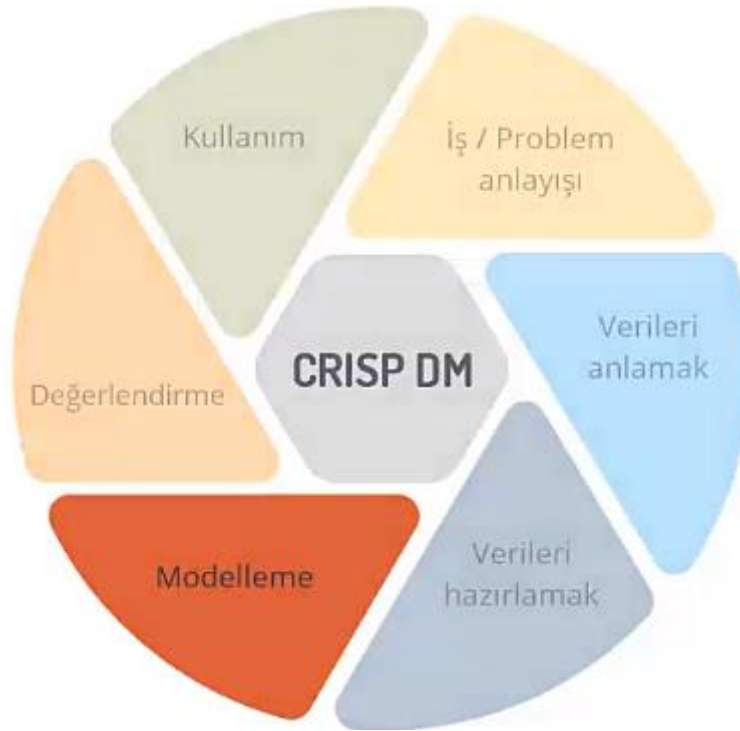
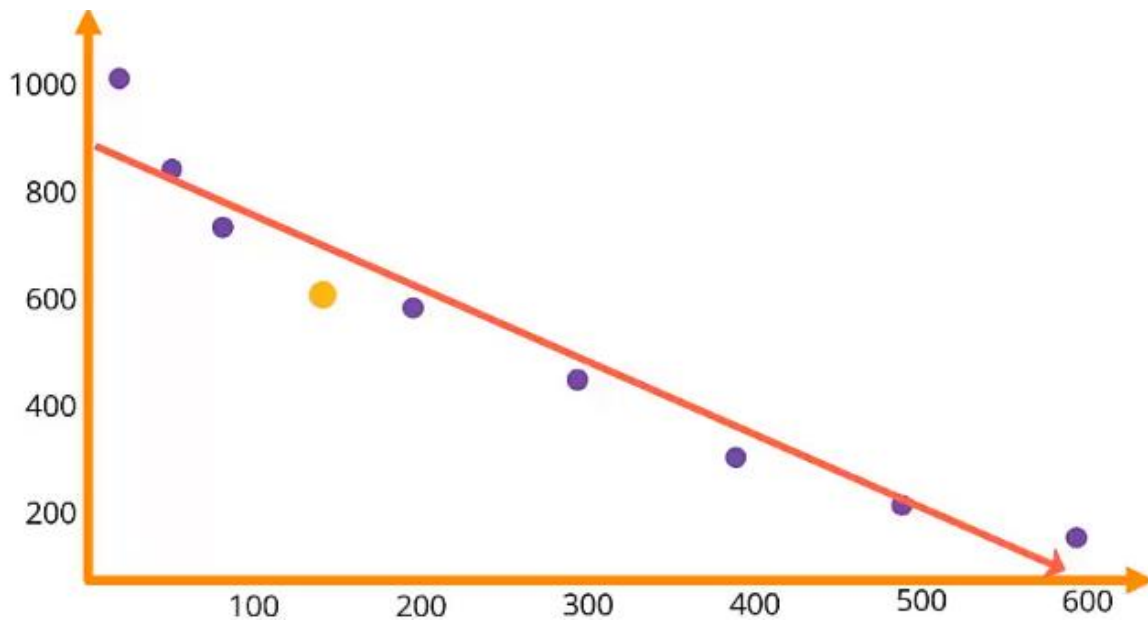


LİNEER REGRESYON

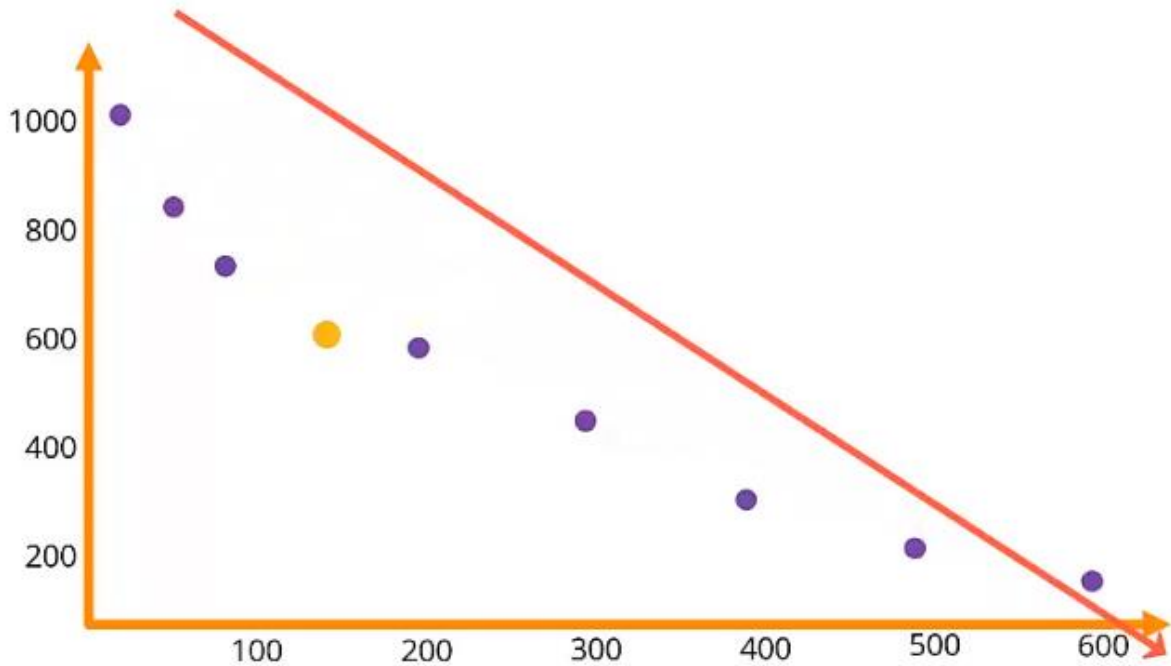


Artık model oluşturup bilgisayarı eğiterek bizlere tahmin vermesini sağlayacağız.

Regresyon, birbirine bağlı iki değişken arasındaki ilişkinin gücünü belirlemeye çalışan istatistiksel bir ölçümdür. **Lineer**: Değişmesi bir doğru ile gösterilebilen. **Lineer Regresyon**: İki ya da daha çok nicel değişken arasındaki ilişkiyi ölçmek için kullanılan analiz metodudur.



Değerler arasındaki fark azsa başarılı bir model kurmuşuz demektir. Eğer regresyon çizgisi noktalardan uzaktaysa modelin başarı oranı düşüktür.



Regresyon çizgisini oluşturmak için Python'un Sckit Learn kütüphanesinden yararlanacağız. Makine öğrenmesi alanında en yaygın kullanılan kütüphanelerden birisidir.

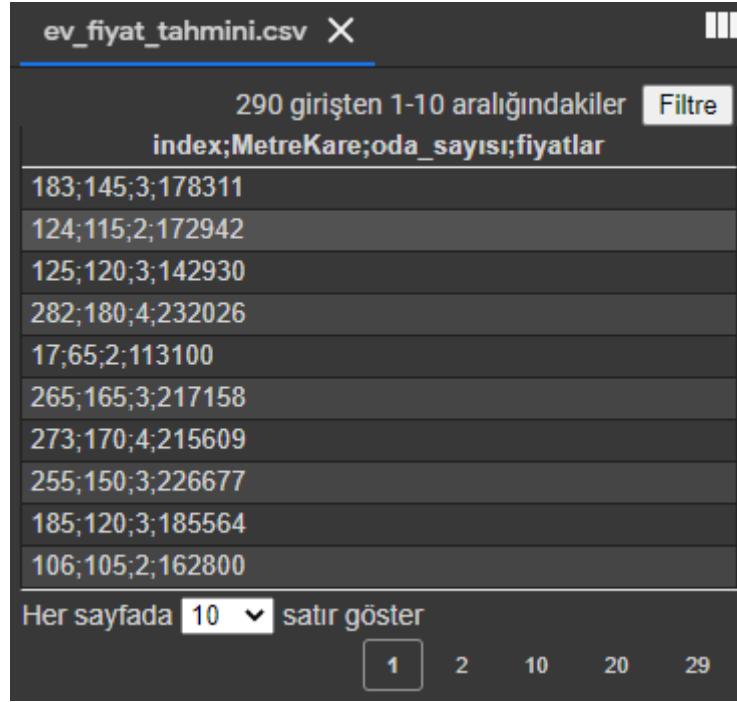
Lineer Regresyon Uygulaması

Uygulamada bir evin metrekaresine veya oda sayısına bakarak satış fiyatını tahmin etmeye çalışacağız. Makineye öğretmek makinenin tahmin vermesini sağlayacağız.

Veri dosyasını düzenlemek ve görselleştirmek için kullanacağımız kütüphaneleri import etme: Sckit learn kütüphanesini veri setini eğitmek için kullanacağız. Kütüphanede birçok makine öğrenmesi algoritması olduğu için sadece Lineer regresyon modelini kullanacağız.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

Veri setimizde veriler virgül yerine noktalı virgül ile ayrılmış.



ev_fiyat_tahmini.csv X

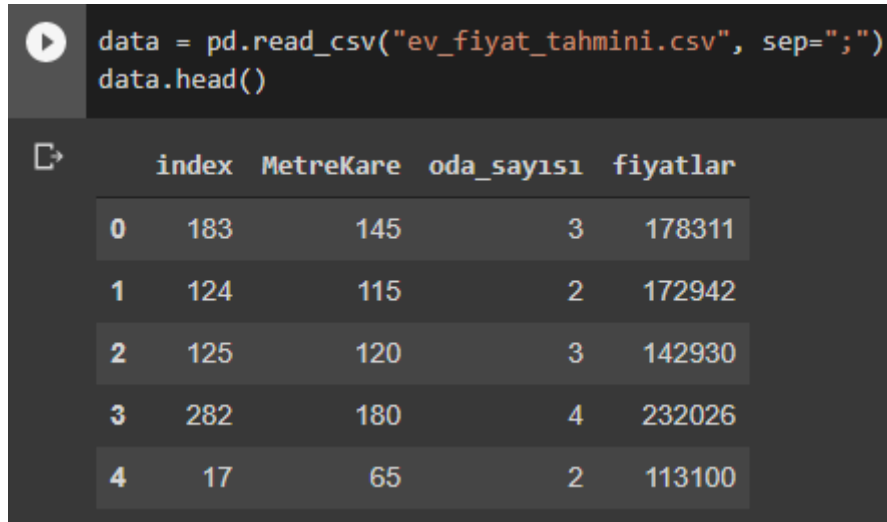
290 girişten 1-10 aralığındakiler Filtre

index;MetreKare;oda_sayısı;fiyatlar
183;145;3;178311
124;115;2;172942
125;120;3;142930
282;180;4;232026
17;65;2;113100
265;165;3;217158
273;170;4;215609
255;150;3;226677
185;120;3;185564
106;105;2;162800

Her sayfada 10 satır göster

1 2 10 20 29

Pandas kütüphanesinde read fonksiyonu varsayılan olarak virgülle ayrılmış verileri okuyordu. Burada yapılması gereken verilerin noktalı virgül ile ayrıldığını belirtmek.



```
data = pd.read_csv("ev_fiyat_tahmini.csv", sep=";")
data.head()
```

	index	MetreKare	oda_sayısı	fiyatlar
0	183	145	3	178311
1	124	115	2	172942
2	125	120	3	142930
3	282	180	4	232026
4	17	65	2	113100

İnternette bulunan veri setleri genellikle noktalı virgülle ayrılmış verilerden oluşuyor. İndex numaraları satır numaralarıdır. Başka veri setinden rastgele veriler alıp oluşturulmuştur. Bu yüzden verilerin yerleri karışıktır.

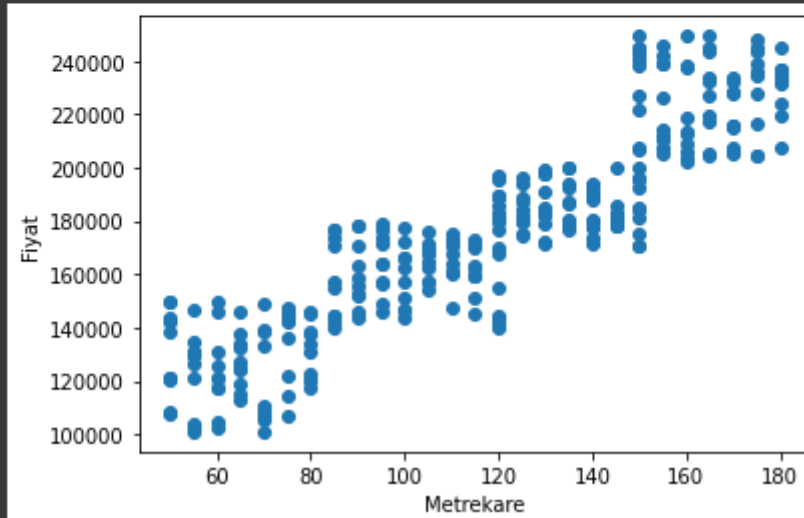
Data.shape() fonksiyonu ile kaç tane verimizin olduğuna bakalım:

```
data.shape  
(290, 4)
```

Metrekare ile fiyatları dağılım grafiğine göre karşılaştırma: plt.scatter() fonksiyonu ile dağılım grafiği oluşturuyoruz.

```
x = data['Metrekare']  
y = data['fiyatlar']
```

```
plt.scatter(x,y)  
  
plt.xlabel("Metrekare")  
plt.ylabel("Fiyat")  
  
plt.show()
```



Grafikte doğrusal artış var. Metrekare artınca fiyatlarında arttığı görülüyor.

Daha doğru tahminler almak için regresyon modelini kullanma:

Bu fonksiyon Sckit learn kütüphanesinden geliyor. Regresyon çizgisini oluşturmak için verilere ihtiyacımız var. Bunlar metrekaare ve satış fiyatlarıydı. `model.fit(x,y)` ile regresyon çizgisi oluşturuluyor. Sckitlearn kütüphanesi model eğitmek için verileri 2 boyutlu numpy dizisi olarak istiyor. Oluşturduğumuz veriler pandas serieslerinden oluşuyor.

```
type(x)  
pandas.core.series.Series
```

Metrekare sütundaki tüm verileri array'e dönüştür: 290 satır, 1 sütun (metrekare olduğu için). Satır numarası yerine -1 yazdığımızda numpy kütüphanesi verinin kaç satırdan oluştuğunu kendisi bulacaktır.

```
[11] x = data['MetreKare'].values.reshape((-1,1))  
     y = data['fiyatlar'].values.reshape((-1,1))
```

```
[12] print(x.shape)  
     print(y.shape)
```

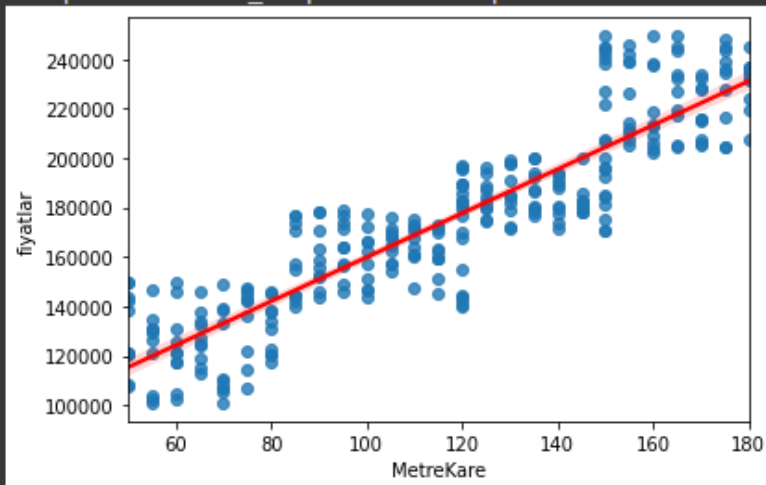
```
(290, 1)  
(290, 1)
```

```
[13] model.fit(x,y)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
sns.regplot(x="MetreKare", y="fiyatlar", data=data, line_kws={"color":"red"})
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5d0bc97f10>
```



Verilere en yakın regresyon çizgisini oluşturduk. Modelden tahmin alalım:

```
model.predict([[150]])  
array([[204406.50495214]])
```

Oluşturduğumuz regresyon modelinin ne kadar başarı sağladığını görmek için `model.score()` içine metrekaire ve fiyat verilerini de yazıyorum. Sonuç 1'e yaklaştıkça modelin başarı oranı artmaktadır.

```
model.score(x,y)  
0.8131235669951359
```

KAYNAKÇA

Bilgeiş “Herkes için Yapay Zekâ II” eğitimi.

KODLUYORUZ
geleceği kodluyoruz >_

 **EMpower**
Enriching young lives in emerging markets