

## Machine Learning

Bilgisayarların insanlara benzer şekilde öğrenmesini sağlamak maksadıyla çeşitli algoritma ve tekniklerin geliştirilmesi için çalışılan bilimsel çalışma alanıdır.

Bağımlı değişkenimiz sayısal olan problemlere regresyon problemi denir

Bağımlı değişkenimiz kategorik olan problemlere sınıflandırma problemi denir

### Değişken Türleri

- Sayısal Değişkenler
- Kategorik Değişkenler(Nominal,Ordinal)  
Nominal : Kadın - Erkek, Sınıflar arasında fark yoktur  
Ordinal : Eğitim Durumu , Sınıflar arası fark vardır, sıralıdır.
- Bağımlı Değişken : İlgilendiğimiz problemdeki hedefimizdir (target,output,response,dependent diye isimlendirebilir)
- Bağımsız Değişken : İlgilendiğimiz problemde target ' a etki ettiğini varsaydığımız değişkendir . (feature olarak da isimlendirilir)

### Öğrenme Türleri

3 farklı öğrenme türümüz vardır.

- Denetimli Öğrenme (Supervised) -> Eğer veri setinde labellarımız yer alıyorsa bu denetimli öğrenmedir. Bağımlı değişken varsa denetimli öğrenmedir
- Denetimsiz Öğrenme (Unsupervised) -> Eğer veri setinde labellarımız yok ise yani bağımlı(hedef) değişkenimiz yoksa gözetimsiz denetimsiz öğrenmedir
- Pekiştirmeli Öğrenme (Reinforcement) -> Deneme yanılma ile öğrenme durumuna denilebilir

### Problem Türleri

Regresyon problem mi ? Sınıflandırma Problemi mi ?

- Eğer bağımlı değişken sayısal ise regresyon problemidir
- Eğer bağımlı değişken kategorik ise sınıflandırma problemidir

### Model Başarı Değerlendirme

Tahminlerim ne kadar başarılı ? Kurmuş olduğumuz modeller tahminlerde bulunduğunda bu tahminlerde belirli sapmalar bekleriz , (hatalı tahminlere artık denir)

- Regresyon Modellerinde Başarı Değerlendirme;

MSE = Başarı değerlendirme ölçüsüdür. Kurmuş olduğumuz model neticesindeki gerçek değerler ile modelin tahmin ettiği değerler arasındaki farkları değerlendirme imkanı sunar .Ortalama hata metriğidir

RMSE = Ortalama hata metriğidir.

MAE =

- Sınıflandırma Modellerinde Başarı Değerlendirme;

Accuracy (Doğru Sınıflandırma Oranı) = Doğru Sınıflandırma Sayısı / Toplam sınıflandırılan Gözlem Sayısı

-- MSE ne kadar düşük olursa o kadar iyidir

-- Accuracy ne kadar fazla ise o kadar iyidir

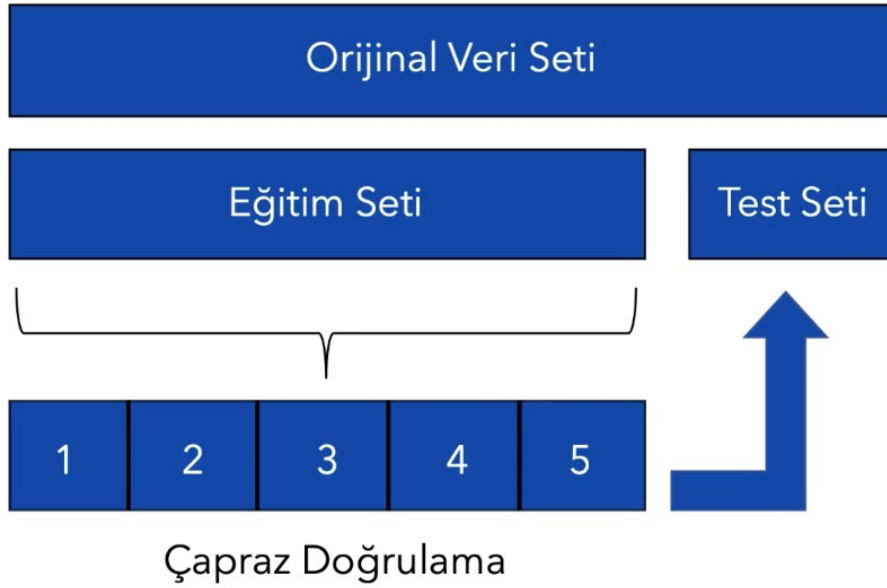
### *Model Doğrulama Yöntemleri*

- Holdout Yöntemi (Sınama Seti Yöntemi):

Orjinal veri setini Eğitim Seti ve Test Seti olarak ikiye bölünür. Eğitim Seti üzerinde modelleme işlemi(train,öğrenme) gerçekleşir daha sonra o modele test setinden sorular sorularak test edilir . Başarı bu şekilde değerlendirilir.

- K-Katlı Çapraz Doğrulama Yöntemi

Çapraz doğrulama tekniği günümüzde yoğun olarak kestirimci makine öğrenimi modellerinin test edilmesi için kullanılıyor. Bir veri setindeki biri hariç tüm alt setlerde eğitilen makine öğrenimi modelinin doğruluğu, öğrendiklerinden yola çıkarak geriye kalan set hakkında tahminlerde bulunmasıyla ölçülüyor. Veri bilimcileri, hangi modelin en iyi tahminleri gerçekleştirebildiğini sağlıklı şekilde saptamak için farklı veri setleri kullanarak, birden çok çapraz doğrulama uyguluyor. Çapraz doğrulama, özellikle algoritmaların eğitimi için kısıtlı veri bulunduğu faydalı bir teknik olarak öne çıkıyor



#### #### Yanlılık (bias)- Varyans Değiş Tokuşu

Model Kurmak : Bağımlı ve bağımsız değişken arasındaki ilişkiyi , ilişkinin özütünü çıkarmak demektir. Bu ilişkiyi öğrenmek demektir.

- Aşırı öğrenme(Overfitting) , Yüksek varyans : Modelin veriyi öğrenmesidir. Modelin veriyi ezberlemesidir. Bizim amacımız; veriyi öğrenmek değil, veri içerisindeki örüntüyü öğrenmektir. Veriyi öğrenmesini değil, verinin yapısını öğrenmesini bekleriz.

(Yüksek Varyans yani değişkenlik fazla )

- Az Öğrenme (Underfitting) , Yüksek yanlılık : Modelin veriyi öğrenememe problemidir.

( Yüksek yanlılık yani bazı gözlemlere daha yanlı , daha yakın demek)

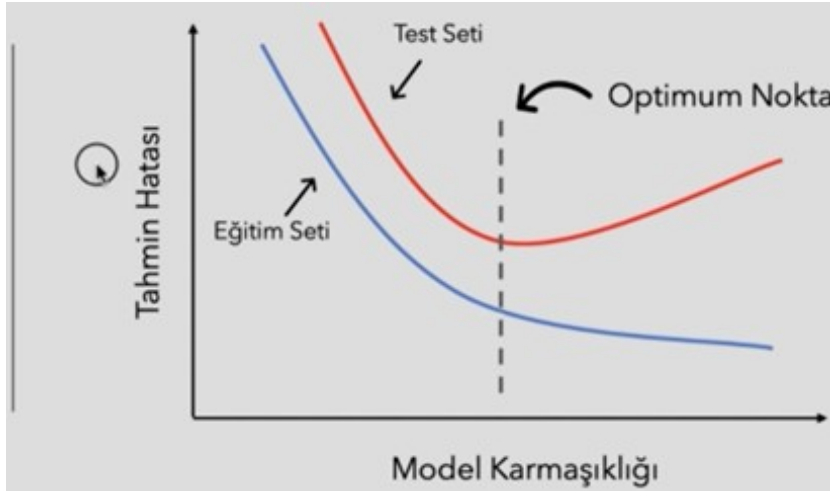
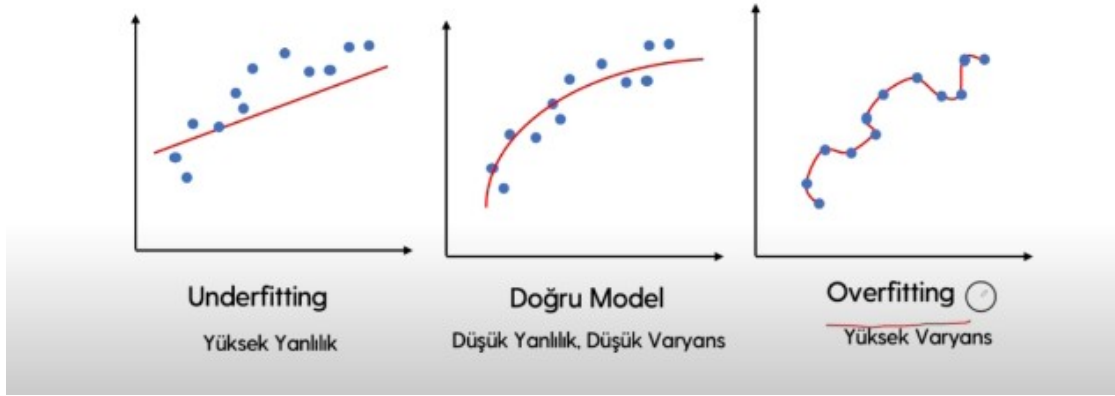
- Doğru Model , Düşük Yanlılık, Düşük Varyans : Veri setinin taşıdığı örüntünün(ilişkinin) temsil edilmesidir.

--- Aşırı öğrenmeye düştüğünüzü nasıl anlarsınız ? Aşırı Öğrenme nasıl tespit edilir? Aşırı öğrenme sorunu, eğitim seti ve test setinin Model Karmaşıklığı ve Tahmin Hatası çerçevesinde birlikte değerlendirilmesi ile tespit edilebilir. Eğitim seti ve test setindeki hata değişimleri incelenir , bu iki hatanın birbirinden ayrılmaya başladığı nokta itibari ile aşırı öğrenme başlamıştır denir .Aşırı öğrenmeyi tespit ettiğimiz noktada model karmaşıklığını , eğitim süresini iterasyonu durdurursak bu problemin önüne geçeriz.

--- Model Karmaşıklığı nedir ? Modelin hassaslaştırılmasıdır. Daha detaylı tahmin yapılabilmesi için özelliklerinin kuvvetlendirilmesidir. Ağaç yöntemlerinde ağaçların dallandırılmasıdır. Karmaşıklaştırma bir süreye kadar kötü değildir ama

bir noktadan sonra ezberlenmeye gidildiği için bir noktadan sonra karmaşılaştırılma yapılmamalıdır.

--- Aşırı Öğrenme nasıl çözülür? Bunun birçok yanıtı olabilir. Veri setinin boyutu artırılabilir. Feature selection yapılabilir. En nihayetinde, eğitim seti ile test setinin hatalarının birbirlerinden ayrıldığı nokta seçilir optimum nokta olarak. Bu noktada durulduğunda aşırı öğrenmenin önüne geçilmiş olunur.



## - Doğrusal Regresyon

Amaç, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi doğrusal olarak modellemektir.

- $y_i = b + w x_i$  (tek değişkenli)

$y_i$  = bağımlı değişken yani tahminimiz

$x_i$  = bağımsız değişken(ler)

- $y_i = b + w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + w_p x_p$  (çok değişkenli)

Bir model kurdum, makine öğrenmesi yöntemi kullandım , öğrendim dediğimiz şeyler buradaki ağırlıklardır. (Yani w)

b = beta, bias, sabit şeklinde ifade edilir

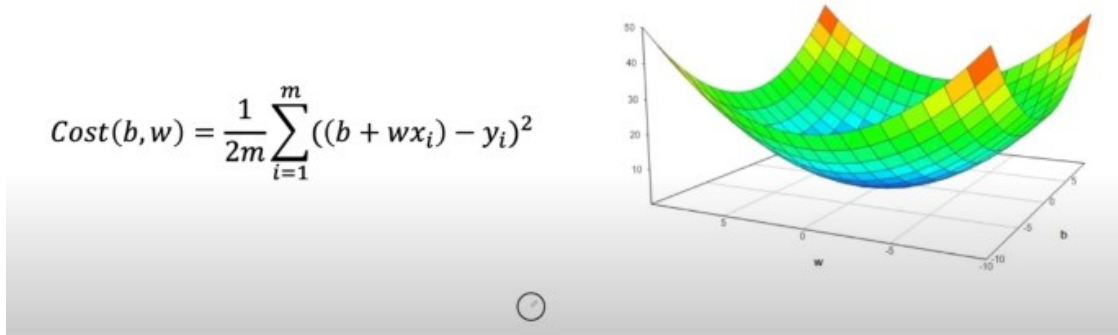
w = ağırlık, katsayı şeklinde ifade edilir

w ifadeleri ilgili değişkenlerin etkilerini ifade etmek için kullanılır

### Parametrelerin Tahmin Edilmesi (Ağırlıkların Bulunması)

Ağırlıklara göre değişkenlerin etkileri biçimlendirilebiliyor dedik. Bu ağırlıkları nasıl bulacağız? Neye göre bulunacak?

- Gerçek değerler ile tahmin edilen değerler arasındaki farkların karelerinin toplamını/ortalamasını minimum yapabilecek b ve w değerlerini bularak.
- Buradaki sabit ve ağırlık ifadeleri öyle bir şekilde bulunmalı ki en uygun değerlere sahip olsun



Bu ağırlıkları bulmanın 2 yolu vardır.

#### 1- Analitik Çözüm : Analitik Çözüm : Normal Denklemler Yöntemi(En Küçük Kareler Yöntemi)

Bu yöntem analitik bir çözümdür. Matrix formunda bir çözümdür. Türeve dayalı bir çözümdür. Bu yöntem kullanıldığında ortaya çıkacak olan doğrusal formülasyon üzerinde analitik yorumlar yapılabilir, neden-sonuç yorumları yapılabilir.

-- Eğer elimizde tek bir değişken varsa yani bir bağımlı bir bağımsız değişken varsa; iki parametreye göre ayrı ayrı kısmi türevler alınıp, 0' a eşitlenir ve bulunması gereken çözülmesi gereken matematiksel ifade burası olur. -- Eğer çok değişken varsa; benzer şekilde ilgili ağırlıklara göre kısmi türevler alınıp, 0'a eşitlendikten sonra ilgili ağırlıklar yalnız bırakıldığında ortaya bir matrix çözümü çıkacaktır. Bu matrix çözüldüğünde tahmin edilen Betalar bulunacaktır. yani  $w_1, w_2, w_3$  ifadeleri bulunacaktır.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

- NEDEN Normal Denklemler Yöntemi değil de , Gradiend Descent yöntemi kullanalım ki ?

!!! Buradaki tersini alma ifadesi (-1 ifadesi) gözlem sayısı ve değişken sayısı çok fazla arttığında bu matrixin tersini alma işlemi zorlaşmakta ve gradiendt descend gibi optimizasyona dayalı problemlerinin ihtiyacı artar.

Çünkü çoklu lineer regresyon probleminde , normal denklemler yönteminde , en küçük kareler yöntemindeki final matrix çözümünün tersini alma işlemi gözlem sayısı ve değişken sayısı çok fazla olduğunda zorlaşmaktadır.

## 2 - Optimizasyon Çözüm : Gradient Descent

Nerden geldik buraya? Hatamızın minimum olmasını istiyoruz. Bununda ağırlıklara bağlı olduğunu gördük. Dolayısıyla bu ağırlıkları bulmamız lazım. Bu ağırlıkları nasıl buluruz ki diye sorduk. 2 yöntemle. 1- Normal Değerler Yöntemi , 2 - Gradient Descent yöntemi

$$b_0 = b = \theta_{10}$$

$$b_1 = w$$

, Optimizasyon Çözümü: Gradient Descent

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

Doğrusal Regresyon için Gradient Descent ( Gradient Descent for Linear Regression )

##### konu ile ilgili tanımlar gelecek

## - Lojistik Regresyon

Amaç, sınıflandırma problemi için bağımlı ve bağımsız değişkenler arasındaki ilişkiyi doğrusal olarak modellemektir.

$$\hat{y}_i = \frac{1}{1 + e^{-(z)}}$$

$$z = b + w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + w_p x_p$$

Doğrusal regresyondan ne farkımız var?

- birinci işlem regresyonun tamamen aynısı fakat buradan çıkan sonucu (z değerini bulan fonksiyon) sigmoid fonksiyonunda z olarak gördüğümüz yere yazıcaz.

Peki neden ? Normalde bağımlı değişkenin 1 ve 0 lardan oluştuğunu düşünürsek, bağımsız değişkenlerinde çeşitli değerlerden oluştuğunu düşünelim. Normalde bir regresyon modeli kurdugumuzda bağımlı değişken 1 yada 0 olduğunda bu modeli kurabiliriz bir problem

olmaz. fakat bu modeli kurdugumuzda tahmin sonuclarını elde etmek istedigimizde deęerler 1 'den büyük yada 0 'dan küçük çıkabilir. Eksi deęerler çıkabilir. Dolayısıyla öyle bir işlem yapmalıyız ki tahmin edilen nihai sonuc 0 ile 1 arasında yer alsın ve bunu belirli bir eşik deęerine göre 0 yada 1 olarak deęerlendirebilelim. İşte bu amacımızı yerine getirebilmek için böyle bir sigmoid fonksiyonu ile dönüştürme işlemi gerçekleşiyor.

Bu doğrusal ifadeden gelen deęer(z) sigmoid fonksiyonunda yerine yazıldığında bize 0 ile 1 arasında olacak sekilde bir olasılık deęeri üretilir. Bu olasılık deęerine göre 1 yada 0 sınıfına ataması yapılır.

Temel mantık doğrusal regresyona benzer. Sadece farklı olarak sondaki bir deęer deęiştiriliyor. Bir dönüştürme işlemine tabi tutuluyor. \*\*Peki nasıl gerçekleştirilir bu iş ?

- Gerçek deęerler ile tahmin edilen deęerler arasındaki farklara ilişkin log loss deęerini minimum yapabilecek ağırlıkları elde edilir.

$$\hat{y}_i = \frac{1}{1 + e^{-(z)}}$$

$$z = b + w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_px_p$$

$$Log\ Loss = \frac{1}{m} \left( \sum_{i=1}^m -y_i * \log(p(\hat{y}_i)) - (1 - y_i) \log(1 - p(\hat{y}_i)) \right)$$

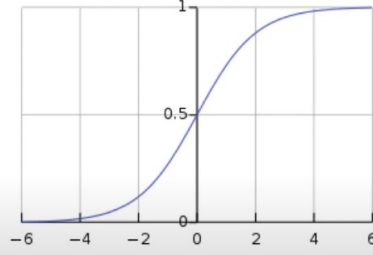
Daha önce MSE adında bir fonksiyonumuz vardı. Bu minimum olacak şekilde ağırlıkları(w) buluyorduk . Yine benzer amacımız var. Log loss fonksiyonu minimum olacak sekilde ağırlıkları bulmaya çalışıyoruz.

Özetle regresyona benzer bir yapımız var. Farklı olarak buradan türeyecek deęeri bir sigmoid fonksiyondan geçirip çeşitli olasılık deęerleri elde ediyoruz. Ve bunun üzerinden yorumlamaları sürdürüyoruz.



**\*\*Gelelim sigmoid fonk nedir ve bu dönüştürme işlemleri nasıl gerçekleşiyor?**

$$\hat{y}_i = P(Y = 1 | X = x) \quad \frac{1}{1 + e^{-(z)}}$$



$$z = b + w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_px_p$$

buradaki x ekseninin ifade ettiği bu doğrusal formdan(z) gelecek olan skorlardır, değerlerdir. Belirli bir yöntem ile w ları bulduğumuzu varsayacak olursak; bu ağırlıklar yerine konduğunda belirli bir gözlem biriminin değerleri geldiğinde yani bağımsız değişken değerleri geldiğinde bu etkileşimden sonra çıkan değer z değeridir bu x eksenini.

Bu sigmoid fonksiyonunun karşılığı ise buradaki mavi çizgi. Buradan elde edilen değer (yani z) sigmoid fonksiyonuna yerleştirildiğinde buradan çıkacak olan değer 0 ile 1 arasında olacaktır. Klasik regresyon kullansaydık, bağımlı değişken 1 ve 0 lardan oluşuydu bu durumda elde edeceğimiz değerlerin 1 ve 0 arasında olması durumu garanti değildi. Öyle bi fonksiyon kullanayım ki çıktısını 0 ile 1 arasına eşlesin. Sigmoid fonksiyonunun görevi budur. Bu doğrusal formdan gelen yapı 0 ile 1 arasına dönüştürülür.

**\*\* Y EKSENİ : Yani tahmin edilen değerler \*\***

X bağımsız değişkenleri değerlerini aldığında y yani bağımlı değişkenin birinci sınıfının gerçekleşmesi olasılığı ile ilgileniyorum. Yani sigmoid fonksiyonundan bu dönüşüm işlemi yapıldığında ilgilendiğimiz problemde 1 sınıfı her neyse (hayatta kaldı/kalamadı senaryosu gibi) dolayısıyla bu fonksiyonun ürettiği değer bir olasılık değeridir. Örneğin 0.78 olsun. İşte bu değer = ilgili problemdeki 1 sınıfının gerçekleşmesi olasılığıdır.

Dolayısıyla sınıflandırma problemleri ile ilgilenirken, aslında elimizde yapacak olduğumuz tahmin sonuçları neticesinde 1 sınıfına ait olma olasılığı olur. Daha sonra bu 1 sınıfına ait olma olasılıklarını biz bir eşik değere göre dönüştürürüz.

Ağırlıkları bulma işlemi yani gradient descent kullanarak nasıl bulacağımızı görelim

## Lojistik Regresyon için Gradient Descent

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)})$$

$$J(\theta) = \frac{1}{m} \left[ \sum_{i=1}^m -y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right]$$

$-1 \mid \log(0.80)$

Cost fonksiyonunda  $h(x_i)$  tahmin edilen değerler,  $y_i$  ler tahmin edilen değerleri ifade eder.

Yine bir cost fonksiyonumuz var. ( $h_{\theta}(x_i)$ ) ifadesi tahmin edilen değerleri,  $y_i$  ise gerçek değerleri ifade etmektedir. Buradaki cost fonksiyonumuz log loss ifadesidir. Cross Entropy formülüdür. Entropy şu anlama gelir: Bizim ilgilendiğimiz konu kapsamında entropi ne kadar yüksek ise çeşitlilik o kadar fazladır. Dolayısıyla entropi nin düşük olmasını çeşitliliğin az olmasını isteriz . Hangi açıdan ? Gerçek değer ile tahmin edilen değer açısından.

Sonuc olarak; Gerçek değerler ile o gerçek değerlerin gerçekleşmesi olasılıkları ifadeleri birbirine ne kadar yakınsa bu durumda loss değerimiz kayıp değerimiz o kadar küçük olacak .

Elimizde sınıflara ait olma olasılıkları var. Sınıfların gerçek değerleri var. Dolayısıyla bunun üzerinden entropy hesabı yapılarak bu işlemler toplandıktan sonra ortalaması alınır ve ortalama hata bulunur.

Dikkat ! belirli bir ağırlık kombinasyonu ile bu sınıflandırma işlemi gerçekleştiğinde sınıflandırmaya giderken ifade etmiştik ki her zaman 1 sınıfına ait olma olasılığı hesaplanır. Dolayısıyla bu 1 sınıfına ait olma olasılığı burada(formulun ilk kısmında) değerlendirilir. 1 - 1 sınıfına ait olma olasılığı diyerek 0 sınıfına ait olma olasılığı da burada değerlendirilerek (formulun + sından sonrası) herbir gözlem birimi için hata hesaplanır.Loss hesaplanır. Kayıp hesaplanır. Bunlar toplanıp ortalaması alındığında ortalama hata elde edilir. Bunun düşük olduğunu bekleriz.

Peki bir iterasyonda bu hesaplandı diyelim. türev( gradient descent) burada nasıl devreye giriyor?

Türevlenebilen ve bir parametre ile şekillenebilen bir fonksiyonu ilgili parametreye göre kısmi türev alındığında bu kısmi türev belli bir öğrenme oranı ile çarpılarak ilgili parametrenin eski değerleri ile işleme sokulur ve parametrenin değeri güncellenir.Bu güncellemeler esnasında hatanın düşmesi beklenilir.

## Sınıflandırma Problemlerinde Başarı Değerlendirme

### Confusion Matrix (Karmaşıklık Matrisi)

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

True gördüğüm yerlerde doğru tahminler yaptığımızı , false gördüğümüz yerlerde yanlış tahminler yaptığımızı anlıyoruz. False 'dan sonra gördüğümüz negatif ifadesi ise yapmam gerekenden farklı ve yanlış olarak yaptığım şeyi ifade etmektedir. Negatif ifadesi : yanlış yaptığım sınıfı ifade eder.

**Accuracy** : Doğru sınıflandırma oranıdır. Doğru yaptığımız işlemlerin toplamıdır.  $(TP+TN)/(TP+TN+FP+FN)$

**Precision** : Pozitif sınıf(1) tahminlerinin başarı oranıdır. Tahmin ettiklerimizde ne kadar başarılıyım ? sorusunun cevabıdır.  $TP/(TP+FP)$

**Recall** Pozitif sınıfın (1) doğru tahmin edilme oranıdır.  $TP/(TP+FN)$

False Pozitif = yanlış alarm üretmek demektir. Yani hata yaptık . 1. tip hata denir.

Eğer elinizdeki sınıflandırma problemi dengeli bir sınıf dağılımına sahip ise accuracy verebilirsiniz. Ama eğer elinizdeki veri setinde sınıflandır dengersiz dağılıyorsa accuracy kullanamazsınız. Bu durumda Recall ve Precision değerlerine bakmak gerekir.

**F1 SKORU** : Precision ve Recall değerlerinin harmonik ortalamasıdır. Precision değeri tahminlerimizin basarsına odaklanmıştır. Recall değeri Gerçekleri yakalama başarımıza odaklanmıştır.  $2(Precision \cdot Recall) / (Precision+Recall)$

Accuracy = Doğru sınıflandırma Sayısı / Toplam sınıflandırılan Gözlem sayısı

### Classification Threshold

Sınıflandırma modellerinde hedef değişken için sigmoid fonksiyonu ile bir olasılık değeri hesaplanır. Bu olasılık değeri bir threshold değeri ile kıyaslanarak ilgili sınıfa atanır. Varsayılan olarak bu threshold değeri 0.5 tir.

Classification Threshold ' u yukarı doğru çekersek (her zaman için değil belki ama) accuracy 'nin düşmesini mi bekleriz ? EVET.

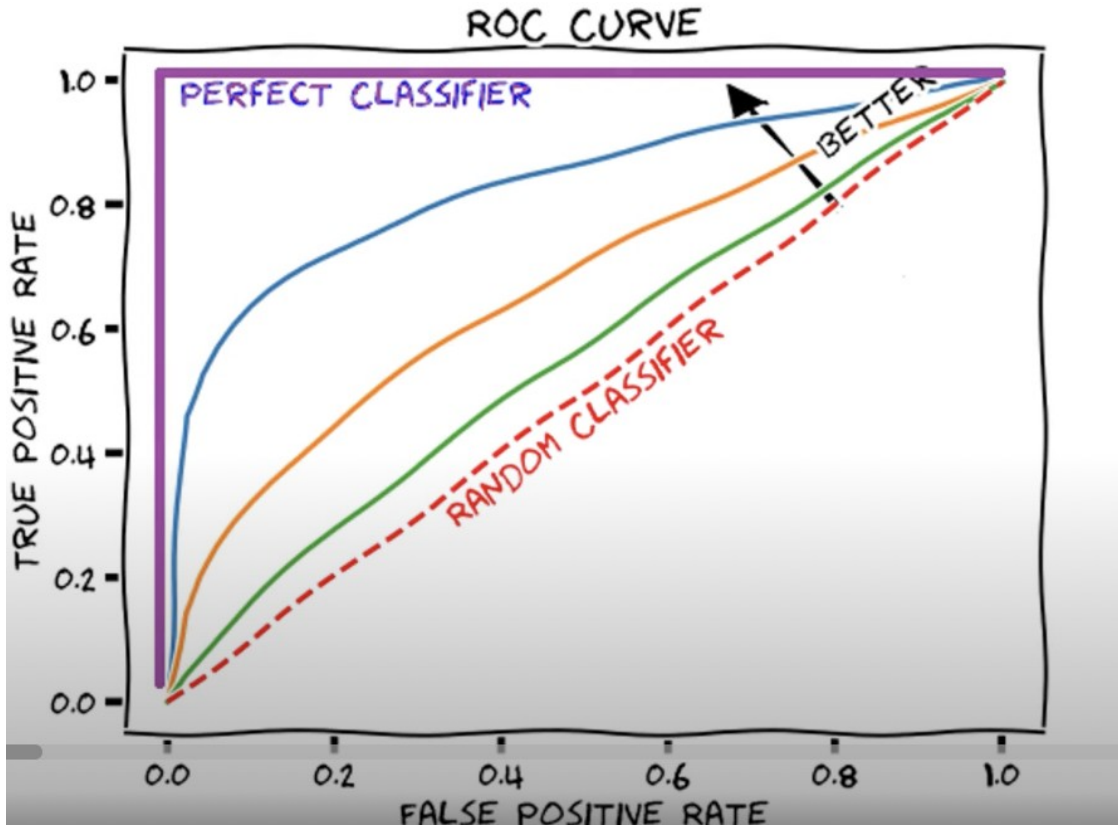
Threshold değiştirdikçe accuracy recall hepsi değişiyor. Ne yapacağım ? Bu problemi çözmek için ROC Curve yöntemi kullanılır.

Eşik değerinin değişmesi ile sınıflandırma tahminlerimiz değişiyor. O zaman öyle bir şey olmalı ki olası bütün eşik değerlerin değişimleri de göz önüne alınarak bir hesaplama yapılsın ve buna göre bir matematiksel ifade karşımıza gelsin.

Bu problemi çözmek için ROC Curve yöntemi kullanılır.

### Roc Eğrisi

Olası eşik değeri değişimlerine karşılık başarı değişimleri nedir sorusunun cevabı için kullanılır.



Bütün olası threshold'ları çıkarayım. Bu olası threshold lara göre karmasıklık matrisini oluşturayım. Bunun üzerinden herbir olası eşik değerine karşılık true positive rate ve false positive rate 'i hesaplayıp grafikte işaretleyeyim . Yani bizi threshold değişikliği kaygısından kurtarır.

Bir eğri ne kadar fazla yer kaplıyorsa o kadar o modelin başarılı olduğu anlamına gelir.

Bu roc eğrisinin altında kalan alanın integrali alınırsa bu durumda area under curve elde edilir(AUC). AUC metriği roc curve eğrisinin tek bir sayısal değer ile ifade edilişidir.

AUC = roc eğrisi altında kalan alandır. AUC tüm olası sınıflandırma eşikleri için toplu bir performans ölçüsüdür.

Öyleyse toparlayalım. Demekki sınıflandırma problemleri söz konusu olduğunda dikkat etmemiz gereken birinci konu; sınıf dağılımı dengesiz mi ? yani 95 ' e 5, 90'a 10 vs gibi.

Eğer veri seti dengesiz ise (95 e 5)bu durumda recall değerine , precision değerine ve bunların harmonik değeri olan F1 skoruna bakıyoruz. Dengeli de olsa dengesiz de olsa buna bakıyoruz (F1 skoruna)

## LOG LOSS

Logloss sınıflandırma problemi için optimize edilecek fonksiyondur. Başarı metriğidir.

Aynı zamanda cross entropi metriğidir. Entropi: çeşitlilik, bilgidir. Ne kadar düşük ise o kadar iyidir. Entropi ne kadar yüksek ise çeşitlilik varyans da o kadar yüksektir. Lojistik regresyonda bu çeşitlilik ve bilgi durumu yapılan tahminleme ile gerçek değerler arasındaki durumu değerlendirmek için kullanılır

$$Log Loss = \frac{1}{m} \left( \sum_{i=1}^m -y_i * \log(p(\hat{y}_i)) - (1 - y_i) \log(1 - p(\hat{y}_i)) \right)$$

Logloss mse değeri gibi sınıflandırma problemlerinde başarı metriğidir. Ne kadar küçük o kadar iyi olarak bakarız olaya.

Hem de Optimize edilmek üzere amaç fonksiyonudur. Düşmesini beklediğimiz optimize etmeye çalıştığımız fonksiyonumuzdur.