Hafta 7(ML Temel Kavramlar - Linear Regression - Logistic Regression)

@mebaysan

08/10/2021

Bu haftanın veri seti

-

İlgili Okuma Listesi

```
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/531549a7-1635-4393-a4fb-c449f4958abc/makine_ogr_giris_carsamba.pdf
```

```
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f983ced3-8ef2-4f6d-9b24-8a85a7701547/makine_ogr_girispers.pdf
```

- Dengesiz Veri Seti (Uygulama)

Benim Yazdığım Yazılar

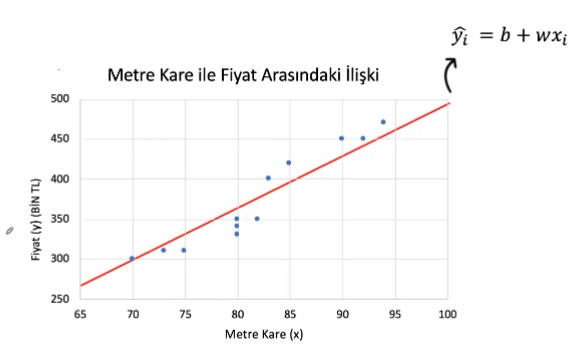
-

Makine Öğrenmesi Nedir?

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

Aşağıdaki örnek bir regresyon problemidir. Bağımlı değişken sayısal bir değişken ise problem regresyondur. "Öğrendik" dediğimiz şey ise kırmızı çizgidir.

metre_kare (x_i)	fiyat (y_i)
70	300
73	310
75	310
80	330
80	340
80	350
82	350
83	400
85	420
90	450
92	450
94	470



Değişken Türleri

- Sayısal
- Kategorik
  - Nominal → Sınıflar arası fark yoktur
  - Ordinal → Sınıflar arası fark vardır
- Bağımlı değişken (target, dependent, output, response)
  - Tahmin etmek istediğimiz değişken

- Bağımsız değişken (feature, independent, input, column, predictor, explanatory)
  - Bağımlı değişkeni tahmin ederken kullanacağımız değişken(ler)

Öğrenme Türleri

- Denetimli Öğrenme (Supervised Learning)
  - Bağımlı değişkenin veri setinde olduğu öğrenme tiplerine denir. Çünkü ne olduğunda ne olduğunu biliriz.
- Denetimsiz Öğrenme (Un-Supervised Learning)
  - Bağımlı değişken veri setinde yoktur.
- Pekiştirmeli Öğrenme (Reinforcement Learning)

Problem Türleri

- Regresyon problemlerinde bağımlı değişken sayısaldır.
- Sınıflandırma problemlerinde bağımlı değişken kategoriktir.

Model Başarı Değerlendirme Yöntemleri

Temeli "tahminlerin ne kadar başarılı?" sorusuna dayanır.

Regresyon Problemlerinde Değerlendirme

- MSE (Mean Squared Error)

    - Gerçek değerlerden tahmin edilen değerler çıkarılır, kareleri alınır. Kareler toplanır ve ortalaması alınır.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- RMSE (Root Mean Squared Error)
    - MSE'nin kare kökü alınır

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

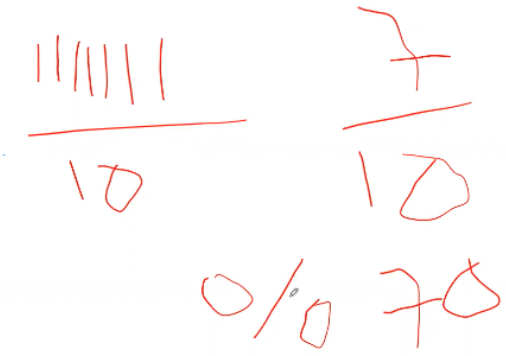
- MAE (Mean Absolute Error)
    - Gerçek değerlerden tahmin edilen değerler çıkarılır ve mutlak değeri alınır. Mutlak değerler toplanıp, ortalaması alınır.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Sınıflandırma Problemlerinde Başarı Değerlendirme

churn

Age	Total_Purchase	Years	Churn	Predicted_Churn
42.0	11066.8	7.22	1	1
41.0	11916.22	6.5	1	0
38.0	12884.75	6.67	0	0
42.0	8010.76	6.71	1	0
37.0	9191.58	5.56	0	1
48.0	10356.02	5.12	1	1
44.0	11331.58	5.23	0	0
32.0	9885.12	6.92	0	0
43.0	14062.6	5.46	1	1
40.0	8066.94	7.11	1	1



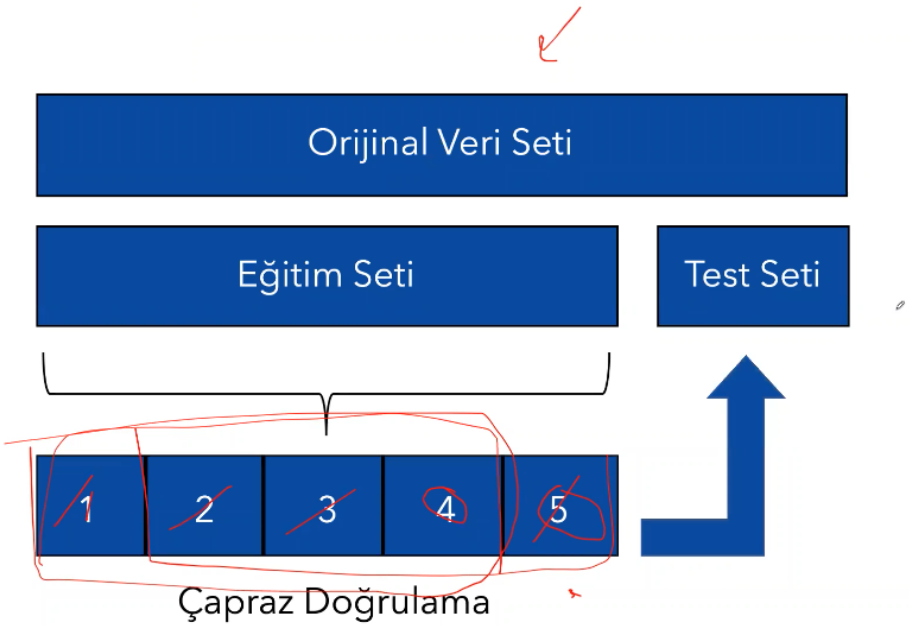
- Accuracy
  - Doğru Sınıflandırma Sayısı / Tahmin Edilen Gözlem Sayısı

Model Doğrulama (Model Validation) Yöntemleri

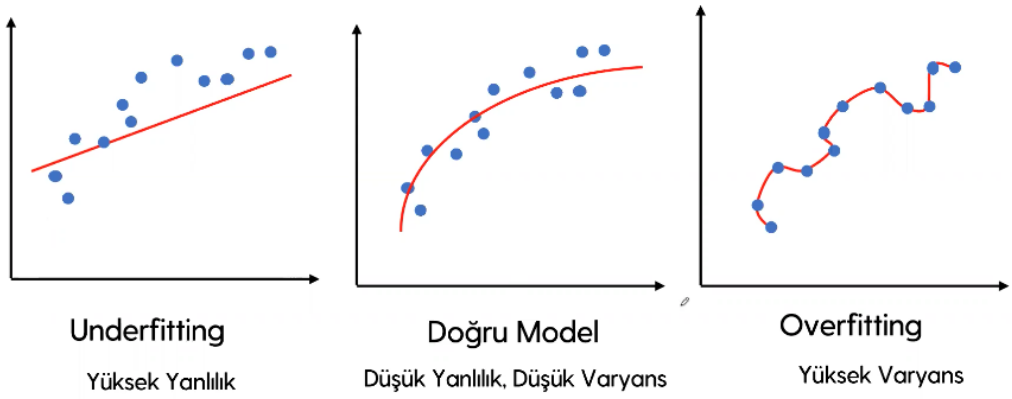
Model, eğitildiği veri dışında hiç görmediği veri üzerinde nasıl çalışacak kaygısını giderme çalışmalarıdır.

- Holdout Yöntemi (Sınama Seti Yöntemi)

- K-Katlı Çapraz Doğrulama Yöntemi (Cross Validation)
  - Örnek olarak veriyi 5 parçaya böldük. 4 parça ile model kurulur, kalan 1 parça ile test edilir. Daha sonra diğer 4 parça ile model kurar ve diğer kalan parça ile test ederiz. Daha sonra bunların ortalamasını alır ortalama hatamıza bakarız.
  - Tüm veri setine de uygulanabilir
  - İstenirse sadece ayrılan eğitim veri setine de uygulanabilir



Yanlılık - Varyans Değiş Tokuşu (Bias-Variance Tradeoff)



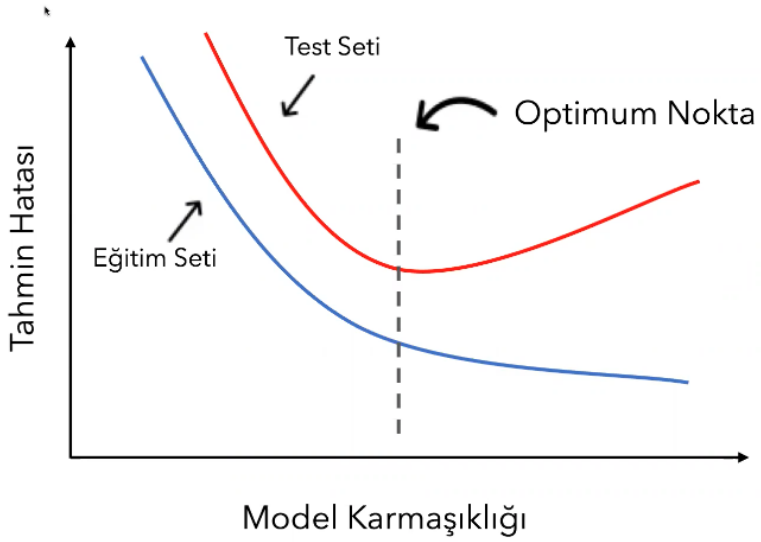
- Underfitting → Az öğrenme
- Overfitting → Aşırı öğrenme
  - Modelin; veri setinin içerisindeki yapıyı değil, veri setinin kendisini öğrenmesidir.

Model Karmaşıklığı

Overfitting'i nasıl yakalarız?

Model karmaşıklığını bir eksene, tahmin hatasını bir eksene koyarız. Eğitim ve test hatasını bir arada gözlemleyerek overfitting'i yakalarız.

Modeli; eğitim ve test setinin ayrıklaşmaya başladığı noktada tutmaya çalışırız. Bu nokta optimum noktadır.



Model karmaşıklığının tanımı modelden modele değişmektedir.

Doğrusal regresyon problemi ile ilgileniyorsak, üstel terimlerin artması model karmaşıklığını artırabilir.

Light GBM için model karmaşıklığı iterasyon sayısına karşılık gelir.

Doğrusal Regresyon (Linear Regression)

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

$$\hat{y}_i = b + wx_i$$

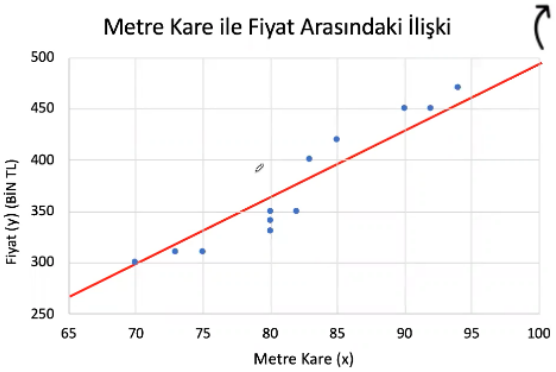
$$\hat{y}_i = b + w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_px_p$$

Gerçek değerler ile tahmin edilen değerler arasındaki farkların karelerinin toplamını/ortalamasını minimum yapabilecek b ve w değerlerini bularak.

$$\hat{y}_i = b + wx_i$$

metre_kare (x_i)	fiyat (y_i)
70	300
73	310
75	310
80	330
80	340
80	350
82	350
83	400
85	420
90	450
92	450
94	470

$\hat{y}_i = b + wx_i$



$$Cost(b,w) = \frac{1}{2m}\sum_{i=1}^{m}((b + wx_i) - y_i)^2$$

ÖRNEK Soru:

metre_kare (x_i)	fiyat (y_i)
70	300
73	310
75	310
80	330
80	340
80	350
82	350
83	400
85	420
90	450
92	450
94	470

Soru: Aşağıda verilen bias ve weight'e göre 90 metre kare bir ev için fiyat tahmini yapınız.

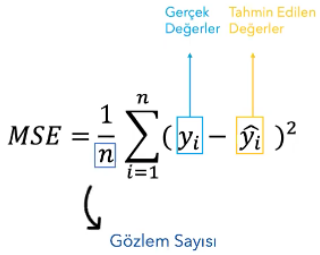
b = 210, w = 4

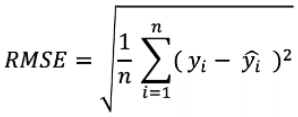
$\hat{y}_i = b + wx_i$

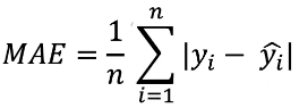
Bu w ve b değerlerine göre: y_hat = 210 + 4 * 90

Regresyon Modellerinde Başarı Değerlendirme (MSE, RMSE, MAE)

fiyat (y_i)	fiyat tahmin ($\hat{y}_i$)	hata ($e_i = y_i - \hat{y}_i$)	hata_kare ($e_i = y_i - \hat{y}_i$)^2
300	320	-20	400
310	280	30	900
310	290	20	400
330	329	1	1
340	330	10	100
350	352	-2	4
350	400	-50	2500
400	410	-10	100
420	460	-40	1600
450	420	30	900
450	410	40	1600
470	480	-10	100



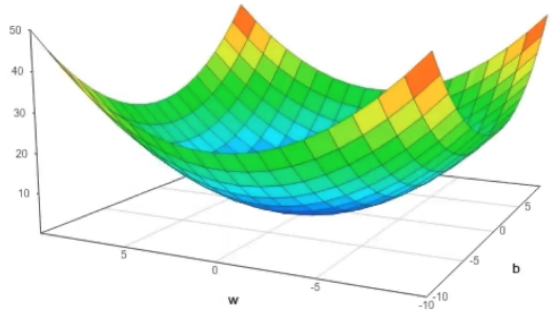




MSE = (400 + 900 + 400 + 1 + 100 + 4 + 2500 + 100 + 1600 + 900 + 1600 + 100) / 12 = 717.08

Parametrelerin Tahmin Edilmesi (Ağırlıkların Bulunması)

$$Cost(b,w) = \frac{1}{2m}\sum_{i=1}^{m}((b + wx_i) - y_i)^2$$



Amacımız Cost'u 0 yapmak. En küçük MSE değerini veren b ve w değerlerini bulmak.

Target / Dependent variable / Actual values:	y_i
y_hat / y_pred / hypothesis / Predicted values: $\hat{y}_i = b + mx_i$ $\hat{y}_i = b_0 + b_1x_i$ $h_\theta(x) = \theta_0 + \theta_1x$	$\hat{y}_i = b + wx_i$
Loss / Residual / Error:	$\hat{y}_i - y_i$
Objective / Loss / Cost function (MSE):	$\frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$
Objective / Loss / Cost function (MSE):	$\frac{1}{2m}\sum_{i=1}^{m}((b + wx_i) - y_i)^2$
Parameters / Coefficients / Weights:	b, w
	- EKK / Normal Denk. Yöntemi - Gradient Descent Yöntemi

EKK / Normal Denklemler Yöntemi

Sci-kit learn, statsmodel vb Python kütüphanelerinde kullanılan parametre tahmin yöntemi bu yöntemdir.

Analitik Çözüm: Normal Denklemler Yöntemi (En Küçük Kareler Yöntemi)

Simple Linear Regression

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2$$

$$b_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

$$b_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

Multiple Linear Regression

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

$$\hat{\beta} = (X^T.X)^{-1}X^T.Y$$

Gradient Descent

Bir optimizasyon yöntemidir. Belirli bir iterasyon ve belirli bir öğrenme hızında katsayı değerlerini güncelleyerek hata metriğine bakar, hata kabul edilebilir bir noktaya düşene kadar işlemleri tekrar eder.

Optimizasyon Çözümü: Gradient Descent

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

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

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

Gradient Descent, yapay öğrenme veya regresyon için doğmuş bir teknik değildir. Zaten var olan bir optimizasyon yöntemidir.

Bir fonksiyonu minimum yapabilecek parametre değerlerini bulmak için kullanır.

$$\text{Repeat until convergence } \{$$

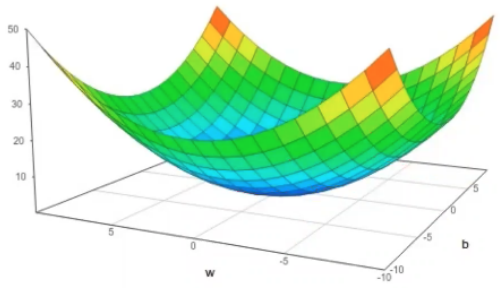
$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\}$$

Gradyanın negatifi olarak tanımlanan "en dik iniş" yönünde iteratif olarak parametre değerlerini güncelleyerek ilgili fonksiyonun minimum değerini verebilecek parametreleri bulur.

Linear Regression'da ise Cost fonksiyonunu (MSE) minimize edebilecek parametreleri bulmak için kullanılır.

Cost fonksiyonunu minimize edebilecek parametreleri bulmak için kullanılır.

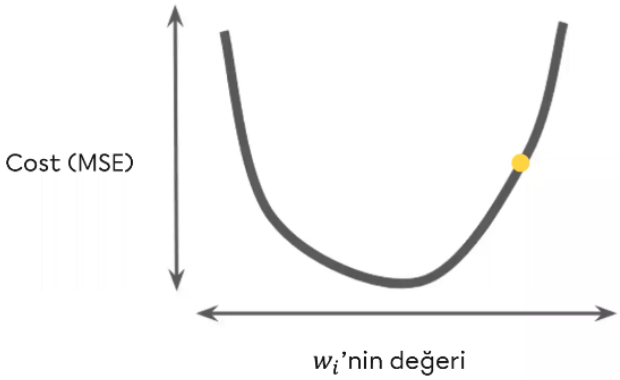


$$Cost(b,w) = \frac{1}{2m}\sum_{i=1}^{m}((b + wx_i) - y_i)^2$$

$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\underset{\theta_0,\theta_1}{minimize} J(\theta_0,\theta_1)$$

Minimum noktayı bulmaya çalışırız:



$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\underset{\theta_0,\theta_1}{minimize} J(\theta_0,\theta_1)$$

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

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

Gradient Descent Formları

- Batch Gradient Descent (Gradyan hesabı için her iterasyonda tüm gözlemler gezilir)
- Stochastic Gradient Descent (Gradyan hesabı için her iterasyonda rastgele 1 örnek gezilir)
- Mini-Batch Stochastic Gradient Descent (Gradyan hesabı için her iterasyonda rastgele 10-1000 örnek gezilir)

Logistic Regression

Bağımlı değişkenimiz KATEGORİK bir değişkendir. Problemimiz ise sınıflandırmadır. ML çalışırken karşılaşacağımız problemlerin %90-95'i sınıflandırma problemidir. Modelin adında Regression geçse de sınıflandırmak için kullanılan bir modeldir.

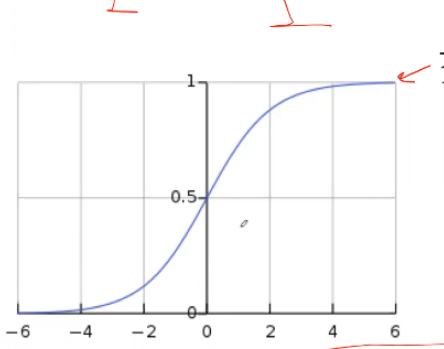
Nasıl? Gerçek değerler ile tahmin edilen değerler arasındaki farklara ilişkin log loss değerini minimum yapabilecek ağırlıkları bularak.

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

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

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

$$\hat{y}_i = P(Y = 1 \mid X = x)$$



$$\frac{1}{1 + e^{-(z)}}$$

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

Bağımsız değişkenler(X) x olduğunda, bağımlı değişkenin(Y) 1 olma olasılığı ile ilgileniyoruz. Bu problemler ile çalışırken gelende hedef sınıfımızı 1 ile temsil edeceğiz.

Soru: Verilen bias ve weight'lere göre aşağıdaki gözlem birimi için 1 sınıfına ait olma olasılığını hesaplayınız.

$b = 5,\ w_1 = 4,\ w_2 = -4,\ w_3 = 3$

$x_1 = 2,\ x_2 = 3,\ x_3 = 0$

İpucu: $\frac{1}{1 + e^{-(z)}}$ $z = b + w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_px_p$

Çözüm:

$z = 5 + (4) * 2 + (-4) * 3 + 3 * (0)$
$z = 1$

$$\frac{1}{1 + e^{-(1)}} = 0.731$$

Python'da Logistic Regression

Logistic Regression İçin Gradient Descent

Burada bizim için entropi'nin tanımı: Entropi ne kadar yüksek ise o kadar çeşitlilik fazladır. Entropi ne kadar az ise çeşitlilik o kadar azdır.

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

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

$$\begin{aligned} &Repeat\ \{ \\ &\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \\ &\} \end{aligned}$$

Sınıflandırma Problemlerinde Başarı Değerlendirme

F1 Score, dengesiz veriye karşı daha iyi bir sonuç verebilir

AUC, bir sınıfa ait olma olasılığının olası threshold'lara göre ne elde edebileceğime dair sinyal verir

Recall, **gerçekteki 1 olan sınıfı** ne kadar başarılı tahmin etmişiz

Precision, 1 olan sınıfların ne kadarını doğru tahmin etmişiz

Accuracy, Doğru sınıflandırma oranı. Yani 1e 1, 0a 0 deme oranımız.

Confusion Matrix(Karmaşıklık Matrisi)

Sınıflandırma problemlerinin temeli buraya dayanmaktadır.

		Tahmin Edilen Sınıf	
		Sınıf = 1	Sınıf = 0
Gerçek Sınıf	Sınıf = 1	True Pozitif (TP)	False Negatif (FN)
	Sınıf = 0	False Pozitif (FP)	True Negatif (TN)

- **Accuracy**: Doğru sınıflandırma oranıdır: **(TP+TN) / (TP+TN+FP+FN)**
- **Precision**: Pozitif sınıf (1) tahminlerinin başarı oranıdır: **TP / (TP+FP)**
- **Recall**: Pozitif sınıfın (1) doğru tahmin edilme oranıdır: **TP / (TP + FN)**
- **F1 SCORE:** 2 * (Precision * Recall) / (Precision + Recall)

Kritik soru: FP mi FN mi daha kritiktir?

FN daha kritiktir. Örnek: Dolandırıcı bir işleme dolandırıcı değil demek.

Örnek bir Confusion Matrix doldurma. 990 normal işlem, 10 sahte işlem. 10 sahte işlemin 5'i doğru 5'i yanlış tahmin edildi. 990 normal işlemin 90'ı yanlış tahmin edildi.

Soru: 1000 kredi kartı işlemi var. 990 normal işlem. 10 sahtekar işlem.
Buna göre Confusion matrisi doldurunuz ve başarı metriklerini hesaplayınız.

		Tahmin Edilen Sınıf Değerleri		
		Fraud İşlem (1)	Normal İşlem (0)	
Gerçek Sınıf Değerleri	Fraud İşlem (1)	5	5	**10**
	Normal İşlem (0)	90	900	**990**
		95	**905**	

- Accuracy = (5 + 900) / 1000 = 0.905
- Precision = 5/(5+90) = 0.05
- Recall = 5 / (5+5) = 0.50
- F1 Score = 2*0.025/ 0.55 = 0.09

Classification Threshold

Sınıflandırma problemlerinde amacımızın; hedef değişkenin hedef sınıfda (1) olma olasılığını bulmak olduğunu söylemiştik. Classification Threshold ise; Bulduğumuz olasılık için eşik değeri temsil etmektedir. Hedef sınıfta olma olasılığı bu eşik değerin üzerinde olasılığa sahip değerler hedef sınıf olarak kabul edilir.

Aşağıdaki örnekte, "Probability of Class 1" değişkeni 0.50 (Classification Threshold) üzerinde olan gözlemler hedef sınıfta (1) olarak kabul (tahmin) edilmektedir ("Predicted_Churn" değişkeni 1 olarak tahmin edilen değerlerden).

Churn	Predicted_Churn	Probability of Class 1
1	1	0,80
1	0	0,48
0	0	0,30
1	0	0,45
0	1	0,55
1	1	0,70
0	0	0,42
0	0	0,35
1	1	0,60
1	1	0,70

$$\text{Accuracy} = \frac{Doğru\ Sınıflandırma\ Sayısı}{Toplam\ Sınıflandırılan\ Gözlem\ Sayısı}$$

Classification threshold = 0.50 ise accuracy nedir?

7/10 = 0.70

Aynı veri setine baktığımızda Classification Threshold'u 0.75 yaptığımızda Accuracy'inin düştüğünü gözlemleyebiliriz.

Churn	Predicted_Churn_0.75	Probability of Class 1
1	1	0,80
1	0	0,48
0	0	0,30
1	0	0,45
0	0	0,55
1	0	0,70
0	0	0,42
0	0	0,35
1	0	0,60
1	0	0,70

$$\text{Accuracy} = \frac{Doğru\ Sınıflandırma\ Sayısı}{Toplam\ Sınıflandırılan\ Gözlem\ Sayısı}$$

Classification threshold = 0.75 ise accuracy nedir?

5/10 = 0.50

Classification Threshold düşürüldükçe accuracy artar.

Churn	Predicted_Churn_0.40	Probability of Class 1
1	1	0,80
1	1	0,48
0	0	0,30
1	1	0,45
0	1	0,55
1	1	0,70
0	1	0,42
0	0	0,35
1	1	0,60
1	1	0,70

$$\text{Accuracy} = \frac{Doğru\ Sınıflandırma\ Sayısı}{Toplam\ Sınıflandırılan\ Gözlem\ Sayısı}$$

Classification threshold = 0.40 ise accuracy nedir?

8/10 = 0.80

Peki neden threshold 0.50 olarak kabul edildi? Threshold değerini neye göre belirliyoruz? Bu sorunun cevabı ROC Curve'de bulunmaktadır.

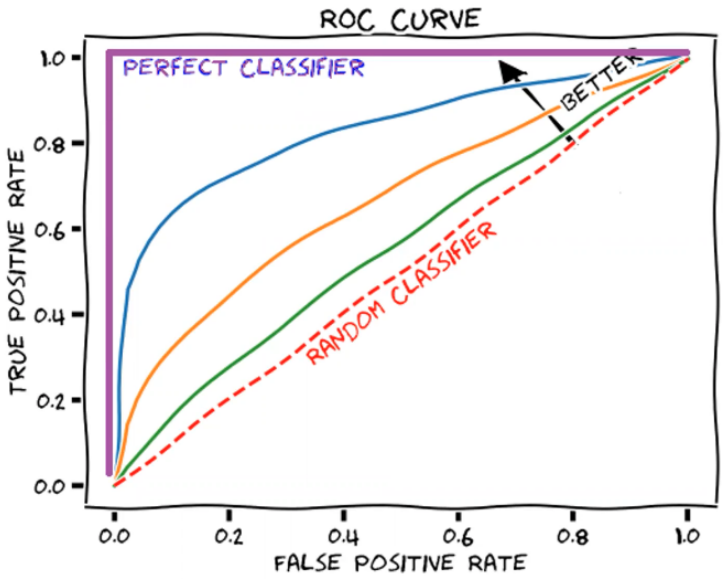
ROC Curve (Receiver Operating Characteristic Curve)

Çeşitli threshold değerlerine True Positive Rate ve False Positive Rate değerlerini görselleştirelim.

Yani olası tüm threshold için TPR ve FPR hesaplanır.

Çizgiler her bir olası threshold değerine göre oluşan TPR ve FPR değerlerinin kesişimleridir.

Kırmızı ve mavi çizgilerin arasındaki alanın mümkün olduğunca geniş olmasını isteriz.



AUC (Area Under Curve)

- ROC eğrisinin tek bir sayısal değer ile ifade edilişidir.
- ROC eğrisi altında kalan alandır.
- AUC, tüm olası sınıflandırma eşikleri için toplu bir performans ölçüsüdür.

Log Loss

Ne kadar düşük o kadar iyi

Ne kadar yüksek o kadar kötü

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

```python
# log loss
yi = [1, 1, 0, 1, 0, 1, 0, 0, 1, 1]
y_prob = [0.80, 0.48, 0.30, 0.45, 0.55, 0.70, 0.42, 0.35, 0.60, 0.70]

# birinci gözlem için 1 ve 0.80 değerlerine göre logloss
-1*(np.log10(0.80))
# 0.096

# üçüncü gözlem için 0 ve 0.30 değerlerine göre logloss
-1 * np.log10(1 - 0.30)
# 0.15

# logloss
log_loss_sum = 0

for i, y in enumerate(yi):
    if y == 1:
        log_loss_sum += -1 * (np.log10(y_prob[i]))
    else:
        log_loss_sum += -1 * np.log10(1 - y_prob[i])


log_loss_sum/len(yi)
# 0.2219
```