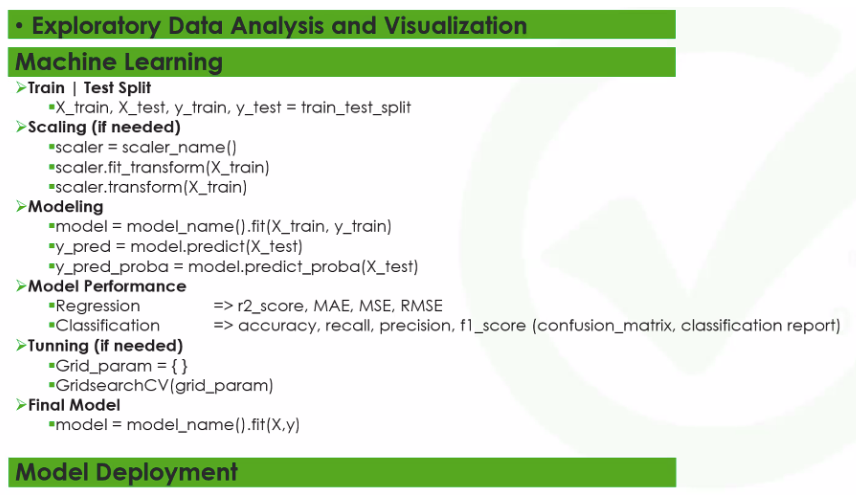
**REGULARIZATION 13.07.2023**

1. Train-test split
2. Fit
3. Scale
4. Prediction
5. Model performance
6. Tuning
7. Final-model

Model performance-->buna sonra bakacagiz. Sadece accuracy’e bakmiyoruz



**Steps of ML**

1. En basta targetimi y, featureslari da X olarak atiyorum. Sonrasinda Train ve test olarak datami boluyorum.(train\_test\_split) (min %70 train, %30test olacak)

2. Elimde X\_train, y\_train, X\_test, y\_test datasi olacak.

3. Model’i linear regression olarak tanimliyorum.(model=LinearRegression()). coefficient ve intercepti cagirarak katsayilari bulduruyorum

4. X-train ile y\_train’i model fit kullanarak egitiyorum (model.fit)

5. X\_test datasetini kullanarak bir y-predict elde ediyorum (model.predict)

Bunu yaparken elde ettigim regresson line’ina (y=aX1+bX2+cX3+dX4+e) X\_test verilerini vererek y\_predict elde etmis oluyorum. Burdaki a,b,c,d,e’yi de coefficient ve interceptleri cagirarak bulmustum.

1. Sonra y-predict ile y-actual’i, yani y-testi kiyasliyorum. Residual’lari elde ediyorum. Bakiyorum hata payim ne kadar diye. Bunu da eval\_metric ile error metricleri cagirarak yapiyorum.
2. Train datasetimin error metriclerini de cagirip, test sonucunda elde ettiklerim ile kiyasliyorum.
3. Boyle boyle denedigim modeller icinde en iyi sonuc aldigim modele, final\_model olarak tum datami sokuyorum.

**Regularization**

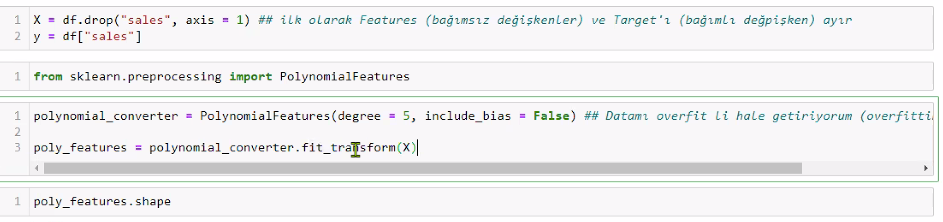
Scaling asamasina gelmeden once:

Yapacagimiz ilk islem-->feature-target ayir

Regularization in amaci oncelikle overfittingin onune gecmek

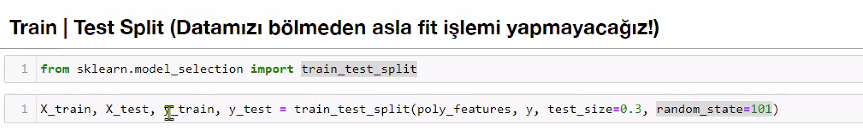
Once overfitli bir data olusturacagim.bunun icin 5. dereceye cevirecegim.

Asagidaki kodu 5.dereceye getirip overfitli hale getirmek icin yapiyorum.



200 satir 55 column dan olusan yeni bir datam var artik.

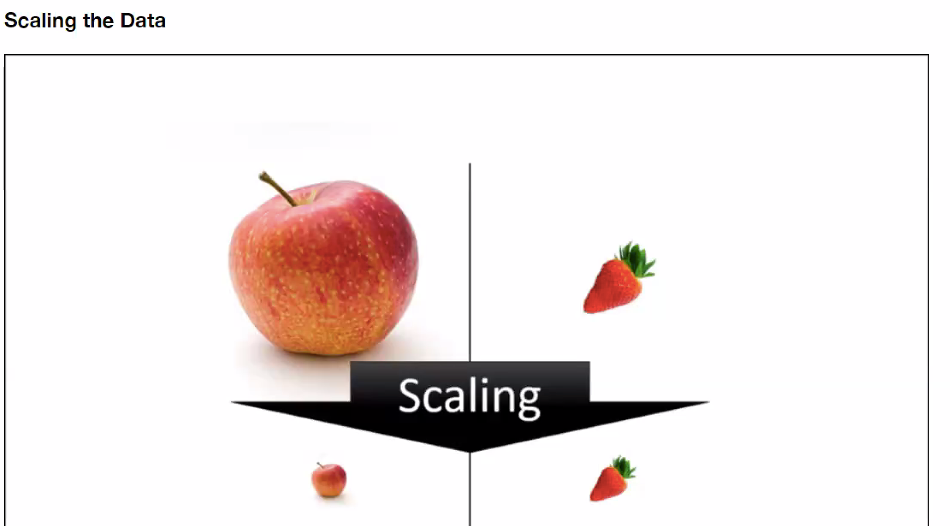
Yani 55 features oldu 5.dereceye cikarinca



Oncelikle train-test olarak ayiriyorum sonra fit ediyorum.

test\_size=30-->Datanin %30’unu test olarak ayirttim.

1. degreeye getirince olusan 55 adet features’im olan poly\_features’i da icine verdim.

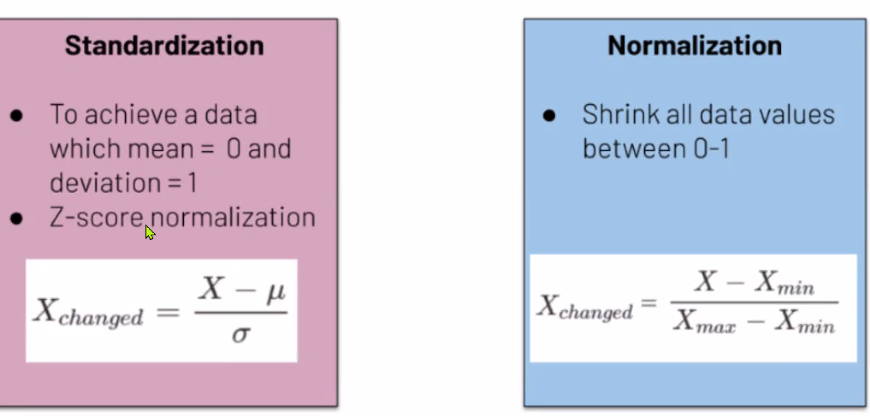


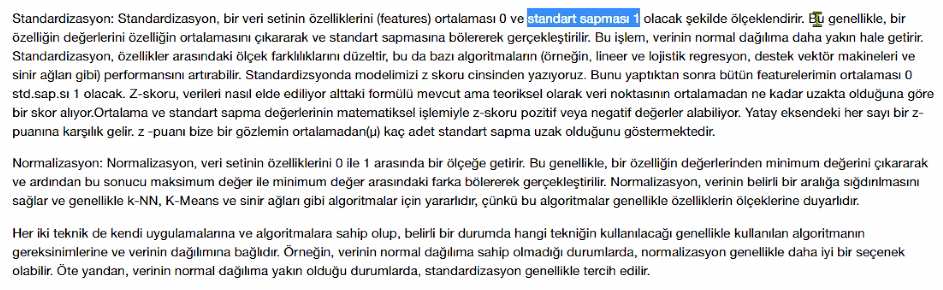
Featurelarin olceklerini uyumlu hale getiriyorum ki, target’a etkisini olcebilelim, kiyaslayabilelim.

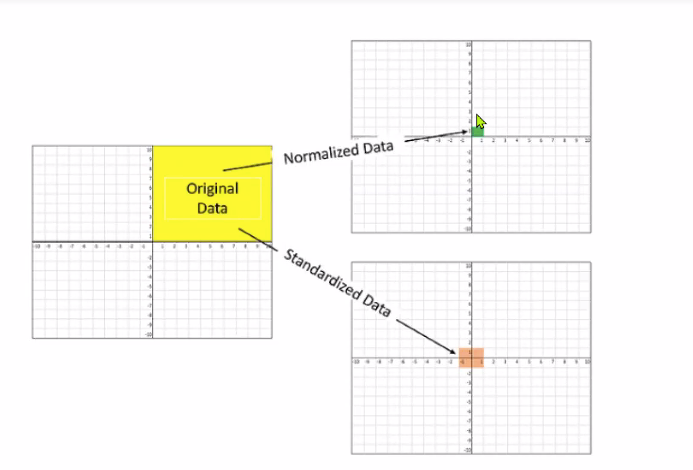
Scaling’in target’a yapilmasi meselesi:

Yayinlarda genelde yapilmadigi yonunde, yapildigi durumlar da varmis ama yapilmasa da genelde ayni sonuc ciktigi icin yapilmasinin gereksiz oldugu, egrileri cok da degistirmedigi yorumlari varmis.

**FEATURE SCALING**







**Normalizasyon:**

En buyuk degeri 1, en kucuk degeri 0’a esitleyerek degerleri bunun arasinda oranlayarak 0-1 arasinda yerlestiriyor.

Buyuk degerlere bilgisayarin daha fazla onem vermesinin onune gecmek ve feature’lari ayni olcuye getirmek icin yaptigimiz bir islem. Scaling

Serilerin normal dagilima yaklasmasini da sagliyor.

Ornegin X1--> balik yasi, X2-->balik yumurtasi olsun. X1--> 1,2,3,4 diye giderken, X2--> 1milyon, 2 milyon diye gidiyor. Bilgisayar balik yumurtasi feature’nun daha onemli oldugunu sanmasin diye

Adaletin onunde esit hale getirmek gibi.

**Standardizasyon:**

Benzer bir islem yapiyor ayni mantikla normalizasyon ile.

Ortalamasi 0, standard sapmasi 1 olacak sekilde yeniden deger atiyor degerlere.

Burda da degerler kuculuyor ama -’ler de giriyor burda icine ortalamasi 0 olmasi icin.

Her ikisi de scale islemi. Usenmeyip ikisini de deneyip onun sonuclarini kullanacagiz.

Sayilari belli araliklara indirip daha iyi analiz yapmamizi sagliyor.

Hata terimlerinin karelerinin toplaminin 0’a yakin olmasini istiyorum. Verilerim de kucuk olursa, hata miktarlarim da kuculmus oluyor normalizasyon ve standardizasyon ile. Verilerimi tren yoluna hapsetmis oluyorum, aralarinda bir regresyon dogrusu cizerek

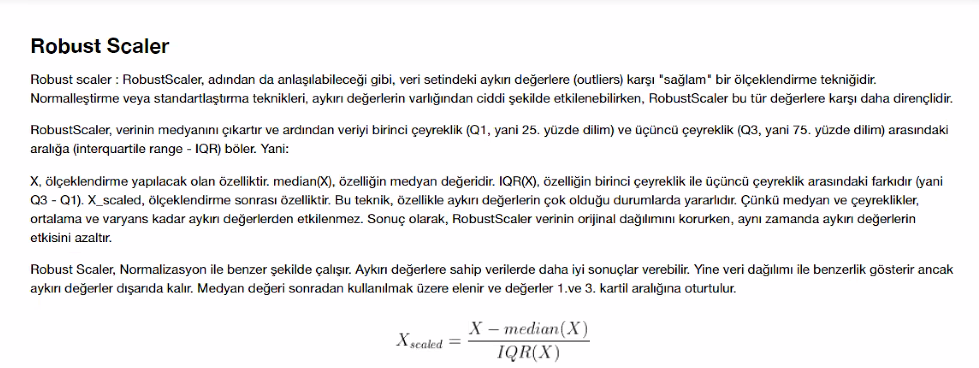
Analizin sonunda dusuk hata degerleri almis oluyorum, ama yine de yorumlayabiliyorum.

Minyaturk gibi dusun, esit olcekte kucultuyor, Selimiye’nin ayni olcegini koruyacak sekilde kucultuyor. Ayni olcekte kucultmezse zaten analizi bozmus oluyor.

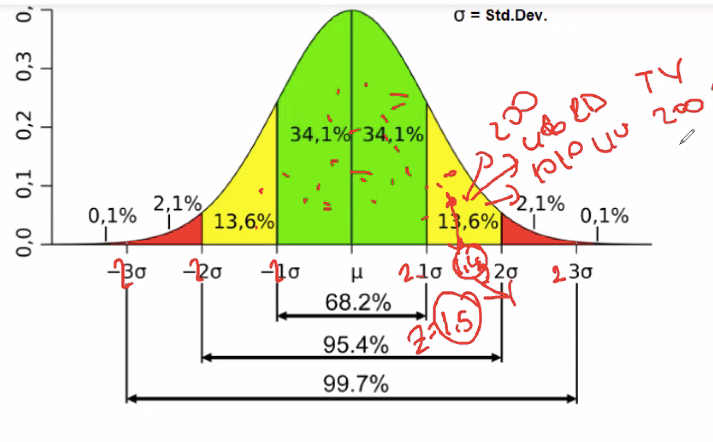
Bir datada normalizasyon yaparsan hepsine normalizasyon yap ki ayni dilden konussunlar. Standardizasyon yapiyorsan da digerlerine de standardizasyon yap

--Veri normal dagilima yakinsa--> standardizasyon

--veri normal dagilima yakin degilse-->normalizas



Robust Scaler--> genelde aykiri deger varsa tercih edilebilir. Ama biz zaten aykiri degerle mucadele edecegiz. Robust scaler yapmak cok da duzeltmiyor. Cok kullanilan bir yontem degil. Saglik datasi ile ugrasiyorsak ama bir robust scaler deneyim oyle karar vermekte fayda var.



Bu standard sapmalari -3Z ile +3Z arasinda degistigini dusunelim. Ornegin,

|  |  |  |
| --- | --- | --- |
| TV | 200 | 1.4 |
| Radio | 40 | 1.4 |
| Newspaper | 10 | 1.4 |
|  |  |  |

1Z ile 2Z arasindaki degerlerden belli bir cizgiye gelenler mesela artik 1.4’e denk gelmis oluyor scaling yapinca.

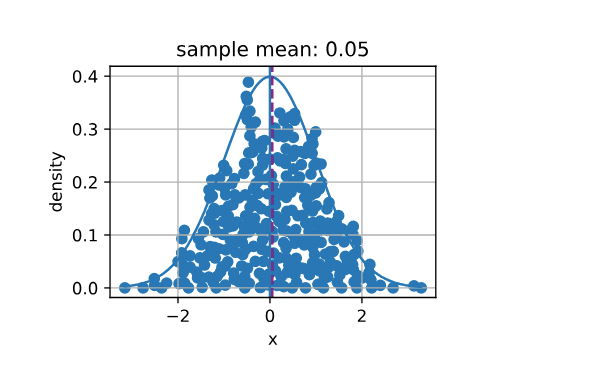
Asagidaki grafikte de ayni seyi goruyorum.mesela Tv column’daki degerler olsun asagidaki grafik. Datayi scale yapmadan once mesela 200 olan TV deki deger, 0, -2 arasinda nereye denk geliyordu ise, ona gore su an -1.5 olmus olacak

1-700 arasinda degisiyordu scale’den once, scale’dan sonra -3, +3 arasinda degisiyor

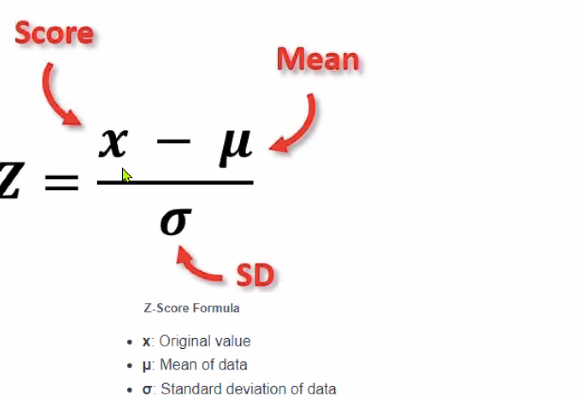
Cogunlugun degerleri:

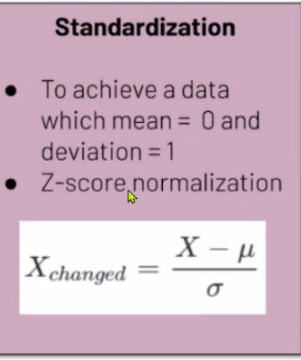
-Ortalamaya yakinlastikca--> z score kuculur

-Ortalamadan uzaklastikca--> z score buyur (mutlak)

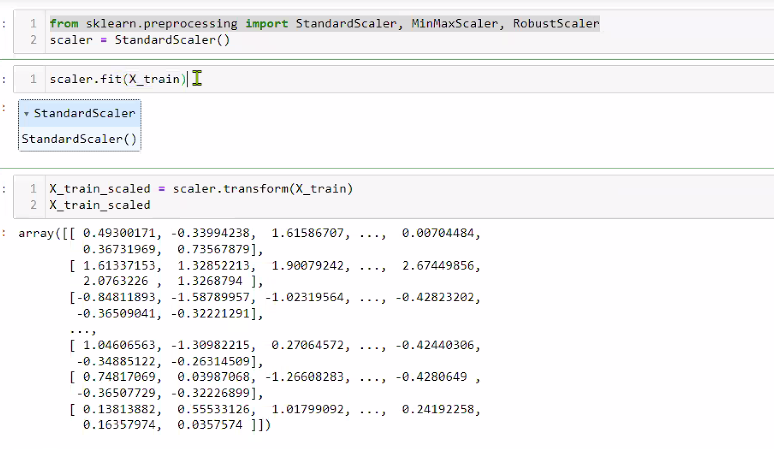


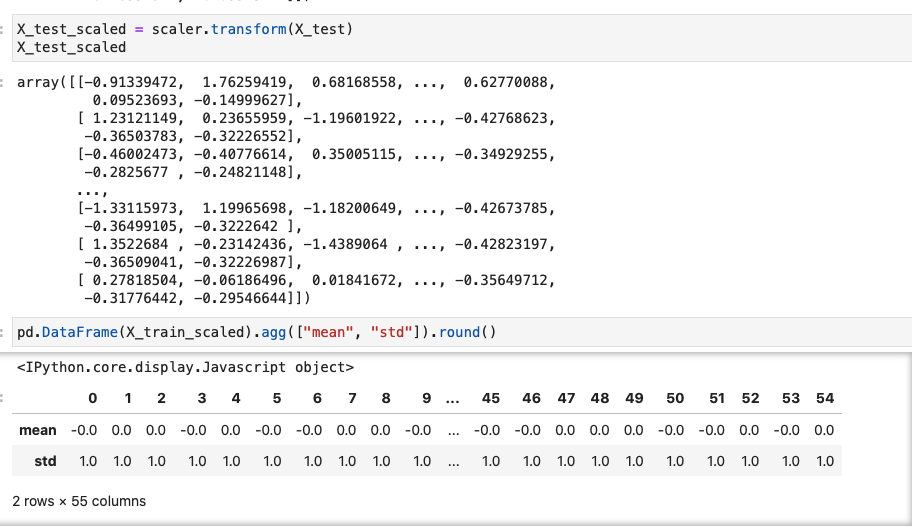
Tum gozlemlere asagidaki formulu uygulayarak buluyor bunu:

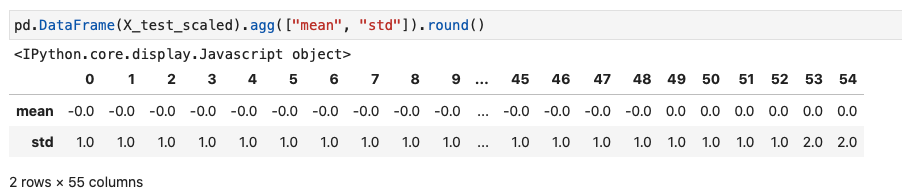




Tabi biz bunlari scikit learn den method cagirarak yapiyoruz







Traini once fit sonra transform ettim.

(test datasini sadece transform edecektim)

**Test datasini asla ve asla fit etmiyorum.** O zaman datayi gormus olur.

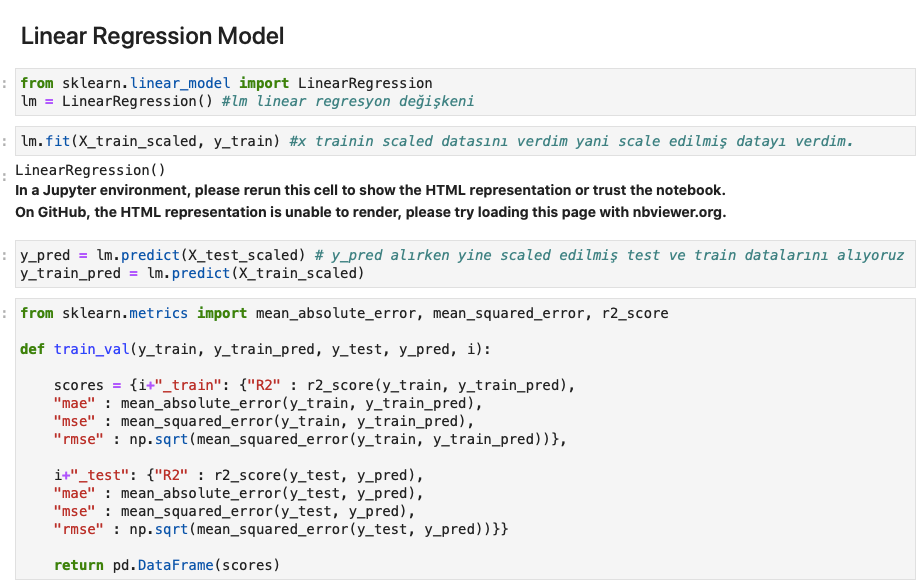
Standard scaler her bir gozleme formulu uygulayip yeni bir z score elde ediyordu.

Bunu train datasinda yapiyor once. Ortalamasi 0, standard sapmasi 1 olacak.

Hem fit islemi hem de transformu ayni train datada yaptigi icin sorun olmaz. Ortalamasi 0, std’si 1 cikar.

Ama ben test datami fit etmedigim yani egitmedigim icin, sadece transform yaptigim icin, std olarak 2’leri goruyorum. Train datasindaki max degerden, daha da fazla olan degerler varmis demekki test datamda o yuzden degerler 2’lerde cikti. Bunu yanlis olarak gormuyrum, bilakis test datasini gormedigi icin bu sonuclar ciktigindan that’s fine.

En son final modelde tum datayla girdigim icin modele, son asamaya kadar test datasini modele hic gostermiyorum.



Adini lm diye verdik bu sefer linear regression’in.

Feature’larin scale’indan sonra modelde iyilesmeler oluyor.

Simdi modelin icine (lm.fit) scale edilmis train datami verdim. Egitimi tamamlandi

Artik predict islemine geciyorum. Icine de yine scale edilmis x\_test’i veriyorum

Yukarida scale edilmis train ve test datasini modele aktariyorum yani

Burda predict var yukarida transform vardi.

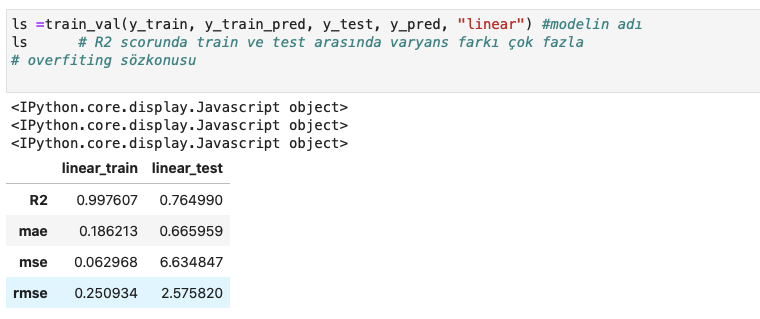
Yukarida transformu hem traine hem teste yapiyordum.

Burda da predicti hem teste hem traine yapiyoruz.

Modelim test datasinda aldigi skorla train datasinda aldigi skor uuymlu mu ona bakiyorum.

Uyumsuzsa 2 sebebi vardi--> underfitting, overfitting

Yukaridaki kod blogu ile buna bakiyorum.



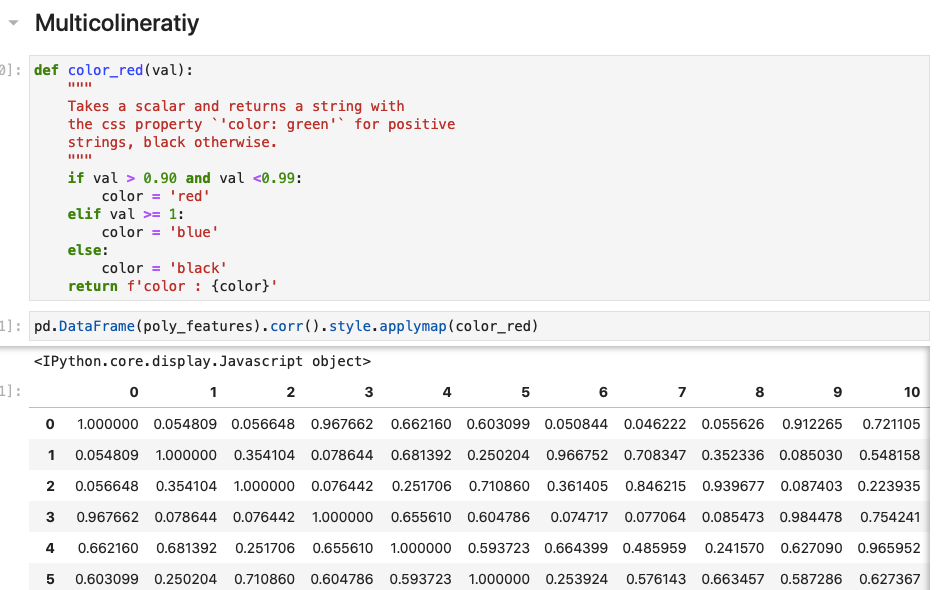
Train datamda aldigim skorla test datasinda aldigim skorlari karsilastirmak icin bu kodu yazdirdim.

RMSE 10 katina cikmis

R2 dusmus.

**linear\_test skorum train setimle uyumlu degil. Boyle bir durum soz konusu oldugunda ben her zaman overfitting var diyecegim.**

Normalde R2 0,99 ise overfitting’den suphelenirim. Ama net bir limit yok overfitting demek icin. Aslinda overfitting icin train ve test arasindaki fark asil onemli olan. Ben zaten overfitting ve multicolineratiy varmis gibi yola devam edecegiz.



0.8’in uzeri dogrusallik sorunu dogurur makalelerde boyle gecer

Ama ML alaninda threshold 0.90-->bu featurelar ayni seyi olcuyor demektir.

Fahreneit ve celcius ayni tablodaysa, aralarinda 0.99 korelasyon var demektir.

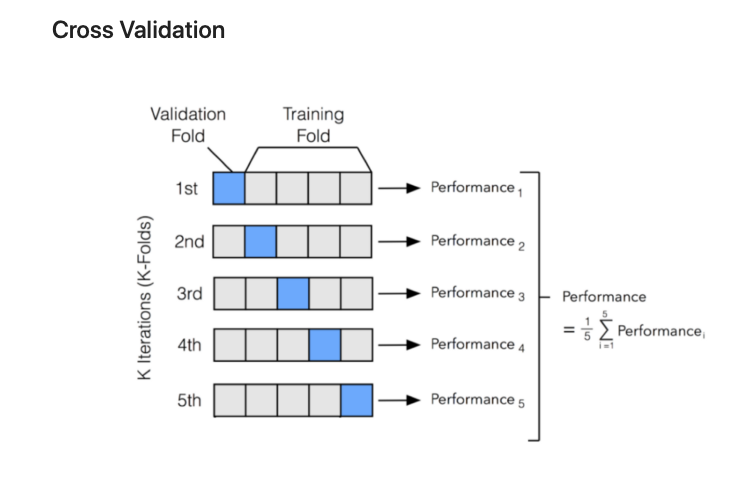
Korelasyon matrisini gormek icin feature’larda, bu kodu yazdik

Heatmap gibi aslinda

1 degerlerinden simetriktir. Diagonal’da kendileri ile korelasyonu vra.

Dolayisiyla sadece 1’in altindakileri, ya da sadece ustundekileri yorumlayabilirim.

Ozetle 0.90 uzeri korelasyon varsa--> multicolinearity sorunuu



**Cross Validation**

Train datasinin icinden bir kisimdan skor aliyor. Test datasi ile skoru ile kiyasliyor. Test datasi olarak ayirdigim kisim acaba datasetinin kotu bir kismindan mi denk geldi, ondan mi skorlar farkli cikti diye bir dogrulama islemi aslinda bu.

Modelin randomly sectigi yerden olusan ilk train dataseti icin hesaplattigimiz R2’yi, Cross validation sonrasi traindataseti icin hesaplattigimiz yeni R2 ile kiyasliyoruz. Train dataseti icin yeni bir R2 hesaplatiyorum ki, aslinda iyi olan overfittingi olmayan bir modelim icin yanlislikla overfitting yorumu yapmayayim. Ya da aslinda overfitting olmayan bir durum icin model basarili diyip gecmeyelim.

Cross Validation icin de, train datasetini 5-10-15.. parca haline boldurup, her bir parca icin ayri bir train-test split yaptirip, yeni R2’ler olusturuyorum ve ortalamasini aldirip yeni bir R2 hesaplatiyorum. Bu yeni R2 aslinda benim test skorumu temsil edecek. Ben bunu train skorlari ile kiyaslayip ona gore yorum yapacagim.

CV ile elde ettigim skorumu test skorumu temsil edecek ben onu train ile kiyaslayacagim, o zaman benim gercek test datasetim %30'luk kisim ignore oldu. ben %70lik datamin icinden CV yaparak hem train hem test skoru elde etmis oldum, birbiri ile kiyasliyorum.

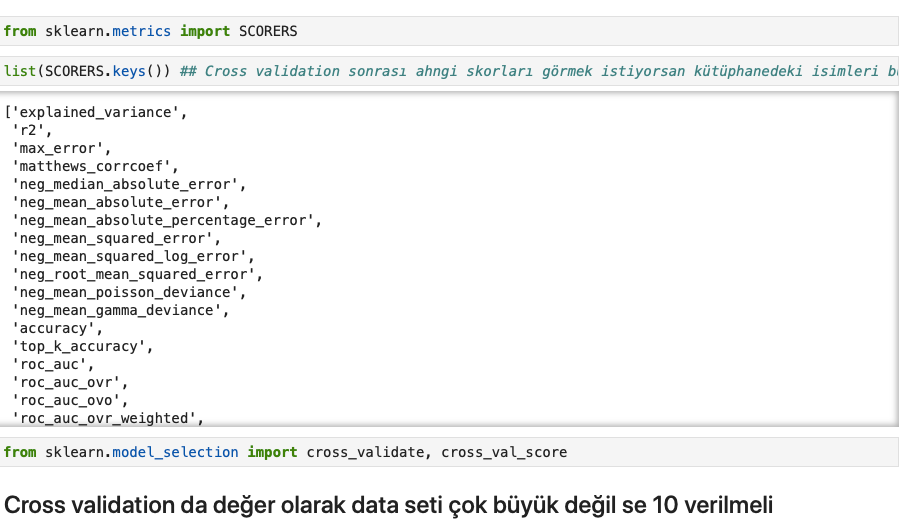
Ama CV sonrasi tum islemlerime risge lasso and others zaten hic yapmamis gibi tum datasetini dikkate alarak devam ediyorum. Bu sadece arada bir validasyon islemi. Overfitting sorunu varsa zaten bir modelde, muhtemelen CV’de de oyle cikacaktir. Cok buyuk ihtimalle.

Burada çıkan skor test-train skorları ile karşılaştırılacak.CV sonrası alınan skorun Train skor ile uyumlu olması modelin genelleme yeteneğinin olduğunu gösterir dolayısyla overfit yok demek olabilir. Ancak bu durumda test datasının datanın kötü bir bölümünden gelmiş olması (100.000 satırlık veriden 1000 satırlık bir test datası ayrıldığında) düşünülebilir bu durum söz konusu değilse yinede test datasının skoru göz önünde bulundurularak overfit durumu var denilir.

CV ile elde ettigim skoru train dataseti R2 ile kiyasliyorum. bunlar uyumlu ise diyorum ki, aslinda sorun Test datasetinin datanin kotu bir yerinden gelmis oldugu yorumunu yapiyorum. Ama bu cok cok dusuk bir ihtimal. normalde train-test split yaptiktan sonra test ve train skorlarini kiyaslayarak overfitting varsa, buyuk ihtimalle overfitting cikacak zaten CV sonrasinda da.

CV ile alinan skor train ile uyumlu olmasi--> modelin genelleme yeteneginin oldugunu gosterir dolayisiyla overfitting yok demektir. Test datasinin datanin kotu bir yerinden gelmis olmasi dusunulebilir.

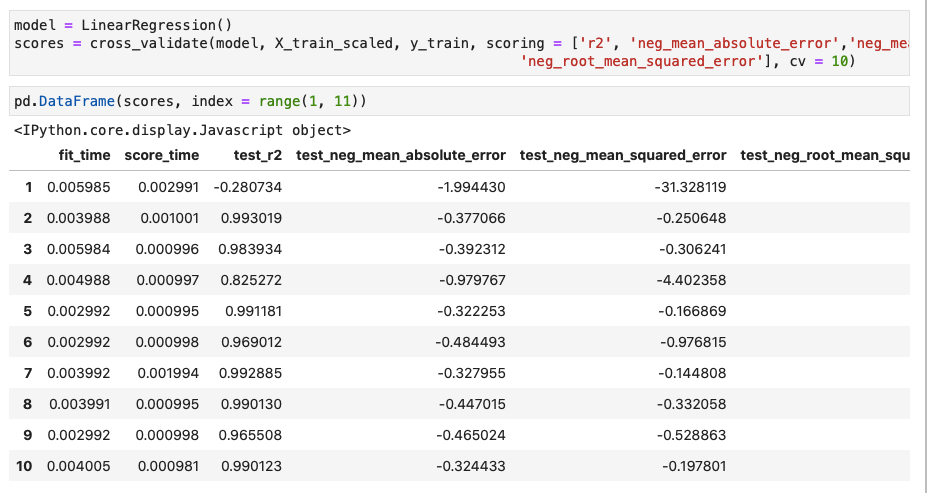
Aslinda Grid Search ile Tunning asamasinda, zaten embedded olarak cross-validation yapmis oluyoruz. Modelin icinde cv seklinde zaten giriyoruz. Mutlaka yapmis oluyoruz yani cross validation.

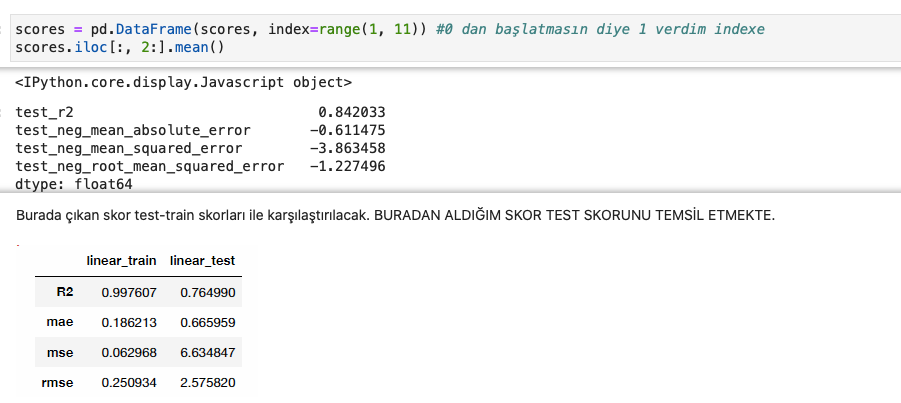


Icine train edilmis datami verdim.

Projede de 10 CV yap. Ne kadar yuksek verirsek datanin degerleri o kadar iyi cikiyor.

Default 5 yeterli olmuyor.





**CV sonrasi**

10 tane CV yap deyince bana 10 tane skor uretti. Ben bunlarin ortalamasini aliyorum.

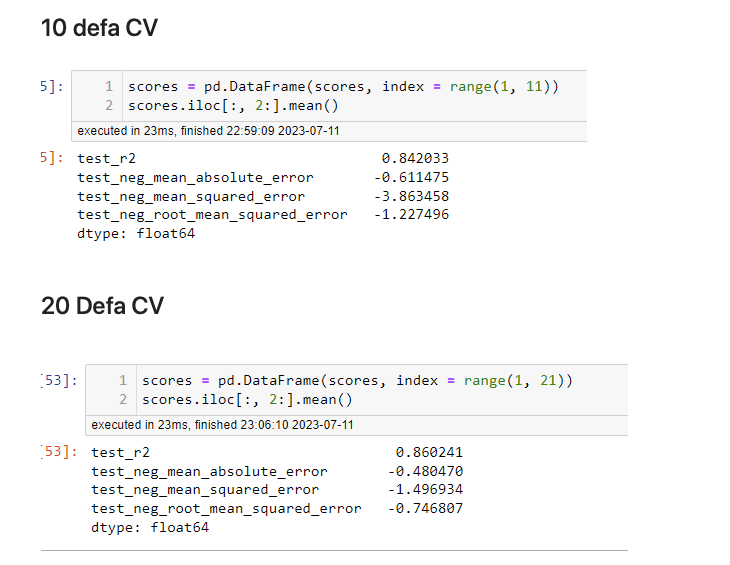
CV Sonrasinda aldigim skor ise 0.84 R2 icin.

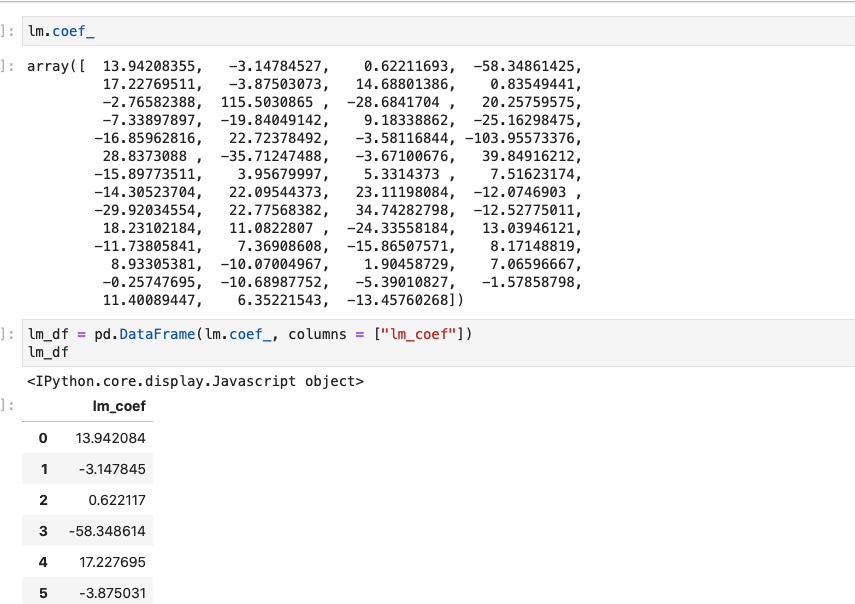
Aldigim skor test skorunu temsil ediyor.

**Mae absolute RMSE arasindaki farkin cok yuksek olmasi outlieri gosterir.**

**CV skorunu ben evet train datasetinden elde ediyorum ama bu benim aslinda test skorumu temsil ediyor. Yani mini bir train-test skoru olusuyor. Dolayisiyla aldigim skorlar test datasini temsil ediyor.**

**CV ile aslinda biz train icinden yeni bir test dataseti cikariyoruz, onun skorlarini buluyoruz dolayisiyla bu skoru train ile kiyasliyoruz. dolayisiyla yeni skorlar ile train uyumlu ise sorun yok diyecegim.**





1. Indeks degerinde cikan 115 degeri--> en onemli feature oldugu icin degil ama

Bize feature selection icin en cok lasso yardim edecek