

# Derin Öğrenme Deep Learning

Hazırlayan: M. Ali Akcayol  
Gazi Üniversitesi  
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016." kitabı kullanılarak hazırlanmıştır.

## İçerik

---

- ▶ Makine öğrenmesi
- ▶ Öğrenme algoritmaları
- ▶ Uygulamalar
- ▶ Performans ölçütleri
- ▶ Öğrenme türleri
- ▶ Kapasite, overfitting ve underfitting
- ▶ Hyperparameters ve validation kümesi

## Makine öğrenmesi

---

- ▶ **Derin öğrenme**, makine öğrenmesinin özel bir türüdür.
- ▶ Çoğu makine öğrenmesi algoritması probleme yönelik ayarlanması gereken **parametrelere sahiptir** (**hyperparameters**).
- ▶ **Makine öğrenmesi algoritmalarının çoğu iki kategoriden birisinde yer alır:**
  - ▶ Denetimli öğrenme (**supervised learning**)
  - ▶ Denetimsiz öğrenme (**unsupervised learning**)
- ▶ **Derin öğrenme algoritmalarının temeli** genellikle optimizasyon algoritmasına (**stochastic gradient descent**) dayanır.
- ▶ Bir **makine öğrenmesi algoritması oluşturulurken, maliyet fonksiyonu** (**cost function**), **model** ve **veriseti** kullanılır.

3

## İçerik

---

- ▶ Makine öğrenmesi
- ▶ **Öğrenme algoritmaları**
- ▶ Uygulamalar
- ▶ Performans ölçütleri
- ▶ Öğrenme türleri
- ▶ Kapasite, overfitting ve underfitting
- ▶ Hyperparameters ve validation kümesi

4

## Öğrenme algoritmaları

---

- ▶ Makine öğrenmesi algoritmaları **veriden öğrenebilen algoritmalarıdır.**
- ▶ **Makine öğrenmesi**, klasik programlarla çözülmesi çok **zor problemlerle ilgilenir.**
- ▶ Makine öğrenmesi algoritmaları ile **bir örnek üzerinde nasıl işlem yapılacağı tanımlanır.**
- ▶ Bir **örnek**, nesnelere, olaylar veya sistemlerden ölçülen **nitelikler (features) topluluğudur.**
- ▶ Bir örnek, bir vektör ile ( $\mathbf{x} \in \mathbb{R}^n$ ) gösterilir. Vektördeki her giriş  $x_i$ , ayrı bir özelliktir.

5

## İçerik

---

- ▶ Makine öğrenmesi
- ▶ Öğrenme algoritmaları
- ▶ **Uygulamalar**
- ▶ Performans ölçütleri
- ▶ Öğrenme türleri
- ▶ Kapasite, overfitting ve underfitting
- ▶ Hyperparameters ve validation kümesi

6

## Uygulamalar

- ▶ **Makine öğrenmesi** çok farklı alanlarda birlikte kullanılan disiplinler arası bir alandır.



7

## Uygulamalar

- ▶ Makine öğrenmesi ile çok sayıda farklı türdeki problem çözülebilir:
  - ▶ Sınıflandırma
  - ▶ Eksik giriş ile sınıflandırma
  - ▶ Regresyon
  - ▶ Transcription
  - ▶ Makine çevirisi
  - ▶ Yapılandırılmış çıkış
  - ▶ Anormallik tespiti
  - ▶ Sentezleme ve örnekleme
  - ▶ Eksik değerleri tahmin etme
  - ▶ Gürültüyü yok etme
  - ▶ Yoğunluk tahmini

8

## Uygulamalar

### Sınıflandırma

- ▶ **Giriş değerlerine** göre ait olduğu  $k$  **kategorisi atanır.**
- ▶ Öğrenen algoritma bir fonksiyon üretir.

$$f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$$

- ▶ Model,  $x$  giriş vektörü için  $y$  çıkışını elde eder.

$$y = f(x)$$

- ▶ **Görüntü içerisinde nesne tanımlama/sınıflandırma**, fotoğraf koleksiyonundan insanların tespit edilmesi gibi nesne tanıma **uygulamaları derin öğrenme ile gerçekleştirilebilir.**

9

## Uygulamalar

### Eksik giriş ile sınıflandırma

- ▶ Sınıflandırma problemi **tüm girişler uygun olmadığında daha zor hale gelir.**
- ▶ Eğer bazı **girişler eksik olursa, öğrenme algoritması** bir fonksiyonu öğrenmek yerine bir **fonksiyon kümesini öğrenir.**
- ▶ Her fonksiyon  $x$  vektöründeki **farklı kayıp girişleri içeren altkümede işlem yapar.**
- ▶ Bu tür durumlar **tıbbi teşhis** uygulamalarında sıklıkla ortaya çıkar.
- ▶ **Görüntüden karakter elde etme** ([optical character recognition](#)) uygulamaları da **eksik bilgiye göre sınıflandırma yapar.**

10

## Uygulamalar

### Regresyon

- ▶ Bu tür problemlerde **makine öğrenmesi algoritması** girişlere göre **sayısal bir değeri tahmin eder.**
- ▶ Bu tür problemlerin çözümünde, **öğrenen algoritma** girişlere göre **fonksiyonun sayısal çıkışını elde eder.**

$$f : \mathbb{R}^n \rightarrow \mathbb{R}$$

- ▶ Sigortalı personelin tazminat **talep tahmini** veya menkul kıymetlerin ileriye dönük **değer tahmini regresyon ile yapılabilir.**

11

## Uygulamalar

### Transcription

- ▶ **Yapılandırılmamış** şekilde gösterilen verilerden **metin formatında veri elde edilir.**
- ▶ Makine öğrenmesi ile bir fotoğraf **içerisindeki metinler çıkartılabilir.**
- ▶ Google Street View, fotoğraflardaki adres numaralarını bu şekilde elde etmektedir.
- ▶ **Konuşma tanıma (speech recognition)** uygulamaları **ses dalgalarını karakterlere dönüştürür.**
- ▶ Günümüzdeki konuşma tanıma uygulamalarında **derin öğrenme yaygın bir şekilde kullanılmaktadır.**

12

## Uygulamalar

### Makine çevirisi

- ▶ Makine öğrenmesi ile **bir dildeki karakter dizgisi** başka bir dildeki **karakter dizgisine dönüştürülür**.
- ▶ Derin öğrenme bu tür uygulamalarda çok başarılı bir şekilde uygulanabilmektedir.

### Yapılandırılmış çıkış

- ▶ Çıkış vektörü farklı elemanlar arasındaki önemli ilişkileri içerir.
- ▶ Örneğin, **giriş doğal dildeki bir cümledir, çıkış ise bir ağaç yapısıdır (parsing, mapping)**.
- ▶ Çıkış ağaç yapısı cümlenin gramer yapısını tanımlar (fiil, isim,...).
- ▶ **Hava fotoğraflarından yolların konumlarını belirlemek** için derin öğrenme kullanılabilir.
- ▶ Görüntüyü tanımlayan metin uygulamalarında (**image captioning**) yapılandırılmış çıkış geçerli bir cümledir.

13

## Uygulamalar

### Anormallik tespiti

- ▶ Makine öğrenmesi ile **bir grup olay** veya **nesneden** bazıları **farklı** ve **anormal** olarak işaretlenir.
- ▶ Kredi kartı sahteciliği uygulamasında **anomaly detection** yapılır.
- ▶ Kişinin harcama alışkanlıkları modellenerek kart üzerindeki hatalı veya anormal işlemler tespit edilir.
- ▶ Kredi kartını çalan kişi farklı bir olasılık dağılımına göre alışveriş yapar.

### Sentezleme ve örnekleme

- ▶ Makine öğrenmesi ile **eğitilen verinin benzeri** örnekler üretilir.
- ▶ **Konuşma sentezleme** uygulamasında, bir **metin giriş** olarak verilir ve **ses dalgaları** ile konuşma şeklinde **çıkış** elde edilir.
- ▶ Derin öğrenme ile öğrenilen müzik eserlerinden yeni bir müzik eseri çıkış olarak elde edilebilir.

14

## Uygulamalar

### Eksik deęerleri tahmin etme

- ▶ Bu tür problemlerde makine öğrenmesi algoritmasına yeni bir giriş verilir ( $x \in \mathbb{R}^n$ ), **eksik girişleri tahmin etmesi istenir.**

### Gürültüyü yok etme

- ▶ Bu tür problemlerde, makine öğrenmesi algoritmasına bozulmuş/gürültülü bir örnek ( $\tilde{x} \in \mathbb{R}^n$ ) verilir, temiz ve doğru örnek ( $x \in \mathbb{R}^n$ ) elde edilir.
- ▶ Makine öğrenmesi algoritmaları,  $\tilde{x}$  **gürültülü girişinden  $x$  doğru girişini tahmin eder.**
- ▶ Genel olarak,  $p(x | \tilde{x})$  şartlı olasılık dağılımını tahmin eder.

15

## Uygulamalar

### Yoęunluk tahmini

- ▶ Yoęunluk tahmini probleminde, makine öğrenmesi algoritması **olasılık dağılımı fonksiyonunu öğrenir.**

$$p_{\text{model}} : \mathbb{R}^n \rightarrow \mathbb{R}$$

- ▶  $p_{\text{model}}(x)$ , eęer  $x$  sürekli ise olasılık yoęunluk (**probability density**) fonksiyonu, kesikli ise olasılık kütle (**probability mass**) fonksiyonu olarak adlandırılır.
- ▶ Bir  $p(x)$  olasılık dağılımı öğrenilirse, eksik veri tahmin edilebilir.
- ▶ Eęer,  $x_i$  girişı eksik ve dięer tüm  $x_{-i}$  girişleri var ise, eksik girişı tahmin etmek için  $p(x_i | x_{-i})$  şartlı olasılığı kullanılabilir.

16



## İçerik

- ▶ Makine öğrenmesi
- ▶ Öğrenme algoritmaları
- ▶ Uygulamalar
- ▶ Performans ölçütleri
- ▶ Öğrenme türleri
- ▶ Kapasite, overfitting ve underfitting
- ▶ Hyperparameters ve validation kümesi

17

## Performans ölçütleri

- ▶ Makine öğrenme algoritmasının performansını değerlendirmek için **nicel ölçütler kullanılmalıdır**.
- ▶ Sınıflandırma, eksik giriş ile sınıflandırma, transcription gibi uygulamalarda **doğruluk (accuracy)** kullanılabilir.

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}}$$

- ▶ Doğruluk değerinden **hata oranı** hesaplanabilir.

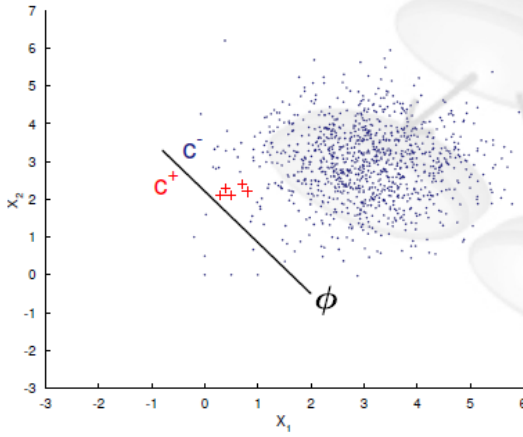
$$error\ rate = 1 - accuracy$$

- ▶ Makine öğrenmesi algoritmalarının **eğitimi**, eğitim verisi ile (**training data**) yapılır.
- ▶ Algoritmaların **performansı**, eğitim süresince görmediği (**test data**) veriler üzerinde ölçülür.

18

## Performans ölçütleri

- **Accuracy** bazı durumlarda kullanılabilir değildir.



		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	0	5	5
	C <sup>-</sup>	7	993	1000
TOPLAM		7	998	1005

$$\text{Accuracy} = (TP+TN)/(N) = (0+993)/1005 = \mathbf{0,99}$$

$$\text{Hata} = (FP+FN)/(N) = (7+5)/1005 = 0,01 \text{ (Çok düşük hata !!!)}$$

19

## Performans ölçütleri

- Makine öğrenme algoritmasının performansını değerlendirmek için **hata matrisi** (confusion matrix) kullanılabilir.
- İki veya daha fazla sınıf için hata matrisi oluşturulabilir.
- "**Tx**" doğruyu, "**Fx**" yanlış göstermektedir.

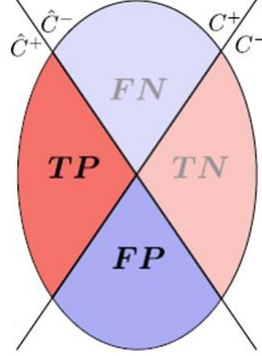
		TAHMİN EDİLEN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	TP True Pozitif (Hits)	FN False Negatif (Miss)	N <sup>+</sup> Gerçek Pozitif sayısı
	C <sup>-</sup>	FP False Pozitif (Miss)	TN True Negatif (Hits)	N <sup>-</sup> Gerçek Negatif sayısı
TOPLAM		N̂ <sup>+</sup> Tahmin Pozitif sayısı	N̂ <sup>-</sup> Tahmin Negatif sayısı	N Toplam Örnek sayısı

		TAHMİN EDİLEN					TOPLAM
		C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	...	C <sub>n</sub>	
GERÇEK	C <sub>1</sub>	T <sub>1</sub>	F <sub>12</sub>	F <sub>13</sub>	...	F <sub>1n</sub>	N <sub>1</sub>
	C <sub>2</sub>	F <sub>21</sub>	T <sub>2</sub>	F <sub>23</sub>	...	F <sub>2n</sub>	N <sub>2</sub>
	C <sub>3</sub>	F <sub>31</sub>	F <sub>32</sub>	T <sub>3</sub>	...	F <sub>3n</sub>	N <sub>3</sub>
	...	...	...	...	...	...	...
	C <sub>n</sub>	F <sub>n1</sub>	F <sub>n2</sub>	F <sub>n3</sub>	...	T <sub>n</sub>	N <sub>n</sub>
TOPLAM		N̂ <sub>1</sub>	N̂ <sub>2</sub>	N̂ <sub>3</sub>	...	N̂ <sub>n</sub>	N Toplam Örnek sayısı

## Performans ölçütleri

- **Precision**, gerçek değeri pozitif olup pozitif değere sınıflandırılan sayısının, pozitif değere sınıflandırılanların toplam sayısına oranıdır.

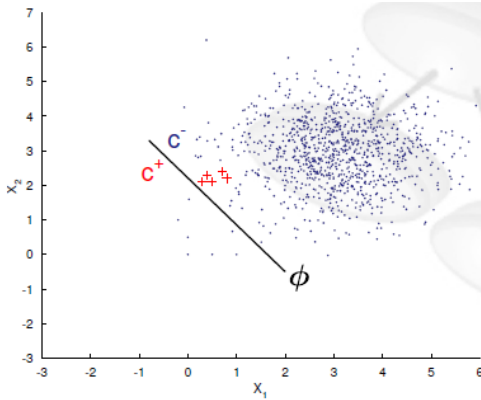
$$\text{Precision} = \frac{TP}{TP + FP}$$



21

## Performans ölçütleri

- Precision, **hedeflenen sınıfın seçilme başarısını ölçmek için daha faydalıdır.**



		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	0	5	5
	C <sup>-</sup>	7	993	1000
TOPLAM		7	998	1005

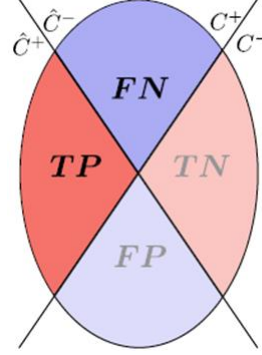
$$\text{Precision} = \frac{TP}{TP + FP} = \frac{0}{0 + 7} = 0 \quad (\text{Çok kötü precision değeri !!!})$$

22

## Performans ölçütleri

- ▶ **Recall**, gerçek değeri pozitif olup pozitif değere sınıflandırılan sayısının, gerçek değeri pozitif olanların tümüne oranıdır.

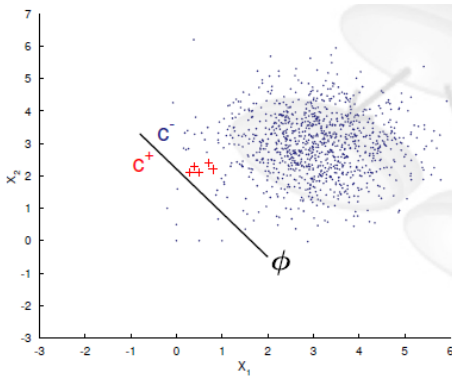
$$\text{Recall} = \frac{TP}{TP + FN}$$



23

## Performans ölçütleri

- ▶ Recall, **hedeflenen sınıfın seçilme başarısını ölçmek için daha faydalıdır.**



		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	0	5	5
	C <sup>-</sup>	7	993	1000
TOPLAM		7	998	1005

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{0}{0 + 5} = 0$$

(Çok kötü recall değeri !!!)

24

## Performans ölçütleri

### Spam filtreleme (spam olanlar hedef sınıf)

- ▶ **Precision:** spam kutusuna alınanlardan gerçekte spam olanların sayısının spam kutusundaki tüm mesajlara oranıdır.
- ▶ **Recall:** gerçekte spam olup da spam kutusuna alınan mesajların, tüm gerçek spam mesajlara oranıdır.

### Duygu analizi (pozitif olanlar hedef sınıf)

- ▶ **Precision:** gerçekte pozitif olup da pozitif sınıflandırılanların sayısının tüm pozitif olarak sınıflandırılanlara oranıdır.
- ▶ **Recall:** gerçekte pozitif olup da pozitif sınıflandırılanların sayısının tüm pozitif olanlara oranıdır.

### Saldırı tespit sisteminde hangisi daha önemlidir?

$$\text{Precision} = \frac{TP}{TP + FP}$$

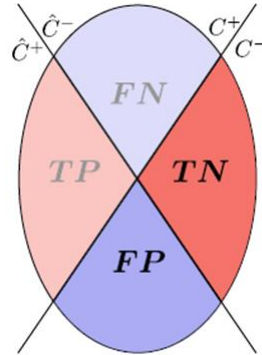
$$\text{Recall} = \frac{TP}{TP + FN}$$

25

## Performans ölçütleri

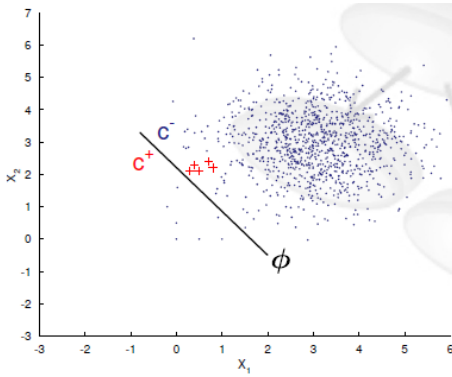
- ▶ **Specificity,** gerçek değeri negatif olup negatif sınıflandırılan sayısının, gerçek değeri negatif olanların tümüne oranıdır.

$$\text{Specificity} = \frac{TN}{TN + FP}$$



26

## Performans ölçütleri

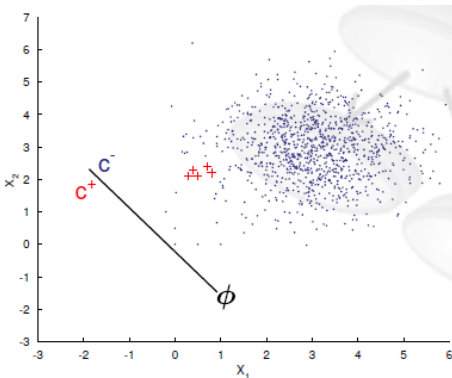


		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	0	5	5
	C <sup>-</sup>	7	993	1000
TOPLAM		7	998	1005

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{993}{993 + 7} = 0,99$$

27

## Performans ölçütleri



		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	0	5	5
	C <sup>-</sup>	0	1000	1000
TOPLAM		0	1005	1005

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{1000}{1000 + 0} = 1,00$$

28

## Performans ölçütleri

- $F_1$ -score (Harmonic mean), iki ölçüt yerine tek ölçüt kullanarak değerlendirme yapmak için kullanılır.

$$F\text{-score} = \frac{(\beta^2 + 1)\text{Precision} \cdot \text{Recall}}{\beta^2(\text{Precision} + \text{Recall})}$$

$$F_1\text{-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

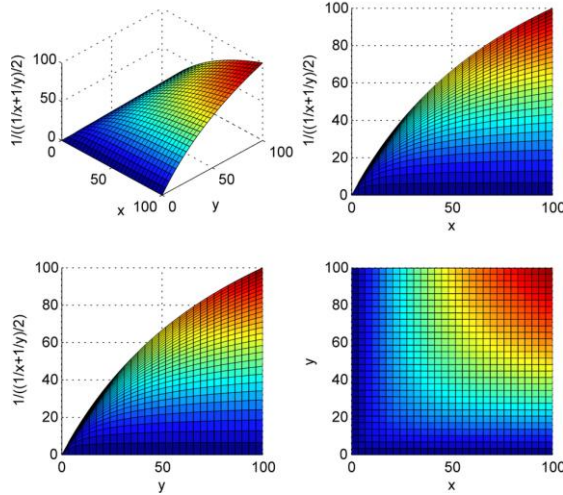
$$x_{\text{harmonic mean}} = \frac{n}{\sum_{k=1}^n \frac{1}{x_k}}$$

$$\begin{aligned} \text{Harmonic mean} &= \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \\ &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

29

## Performans ölçütleri

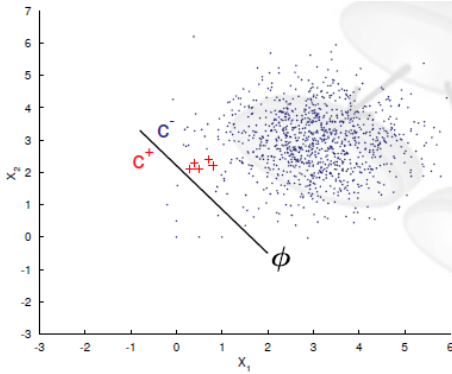
- $F_1$ -score, iki değişkeni tek ölçüt ile değerlendirir.



x (recall) ve y (precision) için  $F_1$ -score değişimi

30

## Performans ölçütleri



		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	0	5	5
	C <sup>-</sup>	7	993	1000
TOPLAM		7	998	1005

$$F_1 \text{ - score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot 0 \cdot 0}{0 + 0} = 0$$

31

## Performans ölçütleri

		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	10	0	10
	C <sup>-</sup>	0	5	5
TOPLAM		10	5	15

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{10}{10 + 0} = 1$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{10}{10 + 0} = 1$$

$$F_1 \text{ - score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot 1 \cdot 1}{1 + 1} = 1$$

32



## Performans ölçütleri

		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	5	5	10
	C <sup>-</sup>	5	5	10
TOPLAM		10	10	20

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{5}{5 + 5} = 0,5$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{5}{5 + 5} = 0,5$$

$$F_1 \text{ - score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot (0,5) \cdot (0,5)}{0,5 + 0,5} = 0,5$$

33

## Performans ölçütleri

		TAHMİN		TOPLAM
		C <sup>+</sup>	C <sup>-</sup>	
GERÇEK	C <sup>+</sup>	5	0	5
	C <sup>-</sup>	15	0	15
TOPLAM		20	0	20

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{5}{5 + 15} = 0,25$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{5}{5 + 0} = 1$$

$$F_1 \text{ - score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot (0,25) \cdot 1}{0,25 + 1} = 0,4$$

34

## İçerik

---

- ▶ Makine öğrenmesi
- ▶ Öğrenme algoritmaları
- ▶ Uygulamalar
- ▶ Performans ölçütleri
- ▶ Öğrenme türleri
- ▶ Kapasite, overfitting ve underfitting
- ▶ Hyperparameters ve validation kümesi

35

## Öğrenme türleri

---

### Denetimli öğrenme

- ▶ Denetimli öğrenme (**supervised learning**), makine öğrenmesinde **sınıflandırma** veya **tümevarımlı (inductive)** öğrenme şeklinde ifade edilir.
- ▶ Denetimli öğrenmede **hedef değerler (targets)** ile **giriş değerleri (inputs)** birlikte **eğitim kümesi (training set)** olarak sağlanır.
- ▶ Öğrenme işleminde bir kayıt kümesi kullanılır ve özellikler kümesi olarak gösterilir.

$$A = \{A_1, A_2, \dots, A_{|A|}\}$$

- ▶ Burada,  $|A|$  kümedeki eleman sayısını gösterir.

36

## Öğrenme türleri

### Denetimli öğrenme

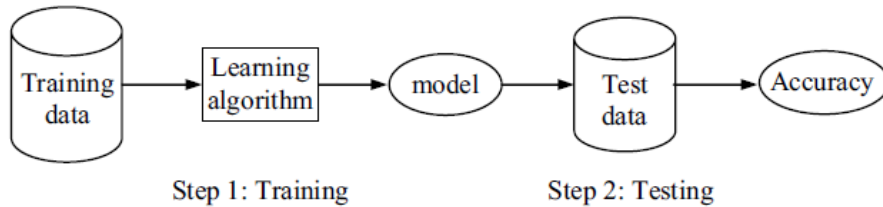
- ▶ Bir veri kümesi aynı zamanda hedef  $C$  özelliğine de (sınıf) sahip olabilir.
- ▶  $C \cap A = \emptyset$  dir ve aşağıdaki gibi ifade edilir:  
$$C = \{c_1, c_2, \dots, c_{|C|}\}, |C| \geq 2$$
- ▶ Verilen bir  $D$  veri kümesi için öğrenmedeki amaç,  **$A$ 'daki özellikler ile  $C$ 'deki sınıflar arasındaki ilişkiyi gösteren bir sınıflandırma/tahmin fonksiyonu** oluşturmaktır.
- ▶ Elde edilen bu fonksiyon, **sınıflandırma modeli**, **tahmin modeli** veya **sınıflandırıcı** olarak adlandırılır.

37

## Öğrenme türleri

### Denetimli öğrenme

- ▶ Öğrenme süreci training ve test aşamalarından oluşur.
- ▶ Modelde **hiper parametreler** varsa, **öğrenme aşamasından önce optimize edilirler.**



38

## Öğrenme türleri

---

### Denetimsiz öğrenme

- ▶ Denetimli öğrenme giriş verileri ile çıkış niteliği arasındaki ilişkiyi ortaya çıkartır.
- ▶ Elde edilen model ile yeni verilerle ileriye dönük tahmin yapılması amaçlanmaktadır.
- ▶ **Denetimsiz öğrenmede** eğitim sürecinde **hedef nitelik bulunmamaktadır.**
- ▶ Denetimsiz öğrenmede **veriler arasında bazı yapısal ilişkilerin** veya **örüntülerin ortaya çıkartılması** amaçlanmaktadır.

39

## İçerik

---

- ▶ Makine öğrenmesi
- ▶ Öğrenme algoritmaları
- ▶ Uygulamalar
- ▶ Performans ölçütleri
- ▶ Öğrenme türleri
- ▶ **Kapasite, overfitting ve underfitting**
- ▶ Hyperparameters ve validation kümesi

40

## Kapasite, overfitting ve underfitting

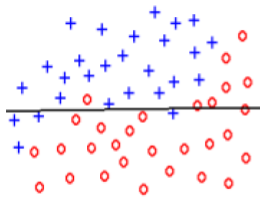
- ▶ Makine öğrenmesi algoritmasının **önceden görülmemiş girişler üzerinde başarılı olması gereklidir.**
- ▶ Önceden görülmemiş veriler üzerindeki performansa **genelleme (generalization)** denir.
- ▶ Makine öğrenmesi algoritmaları **eğitim sürecinde eğitim hatası hesaplar.**
- ▶ Eğitim aşamasında sonra, **unseen data** ile **test hatası (generalization/test error)** hesaplanır.
- ▶ İstatistiksel öğrenme teorisine göre, **test ve eğitim kümeleri uygun toplanmışsa, eğitim ve test hataları birbirine yakın** olur.

41

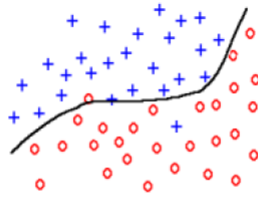
## Kapasite, overfitting ve underfitting

### Underfit, fit ve overfit

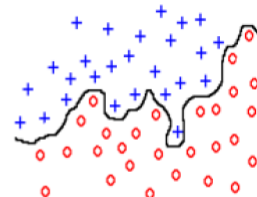
- ▶ **Underfit**, eğitim kümesi için başarı düşüktür (eğitim hatası yüksektir).
- ▶ **Overfit**, eğitim kümesi için başarı yüksektir, ancak test kümesi için başarı düşüktür (eğitim hatası düşük, test hatası yüksektir).
- ▶ **Fit**, hem eğitim kümesi hem de test kümesi için başarı yüksektir (eğitim ve test hatası düşüktür).



underfit



fit



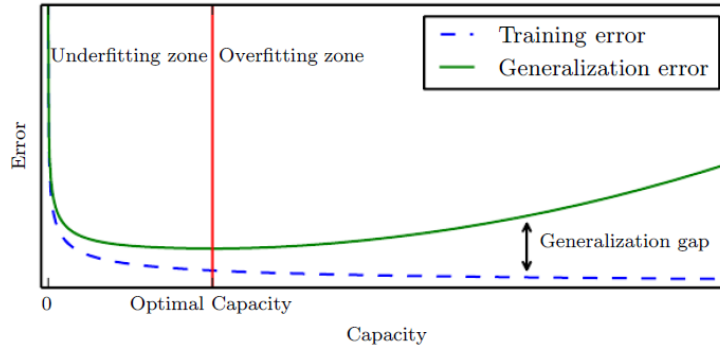
overfit

42

## Kapasite, overfitting ve underfitting

### Underfit, fit ve overfit

- ▶ Modelin eğitimini **optimal kapasitenin** olduğu noktada tamamlamak gerekir.
- ▶ **Generalization error minimum** noktada iken sistemin **genelleme özelliği maksimum** düzeydedir.



43

## İçerik

- ▶ Makine öğrenmesi
- ▶ Öğrenme algoritmaları
- ▶ Uygulamalar
- ▶ Performans ölçütleri
- ▶ Öğrenme türleri
- ▶ Kapasite, overfitting ve underfitting
- ▶ **Hyperparameters ve validation kümesi**

44

## Hyperparameters ve validation kümesi

- ▶ Çoğu makine öğrenmesi algoritması, öğrenme algoritmasının davranışını kontrol etmek için farklı ayarlara/parametrelere sahiptir (**öğrenme hızı**, derin ağda **gizli katman sayısı**, **k-means için k değeri**, ...).
- ▶ Bu ayarlar, **hiper parametre** (**hyperparameters**) olarak adlandırılır.
- ▶ Hiper parametrelerin değerleri, **eğitim aşamasından önce belirlenmelidir**.
- ▶ Hiper parametrelerin set edilmesi aşamasında, eğitim algoritmasının kullanmayacağı doğrulama verisi (validation data) kullanılabilir.
- ▶ **Validation data** genellikle **eğitim verisinden oluşturulur**.
- ▶ **Validation data** içerisinde **test verisi kullanılmaz**.
- ▶ Genellikle, **eğitim verisinin %80'i eğitim, %20'si validation** için kullanılır.