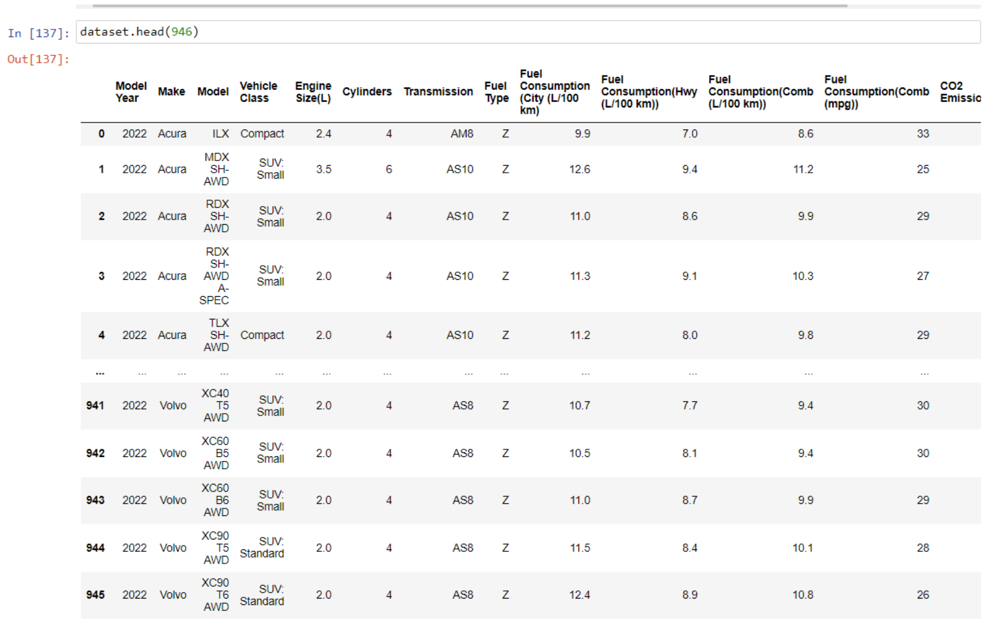
Örüntü Tanıma Arasınav Ödevi

İrem Çakmak

203908028



```
In [137]: dataset.head(946)
Out[137]:
```

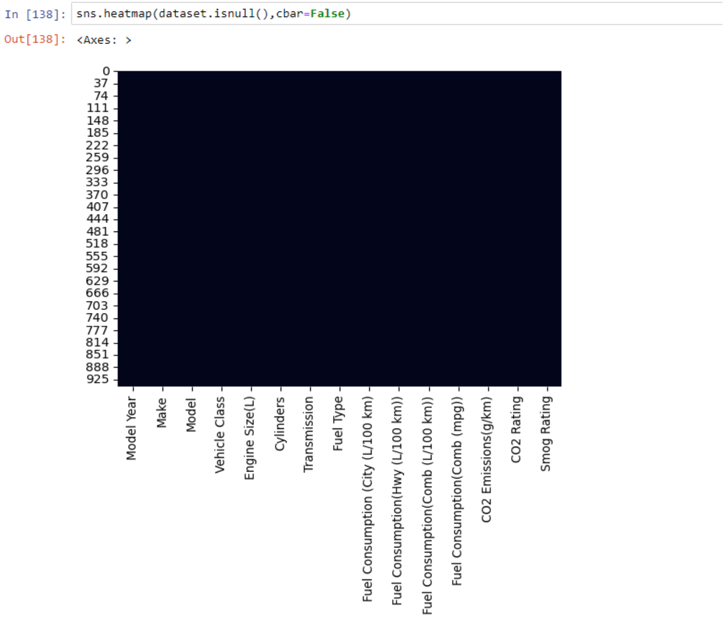
	Model Year	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissic
0	2022	Acura	ILX	Compact	2.4	4	AM8	Z	9.9	7.0	8.6	33	
1	2022	Acura	MDX SH-AWD	SUV: Small	3.5	6	AS10	Z	12.6	9.4	11.2	25	
2	2022	Acura	RDX SH-AWD	SUV: Small	2.0	4	AS10	Z	11.0	8.6	9.9	29	
3	2022	Acura	RDX SH-AWD A-SPEC	SUV: Small	2.0	4	AS10	Z	11.3	9.1	10.3	27	
4	2022	Acura	TLX SH-AWD	Compact	2.0	4	AS10	Z	11.2	8.0	9.8	29	
...	...	...	...	...	...	...	...	...	...	...	...	...	
941	2022	Volvo	XC40 T5 AWD	SUV: Small	2.0	4	AS8	Z	10.7	7.7	9.4	30	
942	2022	Volvo	XC60 B5 AWD	SUV: Small	2.0	4	AS8	Z	10.5	8.1	9.4	30	
943	2022	Volvo	XC60 B6 AWD	SUV: Small	2.0	4	AS8	Z	11.0	8.7	9.9	29	
944	2022	Volvo	XC90 T5 AWD	SUV: Standard	2.0	4	AS8	Z	11.5	8.4	10.1	28	
945	2022	Volvo	XC90 T6 AWD	SUV: Standard	2.0	4	AS8	Z	12.4	8.9	10.8	26	

Veri önişleme:

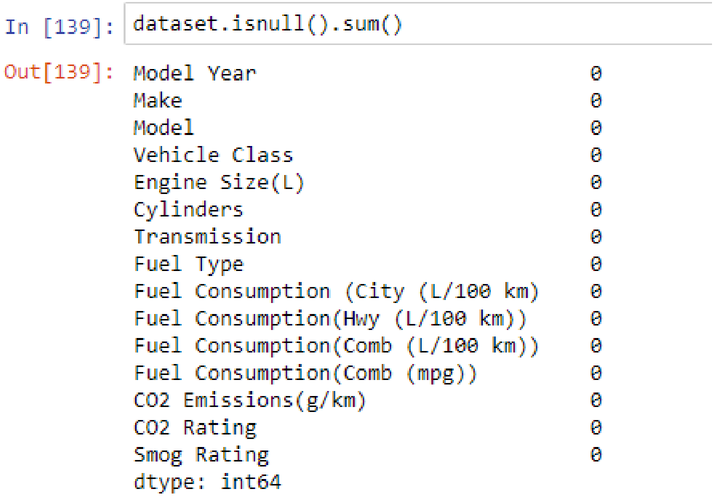
1. **Verisetinde eksik değer kontrolü yapılacaktır, özelliklerdeki sözel ifadeler sayısal değerlere çevirilecektir (onehotencoding),**
2. **Normalizasyon işlemi gerçekleştirilecektir (Standart veya MinMax normalizasyon)**

1. Tablo şeklindeki bir veri kümesinde eksik değerler varsa, sns.heatmap() fonksiyonu kullanılarak bu eksik değerler bir ısı haritası şeklinde görselleştirilebilir.

```
In [138]: sns.heatmap(dataset.isnull(),cbar=False)
Out[138]: <Axes: >
```



Bu kod satırı, bir veri kümesindeki her sütunda kaç tane eksik değer olduğunu hesaplar ve bu sayıları toplayarak toplam eksik değer sayısını verir.

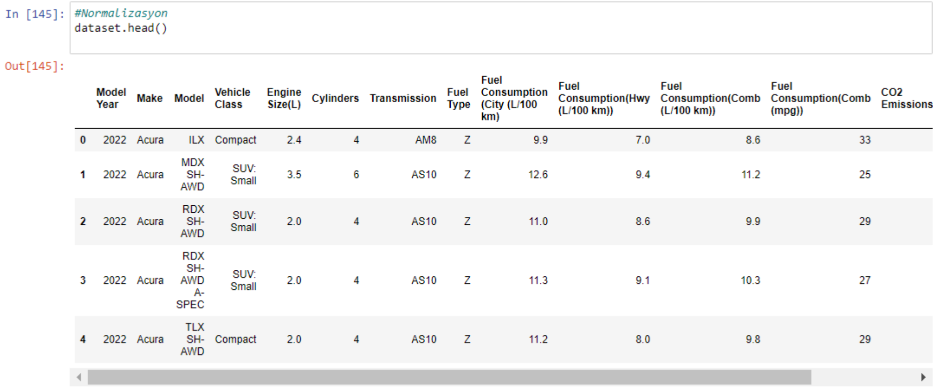


```
In [139]: dataset.isnull().sum()

Out[139]: Model Year                            0
          Make                                  0
          Model                                 0
          Vehicle Class                         0
          Engine Size(L)                        0
          Cylinders                             0
          Transmission                          0
          Fuel Type                             0
          Fuel Consumption (City (L/100 km)     0
          Fuel Consumption(Hwy (L/100 km))      0
          Fuel Consumption(Comb (L/100 km))     0
          Fuel Consumption(Comb (mpg))          0
          CO2 Emissions(g/km)                   0
          CO2 Rating                            0
          Smog Rating                           0
          dtype: int64
```

Burada da eksik verinin olmadığını gösteriyor.

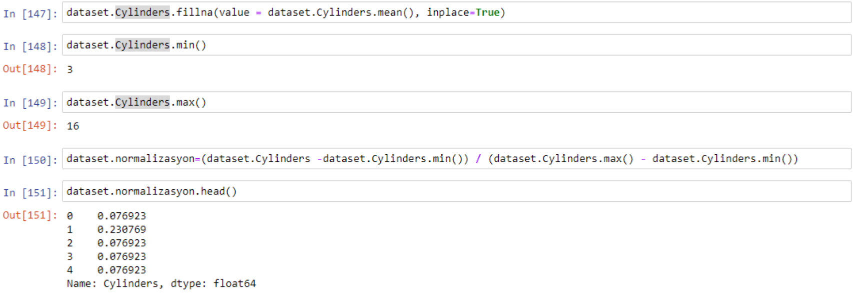
2.

dataset.head() yaparak tablomu listeledim. Bu kod satırı, bir veri kümesindeki "Cylinders" sütunundaki eksik değerleri ortalama değer ile doldurur. Yani, "Cylinders" sütunundaki eksik değerler, o sütundaki diğer verilerin ortalaması alınarak yerine konulur. "inplace=True" parametresi, yapılan değişikliklerin kalıcı olmasını sağlar, yani veri kümesi üzerinde değişiklik yapar. "Cylinders"sütünündaki max min değerlerini alarak dataset.normalizasyon formülüze edip head() yaparak listeledim.



```
In [145]: #Normalizasyon
          dataset.head()

Out[145]:
```

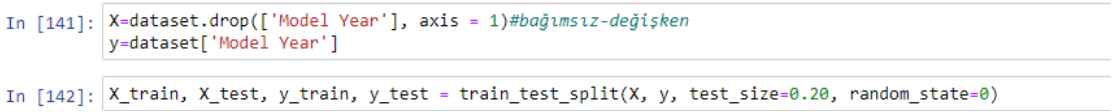
	Model Year	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions
0	2022	Acura	ILX	Compact	2.4	4	AM8	Z	9.9	7.0	8.6	33	
1	2022	Acura	MDX SH-AWD	SUV: Small	3.5	6	AS10	Z	12.6	9.4	11.2	25	
2	2022	Acura	RDX SH-AWD	SUV: Small	2.0	4	AS10	Z	11.0	8.6	9.9	29	
3	2022	Acura	RDX SH-AWD A-SPEC	SUV: Small	2.0	4	AS10	Z	11.3	9.1	10.3	27	
4	2022	Acura	TLX SH-AWD	Compact	2.0	4	AS10	Z	11.2	8.0	9.8	29	



```
In [147]: dataset.Cylinders.fillna(value = dataset.Cylinders.mean(), inplace=True)

In [148]: dataset.Cylinders.min()

Out[148]: 3

In [149]: dataset.Cylinders.max()

Out[149]: 16

In [150]: dataset.normalizasyon=(dataset.Cylinders -dataset.Cylinders.min()) / (dataset.Cylinders.max() - dataset.Cylinders.min())

In [151]: dataset.normalizasyon.head()

Out[151]: 0    0.076923
          1    0.230769
          2    0.076923
          3    0.076923
          4    0.076923
          Name: Cylinders, dtype: float64
```

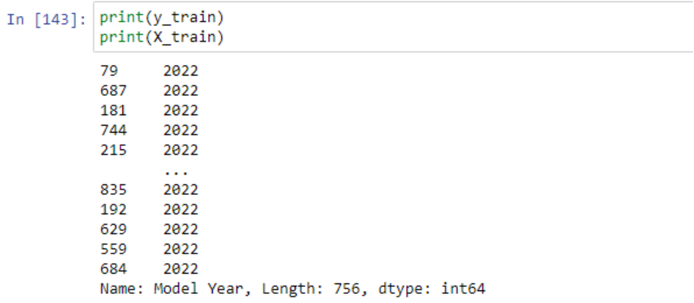
2.Versetinin eğitim ve test olarak ayrılması

Tablodaki veriler arasında ilişki bulmak için, "Model Year" sütunundaki verileri bağımlı değişken olarak ayırıp geri kalan sütunlardaki verileri bağımsız değişken olarak ayırmak isteyebiliriz. Bu kodlar, veri kümesinden "Model Year" sütununu çıkararak bağımsız değişkenleri (X) ve bağımlı değişkeni (y) belirler. "drop()" fonksiyonu kullanılarak, belirtilen sütun veri kümesinden kaldırılır. "axis=1" parametresi, işlemin sütun bazında yapılacağını belirtir.

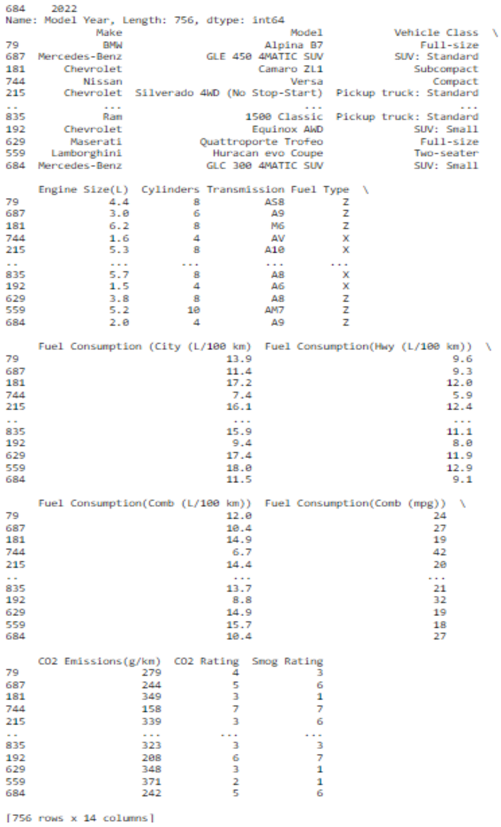


```
In [141]: X=dataset.drop(['Model Year'], axis = 1)#bağımsız-değişken
          y=dataset['Model Year']

In [142]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
```

Bu kod satırı, veri kümesini eğitim ve test setlerine ayırmak için kullanılır.



```
In [143]: print(y_train)
          print(X_train)

          79     2022
          687    2022
          181    2022
          744    2022
          215    2022
                 ...
          835    2022
          192    2022
          629    2022
          559    2022
          684    2022
          Name: Model Year, Length: 756, dtype: int64
```

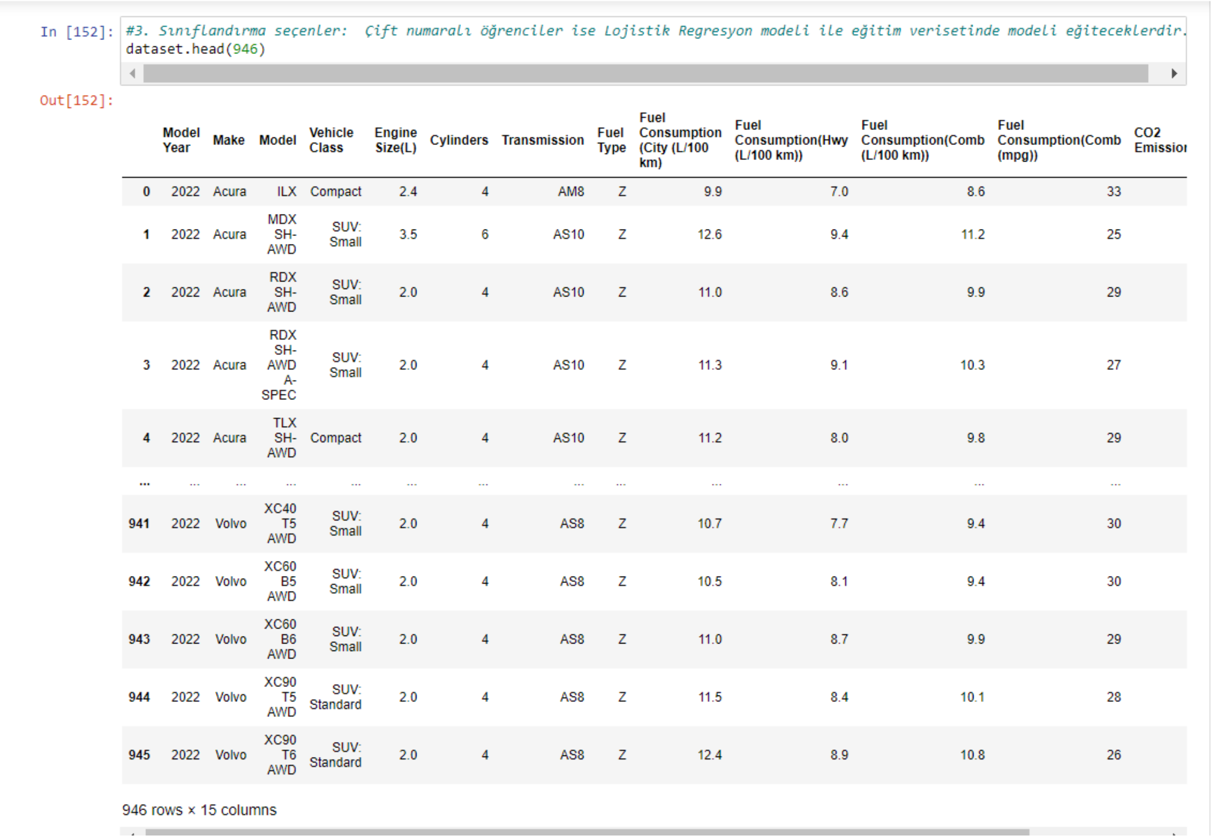


```
684    2022
Name: Model Year, Length: 756, dtype: int64
              Make                           Model          Vehicle Class  \
79             BMW                       Alpina B7              Full-size   
687  Mercedes-Benz               GLE 450 4MATIC SUV          SUV: Standard   
181      Chevrolet                     Camaro ZL1             Subcompact   
744         Nissan                          Versa                Compact   
215      Chevrolet  Silverado 4WD (No Stop-Start)  Pickup truck: Standard   
..             ...                            ...                    ...   
835            Ram                   1500 Classic  Pickup truck: Standard   
192      Chevrolet                    Equinox AWD             SUV: Small   
629       Maserati           Quattroporte Trofeo              Full-size   
559    Lamborghini             Huracan evo Coupe             Two-seater   
684  Mercedes-Benz               GLC 300 4MATIC SUV             SUV: Small   

     Engine Size(L)  Cylinders Transmission Fuel Type  \
79              4.4          8          AS8         Z   
687             3.0          6           A9         Z   
181             6.2          8           M6         Z   
744             1.6          4           AV         X   
215             5.3          8          A10         X   
..              ...        ...          ...       ...   
835             5.7          8           A8         X   
192             1.5          4           A6         X   
629             3.8          8           A8         Z   
559             5.2         10          AM7         Z   
684             2.0          4           A9         Z   

     Fuel Consumption (City (L/100 km)  Fuel Consumption(Hwy (L/100 km))  \
79                                13.9                               9.6   
687                               11.4                               9.3   
181                               17.2                              12.0   
744                                7.4                               5.9   
215                               16.1                              12.4   
..                                 ...                               ...   
835                               15.9                              11.1   
192                                9.4                               8.0   
629                               17.4                              11.9   
559                               18.0                              12.9   
684                               11.5                               9.1   

     Fuel Consumption(Comb (L/100 km))  Fuel Consumption(Comb (mpg))  \
79                                12.0                            24   
687                               10.4                            27   
181                               14.9                            19   
744                                6.7                            42   
215                               14.4                            20   
..                                 ...                           ...   
835                               13.7                            21   
192                                8.8                            32   
629                               14.9                            19   
559                               15.7                            18   
684                               10.4                            27   

     CO2 Emissions(g/km)  CO2 Rating  Smog Rating  
79                   279           4            3  
687                  244           5            6  
181                  349           3            1  
744                  158           7            7  
215                  339           3            6  
..                   ...         ...          ...  
835                  323           3            3  
192                  208           6            7  
629                  348           3            1  
559                  371           2            1  
684                  242           5            6  

[756 rows x 14 columns]
```

3. Sınıflandırma seçenler:

1. **Tek numaralı öğrenciler NaiveBayes, Çift numaralı öğrenciler ise Lojistik Regresyon modeli ile eğitim verisetinde modeli eğiteceklerdir.**
2. **Regresyon seçenler : Doğrusal regresyon (genelde çok değişkenli doğrusal regresyon) modelinde eğitim verisetini eğitecekler.**

1.Lojistik fonsiyonu girilen değerleri 0 ile 1 arasına getirir.one hot encoding kategorik değişkenleri 0-1 değişkene dönüştürdüm. Join yaparak tablolarla birleştirdim. Ve String değerleri one hot encoding işlemi yaparak yeni datasetler oluşturdum. Yeni datasetimle lojistik regrasyon uyguladım.

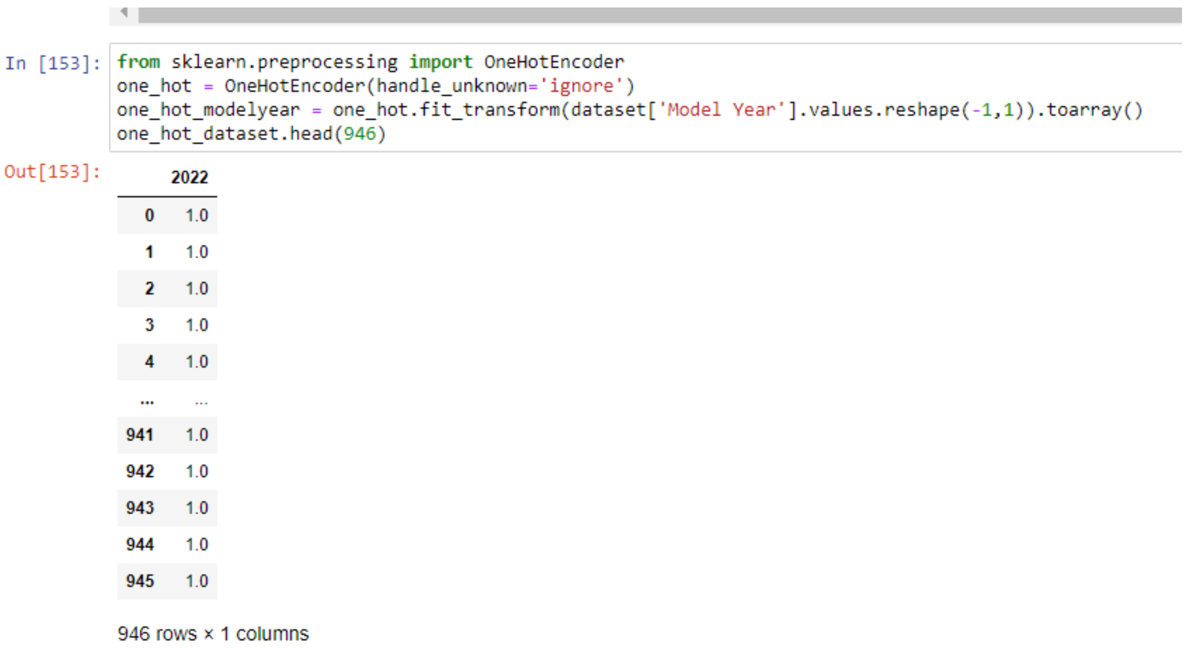


```
In [152]: #3. Sınıflandırma seçenler:  Çift numaralı öğrenciler ise Lojistik Regresyon modeli ile eğitim verisetinde modeli eğiteceklerdir.
          dataset.head(946)
```

Out[152]:

	Model Year	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissior
0	2022	Acura	ILX	Compact	2.4	4	AM8	Z	9.9	7.0	8.6	33	
1	2022	Acura	MDX SH-AWD	SUV: Small	3.5	6	AS10	Z	12.6	9.4	11.2	25	
2	2022	Acura	RDX SH-AWD	SUV: Small	2.0	4	AS10	Z	11.0	8.6	9.9	29	
3	2022	Acura	RDX SH-AWD A-SPEC	SUV: Small	2.0	4	AS10	Z	11.3	9.1	10.3	27	
4	2022	Acura	TLX SH-AWD	Compact	2.0	4	AS10	Z	11.2	8.0	9.8	29	
...	...	...	...	...	...	...	...	...	...	...	...	...	
941	2022	Volvo	XC40 T5 AWD	SUV: Small	2.0	4	AS8	Z	10.7	7.7	9.4	30	
942	2022	Volvo	XC60 B5 AWD	SUV: Small	2.0	4	AS8	Z	10.5	8.1	9.4	30	
943	2022	Volvo	XC60 B6 AWD	SUV: Small	2.0	4	AS8	Z	11.0	8.7	9.9	29	
944	2022	Volvo	XC90 T5 AWD	SUV: Standard	2.0	4	AS8	Z	11.5	8.4	10.1	28	
945	2022	Volvo	XC90 T6 AWD	SUV: Standard	2.0	4	AS8	Z	12.4	8.9	10.8	26	

946 rows × 15 columns

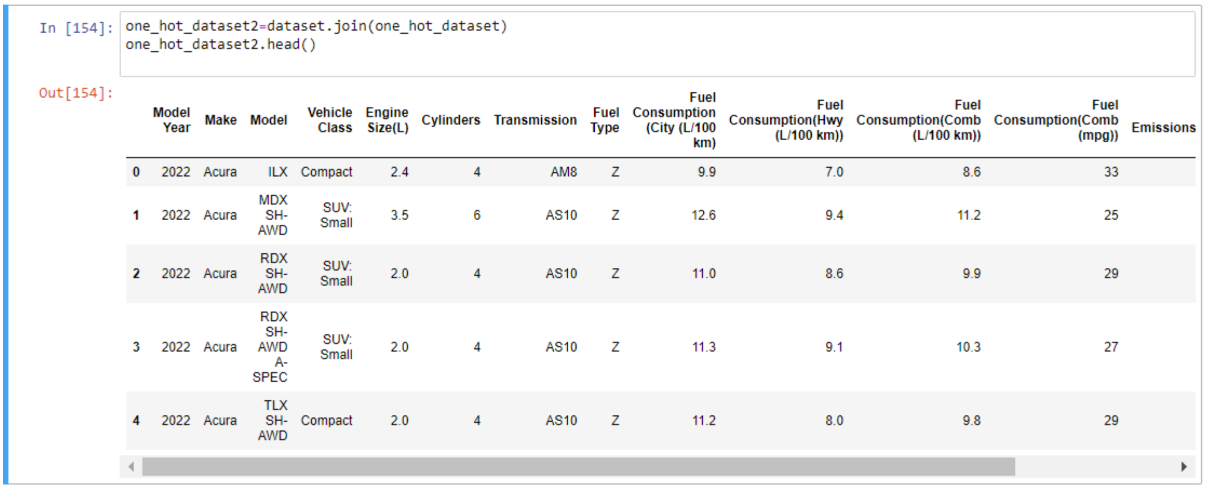


In [153]:
```
from sklearn.preprocessing import OneHotEncoder
one_hot = OneHotEncoder(handle_unknown='ignore')
one_hot_modelyear = one_hot.fit_transform(dataset['Model Year'].values.reshape(-1,1)).toarray()
one_hot_dataset.head(946)
```

Out[153]:

	2022
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0
...	...
941	1.0
942	1.0
943	1.0
944	1.0
945	1.0

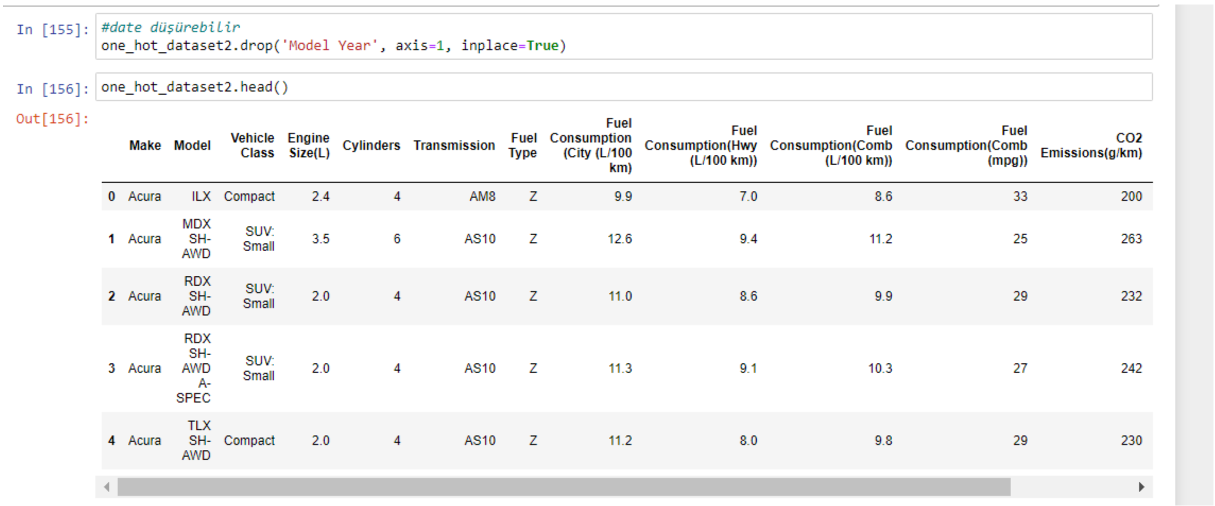
946 rows × 1 columns



In [154]:
```
one_hot_dataset2=dataset.join(one_hot_dataset)
one_hot_dataset2.head()
```

Out[154]:

	Model Year	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	Emissions
0	2022	Acura	ILX	Compact	2.4	4	AM8	Z	9.9	7.0	8.6	33	
1	2022	Acura	MDX SH-AWD	SUV: Small	3.5	6	AS10	Z	12.6	9.4	11.2	25	
2	2022	Acura	RDX SH-AWD	SUV: Small	2.0	4	AS10	Z	11.0	8.6	9.9	29	
3	2022	Acura	RDX SH-AWD A-SPEC	SUV: Small	2.0	4	AS10	Z	11.3	9.1	10.3	27	
4	2022	Acura	TLX SH-AWD	Compact	2.0	4	AS10	Z	11.2	8.0	9.8	29	

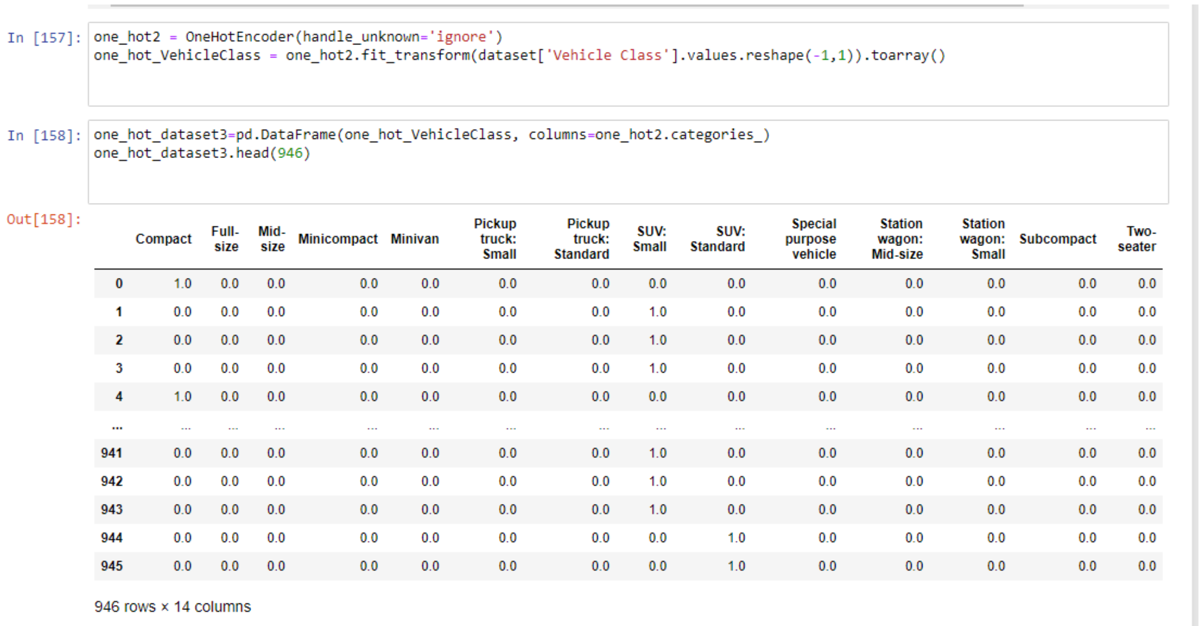


In [155]:
```
#date düşürebilir
one_hot_dataset2.drop('Model Year', axis=1, inplace=True)
```

In [156]:
```
one_hot_dataset2.head()
```

Out[156]:

	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)
0	Acura	ILX	Compact	2.4	4	AM8	Z	9.9	7.0	8.6	33	200
1	Acura	MDX SH-AWD	SUV: Small	3.5	6	AS10	Z	12.6	9.4	11.2	25	263
2	Acura	RDX SH-AWD	SUV: Small	2.0	4	AS10	Z	11.0	8.6	9.9	29	232
3	Acura	RDX SH-AWD A-SPEC	SUV: Small	2.0	4	AS10	Z	11.3	9.1	10.3	27	242
4	Acura	TLX SH-AWD	Compact	2.0	4	AS10	Z	11.2	8.0	9.8	29	230



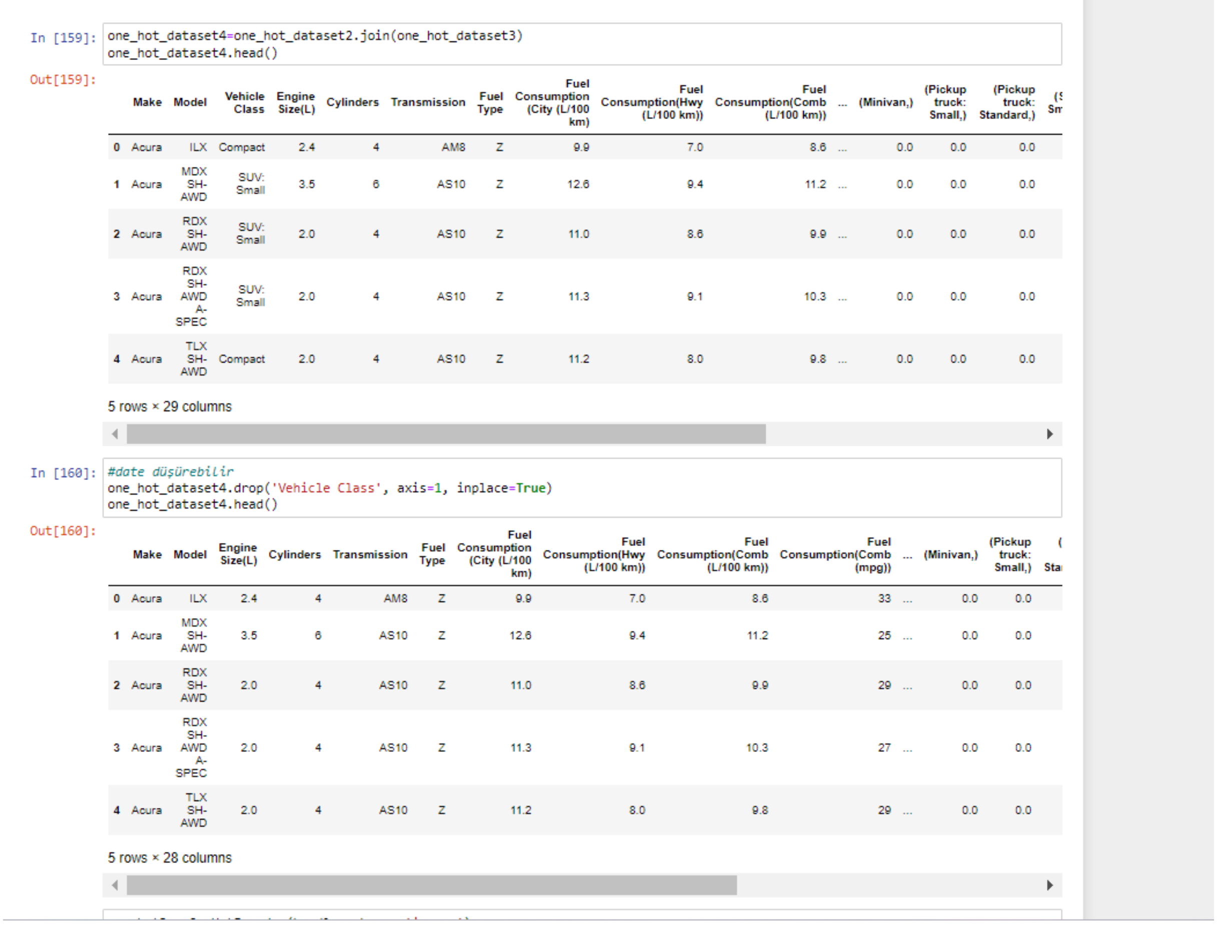
In [157]:
```
one_hot2 = OneHotEncoder(handle_unknown='ignore')
one_hot_VehicleClass = one_hot2.fit_transform(dataset['Vehicle Class'].values.reshape(-1,1)).toarray()
```

In [158]:
```
one_hot_dataset3=pd.DataFrame(one_hot_VehicleClass, columns=one_hot2.categories_)
one_hot_dataset3.head(946)
```

Out[158]:

	Compact	Full-size	Mid-size	Minicompact	Minivan	Pickup truck: Small	Pickup truck: Standard	SUV: Small	SUV: Standard	Special purpose vehicle	Station wagon: Mid-size	Station wagon: Small	Subcompact	Two-seater
0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
941	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
942	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
943	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
944	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
945	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0

946 rows × 14 columns



In [159]:
```
one_hot_dataset4=one_hot_dataset2.join(one_hot_dataset3)
one_hot_dataset4.head()
```

Out[159]:

	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	...	(Minivan,)	(Pickup truck: Small,)	(Pickup truck: Standard,)	(S Sm
0	Acura	ILX	Compact	2.4	4	AM8	Z	9.9	7.0	8.6	...	0.0	0.0	0.0	
1	Acura	MDX SH-AWD	SUV: Small	3.5	6	AS10	Z	12.6	9.4	11.2	...	0.0	0.0	0.0	
2	Acura	RDX SH-AWD	SUV: Small	2.0	4	AS10	Z	11.0	8.6	9.9	...	0.0	0.0	0.0	
3	Acura	RDX SH-AWD A-SPEC	SUV: Small	2.0	4	AS10	Z	11.3	9.1	10.3	...	0.0	0.0	0.0	
4	Acura	TLX SH-AWD	Compact	2.0	4	AS10	Z	11.2	8.0	9.8	...	0.0	0.0	0.0	

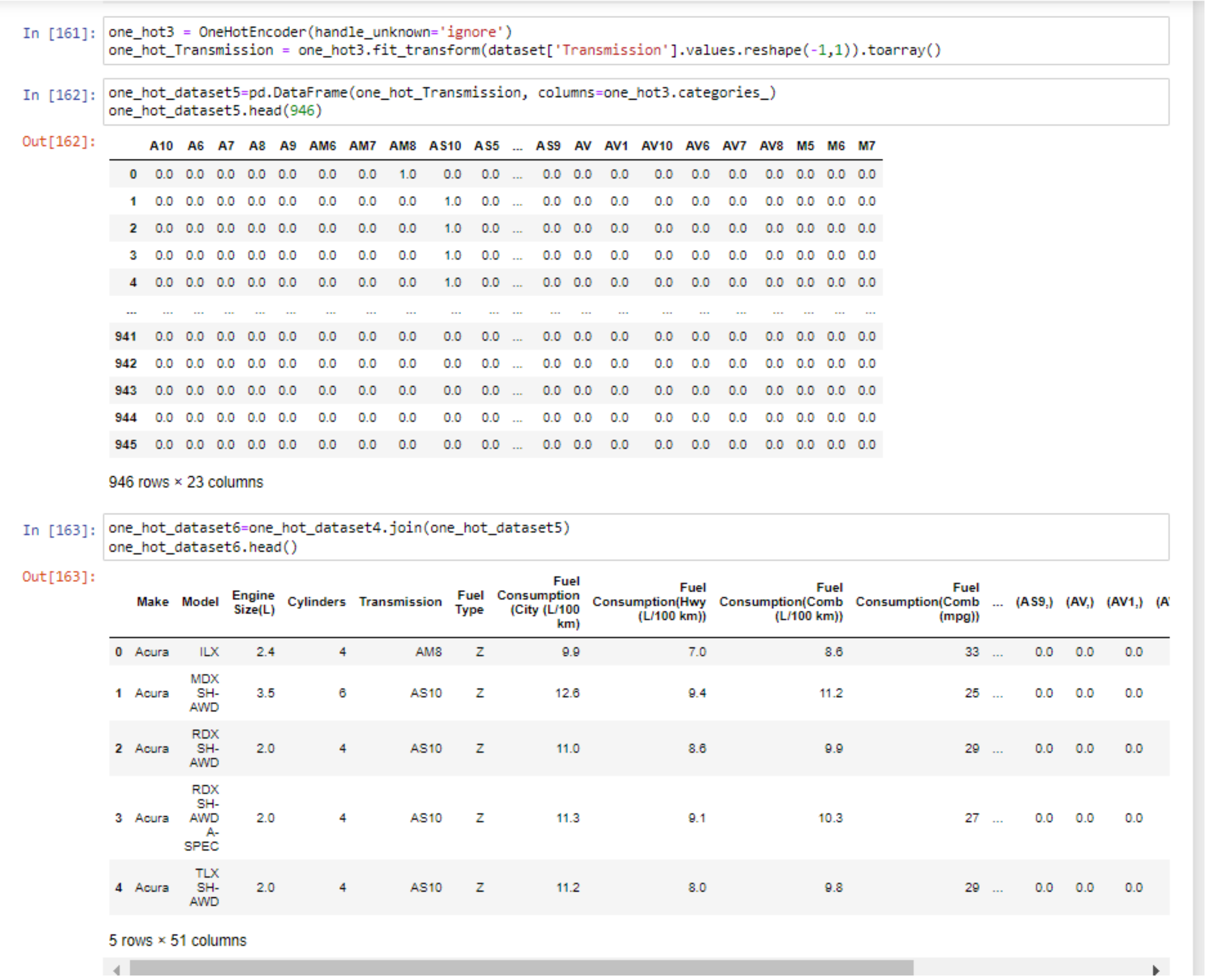
5 rows × 29 columns

In [160]:
```
#date düşürebilir
one_hot_dataset4.drop('Vehicle Class', axis=1, inplace=True)
one_hot_dataset4.head()
```

Out[160]:

	Make	Model	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	...	(Minivan,)	(Pickup truck: Small,)	(Sta
0	Acura	ILX	2.4	4	AM8	Z	9.9	7.0	8.6	33	...	0.0	0.0	
1	Acura	MDX SH-AWD	3.5	6	AS10	Z	12.6	9.4	11.2	25	...	0.0	0.0	
2	Acura	RDX SH-AWD	2.0	4	AS10	Z	11.0	8.6	9.9	29	...	0.0	0.0	
3	Acura	RDX SH-AWD A-SPEC	2.0	4	AS10	Z	11.3	9.1	10.3	27	...	0.0	0.0	
4	Acura	TLX SH-AWD	2.0	4	AS10	Z	11.2	8.0	9.8	29	...	0.0	0.0	

5 rows × 28 columns



In [161]:
```
one_hot3 = OneHotEncoder(handle_unknown='ignore')
one_hot_Transmission = one_hot3.fit_transform(dataset['Transmission'].values.reshape(-1,1)).toarray()
```

In [162]:
```
one_hot_dataset5=pd.DataFrame(one_hot_Transmission, columns=one_hot3.categories_)
one_hot_dataset5.head(946)
```

Out[162]:

	A10	A6	A7	A8	A9	AM6	AM7	AM8	AS10	AS5	...	AS9	AV	AV1	AV10	AV6	AV7	AV8	M5	M6	M7
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
941	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
942	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
943	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
944	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
945	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

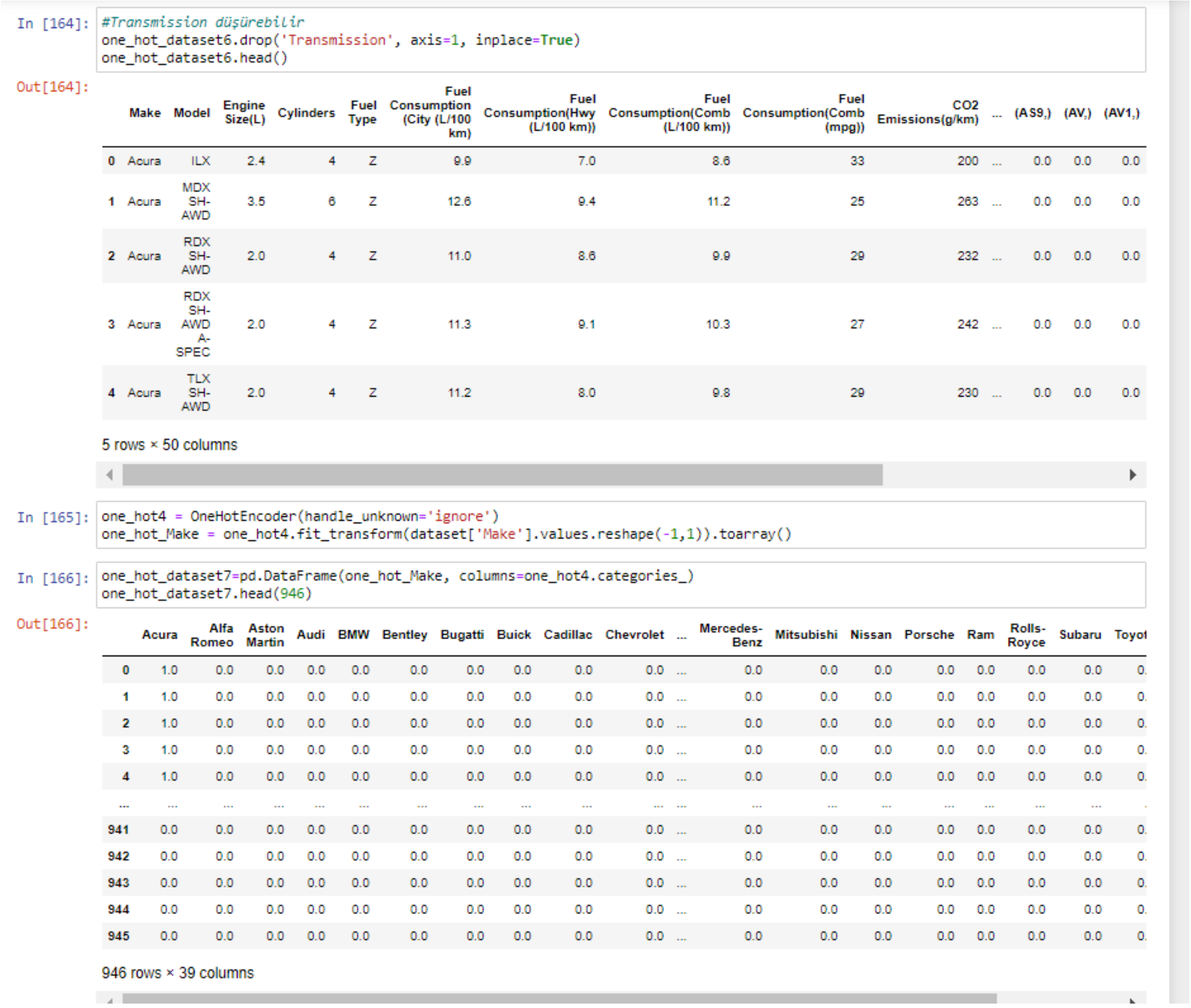
946 rows × 23 columns

In [163]:
```
one_hot_dataset6=one_hot_dataset4.join(one_hot_dataset5)
one_hot_dataset6.head()
```

Out[163]:

	Make	Model	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	...	(AS9,)	(AV,)	(AV1,)	(A
0	Acura	ILX	2.4	4	AM8	Z	9.9	7.0	8.6	33	...	0.0	0.0	0.0	
1	Acura	MDX SH-AWD	3.5	6	AS10	Z	12.6	9.4	11.2	25	...	0.0	0.0	0.0	
2	Acura	RDX SH-AWD	2.0	4	AS10	Z	11.0	8.6	9.9	29	...	0.0	0.0	0.0	
3	Acura	RDX SH-AWD A-SPEC	2.0	4	AS10	Z	11.3	9.1	10.3	27	...	0.0	0.0	0.0	
4	Acura	TLX SH-AWD	2.0	4	AS10	Z	11.2	8.0	9.8	29	...	0.0	0.0	0.0	

5 rows × 51 columns



In [164]:
```
#Transmission düşürebilir
one_hot_dataset6.drop('Transmission', axis=1, inplace=True)
one_hot_dataset6.head()
```

Out[164]:

	Make	Model	Engine Size(L)	Cylinders	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	...	(AS9,)	(AV,)	(AV1,)
0	Acura	ILX	2.4	4	Z	9.9	7.0	8.6	33	200	...	0.0	0.0	0.0
1	Acura	MDX SH-AWD	3.5	6	Z	12.6	9.4	11.2	25	263	...	0.0	0.0	0.0
2	Acura	RDX SH-AWD	2.0	4	Z	11.0	8.6	9.9	29	232	...	0.0	0.0	0.0
3	Acura	RDX SH-AWD A-SPEC	2.0	4	Z	11.3	9.1	10.3	27	242	...	0.0	0.0	0.0
4	Acura	TLX SH-AWD	2.0	4	Z	11.2	8.0	9.8	29	230	...	0.0	0.0	0.0

5 rows × 50 columns

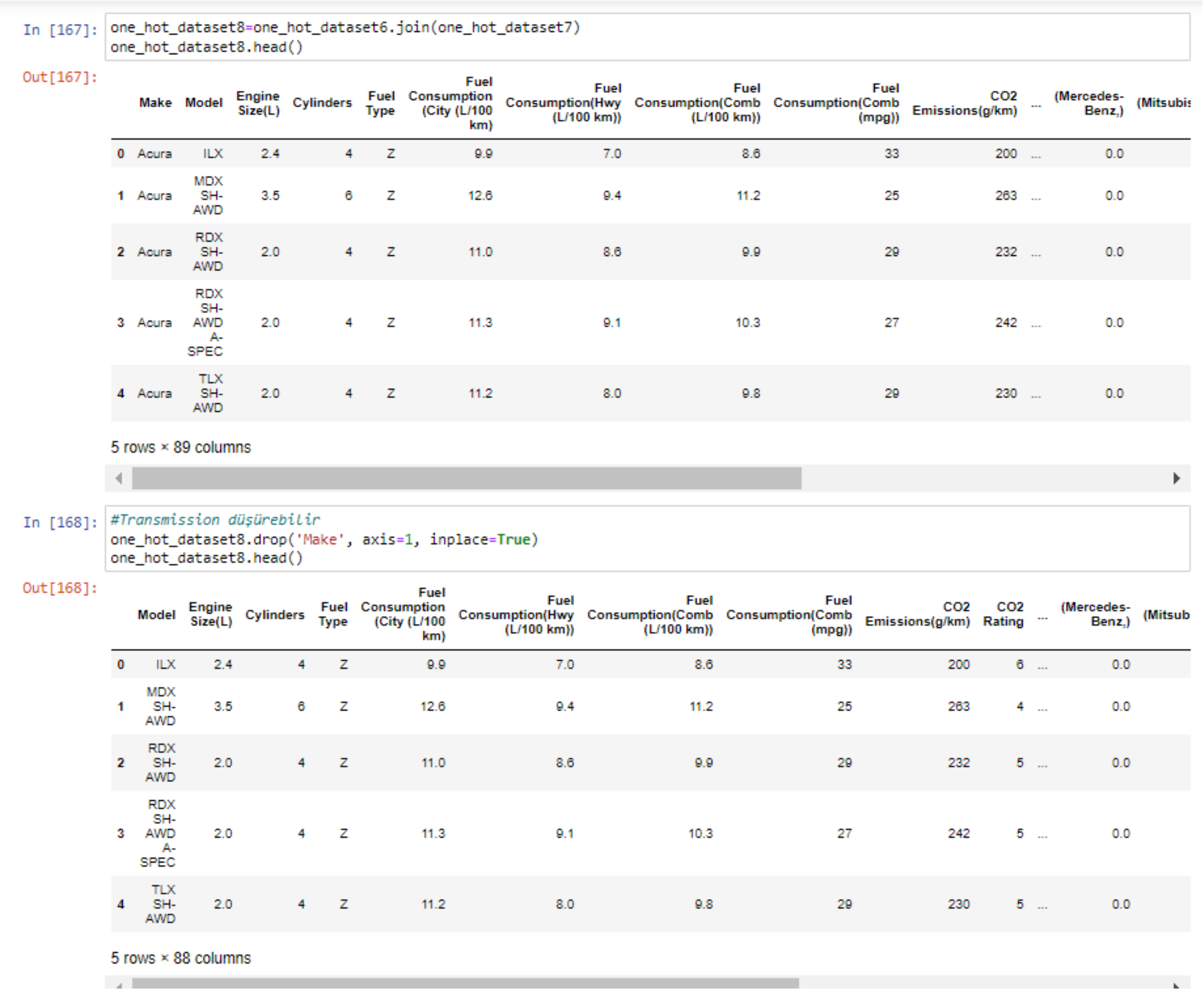
In [165]:
```
one_hot4 = OneHotEncoder(handle_unknown='ignore')
one_hot_Make = one_hot4.fit_transform(dataset['Make'].values.reshape(-1,1)).toarray()
```

In [166]:
```
one_hot_dataset7=pd.DataFrame(one_hot_Make, columns=one_hot4.categories_)
one_hot_dataset7.head(946)
```

Out[166]:

	Acura	Alfa Romeo	Aston Martin	Audi	BMW	Bentley	Bugatti	Buick	Cadillac	Chevrolet	...	Mercedes-Benz	Mitsubishi	Nissan	Porsche	Ram	Rolls-Royce	Subaru	Toyot
0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
2	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
3	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
4	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
941	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
942	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
943	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
944	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
945	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.

946 rows × 39 columns



In [167]:
```
one_hot_dataset8=one_hot_dataset6.join(one_hot_dataset7)
one_hot_dataset8.head()
```

Out[167]:

	Make	Model	Engine Size(L)	Cylinders	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	...	(Mercedes-Benz,)	(Mitsubis
0	Acura	ILX	2.4	4	Z	9.9	7.0	8.6	33	200	...	0.0	
1	Acura	MDX SH-AWD	3.5	6	Z	12.6	9.4	11.2	25	263	...	0.0	
2	Acura	RDX SH-AWD	2.0	4	Z	11.0	8.6	9.9	29	232	...	0.0	
3	Acura	RDX SH-AWD A-SPEC	2.0	4	Z	11.3	9.1	10.3	27	242	...	0.0	
4	Acura	TLX SH-AWD	2.0	4	Z	11.2	8.0	9.8	29	230	...	0.0	

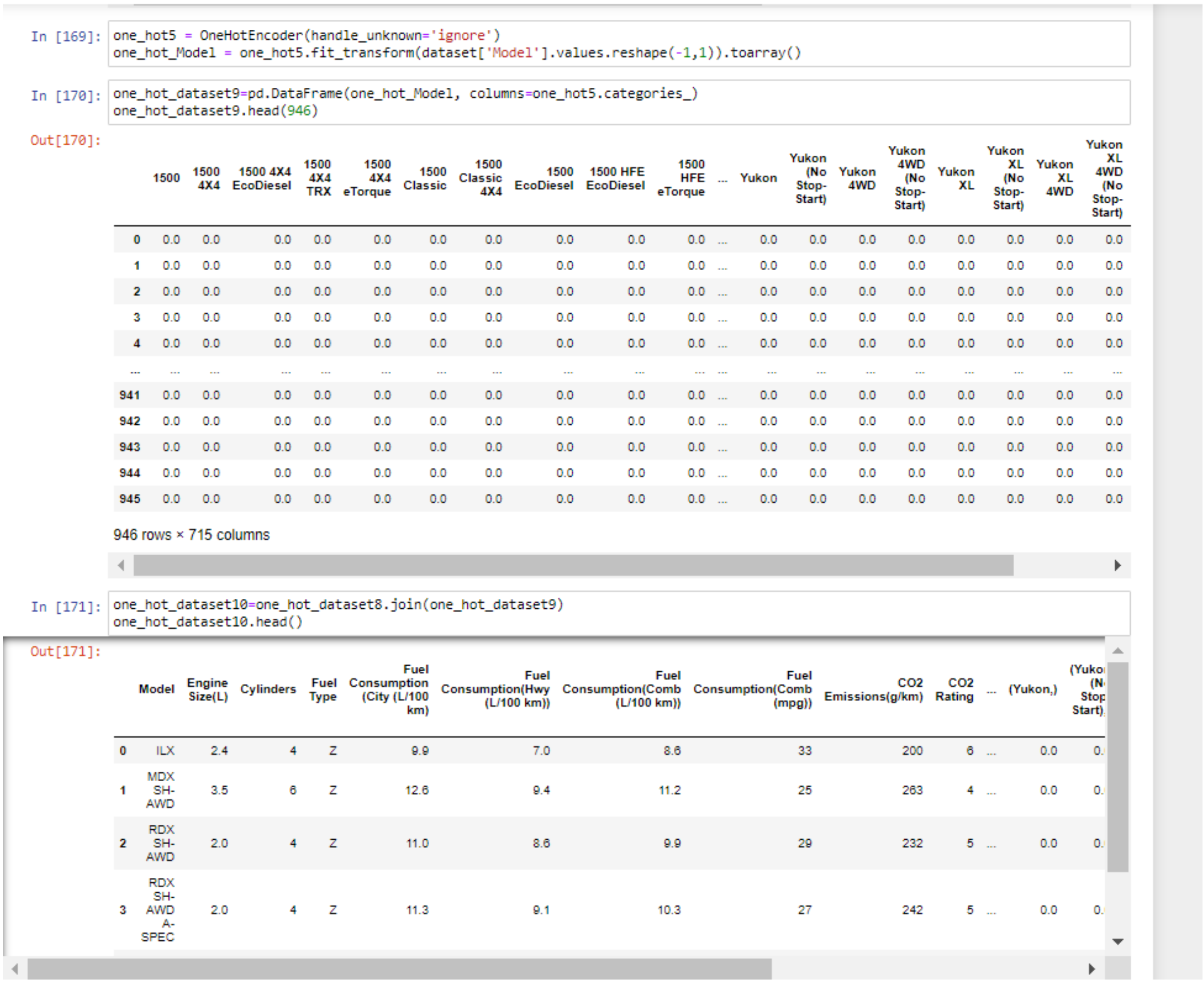
5 rows × 89 columns

In [168]:
```
#Transmission düşürebilir
one_hot_dataset8.drop('Make', axis=1, inplace=True)
one_hot_dataset8.head()
```

Out[168]:

	Model	Engine Size(L)	Cylinders	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	CO2 Rating	...	(Mercedes-Benz,)	(Mitsub
0	ILX	2.4	4	Z	9.9	7.0	8.6	33	200	6	...	0.0	
1	MDX SH-AWD	3.5	6	Z	12.6	9.4	11.2	25	263	4	...	0.0	
2	RDX SH-AWD	2.0	4	Z	11.0	8.6	9.9	29	232	5	...	0.0	
3	RDX SH-AWD A-SPEC	2.0	4	Z	11.3	9.1	10.3	27	242	5	...	0.0	
4	TLX SH-AWD	2.0	4	Z	11.2	8.0	9.8	29	230	5	...	0.0	

5 rows × 88 columns



In [169]:
```
one_hot5 = OneHotEncoder(handle_unknown='ignore')
one_hot_Model = one_hot5.fit_transform(dataset['Model'].values.reshape(-1,1)).toarray()
```

In [170]:
```
one_hot_dataset9=pd.DataFrame(one_hot_Model, columns=one_hot5.categories_)
one_hot_dataset9.head(946)
```

Out[170]:

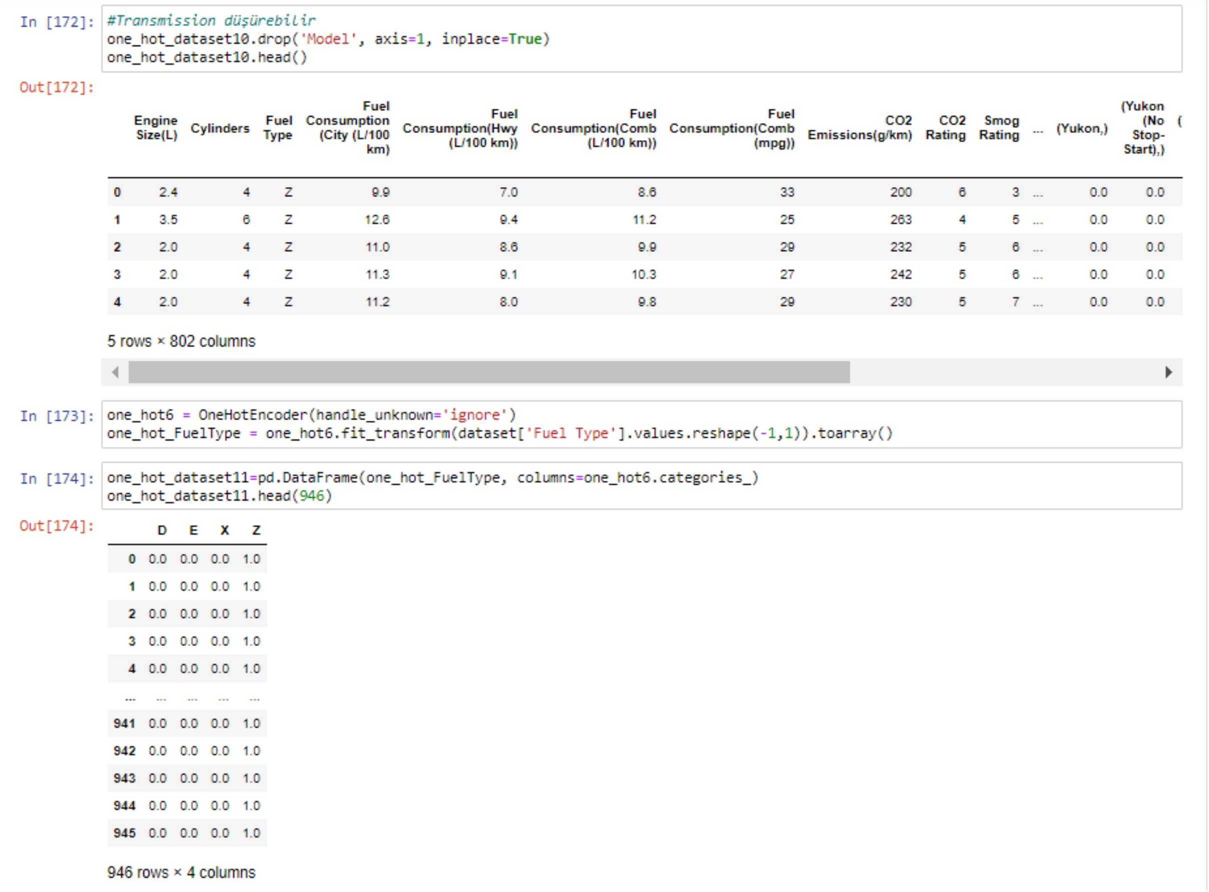
	1500	1500 4X4	1500 4X4 EcoDiesel	1500 4X4 TRX	1500 4X4 eTorque	1500 Classic	1500 Classic 4X4	1500 EcoDiesel	1500 HFE EcoDiesel	1500 HFE eTorque	...	Yukon	Yukon (No Stop-Start)	Yukon 4WD	Yukon 4WD (No Stop-Start)	Yukon XL	Yukon XL (No Stop-Start)	Yukon XL 4WD	Yukon XL 4WD (No Stop-Start)
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
941	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
942	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
943	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
944	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
945	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

946 rows × 715 columns

In [171]:
```
one_hot_dataset10=one_hot_dataset8.join(one_hot_dataset9)
one_hot_dataset10.head()
```

Out[171]:

	Model	Engine Size(L)	Cylinders	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	CO2 Rating	...	(Yukon,)	(Yuko (N Stop Start),
0	ILX	2.4	4	Z	9.9	7.0	8.6	33	200	6	...	0.0	0.
1	MDX SH-AWD	3.5	6	Z	12.6	9.4	11.2	25	263	4	...	0.0	0.
2	RDX SH-AWD	2.0	4	Z	11.0	8.6	9.9	29	232	5	...	0.0	0.
3	RDX SH-AWD A-SPEC	2.0	4	Z	11.3	9.1	10.3	27	242	5	...	0.0	0.



In [172]:
```
#Transmission düşürebilir
one_hot_dataset10.drop('Model', axis=1, inplace=True)
one_hot_dataset10.head()
```

Out[172]:

	Engine Size(L)	Cylinders	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	CO2 Rating	Smog Rating	...	(Yukon,)	(Yukon (No Stop-Start),)	(
0	2.4	4	Z	9.9	7.0	8.6	33	200	6	3	...	0.0	0.0	
1	3.5	6	Z	12.6	9.4	11.2	25	263	4	5	...	0.0	0.0	
2	2.0	4	Z	11.0	8.6	9.9	29	232	5	6	...	0.0	0.0	
3	2.0	4	Z	11.3	9.1	10.3	27	242	5	6	...	0.0	0.0	
4	2.0	4	Z	11.2	8.0	9.8	29	230	5	7	...	0.0	0.0	

5 rows × 802 columns

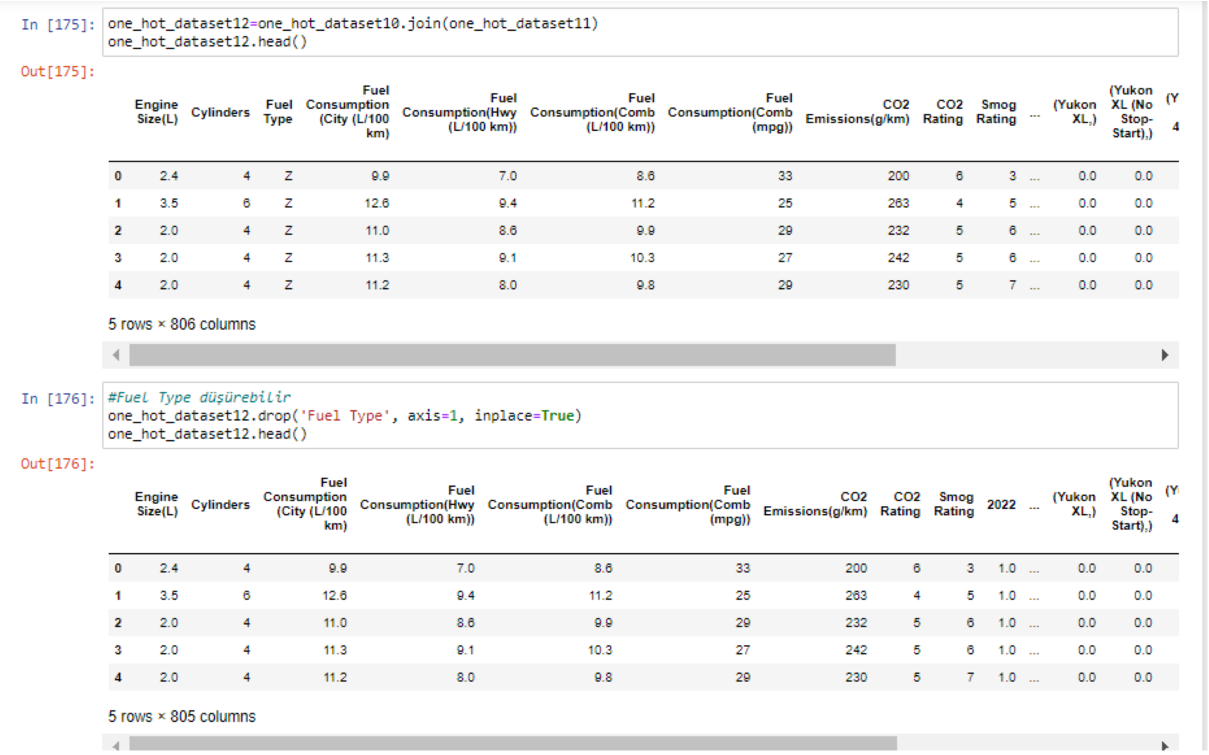
In [173]:
```
one_hot6 = OneHotEncoder(handle_unknown='ignore')
one_hot_FuelType = one_hot6.fit_transform(dataset['Fuel Type'].values.reshape(-1,1)).toarray()
```

In [174]:
```
one_hot_dataset11=pd.DataFrame(one_hot_FuelType, columns=one_hot6.categories_)
one_hot_dataset11.head(946)
```

Out[174]:

	D	E	X	Z
0	0.0	0.0	0.0	1.0
1	0.0	0.0	0.0	1.0
2	0.0	0.0	0.0	1.0
3	0.0	0.0	0.0	1.0
4	0.0	0.0	0.0	1.0
...	...	...	...	...
941	0.0	0.0	0.0	1.0
942	0.0	0.0	0.0	1.0
943	0.0	0.0	0.0	1.0
944	0.0	0.0	0.0	1.0
945	0.0	0.0	0.0	1.0

946 rows × 4 columns



In [175]:
```
one_hot_dataset12=one_hot_dataset10.join(one_hot_dataset11)
one_hot_dataset12.head()
```

Out[175]:

	Engine Size(L)	Cylinders	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	CO2 Rating	Smog Rating	...	(Yukon XL,)	(Yukon XL (No Stop-Start),)	(Y 4
0	2.4	4	Z	9.9	7.0	8.6	33	200	6	3	...	0.0	0.0	
1	3.5	6	Z	12.6	9.4	11.2	25	263	4	5	...	0.0	0.0	
2	2.0	4	Z	11.0	8.6	9.9	29	232	5	6	...	0.0	0.0	
3	2.0	4	Z	11.3	9.1	10.3	27	242	5	6	...	0.0	0.0	
4	2.0	4	Z	11.2	8.0	9.8	29	230	5	7	...	0.0	0.0	

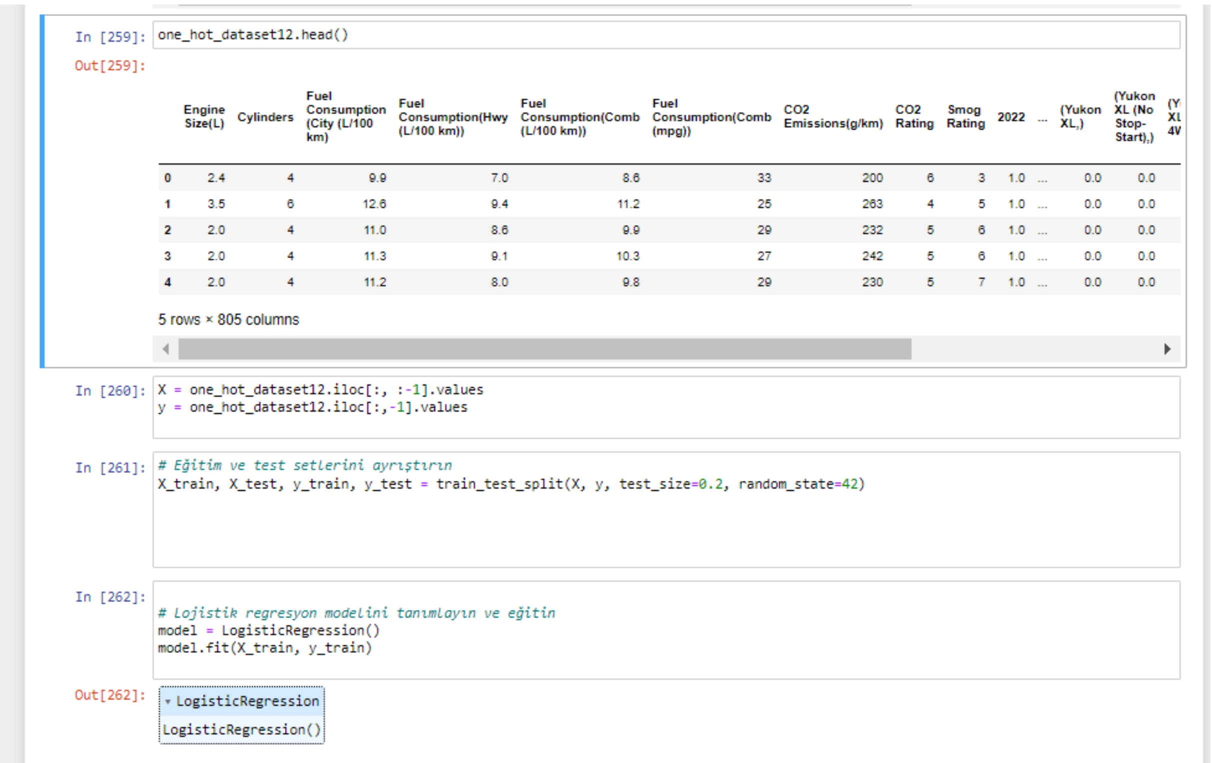
5 rows × 806 columns

In [176]:
```
#Fuel Type düşürebilir
one_hot_dataset12.drop('Fuel Type', axis=1, inplace=True)
one_hot_dataset12.head()
```

Out[176]:

	Engine Size(L)	Cylinders	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	CO2 Rating	Smog Rating	2022	...	(Yukon XL,)	(Yukon XL (No Stop-Start),)	(Y 4
0	2.4	4	9.9	7.0	8.6	33	200	6	3	1.0	...	0.0	0.0	
1	3.5	6	12.6	9.4	11.2	25	263	4	5	1.0	...	0.0	0.0	
2	2.0	4	11.0	8.6	9.9	29	232	5	6	1.0	...	0.0	0.0	
3	2.0	4	11.3	9.1	10.3	27	242	5	6	1.0	...	0.0	0.0	
4	2.0	4	11.2	8.0	9.8	29	230	5	7	1.0	...	0.0	0.0	

5 rows × 805 columns



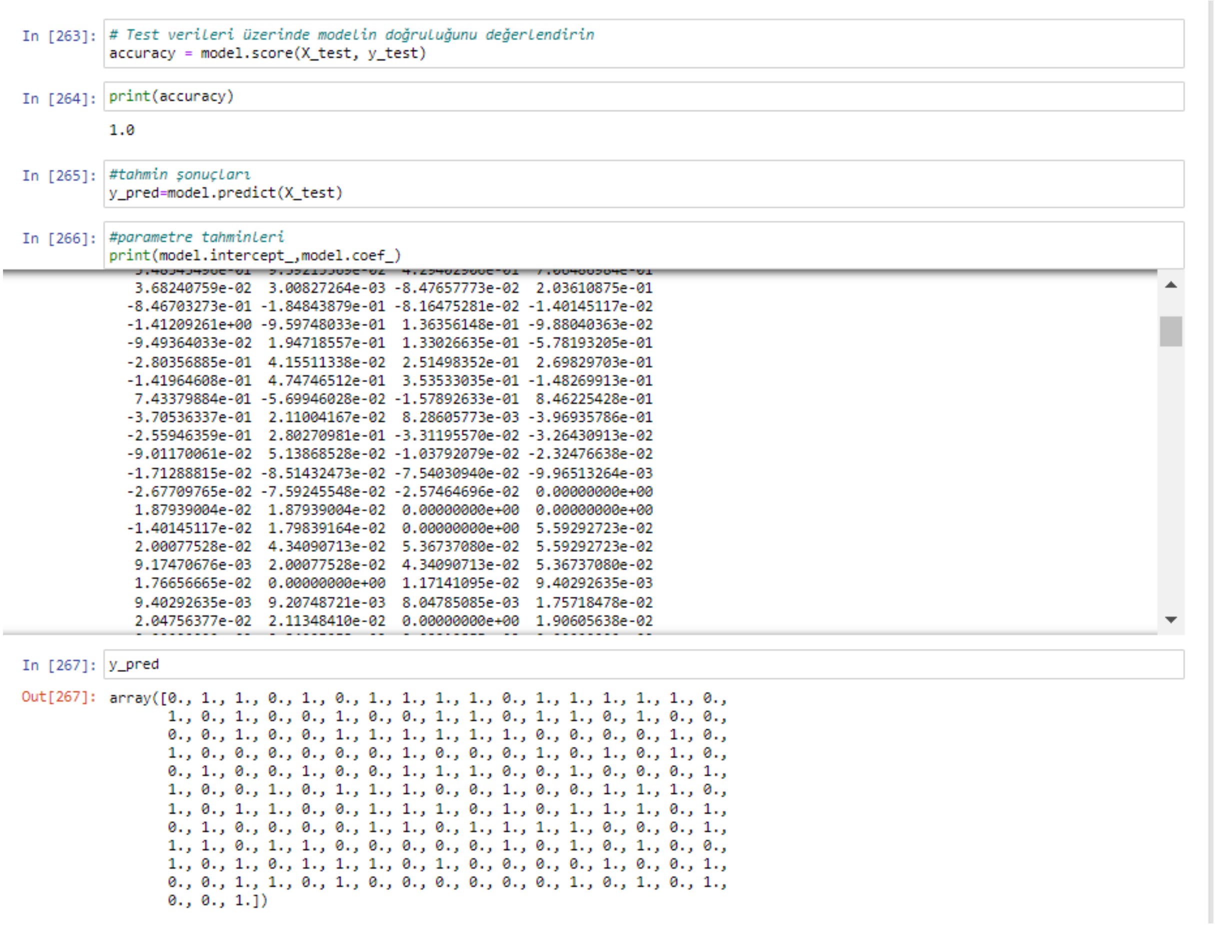
```
In [259]: one_hot_dataset12.head()
```

Out[259]:

	Engine Size(L)	Cylinders	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	CO2 Rating	Smog Rating	2022	...	(Yukon XL,)	(Yukon XL (No Stop-Start),)	(Y XL 4V
0	2.4	4	9.9	7.0	8.6	33	200	6	3	1.0	...	0.0	0.0	
1	3.5	6	12.6	9.4	11.2	25	263	4	5	1.0	...	0.0	0.0	
2	2.0	4	11.0	8.6	9.9	29	232	5	6	1.0	...	0.0	0.0	
3	2.0	4	11.3	9.1	10.3	27	242	5	6	1.0	...	0.0	0.0	
4	2.0	4	11.2	8.0	9.8	29	230	5	7	1.0	...	0.0	0.0	

5 rows × 805 columns

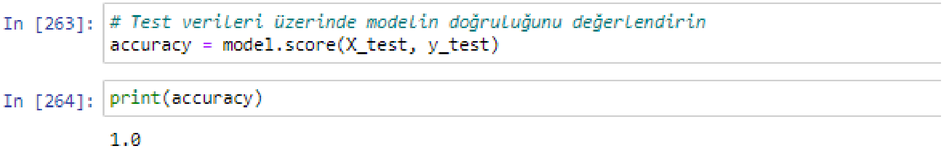
```
In [260]: X = one_hot_dataset12.iloc[:, :-1].values
          y = one_hot_dataset12.iloc[:,-1].values

In [261]: # Eğitim ve test setlerini ayrıştırın
          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

In [262]: # Lojistik regresyon modelini tanımlayın ve eğitin
          model = LogisticRegression()
          model.fit(X_train, y_train)

Out[262]: ▾ LogisticRegression
          LogisticRegression()
```



```
In [263]: # Test verileri üzerinde modelin doğruluğunu değerlendirin
          accuracy = model.score(X_test, y_test)

In [264]: print(accuracy)

          1.0

In [265]: #tahmin şonuçları
          y_pred=model.predict(X_test)

In [266]: #parametre tahminleri
          print(model.intercept_,model.coef_)

   3.68240759e-02  3.00827264e-03 -8.47657773e-02  2.03610875e-01
  -8.46703273e-01 -1.84843879e-01 -8.16475281e-02 -1.40145117e-02
  -1.41209261e+00 -9.59748033e-01  1.36356148e-01 -9.88040363e-02
  -9.49364033e-02  1.94718557e-01  1.33026635e-01 -5.78193205e-01
  -2.80356885e-01  4.15511338e-02  2.51498352e-01  2.69829703e-01
  -1.41964608e-01  4.74746512e-01  3.53533035e-01 -1.48269913e-01
   7.43379884e-01 -5.69946028e-02 -1.57892633e-01  8.46225428e-01
  -3.70536337e-01  2.11004167e-02  8.28605773e-03 -3.96935786e-01
  -2.55946359e-01  2.80270981e-01 -3.31195570e-02 -3.26430913e-02
  -9.01170061e-02  5.13868528e-02 -1.03792079e-02 -2.32476638e-02
  -1.71288815e-02 -8.51432473e-02 -7.54030940e-02 -9.96513264e-03
  -2.67709765e-02 -7.59245548e-02 -2.57464696e-02  0.00000000e+00
   1.87939004e-02  1.87939004e-02  0.00000000e+00  0.00000000e+00
  -1.40145117e-02  1.79839164e-02  0.00000000e+00  5.59292723e-02
   2.00077528e-02  4.34090713e-02  5.36737080e-02  5.59292723e-02
   9.17470676e-03  2.00077528e-02  4.34090713e-02  5.36737080e-02
   1.76656665e-02  0.00000000e+00  1.17141095e-02  9.40292635e-03
   9.40292635e-03  9.20748721e-03  8.04785085e-03  1.75718478e-02
   2.04756377e-02  2.11348410e-02  0.00000000e+00  1.90605638e-02

In [267]: y_pred

Out[267]: array([0., 1., 1., 0., 1., 0., 1., 1., 1., 1., 0., 1., 1., 1., 1., 1., 0.,
                 1., 0., 1., 0., 0., 1., 0., 0., 1., 1., 0., 1., 1., 0., 1., 0., 0.,
                 0., 0., 1., 0., 0., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0., 1., 0.,
                 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 1., 0., 1., 0.,
                 0., 1., 0., 0., 1., 0., 0., 1., 1., 1., 0., 0., 1., 0., 0., 0., 1.,
                 1., 0., 0., 1., 0., 1., 1., 1., 0., 0., 1., 0., 0., 1., 1., 1., 0.,
                 1., 0., 1., 1., 0., 0., 1., 1., 1., 0., 1., 0., 1., 1., 1., 0., 1.,
                 0., 1., 0., 0., 0., 0., 1., 1., 0., 1., 1., 1., 1., 0., 0., 0., 1.,
                 1., 1., 0., 1., 1., 0., 0., 0., 0., 0., 1., 0., 1., 0., 1., 0., 0.,
                 1., 0., 1., 0., 1., 1., 1., 0., 1., 0., 0., 0., 0., 1., 0., 0., 1.,
                 0., 0., 1., 1., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 1.,
                 0., 0., 1.])
```

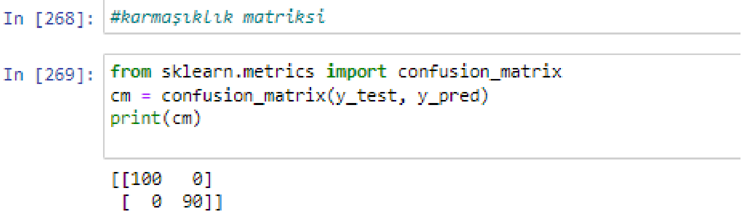
3. Test verisetinde doğruluk ve karmaşıklık matrisi değerlerini raporlayacaklardır.

Bu kod satırı, oluşturulan modelin test setindeki başarısını ölçmek için kullanılır.



```
In [263]: # Test verileri üzerinde modelin doğruluğunu değerlendirin
          accuracy = model.score(X_test, y_test)

In [264]: print(accuracy)

          1.0
```

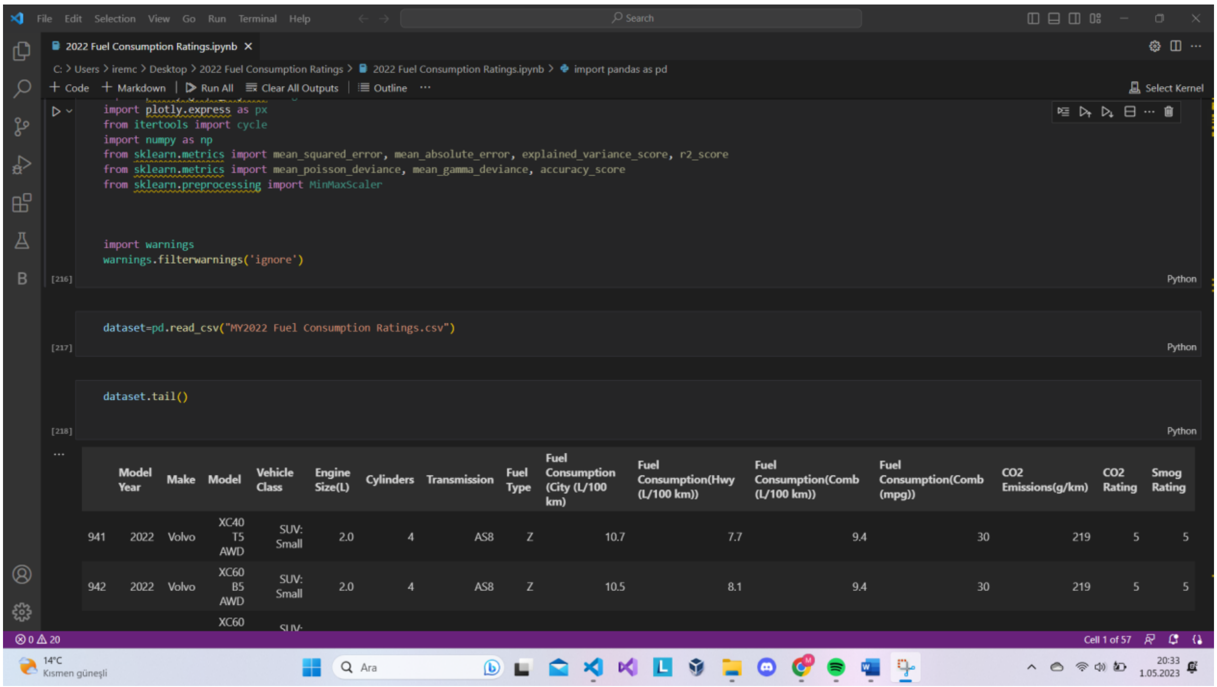
Modelin doğruluğunu ölçmek için kullanılan bir başka yöntem



```
In [268]: #karmaşıklık matriksi

In [269]: from sklearn.metrics import confusion_matrix
          cm = confusion_matrix(y_test, y_pred)
          print(cm)

          [[100   0]
           [  0  90]]
```

Sınav raporu yazı büyüklüğü 11 punto olacaktır. Word formatında olmalıdır. Raporda ilgili kısımlar kod blokları ile açıklanacaktır.

Ödev raporunda bir adet öğrenci kod bloğu ile işletim sistemi saat, zaman ve kullanıcı bilgisinin bulunduğu ekran görüntüsü de olmalıdır.


File
Edit
Selection
View
Go
Run
Terminal
Help
Search
2022 Fuel Consumption Ratings.ipynb
C: > Users > iremc > Desktop > 2022 Fuel Consumption Ratings > 2022 Fuel Consumption Ratings.ipynb > import pandas as pd
+ Code
+ Markdown
Run All
Clear All Outputs
Outline
Select Kernel

```
import plotly.express as px
from itertools import cycle
import numpy as np
from sklearn.metrics import mean_squared_error, mean_absolute_error, explained_variance_score, r2_score
from sklearn.metrics import mean_poisson_deviance, mean_gamma_deviance, accuracy_score
from sklearn.preprocessing import MinMaxScaler



import warnings
warnings.filterwarnings('ignore')
```

```
dataset=pd.read_csv("MY2022 Fuel Consumption Ratings.csv")
```

```
dataset.tail()
```

	Model Year	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption (City (L/100 km)	Fuel Consumption(Hwy (L/100 km))	Fuel Consumption(Comb (L/100 km))	Fuel Consumption(Comb (mpg))	CO2 Emissions(g/km)	CO2 Rating	Smog Rating
941	2022	Volvo	XC40 T5 AWD	SUV: Small	2.0	4	AS8	Z	10.7	7.7	9.4	30	219	5	5
942	2022	Volvo	XC60 B5 AWD	SUV: Small	2.0	4	AS8	Z	10.5	8.1	9.4	30	219	5	5
			XC60	SUV:											

Benzer öğrenci ödevlerine kopya işlemi uygulanacaktır.