



VERİ BİLİMİNE GİRİŞ

**Proje2- Scikit-Learn ile Makine
Öğrenmesi (Tahminleme)**


Konu : Laptop Fiyat Tahmini

Öğrenciler:

05150000635- Şevval KOÇMAR

05150000609- Hasan REİSOĞLU

05150000607- Sedanur EKER



Arkaplan ve Motivasyon

Günümüz koşullarında artık neredeyse her bireyin bir bilgisayar var. Gelişen teknolojiyle birlikte sürekli yeni özelliklere ve donanımlara sahip yeni bilgisayar modelleri üreilmeye devam ediyor. İnsanlar teknolojiyi takip ederek hem isteklerini karşılayan hem de bütçesine uygun bilgisayar modelleri bulmayı amaçlıyor. Ama bunun için tek tek bütün modelleri ve özelliklerini fiyatlarıyla birlikte ayrı ayrı karşılaştırma yapmak zahmetli bir durum. Biz de istediğimiz yapıya uygun veri seti imkanımız olduğu için ve buna uygun featurelar mevcut olduğu için istenen özelliklere göre tahmini bir dizüstü bilgisayar fiyatı çıkararak kullanıcılara yarar sağlamak istedik.

Projenin Amaçları (Proje Tanımı)

- 1-Bir dizüstü bilgisayarın donanım özelliklerinin (Cpu,Ram,Memory,Gpu... gibi) fiyata etkisi
- 2- Üretici şirketin (Apple, HP, Acer... gibi) fiyata etkisi
- 3- Seçilen işletim sisteminin fiyata etkisi

Bu parametreler ile, bilgisayarların özelliklerinin fiyatlarında ne gibi farklılıklara yol açtığını, bu piyasayı belirleyen en önemli ögenin ne olduğunu ve aynı özelliklere sahip farklı bilgisayar markalarındaki fiyat farklarının nasıl değiştiğini araştırdık.

Veri

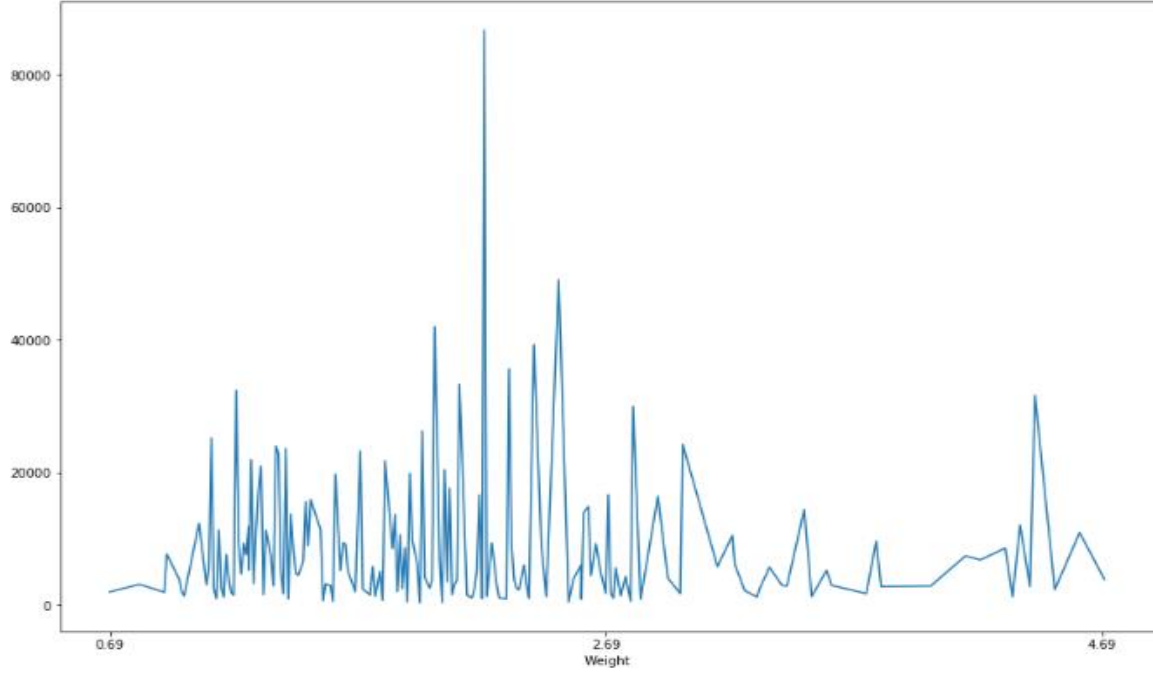
- Veri kaynağımız: Kaggle → Laptop Prices
- İçinde herhangi bir çözüm dosyası bulunmayan bu veri setinin içinde şirket, ürün, tip adı, inç, ekran çözünürlüğü, CPU, RAM, bellek, işletim sistemleri, ağırlık ve Euro üzerinden fiyat başlıklarından oluşan sütunlar vardır.

Veri İşleme

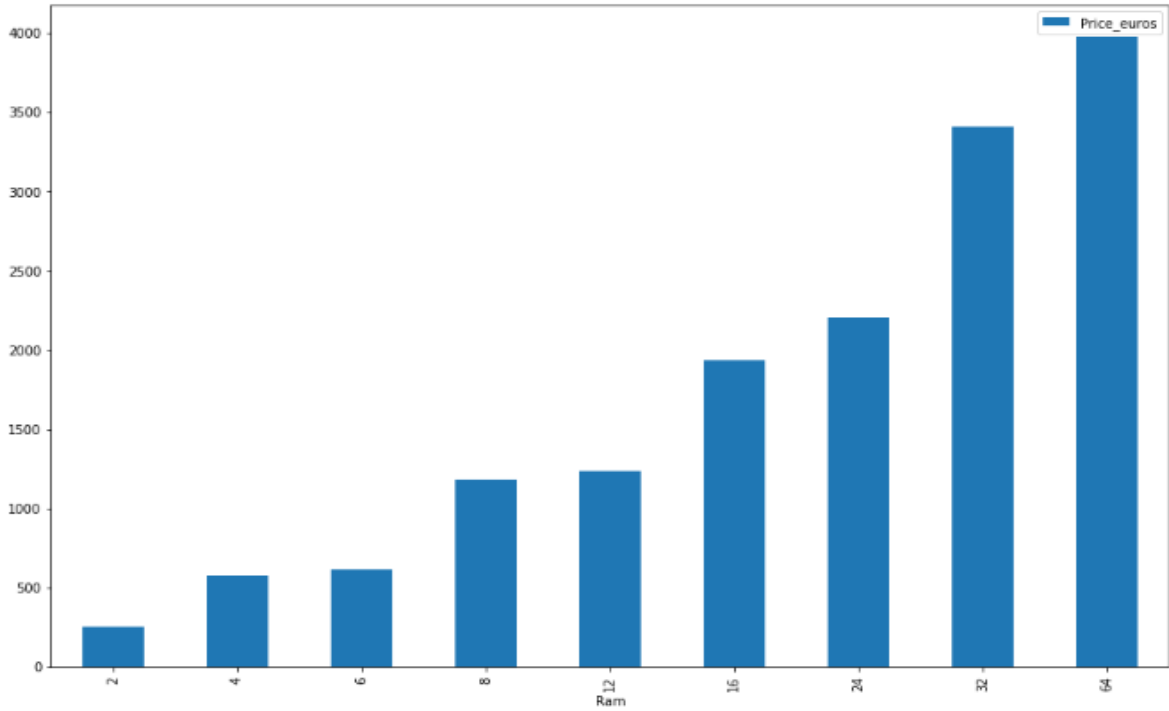
- ScreenResolution sütununu ayırmak için string kısmını atarak 2560x1600 formatında kalan verileri genişlik (Width) ve yükseklik (Height) olmak üzere iki sütuna ayırdık ve bu değerleri bölüp o sütunlara atadık.
- OpSys, TypeName, Cpu, Gpu, Company sütunlarına one-hot-encoding uyguladık.
- Veri setimizin içinde mevcut olan “Product” sütununu marka modellerinin ürün fiyatlarının piyasadaki değerleri belli olduğundan dolayı ve bizim amacımızın ürün özelliklerine göre bir fiyat tahminleme olduğundan dolayı regresyon değerlerimizin daha doğru sonuç vermesi için bu sütunu çıkardık.
- Ram sütunundaki “GB” ve Weight sütunundaki “kg” stringlerini çıkarıp kalan sayısal verileri numerik formata çevirdik.
- “Memory” sütunu içinde 256GB SSD, 128GB Flash Storage, 500GB HDD, 128GB SSD + 1TB HDD, 1.0TB Hybrid gibi olan değerleri dört sütuna ayırdık. Bunun sonucunda memory sütunu artık SSD, HDD, Flash, Hybrid sütunlarına ayrıldı. Daha sonraki adımda içlerinde kalan GB değerlerini de “GB” stringini atarak sayısal verinin kalmasını sağladık ve bunu numerik tipe çevirdik. GB değerleri dışında “TB” değerleri de olduğu için aynı tipte olması adına TB stringini atıp TB değerine karşılık gelen GB değerine çevirdikten sonra verimiz tek tip olmuş oldu.

Keşif Analizi

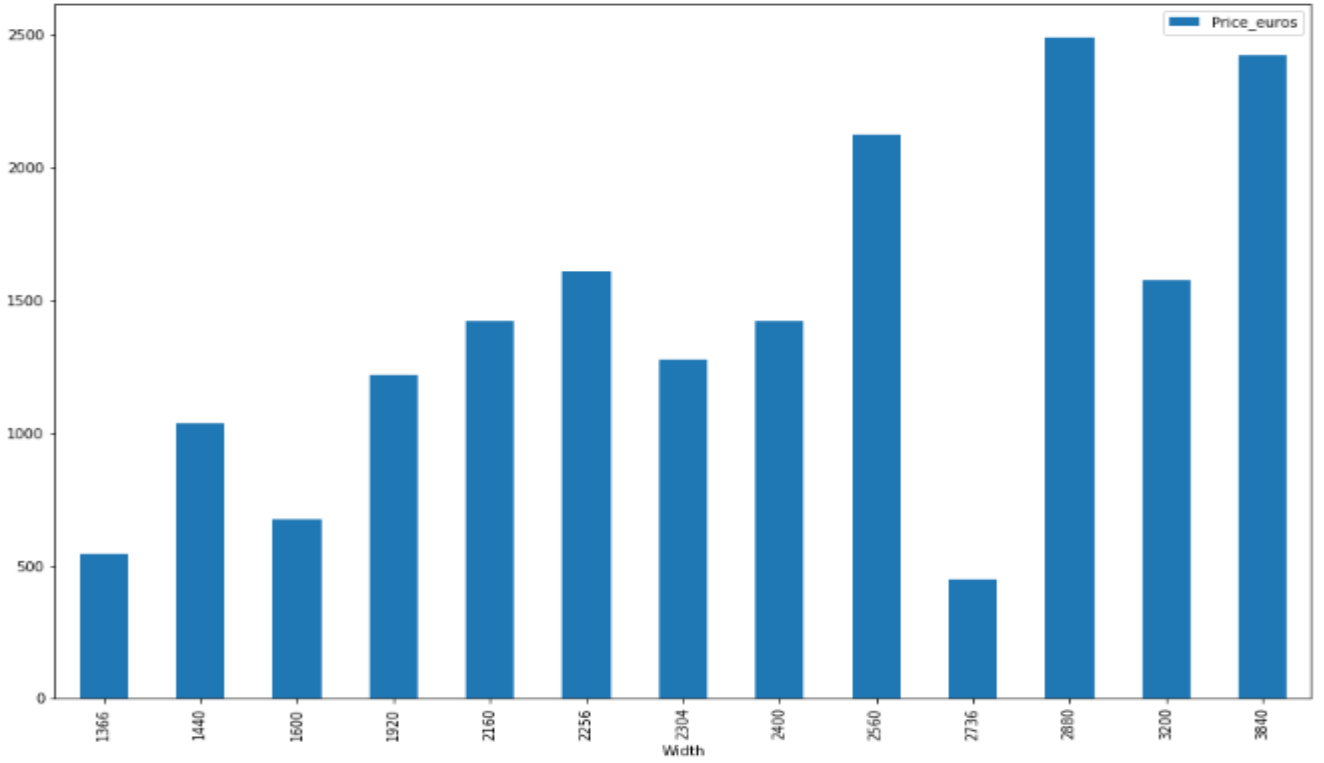
- Laptop ağırlığının fiyata olan etkisi



- Ram e göre fiyat değerleri ortalaması



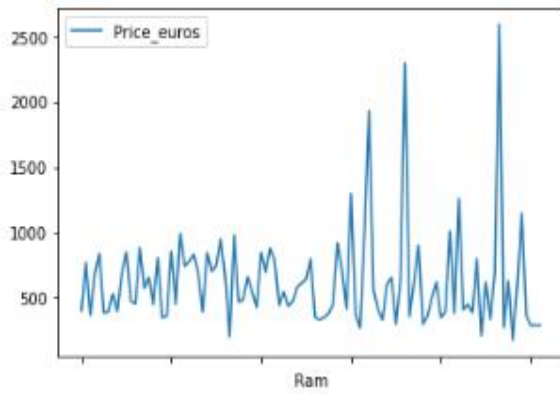
➤ Genişliğe göre fiyat değerleri ortalaması



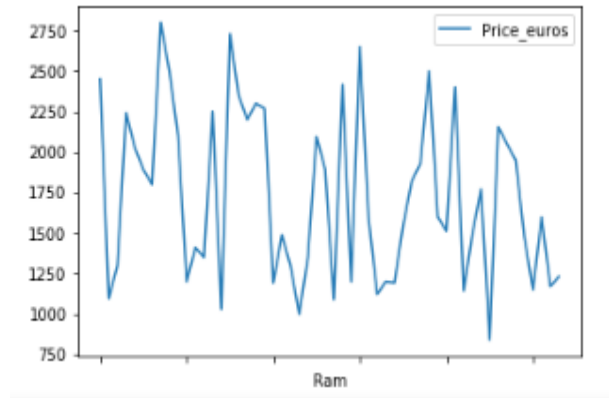
➤ Şirketlere göre RAM değerlerinin fiyata etkisi

Company	
Acer	AxesSubplot(0.125,0.125;0.775x0.755)
Apple	AxesSubplot(0.125,0.125;0.775x0.755)
Asus	AxesSubplot(0.125,0.125;0.775x0.755)
Chuweni	AxesSubplot(0.125,0.125;0.775x0.755)
Dell	AxesSubplot(0.125,0.125;0.775x0.755)
Fujitsu	AxesSubplot(0.125,0.125;0.775x0.755)
Google	AxesSubplot(0.125,0.125;0.775x0.755)
HP	AxesSubplot(0.125,0.125;0.775x0.755)
Huawei	AxesSubplot(0.125,0.125;0.775x0.755)
LG	AxesSubplot(0.125,0.125;0.775x0.755)
Lenovo	AxesSubplot(0.125,0.125;0.775x0.755)
MSI	AxesSubplot(0.125,0.125;0.775x0.755)
Mediacom	AxesSubplot(0.125,0.125;0.775x0.755)
Microsoft	AxesSubplot(0.125,0.125;0.775x0.755)
Razer	AxesSubplot(0.125,0.125;0.775x0.755)
Samsung	AxesSubplot(0.125,0.125;0.775x0.755)
Toshiba	AxesSubplot(0.125,0.125;0.775x0.755)
Vero	AxesSubplot(0.125,0.125;0.775x0.755)
Xiaomi	AxesSubplot(0.125,0.125;0.775x0.755)

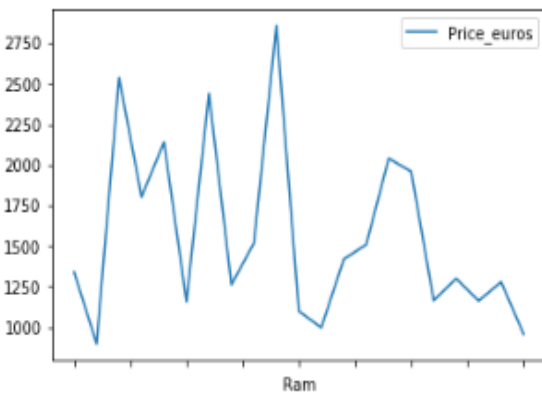
Acer :



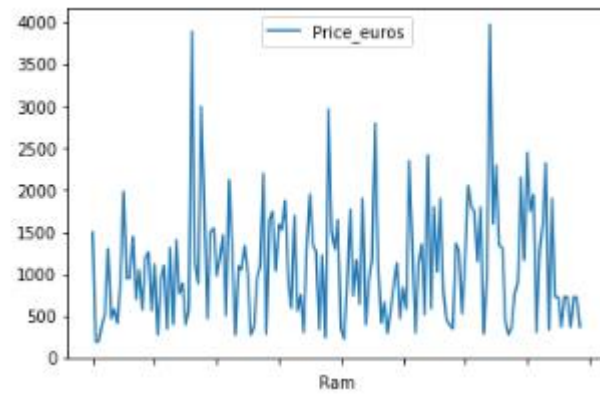
MSI:



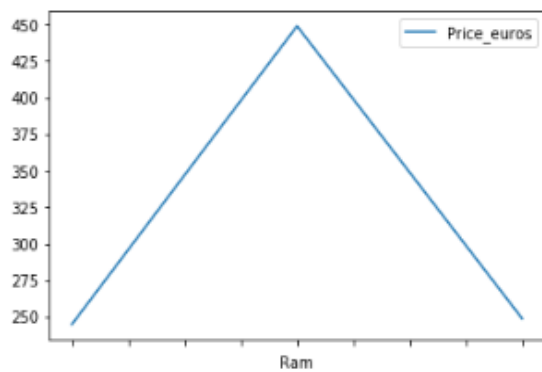
Apple:



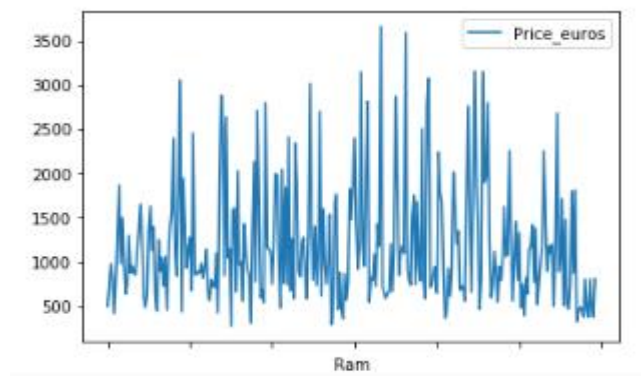
Asus:



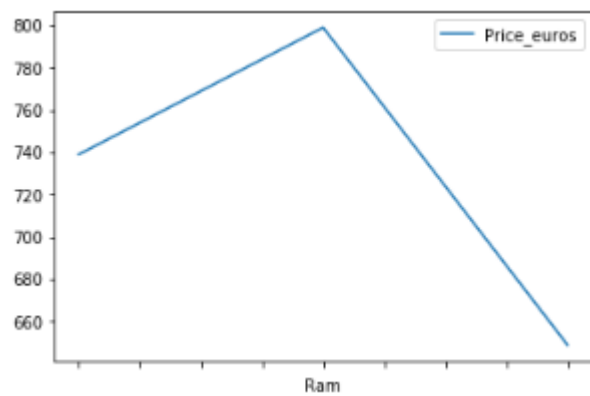
Chuwi:



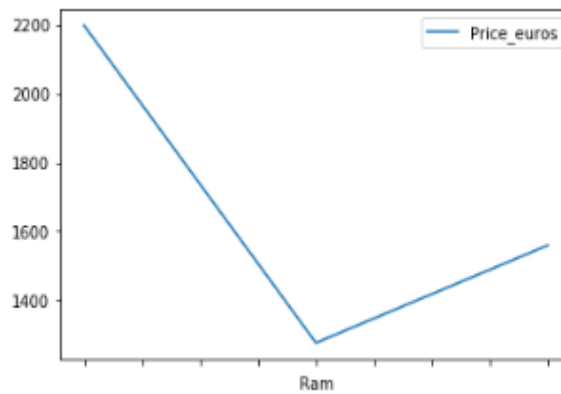
Dell:



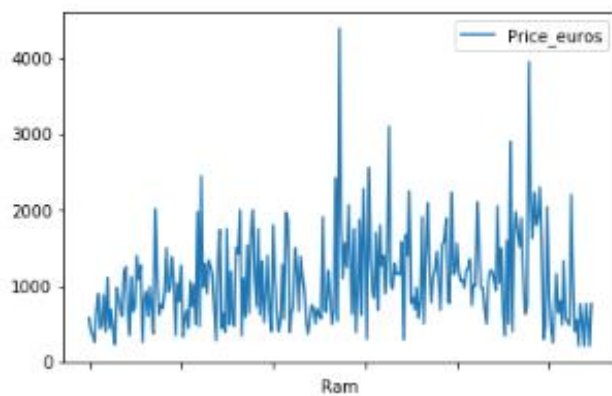
Fujitsu:



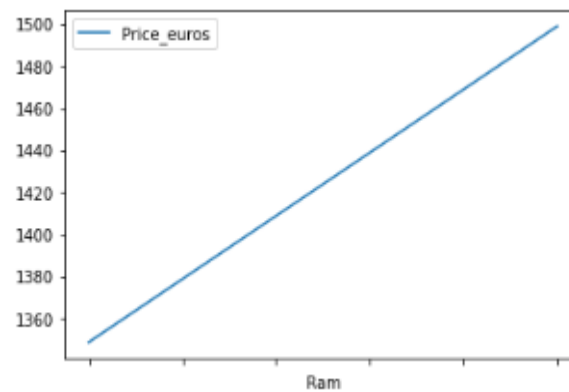
Google:



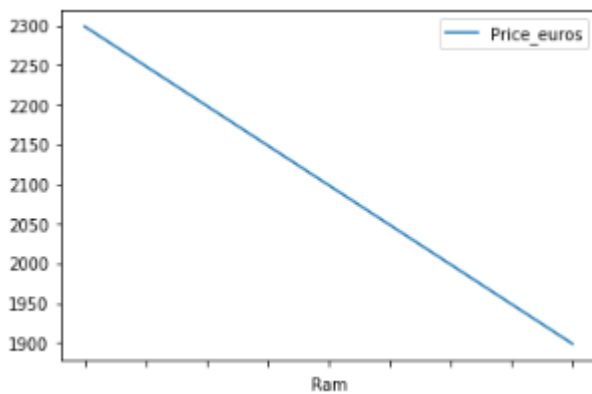
HP:



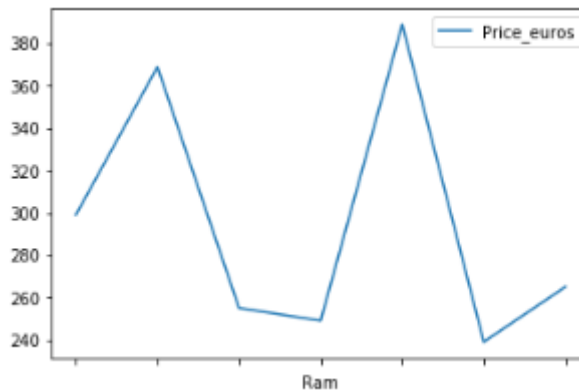
Huawei:



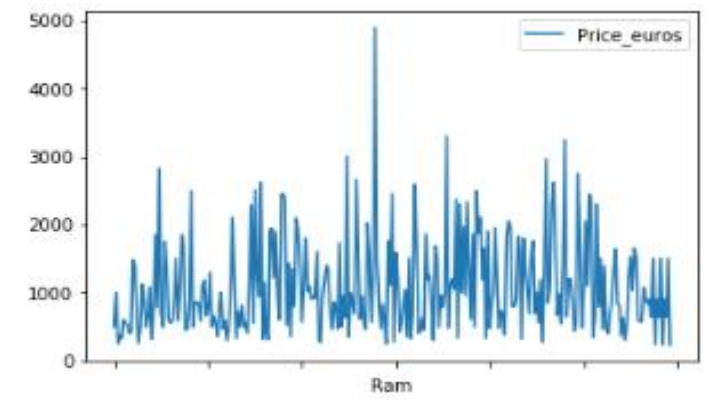
LG:



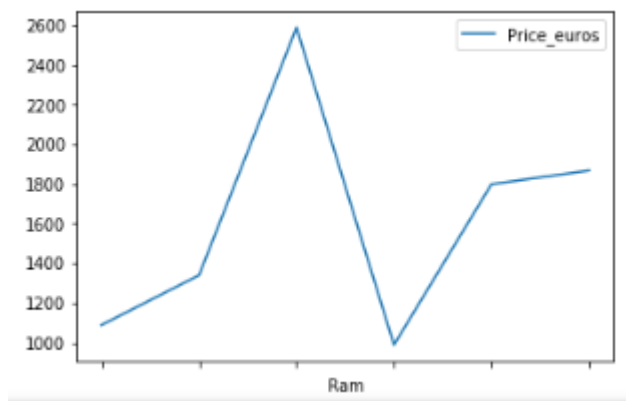
Mediacom:



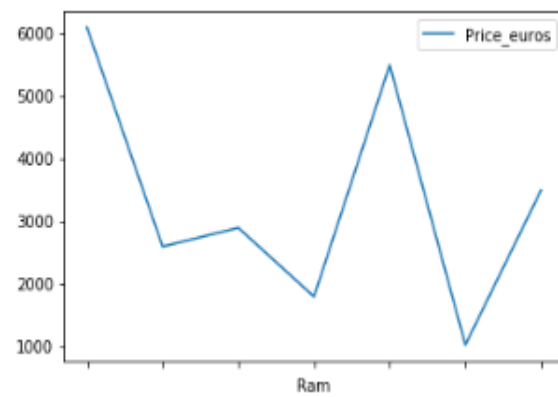
Lenovo:



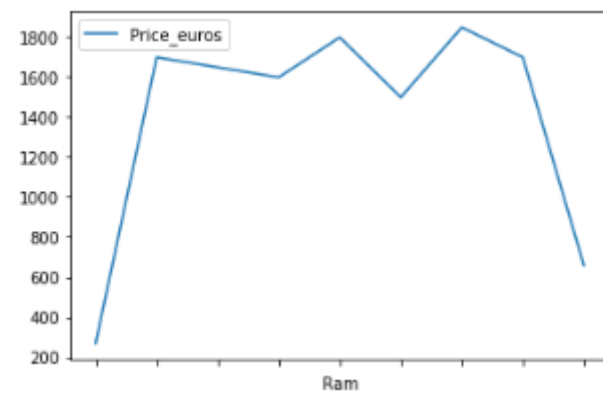
Microsoft:



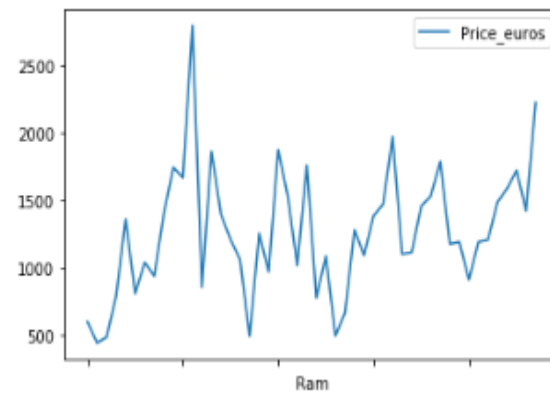
Razer:



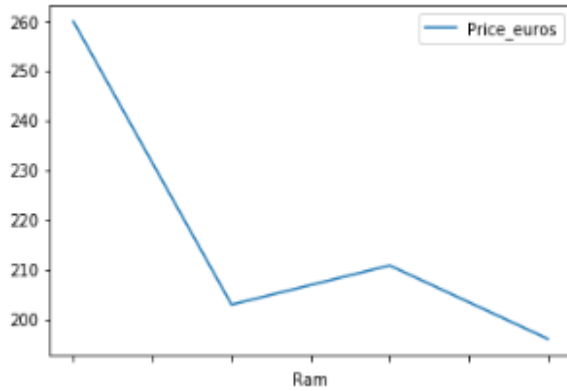
Samsung:



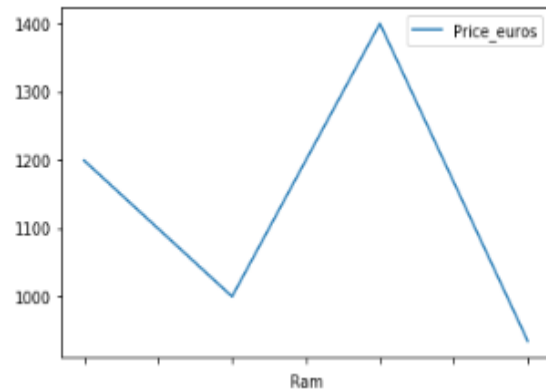
Toshiba:



Vero:



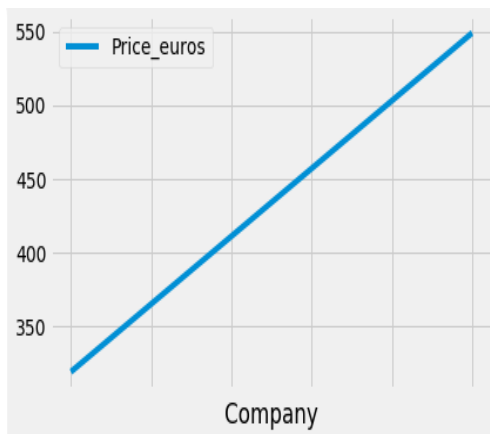
Xiaomi:



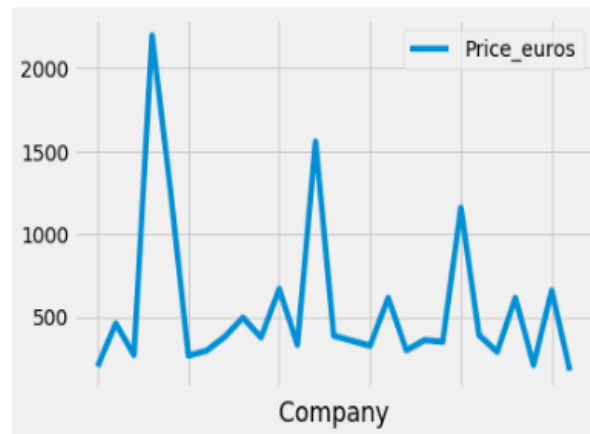
➤ Şirketlerin işletim sistemlerinin fiyata etkisi

OpSys	
Android	AxesSubplot(0.08,0.07;0.87x0.81)
Chrome OS	AxesSubplot(0.08,0.07;0.87x0.81)
Linux	AxesSubplot(0.08,0.07;0.87x0.81)
Mac OS X	AxesSubplot(0.08,0.07;0.87x0.81)
No OS	AxesSubplot(0.08,0.07;0.87x0.81)
Windows 10	AxesSubplot(0.08,0.07;0.87x0.81)
Windows 10 S	AxesSubplot(0.08,0.07;0.87x0.81)
Windows 7	AxesSubplot(0.08,0.07;0.87x0.81)
macOS	AxesSubplot(0.08,0.07;0.87x0.81)

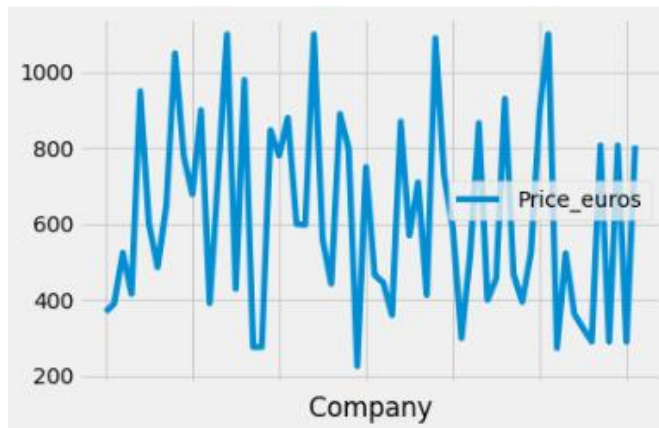
Android:



Chrome OS:



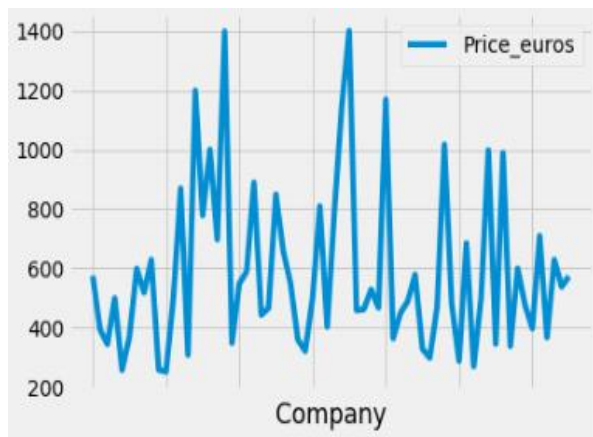
Linux:



Mac OS X:



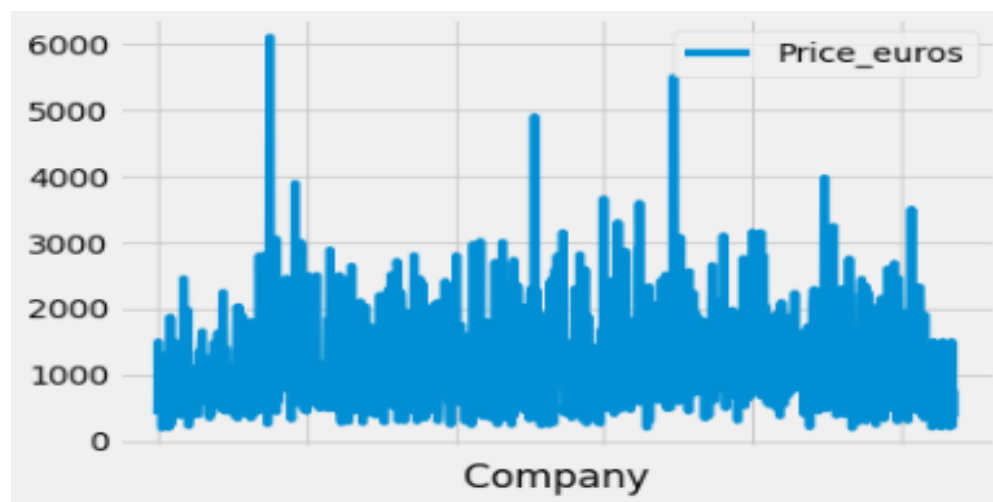
No OS:



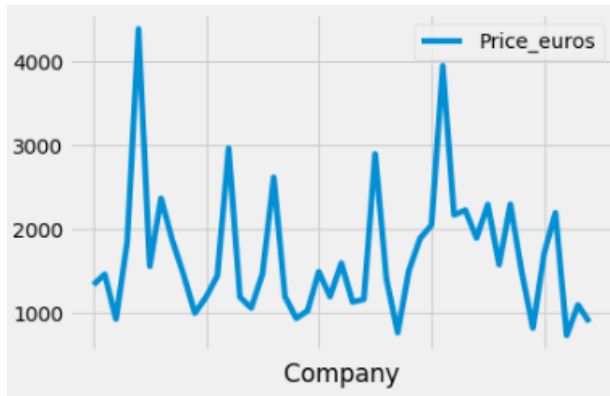
Windows 10 S:



Windows 10:



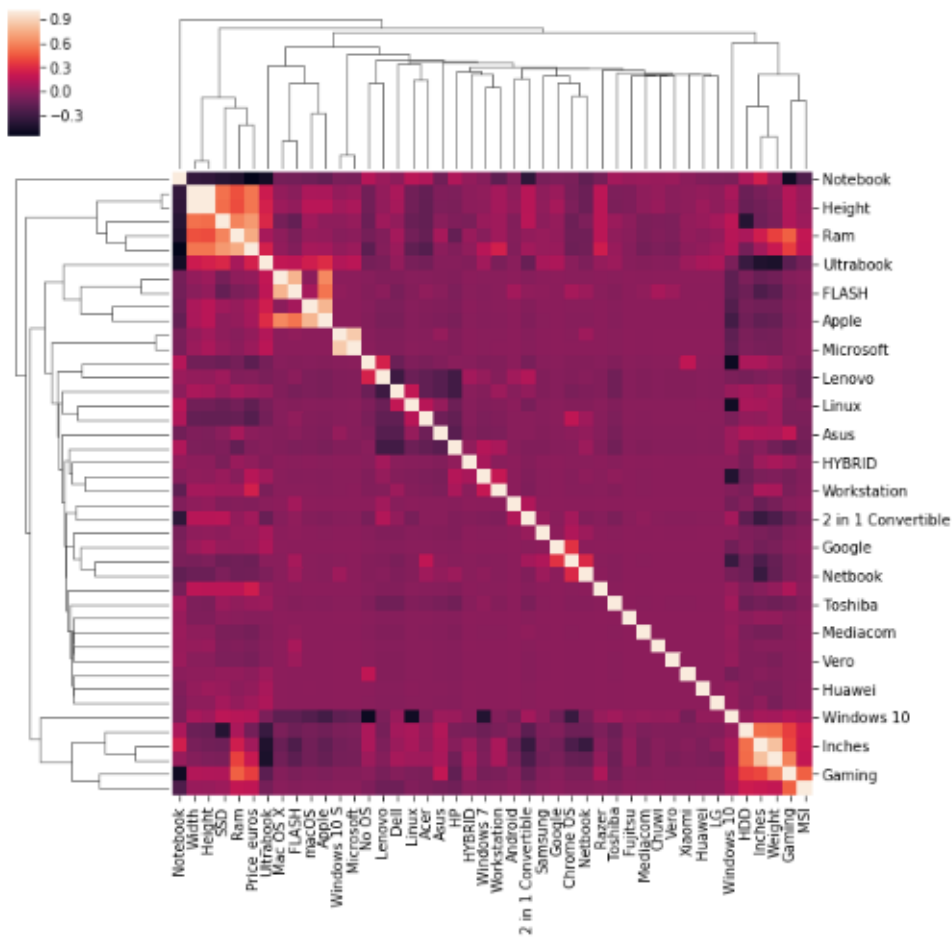
Windows 7:



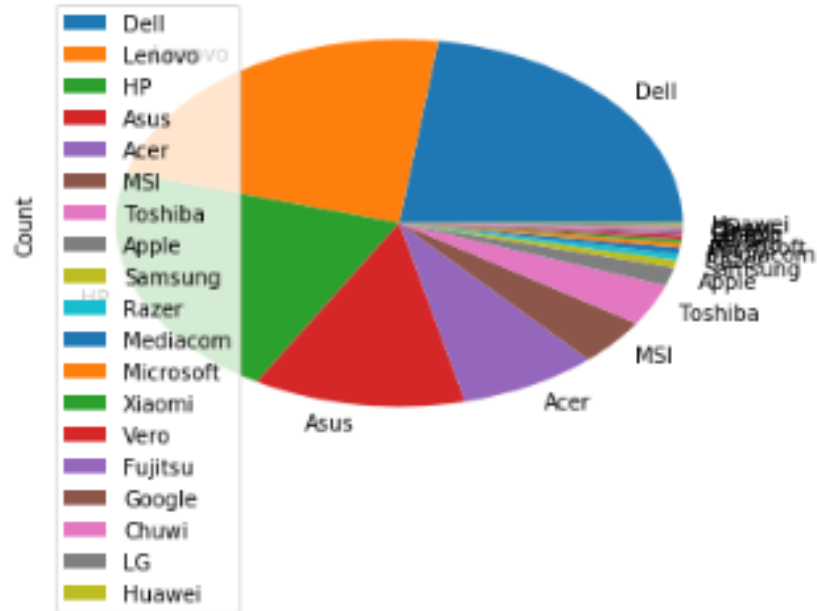
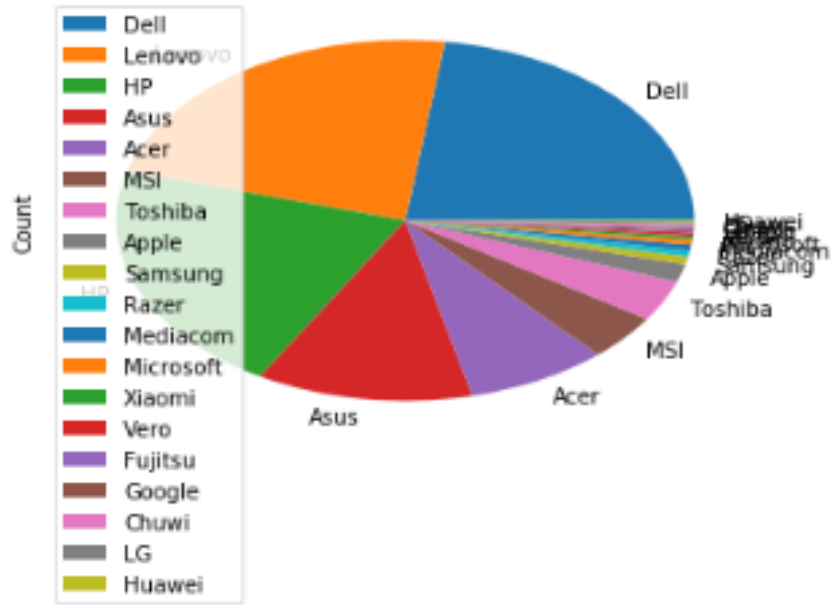
Mac OS:



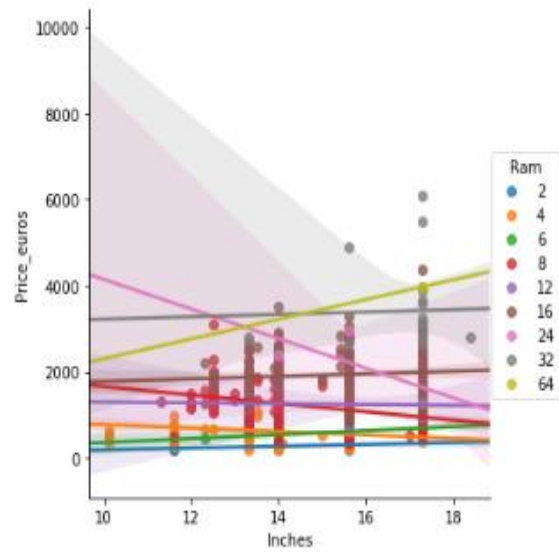
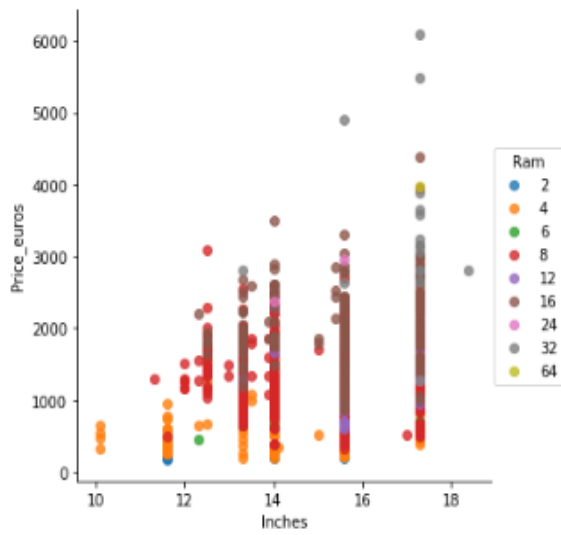
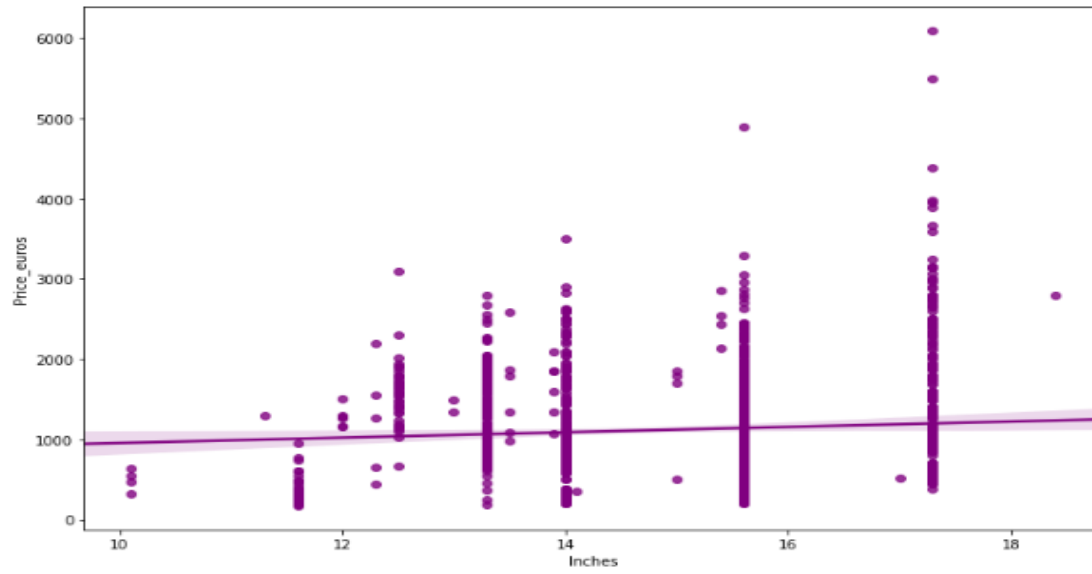
➤ Sayısal değerler arasındaki kolerasyon



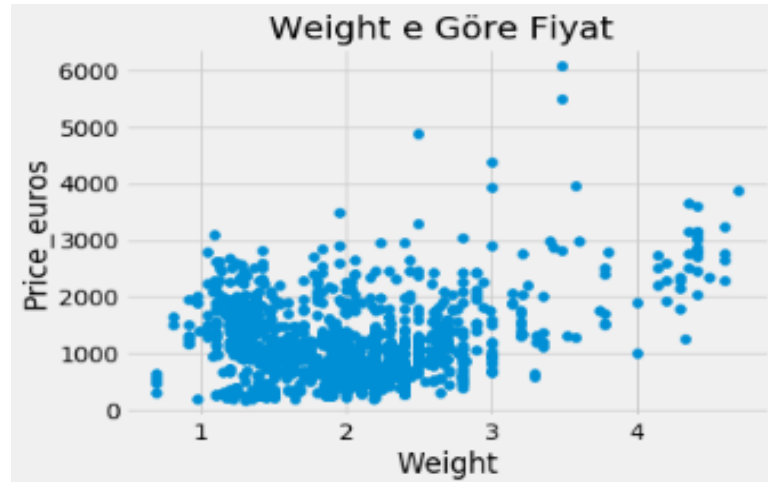
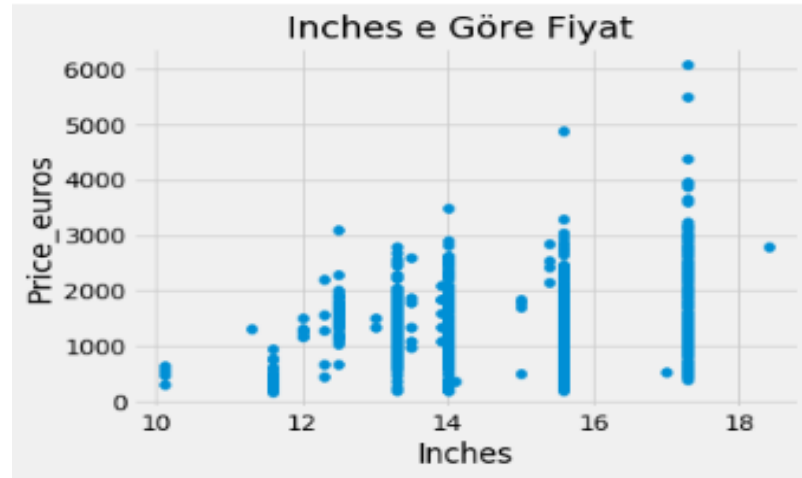
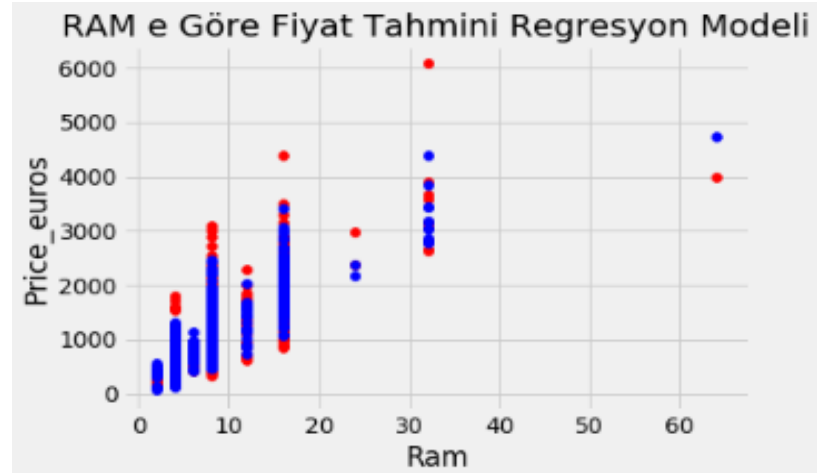
➤ Şirketlerin ürün grafikleri



➤ İnç-Fiyat grafikleri



➤ Tahminlere göre regresyon modelleri



Analiz Metodolojisi

- K-Fold Cross Validation, sınıflandırma modellerinin değerlendirilmesi ve modelin eğitilmesi için veri setini parçalara ayırma yöntemlerinden biridir. Bu bağlamda biz de veri setimizi 1/5 lik parçalara bölerek Cross Validation uyguladık.
- Veri seti modelimizde bütün one hot verilerle birlikte yaptığımız analizde performans metriklerinin sağlıklı sonuç üretmemesinden dolayı buna sebep olan CPU ve GPU sütunlarını çıkararak bir cross validation daha yaptık. Bu karşılaştırmada CPU ve GPU sütunlarındaki çok çeşitlilik sebebiyle çok fazla one-hot-encoding sütunu üretilmesinden dolayı modelin nasıl etkilendiğini görebiliriz.

CPU ve GPU sütunları çıkarılmış data için Cross Validation

```
In [183]: from sklearn.cross_validation import cross_val_score, cross_val_predict
          from sklearn import metrics
          scores = cross_val_score(model, X, Y, cv=10)
          print ('Cross-validated scores:', scores)

Cross-validated scores: [0.70821153 0.82402399 0.79008779 0.79435723 0.77402318 0.80906682
0.63293749 0.63883669 0.69015432 0.75351014]
```

```
In [184]: predictions_cross = cross_val_predict(model, X_test, Y_test, cv=10)
```

```
In [185]: accuracy = metrics.r2_score(Y_test, predictions_cross)
          print ('Cross-Predicted Accuracy:', accuracy)

Cross-Predicted Accuracy: 0.6843204783297963
```

```
In [186]: regressor.intercept_
```

```
Out[186]: 635.2227565433664
```

```
In [187]: regressor.coef_
```

```
Out[187]: array([-5.21202954e+01,  4.81831499e+01,  2.28854762e+02,  5.96276685e-01,
-6.77783960e-01, -2.64821953e+02, -1.02536504e+02, -3.48542091e+01,
-1.36021384e+02, -1.38347807e+02,  7.96748993e+01, -7.80795409e+01,
 4.30971233e+02,  2.44015266e+02, -5.37523903e+01, -6.47366261e+01,
-2.72958244e+02, -2.84395118e+02,  6.69593630e+01,  6.08883015e+02,
-2.43349723e+02,  1.07993882e+02, -1.71498990e+02, -4.47540648e+02,
-1.03849028e+02, -2.11565015e+02,  3.39114350e+02, -2.24155860e+01,
 6.66690760e+01,  4.03804614e+02, -9.85133756e+01,  7.83792891e+01,
-4.17548038e+02,  4.46560926e+02,  6.27482335e+02,  1.32170410e+02,
 8.57308265e+01, -4.24209810e+02, -1.47415493e+02,  8.21162265e-01,
-1.55593149e-03,  9.39485596e-01, -1.81174965e-01])
```

```
In [188]: Y_pred
```

```
Out[188]: array([ 246.96411845, 2347.9104074 , 1690.54541334,  506.61070426,
2002.07230479,  497.59547787,  340.6017101 ,  290.15747339,
1006.44157677, 1794.85056733, 1587.20718125,  692.75730346,
 327.33547804, 1175.47044818, 2345.78646879, 1009.56484105,
545.55384087, 1719.80150389, 2223.95820287, 2224.31939889,
388.45877279, 1736.94894711,  561.53584703,  988.44268056,
681.17388709,  863.43840513,  549.51739071,  629.15287647,
603.26154053, 1316.25018948,  774.0168628 ,  435.33791928,
1071.85823245,  706.78181227, 1757.42302423, 1131.49302552,
2446.07188983,  669.68225951, 1080.0453506 , 2235.68781302,
```

CPU ve GPU içeren one hot verilerin olduğu sete birde cross validation uyguladık

```
In [169]: from sklearn.cross_validation import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X2, Y2, test_size = 1/5, random_state = 0)
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarning: This module was deprecated in favor of the model_selection module into which all the refactored classes and functions are moved. Also the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20. "This module will be removed in 0.20.", DeprecationWarning)

```
In [170]: from sklearn.cross_validation import cross_val_score, cross_val_predict
from sklearn import metrics
scores = cross_val_score(model, X2, Y2, cv=10)
print ('Cross-validated scores:', scores)
```

Cross-validated scores: [-1.06388937e+15 -1.30210291e+14 -9.15192897e+14 -1.89148927e+15
-1.42126313e+12 -1.15878772e+14 -1.98182195e+10 -5.33696160e+13
-3.88719870e+14 -3.58527528e+14]

```
In [171]: predictions_cross = cross_val_predict(model, X_test, Y_test, cv=10)
```

```
In [172]: accuracy = metrics.r2_score(Y_test, predictions_cross)
print ('Cross-Predicted Accuracy:', accuracy)
```

Cross-Predicted Accuracy: -5.373466782128512e+16

- Linear Regression, sayısal girdi ve çıktı değerleri arasında ilişki kurmayı sağlar. İlişki doğrusal olarak modellenir.

Modeli Eğitme

X_train ve buna karşılık gelen y_train verisiyle modelimizi eğiteceğiz. Bunun için scikit-learn kütüphanesinden LinearRegression sınıfı import edilir

```
In [175]: from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
```


regressor nesnesi oluşturuldu. Şimdi fit() metodunu kullanarak model eğitilir. Parametre olarak yukarıda oluşturulan X_train, y_train eğitim setleri olarak kullanılır

Böylece basit lineer nesne oluşturmuş ve model eğitilmiş olur.

```
In [178]: model = regressor.fit(X_train, Y_train)
          model

Out[178]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

In [ ]:

In [179]: regressor.score(X_train, Y_train)

Out[179]: 0.7881761109835971

In [180]: regressor.score(X_test, Y_test)

Out[180]: 0.7294701587628417

In [182]: Y_pred = regressor.predict(X_test)
          Y_pred

Out[182]: array([ 246.96411845, 2347.9104074 , 1690.54541334,  506.61070426,
                  2002.07230479,  497.59547787,  340.6017101 ,  290.15747339,
                  1006.44157677, 1794.85056733, 1587.20718125,  692.75730346,
                  327.33547804, 1175.47044818, 2345.78646879, 1009.56484105,
                  545.55384087, 1719.80150389, 2223.95820287, 2224.31939889,
                  388.45877279, 1736.94894711,  561.53584703,  988.44268056,
```

Analiz sonuçları:

- **Mean Absolute Error (MAE)**

MAE, yönlerini dikkate almadan bir dizi tahmindeki hataların ortalama büyüklüğünü ölçer. Sürekli değişkenler için doğruluğu ölçer.

Kaydedilen sonuç : 283.7677532435216

- **Mean Squared Error (MSE)**

İstatistikte , bir kestiricinin (gözlemlenmemiş bir miktarı tahmin etmek için bir prosedürün) ortalama kareli hatası (MSE) veya ortalama kareli sapma (MSD), hataların veya sapmaların karelerinin ortalamasını ölçer – ki bu tahmin ediciler arasındaki farktır ve tahmin edilmektedir.

MSE, karesel hata kaybının veya kuadratik kayıpların beklenen değerine karşılık gelen bir risk fonksiyonudur .

Fark, rassallık nedeniyle veya tahmincinin daha doğru bir tahminde bulunabilecek bilgileri hesaba katmadığından kaynaklanır.

Kaydedilen sonuç : 159244.61583987987

- **Coefficient of Determination(R^2)**

Kaydedilen sonuç : 0.729470

R^2 Score ve Mean Absolute,Squared Score Hesaplama

```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
print('R^2 score is: %f' %(r2_score(Y_test,Y_pred)))
mae=mean_absolute_error(Y_test,Y_pred)
print('Mean absolute score is: %s' %(mae))
mse=mean_squared_error(Y_test,Y_pred)
print('Mean squared score is: %s:%mse )
#print("Percent of Mean absolute Error is : %",yuzde)#regressor.score(X_test,Y_test)
```

```
R^2 score is: 0.729470
Mean absolute score is: 283.7677532435216
Mean squared score is: 159244.61583987987:
```

İş Bölümü:

Veri Seti Bulma : Şevval Koçmar, Sedanur Eker

Veri Düzenleme : Şevval Koçmar, Hasan Reisoğlu, Sedanur Eker

Veri Görselleştirme : Şevval Koçmar, Hasan Reisoğlu, Sedanur Eker

Veri Eğitme : Şevval Koçmar, Hasan Reisoğlu, Sedanur Eker

Rapor Hazırlama : Sedanur Eker