

→ Supervised machine Learning Gözetimli makine öğrenmesi

$x \rightarrow y$
input \rightarrow output

> Çakışma algoritması

* (right answer is correct label y for given input x)

Ex	Input (x)	Output (y)	Application
	email	spam or not	spam filtering
	audio	text transcripts	speech recognition
	ad, user info	click or not	online advertising
	image, radar info	position of other cars	self-driving car

Regression - regresyon \rightarrow değişkenler arasındaki ilişkiyi tahmin etmek için kullanılan istatistiksel bir teknik.

1) Regression algorithms, which is a type of supervised learning algorithm learns to predict numbers out of infinitely many possible numbers.

2) Classification algorithm

class or category when referring to the output, it means something.

classification algorithms predict categories. (sayılar, sınıflar, kategoriler)

* small numbers of possible outputs.

* Öğüt, bağımlı değişken ve bağımsız değişken yani girdi ve çıktı bir aradaysa buva gözetimli (supervised) öğrenme denir. Bu öğrenme tipinde makineyi "etiketlenmiş" verileri kullanarak eğitirsiniz ve bazı verilerin zaten doğru yanıtla eşleştirildiği anlamına gelir.

Up supervised Learning

* Gözetimsiz öğrenmede çıktıları öğrenmenin içinde bulunmaz. Örneklerden birimler benzer özelliklerine göre bir araya getirilir. Gözetimsiz öğrenme, etiketlenmemiş veri kitlelerini analiz etmek ve kümelemek için yapıy öğrenme algoritmalarını kullanır.

1) Cluster algorithm (kümeleme algoritması)

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

* Group similar data points together.

* In unsupervised learning, data only comes with inputs x , but not output labels y . Algorithm has to find structure in the data.

2) Anomaly detection (kusur/kata saptama)

Find unusual data points.

3) Dimensionality reduction (boyut azaltma)

Compress data using fewer numbers.

Not: Kısa, gözetimli öğrenme modelleri gözetimsiz öğrenme modellerinden daha doğru olma eğilimindedir, verileri uygun şekilde etiketlemek için önceden insan müdahalesi gerektirir.

Gözetimsiz öğrenme modelleri olarak:

- Spam algılama
- duygu analizi
- hava durumu tahmini
- fiyatlandırma tahmini

→ shift + enter
kütüphane çağırarak
için.

↓
← + colunlar X
(input) (output) (output)

Gözetimsiz (denetimsiz) öğrenme modelleri olarak:

- Öneri motorları
- müşteri kısıtlıdır
- tıbbi görüntüleme

Jupyter Notebook

Jupyter notebook, dilleri programlara dilin için etkilidir bir ortam

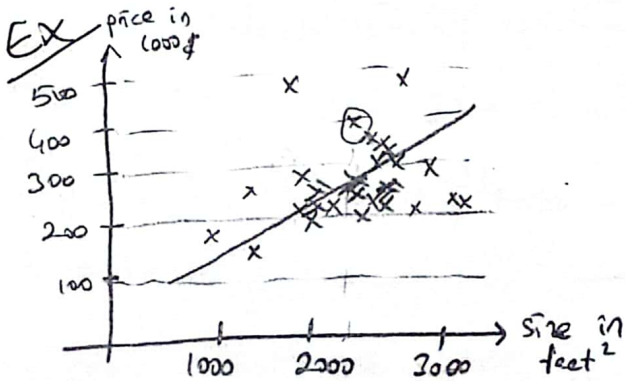
Saklayan açık kaynak kodlu bir programdır

Örneğin, word içerisinde python kodları çalıştırabilirsiniz ve markdown destekli notlar alıp bir alt satırda da notları öğrenmesi kodlarını çalıştırabilirsiniz

* Linear regression model

* Linear regression, denetimli öğrenme modellerinde girdi X ile çıktı Y arasındaki en uygunluğa sahip bir regresyon modeli algoritmasıdır

Bir veya birden fazla girdi ile çıktı arasındaki istatistiksel ilişkiyi tanımlamak için bir denklem oluşturur. ve bu oluşturulan denkleme göre de bir lineer doğru çıkar.



$$X: 2104 \rightarrow Y: 400$$

$$(X, Y) = (2104, 400)$$

Data table		Data used to train the model
Training set		
	Size in feet²	price in \$ 1000's
X=input Variable	2104	400
	1416	232
	1534	315
	852	178

	3210	370
		Y=output (target) Variable

m = 47 satır (number of training examples)

$(x^{(i)}, y^{(i)}) = i^{th}$ training example
(1., 2., 3.)

$$\underline{x^{(2)} \neq x^2}$$

Training set \rightarrow features

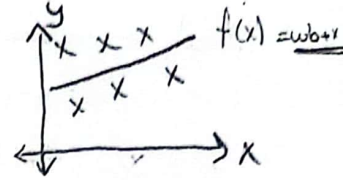
\rightarrow targets \rightarrow The output targets are the right answers to the model.

Learning algorithm

$X \rightarrow$ function $f \rightarrow \hat{y}$ (y-hat) (y için bir tahmin)
(feature) (model) (prediction)
(estimated y(target))

* size $\rightarrow [f] \rightarrow$ price (estimated)

$$f = f_{w,b}(x) = wx + b \quad (w \text{ and } b; \hat{y}) \rightarrow f(x)$$



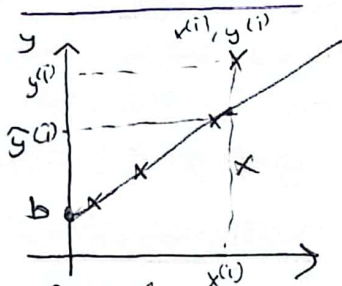
(Linear regression with one variable)
Univariate linear regression

$$y = b_0 + b_1 \cdot x \quad \rightarrow \quad b_0 = \text{sabit katsayı değeri}$$

$b_1 = \text{girdi katsayısı}$

$x = \text{girdi}$

Cost function (maliyet fonk.)



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

$$\frac{1}{2m} \sum_{i=1}^m (\underbrace{\hat{y}^{(i)} - y^{(i)}}_{\text{error}})^2 = \text{Cost function} = J(w, b)$$

(squared error cost function)

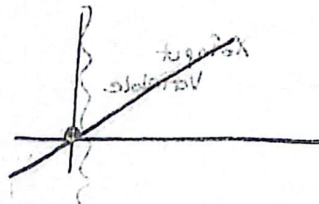
$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m (f_{w,b}(x^{(i)}) - y^{(i)})^2$$

* Bu fonksiyonda sadece w ve b ayarlanabilen parametrelerdir

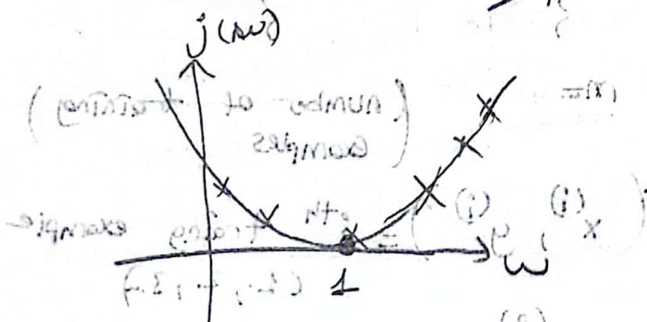
Simplified function

$$f_w(x) = w \cdot x$$

($b=0$ kabul edilir)



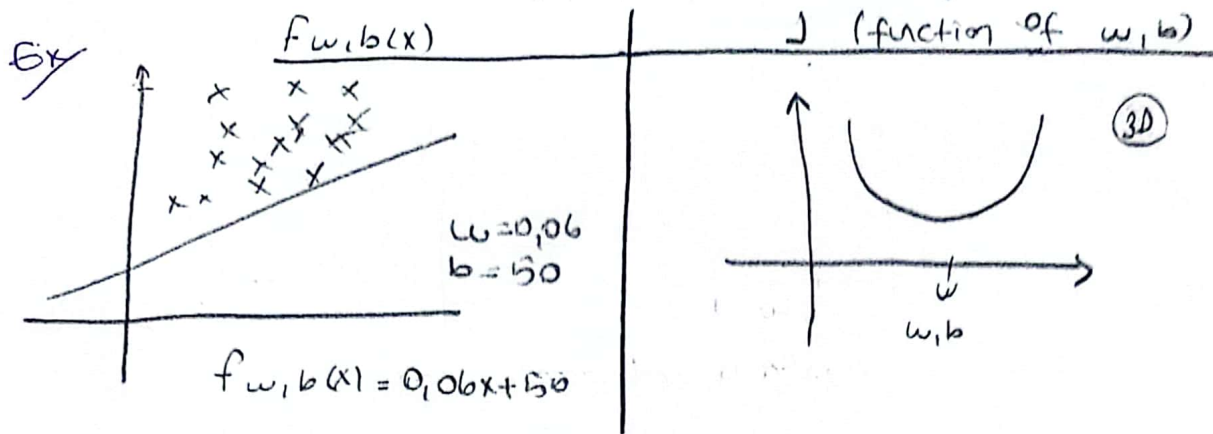
$$J(w) = \frac{1}{2m} \sum_{i=1}^m (f_w(x^{(i)}) - y^{(i)})^2$$



$w=1$ en uygun değer

* Maliyet J nispeten küçük, Sıfıra yakın olduğunda, modelin w ve b değerleri seçeneğe uygun verilerle daha iyi uyduğunu gösterir

* The goal of linear regression is to find the parameters w and b that results in the smallest possible value for the cost function.



Gradient descent (gradyan inisi)

Gradyen inisi metodu, çok güçlü ve çok genel optimizasyon metodudur. Gradyen inisi, makine öğrenme alanında popüler bir yöntemdir, çünkü makine öğrenmenin amaçlarından biri, eğitim verisi göz önüne alındığında, en yüksek doğruluğu bulmak veya hata oranını en aza indirmektir.

* Gradyen inisi malîyet fonksiyonunu en aza indirgeyerek asgari hatayı bulmak için kullanılır.

* J 'nin değerleri sürekli azalsın. Çünkü $J \geq 0$ olmalı.

→ have some function $J(w, b)$

want $\min_{w, b} J(w, b)$

outline! Start with some w, b

keep changing w, b to reduce $J(w, b)$

until we settle at or near a minimum.

Algorithm: $w \leftarrow w - \alpha \frac{\partial}{\partial w} J(w, b)$

yaklaşık olarak
tekrarlanabilir

learning rate \rightarrow 0 ile 1 arasında küçük bir sayı

$$b \leftarrow b - \alpha \frac{\partial}{\partial b} J(w, b)$$

Correct simultaneous update

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

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

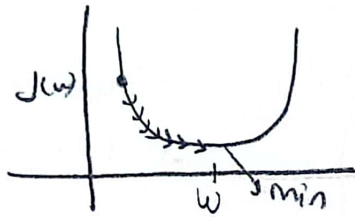
$$w = \text{tmp-}w$$

$$b = \text{tmp-}b$$

* Gradyen inisi, J malîyet fonk. minimize eden w ve b parametrelerinin değerlerini bulmak için kullanılan bir algoritmadır.

α : öğrenme oranı

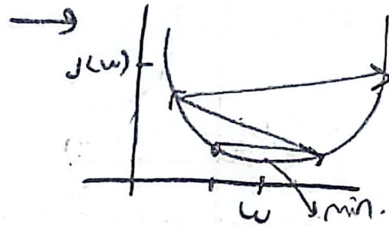
α is too small \rightarrow



Gradient descent may be slow.

Minimuma yaklaşıp
(yavaş yaklaşıp)

α is too large
(Tleni gñ yaklaşıp)



Minimumdan uzaklaş



★ Can reach local min with fixed learning rate

$$w = w - \alpha \cdot \frac{\partial}{\partial w} J(w)$$

(gini aybana) trajektör

Near a local min,

- derivative become smaller
- update steps become smaller

Can reach min. without decreasing learning rate α .

$$\begin{aligned} w &= w - \alpha \cdot \frac{\partial}{\partial w} J(w, b) \rightarrow \frac{1}{M} \sum_{i=1}^M (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)} \\ b &= b - \alpha \cdot \frac{\partial}{\partial b} J(w, b) \rightarrow \frac{1}{M} \sum_{i=1}^M (f_{w,b}(x^{(i)}) - y^{(i)}) \end{aligned}$$

★ If we are using a squared error cost function with linear regression the cost function doesn't and will never have multiple local minima. It has a single global minimum because of this bowl-shape. This cost function is a convex function.

"Batch" gradient descent

Each step of gradient descent uses all the training examples.