MACHINE LEARNING BOOTCAMP

GDG Algiers

Women Techmakers Algiers

Hello there,

I am Nour Fadila Zemouri

Master 2 degree student at University of Algiers 1 - Benyousef Benkheda -

Field of study : ARTIFICIAL INTELLIGENCE

What we will see ?

1. What is supervised Machine Learning
2. Some supervised Machine Learning Examples
3. Some common supervised Machine Learning Algorithms
4. Challenge

What we will see ?

1. What is supervised Machine Learning
2. Some supervised Machine Learning Examples
3. Some common supervised Machine Learning Algorithms
4. Challenge

Before we start what about a quick Recall?

Machine Learning?

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959

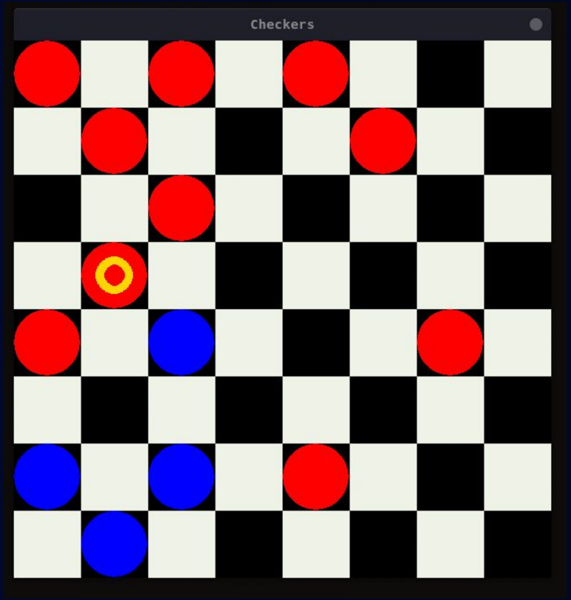
Let make it more engineer oriented

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.
—Tom Mitchell, 1997

Let put it in a example

Let consider a checkers playing:

- The experience E : would be the experience of having the program play tens of thousands of games itself.
- The task T : would be the task of playing checkers,
- The performance measure P : will be the probability that wins the next game of checkers against some new opponent.



Machine Learning systems

Machine Learning

- **Supervised Machine Learning**: Teach the computer how to do something
- **Unsupervised Machine Learning**: Let the computer learn by itself
- **Reinforcement Learning**: How intelligent agents must act in an environment in order to maximize the notion of cumulative reward

Machine Learning systems

- Machine Learning
  - Supervised Machine Learning
  - Unsupervised Machine Learning
  - Reinforcement Learning

Our focus !!

Supervised Machine Learning

Let’s define it

We provides the algorithm with pairs of inputs and desired outputs (target).

Serial	X1	X2	Y
1	96	1989	no
2	128	1986	yes
3	81	1996	no

That way when the algorithm is facing a new input he will find the way to produce the desired output

Supervised Machine Learning Types

Based on the type of our output (target) Supervised Machine Learning can be categorized into two.

Supervised Machine Learning

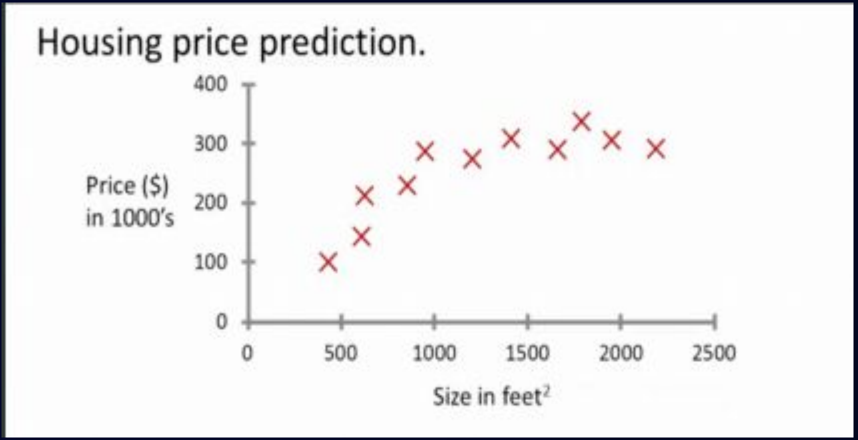
- **Classification** — Target values are discrete classes
- **Regression** — Target values are continuous values

Regression

Let us take an example:

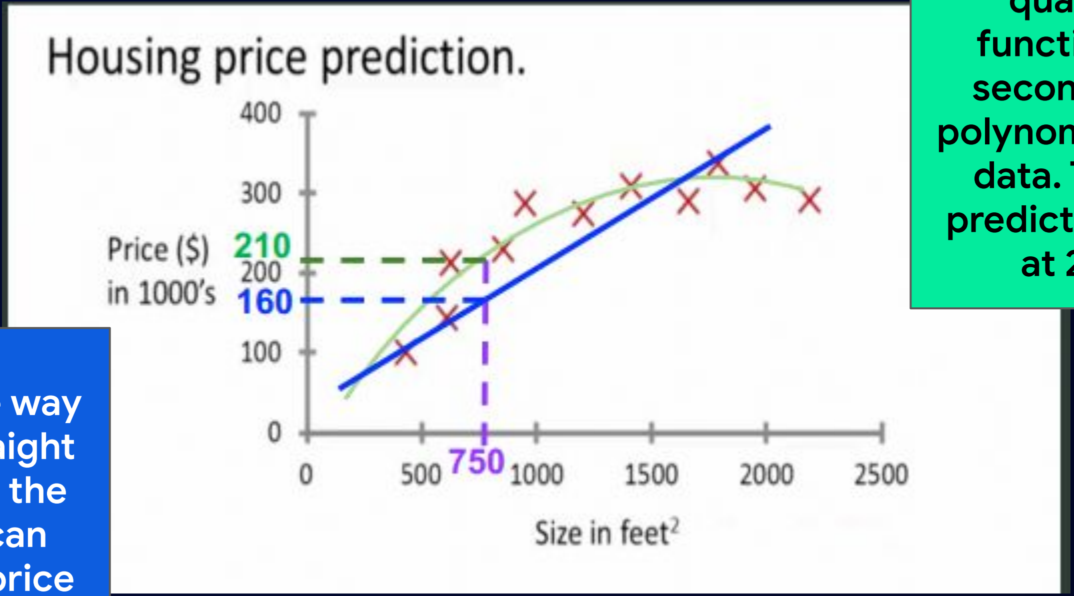
- You are given a plotting data for some houses.

- Given this data, let's say you have a friend who owns a house that is say 750 square feet, and they are hoping to sell the house, and they want to know how much they can get for the house.



Notice that the learning algorithm is trying to predict a "continuous valued output"

Regression



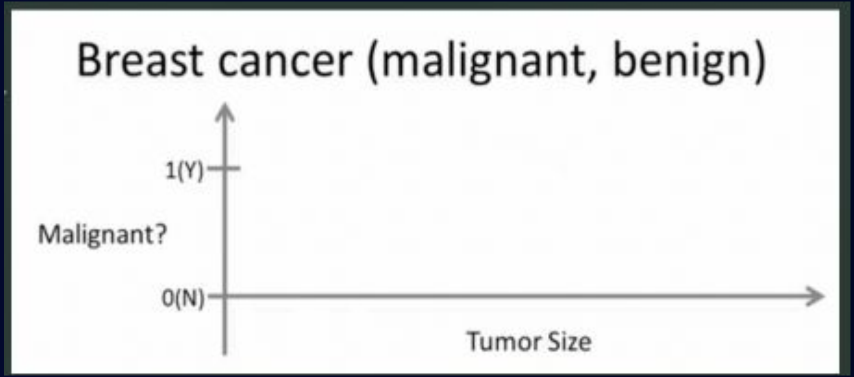
Another possible is to fit a quadratic function, or a second-order polynomial to this data. This can predict the price at 210k$

One possible way is to fit a straight line through the data. This can predict the price at 160k$

Classification

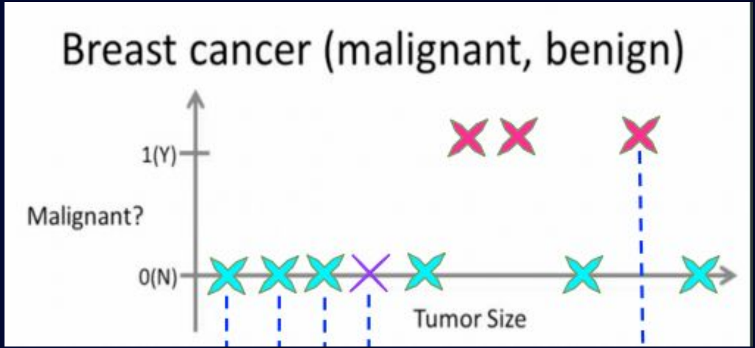
Let see an example in this case:

- Doctors look at medical records and try to predict of a breast cancer as malignant (harmful) or benign (harmless).
- Suppose that you have a set of data collected. The examples of tumors we are seeing are malignant, which is one, or benign , which is zero.



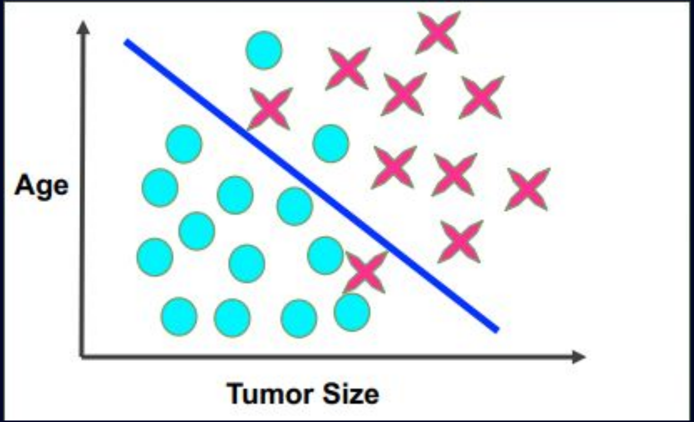
Notice that the learning algorithm is trying to predict a "discrete valued output (0 or 1) "

Classification



One feature (tumor size)

Two features (tumor size and age)



Classification Types

Classification

- **Binary classification**: refers to predicting one of two classe
- **Multi-class classification**: involves predicting one of more than two classes.
- **Multi-label classification**: involves predicting one or more classes for each example

What we will see ?

1. What is supervised Machine Learning
2. Some supervised Machine Learning Examples
3. Some common supervised Machine Learning Algorithms
4. Challenge

	Classification	Regression	Output (target)
House pricing	✗	✓	• The price (continuous value)
How is the weather forecasting?	✗	✓	• The forecast (continuous value)
Is the Email a spam	✓	✗	• Two classes : Spam or not a spam
Is it a cat or a dog?	✓	✗	• Two classes : cat or dog
Salary of a person	✗	✓	• The salary (continuous value)

What we will see ?

1. What is supervised Machine Learning
2. Some supervised Machine Learning Examples
3. Some common supervised Machine Learning Algorithms
4. Challenge

Data Split

(n > 1,000 instances)

Train-Test Split

Training dataset: Used to train our model

Test dataset: Used to evaluate the trained model.

Common split percentages include:
Train: 50%, Test: 50%
Train: 70%, Test: 30% (n < 10,000)
Train: 80%, Test: 20%

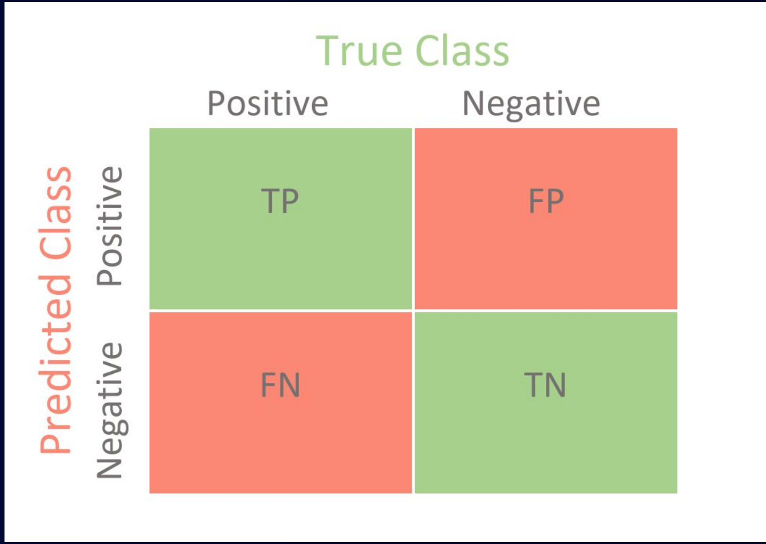
Train-Validation-Test Split

Training dataset: Used to train a few candidate models

Validation dataset: Used to evaluate the candidate models

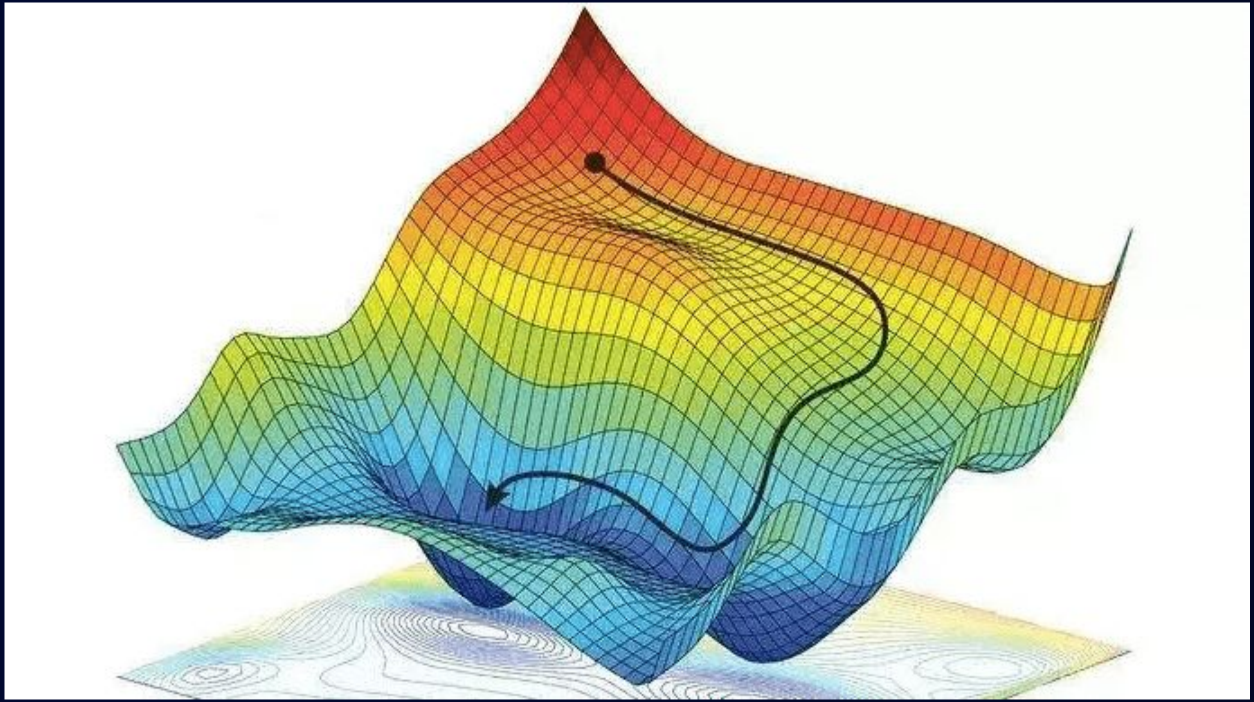
Test dataset: Used to evaluate the candidate model.

Confusion Matrix



A Confusion Matrix is a tool to measure the performance of a Machine Learning model by checking how often its predictions are accurate in relation to reality in classification problems.

You should also know Gradient Descent



What is Gradient Descent ?

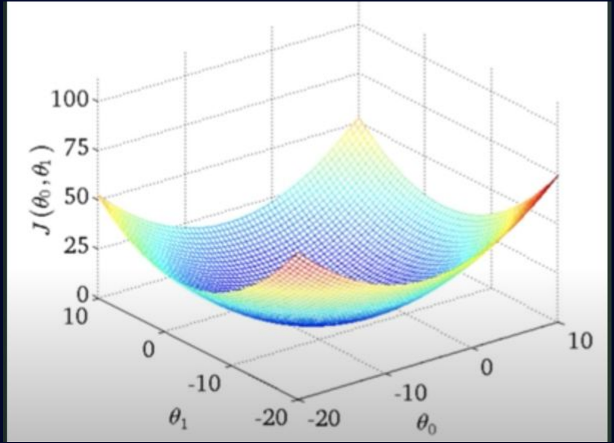
Gradient descent is a popular optimization algorithm that is used when training a machine learning model.

Based on a convex function, the algorithm tweaks its parameters iteratively to minimize a given function called cost function.

What is Convex function ?

Informally a convex function means a bowl shaped function.

This function doesn't have any local optima except for the one global optimum.



What is Learning rate ?

The learning rate (α) controls how big a step is taken downhill with creating descent.

- If α isvery large, then that corresponds to a very aggressive gradient descent procedure where a huge steps are taken downhill
- If α is very small, then little baby stepsare taken downhill.

Gradient Descent Types

Gradient Descent

Batch Gradient Descent

Batch gradient descent calculates the error for each example within the training dataset, but only after all training examples have been evaluated does the model get updated

Stochastic Gradient Descent

stochastic gradient descent (SGD) updates the parameters for each training example one by one.

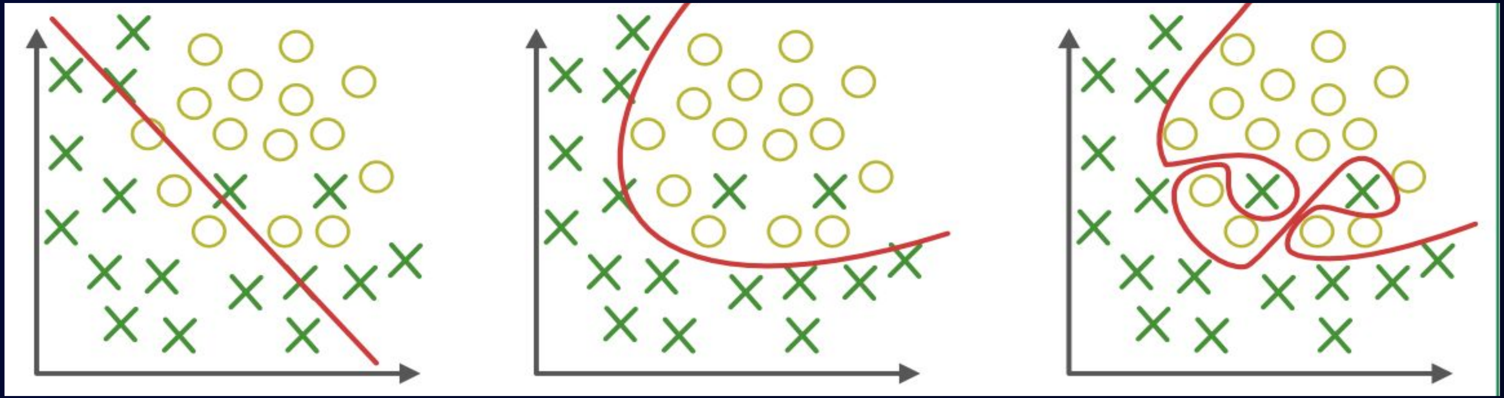
Mini-Batch Gradient Descent

Mini-batch gradient descent is a combination of the concepts of SGD and batch gradient descent. It simply splits the training dataset into small batches (best are 50 and 256) and performs an update for each of those batches.

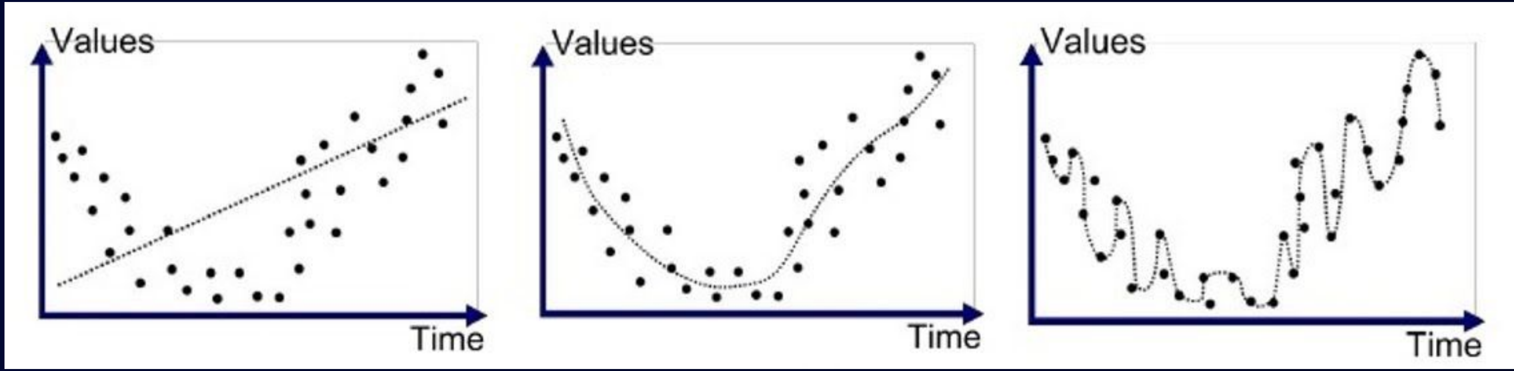
Some common supervised Machine Learning Algorithms

k-Nearest Neighbors
Linear Regression
Logistic Regression
Support Vector Machines (SVMs)
Decision Trees and Random Forests
Neural networks

But before we start learning about Algorithms, Take a look at this classification examples !!

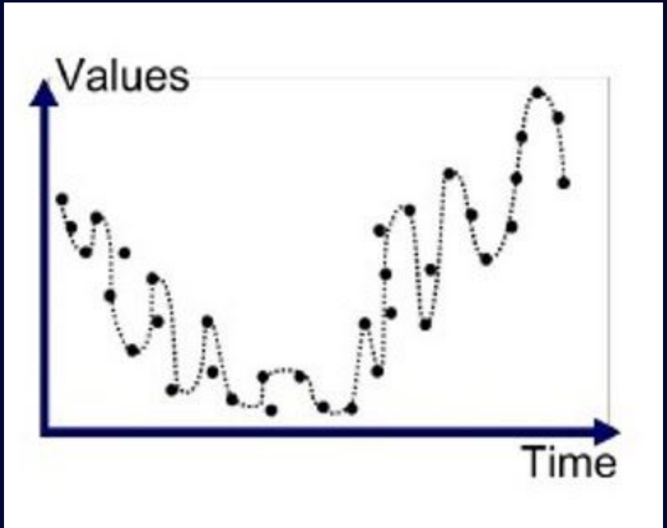


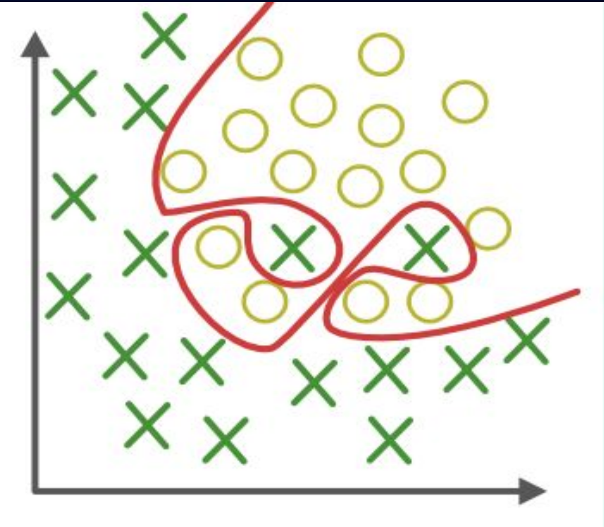
Now at this regression examples !!



Did you Notice something ?

Let’s take a closer look





Over-fitting

What is
Over-fitting ?

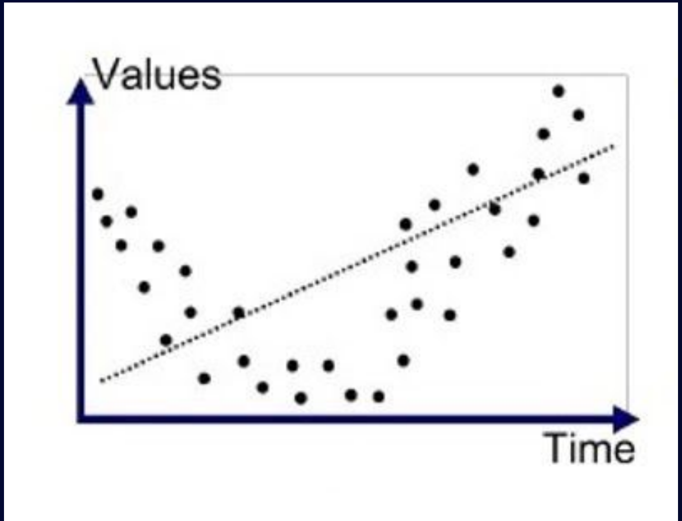
When a model performs very well for training data but has poor performance with test data (new data), it is known as overfitting.

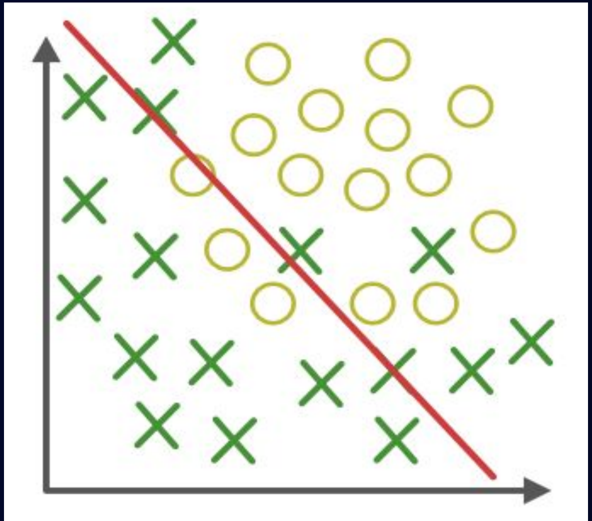
Reasons for Over-fitting ?

- Data used for training is not cleaned and contains noise in it.
- Due to low bias and high variance.
- The size of the training dataset used is not enough
- The model is too complex

Ways to deal with Over-fitting ?

- Using Regularization techniques
- Using K-fold cross-validation
- Training model with sufficient data





Under-fitting

What is **Under-fitting** ?

When a model has not learned the patterns in the training data well and is unable to generalize well on the new data.

An underfit model has poor performance on the training data and will result in unreliable predictions.

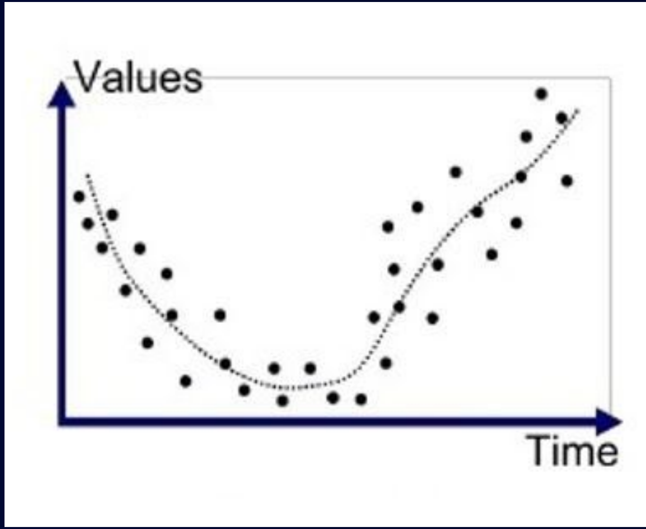
Reasons of **Under-fitting ?**

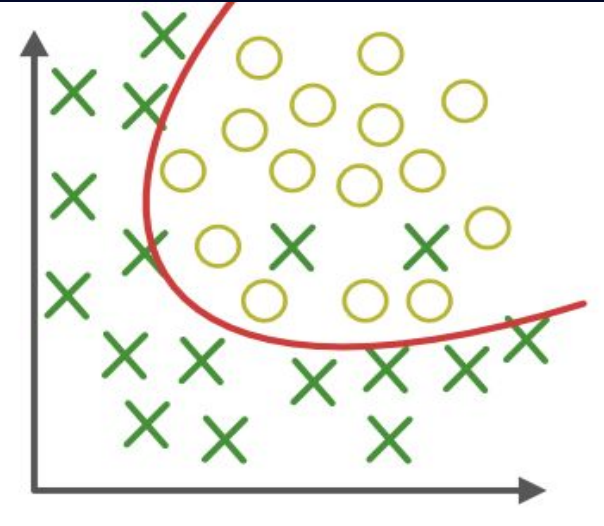
- Data used for training is not cleaned and contains noise in it
- The size of the training dataset used is not enough
- The model is too simple
- The model has a high bias and low variance.

Ways to deal with Under-fitting ?

- Increase the number of features in the dataset
- Increase model complexity
- Reduce noise in the data
- Increase the duration of training the data

Let’s take a closer look





Good-fitting

How to achieve Good-fitting ?

In order to achieve a good fit, you need to stop training at a point where the error starts to increase.

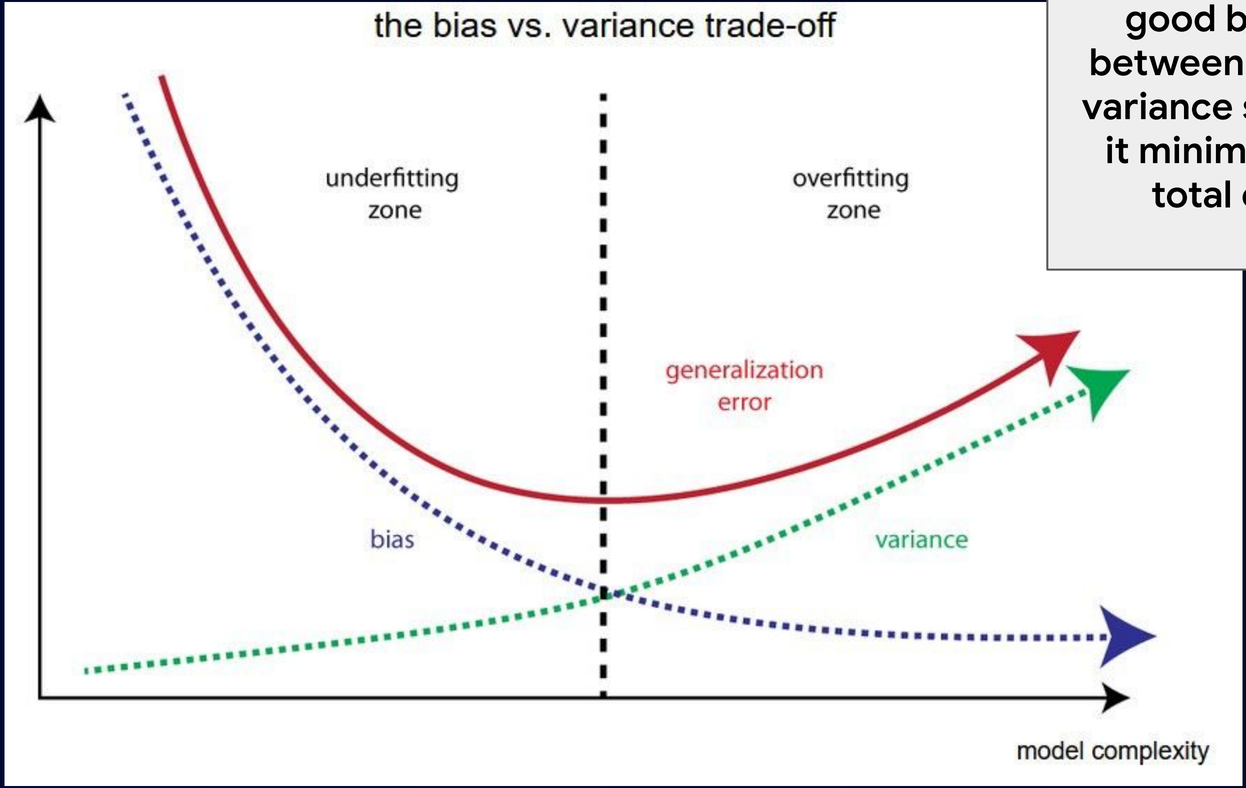
What is Bias ?

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.

What is Variance?

Variance is the variability of model prediction for a given data point or a value which tells us spread of our data.

Take a look!!

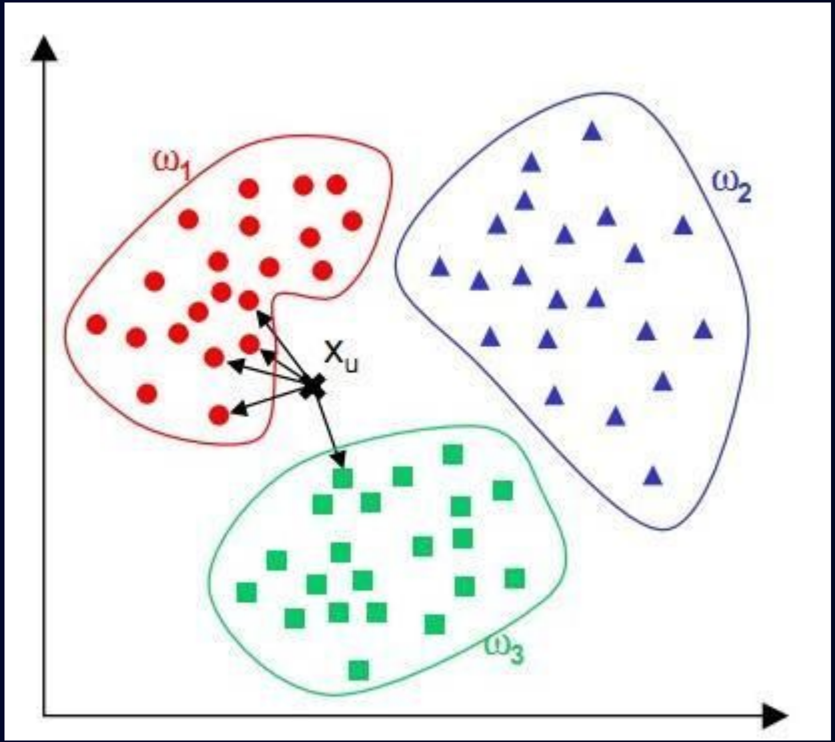


We need to find a good balance between bias and variance such that it minimizes the total error.

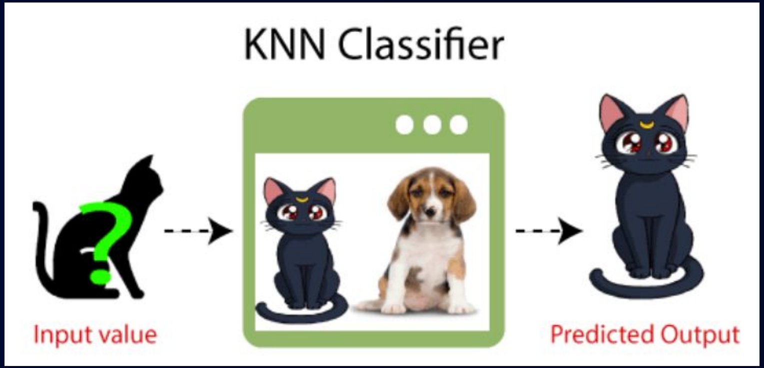
K-Nearest Neighbors (KNN)

How does it work?

- Also called a lazy learner algorithm
- Mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm



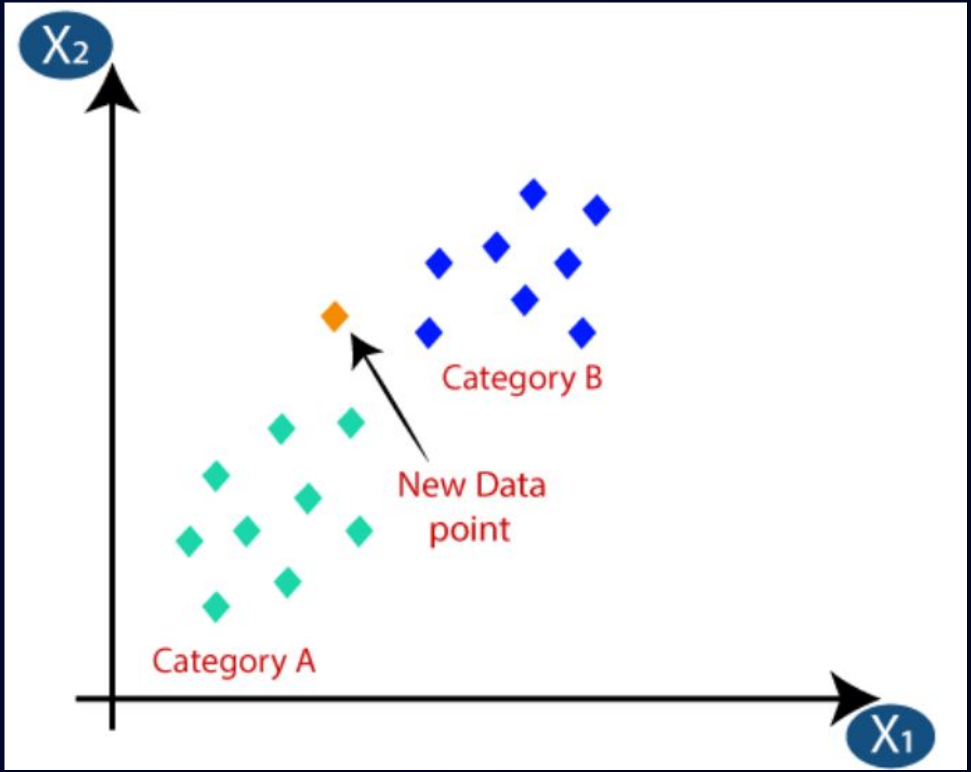
Example : is it a cat or dog?



Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

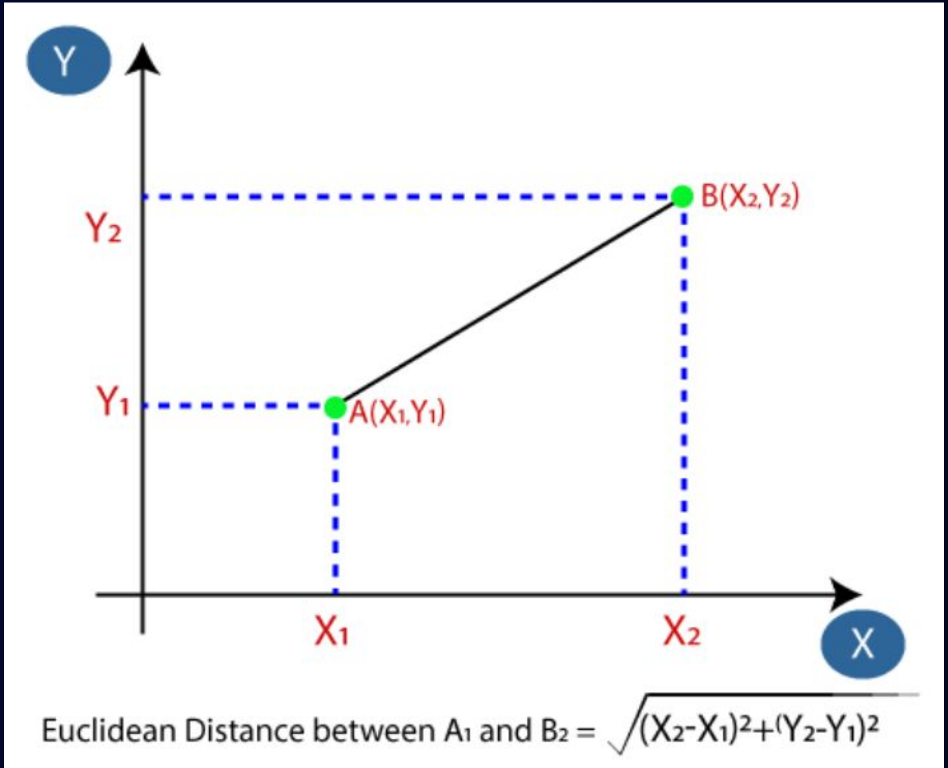
How does it work?

Step 1: Select the number K of the neighbors



How does it work?

Step 2: Calculate the Euclidean distance of K number of neighbors

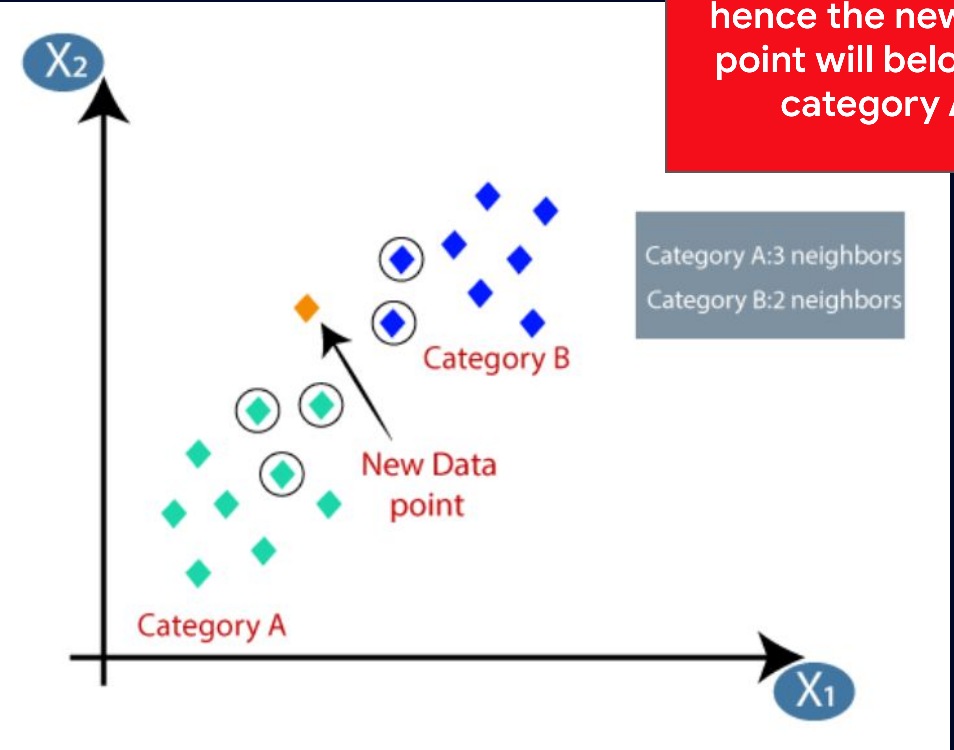


How does it work?

Step 3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step 4: Among these k neighbors, count the number of the data points in each category.

Step 5: Assign the new data points to that category for which the number of the neighbor is maximum.



As we can see the 3 nearest neighbors are from category A, hence the new data point will belong to category A.

Testing is the key!!

There are no particular ways to determine the best value for "K", so we need to try some values to find the best out of them.

The most preferred value for K is 5.

A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.

Large values for K are good, but it may find some difficulties.

Advantages

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages

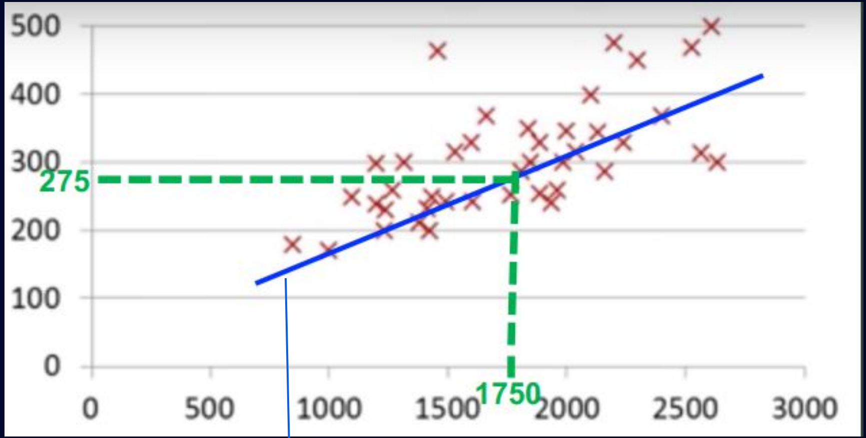
- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

Let practice!!

Linear Regression
One Variable

Let us go back to our House Pricing example

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...



How do you proceed to draw this line?

Let us go back to our House Pricing example

How do you proceed to draw this line?

I am sure what came to your mind is a simple function:

$$Y = ax + b$$

Let me tell you are not wrong! We will just replace the a and b with theta one and theta two :

$$h = \theta 0 + \theta 1x$$

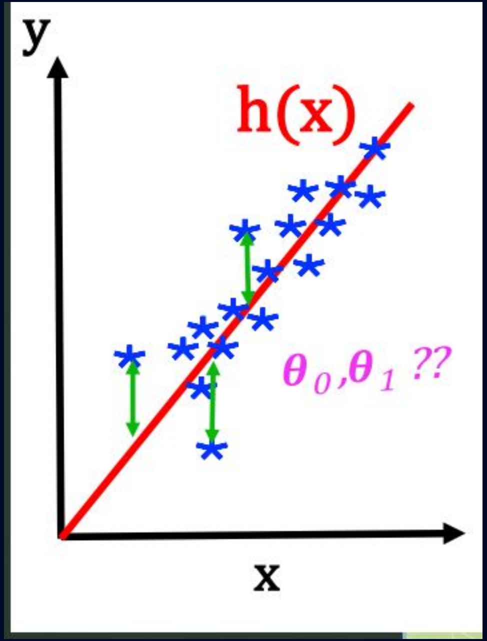
Hypothesis Function

Hypothesis function = θ0 + θ1x

How to choose theta 0 (θ0) and theta 1 (θ1) ?

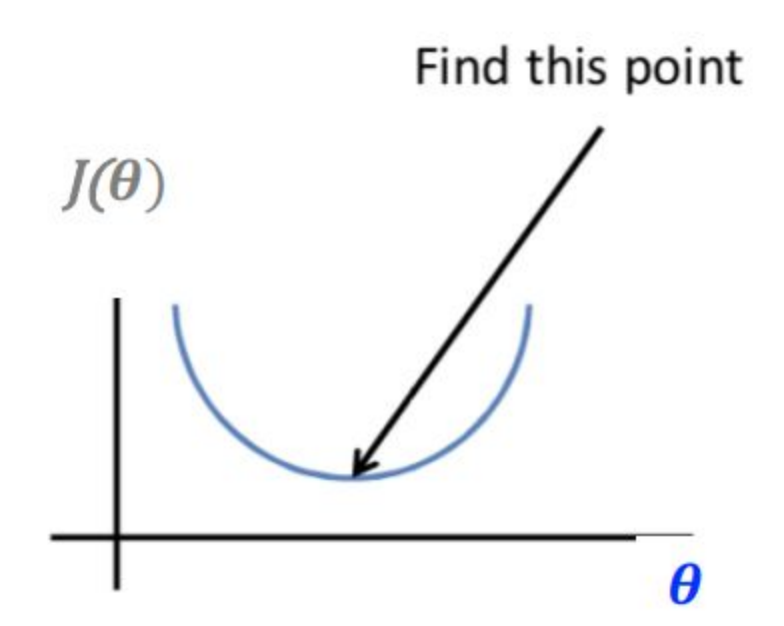
We choose them so that h(x) is close to our output y (target).

The Goal is to minimize a certain function called the Error Function (cost function)



Cost (Error) function

$$Cost\ funtion = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$



We are trying to minimize the cost function

Gradient Descent Algorithm

- **Input:** Data x, Labels y, Learning Rate alpha (α)
- θ_0, θ_1 = random values
- **Repeat:**

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

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

- **Until Convergence**
- **Output:** θ_0, θ_1

Linear Regression
Multiple Variables

Let us go back to our House Pricing example

Size ($feet^2$) X1	Number of bedrooms X2	Number of floors X3	Age of home (years) X4	Price ($1000) Y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

m

n= 4 Variables

Let us go back to our House Pricing example

$$h = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

Vectorisation

$$h = \begin{pmatrix} \theta_0, \theta_1, \theta_2, \theta_3, \theta_4 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

n+1 by 1 Matrix

1 by n+1 Matrix

How does it work?

Hypothesis Function

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta^T x$$

Cost Function

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

Gradient Descent

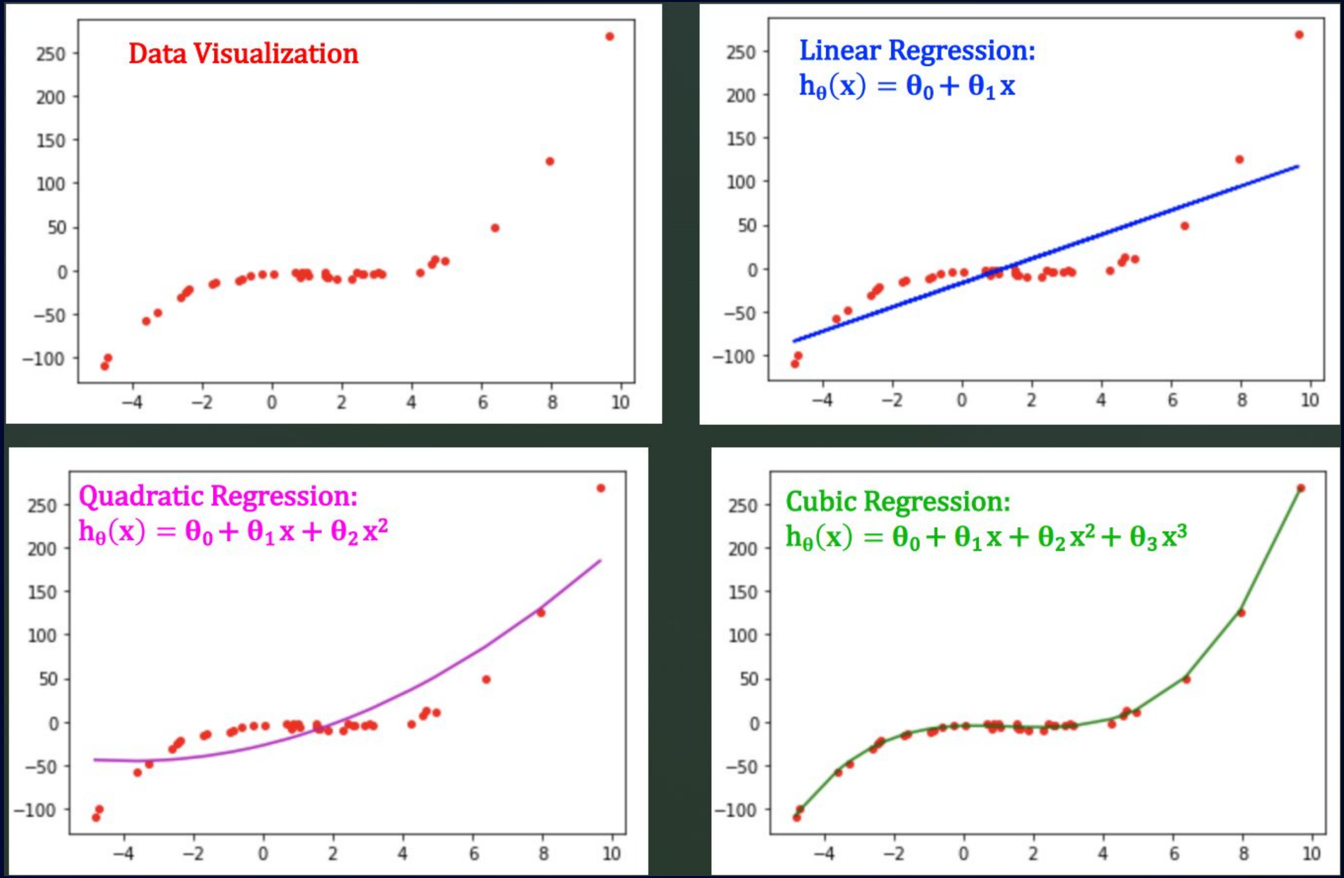
Repeat $\{$

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

(simultaneously update θ_j for $j = 0, \ldots, n$)

$\}$

Wanna go deep!!



Let practice!!

Normal Equation

Normal Equation?

It is a analytical method to solve for the optimal value of the parameters θ.

Let us go back to our House Pricing example

Size ($feet^2$)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
X1	X2	X3	X4	Y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

m

n= 4 Variables

Let us go back to our House Pricing example

$$X = \begin{pmatrix} 1, 2104, 5, 1, 45 \\ 1, 1416, 3, 2, 40 \\ 1, 1534, 3, 2, 30 \\ 1, 852, 2, 1, 36 \end{pmatrix}$$

X3

X4

$$Y = \begin{pmatrix} 460 \\ 232 \\ 315 \\ 178 \end{pmatrix}$$

Calculate Theta

$$\theta = (X^T X)^{-1} X^T y$$

Hypothesis function

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n = \theta^T x$$

Gradient Descent VS Normal Equation?

Gradient Descent

- Need to choose α
- Iterative Algorithm
- Feature scaling can be used
- Works well when n is large n ≥ 10^6

Normal Equation

- No Need to choose α
- Analytical approach
- No need for Feature scaling
- Works well when n is small
- Slow if n is very large

Logistic Regression
Binary Classification

How does it work?

Take an input vector x and assign it to one of 2 classes y.

$y \in \{0,1\}$

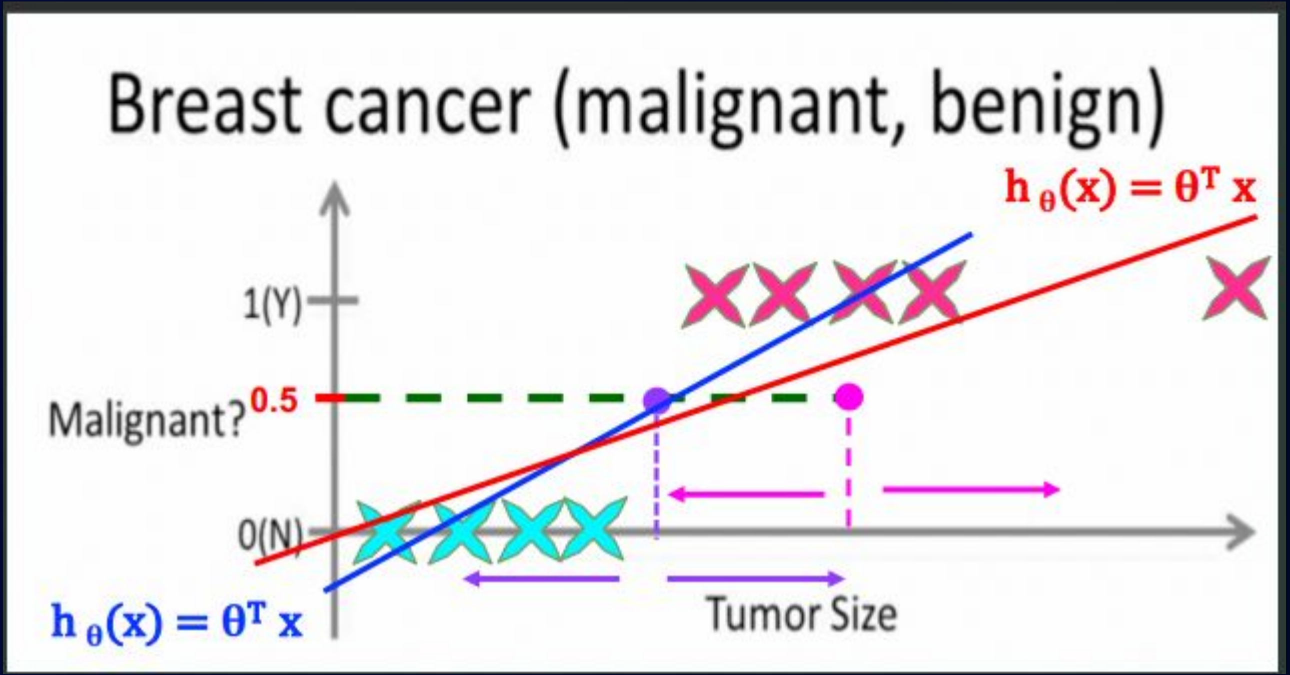
0 : Negative Class

E.g. Benign Tumor, Not Spam Email, ...

1 : Positive Class

E.g. Malignant Tumor, Spam Email, ...

Let us go back to our Breast cancer example

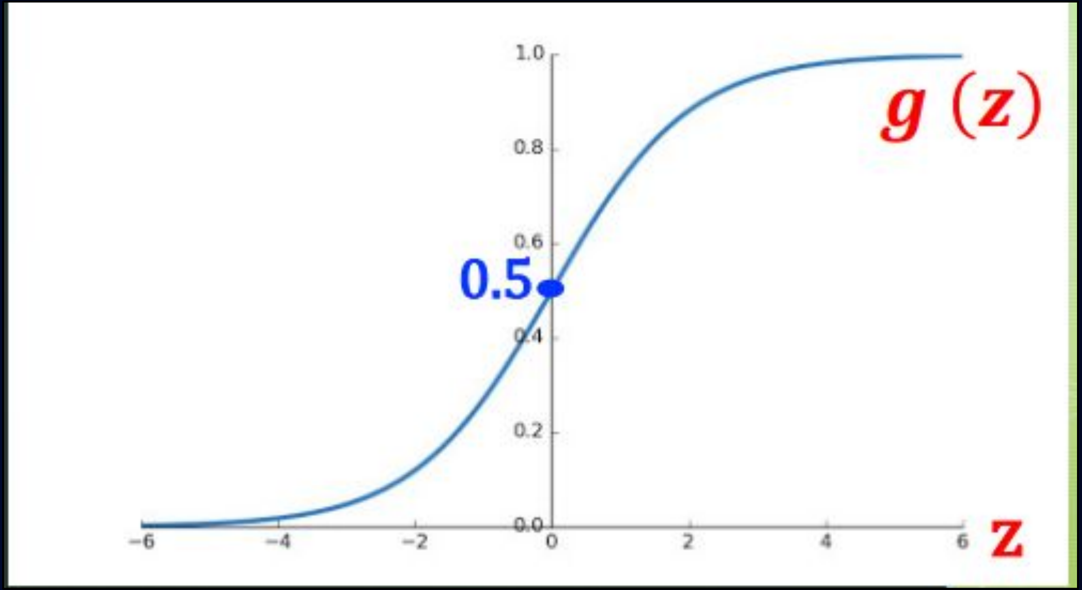


Y

Segmoid (Logistic) function?

Is a function that aims to predict the class to which a particular sample belongs.

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



Predict y = 1 if h(x) ≥ 0.5 ⟶ z ≥ 0
Predict y = 0 if h(x) < 0.5 ⟶ z < 0

How does it work?

Hypothesis Function

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

Cost Function

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

$$cost(h_\theta(x^{(i)}) - y^{(i)}) = \begin{cases} -\log(h_\theta(x)) \; if \; y = 1 \\ -\log(1 - h_\theta(x)) \; if \; y = 0 \end{cases}$$

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

How does it work?

Gradient Descent

Repeat {

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

(simultaneously update θ_j for $j = 0, \ldots, n$)

}

Decision Boundary?

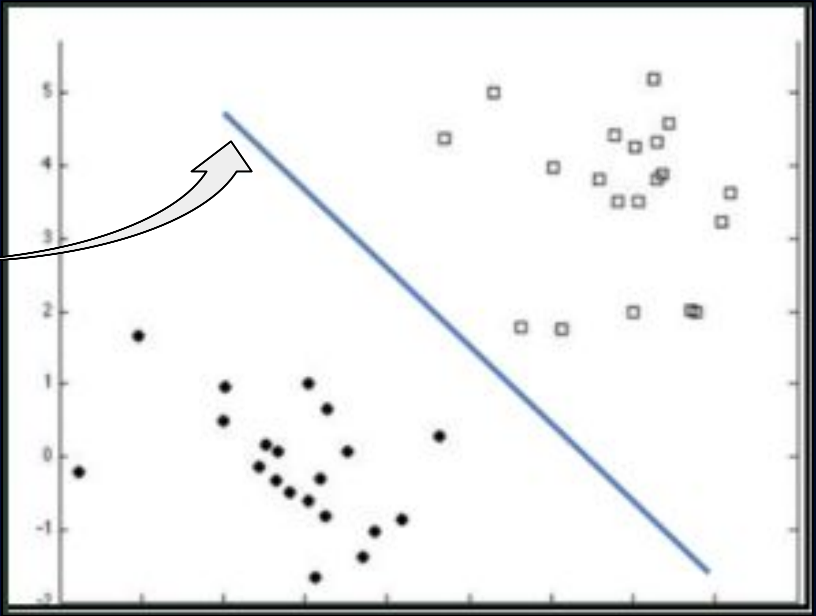
The Decision Boundary is the boundary between two classes,

where:

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} = 0.5$$

$$1 + e^{-\theta^T x} = 2$$

$$\theta^T x = 0$$

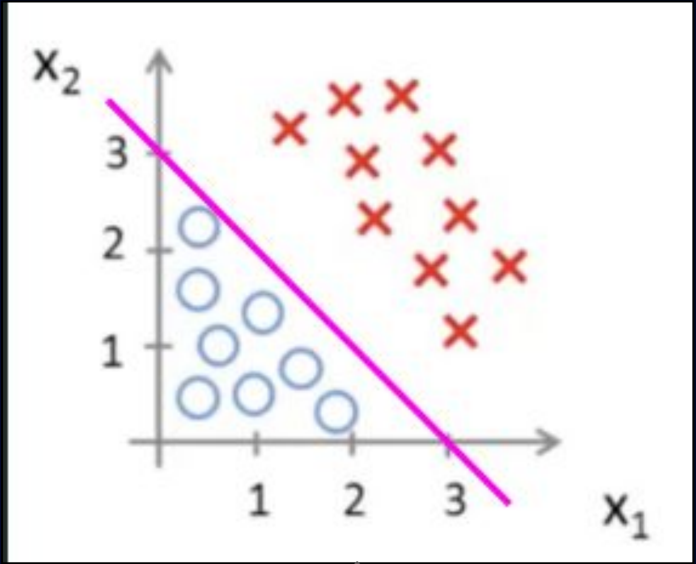


Decision Boundary?

Example:

$$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

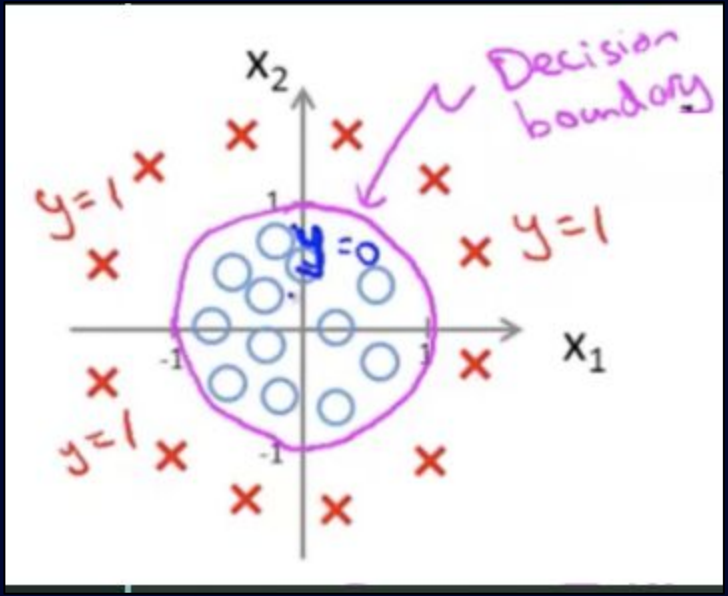
$$h_\theta(x) = g(\theta^T x)$$
$$h_\theta(x) = g(\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2)$$
$$\theta^T x = -3 + x_1 + x_2$$
$$y = 1 \; if \; \theta^T x \geq 0$$
$$y = 1 \; if \; 3 + x_1 + x_2 \geq 0$$
$$y = 1 \; if \; x_1 + x_2 \geq 3$$

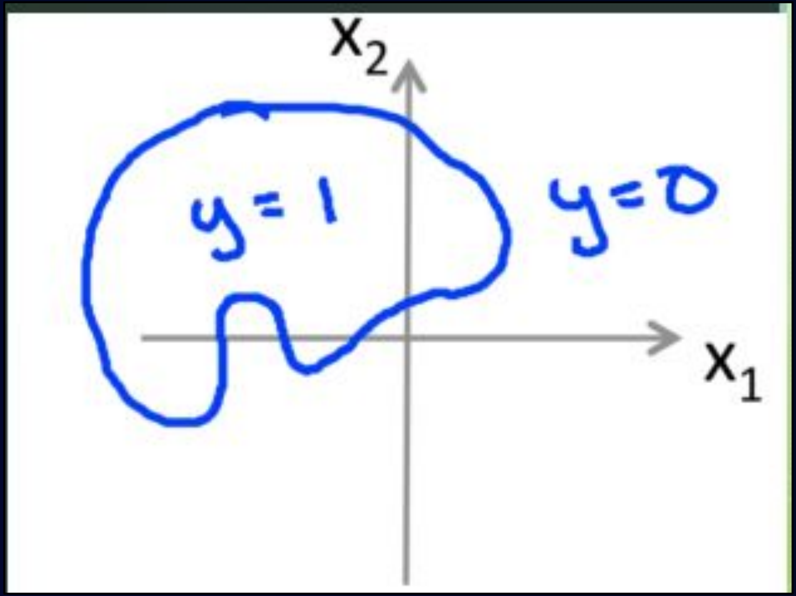


Decision Boundary: $x_1 + x_2 = 3$

$$h_\theta(x) = 0.5$$

Other Decision Boundary Type?





Non-Linear Decision Boundary

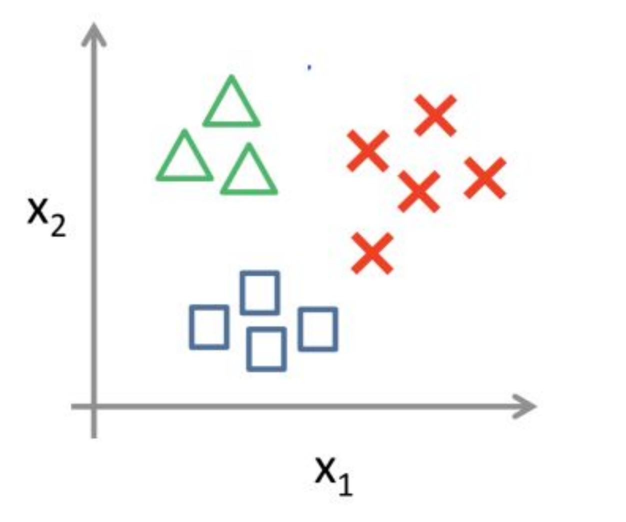
Logistic Regression
Multi class Classification

How does it work?

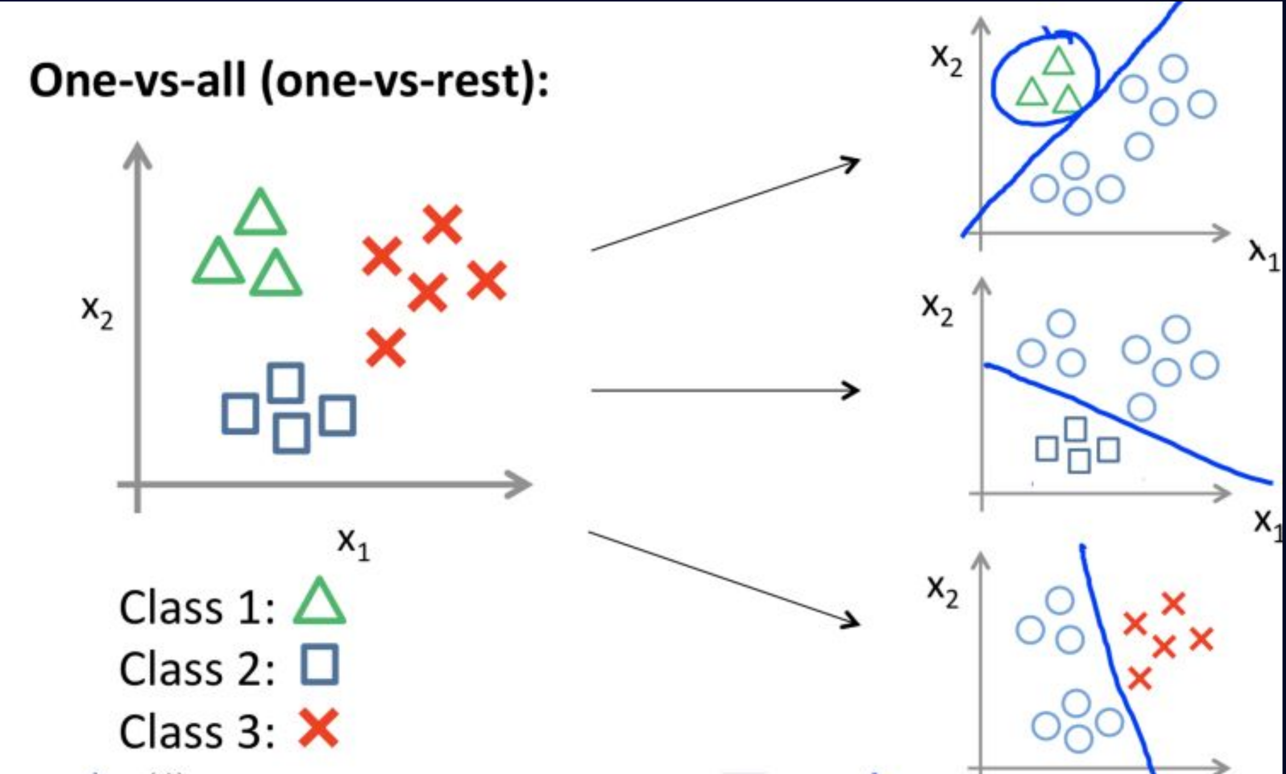
Multiclass classification involves predicting one of more than two classes, y = {1, 2, 3, ..., K} for K possible classes

Example : Emails Classification (K = 3)

- Work → y = 1
- Friends → y = 2
- Family → y = 3



From Binary to Multiclass



$$h_\theta^{(i)}(x) = P(y = i | x; \theta) \qquad (i = 1, 2, 3)$$

Let practice!!


SEE YOU TOMORROW

Thank you

What we will see ?

1. What is supervised Machine Learning
2. Some supervised Machine Learning Examples
3. Some common supervised Machine Learning Algorithms
4. Challenge

In this challenge you will have to implement Linear Regression algorithm for one variable using what we've seen in the first session.

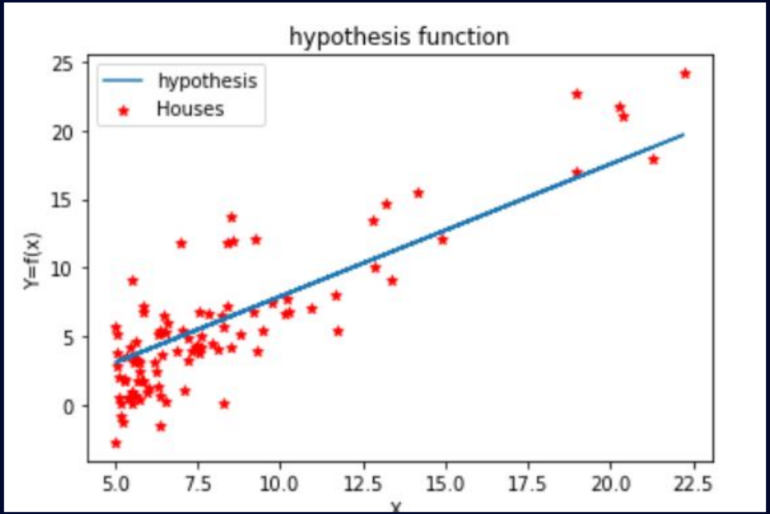
Database:

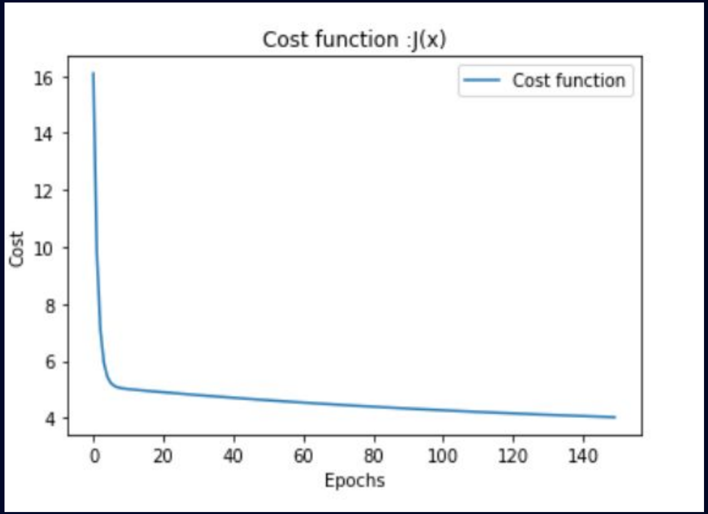
- https://drive.google.com/drive/folders/1POXWinRoZmj3KOymX_X87Bm9hvF_uzOK?usp=sharing

Script skeleton:

- https://colab.research.google.com/drive/1mG8aRXDYFGGo6K_3FkZfJ45PBKp2OfNI?usp=sharing

The desired Output:





🔗 Resources

- [Machine Learning | Andrew Ng]: https://www.youtube.com/watch?v=PPLop4L2eGk&list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN
-

Questions ?

Thank you