Machine Learning (ML)

Machine learning had grown up as a sub-field of AI or artifical intelligence.

- machine learn to do it by itself

Two main types of ML are 1- Supervised learning
2- Unsupervised learning

* Supervised learning is used most in many real-world applications and most rapid advancements and innovation

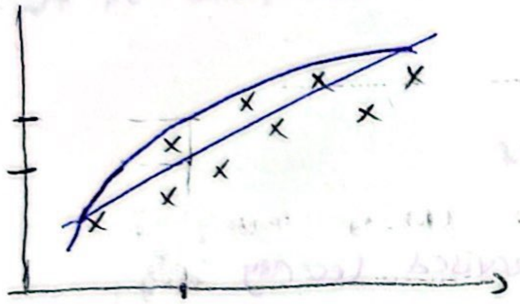
Supervised learning

X input ⟶ Y output label

* Learns from being given "right answer" → labeled data

input (x)	output (y)	Application
email	spam? (0/1)	spam filtering
audio	text transcripts	speech recognition
English	Spanish	machine translation
ad, user info	click? (0/1)	online advertising

• **Regression**: Predict a number <u>infinitely</u> many possible outputs

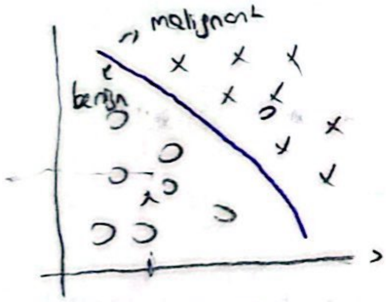


• Bir çizgiye ya da eğriye göre tahminde bulunulur.

• Verilere uyacak en uygun çizgiyi seçerek algoritma elde etmek

• **Classification**: Predict categories small number of possible outputs.

• Only two values 0 or 1 → tümör iyi huylu, kötü huylu

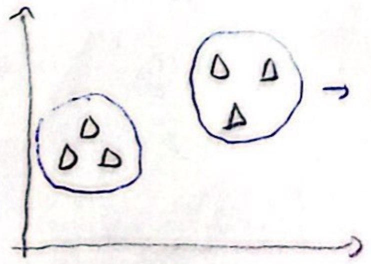


→ Sınıflandırmanın regresyondan farkı: sınıflandırmanın 0,1,2 gibi küçük sınırlı olası çıktı olması dır ancak 0.5, 1.7 gibi aradaki tüm olası sayıları saymamasıdır

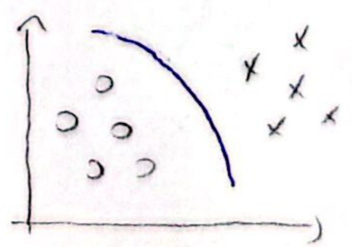
Unsupervised Learning

Find something interesting in unlabeled data.

- An unsupervised learning algorithm → two different groups or two different clusters



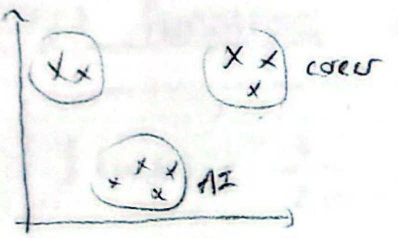
Unsupervised



Supervised

* Data only comes with inputs x, but not output labels y.
* Algorithm has to find <u>structure</u> in the data

1- Clustering: Group similar data points together.



→ takes data without labels and tries to automatically group them into clusters.

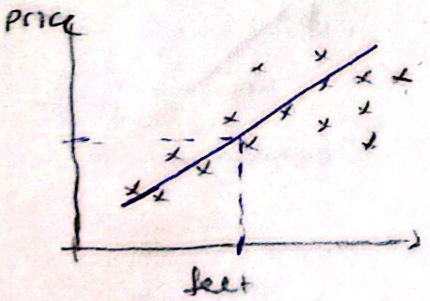
2- Anomaly detection; Find unusual data points.

→ Finansal sistemde dolandırıcılık tespiti.

3- Dimensionality reduction: Compress data using fewer numbers.

- Take big dataset and compress it to a much smaller data-set while losing as little information as possible.

Linear Regression



Girilen değere göre o büyüklükteki evin fiyatı tahmin edilebilir.

Supervised learning model, data has "right answers".

Training set: ev fiyatını öğrenebilmek için, modeli training set ile eğitilir böylece model müşterinin evini tahmin eder.

	x	y
1)	2104	400
⋮		
47)	3210	870

$x^{(1)} = 2104$ → input $y^{(1)} = 400$ → output

$m = 47$ → number of training example

$(x^{(1)}, y^{(1)}) = (2104, 400)$

Scanned with CamScanner

training set — features (x), targets (y)

↓

learning algorithms

↓

$x \rightarrow \boxed{f} \rightarrow \hat{y}$ (tahmin, yaklaşık değer)

feature → model → prediction (estimated y)

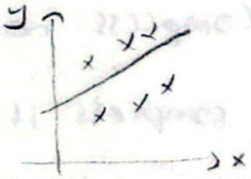
ex için

size → $\boxed{f}$ → price

How to represent f ?

$$f_{w,b}(x) = wx + b = \hat{y}$$

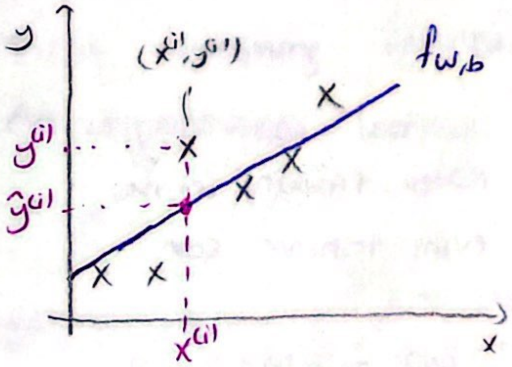
$$f(x) = wx + b$$



linear function

→ linear fonk. daha basit ve kolay olduğu için bu kullanılır.

* Linear regression with one variable = Univariate linear regression



$$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$$

$$f_{w,b}(x^{(i)}) = wx^{(i)} + b$$

cost function: Squared error cost function

$$\boxed{J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2}$$

$\hat{y}^{(i)} - y^{(i)} = \text{error}$

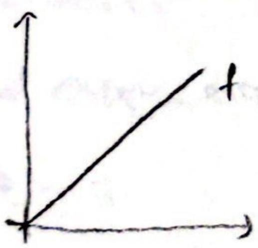
m= number of traning examples

w, b → parameters

Cost function intuition (sezgi)

$\underset{w,b}{\text{minimize}}\ J(w,b)$

Simplified $f_w(x) = wx$, $b=0$



$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} (\underbrace{f_w(x^{(i)})}_{wx^{(i)}} - y^{(i)})^2$$

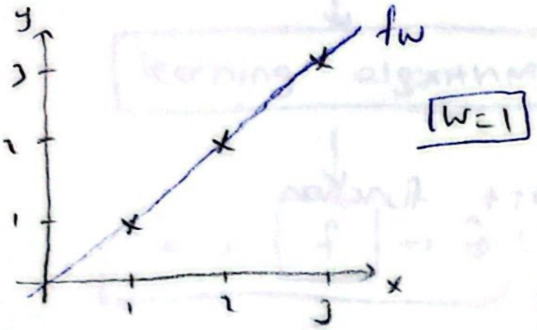
$\underset{w}{\text{minimize}}\ J(w)$

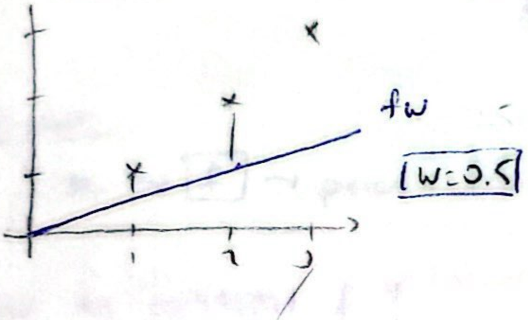
- goal of linear regression; $\underset{w}{\text{minimize}}\ J(w)$

- general case: $\underset{w,b}{\text{minimize}}\ J(w,b)$

$f_w(x)$

• w: fixed, function of x





$J(w)$:

• function of w

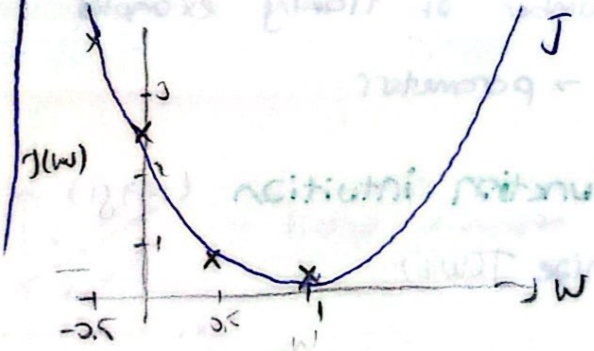
$$J(w)=\frac{1}{2m}\sum_{i=1}^{m}\left(f_w(x^{(i)})-y^{(i)}\right)^2$$
$$=\frac{1}{2m}\sum_{i=1}^{m}\left(wx^{i}-y^{i}\right)^2$$

• $J(1)=0$

• $J(0.5)=\frac{1}{2.3}\left[(0.5-1)^2+(1-2)^2+(1.5-3)^2\right]\approx 0.5$

• $J(0)=\frac{1}{2.3}\left[1^2+2^2+3^2\right]\approx 2.3$

• $J(-0.5)\approx 5.25$

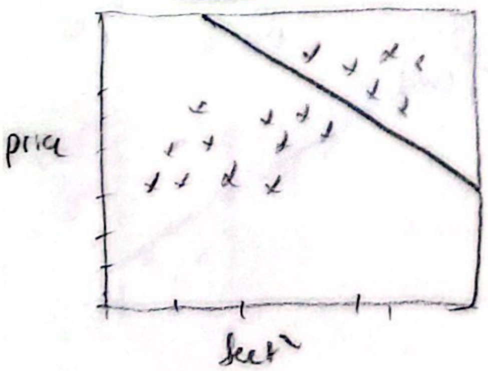


→ Choose w to minimize J(w)

w=1 ✓

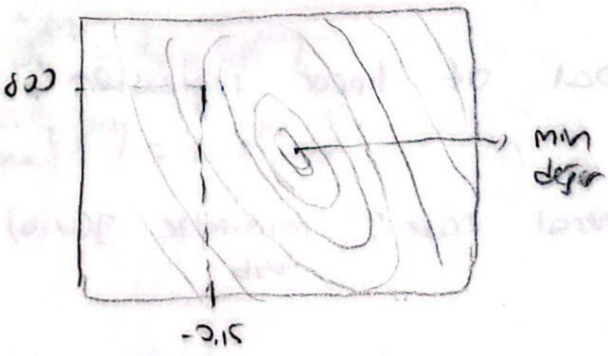
* J değerlerini küçülten w değerini seçerek lineer regresyona uygun bir cost fonksiyonu kullanmış oluruz.

$f_{w,b}$



$f(x) = -0.15x + 800$

J



– Dereceli azalma algoritması

Gradient Descent

$J(w,b)$ → küçültmek istediğimiz cost fonk.

↳ for linear regression or any function

$$\min_{w_1, \dots w_n, b} J(w_1, w_2, \dots, w_n, b)$$

Outline:

- Start with some w,b (set w=0, b=0)
- keep changing w,b to reduce J(w,b)
- Until we settle at or near a minimum

→ cost fnk may have more than 1 min

yokuş aşağı iniş

Gradient descent algorithm

$$w = w - \alpha \frac{d}{dw} J(w,b)$$

$$b = b - \alpha \frac{d}{db} J(w,b)$$

α → Learning rate
$(0<\alpha<1)$

→ yokuş aşağı ne kadar büyük adım atacağımızı kontrol eder.

- α büyük, adımlar büyük.

→ Repeat until convergence.

→ Simultaneously update w and b.

Simultaneous update:

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w,b)$$

$$tmp\_b = b - \alpha \cdot \frac{\partial}{\partial w} J(w,b)$$

$$w = tmp\_w$$

$$b = tmp\_b$$

Scanned with CamScanner

$$w = w - a \frac{\partial}{\partial w} J(w,b)$$

$$b = b - a . \frac{\partial}{\partial w} J(w,b)$$

- $\frac{\partial}{\partial w} J(w,b)$ → derivative, yön belirtir.

- $\boxed{a \rightarrow \text{learning rate}}$

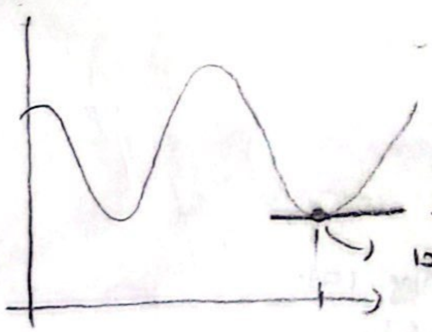
- a very small → adım da çok küçük.

→ Gradient descent be slow

- a too large → adım çok büyük

→ Gradient descent may overshoot, never reach minimum.

→ Fail to converge, diverge



$$w = w - a . \underbrace{\frac{d}{dw} J(w)}_{0}$$

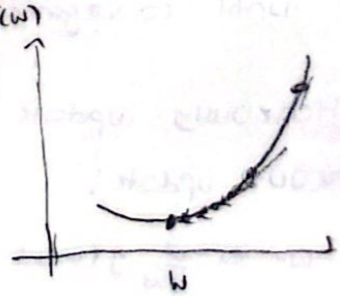
$w = w$

→ local min noktasındaysanız derivative 0 olur w değerini değiştirmez.

* Near a local min,

- Derivative becomes smaller.
- Update steps become smaller

Can reach minimum without decreasing learning rate.



- Linear regression

$f_{w,b}(x) = wx + b$

- cost function

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})^2$$

- Gradient descent algorithm.

$$w = w - a \boxed{\frac{\partial}{\partial w} J(w,b)} \longrightarrow \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)}) . x^{i}$$

$$b = b - a \boxed{\frac{\partial}{\partial b} J(w,b)} \longrightarrow \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})$$

"Batch" gradient descent

Batch: Each step of gradient descent uses all the training examples m

$$\sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})^2 \qquad m=47$$

Scanned with CamScanner