داده کاوی

تمرین سری اول

قسمت عملی

سوال اول)

A

در کد زیر ما ماتریس B که شامل b0,b1,b2 هست را مقدار دهی اولیه میکنیم و در ادامه ماتریس X را میسازیم و حال در ۵۰ گام مقدار B را آپدیت میکنیم به این صورت که در هر تکرار حلقه مقدار گرادیان را محاسبه میکنیم و مقدار B با توجه به گرادیان آپدیت میکنیم.

```
#initialize B0,B1,B2 to zero and finding best value later
B = np.array([[1],[1],[1]])
oneMatrix = np.full((len(x1),1),1)
#creating X matrix
X = np.concatenate((oneMatrix,x1.reshape(len(x1),1),x2.reshape(len(x2),1)),axis = 1)
#algorithm nelow is for training values of B using batch gradient desent
for i in range(50):
    B_gradient = np.transpose(X)@X@B - np.transpose(X)@y.reshape(len(y),1)
    B = B - 0.01*B_gradient
print(B)
```

B

در کد زیر همانند کد بالا عمل میکنیم با این تفاوت که ماتریس X ترکیب تمامی داده ها نیست و فقط یک داده به صورت رندوم انتخاب میشود تا نتیجه را در هر دست آپدیت کنیم.

```
#algorithm below is for stochastic gradient desent
B2 = np.array([[1],[1],[1]])
for i in range(300):
    random_number = random.randint(0, len(x1)-1)
    XTWO = np.array([[1,x1[random_number],x2[random_number]]])
    B2_gradient = np.transpose(XTWO)@XTWO@B2 - 0.1*np.transpose(XTWO)*y[random_number]
    B2 = B2 - 0.01*B2_gradient
print(B2)
```

C

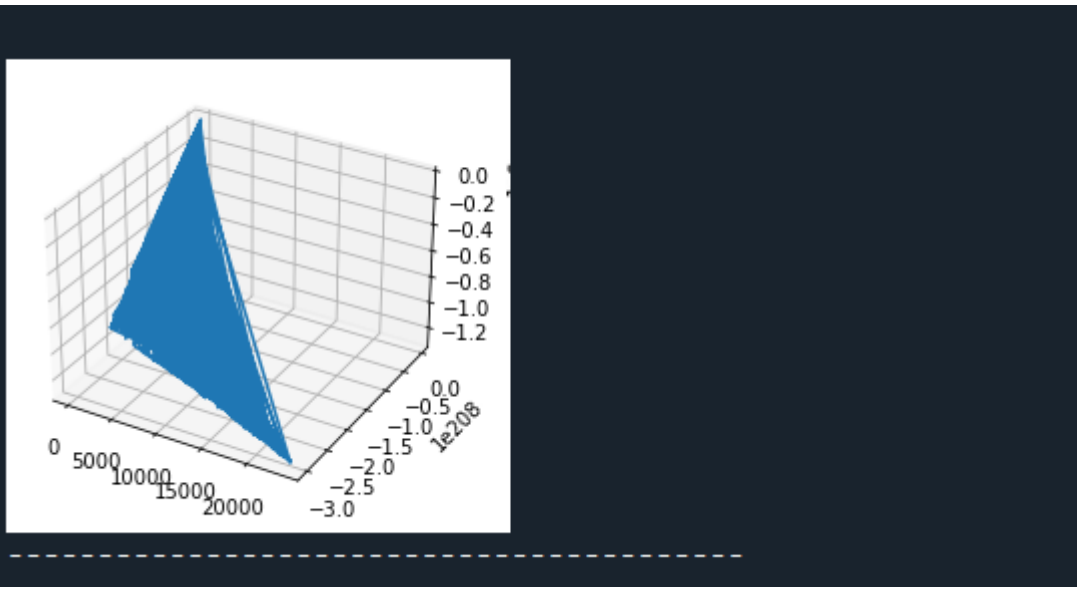
با توجه به خروجی زیر میتوان متوجه شد که روش گرادیان تصادفی نتیجه قابل قبول تری را میتواند به ما بدهد.

```
B in batch gradient desent
[[-8.77661084e+205]
 [-4.62818824e+206]
 [-1.11566763e+207]]
--------------------------------------------------
B in stochastic gradient desnt
[[158.42822964]
 [-18.22294791]
 [ 19.79376741]]
```

D

با به دست اوردن مقادیر y تست با توجه به نوع الگوریتم میتوانیم نمودار سه بعدی را رسم نماییم.

```
#plotting test data real value and gradient desnt and stochastic
print('-----------------------------------------')
X_grad = 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)
y_gradient = X_grad@B
y_stochastic = X_grad@B2
ax = plt.axes(projection='3d')
ax.plot3D(y_test,y_gradient.reshape(len(y_gradient)),y_stochastic.reshape(len(y_stochastic)))
plt.show()
print('-----------------------------------------')
```



E

```python
#code below is for calculating the error
def error(arr1,arr2):
    length = len(arr1)
    totalErr = 0
    for i in range(length):
        totalErr += (arr1[i]-arr2[i])**2
    return totalErr
#for training data set
y_grad_train = X@B
y_stoch_train = X@B2
print('train')
print('err for grad desent:',error(y,y_grad_train.reshape(len(y_grad_train)))
      ,'err for stochastic:',error(y,y_stoch_train.reshape(len(y_stoch_train))))
print('test')
print('err for grad desent:',error(y_test,y_gradient.reshape(len(y_gradient)))
      ,'err for stochastic:',error(y_test,y_stochastic.reshape(len(y_stochastic))))
```

```
----------------------------------------
train
err for grad desent: inf err for stochastic: 23136375245551.55
test
err for grad desent: inf err for stochastic: 8610095619157.735
C:\Users\User\Desktop\data mining\hw\1\9731113_DM01\question1.py:57: RuntimeWarning: overflow
encountered in double_scalars
  totalErr += (arr1[i]-arr2[i])**2
```

مقدار خطا برای روش گرادیان نزولی بسیار زیاد است و سرریز میکند اما برای روش گرادیان تصادفی نمایش داده شده است.

سوال دوم)

A

کد

خواندن داده ها و نشان دادن ۵ داده اول و ۵ داده آخر

```python
#a
myData = pd.read_csv('players.csv')
print('first 5 data')
print(myData[:5])
print('last 5 data')
print(myData[len(myData)-5:len(myData)])
```

خروجی

```
first 5 data
       ID              Name  ... RBRating  GKRating
0  158023          L. Messi  ...       65        22
1   20801  Cristiano Ronaldo  ...       64        23
2  200389          J. Oblak  ...       35        92
3  192985      K. De Bruyne  ...       78        24
4  190871         Neymar Jr  ...       65        23

[5 rows x 90 columns]
last 5 data
          ID           Name           FullName  ...  CBRating  RBRating  GKRating
19015  257371    M. Nzongong      Mike Nzongong  ...        40        42        18
19016  259160        L. Bell         Lewis Bell  ...        35        41        13
19017  259157        Y. Arai         Yasin Arai  ...        39        44        17
19018  253763     R. Dinanga    Ricardo Dinanga  ...        30        34        16
19019  241493  S. Cartwright  Samuel Cartwright  ...        51        47        15

[5 rows x 90 columns]
```

B

کد

با استفاده از isnull و با دیدن خروجی آن هایی که True میباشند به آن معناست که missing value ما هستند.

```
#b
print('----------------------PART B----------------------------------------')
print(myData.isnull())
```

خروجی

```
----------------------PART B----------------------------------------
          ID   Name  FullName    Age  ...  LBRating  CBRating  RBRating  GKRating
0      False  False     False  False  ...     False     False     False     False
1      False  False     False  False  ...     False     False     False     False
2      False  False     False  False  ...     False     False     False     False
3      False  False     False  False  ...     False     False     False     False
4      False  False     False  False  ...     False     False     False     False
...      ...    ...       ...    ...  ...       ...       ...       ...       ...
19015  False  False     False  False  ...     False     False     False     False
19016  False  False     False  False  ...     False     False     False     False
19017  False  False     False  False  ...     False     False     False     False
19018  False  False     False  False  ...     False     False     False     False
19019  False  False     False  False  ...     False     False     False     False

[19020 rows x 90 columns]
```

C

کد

محاسبه خروجی با استفاده از توابع min,max,sum

```
#c
print('---------------------PART C----------------------------------------')
print('min',min(myData['Weight']))
print('max',max(myData['Weight']))
print('mean',sum(myData['Weight'])/len(myData['Weight']))
```

خروجی

```
---------------------PART C----------------------------------------
min 50
max 110
mean 75.05241850683491
```

D

کد

تابع value_counts تعداد تکرار هر value را در لیست پانداسی میدهد که از بزرگ به کوچک مرتب شده هستند.

```
#d
print('---------------------PART D----------------------------------------')
counts = myData['Nationality'].value_counts()
print('max:',counts[0:1],'\nmin:',counts[len(counts)-1:len(counts)])
```

خروجی

```
---------------------PART D----------------------------------------
max: England    1706
Name: Nationality, dtype: int64
min: Indonesia    1
Name: Nationality, dtype: int64
```

E

کد

یکی یکی داده هارا مورد پردازش قرار میدهیم و هر داده ای که شرط مورد نظر مارا داشت به عنوان خروجی چاپ میشود.

```python
#e
print('--------------------PART E--------------------------------------')
for i in range(len(myData)):
    if myData['Growth'][i] < 3 and myData['Potential'][i] < 84:
        print('palyer name:',myData['FullName'][i],'Growth:',myData['Growth'][i],
              'Potential:',myData['Potential'][i])
```

بخشی از خروجی

```
palyer name: Karl Sheppard Growth: 0 Potential: 61
palyer name: Juan Raúl Neira Growth: 2 Potential: 63
palyer name: Chris Taylor Growth: 0 Potential: 61
palyer name: Nigel Atangana Growth: 0 Potential: 61
palyer name: Theo Robinson Growth: 0 Potential: 61
palyer name: Anvit Swaminathan Growth: 0 Potential: 61
palyer name: Won Gun Kim Growth: 1 Potential: 62
palyer name: Matías Pato Growth: 0 Potential: 61
palyer name: George Thomson Growth: 0 Potential: 61
palyer name: Devindra Pillai Growth: 0 Potential: 61
palyer name: Hyo Gi Kim Growth: 0 Potential: 61
palyer name: Joe Martin Growth: 0 Potential: 61
palyer name: Senwen Luo Growth: 1 Potential: 62
palyer name: Josh Falkingham Growth: 0 Potential: 61
palyer name: Tommy Grupe Growth: 0 Potential: 61
palyer name: Ali Al Khaibari Growth: 0 Potential: 61
palyer name: Ronald Mukiibi Growth: 0 Potential: 61
palyer name: Tibor Joza Growth: 0 Potential: 61
palyer name: Jobi McAnuff Growth: 0 Potential: 61
palyer name: Chenglin Zhang Growth: 0 Potential: 61
palyer name: Jhony Rodríguez Growth: 2 Potential: 63
palyer name: Georgi Pashov Growth: 0 Potential: 61
palyer name: Hélton Fernando Carvalhal Lima Growth: 0 Potential: 61
palyer name: Jinhao Bi Growth: 0 Potential: 61
palyer name: Abdullah Al Owayshir Growth: 0 Potential: 61
palyer name: Héctor Pérez Growth: 1 Potential: 61
palyer name: Adrián Martínez Growth: 1 Potential: 61
palyer name: Ian Bermingham Growth: 0 Potential: 60
palyer name: David Fitzpatrick Growth: 0 Potential: 60
palyer name: Francesco Rossi Growth: 0 Potential: 60
palyer name: Óskar Sverrisson Growth: 0 Potential: 60
palyer name: Hamad Al Juhayyim Growth: 0 Potential: 60
palyer name: Víctor Centurión Growth: 0 Potential: 60
palyer name: Wilson Mena Growth: 0 Potential: 60
palyer name: Xin Luo Growth: 0 Potential: 60
palyer name: Jung Nam Hong Growth: 0 Potential: 60
palyer name: Osama Malik Growth: 0 Potential: 60
palyer name: Ahmed Ashraf Growth: 0 Potential: 60
palyer name: Ryohei Yamazaki Growth: 0 Potential: 60
palyer name: Abdulaziz Majrashi Growth: 0 Potential: 60
palyer name: Abdulaziz Al Dawsari Growth: 0 Potential: 60
palyer name: Desley Ubbink Growth: 1 Potential: 61
```

F

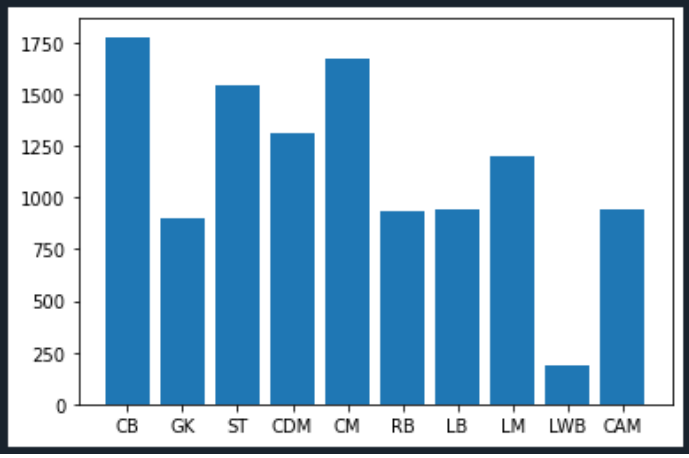
کد

از آنجایی که این سوال ادامه سوال قبلی است با استفاده از کد قسمت قبل و اضافه کردن یک دیکشنری که پست های متفاوت در آن نمایش داده شده است تعداد تکرار هر پست را محاسبه میکنیم و آن را روی نمودار میله ای نمایش میدهیم.

```
#e
print('---------------------PART E----------------------------------------')
positions = {'CB':0,'GK':0,'ST':0,'CDM':0,'CM':0,'RB':0,'LB':0,'LM':0,'LWB':0,'CAM':0}
def updatePositions(pos_arg):
    for i in positions:
        if i in pos_arg.split(','):
            positions[i] += 1
for i in range(len(myData)):
    if myData['Growth'][i] < 3 and myData['Potential'][i] < 84:
        print('palyer name:',myData['FullName'][i],'Growth:',myData['Growth'][i],
              'Potential:',myData['Potential'][i])
        pos = myData['Positions'][i]
        updatePositions(pos)
#f
print('---------------------PART F----------------------------------------')
plt.bar(positions.keys(),positions.values())
plt.show()
```

خروجی

```
----------------------PART F----------------------------------------
```



G

کد

داده جدیدی از فیلتر کردن داده های اولیه به دست می اوریم و با استفاده از تابع value_counts نام هر باشگاه و تعداد بازیکنان آینده دار آن را محاسبه میکنیم.

توجه شود که برای فیلتر بازیکنان آینده دار آن دسته از بازیکنانی که سن کمتر از ۲۴ و پتانسیل بالای ۸۰ را دارا بودند در نظر گرفته شده است.

```
#g
print('----------------------PART G----------------------------------------')
newData = myData[(myData['Potential'] > 80) & (myData['Age'] < 24)]
clubCounts = newData['Club'].value_counts()
print('max:',clubCounts[0:1])
print('min',clubCounts[len(clubCounts)-1:len(clubCounts)])
```

خروجی

```
----------------------PART G----------------------------------------
max: Manchester United    16
Name: Club, dtype: int64
min San Lorenzo de Almagro    1
Name: Club, dtype: int64
```

H

کد

محاسبه کردن تعداد بازیکنان مورد نظر با قرار دادن دو محدودیت در شرط

```
#h
print('----------------------PART H----------------------------------------')
counter = 0
for i in range(len(myData)):
    if myData['ContractUntil'][i] == 2021 and myData['NationalTeam'][i] == 'Not in team':
        counter += 1
print(counter)
```

خروجی

```
----------------------PART H----------------------------------------
6727
                      PART I
```

I

کد

پردازش همه داده ها در حلقه و چک کردن شرط مورد نظر

```
#i
print('----------------------PART I----------------------------------------')
for i in range(len(myData)):
    if myData['Club'][i] == 'Chelsea' and myData['Age'][i] < 24:
        print('name:',myData['FullName'][i],'value:',myData['ValueEUR'][i])
```

خروجی

```
----------------------PART I----------------------------------------
name: Kai Havertz value: 121000000
name: Ben Chilwell value: 34500000
name: Christian Pulisic value: 41500000
name: Mason Mount value: 43000000
name: Reece James value: 29500000
name: Tammy Abraham value: 29000000
name: Fikayo Tomori value: 15500000
name: Callum Hudson-Odoi value: 10000000
name: Charly Musonda value: 4800000
name: Billy Gilmour value: 4400000
name: Dujon Sterling value: 2300000
name: Jack Wakely value: 500000
```

J

کد

نشان دادن موارد خواسته شده در صورت تطابق اسم با اسم مورد نظر

```
#j
print('----------------------PART J----------------------------------------')
for i in range(len(myData)):
    if myData['Name'][i] == 'E. Hazard':
        print('Positions:',myData['Positions'][i],'income:',myData['WageEUR'][i],
              'Club',myData['Club'][i])
        break
```

خروجی

```
----------------------PART J----------------------------------------
Positions: LW income: 350000 Club Real Madrid
```