**ML 07.07.2023**

**Supervised Learning**

**En basta yapilacaklar:**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**import scipy.stats as stats**

**plt.rcParams["figure.figsize"] = (10, 6)**

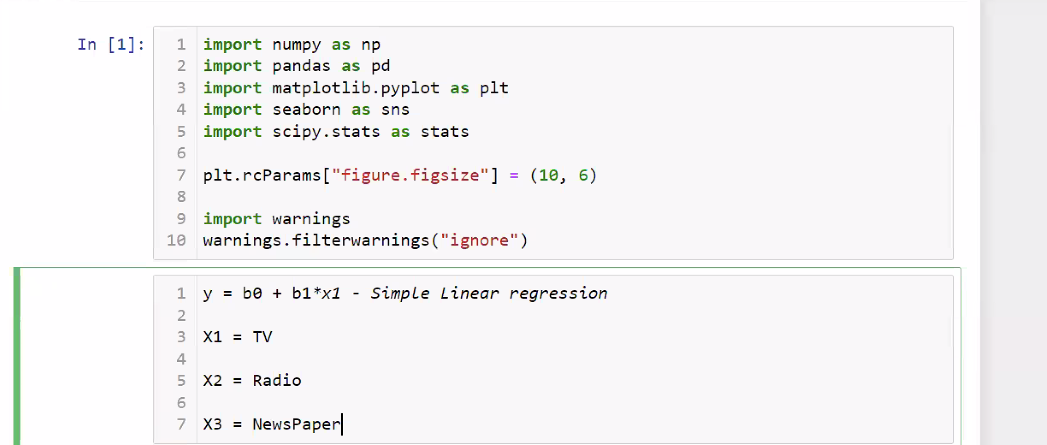
**import warnings**

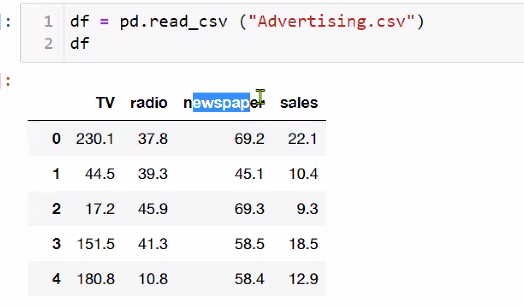
**warnings.filterwarnings("ignore")**

**!pip install missingno**

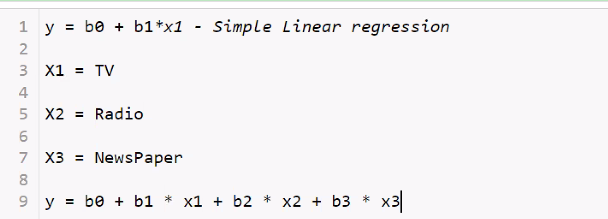
**!pip install scikit-learn**

**!pip install sklearn**

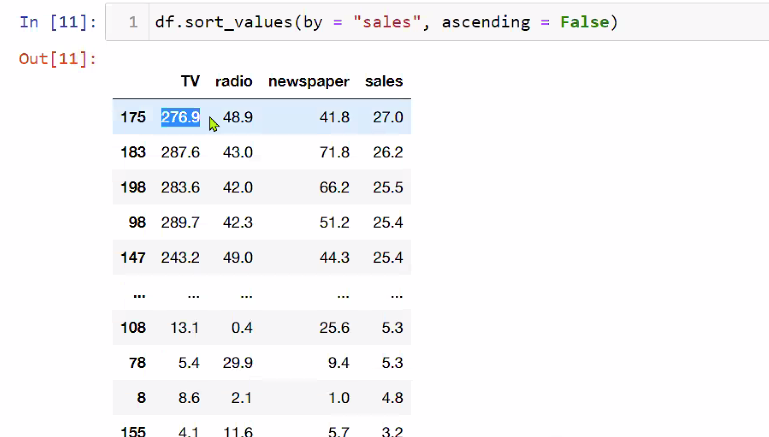




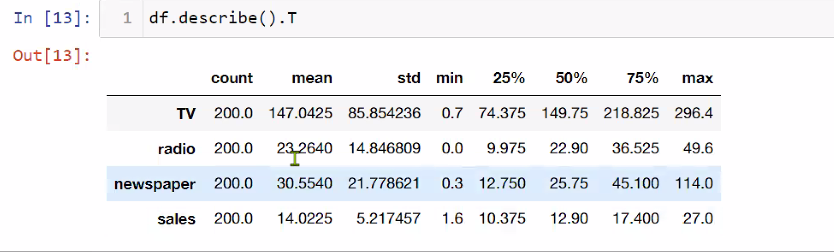
Burda multilinear regresyonda y’yi nasil yazarim?



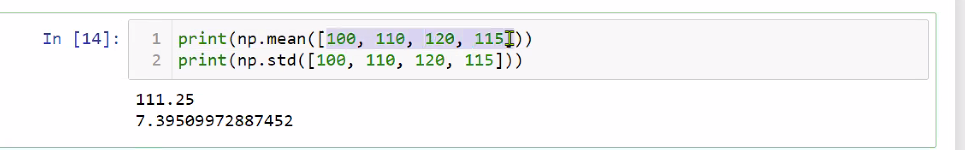
Multilinear’de birden fazla degisken, dolayisiyla birden fazla x var.



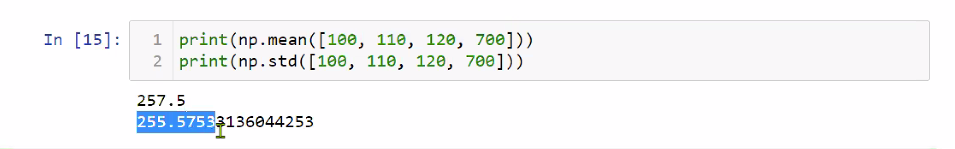
Sales’lari siralatiyorum sort\_values ile, bakiyorum reklam harcamalarina gore.



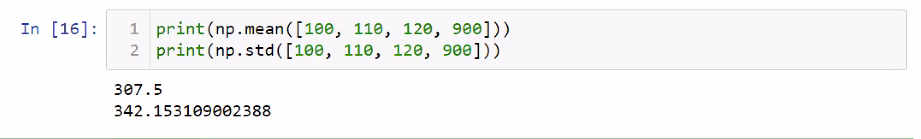
std mean’den buyukse outlier var demektir.



Mean 111, std 7 burda. Yani datalar birbirine yakin gibi



Ama ben outlier girersem data olarak, std mean’e cok yaklasir.



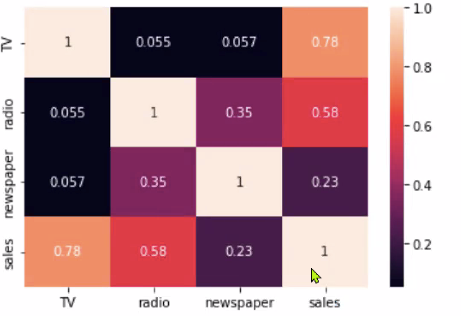
900 girince std mean’i de gecti--> bariz outlier var deriz.

Pairplotla da korelasyonlarina bakiyorum degiskenlerin.

Korelasyon dedigimde bir yonu olmali, kuvveti olmali.

En yuksek korelasyon tv ile sonra radio ile, newsaper ile cok az, hatta newspaper ile yok bile denebilir.

Heatmap cizdiriyorum.

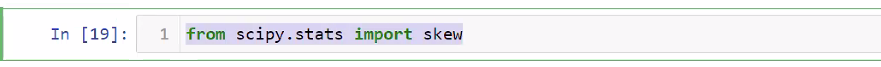


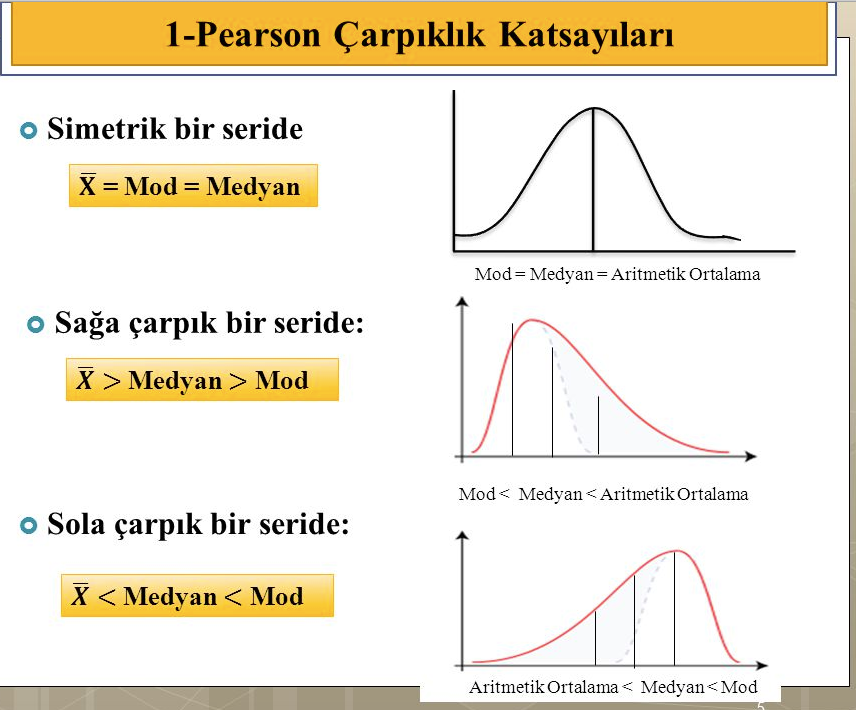
Korelasyon yoksa linearity’den bahsedemeyiz.

R2 0’a yakinsa zaten anlami yok deriz.

Basiklik ve cariklik degerlerinin -1 ve +1.5 arasinda olmasi verinin normal dagildigini gosterir. Literaturde bu rakami farkli gosterenler var.

**Skewness**

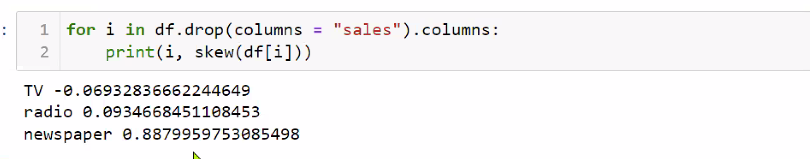




Saga carpik seride ortalama median’dan median da moddan daha buyuk oluyor.

Sola carpikta ise ortalama medyandan medyan da moddan kucuk oluyor.

Simdi target’i dropluyorum:



Bağımsız değişkenler (features) üzerinden bağımlı değişkeni (target) tahmin etmek istiyoruz. Bu yüzden tüm işimiz bağımsız değişkenler ile. O yuzden target’i yani sales’i droladim.

Bu kodla da Sales disindaki tum columnlar icin herbirine git skewness’ini al getir demis olduk.

for j in df.drop(columns = "sales").columns:

g = sns.distplot(df[j], color= "b", label= "Skewness : %.2f"%(df[j].skew()))

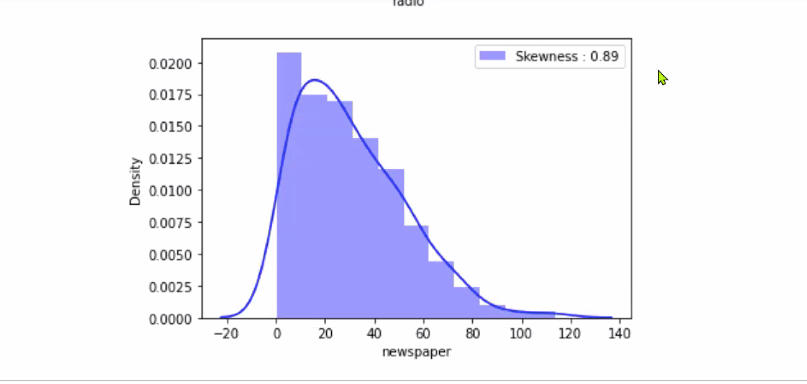
g = g.legend(loc= "best")

plt.show()



Eger bir dagilimda 2 tepe goruyorsan normal dagilim denebilir.

Tv ve radio icin normal dagilim var diyebiliriz.



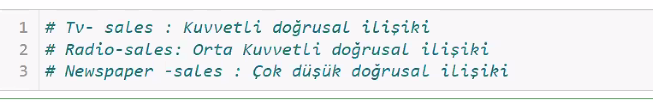
Newspaper’da saga carpiklik var.



# Tv- sales : Kuvvetli doğrusal ilişiki

# Radio-sales: Orta Kuvvetli doğrusal ilişiki

# Newspaper -sales : Çok düşük doğrusal ilişiki



**Train-Test Split**

Datayi ikiye bolecegiz.

Train datasi egitim yapacagimiz data olacak. Bunla elde ettigimiz skorlari test ile kiyaslayacagiz.

Trainde elde ettigimiz skor, testte de iyi sonuc veriyor mu diye bakacagiz

Asil kararimizi test sonucunda verecegiz.

En son da skorlar birbirine yakinsa, modelim genellestirilebilir diyecegim.

Normalizasyon regularizasyon hep train data uzerinde yapiliyor.

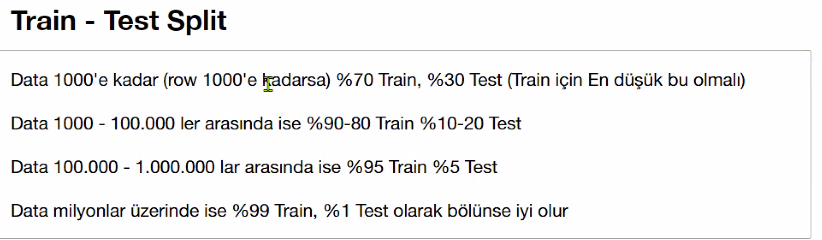
Fit kavrami--> train datasi uzerinde yapiyoruz.

Transform--> test datasina da uyguluyoruz

Hedefimiz her zaman train datasini yuksek tutmak.

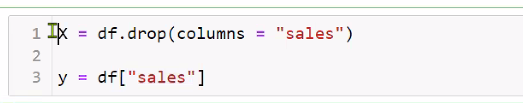
Gozlem sayim az ise ama sorun..limiti var bunun

Gozlem sayim 1000’e kadar ise--> %70 train-%30 test olarak datami bolecegim ve bu en dusuk oran olmali. En az %70’i train olmali



**Feature-Target ayrimi**

X-->feature, y-->target

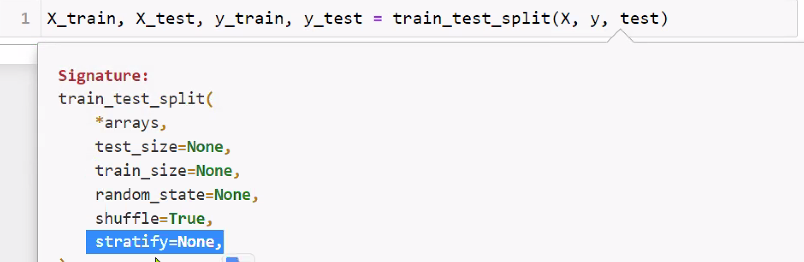


Sales’i dropladigim zaman sadece feature’lar kaldi, bu benim features’um oldu.

Sales da zaten target, dogrudan y’e atiyorum



Bunu sklearn kutuhanesinedn train\_test\_split i cagiriyorum.



Icindeki parametrelere bakinca, **test\_size** girmemi istiyor, %30 test size belirledim. Gozlem sayisi 1000’den az. O zaman data setinin %30’unu aliyor, test dataseti olarak ayirmis oluyor.



**random\_state=42- o**yle bir random secim yapiyor ki, herkesin ayni skoru alabilmesi icin

Bazen bir test verir, random\_state icin de rakam verir mesela. Herkesin ayni random\_state’i vermesi gerek ki, herkes ayni datayla calissin, sonuclar kiyaslanabilsin.

[https://www.youtube.com/watch?v=uM69OaS3d1g](https://www.youtube.com/watch?v=uM69OaS3d1g" \t "/Users/merthan/Desktop/PREVENTING EXCHANGE RATE HIKES/x/_blank)

(The number "42" was apparently chosen as a tribute to the "Hitch-hiker's Guide" books by Douglas Adams, as it was supposedly the answer to the great question of "Life, the universe, and everything" as calculated by a computer (named "Deep Thought") created specifically to solve it.)

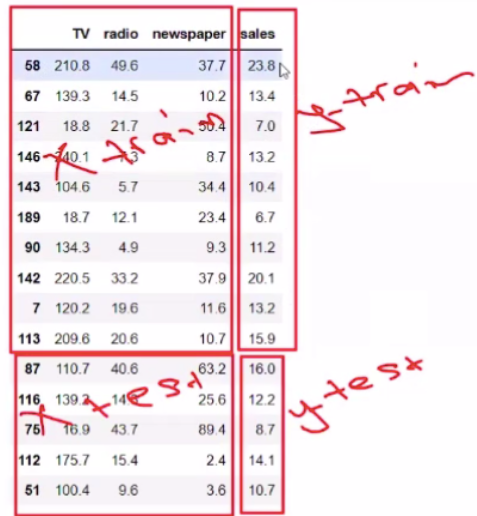
**Shuffle**--> pek kullanilmiyor. Karistirma duzenli getir dersem False derim ama

Unbalanced bir durum varsa. Binary bir target var diyelim. Yani 0-1 lerden olusuyor. Stratify kullaniliyor bu durumda. Hem 0’lardan hem 1’lerden aliyor hem teste hem traine.

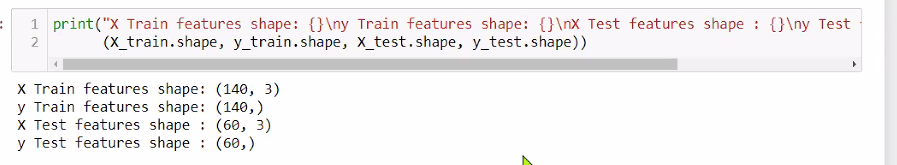
**Strarify** = y : (default None)

hedefin (targetin) orijinal veri kümesinde dağıtıldığı gibi eğitim ve test kümesinde eşit olarak dağıtılmasını sağlar.

Bu şekilde target oranı test ve train için korunmuş olacaktır.



y-train ile y-test kiyaslayacak sonunda.



print("X Train features shape: {}\ny Train features shape: {}\nX Test features shape : {}\ny Test features shape : {}".format

(X\_train.shape, y\_train.shape, X\_test.shape, y\_test.shape)) (edited)

#Print("Train features shape: ", X\_train.shape)

#Print("Train target shape: ", y\_train.shape)

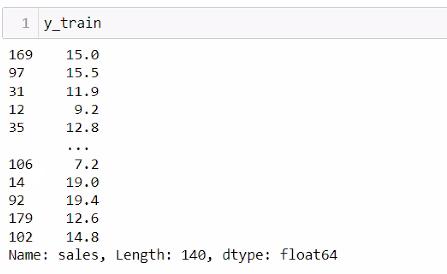
#Print("Test features shape: ", X\_test.shape)

#Print("Test target shape: ", y\_test.shape)

X train setimde 140 satir 3 sutun, testinde ise 60 satir 3 sutun var (tv, radio, newspaper)

Y trainde 140 y testinde 60 satir var.

Y\_train cagirdigimda, 140 satir oldugunu goruyorum.



**Model Fitting and Compare Actual and predicted Labels**

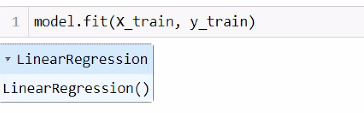


Model olarak linear regresyonu tanimliyorum

Model=LinearRegression() seklinde tanimladim.



Fit etmek demek egitim vermek demektir.



Model egitimi tamamlanmis oluyor bu kodla.

Feature’lardan elde ettigi bilgi ile targeti tahmin etmek icin bir egitim almis oldu model.

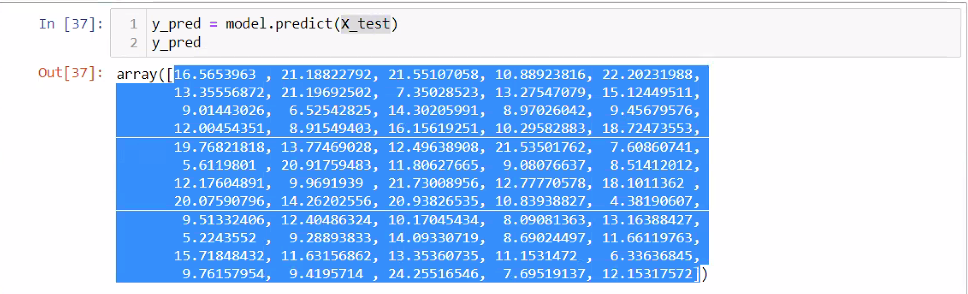
Modeli fit ettikten sonra sira tahmin alma islemine geliyor.

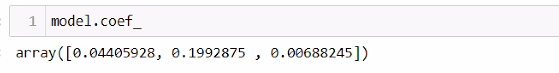
(normalizasyon, regularizasyon yaptigimiz zaman transform yaacagiz. Simdi gerek yok)

y\_pred= model.predict(X\_test)

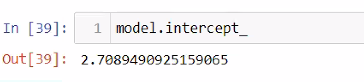
**Train--> fit ile**

**Test--> predict ile yapiliyor.**

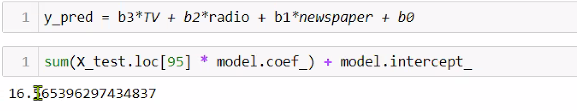




Bunlar katsayilar, yani egimler. Tv icin, radio icin ve newspaper icin. Bunlari cagirirken coefficient’ini cagiriyorum. model.coef\_



Interceptini de model.intercept\_ ile cagiriyorum. X 0 iken y’i kestigi noktayi verir bize.



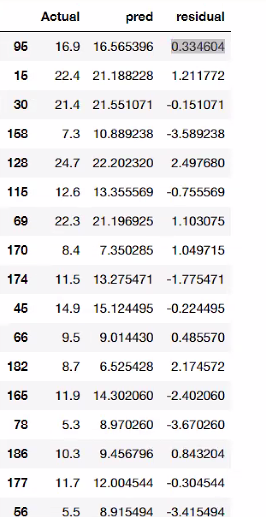
1. Indexteki degeri getirdim, manuel olarak test ettim. Tahmini getiriyor manuel olarak.



Actual, predict ve Residualimdan olusan bir dict olusturdum

**comparing = pd.DataFrame(my\_dict)**

**comparing**

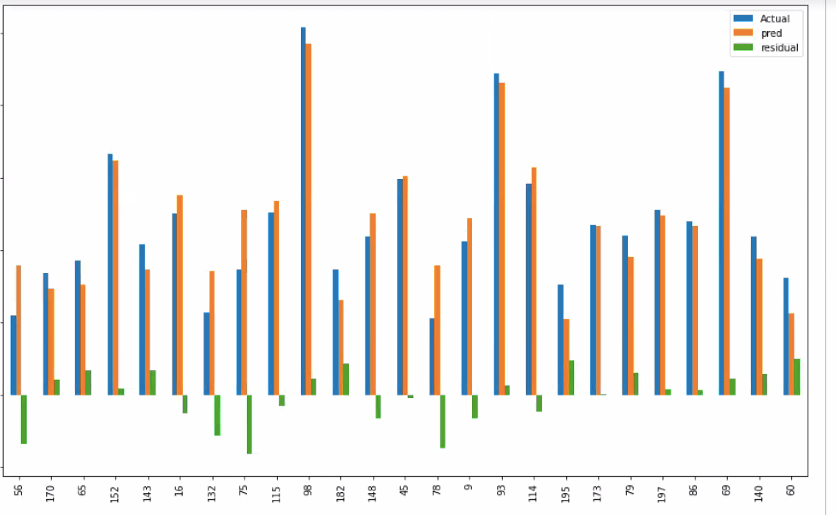


|  |
| --- |
| 1. Train ve test olarak datami boluyorum.(train\_test\_split) 2. Model’i linear regression olarak tanimliyorum.(model=LinearRegression()) 3. X-train ile y\_train’i model fit kullanarak egitiyorum (model.fit) 4. X\_test datasetini kullanarak bir y-predict elde ediyorum (model.predict) 5. Sonra y-redict ile y-actual’i, yani y-testi kiyasliyorum. Residual’lari elde ediyorum. Bakiyorum hata payim ne kadar diye. |

Result\_sample tanimliyorum.

result\_sample = comparing.sample(25)

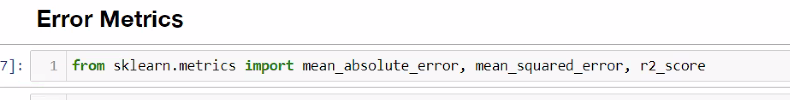




Gercek degerler ile tahmin edilen degerler arasindaki farklari yesil renkte residual olarak vermis.

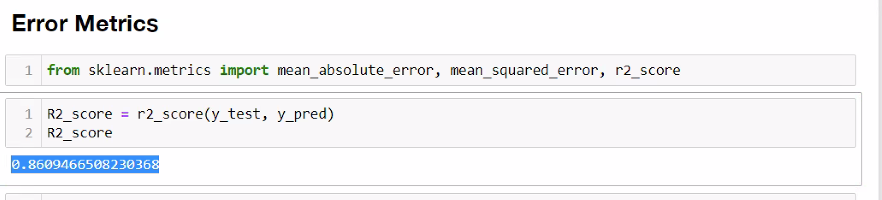
Buna error metric ile bakacagiz.

**Error Metrics**



Modelim ne kadar dogru calisiyor ona bakiyorum. Bunun icin cesitli metric’ler var. Bunlari sklearn’den cagiriyorum:

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score



<https://medium.com/academy-team/makine-%C3%B6%C4%9Frenmesi-regresyon-modellerindek[…]ne-g%C3%BCncel-bak%C4%B1%C5%9F-mae-mse-rmse-r-kare-d1ab758c5759>

R2 score 0.86 cikmis, y\_test, y\_pred

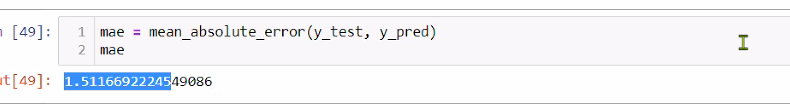
Kullandigimiz reklamlarla, satislardaki degisimin %86’sini aciklayabilmisiz.

Target’daki degisimin %86’sini bu featurelarla aciklayabiliyorum yani.

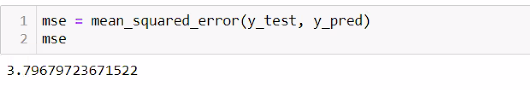
Aciklanan variance yani bu.

**Mean absolute error**

Residual’larin mutlak degerini aliyor. Ortalamasini aliyor.

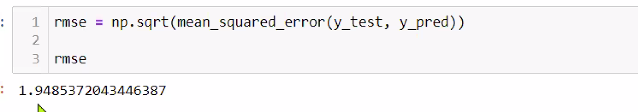


**Mean Square Error**



Hatalarin karesini aliyor topluyor, gozlem sayisina boluyor. Karelerinin ortalamasi yani

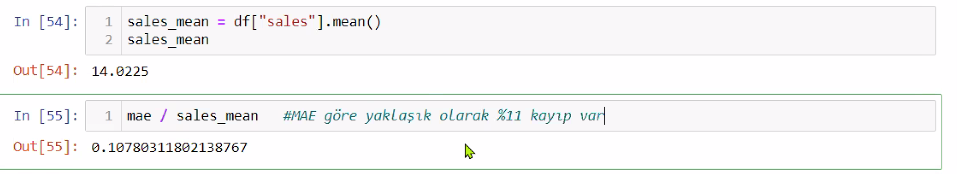
**Root Mean Square Error (RMSE)**



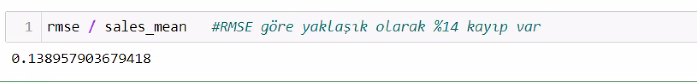
Mean Square Error’in karekokunu alarak buluyor. Buyuk hatalarina daha fazla hata veriyor. Yani cezalandirarak uzaklastiriyor kendinden.

Hata kucukse etkisi sinirli kaliyor.

Hangisi daha onemli R2 mi RMSE mi? Medium makalesinde var.



Kayiplar residuallardan kaynaklaniyor.



100bin tl lik arabanin fiyatini yaklasik olarak 89 bin ya da 86 bin olarak ya da 111 bin ya da 114 bin olarak tahmin edebilir demek bu.

|  |
| --- |
| Adem hocamdan:  \*\* R2 (R-squared), MSE (Mean Squared Error), MAE (Mean Absolute Error) ve RMSE (Root Mean Squared Error)\*\* gibi değerler, bir makine öğrenimi modelinin performansını değerlendirmek ve farklı modellerin veya parametre ayarlarının karşılaştırılması için kullanılan ölçütlerdir.  \*\* R2 (R-squared):\*\* R2, bir modelin bağımlı değişkenin varyansının ne kadarını açıkladığını gösteren bir istatistiksel ölçüdür. Değer aralığı 0 ile 1 arasında olup, 1'e ne kadar yakınsa, modelin bağımlı değişkeni daha iyi açıkladığı anlamına gelir. R2'nin yüksek olması, modelin gözlem verileri üzerinde iyi bir uyuşma sağladığını gösterir.  \*\* MSE (Mean Squared Error):\*\* MSE, modelin tahminlerinin gerçek değerlerden ne kadar uzak olduğunu ölçen bir hata ölçüsüdür. Her tahminin gerçek değerle arasındaki farkın karesi alınır, bu farkların ortalamasıdır. MSE, büyük hataları daha fazla cezalandırır ve kareler nedeniyle daha yüksek ağırlığı büyük hatalara verir.  \*\* MAE (Mean Absolute Error):\*\* MAE, modelin tahminlerinin gerçek değerlerden ortalama olarak ne kadar uzak olduğunu ölçen bir hata ölçüsüdür. Her tahminin gerçek değerle arasındaki mutlak farkın ortalamasıdır. MAE, büyük hataların daha az etkili olduğu MSE'ye kıyasla daha robust (daha dirençli) bir hata ölçüsüdür.  \*\* RMSE (Root Mean Squared Error):\*\* RMSE, MSE'nin kareköküdür. MSE gibi, gerçek ve tahmin değerleri arasındaki farkları ölçer, ancak RMSE'nin birimleri gerçek değer birimine eşittir. RMSE, büyük hatalara daha fazla ağırlık verir ve MSE'ye benzer bir hata ölçüsüdür.  Bu ölçütleri yorumlarken, daha düşük MSE, MAE ve RMSE değerleri, daha iyi bir tahmin performansını gösterir. Yani, bu değerler ne kadar küçükse, modelin tahminleri gerçek değerlere o kadar yakındır ve daha az hata yapmaktadır. Bunun yanı sıra, daha yüksek R2 değerleri, modelin bağımlı değişkeni daha iyi açıkladığını ve gözlem verilerine daha iyi uymayı başardığını gösterir.  Ancak, her bir ölçütün yorumlanması, uygulanan problem alanına ve veri setine bağlıdır. Bir ölçütün iyi veya kötü olduğunu kesin olarak belirlemek, problemin gerekliliklerine ve diğer faktörlere göre değişir. Ayrıca, bu değerleri tek başına yorumlamak yerine, farklı modellerin veya parametre ayarlarının sonuçlarını karşılaştırmak daha anlamlı olabilir. |

**Adjusted R2 (Duzenlenmis R2 Skor)**

**def adj\_r2(y\_test, y\_pred, df):**

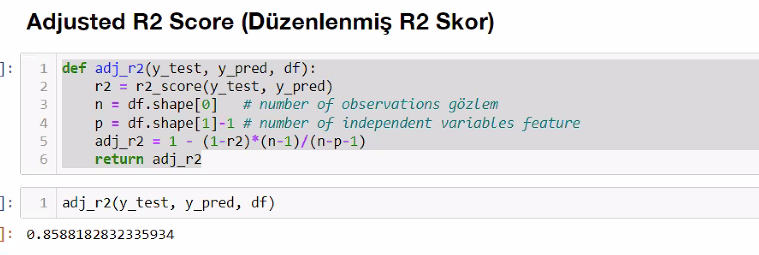
**r2 = r2\_score(y\_test, y\_pred)**

**n = df.shape[0] # number of observations gözlem**

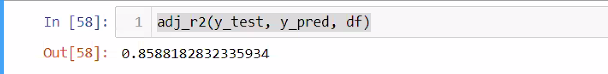
**p = df.shape[1]-1 # number of independent variables feature**

**adj\_r2 = 1 - (1-r2)\*(n-1)/(n-p-1)**

**return adj\_r2**



adj\_r2(y\_test, y\_pred, df)



Sadece feature ekleyerek R2’yi degistiriyorum :

**def adj\_r2(y\_test, y\_pred, df):**

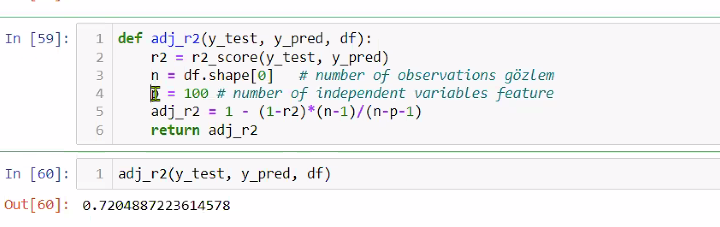
**r2 = r2\_score(y\_test, y\_pred)**

**n = df.shape[0] # number of observations gözlem**

**p = 100 # number of independent variables feature**

**adj\_r2 = 1 - (1-r2)\*(n-1)/(n-p-1)**

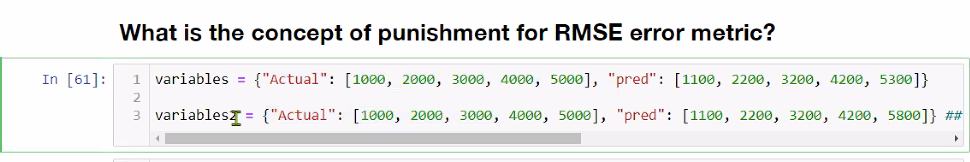
**return adj\_r2**

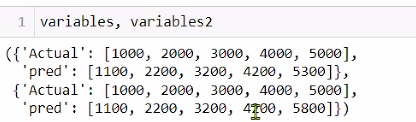


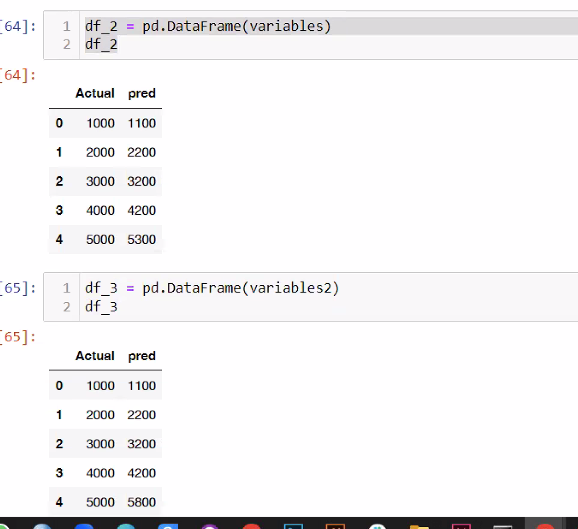
100 feature oldugunda, alacagim skor degisiyor. R2 0.72’ye dustu

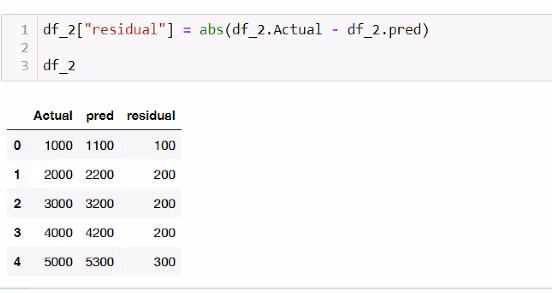
**RMSE**

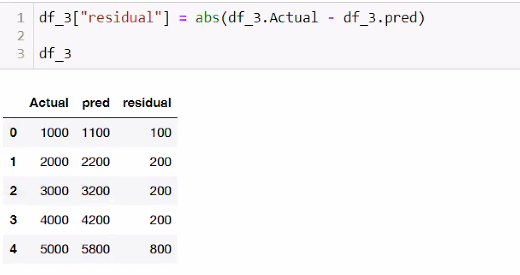
What is the concept of punishment for RMSE error metric?



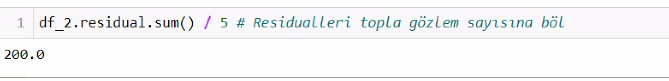


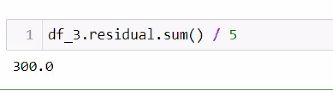






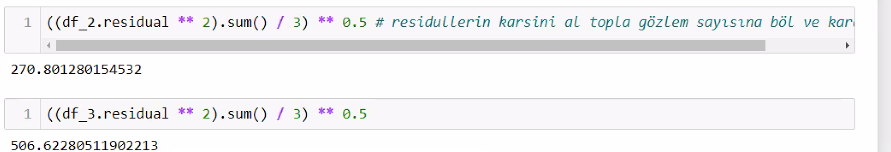
Islemimiz--> residuallari topla gozlem sayisine bol.





Aynisini RMSE ile yapalim:

((df\_2.residual \*\* 2).sum() / 3) \*\* 0.5 # residullerin karsini al topla gözlem sayısına böl ve karakökünü al



Ama RMSE ile baktigimda sonuc df2 ve df3 arasinda nerdeyse 2 kat fark var.

RMSE daha hassas bir metric aslinda.