

Bilgisayar Mühendisliğinde Matematik Uygulamaları

Mert ATAY - 180202090

25 Mayıs 2021

Programlama Dili : Python
Kullanılan Kütüphaneler : Pandas, Numpy, Scikit-learn, Seaborn

1 Örnek 1:

1.1 Problemimizin Tanımı:

Bu örnekteki problemimizde herhangi bir üniversiteye başvuru yapan öğrenci adayların üniversite yönetimi tarafından kabul edilip edilmeyeceğini belirlemek için logistik regresyon analizi yöntemiyle sonuç elde edilmek isteniyor.

1.2 Problemde Uygulanan Adımlar

Bu problemimizde uygulanacak adımları sırasıyla veri seti oluşturulup, kodda çözüm için neler yaptığımızla devam edip, en son olarak programımızın sonuçlarını göstereceğim.

1.2.1 1.Adım : Veri Seti Oluşturma

Öncelikle binary classification yapısına göre burada iki olası sonuç vardır: Ya "admitted"(1 değeri) olarak veya "rejected"(0 değeri) olarak sonuçlanacaktır.

Eğer Python'da bir lojistik regresyon oluşturmak istersek burada:

- Bağımlı değişken, bir kişinin kabul edilip edilmeyeceğini temsil eder,
- Bağımsız değişkenlerimiz ise GMAT puanı, GPA ve iş deneyimi süresidir.

Veri setini aşağıda görebilirsiniz:

	gmat	gpa	work_experience	admitted
0	780	4	3	1
1	750	3.9	4	1
2	690	3.3	3	0
3	710	3.7	5	1
4	680	3.9	4	0
5	730	3.7	6	1
6	690	2.3	1	0
7	720	3.3	4	1
8	740	3.3	5	1
9	690	1.7	1	0
10	610	2.7	3	0
11	690	3.7	5	1
12	710	3.7	6	1
13	680	3.3	4	0
14	770	3.3	3	1
15	610	3	1	0
16	580	2.7	4	0
17	650	3.7	6	1
18	540	2.7	2	0
19	590	2.3	3	0
20	620	3.3	2	1
21	600	2	1	0
22	550	2.3	4	0
23	550	2.7	1	0
24	570	3	2	0
25	670	3.3	6	1
26	660	3.7	4	1
27	580	2.3	2	0
28	650	3.7	6	1
29	660	3.3	5	1
30	640	3	1	0
31	620	2.7	2	0
32	660	4	4	1
33	660	3.3	6	1
34	680	3.3	5	1
35	650	2.3	1	0
36	670	2.7	2	0
37	580	3.3	1	0
38	590	1.7	4	0
39	690	3.7	5	1

Bu veri setiyle çalıştım. Kodda ise nasıl oluşturulduğunu görebilirsiniz:

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.linear_model import LogisticRegression
4 from sklearn import metrics
5 import seaborn as sn
6 import matplotlib.pyplot as plt
7
8 #veri setini excel dosyasından okuyoruz
9 dataset = pd.read_excel('dataset.xlsx',index_col=0)
10 #veri setini dataframe olacak şekilde pandas kut. yardimiyla
    tanımlıyoruz.
11 df = pd.DataFrame(dataset)
12 #Dataframe i gösteriyoruz
13 print (df)
```

Listing 1: Veri Seti Oluşturma

Sonuç olarak da aşağıya bu terminal sonucu gelecektir:

```
PS C:\Users\Mert Atay\Desktop\ödev bmmu> python logisticregresyon.py
   gmat  gpa  work_experience  admitted
0    780  4.0                3          1
1    750  3.9                4          1
2    690  3.3                3          0
3    710  3.7                5          1
4    680  3.9                4          0
5    730  3.7                6          1
6    690  2.3                1          0
7    720  3.3                4          1
8    740  3.3                5          1
9    690  1.7                1          0
10   610  2.7                3          0
11   690  3.7                5          1
12   710  3.7                6          1
13   680  3.3                4          0
14   770  3.3                3          1
15   610  3.0                1          0
16   580  2.7                4          0
17   650  3.7                6          1
18   540  2.7                2          0
19   590  2.3                3          0
20   620  3.3                2          1
21   600  2.0                1          0
22   550  2.3                4          0
23   550  2.7                1          0
24   570  3.0                2          0
25   670  3.3                6          1
26   660  3.7                4          1
27   580  2.3                2          0
28   650  3.7                6          1
29   660  3.3                5          1
```

Figure 1: Veri Seti Oluşturma (Sonucu)

1.2.2 2.Adım: Logistic Regresyon Modeli

2.adımda modelimizi oluşturan kodları ve onların sonuçlarını göreceksiniz. Öncelikle analiz için x ve y değerlerimizin olması gerekiyor. Bunları şu şekilde aldım.

```
1 x = df[['gmat', 'gpa', 'work_experience']]
2 y = df['admitted']
```

Listing 2: x ve y Değerleri

Daha sonra bu değerlerden eğitilecek verileri ayrı ve test verileri ayrı olacak şekilde seçtim.

```
1 #eğitilecek x degerlerinden bir test veri seti elde ediyoruz.
  Belirtilen test_size = veri
2 #setinin ceyregini kullanıyor.
3 X_train,X_test,y_train,y_test = train_test_split(X,y,test_size
  =0.25,random_state=0)
```

Listing 3: Kullanılacak verilerin ayrıştırılması

Şimdi modelimizi oluşturabilecek kıvama geldi. Modelimizin ismini logisticregression değişkenine tanımlayarak model oluşturuldu.Daha sonra eğilecek değerler verildi ve en son da tahmin (predict) işlemleri yapıldı.

```
1 #lojistik regresyon modelini belirtiyoruz
2 logistic_regression= LogisticRegression()
3 #analize giricek modeli e itiyoruz
4 logistic_regression.fit(X_train,y_train)
5 #yukar dan al nan x elemanlar ndan olu an test verisetinden
  e itilen ve
6 #tahmin edilen y de erlerini de i kene at yoruz
7 y_pred=logistic_regression.predict(X_test)
8 print("-----")
9 print (X_test)
10 print("-----")
11 print (y_pred)
```

Listing 4: Logistic Regression

Eğer örnek bir veri seti görmek istersek :

```
-----
      gmat  gpa  work_experience
22    550  2.3                4
20    620  3.3                2
25    670  3.3                6
4     680  3.9                4
10    610  2.7                3
15    610  3.0                1
28    650  3.7                6
11    690  3.7                5
18    540  2.7                2
29    660  3.3                5
-----
[0 0 1 1 0 0 1 1 0 1]
```

Figure 2: Test Verileri

```
Skor:0.9666666666666667
-----
Doğruluğu: 0.8
-----
```

Figure 3: Skor ve Doğruluk Değeri

Bu verilerde test amaçlı olarak kullanılıp eğitilen modele göre bir doğruluk payı değeri alırlar. Bu değerin 0 ile 1 arasında olması gerekir. Bu değer 0'a ne kadar çok yaklaşırsa tahminler olumsuz sonuçlar ortaya çıkaracaktır. Fakat 1'e ne kadar çok yaklaşırsa bir o kadar olumlu sonuç ortaya çıkma şansı doğurur. Yukarıda gördüğünüz örnek verilerde bir dizide bulunan 0 ve 1 değerleri gördünüz. Test verilerinde kabul edilen sayısı ve kabul edilmeyenler eğitilen modele göre bu sonucu almıştır.

Son olarak bu doğruluk değeri ve eğitilen modelimizin doğruluk değerini görelim.

1.2.3 3.Adım: Sonuçlar

```
1 print("Skor:" + str(logistic_regression.score(X_train,y_train)))
2 print("-----")
3 print('Do rulu u: ',metrics.accuracy_score(y_test, y_pred)) #
  Do ruluk skoru
4 print("-----")
5
6
7
8 confusion_matrix = pd.crosstab(y_test, y_pred, rownames=['Gerçek'],
  colnames=['Tahmin'])
9 sn.heatmap(confusion_matrix, annot=True)
10
11 plt.show()#Ekranda goster
```

Listing 5: Logistic Regression Sonuç Bastırımı

Şimdiki sonuçta gerçek ve tahmin değerleri arasındaki 4 farklı şekilde bulunan:

- Doğru Pozitifler: 4
- Doğru Negatifler: 4
- Yanlış Pozitifler : 1
- Yanlış Negatifler : 1

şeklde elde edilmiştir.

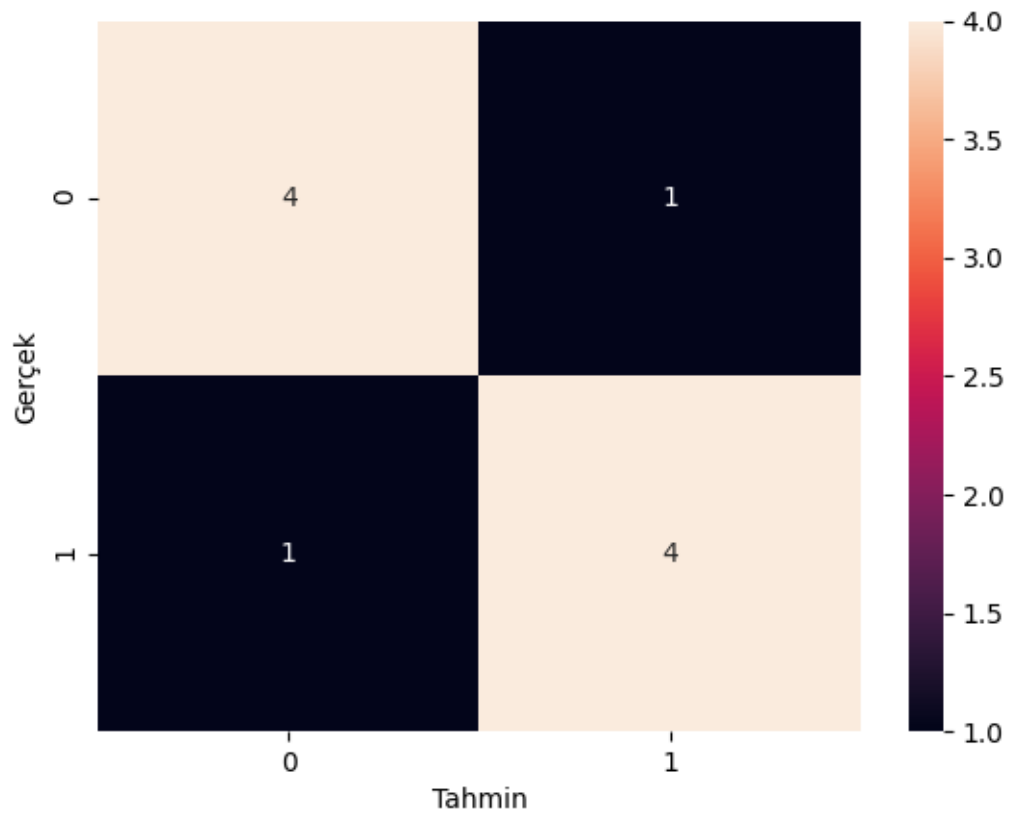


Figure 4: Sonuç Grafiği

2 Örnek 2:

2.1 Problemimizin Tanımı:

Bu örnekteki problemimizde herhangi bir araba galerisinde bulunan arabaların katettiği yolun km olarak arabanın değerine göre hangi durumda olduğunu anlamak için lineer regresyon analizi yöntemiyle sonuç elde edilmek isteniyor.

2.2 Problemden Uygulanan Adımlar

Aynı şekilde bu problemimizde uygulanacak adımları sırasıyla veri seti oluşturulup, kodda çözüm için neler yaptığımızla devam edip, en son olarak programımızın sonuçlarını göstereceğim.

2.2.1 1.Adım : Veri Seti Oluşturma

Lineer regresyonda, bir dizi noktaya en uygun düz çizgiyi veya hiper düzlemi bulmak için kullanılmakta idi. Bir diğer ifadeyle lineer regresyon, en uygun düz çizgi (regresyon çizgisi) kullanarak bağımlı değişken (Y) ile bir veya daha fazla bağımsız değişken (X) arasında bir ilişki kurar. Aşağıdaki grafikte kırmızı çizgi en uygun düz çizgi olarak adlandırılır.

Ben de bu veri setinde arabaların kilometre ömürlerine göre fiyatlarının nasıl bir ilişki içerisinde olacağını göstereceğim.

Hadi veri setimizi görelim:

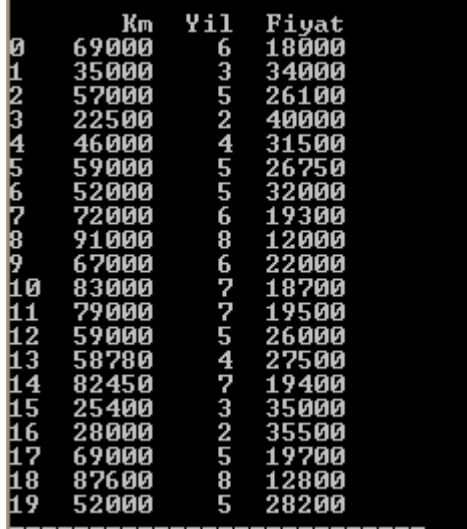
Km	Yıl	Fiyat
69000	6	18000
35000	3	34000
57000	5	26100
22500	2	40000
46000	4	31500
59000	5	26750
52000	5	32000
72000	6	19300
91000	8	12000
67000	6	22000
83000	7	18700
79000	7	19500
59000	5	26000
58780	4	27500
82450	7	19400
25400	3	35000
28000	2	35500
69000	5	19700
87600	8	12800
52000	5	28200

Bu veri setiyle çalıştım. Kodda ise nasıl oluşturulduğunu görebilirsiniz:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 from sklearn.linear_model import LinearRegression
5
6 data = pd.read_csv('fiyatlar.csv') # veriyi alıyoruz
7 X = data.iloc[:, 0].values.reshape(-1, 1) # Verideki degerleri
   numpy dizisine donduruyoruz
8 Y = data.iloc[:, 2].values.reshape(-1, 1) # -1, satirlarin
   boyutunun hesaplandigi, ancak 1 sutun olduğu anlamına gelir
9 print("-----")
10 print(data)
```

Listing 6: Veri Seti Oluşturma

Sonuç olarak da aşağıya bu terminal sonucu gelecektir:



	Km	Yil	Fiyat
0	69000	6	18000
1	35000	3	34000
2	57000	5	26100
3	22500	2	40000
4	46000	4	31500
5	59000	5	26750
6	52000	5	32000
7	72000	6	19300
8	91000	8	12000
9	67000	6	22000
10	83000	7	18700
11	79000	7	19500
12	59000	5	26000
13	58780	4	27500
14	82450	7	19400
15	25400	3	35000
16	28000	2	35500
17	69000	5	19700
18	87600	8	12800
19	52000	5	28200

Figure 5: Veri Seti

2.2.2 2.Adım: Lineer Regresyon Modeli

2.adımda modelimizi oluşturan kodları ve onların sonuçlarını göreceksiniz. Öncelikle analiz için x ve y değerlerimizin olması gerekiyor. Bunları şu şekilde aldım.

```
1 X = data.iloc[:, 0].values.reshape(-1, 1) # Verideki degerleri
   numpy dizisine donduruyoruz
2 Y = data.iloc[:, 2].values.reshape(-1, 1) # -1, satirlarin
   boyutunun hesaplandigi, ancak 1 sutun olduğu anlamına gelir
```

Listing 7: x ve y Değerleri

Daha sonra bu değerleri kullanan eğitim modelimizi oluşturdum.

```
1 linear_regressor = LinearRegression() # model objemizi olusturduk
2 linear_regressor.fit(X, Y) # lineer regresyon modelimizi egittik
3 Y_pred = linear_regressor.predict(X) # tahminler yaptik
4 print(X)#e itti imiz x de erleri
5 print("-----")
6 print(Y_pred)#tahminler
```

Listing 8: Model

Şimdi modelimizi oluşturduk. Modelimizi linearregressor değişkenine tanımladık. Daha sonra eğilecek değerler verildi ve en son da tahmin (predict) işlemleri yapıldı. Bu işlemlerin sonucunda galeride bulunan arabaların km ömürlerine göre fiyatlarının ne seviyede olduğunu ilişkilendirmeye çalıştım. En son da bu değerleri görelim.

```
[21789.35155452]
[34298.34352801]
[26204.2898981]
[38897.23763591]
[30251.31671306]
[25468.46684084]
[28043.84754126]
[20685.61696862]
[13695.29792461]
[22525.17461178]
[16638.59015367]
[18110.23626819]
[25468.46684084]
[25549.40737714]
[16840.94149441]
[37830.29420288]
[36873.72422844]
[21789.35155452]
[14946.19712196]
[28043.84754126]
```

Figure 6: Tahmin değerleri

```
Skor: 0.9354244112393256
```

Figure 7: Doğruluk Skoru

2.2.3 3.Adım: Sonuçlar

```
1 print("Skor: " + str(linear_regressor.score(X,Y)))
2
3 plt.xlabel("Km ler")
4 plt.ylabel("Tahmini fiyatlar")
5 plt.scatter(X, Y)#x ve y de erlerine g re noktalar m z
   grafi e yerle tirdik.
```

```
6 plt.plot(X, Y_pred, color='red')  
7 plt.show()
```

Listing 9: Lineer Regression Sonuç Bastırımı

Sonuç grafiğini de aşağıda görebilirsiniz.

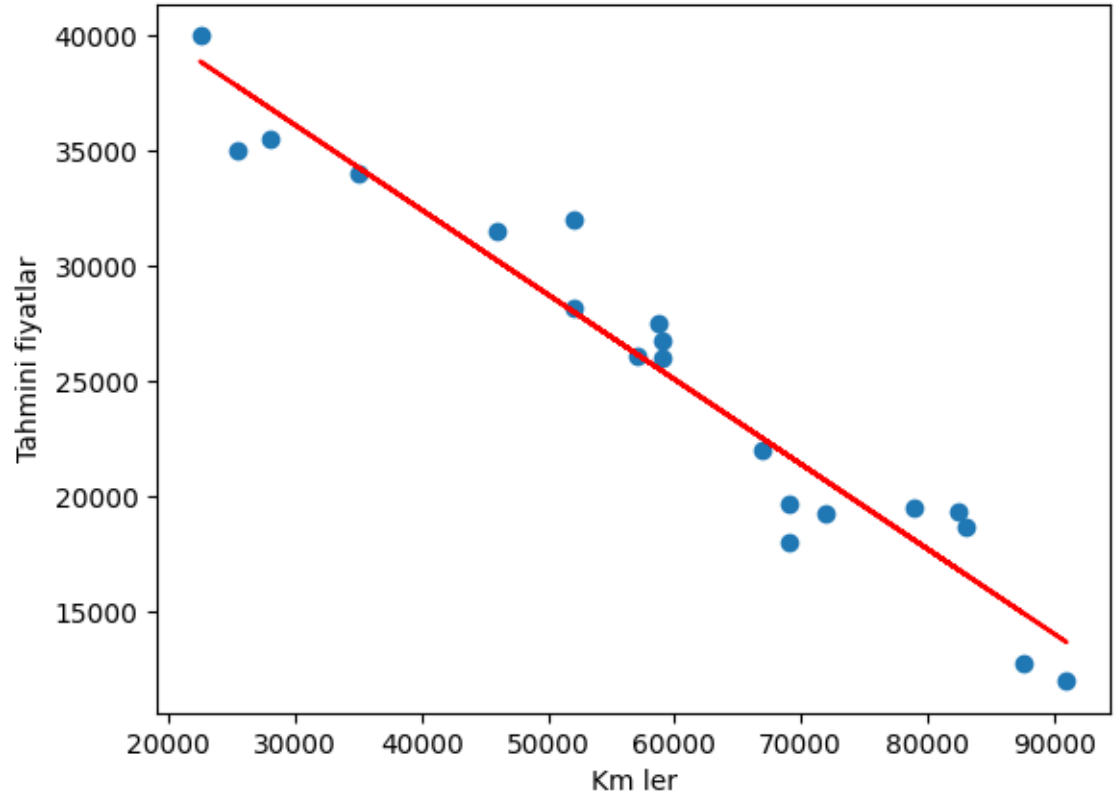


Figure 8: Sonuç Grafiği