به نام **خدا**

""تمرین اول داده کاوی""

محمدرضا قادری ۹۶۲۷۰۵۷

۱) برای این بخش ابتدا کتابخانه اضافه میکنیم

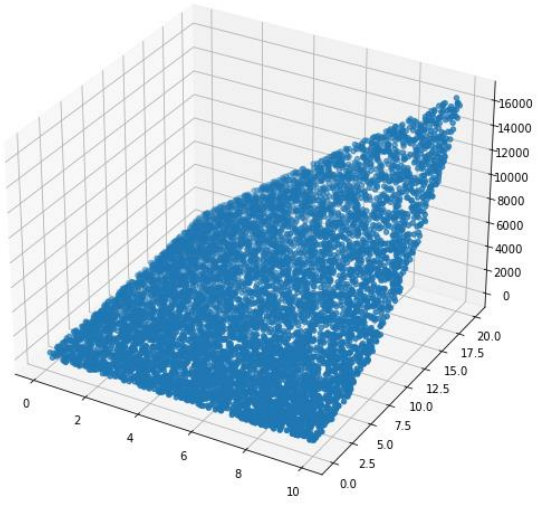
```
In [111]: import pandas as pd
          import numpy as np
          import scipy.io
          import matplotlib.pyplot as plt
          import random
```

سپس فایل‌ها را می‌خوانیم

```
In [112]: data = np.load('data.npz')
          print(data.files)
          x1 = data['x1']
          x2 = data['x2']
          y = data['y']
          a = 0.01

          ['y', 'x2', 'x2_test', 'x1', 'y_test', 'x1_test']
```

در ادامه می‌توانیم plot برای داده های موجود رسم کنیم

```
In [69]: fig = plt.figure()
         ax = plt.axes(projection ='3d')
         ax.scatter(x1, x2, y)
         plt.show()
```



بدلیل اینکه رگرسوین داریم فرض میکنیم یک b به عنوان ضرایب داریم یکی برای x1 یکی برای x2 و دیگری ثابت و در یک دوره به دنبال این ضرایب می‌گردیم

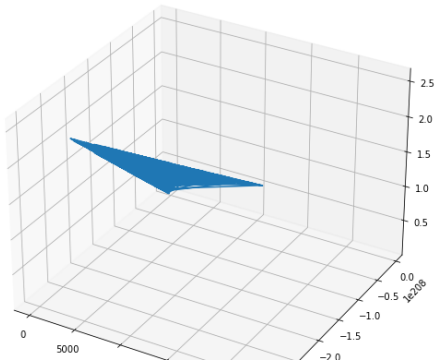
```
In [113]: b = np.array([[1],[1],[1]])
matrix1 = np.full((len(x1),1),1)
x = np.concatenate((matrix1,x1.reshape(len(x1),1),x2.reshape(len(x2),1)) , axis = 1)

In [114]: for i in range (50):
    b1 = np.transpose(x)@x@b - np.transpose(x)@y.reshape(len(y),1)
    b = b - a*b1
print(b)

[[-8.77661084e+205]
 [-4.62818824e+206]
 [-1.11566763e+207]]
```

در ادامه برای استوکستیک نیز می‌توانیم یک عدد رندوم را داشته باشیم تا برهمین روند به b ها برسیم برای اینکه bهای جدید زیاد تاثیر نگذارند ضریب ۰/۰۰۱ برایشان ست کردیم

```
In [140]: b_new = np.array([[1],[1],[1]])

In [141]: for i in range (50):
    randoms = random.randint(0, len(x1)-1)
    xx = np.array([[1 , x1[randoms], x2[randoms]]])
    b2 = np.transpose(xx)@xx@b_new - 0.1*np.transpose(xx)*y[randoms]
    b_new = b_new - a*b2
print(b_new)

[[-8.51368331]
 [-3.81196129]
 [35.05972413]]
```

برای رسم این توابع از plot ۳بعدی استفاده می‌کنیم.

```
In [151]: x1_test = data['x1_test']
x2_test = data['x2_test']
y_test = data['y_test']

In [153]: xxx  = np.concatenate((np.full((len(x1_test) , 1),1), x1_test.reshape(len(x1_test),1), x2_test.reshape(len(x2_test),1)), axis=1)

In [154]: y1 = xxx@b
y2 = xxx@b_new
ax = plt.axes(projection ='3d')
ax.plot3D(y_test, y1.reshape(len(y1)), y2.reshape(len(y2)))

Out[154]: [<mpl_toolkits.mplot3d.art3d.Line3D at 0x4e65001f0>]
```



برای محاسبه خطا از SEE استفاده که جمع توان دوی اختلاف بین اعداد محاسبه شده است برای مثال برای استوکستسک

```
In [167]: arr2 = y2.reshape(len(y2))
error2 = 0
for i in range(len(y_test)):
    error2 += (y_test[i] - arr2[i])**2
print (error2)

4.4060660717834125e+17
```

۲) در این قسمت از دو کتابخانه pandas و matplotlib استفاده می‌کنیم پس هر دو را import می‌کنیم.

```
In [32]: import pandas as pd
         import matplotlib.pyplot as plt
```

a) برای این قسمت ابتدا csv را می‌خوانیم و پوینتر رو به اون انتصاب می‌دهیم df حالا می‌توانیم ابتدا وانتهای فایل رو با دستور head و tail مشخص کنیم. در انتها هم هر دو را concat می‌کنیم و جواب را بیرون می‌گذاریم.

```
In [6]: df = pd.read_csv("players.csv")
        top = df.head(1)
        bot = df.tail(1)
        contact = pd.concat([top,bot])
        print(contact)
```

```
           ID          Name          FullName  Age  Height  Weight  \
0      158023      L. Messi      Lionel Messi   33     170      72
19019  241493  S. Cartwright  Samuel Cartwright   19     185      75

                                        PhotoUrl Nationality  Overall  \
0      https://cdn.sofifa.com/players/158/023/21_60.png   Argentina       93
19019  https://cdn.sofifa.com/players/241/493/21_60.png     England       49

       Potential  ...  LMRating  CMRating  RMRating LWBRating CDMRating  \
0             93  ...        93        90        93        69        68
19019         65  ...        38        36        38        46        45

      RWBRating  LBRating  CBRating  RBRating GKRating
0            69        65        55        65       22
19019        46        47        51        47       15

[2 rows x 90 columns]
```

b) برای مشخص کردن اینکه cell ایی خالی باشد.

```
In [96]: df.isnull()
```

Out[96]:

	ID	Name	FullName	Age	Height	Weight	PhotoUrl	Nationality	Overall	Potential	...	LMRating	CMRating	RMRating	LWBRating	CDMRating	RW
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
19015	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
19016	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
19017	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
19018	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	
19019	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	

19020 rows × 90 columns

c) تمامی مقادیر رو از ستون Weight می‌خوانیم و با استفاده از max و min و mean مقادیر مورد نظر رو بدست می‌آوریم.

```
In [8]: Weight = df["Weight"]
        max_Weight = Weight.max()
        min_Weight = Weight.min()
        mean_Weight = Weight.mean()
        print(max_Weight, min_Weight,mean_Weight)
```

```
110 50 75.05241850683491
```

d) در این قسمت ستون Nationality را انتخاب می‌کنیم و تابع value_counts را استفاده می‌کنیم تا اگر کشور تکراری موجود بود برای ما بشمارد در انتها این را به یک dictionary تبدیل می‌کنیم و مقادیر max,min را از دیکشنری بیرون می‌کشیم.

```
In [12]: country = df["Nationality"].value_counts()
         nationality_dict = country.to_dict()
         all_values = nationality_dict.values()
         max_country = max(nationality_dict, key=nationality_dict.get)
         min_country = min(nationality_dict, key=nationality_dict.get)
         max_value = max(all_values)
         min_value = min(all_values)
         print(max_country , max_value)
         print(min_country , min_value)

         England 1706
         Malta 1
```

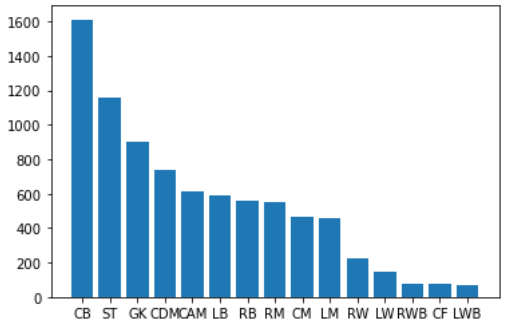
e) در این حالت برای پوینتر شرط تعریف می‌کنیم که برای مثال چه ستون‌هایی دارای چه شرایطی باشند در اینجا دو شرط داریم که باهم and شده‌اند.

```
In [69]: result = df[(df["Growth"] < 3) & (df["Potential"] < 84)]
         print(result.head(5))

              ID            Name                FullName  Age  Height  Weight  \
139  169195  Renato Augusto           Renato Augusto   32     186      86
140  169416        C. Vela              Carlos Vela   31     177      77
143  215333       D. Zapata             Duván Zapata   29     189      88
144  216547           Rafa  Rafael A. Ferreira Silva   27     172      66
147  220407     M. Dúbravka          Martin Dúbravka   31     190      80

                                          PhotoUrl Nationality  Overall  \
139  https://cdn.sofifa.com/players/169/195/21_60.png      Brazil       83
140  https://cdn.sofifa.com/players/169/416/21_60.png      Mexico       83
143  https://cdn.sofifa.com/players/215/333/21_60.png    Colombia       83
144  https://cdn.sofifa.com/players/216/547/21_60.png    Portugal       83
147  https://cdn.sofifa.com/players/220/407/21_60.png    Slovakia       83

     Potential  ...  LMRating  CMRating  RMRating LWBRating CDMRating  \
139         83  ...        82        83        82        79        80
140         83  ...        83        78        83        62        60
143         83  ...        75        68        75        56        57
144         83  ...        83        77        83        72        65
147         83  ...        37        41        37        33        38

     RWBRating  LBRating  CBRating  RBRating GKRating
139         79        77        75        77       21
140         62        58        50        58       22
143         56        54        56        54       18
144         72        68        56        68       20
147         33        32        32        32       83

[5 rows x 90 columns]
```

f) با استفاده قسمت d و e همان مراحل را برای dictionary حالت قبل می‌رویم و با توجه به BestPosition اون بازیکن تعداد بازیکنان در اون موقعیت را میشماریم و یک plot از نوع bar رسم می‌کنیم با توجه به key و value های دیکشنری جدید.

```
In [38]: country = result["BestPosition"].value_counts()
         nationality_dict = country.to_dict()
         keys = nationality_dict.keys()
         values = nationality_dict.values()

         plt.bar(keys, values)
```

```
Out[38]: <BarContainer object of 15 artists>
```



g) در ابتدا شرط رشد بازیکنان را چک می‌کنیم ، سپس برا اساس باشگاه‌ها بازیکنان رو به یک dictionary نگاشت می‌کنیم و در انتها ۱۰ بیشترین بازیکن آینده دار در باشگاه‌ها رو نمایش دادیم.

```
In [51]: result = df[(df["Growth"] > 0)]
         club = result["Club"].value_counts()
         club_dict = club.to_dict()
         all_values = club_dict.values()
         max_club = max(club_dict, key=club_dict.get)
         max_value = max(all_values)
         print(max_club , max_value)
         print(club.nlargest(10))
```

```
Free agent 75
Free agent                      75
Chamois Niortais Football Club  29
CD Mirandés                     29
RC Celta                        29
Heracles Almelo                 28
Hertha BSC                      28
LOSC Lille                      28
Vitória Guimarães               28
Famalicão                       28
AZ Alkmaar                      27
Name: Club, dtype: int64
```

h) در این قسمت دوتا شرط ContractUntil را تا سال ۲۰۲۱ و NationalTeam را Not in team را باهم and می‌کنیم و درنهایت یک دیکشنری از نام بازیکنان و id آنها میسازیم و در انتها اندازه دیکشنری برای ما مهم است.

```
In [72]: result = df[(df["ContractUntil"] == 2021) & (df["NationalTeam"] == "Not in team")]
         state = result["Name"]
         state_dict = state.to_dict()
         print(len(state_dict))
```

```
6727
```

i) در این قسمت ما باید برای csv دوتا شرط سن و باشگاه رو تعیین می‌کنیم که بازیکن بایستی کمتر از ۲۴ سال داشته باشد و باشگاهش چلسی باشد. در انتها رو ستون ValueEUR یک جمع انجام میدهیم و نتیجه حاصل می‌شود.

```
In [60]: result = df[(df["Age"] < 24) & (df["Club"] == "Chelsea")]
         print(result["ValueEUR"].sum())
```

```
336000000
```

j) در این قسمت به csv میگیم تو ستون اسم دنبال نام E. Hazard بگرد و این سطر رو برگردون و در نمایش آن ذکر می‌کنیم فقط 'Positions' , 'WageEUR', 'Club' را نمایش دهد.

```
In [63]: result = df[(df["Name"] == "E. Hazard")]
         print(result[['Positions' , 'WageEUR', 'Club']])

    Positions  WageEUR         Club
22        LW   350000  Real Madrid
```