VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE TEHNOLOGIJE

Data mining

Semestralni projekat

Analiziranje podataka o konzumiranju kafe i kodiranja programa u Python-u

Predmetni nastavnici:

dr Stefana Janićijević

Studenti:

Anja Gruber 361/19

Datum predaje 29.09.2020.

Beograd,

Septembar 2020.

Sadržaj

Uvod

U ovom radu su preuzeti podaci sa sajta Kaggle u svrhu kreiranja programa u Python-u, čija je glavna svrha da predstavi korišćenje data mining-a kao alata za dobijanje značajnih informacija. Prikazani podaci su korigovani tako da se sa njima može upravljati i proračunavati, u smislu što su svi tekstualni podaci modifikovani u numeričku formu, a prazne vrednosti uklonjene. U nastavku su prikazani grafikoni koji služe za lakše razumevanje podataka, zatim odrađene predikcije algoritmima u programu.

Pitanja i problemi kojima se program bavi su: da li različite vrste kafe daju različiti uticaj na uspešnost programiranja, da li žene ili muškarci količinski više konzumiraju kafu, da li je uticaj kafe isti na muškarce i žene, dnevni unos kafe osobe koja programira.

Na samom kraju je izveden zaključak i prikazana literatura.

Ključne reči

Data mining, python, analiza, podaci, algoritam, kafa, programiranje, grafikon.

Prikaz i čišćenje podataka

U tabeli podataka se nalazi 100 uzoraka podataka, ali radi preglednosti prikazano je prvih nekoliko (slika 1). U pitanju su podaci koji vezuju uspeh programiranja sa konzumiranjem kafe.

```
print("number of data: "+ str(len(df.index)))
```

```
number of data: 100
```

	CoffeeType	Gender	CoffeeCupsPerDay	CodingHours	CoffeeTime	CodingWithoutCoffee	CoffeeSolveBugs	AgeRange	Country
0	Caffè latte	Female	2	8	Before coding	Yes	Sometimes	18 to 29	Lebanon
1	Americano	Female	2	3	Before coding	Yes	Yes	30 to 39	Lebanon
2	Nescafe	Female	3	5	While coding	No	Yes	18 to 29	Lebanon
3	Nescafe	Male	2	8	Before coding	No	Yes	NaN	Lebanon
4	Turkish	Male	3	10	While coding	Sometimes	No	18 to 29	Lebanon

Slika 1. Prikaz obima uzorka i prvih nekoliko redova iz tabele podataka

Pojašnjenje šta predstavljaju kolone u tabeli:

1) CoffeeType – vrsta kafe,
2) Gender – pol,
3) CoffeeCupsPerDay – broj dnevno popijenih šoljica kafe,
4) CodingHours – broj sati dnevno provedenih u programiranju,
5) CoffeeTime – vreme u toku dana kada osoba konzumira kafu u odnosu na radno vreme,
6) CodingWithoutCoffee – da li osoba programira bez konzumiranja kafe,
7) CoffeeSolveBugs – da li konzumiranje kafe pomaže prilokom rešavanja bagova,
8) AgeRange – u koji raspon godina pripada osoba,
9) Country – država u kojoj su uzeti uzorci podataka.

Podaci su uzeti samo iz države Lebanon gde je i najbolja proizvodnja kafe, elem za analiziranje i dalji rad sa podacima, jedna lokacija nije relevantna i kao takva neće biti uzeta u obzir (Slika 2).

```
df.drop('Country', axis=1, inplace=True)
```

Slika 2.Uklanjanje kolone *Country* (država)

Podaci koji su numeričkog tipa su ,,CoffeeCupsPerDay'' i ,,CodingHours''. Dok su ostali podaci tipa objekat, odnosno kategorički (slika 3).

```
df.dtypes
```

```
CoffeeType            object
Gender                object
CoffeeCupsPerDay       int64
CodingHours            int64
CoffeeTime            object
CodingWithoutCoffee   object
CoffeeSolveBugs       object
AgeRange              object
Country               object
dtype: object
```

Slika 3.Prikaz tipova podataka iz tabele za svaku kolonu

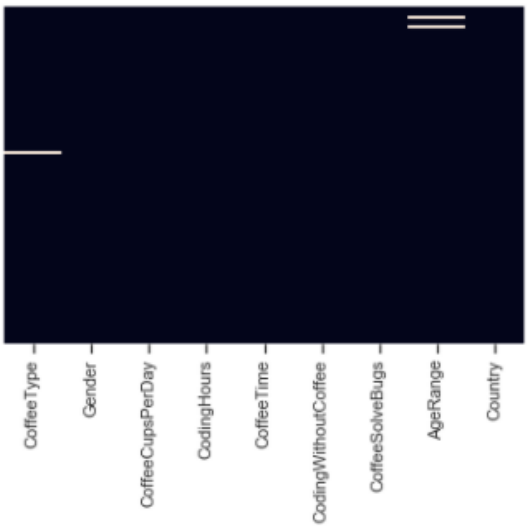
Koristeći metodu describe() (slika 4.), vidi se da postoje prazne vrednosti za neke kolone za neke uzorke, konkretnije za ,,CoffeeType'' i za ,,AgeRange''. Za sve kolone se vidi ukupan broj jedinstvenih vrednosti. Kod numeričkih podataka prikazana je minimalna, maksimalna i srednja vrednost, kao i standardna devijacija. Dok je kod kategoričkih prikazano koja vrednost se najčešće ponavlja i koliko puta. Iz tih podataka se može zaključiti da je veći uzorak uzet nad muškim polom, da su ljudi podeljeni kad je u pitanju konzumiranje kafe i kodiranje, ali da oni koji to rade najčešće konzumiraju kafu tokom kodiranja i da je to u proseku približno tri šolje kafe i to najčešće Nescafe. Takođe, ljudi su podeljenih mišljenja da li kafa pomaže prilikom rešavanja bagova u programu.

```
df.describe(include="all")
```

	CoffeeType	Gender	CoffeeCupsPerDay	CodingHours	CoffeeTime	CodingWithoutCoffee	CoffeeSolveBugs	AgeRange	Country
count	99	100	100.000000	100.000000	100	100	100	98	100
unique	8	2	NaN	NaN	7	3	3	5	1
top	Nescafe	Male	NaN	NaN	While coding	Sometimes	Sometimes	18 to 29	Lebanon
freq	32	74	NaN	NaN	61	51	43	60	100
mean	NaN	NaN	2.890000	6.410000	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	1.613673	2.644205	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN	1.000000	1.000000	NaN	NaN	NaN	NaN	NaN
25%	NaN	NaN	2.000000	4.000000	NaN	NaN	NaN	NaN	NaN
50%	NaN	NaN	2.500000	7.000000	NaN	NaN	NaN	NaN	NaN
75%	NaN	NaN	4.000000	8.000000	NaN	NaN	NaN	NaN	NaN
max	NaN	NaN	8.000000	10.000000	NaN	NaN	NaN	NaN	NaN

Slika 4.Prikaz osnovnih statističkih podataka pomoću metode describe()

Ono što bi bilo dalje poželjno uraditi jeste uraditi encoding kategoričkih podataka u numeričku formu, iz razloga što je za kreiranje modela za predikciju tako poželjno. Pre toga se treba pozabaviti praznim vrednostima i zameniti ih ili ih ukloniti. Shodno malom broju uzorka, prazne vrednosti će biti zamenjene onim koje se najčešće pojavljuju za tu kolonu (Slika 5 i 6).



Slika 5.Prikaz praznih vrednosti

```
df['CoffeeType'] = df['CoffeeType'].fillna(df['CoffeeType'].value_counts().index[0])
```

```
print(df.isnull().values.sum())
```

```
2
```

```
df['AgeRange'] = df['AgeRange'].fillna(df['AgeRange'].value_counts().index[0])
```

```
print(df.isnull().values.sum())
```

```
0
```

```
print("number of data: "+ str(len(df.index)))
```

```
number of data: 100
```

Sika 6.Prikaz zamene praznih vrednosti

Kako bi se kategorički podaci prikazali u numeričkoj formi, tip podataka treba da bude category, a ne object. (slika 7). Na slici 8 su prikazani kodovi za kategoričke podatke, a na slici 9 prvih nekoliko redova tabele nakon zamene podataka.

```
df["CoffeeTime"] = df["CoffeeTime"].astype('category')
df["CodingWithoutCoffee"] = df["CodingWithoutCoffee"].astype('category')
df["CoffeeSolveBugs"] = df["CoffeeSolveBugs"].astype('category')
df["CoffeeType"] = df["CoffeeType"].astype('category')
df.dtypes
```

```
CoffeeType             category
Gender                   object
CoffeeCupsPerDay          int64
CodingHours               int64
CoffeeTime             category
CodingWithoutCoffee    category
CoffeeSolveBugs        category
AgeRange                 object
dtype: object
```

Slika 7. Prikaz tipova podataka

```
CoffeeType

2     Caffè latte
1     Americano
6     Nescafe
7     Turkish
0     American Coffee
5     Espresso (Short Black)
3     Cappuccino
4     Double Espresso (Doppio)
```

```
AgeRange

23.5     18 to 29
34.5     30 to 39
44.5     40 to 49
14.5     12 to 17
54.5     50 to 59
```

```
CoffeeTime

2     Before coding
4     While coding
3     Before and while coding
1     In the morning
6     All the time
5     After coding
0     No specific time
```

```
CodingWithoutCoffee

2     Yes
0     No
1     Sometimes
```

```
CoffeeSolveBugs

1     Sometimes
2     Yes
0     No
```

```
Gender

0     Female
1     Male
```

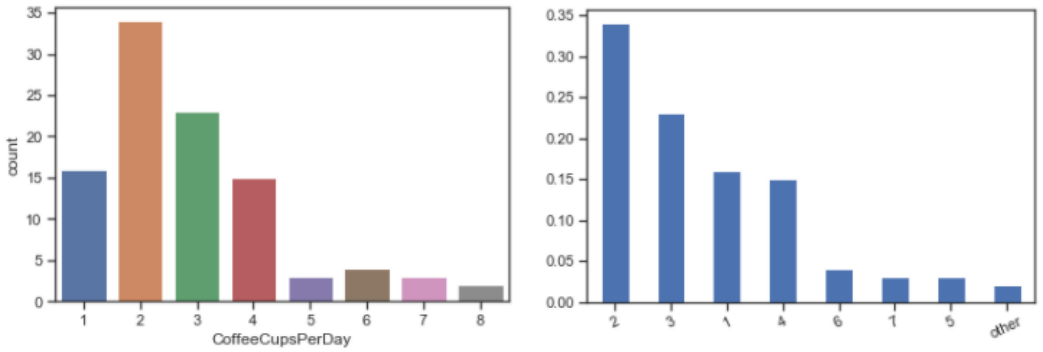
Slika 8. Prikaz kodova kategoričkih u numeričke podatke

	CoffeeCupsPerDay	CodingHours	CoffeeType_code	CodingWithoutCoffee_code	CoffeeSolveBugs_code	Gender_code	Age_mean	CoffeeTime_code
0	2	8	2	2	1	1	23.5	2
1	2	3	1	2	2	1	34.5	2
2	3	5	6	0	2	1	23.5	4
3	2	8	6	0	2	0	23.5	2
4	3	10	7	1	0	0	23.5	4

Slika 9. Prikaz tabele podataka nakon modifikovanja

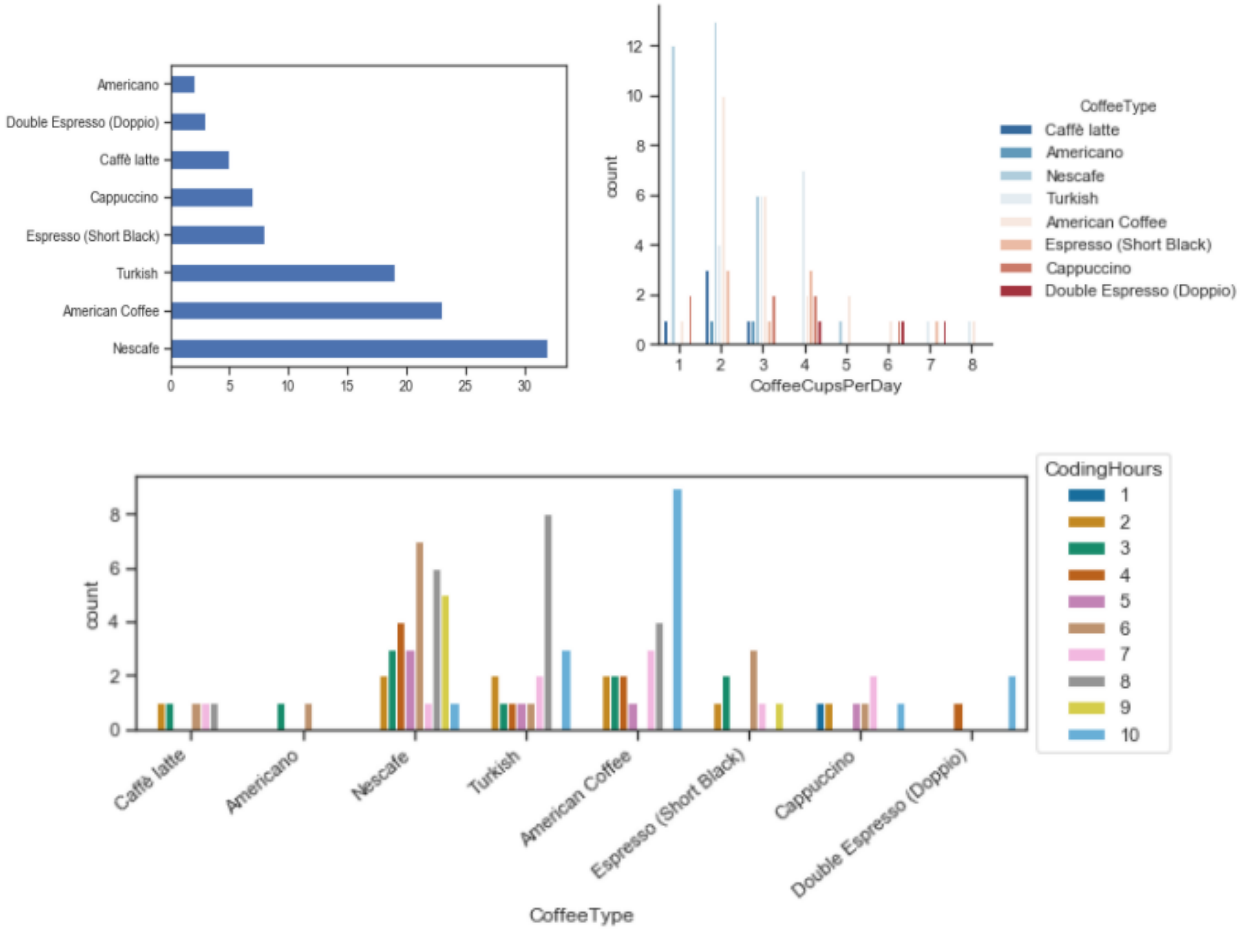
Analiza podataka

Prikazani su podaci o tome koliko uzoraka konzumira koliko šoljica kafe dnevno i koje vrste, kao i koliko vrsta kafe utiče na kvalitet i kvantitet programiranja. Na prvom grafiku su sirovi podaci o zbiru svake vrednosti za broj popijenih šoljica na dnevnom nivou, dok su na drugom u procentualnom obliku hijerarhijski prikazani. Najčešći slučajevi su dve, zatim tri šoljice kafe.



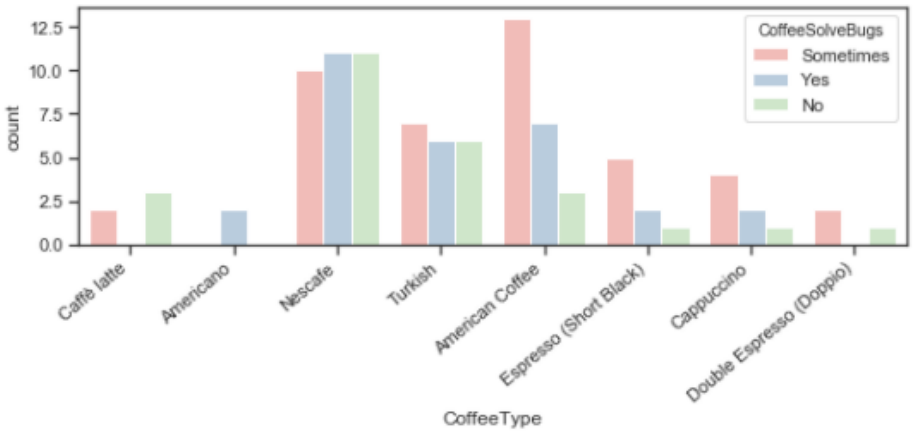
Grafik 1 i 2

Na trećem i četvrtom grafiku je prikazano koja vrsta kafe se koliko konzumira, Nescafe, američka i turska kafa su najčešći izbor, dok se na petom grafiku vidi i da ljudi koji piju te kafe najviše sati dnevno kodiraju.



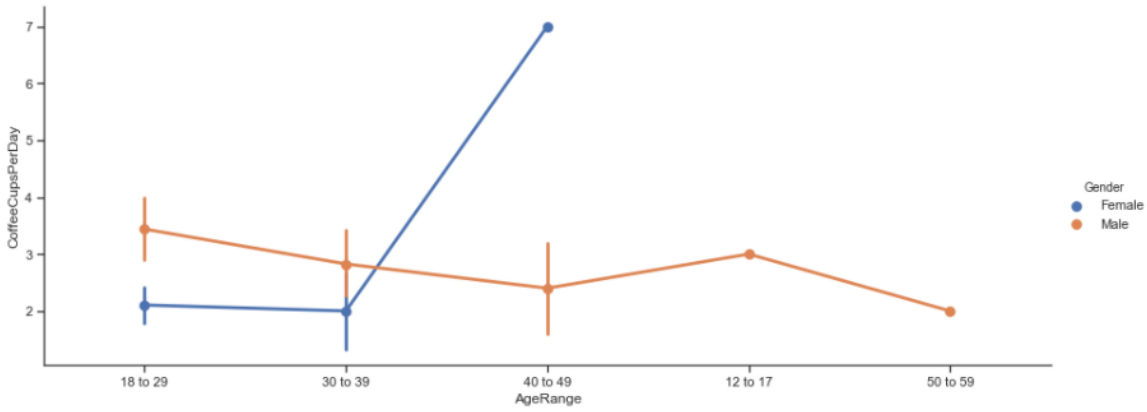
Grafik 3, 4 i 5

Grafik 6 prikazuje da vrsta kafe ne utiče značajno na podeljeno mišljenje ljudi o tome koliko kafa utiče na rešavanje bagova.



Grafik 6

Sa grafika 7 se vidi da, kad je muški rod u pitanju, najviše soljica kafe dnevno popiju osobe koje imaju od 18 do 29 godina, dok je kod žena od 40 do 49.



Grafik 7

Sa grafika 8 se vidi da muškarci i žene koje konzumiraju kafu dok kodiraju imaju prilično sličan stav kad se dovodi u pitanje da li kafa utiče na rešavanje bagova, dok kod onih koji ponekad konzumiraju, ponekad ne, saglasan je stav da ponekad utiče na rešavanje baga, a nesaglasan za sigurno da ili ne. Za osobe koje kažu da ne konzumiraju kafu dok kodiraju su prikazani podaci na osnovu mišljena, a ne iskustva, zbog čega doinira mišljenje kod oba pola da kafa ne utiče na rešavanje bagova.

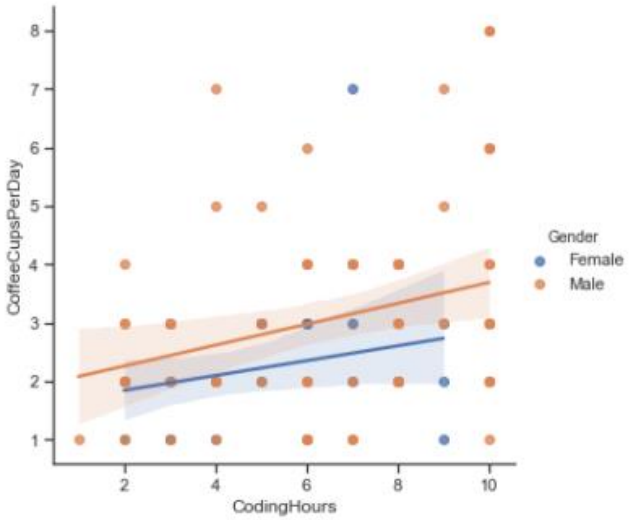
Grafik 8

Na slici 10 je prikazana pivot tabela sa statističkim podacima o popijenim šoljicama kafe grupisanim po polu, kodiranju bez kafe i rešavanju bagova i prema kojoj se vidi ko konzumira najviše kafe u proseku. Osobe ženskog pola koje ne konzumiraju kafu tokom kodiranja, ali imaju mišljenje da konzumiranje kafe utiče na rešavanje bagova, u proseku popiju 4 šojice kafe dnevno, dok muškog pola koji konzumiraju kafu tokom kodiranja i misle da ona ponekad utiče na rešavanje bagova, u proseku popiju oko 6 šoljica kafe dnevno.

			mean	count	min	max
Gender	CodingWithoutCoffee	CoffeeSolveBugs				
Female	No	No	2.000000	1	2	2
		Sometimes	2.000000	1	2	2
		Yes	2.666667	3	2	3
	Sometimes	No	1.500000	2	1	2
		Sometimes	2.333333	9	1	3
		Yes	2.000000	1	2	2
	Yes	No	1.500000	4	1	3
		Sometimes	1.500000	2	1	2
		Yes	4.000000	3	2	7
Male	No	No	3.000000	3	3	3
		Sometimes	5.750000	4	4	8
		Yes	4.714286	7	2	8
	Sometimes	No	3.875000	8	2	6
		Sometimes	2.684211	19	1	4
		Yes	3.500000	12	2	5
	Yes	No	1.555556	9	1	2
		Sometimes	2.500000	8	1	4
		Yes	1.750000	4	1	3

Slika 10. Pivot tabela sa statističkim podacima

Grafik 9 predstavlja scatterplot sa regresionim linijama. Na grafiku 9 je prikazana korelacija između šoljica kafe po danu i sata kodiranja sa polom, gde se vidi da i kod žena i muškaraca sa brojem sati provedenih u programiranju se povećava broj šoljica kafe po danu.



Grafik 9

Na osnovu grafika 9, može se uraditi provera modela čiji bi zadatak bio predvideti dnevni unos kafe osobe koja programira u zavisnosti od pola i broja sati provedenih u programiranju. Na slici 11 se vidi da je kreiran novi dataframe u koji su smešteni samo podaci za predvidjanje.

```
prediction_df = df[['CoffeeCupsPerDay','Gender_code', 'CodingHours']]
prediction_df.head()
```

	CoffeeCupsPerDay	Gender_code	CodingHours
0	2	0	8
1	2	0	3
2	3	0	5
3	2	1	8
4	3	1	10

```
prediction_df.corr()
```

	CoffeeCupsPerDay	Gender_code	CodingHours
CoffeeCupsPerDay	1.000000	0.229175	0.313692
Gender_code	0.229175	1.000000	0.231018
CodingHours	0.313692	0.231018	1.000000

Slika 11.Prikaz novog dataframe-a

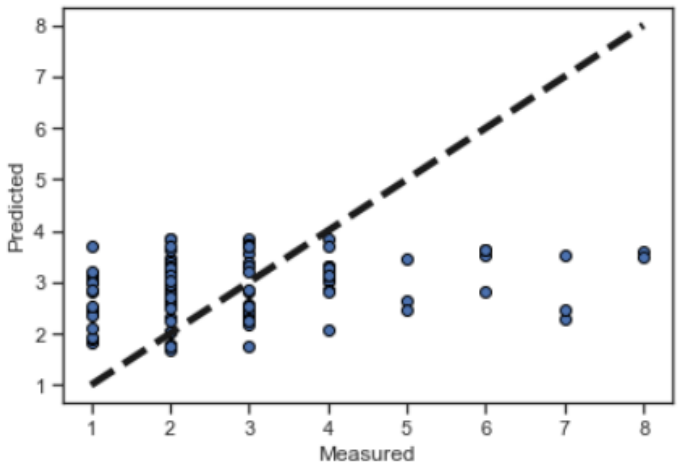
Predikcija

Podaci su podaljeni na podatke za trening i testranje, tako što je 0,25% podataka iz tabele sa podacima odvojeno za testiranje, a ostatak za učenje modela. Koristeći logističku regresiju kao model za predviđanje dobija se tačnost predviđanja od 32% (slika 12). Odnos stvarnih i predviđenih vrednosti dobijen pomoću cross val predict, prikazan je na grafiku 10.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

y_pred =classifier.predict(X_test)

accuracy_score(y_test, y_pred)*100
```

```
32.0
```

Slika 12.Prikaz predikcije



Grafik 10

Kada se za predikciju koristi cross validacija i svm model, dobija se tačnost predviđanja od 34% (slika 13).

```
#Accuracy of Model with Cross Validation
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn import svm
from sklearn.model_selection import cross_val_score
# retrieve the array
data = prediction_df.values
# split into inpiut and output elements
X, y = data[:,1:3], data[:, 0]
model = svm.SVC()
accuracy = cross_val_score(model, X, y, scoring='accuracy', cv = 2)
print(accuracy)
#get the mean of each fold
print("Accuracy of Model with Cross Validation is:",accuracy.mean() * 100)

model.fit(X, y)
print('Broj kafa: ', model.predict([[1, 6]]))
```

```
[0.34 0.34]
Accuracy of Model with Cross Validation is: 34.0
Broj kafa:  [2]
```

Slika 13. Prikaz predikcije

Na slici 14 je prikazana tačnost predviđanja za logistic regression model, K neighbors classifier, decision tree classifier, gaussian NB i SVC.

```
[0.32 0.34]
Accuracy of LR Model with Cross Validation is: 33.0
[0.26 0.32]
Accuracy of KNN Model with Cross Validation is: 29.000000000000004
[0.34 0.34]
Accuracy of CART Model with Cross Validation is: 34.0
[0.18 0.24]
Accuracy of NB Model with Cross Validation is: 21.0
[0.34 0.34]
Accuracy of SVM Model with Cross Validation is: 34.0
```

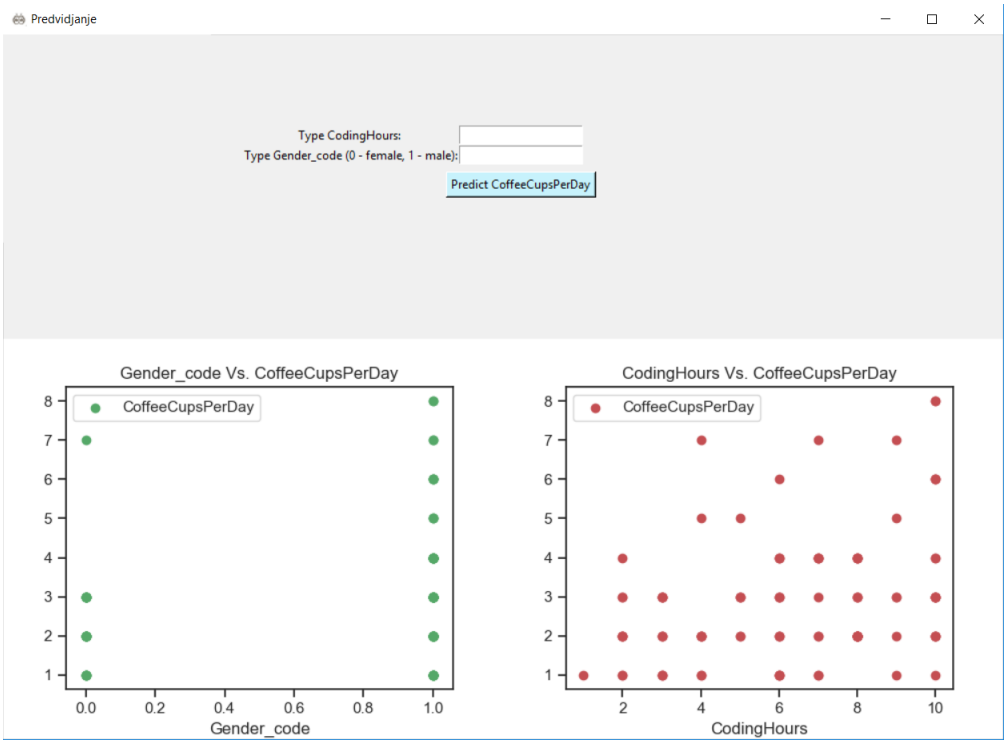
Slika 14.Prikaz tačnosti modela

Zaključak

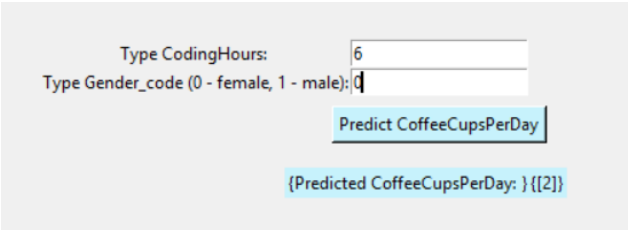
Analizirani podaci o vezi između konzumiranja kafe i programiranja daju sledeće zaključke:

1) Muškarci piju više kafe dnevno nego žene.
2) Najviše se pije Nescafe vrsta kafe.
3) Jedna osoba u proseku dnevno popije 2 do 3 šoljice kafe.
4) Kad je muški rod u pitanju, najviše soljica kafe dnevno popiju osobe koje imaju od 18 do 29 godina, dok je kod žena od 40 do 49.
5) Osobe koje konzumiraju kafu dok kodiraju kažu da kafa pozitivno utiče na rešavanje bagova.
6) Sa brojem sati provedenih u programiranju se povećava broj popijenih šoljica kafe po danu.

Bilo bi poželjno da je uzet veći uzorak podataka, kako bi tačnije mogla da se odrade predviđanja. U ovom radu je urađeno predviđanje koliko osoba popije dnevno kafe ako je poznato kog je pola i koliko sati programira sa tačnošću od 34%. Predviđanje je iskorišćeno za malu aplikaciju gde se unose poznati podaci i prikazuje predviđeni broj kafe koju će osoba popiti (slika 15). Kad korisnik unese podatke I klikne na dugme iskočiće broj predviđenih kafa ispod dugmeta kao na slici 16.



Slika 15.Prikaz aplikacije



Slika 16.Prikaz rezultata

Literatura

[1] https://www.w3schools.com/python
[2] https://pandas.pydata.org
[3] https://datatofish.com
[4] https://www.kaggle.com
[5] https://www.educative.io
[6] https://mode.com/python-tutorial
[7] https://www.geeksforgeeks.org
[8] https://dfrieds.com
[9] https://en.wikipedia.org/wiki/Orange_(software)
[10] Materijali sa predavanja data mining-a, ITS 2020.