite funkcije za rad s DataFrameovima s kojima je moguće dobiti neke vrijednosti raspoloživih podataka (info, head, tail, describe):

Primjer 3.4.

```
print(mtcars.head(5))
print(mtcars.tail(3))
print(mtcars.info())
print(mtcars.describe())
```

Dodavanje novog stupca, izdvajanje pojedinog stupca ili određenih redova iz DataFramea moguće je izvesti pomoću zagrada []:

Primjer 3.5.

```
print(mtcars['car'])
print(mtcars.cyl)

print(mtcars.cyl > 6)
print(mtcars[mtcars.cyl > 6])
print(mtcars[(mtcars.cyl == 4) & (mtcars.hp > 100)].car)
print(mtcars[['car','cyl']])

print(mtcars.cyl[2:4])
print(mtcars[5:12])
print(mtcars.mpg[3:5])

mtcars['jedinice'] = np.ones(len(mtcars))
mtcars['heavy'] = mtcars.wt > 4.5
print(mtcars[['car','heavy']])
print(mtcars.query('cyl == [4,6]').car)

print(mtcars.iloc[1:3, 5:10])
print(mtcars.iloc[:, 3:5])
print(mtcars.iloc[:, [0,4,7]])
print(mtcars.iloc[[1,29], :])
```

Grupiranje podataka u DataFrameu je brz način dobivanja karakterističnih vrijednosti raspoloživih podataka.

```
new_mtcars = mtcars.groupby('cyl')
print(new_mtcars.count())
print(new_mtcars.sum())
print(new_mtcars.mean())
```

II.2. Eksplorativna analiza podataka

Eksplorativna analiza podataka je pristup analizi podataka na način da se sumiraju određene karakteristike podataka te prikažu 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: box plot, histogram, barplot, scatter plot i density plots. Detaljan opis korištenja grafičke biblioteke matplotlib za ove potrebe može se pronaći na adresi https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html . Isprobajte nekoliko primjera.

III. Priprema za vježbu:

1.

IV. Rad na vježbi:

1. Isprobajte Python primjere iz II. Opis vježbe u Spyder IDE. Razmislite o svakoj liniji programskog koda i što je njen rezultat. Pogledajte u Variable Explorer Spyder IDE sve varijable koje ste kreirali nakon pokretanja svakog primjera.
2. Klonirajte vaš repozitorij `rusu_lv_2019_20` na računalo pomoću gitbash. Zatim povucite moguće promjene iz izvornog repozitorija pomoću naredbi:

```
git remote add upstream https://gitlab.com/rgrbic/rusu_lv_2019_2020
git fetch upstream
git merge upstream/master
```

3. Riješite dane zadatke, pri čemu Python skripte trebaju imati naziv `zad_x.py` (gdje je `x` broj zadatka) i trebaju biti pohranjene u direktorij `rusu_lv_2019_20/LV3/solutions/`. Svaki zadatak rješavajte u zasebnoj *git* grani koju spojite s glavnom granom kada riješite pojedini zadatak. Pohranite skripte u lokalnu *git* bazu kao i u `rusu_lv_2019_20` repozitorij na vašem korisničkom računu. Svaki puta kada načinite promjene koje se spremaju u *git* sustav napišite i odgovarajuću poruku prilikom izvršavanja `commit` naredbe.
4. Nadopunite postojeću tekstualnu datoteku `rusu_lv_2019_20/LV3/Readme.md` s kratkim opisom vježbe i kratkim opisom rješenja vježbe te pohranite promjene u lokalnu bazu. Na kraju pohranite promjene u udaljeni repozitorij.

Zadatak 1

Za `mtcars` skup podataka (nalazi se `rusu_lv_2019_20/LV3/resources`) napišite programski kod koji će odgovoriti na sljedeća pitanja:

1. Kojih 5 automobila ima najveću potrošnju? (koristite funkciju sort)
2. Koja tri automobila s 8 cilindara imaju najmanju potrošnju?
3. Kolika je srednja potrošnja automobila sa 6 cilindara?
4. Kolika je srednja potrošnja automobila s 4 cilindra mase između 2000 i 2200 lbs?
5. Koliko je automobila s ručnim, a koliko s automatskim mjenjačem u ovom skupu podataka?
6. Koliko je automobila s automatskim mjenjačem i snagom preko 100 konjskih snaga?
7. Kolika je masa svakog automobila u kilogramima?

Zadatak 2

Napišite programski kod koji će iscrtati sljedeće slike za `mtcars` skup podataka:

1. Pomoću barplot-a prikažite na istoj slici potrošnju automobila s 4, 6 i 8 cilindara.
2. Pomoću boxplot-a prikažite na istoj slici distribuciju težine automobila s 4, 6 i 8 cilindara.
3. Pomoću odgovarajućeg grafa pokušajte odgovoriti na pitanje imaju li automobili s ručnim mjenjačem veću potrošnju od automobila s automatskim mjenjačem?
4. Prikažite na istoj slici odnos ubrzanja i snage automobila za automobile s ručnim odnosno automatskim mjenjačem.

Zadatak 3

Na stranici http://iszz.azo.hr/iskzl/exc.htm moguće je dohvatiti podatke o kvaliteti zraka za Republiku Hrvatsku. Podaci se mogu preuzeti korištenjem RESTfull servisa u XML ili JSON obliku. U `rusu_lv_2019_20/LV3/resources/` skriptu koja dohvaća podatke te ih pohranjuje u odgovarajući `DataFrame`. Prepravite/nadopunite skriptu s programskim kodom kako bi dobili sljedeće rezultate:

1. Dohvaćanje mjerenja dnevne koncentracije lebdećih čestica PM_{10} za 2017. godinu za grad Osijek.
2. Ispis tri datuma u godini kada je koncentracija PM_{10} bila najveća.
3. Pomoću barplot prikažite ukupni broj izostalih vrijednosti tijekom svakog mjeseca.
4. Pomoću boxplot usporedite PM_{10} koncentraciju tijekom jednog zimskog i jednog ljetnog mjeseca.
5. Usporedbu distribucije PM_{10} čestica tijekom radnih dana s distribucijom čestica tijekom vikenda.

V. Izvještaj s vježbe

Kao izvještaj s vježbe prihvaća se web link na repozitorij pod nazivom `rusu_lv_2019_20` koji sadrži rješenja unutar direktorija `rusu_lv_2019_20/LV3/solutions/`.