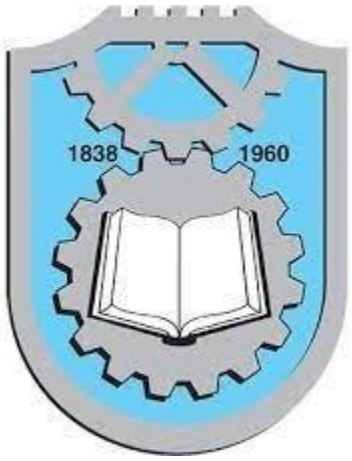
FAKULTET INŽENJERSKIH NAUKA

Kragujevac, Sestre Janjić 6



Projekat iz predmeta Osnovi mašinskog učenja

PREDVIĐANJE SAGOREVANJA KALORIJA

Profesor:

Vladimir Milovanović

Student:

Kristina Trajković 605/2019

Kragujevac, septembar 2022. godine

Ideja projekta

Glavni cilj projekta je kako mozemo da napravimo sistem masinskog ucenja koji moze da predvidi kolicinu sagorelih kalorija tokom vezbanja. Skup podataka kojikoristimo sadrži vrednosti paramatara: trajanje tokom kojeg osoba vezba, prosecan broj otkucaja u minuti, telesna temperatura, visina i tezina osobe. Datoteka sadrži 1500 primera (redova), 8 parametara (kolona).

Realizacija

Projekat je odrađen u Google Colab-u (python).

Faze realizacije:

1. Importovanje biblioteka
2. Analiza podataka
3. Standardizacija podataka
4. Razdvajanje na trening i test skup
5. Treniranje modela
6. Evaluacija modela
7. Sistem za predviđanje

1. Importovanje biblioteka

- numpy i pandas - za rad sa nizovima i tabelama
- matplotlib.pyplot – za kreiranje dijagrama
- train_test_split - za razdvajanje skupa podataka na trening i test skup
- metrics – za procenu naseg modela

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn import metrics
```

2. Analiza podataka

Obuhvata učitavanje skupa podataka, veličinu skupa i statističke vrednosti.

Prikupljanje i obrada podataka

```
# Ucitavanje podataka
calories = pd.read_csv('/content/calories.csv')
```

```
# stampanje 5 redova
calories.head()
```

	User_ID	Calories
0	14733363	231.0
1	14861698	66.0
2	11179863	26.0
3	16180408	71.0
4	17771927	35.0

```
exercise_data = pd.read_csv('/content/exercise.csv')
```

```
exercise_data.head()
```

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp
0	14733363	male	68	190.0	94.0	29.0	105.0	40.8
1	14861698	female	20	166.0	60.0	14.0	94.0	40.3
2	11179863	male	69	179.0	79.0	5.0	88.0	38.7
3	16180408	female	34	179.0	71.0	13.0	100.0	40.5
4	17771927	female	27	154.0	58.0	10.0	81.0	39.8

Naredni korak je da dobijemo informacije o podacima: broj tacaka podataka koje imamo je 15000, broj kolona 9, broj normalnih vrednosti, i imamo tip podataka svake kolone.

```
# broj redova i kolona
calories_data.shape
```

```
(15000, 9)
```

```
# informaicije o podacima
calories_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   User_ID     15000 non-null  int64  
 1   Gender      15000 non-null  object 
 2   Age         15000 non-null  int64  
 3   Height      15000 non-null  float64
 4   Weight      15000 non-null  float64
 5   Duration    15000 non-null  float64
 6   Heart_Rate  15000 non-null  float64
 7   Body_Temp   15000 non-null  float64
 8   Calories    15000 non-null  float64
dtypes: float64(6), int64(2), object(1)
memory usage: 1.0+ MB
```

Prebacujemo sve podatke u numericke vrednosti tako da racunar moze da ih razume.
Pol je u tektusalnom obliku pa ga kovertujemo u numericki.

Pretvaranje tekstualnih podataka u numericke vrednosti

```
calories_data.replace({"Gender":{'male':0,'female':1}}, inplace=True)
```

```
calories_data.head()
```

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	0	68	190.0	94.0	29.0	105.0	40.8	231.0
1	14861698	1	20	166.0	60.0	14.0	94.0	40.3	66.0
2	11179863	0	69	179.0	79.0	5.0	88.0	38.7	26.0
3	16180408	1	34	179.0	71.0	13.0	100.0	40.5	71.0
4	17771927	1	27	154.0	58.0	10.0	81.0	39.8	35.0

Statisticke mere o podacima: koji je opseg vrednosti u svakoj koloni, ovde dobija srednje vrednosti iz svake kolone.

Analiza podataka

```
# statistika podataka
calories_data.describe()
```

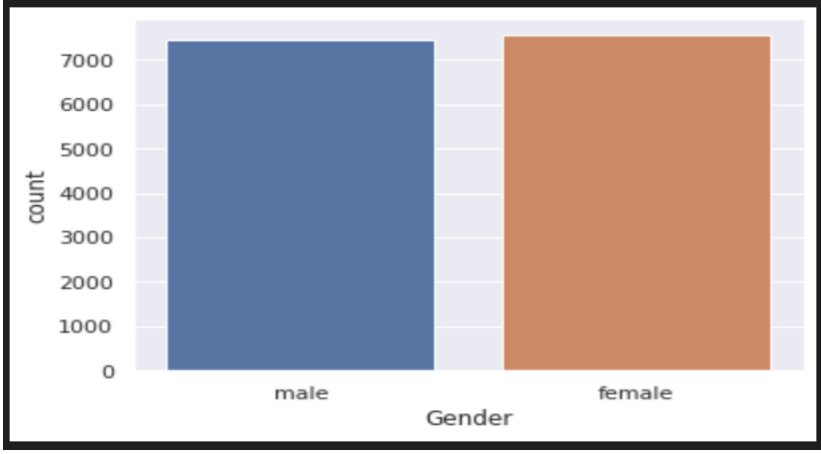
	User_ID	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
count	1.500000e+04	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000
mean	1.497736e+07	42.789800	174.465133	74.966867	15.530600	95.518533	40.025453	89.539533
std	2.872851e+06	16.980264	14.258114	15.035657	8.319203	9.583328	0.779230	62.456978
min	1.000116e+07	20.000000	123.000000	36.000000	1.000000	67.000000	37.100000	1.000000
25%	1.247419e+07	28.000000	164.000000	63.000000	8.000000	88.000000	39.600000	35.000000
50%	1.499728e+07	39.000000	175.000000	74.000000	16.000000	96.000000	40.200000	79.000000
75%	1.744928e+07	56.000000	185.000000	87.000000	23.000000	103.000000	40.600000	138.000000
max	1.999965e+07	79.000000	222.000000	132.000000	30.000000	128.000000	41.500000	314.000000

4.Vizuelizacija podataka

Neke podatke mozemo da prikazemo preko dijagrama I grafika.

```
    # prikaz roda u vidu grafa
    sns.countplot(calories_data['Gender'])
```

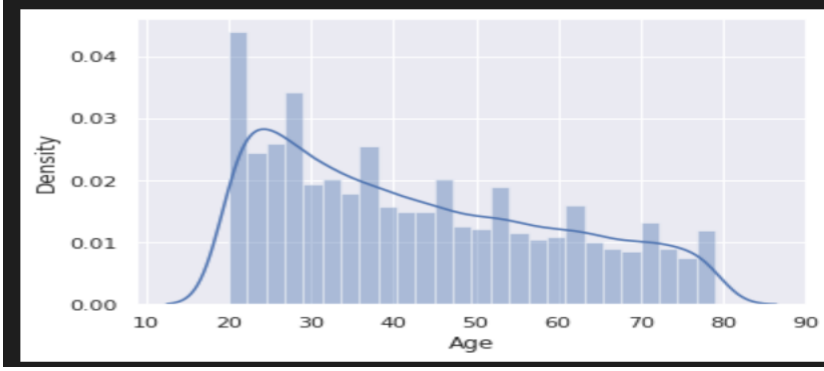
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decora
the only valid positional argument will be `data`, and
misinterpretation.
  FutureWarning

<matplotlib.axes._subplots.AxesSubplot at 0x7fcbbd7561
```



Vizuelizacija podataka je korisna jer nam govori koji opseg vrednosti je vise prisutan u nekom konkretnom skupu podataka.

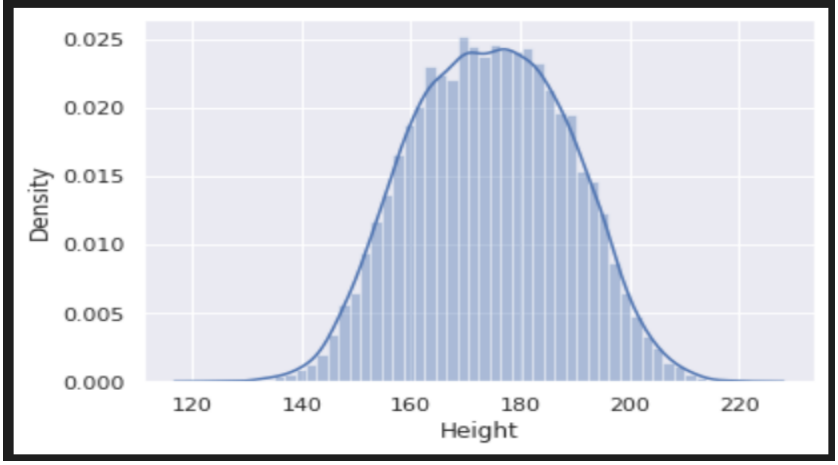
```
    # Prikaz grafika "godine"
    sns.distplot(calories_data['Age'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distrib
future version. Please adapt your code to use either `
function for histograms).
  warnings.warn(msg, FutureWarning)

<matplotlib.axes._subplots.AxesSubplot at 0x7fcbbd2005
```



```
    # prikaz grafika "visina"
    sns.distplot(calories_data['Height'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distri
future version. Please adapt your code to use either
function for histograms).
  warnings.warn(msg, FutureWarning)

<matplotlib.axes._subplots.AxesSubplot at 0x7fcbb1ed3
```



5.Korelacija u skupu podataka

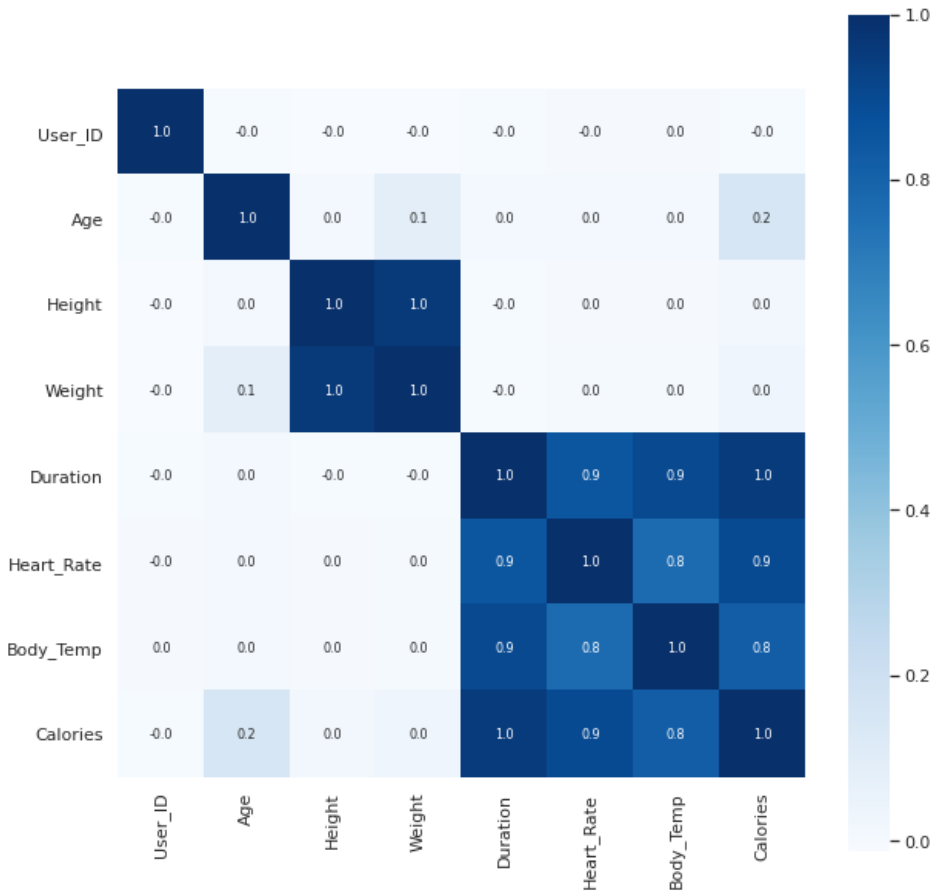
Pronalazenje koleracije u skupu podataka(odnos izmedju dve kke).
calories_data.corr je funckija koja nam pomaze da pronadjemo vrednost korelacije
2 tipa korelacije - pozitivna i negativna : ako vreme tokom koje osoba trenira vece onda ce broj kalorija koje sagori biti veci sto znaci da su te dve karakteristike u pozitivnoj korelaciji.
Ako se jedna kolona poveceva druga smanjuje one u u negativnoj korelaciji.

Pronalazenje korelacije u skupu podataka

1. Pozitivna korelacija
2. Negativna korelacija

```
correlation = calories_data.corr()
```

```
# pravljenje karte toplote

plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':8}, cmap='Blues')
```



6.Razdvajanje na trening i test skup

Razdvajanje vršimo pomoću train_test_split-a. Pri podeli 80% podataka stavljamo u trening skup, a 20% podataka u test skup. Podešeno je da se na osnovu ishoda ravnomerno podele podaci u skupovima.

Podatke o obuci koristimo za masinsko ucenje a podatke o testiranju ce se koristiti za procenu

naseg modela.

Podela podataka:

- Ukupno: 1500
- Treniranje: 1200
- Testiranje: 300

```
X = calories_data.drop(columns=['User_ID','Calories'], axis=1)
Y = calories_data['Calories']
```

```
Podela podataka na podatke o obuci i podatke o testu

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)

    print(X.shape, X_train.shape, X_test.shape)

(15000, 7) (12000, 7) (3000, 7)
```

7. Obuka modela

Ucitavamo model pomocu xgboost regressora.
Funkcija model.fit pomaze nasem modelu da nauci, tako da kada date ove podatke modelu on pronalazi obrazac izmedju podataka npr, moze da razume ako je vreme treniranja duze da je broj sagorelih kalorija veci, ili ako je broj otkucaja srca veci broj kalorija ce biti veci.

```
Ucenje modela

XGBoost Regressor

    # ucitavanje modela
    model = XGBRegressor()

    # obucavanje modela
    model.fit(X_train, Y_train)
```

8.Sistem za predviđanje

Predvidjanje na osnovu podataka testa

```
test_data_prediction = model.predict(X_test)
```

```
print(test_data_prediction)
```

```
[129.06204  223.79721   39.181965 ... 145.59767   22.53474   92.29064 ]
```

Predviđamo vrednost ishoda na osnovu trening i test podataka i računamo vrednost preciznosti na osnovu tih predviđanja i stvarnih vrednosti. Sa podacima testa model moze da pronadje koliko kalorija se potrosi, pa mi uporedjujemo vrednosti koje je predvideo nas model sa originalnim vrednostima. Da bismo uporedili koristimo merenje srednje apsolutne greske(matrics iz biblioteke).

Srednja vrednost apsolutne greske

```
mae = metrics.mean_absolute_error(Y_test, test_data_prediction)
```

```
print("Mean Absolute Error = ", mae)
```

```
Mean Absolute Error =  2.7159012502233186
```