Machine Learning Diploma

Level3: Machine Learning

Session7

Agenda

- ➔ Classification Models
- ➔ Logistic Regression Algorithm
- ➔ Sklearn Logistic Regression
- ➔ Evaluating Classification Models Performance

1. Classification Models

Classification:

- ➔ Classification is a supervised learning concept which basically categorizes a set of data into classes.
- ➔ It is a predictive modelling problem where a class label is predicted for a given example of input data.

Classification problem try to learn categorical class.
such as “red” or “blue” or “yellow”

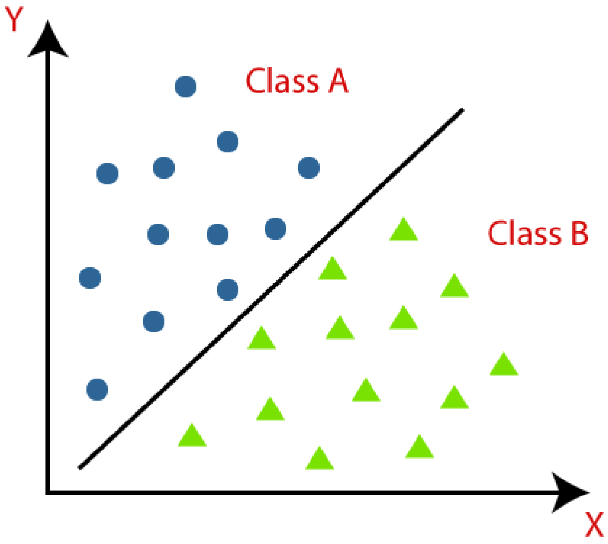
Classification:

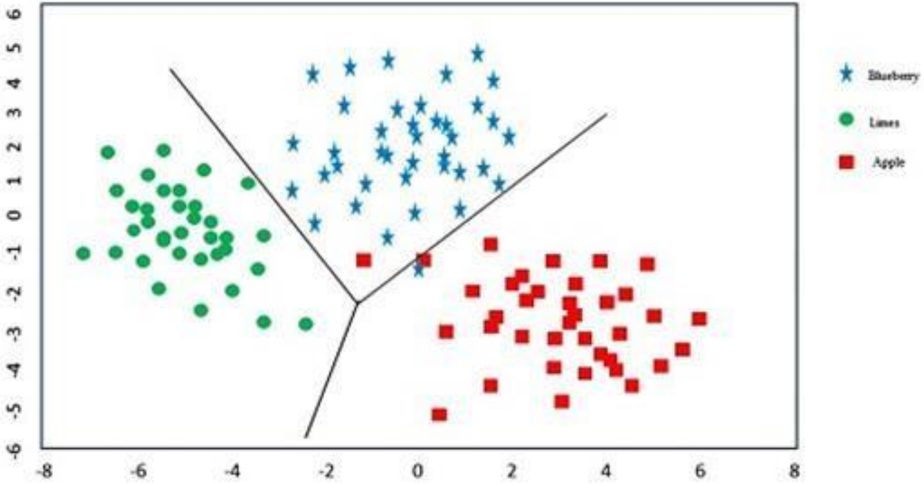
- ➔ Classification is a supervised learning concept which basically categorizes a set of data into classes.
- ➔ It is a predictive modelling problem where a class label is predicted for a given example of input data.

Classification problem try to learn categorical class.
such as “red” or “blue” or “yellow”

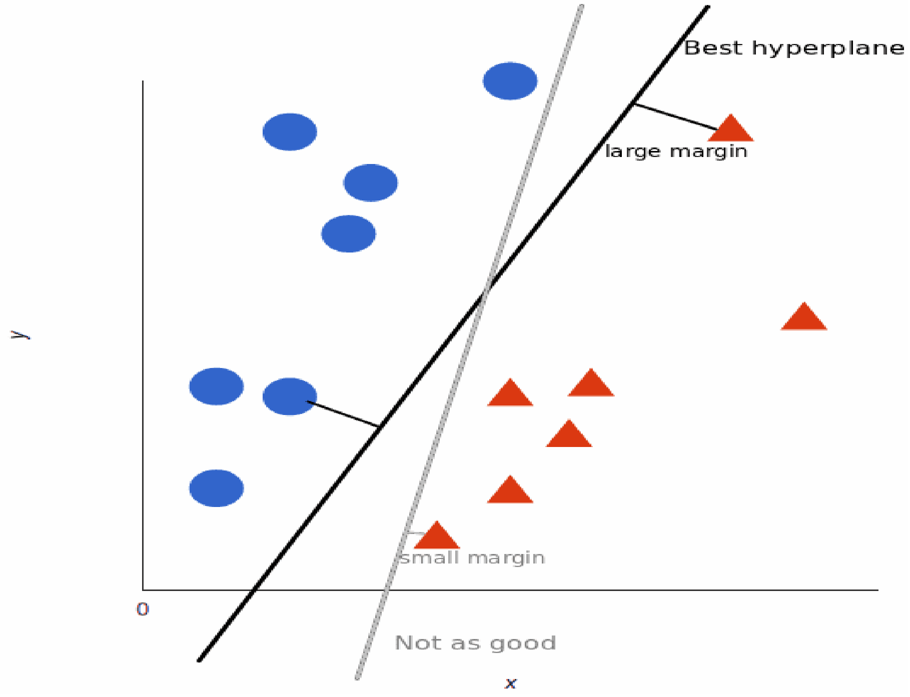
Classification :

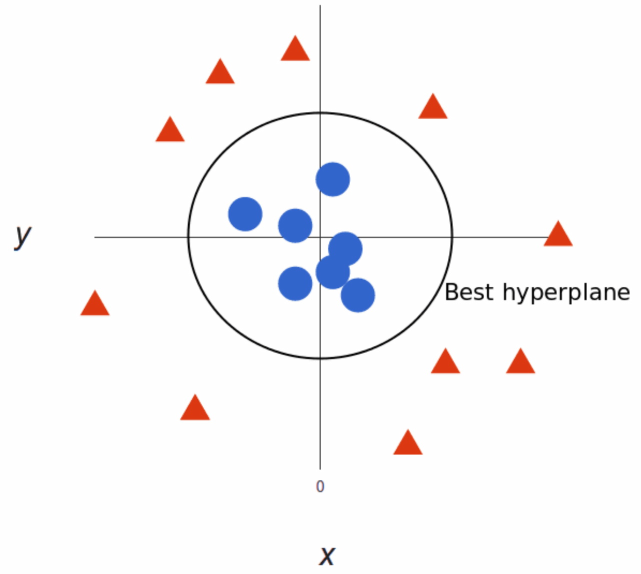
- Classification algorithm is a Supervised Learning technique that is used to identify the class of new observations on the basis of training data.



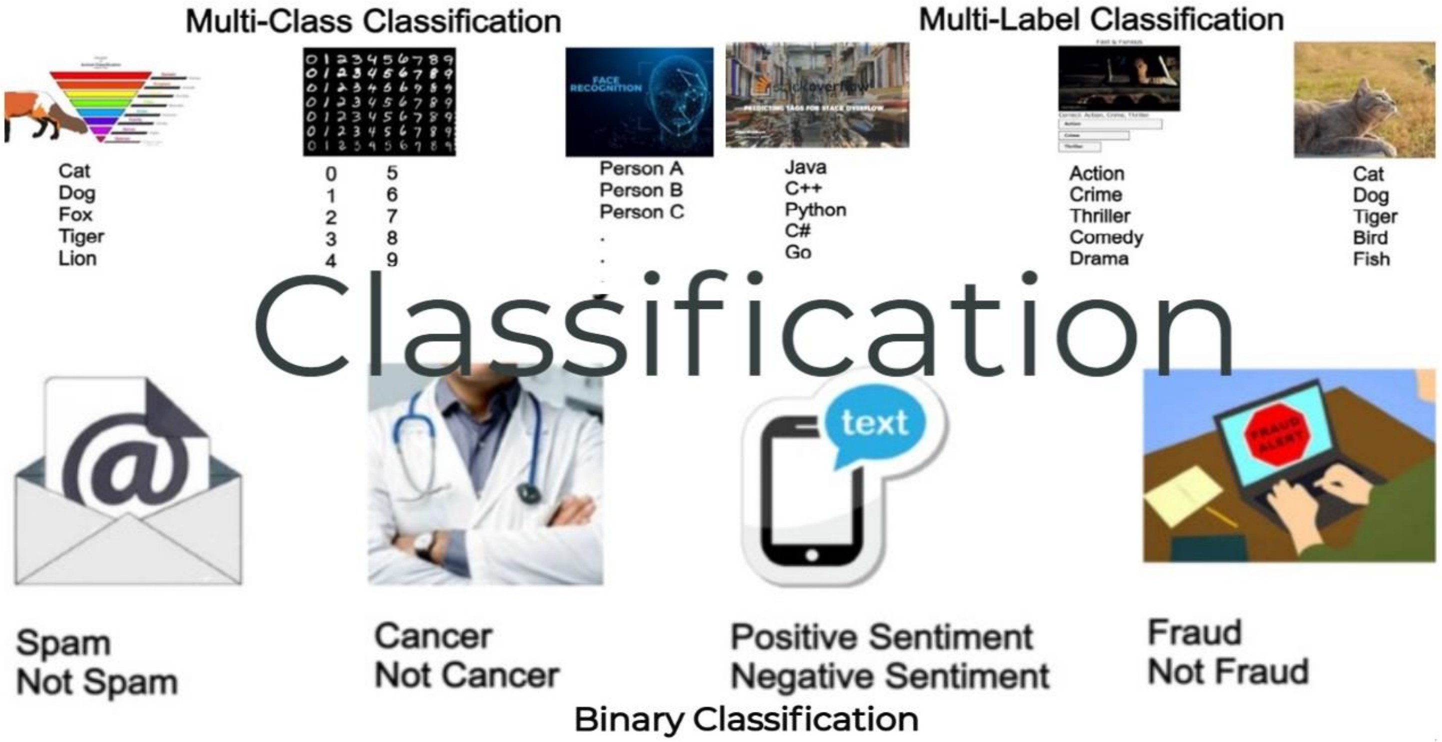


Classification :





Classification use case :



2. Logistic Regression Algorithm

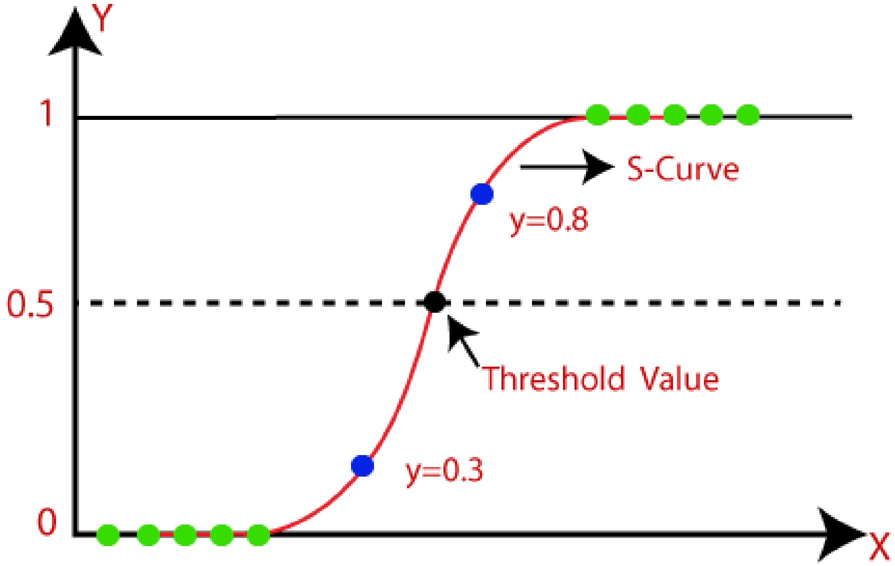
Logistic Regression:

- **Logistic Regression** is a supervised learning classification algorithm used to predict the probability of a **Target Categorical Variable**.
- The outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the **Probabilistic Values** which lie between 0 and 1.
- In Logistic Regression, instead of fitting a regression line, we fit an **"S"** shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something.

AMIT

Logistic Function:

➔ The **Logistic/Sigmoid** function is a mathematical function used to map the predicted values to probabilities.



AMIT

Logistic Regression Equation:

- ➔ The below is the Hypothesis of Linear Regression.
- ➔ Logistic Regression just has a transformation based on it.
- ➔ For logistic regression, focusing on binary classification here, we have class 0 and class 1.

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_j x_j \quad = \theta^T x \quad = \begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_j \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \ldots \\ x_j \end{bmatrix}$$

$$\textit{Where } x_0 = 1$$

Logistic Regression Equation:

$$\boxed{\hat{Y} = \boldsymbol{h_\theta(x)} = \boldsymbol{\sigma(\theta^T . x)}}$$

$\hat{Y}$: probability that an instance x is belongs to a class
$\boldsymbol{h_\theta}$: model parameters that determine boundary decision
$\boldsymbol{x}$: features
$\boldsymbol{\sigma}$: logistic (sigmoid — Logit) S-shaped function
- Map any real-valued number to between 0, 1

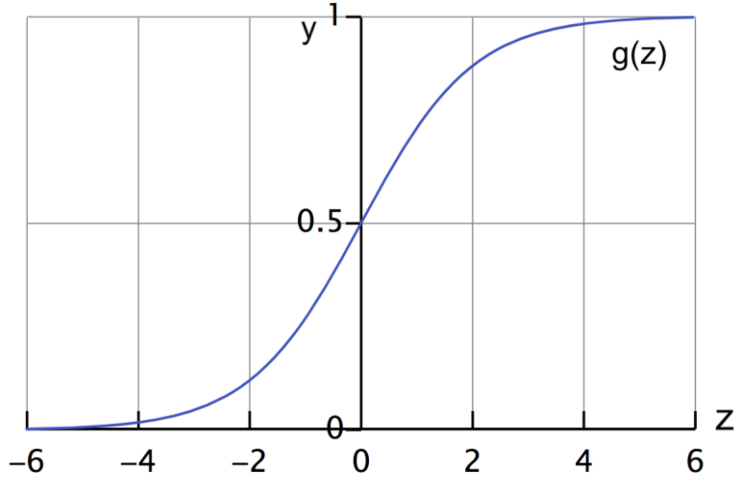
AMIT

Logistic Regression Equation:

- ➔ To compare with the target, we want to constrain predictions to some values between 0 and 1.
- ➔ That's why **Sigmoid Function** is applied on the raw model output and provides the ability to predict with probability.

$$\textit{Sigmoid Function}: \quad g(z) = \frac{1}{1 + e^{(-z)}}$$

$$\textit{Hypothesis}: \quad h_\theta(x) = \frac{1}{1 + e^{(-\theta^T x)}}$$



Logistic Regression Equation:

- ➔ What hypothesis function returns is the probability that $y = 1$, given x, parameterized by θ.
- ➔ Written as: $h(x) = P(y = 1 | x; \theta)$.
- ➔ Decision boundary can be described as:
  Predict 1, if $\theta^T x \geq 0 \rightarrow h(x) \geq 0.5$ Predict
  0, if $\theta^T x < 0 \rightarrow h(x) < 0.5$

AMIT

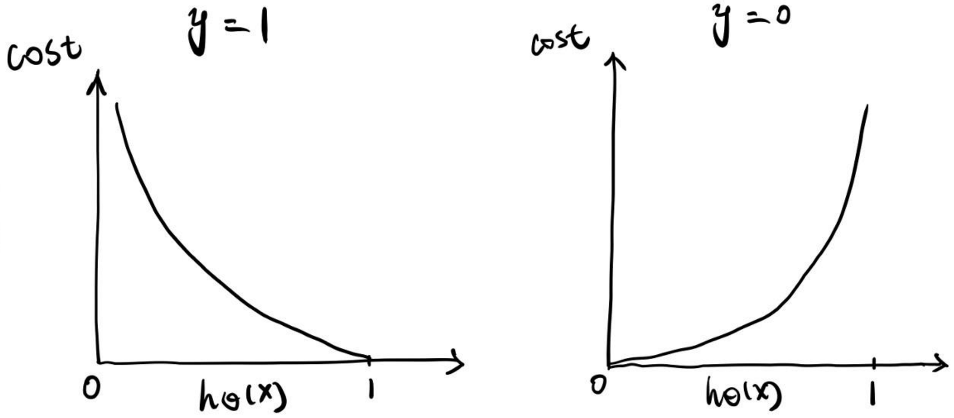
Cost Function:

- ➔ We want to assign more punishment when predicting 1 while the actual is 0, and when predicting 0 while the actual is 1.
- ➔ The Loss Function of Logistic Regression is doing this exactly which is called **Logistic Loss**.

Cost Function:

- If y = 1, the plot on left, when prediction = 1, the cost = 0, when prediction = 0, the learning algorithm is punished by a very large cost.
- Similarly, if y = 0, the plot on right, predicting 0 has no punishment but predicting 1 has a large value of cost.

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



Cost Function:

➔ If y = 1, the plot on left, when prediction = 1, the cost = 0, when prediction = 0, the learning algorithm is punished by a very large cost

$$Cost(h_\theta(x), y) \quad = \quad -y \; log(h_\theta(x)) - (1-y)log(1-h_\theta(x))$$

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

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

AMIT

Optimization Function:

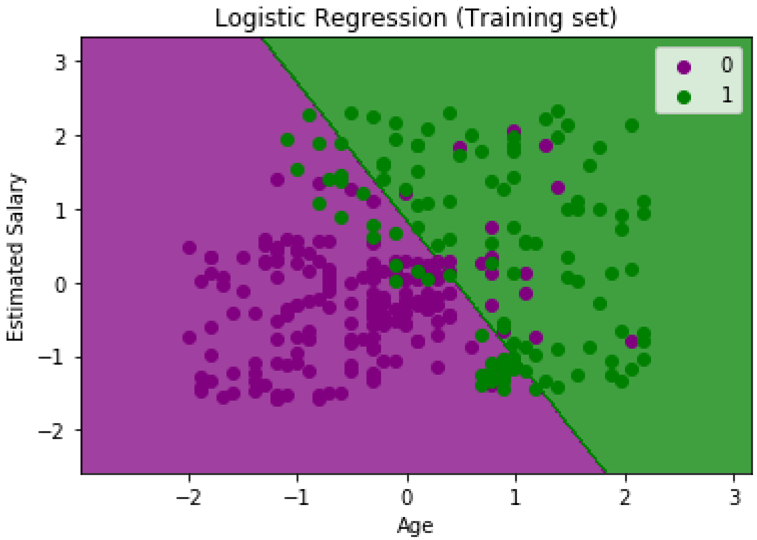
- ➔ We can start to fit by minimizing J(θ) as a function of θ to find optimal parameters.
- ➔ We can still apply **Gradient Descent** as the optimization algorithm.
- ➔ It takes partial derivative of J with respect to θ (the slope of J), and updates θ via each iteration with a selected learning rate **α** until the **Gradient Descent** has converged.

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

AMIT

3. Sklearn Logistic Regression

Sklearn Logistic Regression:



4. Evaluating Classification Models Performance

Evaluating Classification Models Performance

- → Confusion Matrix
- → Accuracy
- → Precision
- → Recall
- → F1 Score

Evaluating Classification Models Performance

		Actual class	
		Positive	Negative
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative

True Class \ Predicted Class	0	1	2	3	4	5	6	7	8	9
0	38	0	0	0	0	0	0	0	0	0
1	0	35	0	0	0	0	0	0	2	0
2	0	0	39	0	0	0	0	0	0	0
3	0	0	0	38	0	1	0	0	2	0
4	0	0	0	0	40	0	0	1	0	0
5	0	0	0	0	0	27	0	0	0	0
6	0	0	0	0	0	1	29	0	0	0
7	0	0	0	0	0	0	0	35	0	1
8	0	2	0	0	0	0	0	0	32	0
9	0	0	0	0	0	0	0	1	1	35

Confusion Matrix:

There are four quadrants in the confusion matrix, which are symbolized as below.

True Positive (TP: f_{++}) : The number of instances that were positive (+) and correctly classified as positive (+v).

True Negative (TN: f_{--}): The number of instances that were negative (-) and correctly classified as (-).

False Negative (FN: f_{+-}): The number of instances that were positive (+) and incorrectly classified as negative (-). It is also known as **Type 2 Error**.

False Positive (FP: f_{-+}): The number of instances that were negative (-) and incorrectly classified as (+). This also known as **Type 1 Error**.

		Actual class	
		Positive	Negative
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative

Accuracy:

This term tells us how many right classifications were made out of all the classifications. In other words, how many TPs and TNs were done out of TP + TN + FP + FNs. It tells the ratio of “True” s to the sum of “True” s and “False” s.

		Actual class	
		Positive	Negative
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative

$$\textbf{\textit{Accuracy}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall (Sensitivity):

Out of all the actual real positive cases, how many were identified as positive.

		Actual class	
		Positive	Negative
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative

$$\text{Recall} = \frac{\mathit{True\ Positive}}{\mathit{True\ Positive} + \mathit{False\ Negative}}$$

$$= \frac{\mathit{True\ Positive}}{\mathit{Total\ Actual\ Positive}}$$

Precision:

Out of all that were marked as positive, how many are actually truly positive.

		Actual class	
		Positive	Negative
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative

$$Precision = \frac{True\,Positive}{True\,Positive + False\,Positive}$$

Precision:

F1 score is a weighted average of Precision and Recall, which means there is equal importance given to FP and FN. This is a very useful metric compared to "**Accuracy**".

		Actual class	
		Positive	Negative
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative

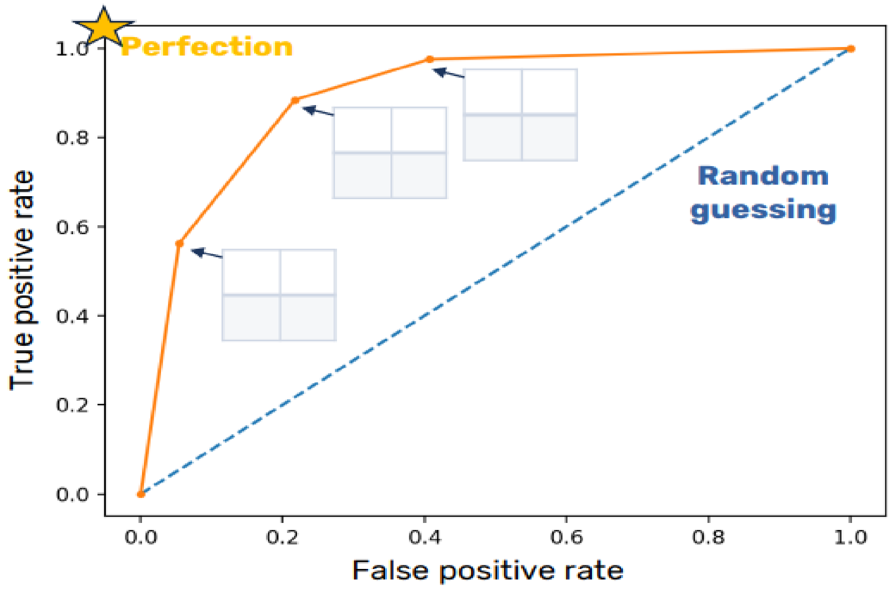
$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

$$\text{F1 Score} = \frac{\boxed{TP}}{\boxed{TP} + \frac{1}{2}\left(\boxed{FP} + \boxed{FN}\right)}$$

AMIT

Area Under Curve (AUC) and ROC Curve:

A ROC Curve is drawn by plotting TPR or True Positive Rate or Recall or Sensitivity (which we saw above) in the y-axis against FPR or False Positive Rate in the x-axis. FPR = 1- Specificity (which we saw above).



Confusion Matrix:

- ➔ Recall: Ability of a model to find all the relevant cases within a dataset.
- ➔ Precision: Ability of a model to identify only the relevant data points.
- ➔ F1 Score: harmonic mean of precision and recall.

Confusion Matrix:

		Actual class		
		Positive	Negative	
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)	**Precision:** $\frac{TP}{(TP + FP)}$
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative	**Negative Predictive Value:** $\frac{TN}{(TN+FN)}$
		Recall or Sensitivity: $\frac{TP}{(TP + FN)}$	**Specificity:** $\frac{TN}{(TN + FP)}$	**Accuracy:** $\frac{TP + TN}{(TP + TN + FP + FN)}$

Case Study:

Case 1	Case 2	Case 3
		
COVID 19/ Healthy	**Spam/Not Spam**	**Good/Bad loan**
Cost of **FN** > Cost of **FP**	Cost of **FP** > Cost of **FN**	Cost of **FN** > Cost of **FP**
Recall	**Precision**	**Recall**

$$recall = \frac{true\ positives}{true\ positives\ +\ false\ negatives}$$

$$precision = \frac{true\ positives}{true\ positives\ +\ false\ positives}$$

$$recall = \frac{true\ positives}{true\ positives\ +\ false\ negatives}$$

Any Questions?

THANK YOU!

AMIT