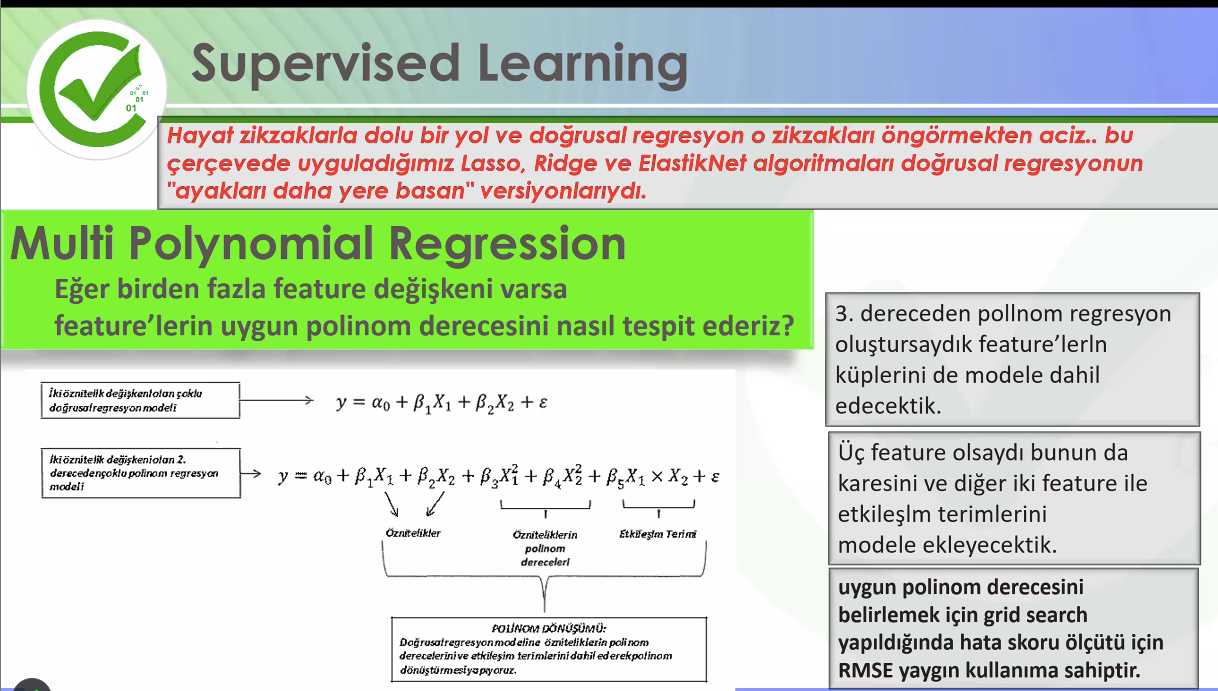
**ML 10.07.2023**

**Multipolynomial Regression**

Egitim setini daha iyi tahmin edebiliyor ama test setinde de mi oyle orasi supheli.

ML’de cok da kullanilmiyor. Cunku modeli asiri komplex hale getirmis oluyoruz. Bias-variance trade-off nedeniyle Overfittinge gidebilir.



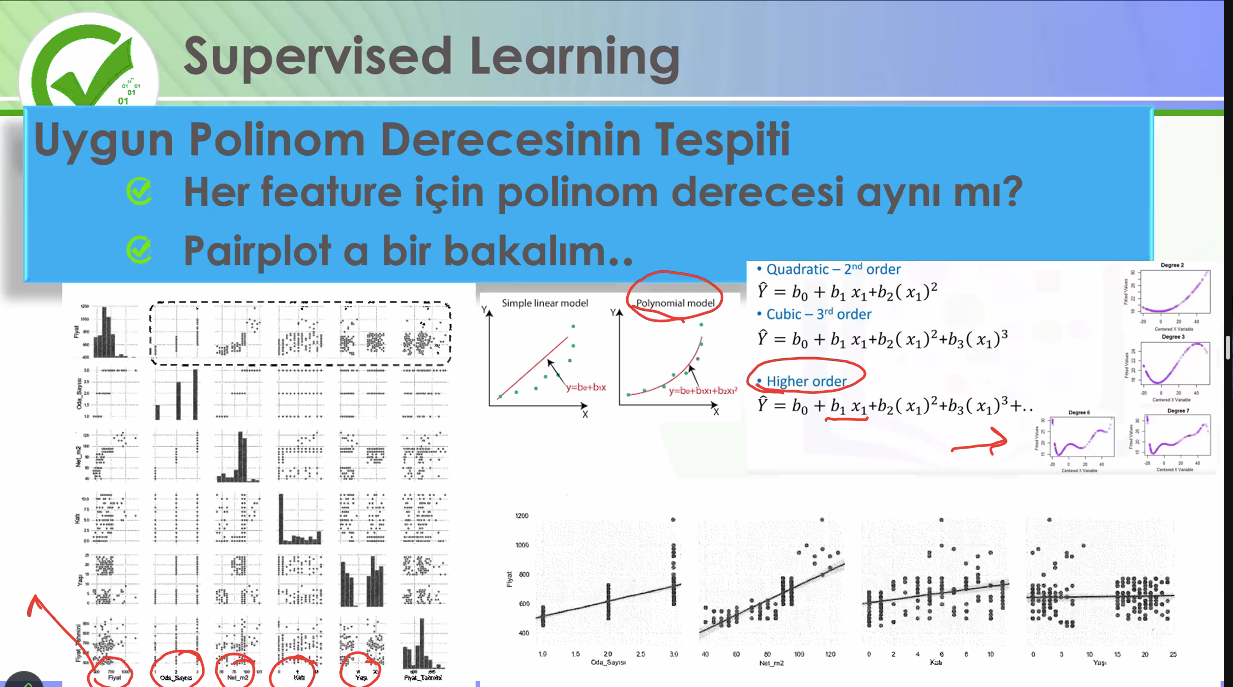
Scikit learn kutuhanesinin olynomial fonk.nu kullaniyoruz.

1. derece demek, 3 feature icin, 9 degisken eklemek demek. Cok kompleks hale geliyor model.

Uygun polynom derecesini belirlemek icin grid search yaptigimizda RMSE’ye bakilabilir.

##An Introduction to Statistical Learning kitabi-Python versiyonu-yeni cikti.

Emlak fiati ile ilgili oda sayisi, net m2, binanin yasi gibi parametreler vardi. Hangisi hangi derecede girecek modele. Bunun tespiti onemli.

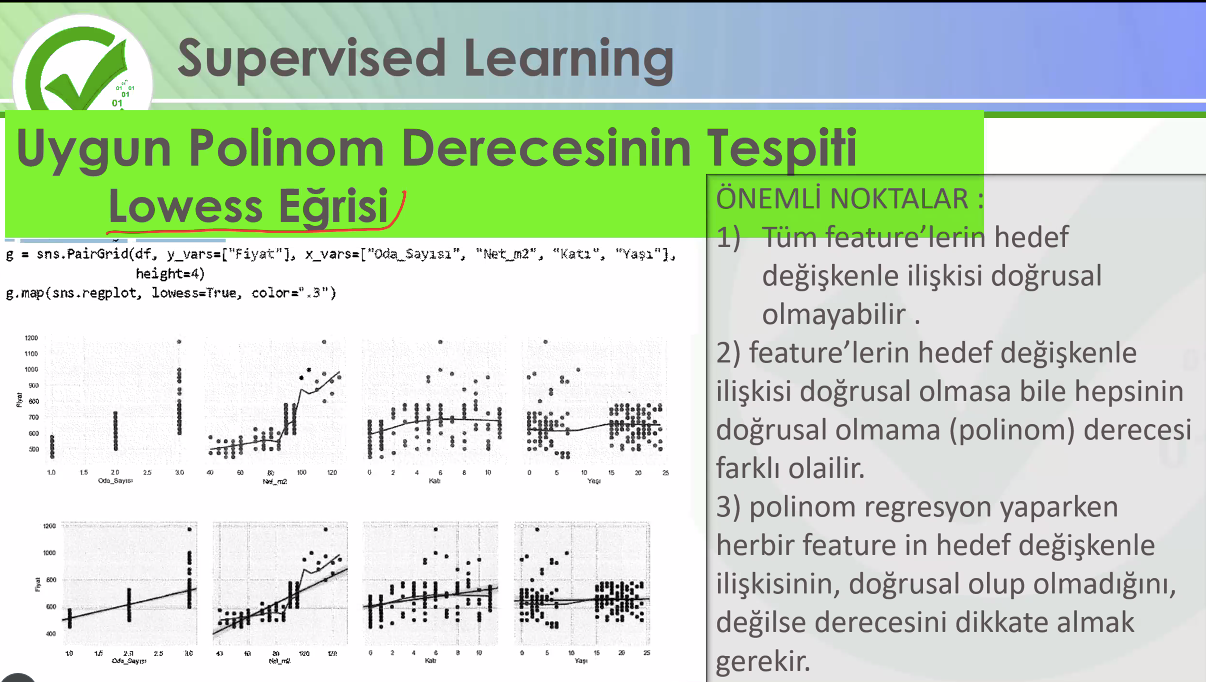


Train sette %84’te olan tahmin performansi test sette %76’ya dusuyorsa variance sorunu var.

1.Target variable Feature ile dogrusal iliskiye mi sahip?

2.Egrisel iliskiye sahipse ayni derecede mi egrisel olanlarla iliskisi?

**Lowess Egrisi**



Lowess egrisi-->Target ile iliskisinin dogrusal olup olmadigi hakkinda bilgi veriyor.

yakin olanlari yuksek agirliklandiriyor

Lokal regresyon egrilerine bakinca, net m2’de dogrusal olmayan bir durum soz konusu

Kat ve yasta da dogrusal olmadigini goruyoruz.

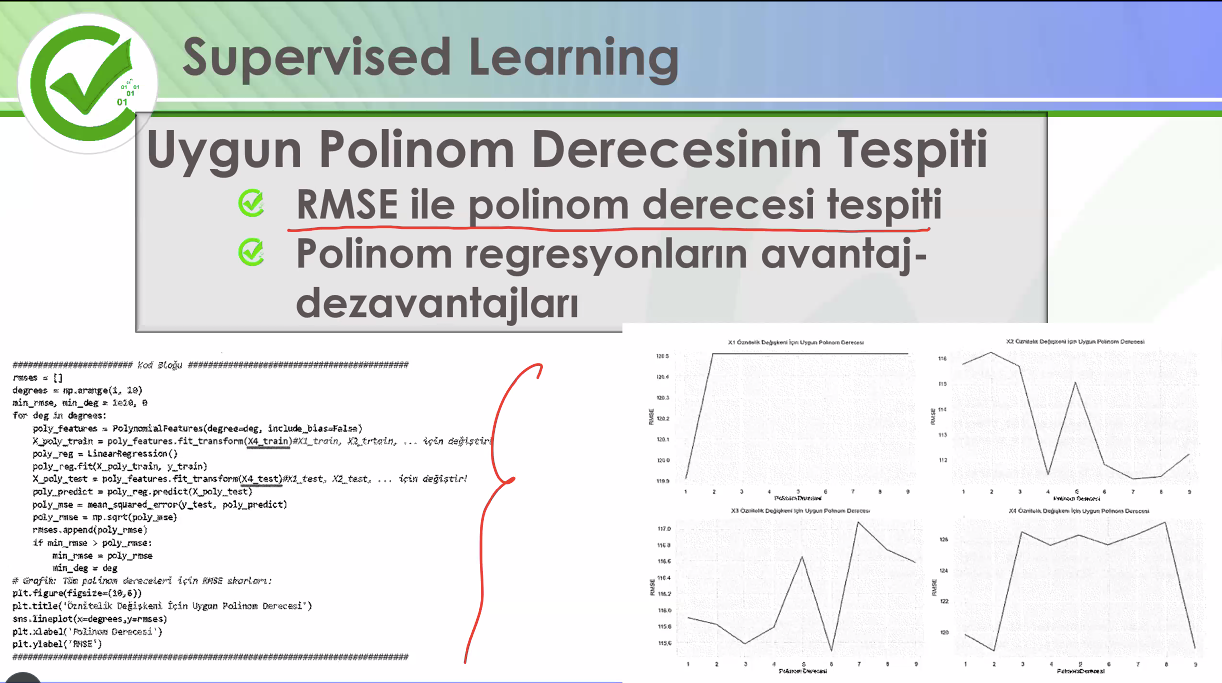
1.Tum featurelarin target ile iliskisi dogrusal olmayabilir

2.Iliski egrisel ise illa ayni derecede egrisellik olmak zorunda degil

3.Regresyon yapilinca iliskinin dogrusal olup olmadigini sorgulayacagiz, degilse hangi derece diye sorgulayacagiz

Yaklasimimizi RMSE uzerinden degerlendirecegiz. R2 bu konuda outliera daha duyarli.

Herbir feature’a gore kismi regresyonunu uygulayacagiz.



Feature’lari 1’den 10’a kadar polynom derecesi alacak sekilde siralandiriyor.

En dusuk RMSE’yi veren dikkate aliniyor.

Herbir polynom derecesi icin RMSE’si min olan bizim icin gecerli olacak.

En dusuk RMSE 1. derecede saglanmissa o zaman dogrusal iliski var diyecegiz.

**Oda sayisi** acisindan bakarsan mesela, **X1**, en dusuk RMSE’yi 1. derecede vermis. O zaman X1, 1. derecede modele girecek.

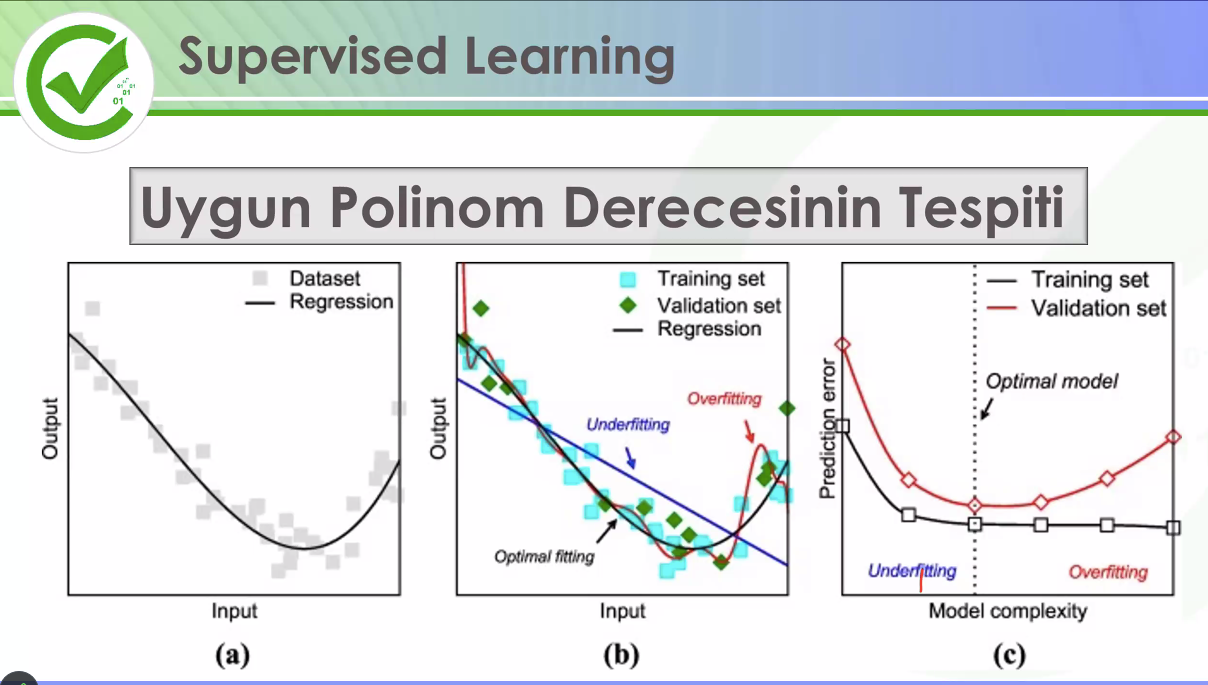
Modelin kompleksligini artirdikca overfittinge gitme riski var. Train iyi cikar, test basarisi azalir.

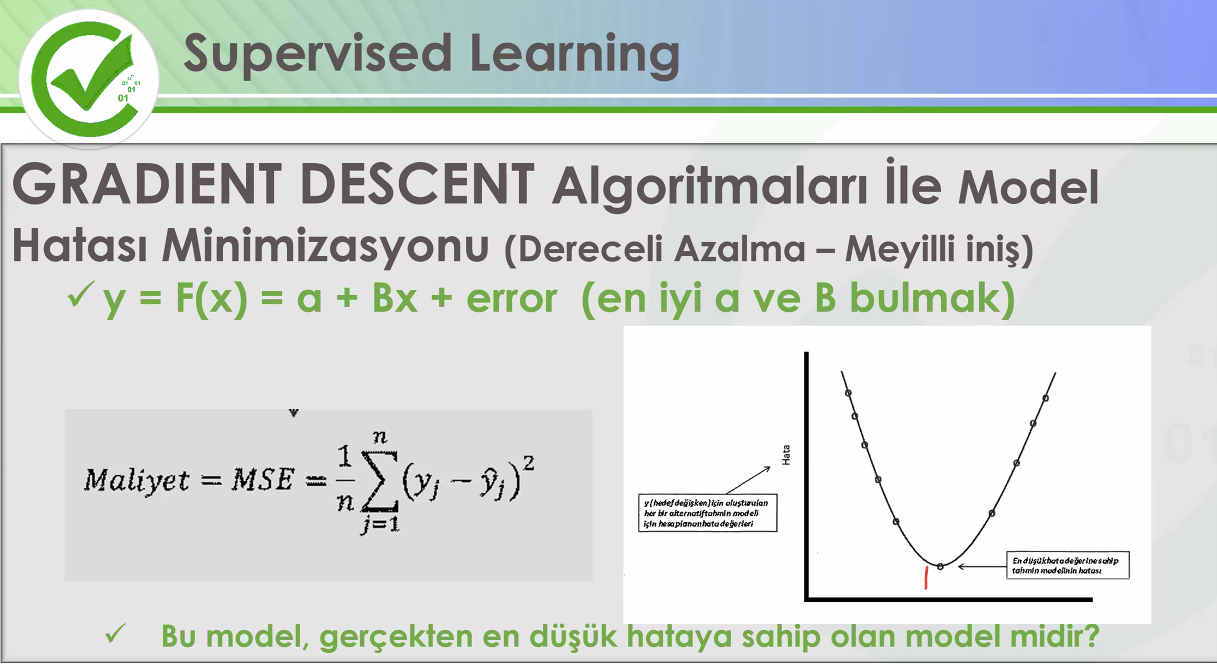
Olabildigince dusuk derece tercih ederiz.

**Net m2** feature, **X2** diyelim. Burda RMSE acisindan degerlendirince 4.yu secmem mantikli

**Dezavantajlari:**

Polynom reg-->outlier’a karsi cok hassas oluyorlar.





Soyut bir konu

Bir optimizasyon algoritmasi calistiriliyor.

En iyileme-->best hale getirmek

Arka planda ne oluyor ne calisiyor da optimize hale geliyor?

Arka planda calisan mantik--? gradient-descent algoritmasi calisiyor

Target variable’i, kayip bilgi cinsinden minimum degerini bulmak

Amacimiz en iyi a ve b’yi bulmaya calisiyorduk.

Regresyon algoritmasi bana oyle bir a ve b degeri vermeli ki, modelimin sonucunda elde ettigim hatalar minimum olsun.

Msq-->olayin maliyeti. Error metriclerinden biri

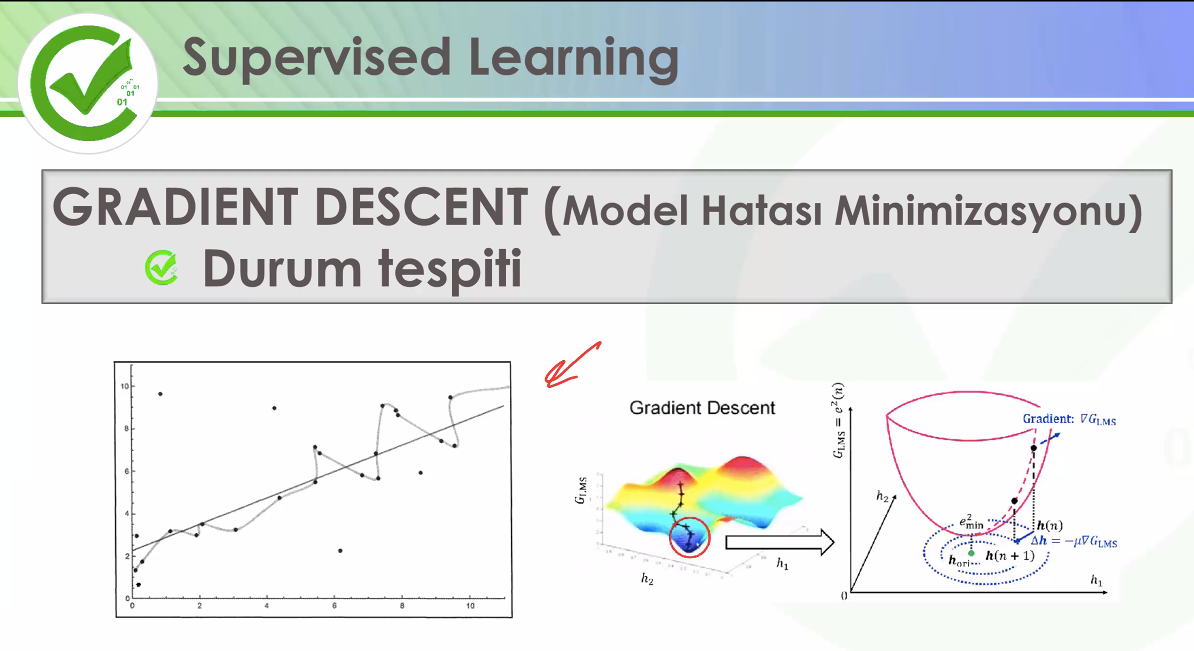
Tahmin modelinin maliyet fonksiyonudur.

Derdimiz de bilgi kaybini azaltmak

Bilgi kaybina bagli olarak ne kadar risk aldigimizi ne kadar maliyete katlandigimizi gosteri mse. Grafik de mse maliyet fonksiyonunun grafigi.

Derdimiz min’u yakalamaktir.

En dusuk hataya sahip olan modeli nasil bulacagiz?



X degiskeni, y’yi en iyi sekilde tahmin edecek b parametresini bulmaya calisiyor.

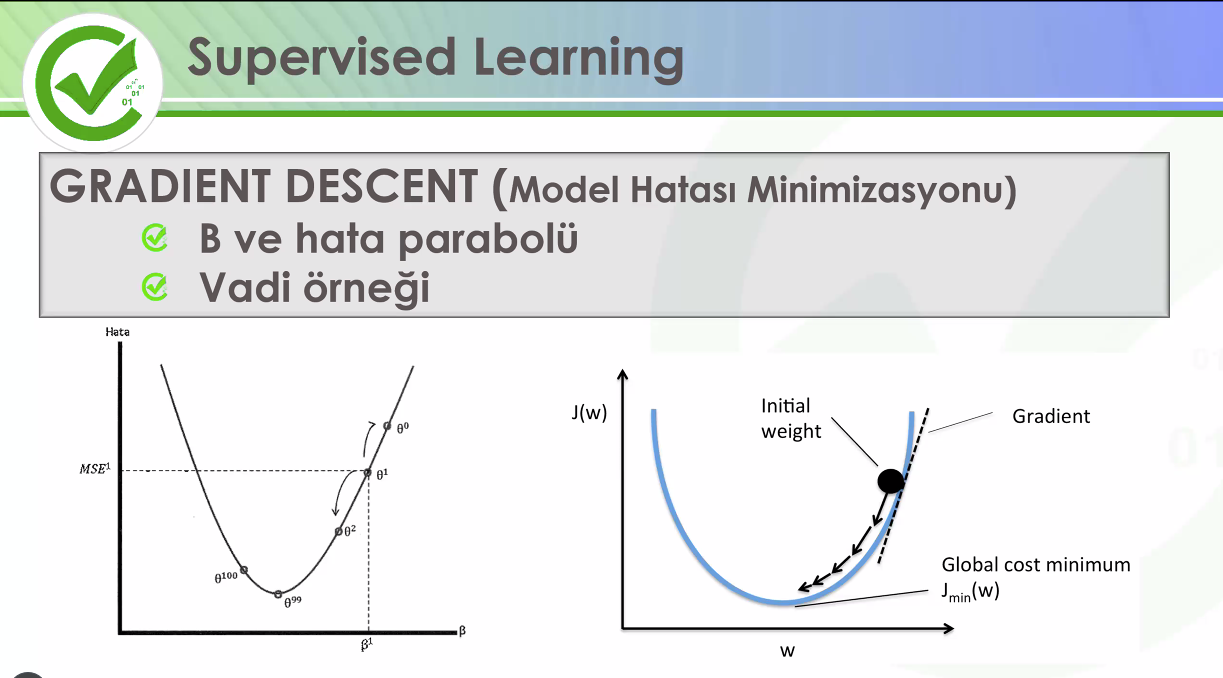
Parabol uzerindeki herbir nokta alternatif modellerin hatalarini gosterir. Min nokta da en dusuk hatayi gosterir.

Gradient descent algoritmasi daha cok b’yi tespit etmeye odaklanir ve ondan yola cikarak bulmaya calisir.

Amacimiz en dusuk noktayi bulmak.

Global min ve lokal min var. Lokal min-->tum vadideki min noktalardan biridir.

Global min--> tum vadideki min noktalardan en min olanidir.



Iterative approach denilen tekrar tekrar yeniden yaptigimizda, tekrar yukari cikmaya baslamadan onceki nokta aslinda minimumdur.

**Gradient-descent** icin en cok verilen ornek vadi ornegidir.

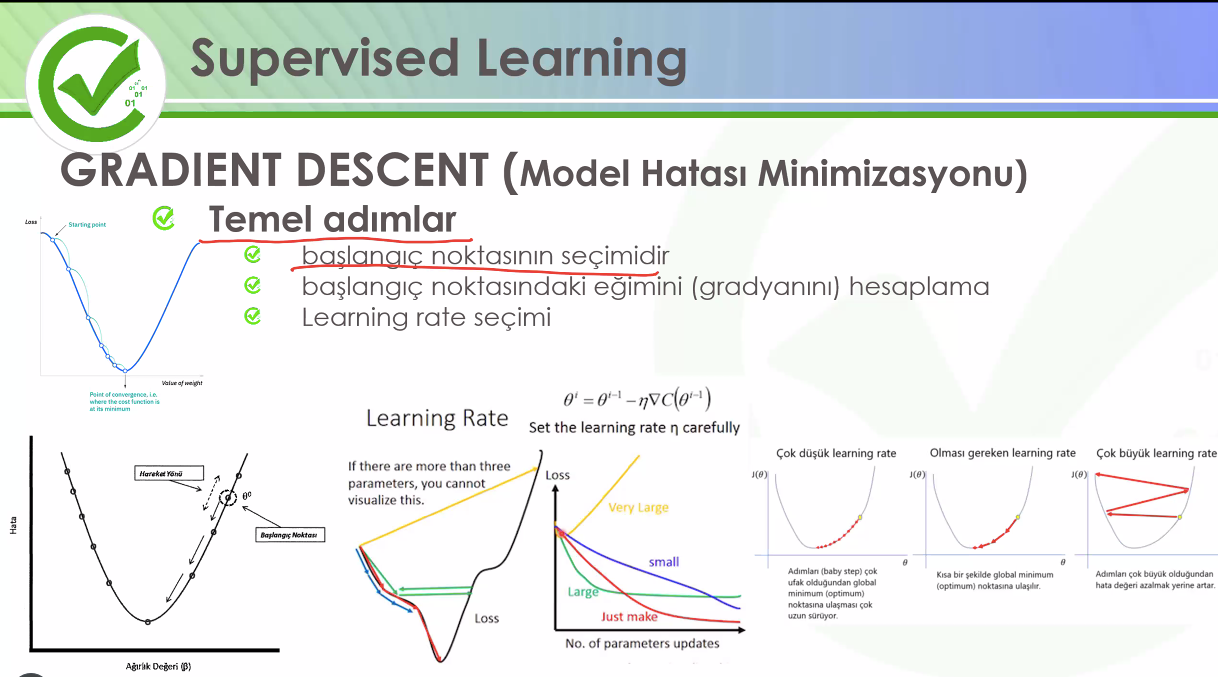
Gozleri bagli bir sekilde tepelerin birinden yamacin bir yerine birakilan bir kisiyi dusunun.

Gozleri bagli sekilde etrafinda kucuk kucuk adimlar atar. Eger attigi adim yukari dogru bir meyil olusturuyorsa, oraya gitmez geri doner. Asagi dogru bir adim ise, bir adim daha atar oraya dogru devam eder. asagi dogru inmeye baslar. Her seferinde bu islemi devam ettirir. Devam da etse, geri de donse yokus yukari ise minimuma geldigini bilir.

Gradient-->meyil, ML acisindan ise maliyet func.inin, herbir hata icin egimidir.

Arka Bunu turev alarak yapar. b degerine gore.

Algoritma gradient’in 0 oldugu yani min oldugu noktaya kadar tekrarlamis olur.



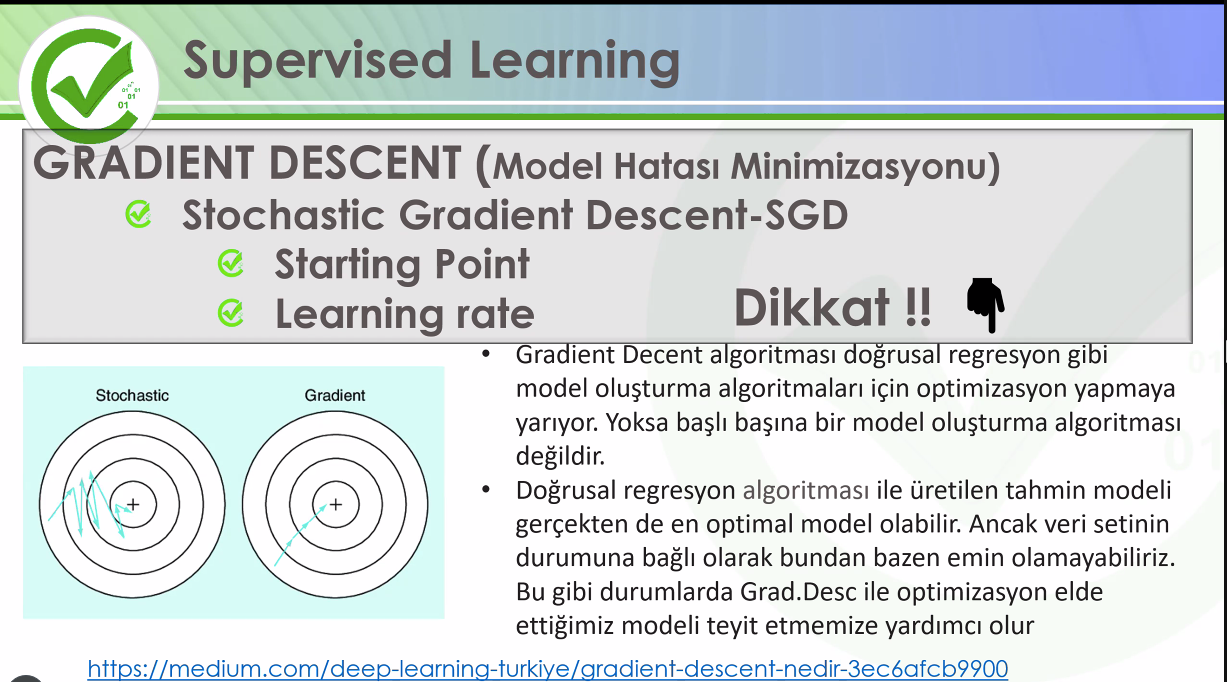
1.

1. Maliyet fonk ilk adimini hangi yonde atacak buna bakar. B degerinin turevini hesaplar
2. Learning rate--> otimum noktaya dogru hangi adimlarla ilerlenecek. Adim boyu

Baby-step olursa global min noktasina ulasmasi uzun surer. Cok buyuk learning rate olursa da, global noktayi yakalayamadan karsi vadiye gecmis olur.

1 milyon gozlem noktasi var diyelim-->herbiri icin yaparsa cok zaman kaybi olur. Algoritmada her bin ya da 10bin icin learning rate belirlemesini isteyebiliriz.

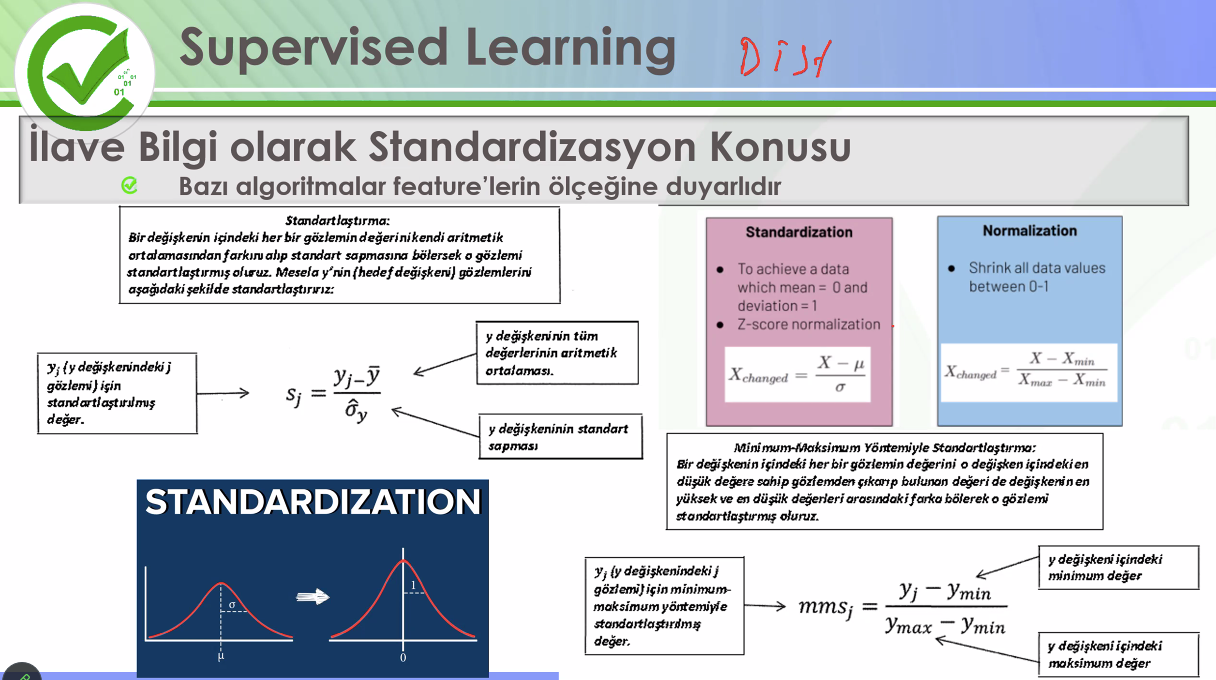
Vadi maliyet fonk.nunu temsil ediyor. Min noktasi da hatanin en az oldugu nokta



ML performansini iyilestirmek icin kullaniliyor.

Bu surec modelin ogrenmesi olarak adlandirilir.

Modelin tahminlerini gercekler degerlere daha yakin bi yere getirmeye calisir.



Bazilari Distance-base-->mesafeye dayali calisir algoritma.

Classification’da onemli olan konulardan biri distance meselesidir.

Feature scaling--> butun ozellikleri ayni olcuye getirmek

Standardization ya da normalization kullanilir.

Aritmetik ortalamasindan farkini alip, standard sapmaya boluyoruz. Ortalamasinin 0, std sapmanin1 olmasini saglar. Outlier’a karsi direnclidir.

**Min max standardlastirmasi-**-> **min max scaling**--> bir yontemidir. Min max scale

En dusuk degerini cikari, en yuksek degerle en dusuk deger arasindaki farka bolerek yapiyorsun.

Feature scaling--> train datada yapilir.

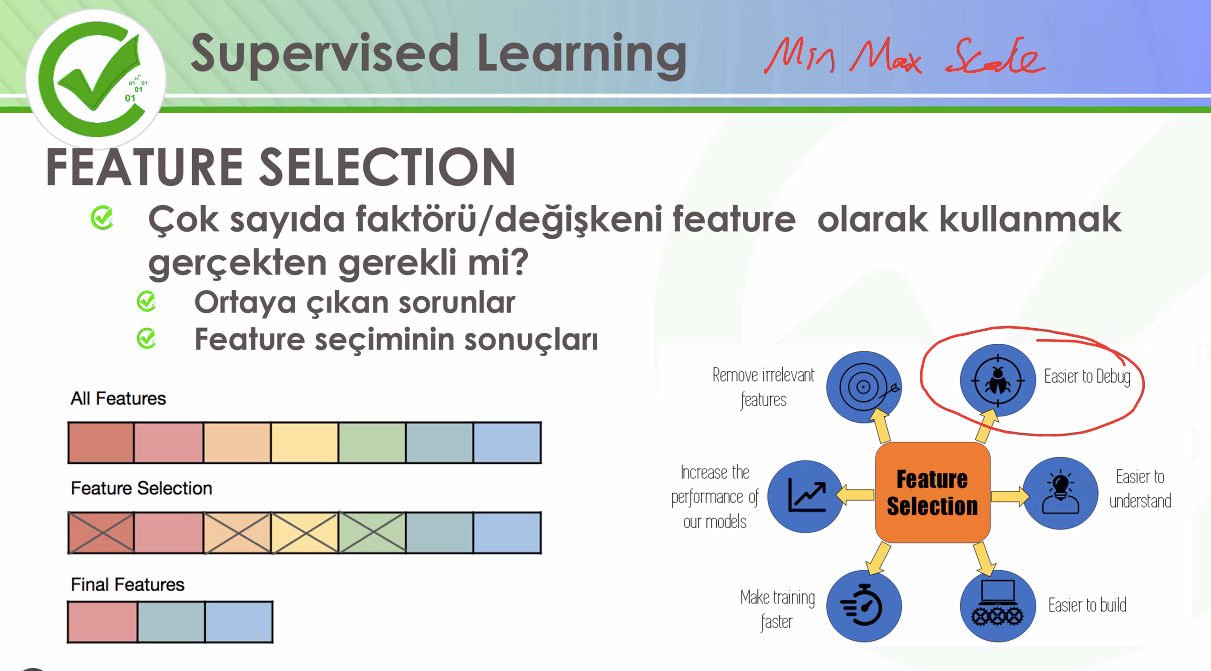
Dagilimin min 0, max 1 gibi davraniyor normalizasyon yaparken. Outlier etkisini biraz azaltabiliyor.

Standard scaler,

Min max scaler olarak gorecegiz notebook’larda

Hangi yontemi kullanacagiz?

-her ikisini de dene, hangisi iyiyse onu kullan.-literaturde net degil aslinda.

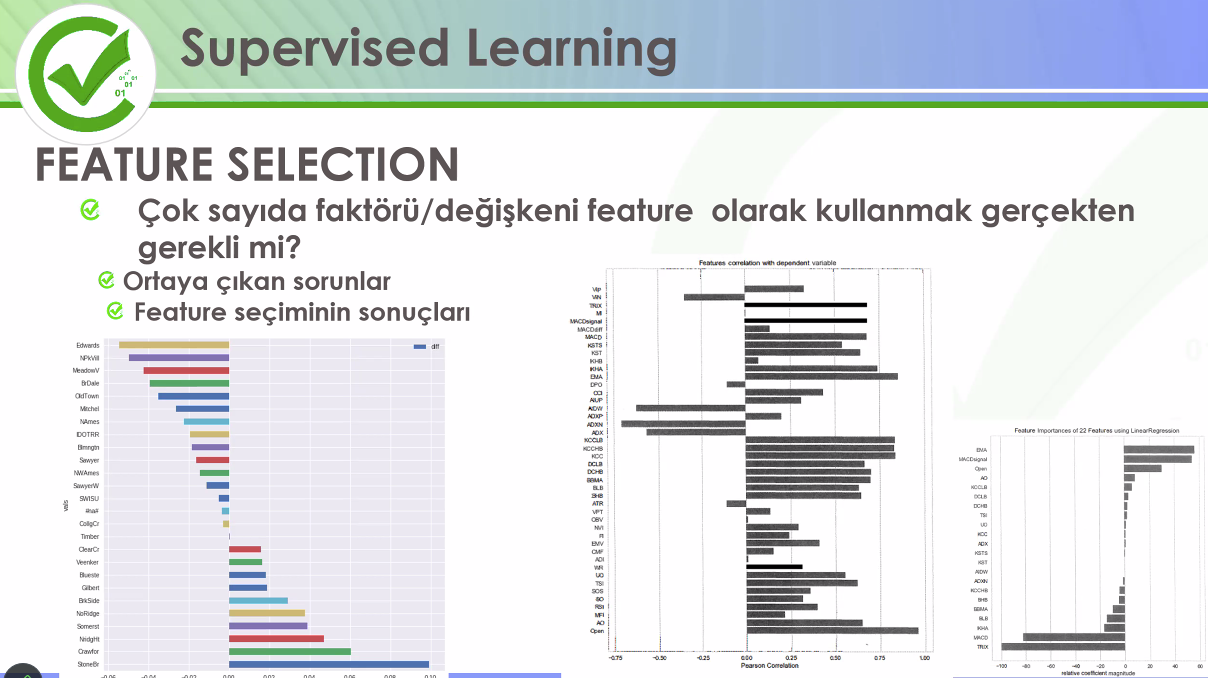


Big data ile ugrasmak zor-->feature selection onemli-->

Feature sayisi cok artarsa model karmasiklasir. Overfitting riski olusur.

Feature sayisi arttikca anlamsiz datalar da modele girmis olabilir.

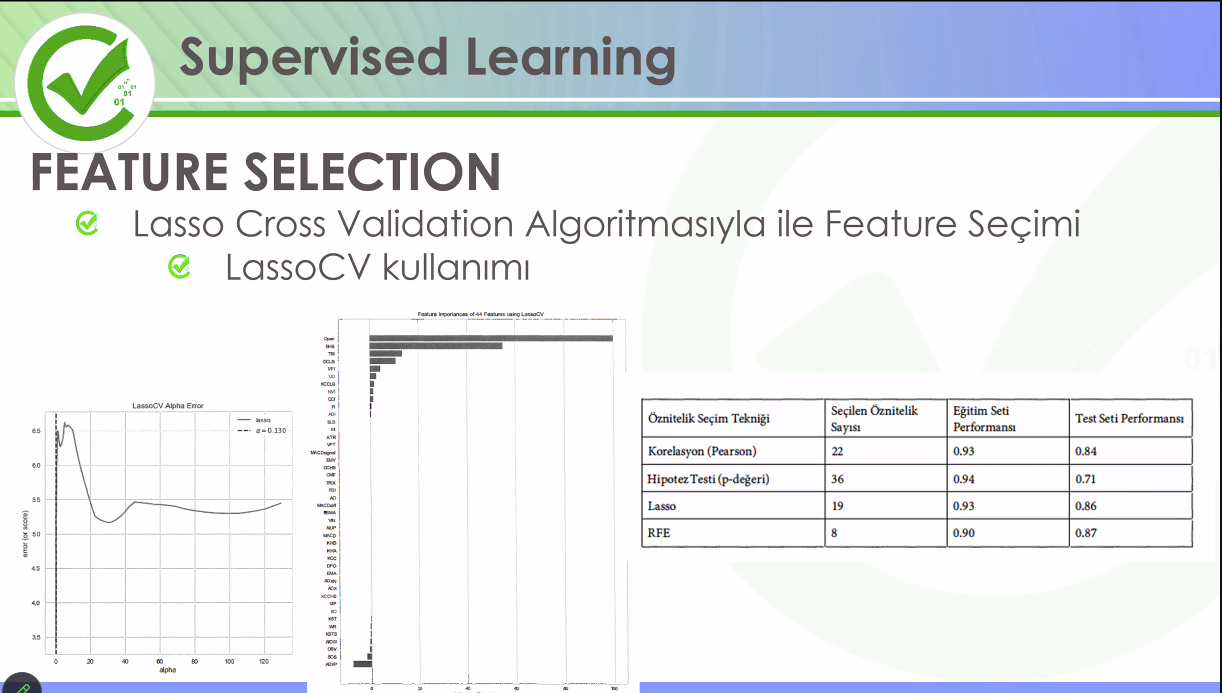
Zaman artar.



**Feature selection methodlari:**

**En kolayi:**

1. Lasso



44 feature var. Herbirini almak mi gerekiyor?

Korelasyon ile 44 taneden 22 tanesi secilmis.

Hipotez testi ile 36’ya dusmus.

Lasso’ya gelince 44’ten 19’a dusmus.

Cok fazla feature ile calismak dogru degil

Overfitting sorunu var. Zaman sorunu var. Modeli anlamsizlastirabilir.

Bilgi kaybi olabilir ama getirisi daha fazla oldugu icin tercihim feature selection yapmak.

**ML Notebook**

def poly(d):

train\_rmse\_errors = []

test\_rmse\_errors = []

number\_of\_features = []

for i in range(1, d):

polynomial\_converter = PolynomialFeatures(degree = i, include\_bias =False)

poly\_features = polynomial\_converter.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(poly\_features, y, test\_size=0.3, random\_state=101)

model = LinearRegression(fit\_intercept=True)

model.fit(X\_train, y\_train)

train\_pred = model.predict(X\_train)

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

train\_RMSE = np.sqrt(mean\_squared\_error(y\_train,train\_pred))

test\_RMSE = np.sqrt(mean\_squared\_error(y\_test,test\_pred))

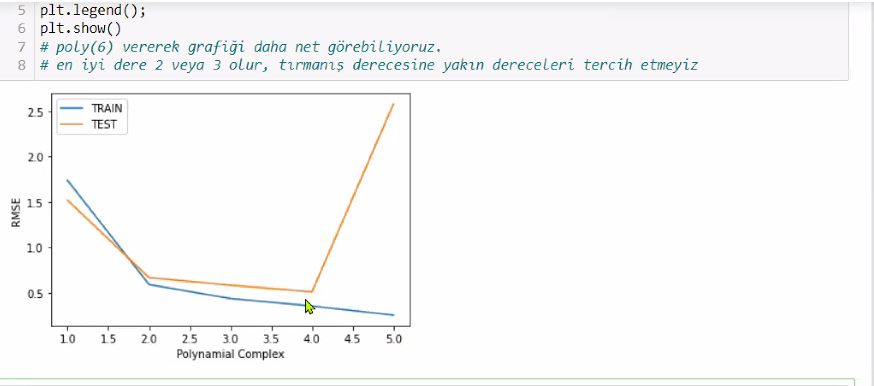
train\_rmse\_errors.append(train\_RMSE)

test\_rmse\_errors.append(test\_RMSE)

number\_of\_features.append(poly\_features.shape[1])

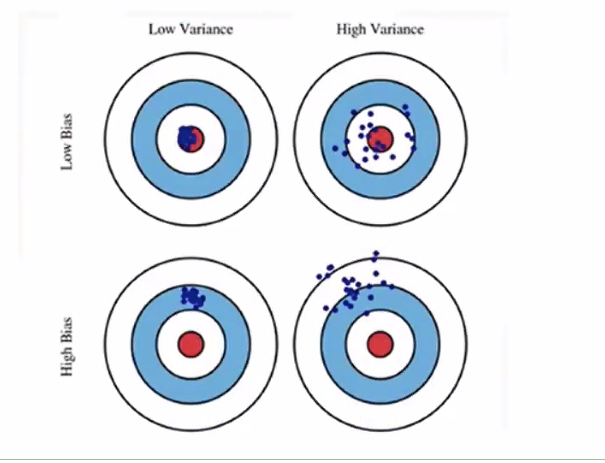
return pd.DataFrame({"train\_rmse\_errors": train\_rmse\_errors, "test\_rmse\_errors":test\_rmse\_errors, "number of features":number\_of\_features},

index=range(1,d))



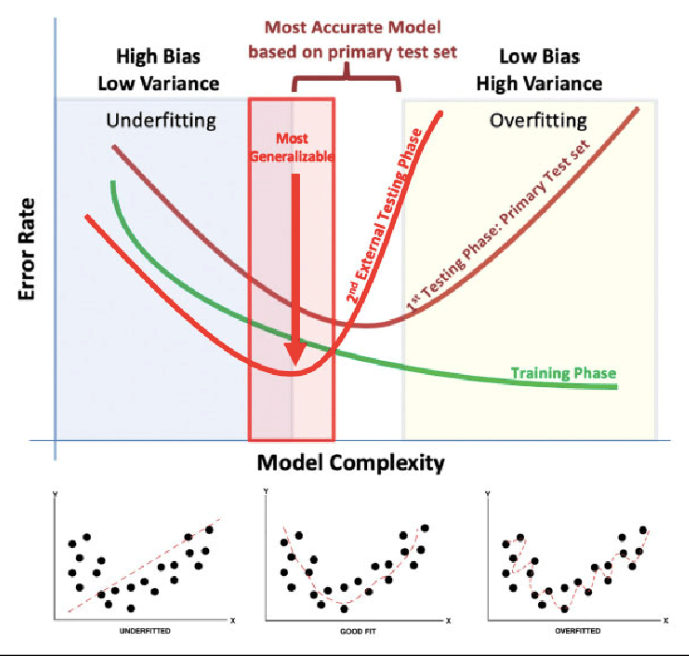
4 patlama noktasi--> 4’ten sonra yukselis basliyor.

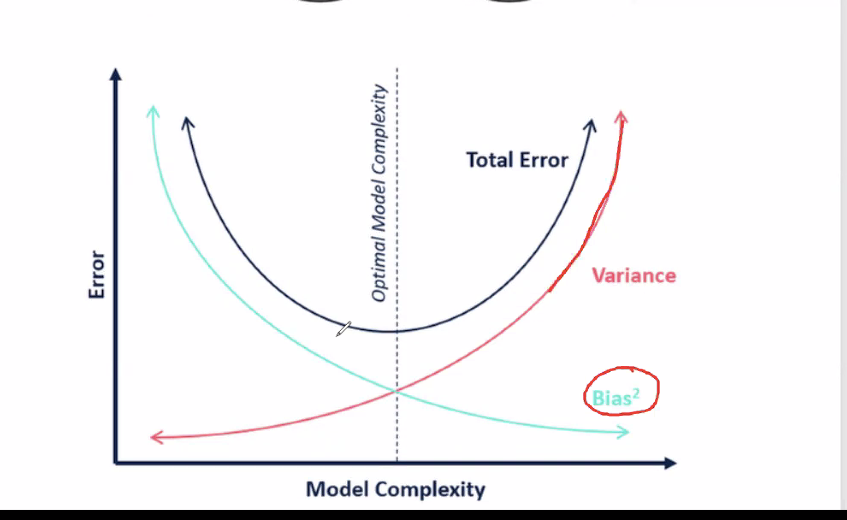
O yuzden 4’ten az geride durmali



Variance-bias trade-off

Bias-->





**Bias yuksek-**->Model henuz ogrenemedi, egitime devam etmesi lazim--yani underfitting

Ordan daha iyi skor varsa o zaman underfitting var demektir.

Modelin karmasikligi yeterince yuksek degil-->genelleme yapmak zor

Hem egitim hem test datasinda dusuk performans gosterir.

O zaman modelin kompleksligini artiracagiz, deneyecegiz falan

**Variance yuksek**--> egitim verilerindeki kucuk degisikliklere duyarliligi fazla

Yuksek R2 skoru dusuk bias demektir. R2-->1 ciktiysa suphelenmem lazim.

R2-->cok yuksekse overfitting’den suphelenmem gerek.

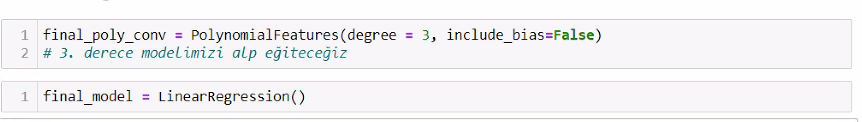
Modelimde bias-variance arasinda denge kurmam lazim.

low bias high varyans = owerfitting

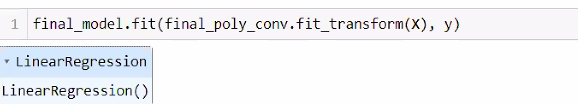
Hiht bias low varyans = underfiting

ideal olan nedir low bias low varyans

**Finalizing Model Choice**

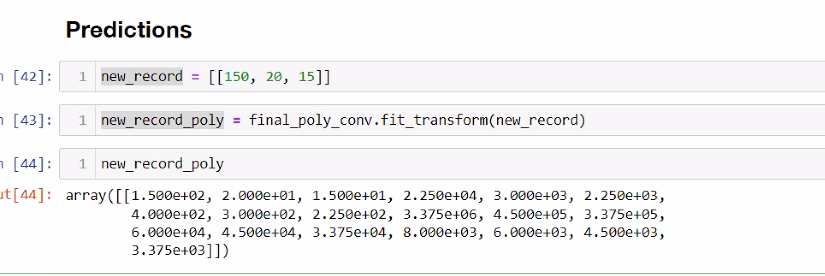


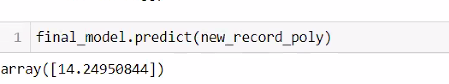
Final modelimi kuruyorum.



Degree 3’te olustu bu.

**Predictions**





new\_record degerlerini girersem, 3 feature icin 3 deger--> sales’im 14.24 olur diyor. final\_model.

**Overfitting**

Overfitting’i 5. degree olarak belirlemistim.

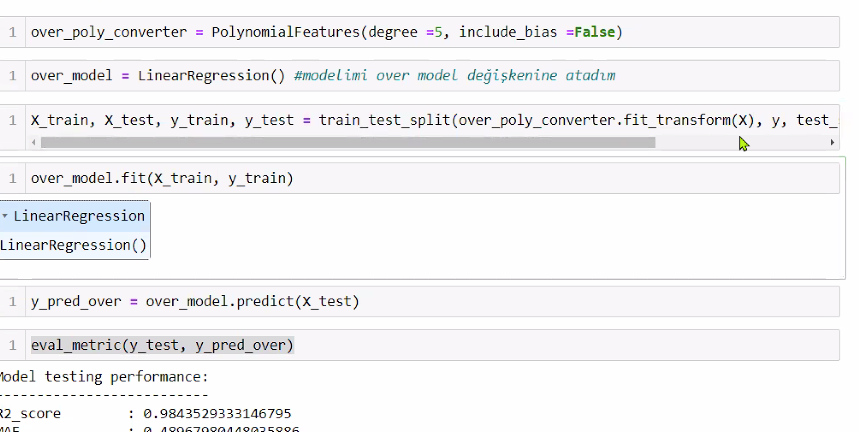
Over\_poly dedim bu 5. derece olan dataya. Overfittingi gostermek icin olusturdum bunu.

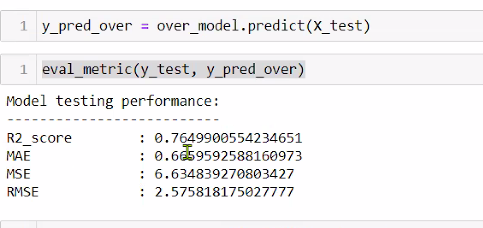


**over\_poly\_converter = PolynomialFeatures(degree =5, include\_bias =False)**

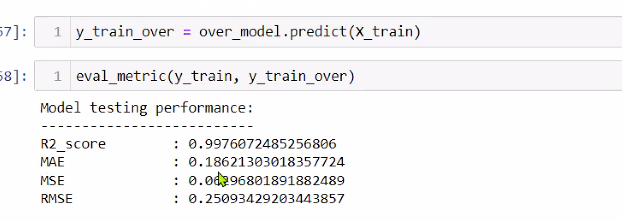
**over\_model = LinearRegression() #modelimi over model değişkenine atadım**

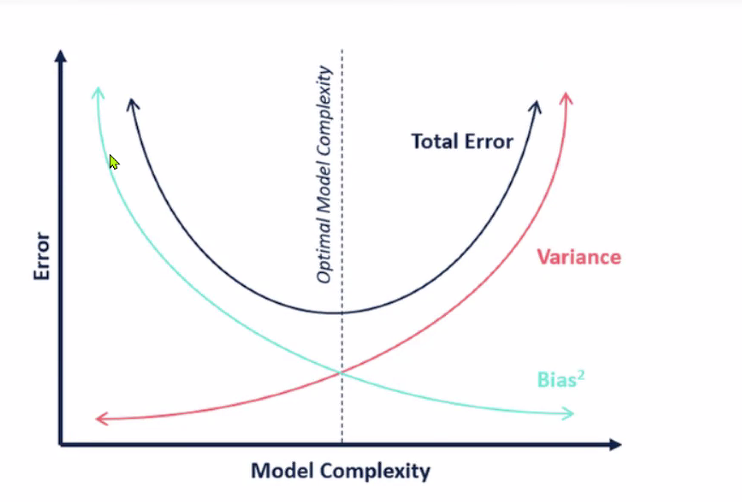
**over\_model.fit(X\_train, y\_train)**





Skorlari degerlendirirsek-->y-predin y-testten farkinin olculeri olan error-metricler ile, train datalarinin error-metriclerini kiyasliyorum. 5. dereceye cikarttigimda, overfitting oldugu anlasiliyor. R2 skorum trainde 0.99 iken, 0.76’ya dusuyor. Ilk baktigim aslinda RMSE. Cunku RMSE bize overfittingi cok iyi verir. Cunku hatalari cezalandiran metric RMSE. Daha sensitive metric RMSE.





Train datasinda kucuk degisikliklere duyarli olmasi aslinda istenen bir sey degil. Herseye hakim cok iyi ogrenmis. Bunu da istemeyiz. O yuzden bias-variance’in optimumunu bulmak gerek.

Ridge ve Lasso ile mucadele edecegiz mesela overfitting ile. Su an manuel calisiyoruz. Ama ileride modelin icinde gomulu parametreler olarak gelecek bize overfittingi asma yontemleri logistic regression’da. Model diyecek ki bize lasso’yu sec.

Mesela dereceler verecegim 3,5,2 diye, soracagim hangisini seceyim diye. Tum sorularima cvp alinca, modelimi bunlar uzerinden kuracagim zaten artik. 10 tane hyperparametre veriyor mesela sana. Sen bunu deniyorsun, olmazsa baska kombinasyonlar denetiyorsun.

OZET:

|  |
| --- |
| 1. Patlama oldugu noktadana bir geriye git. Guvenli bolge o bolge. 2. Low bias high variance olunca-->overfitting 3. Bias yuksek oldugu noktalar--> underfitting |