BCSE00133

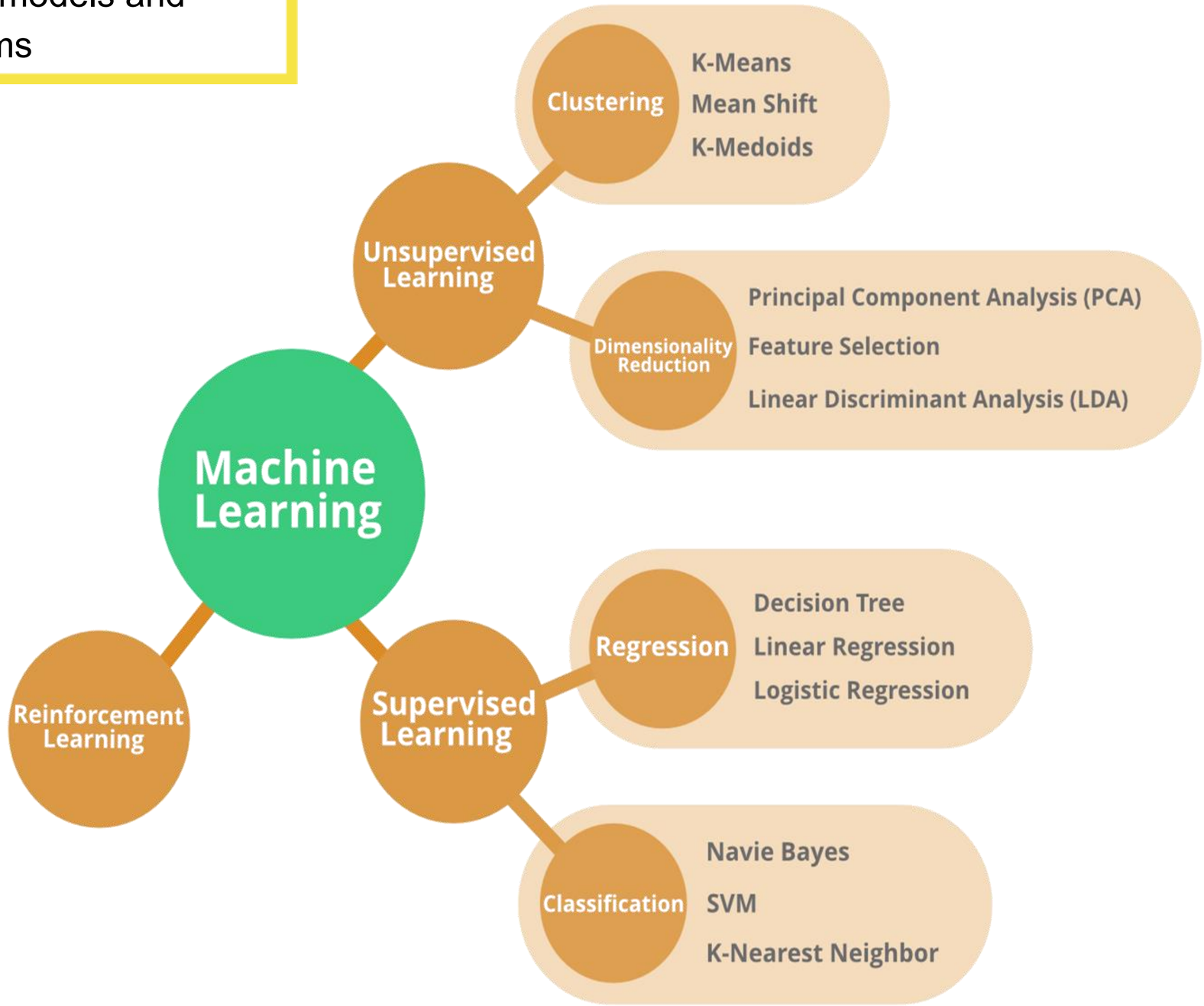
Logistic Regression and KNN

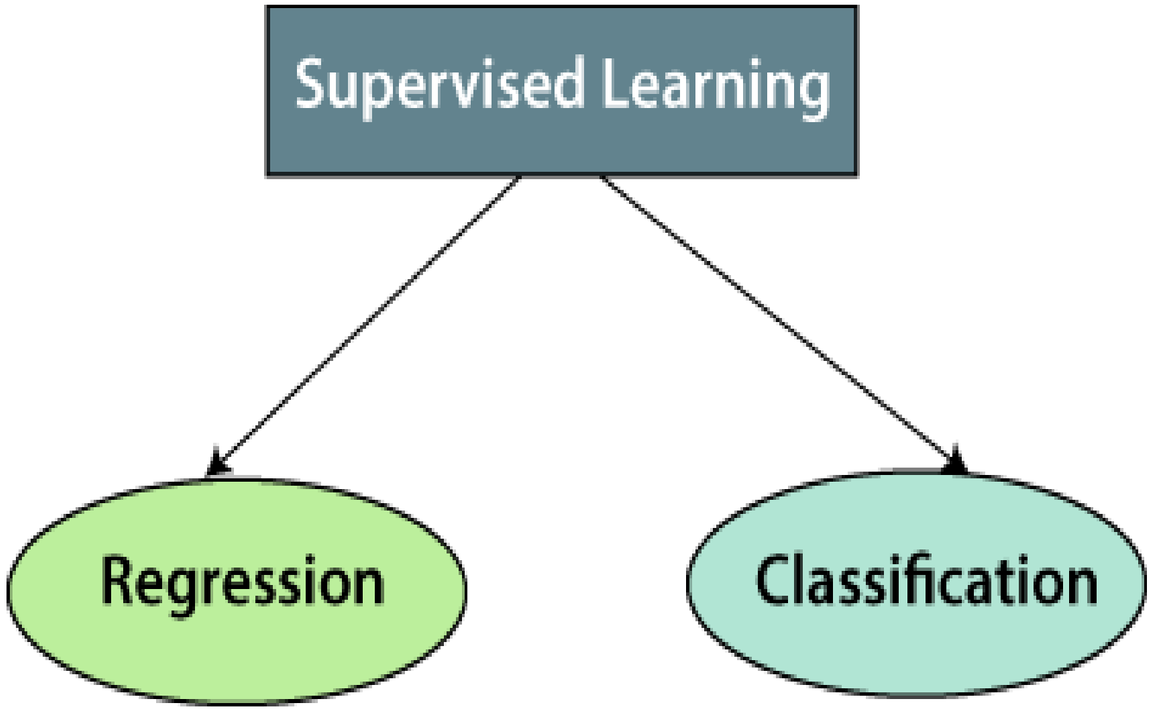
Machine Learning Lab

DR GAURAV KUMAR

ASST. PROF, CEA, GLA UNIVERSITY

Machine learning models and algorithms





Different types of Classification Algorithms -

For Linear Models

- Logistic Regression
- Support Vector Machines

Non-linear Models

- K-Nearest Neighbors
- Naïve Bayes
- Decision Tree Classification

Continuous & Categorical Values

Continuous variables

A variable is said to be continuous if it can assume an infinite number of real values within a given interval.

For instance, consider the height of a student.

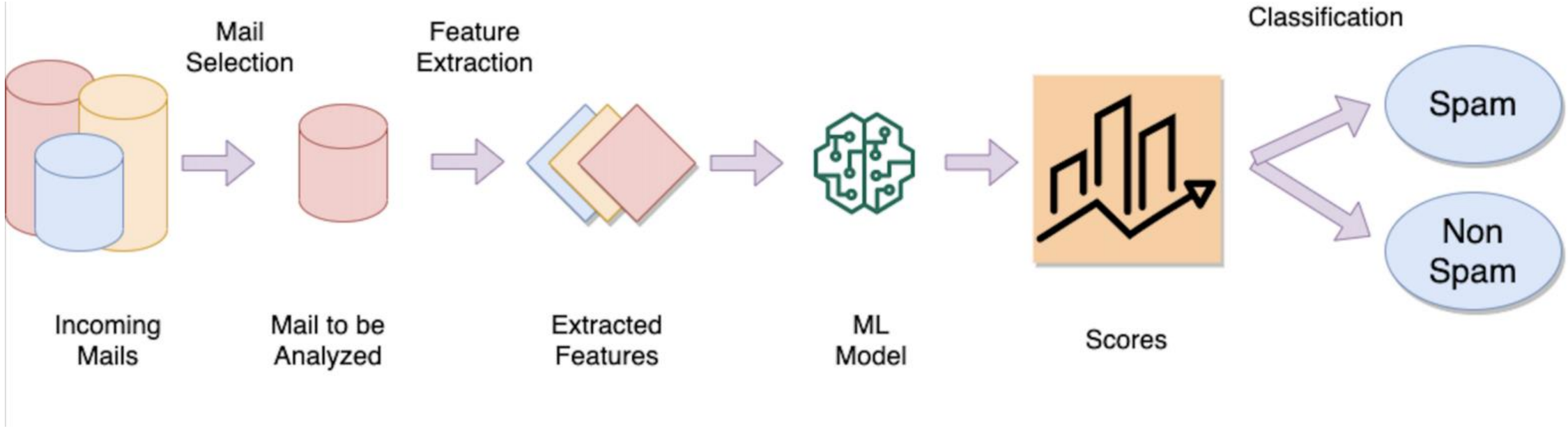
Categorical variables

A categorical variable (also called a qualitative variable) refers to a characteristic that can’t be quantifiable.

For Instance: Yes, No, Good, Bad, Excellent etc.

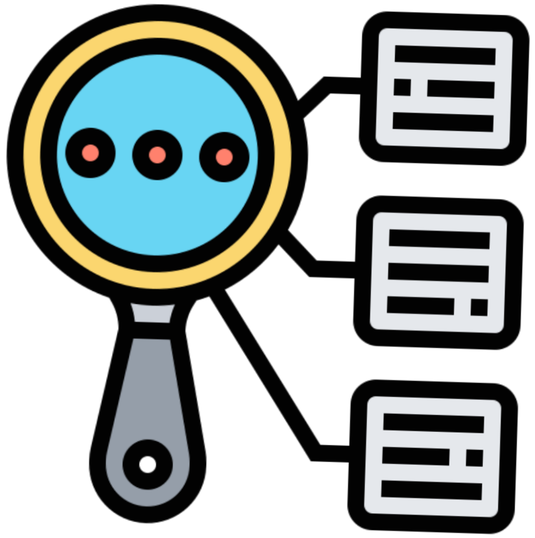
Example of Classification Algorithm

The best example of an ML classification algorithm is **Email Spam Detector**.



The main goal of the Classification algorithm is to identify the category of a given dataset, and these algorithms are mainly used to predict the output for the categorical data.

Classification Algorithm



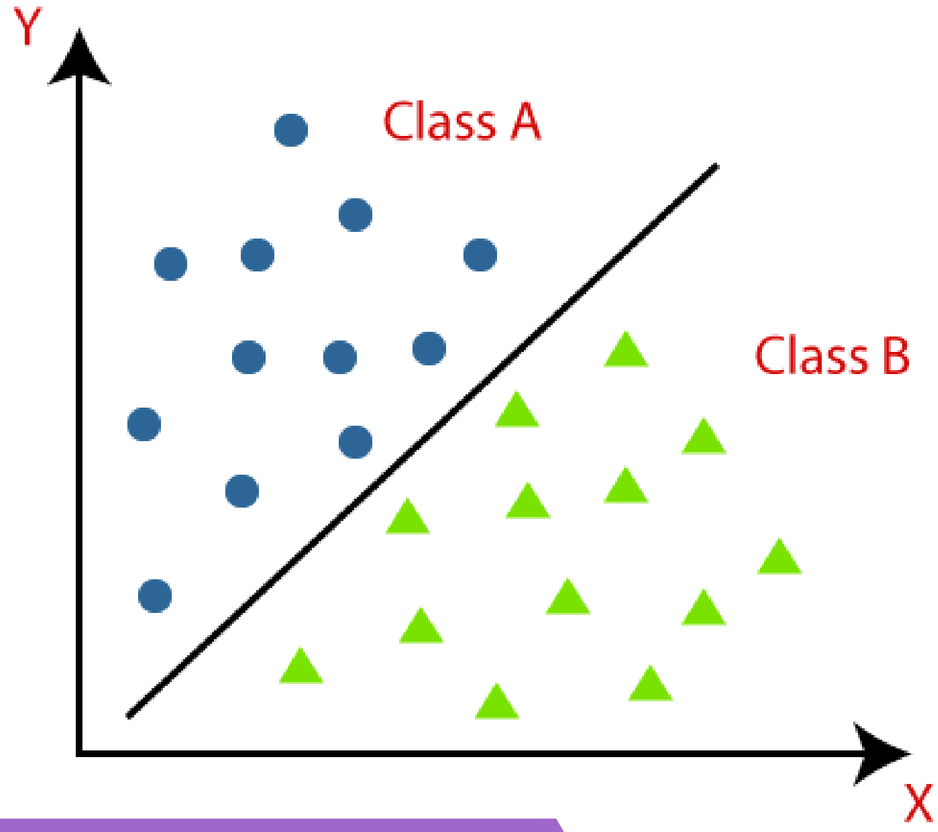
- The Classification algorithm is a Supervised Learning technique that is used to **identify the category of new observations** on the basis of training data.

- In Classification, a program learns from the given dataset or observations and then classifies new observations into a number of classes or groups. Such as, **Yes or No, 0 or 1, Spam or Not Spam, cat or dog**, etc.

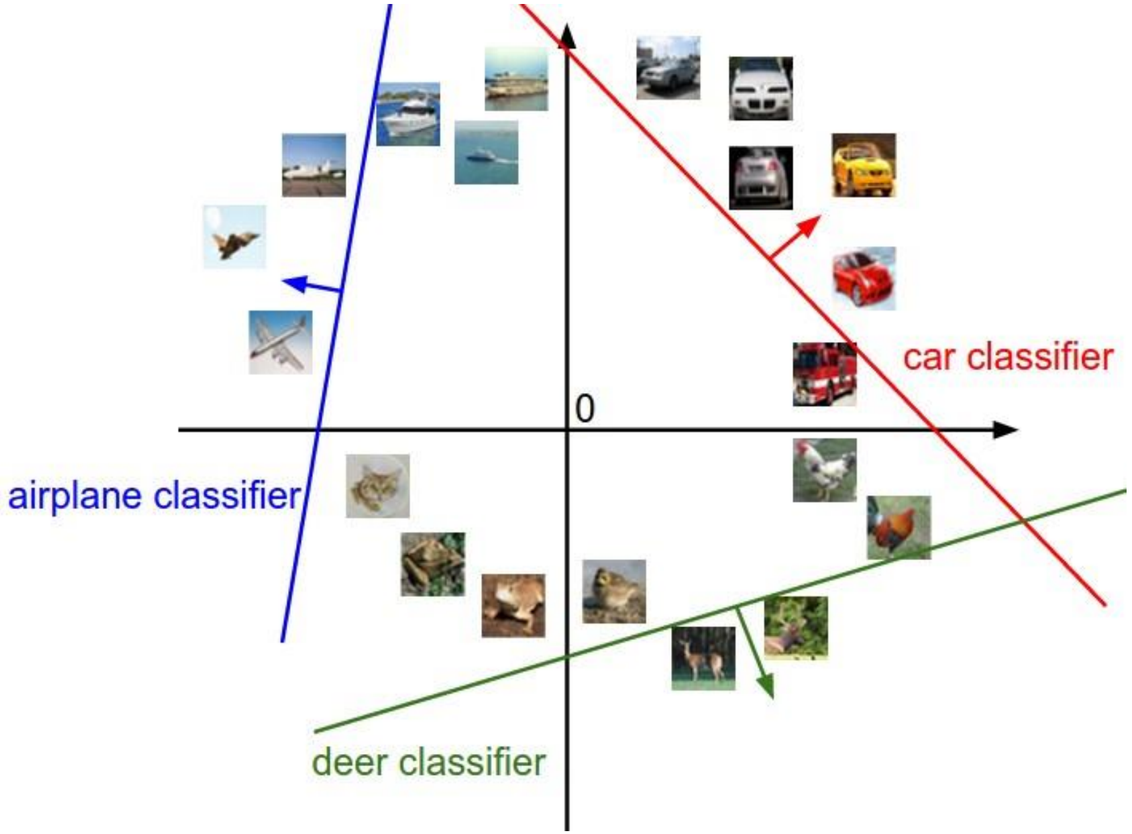
- Classes can be called as **targets/labels or categories**

Example of Linear Classification Algorithm

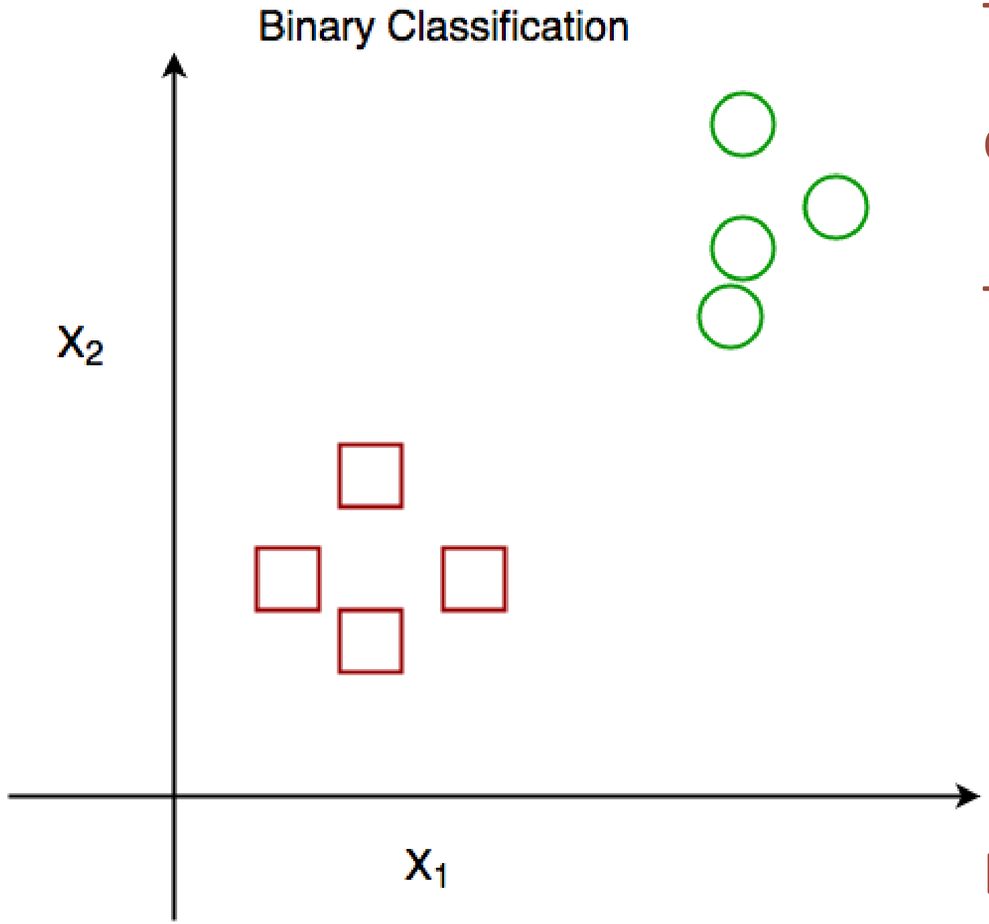
In the diagram, there are two classes, class A and Class B.



In the diagram, there are three classes, class Car, Class Deer and Class Airplane.



Types of Classification Algorithm



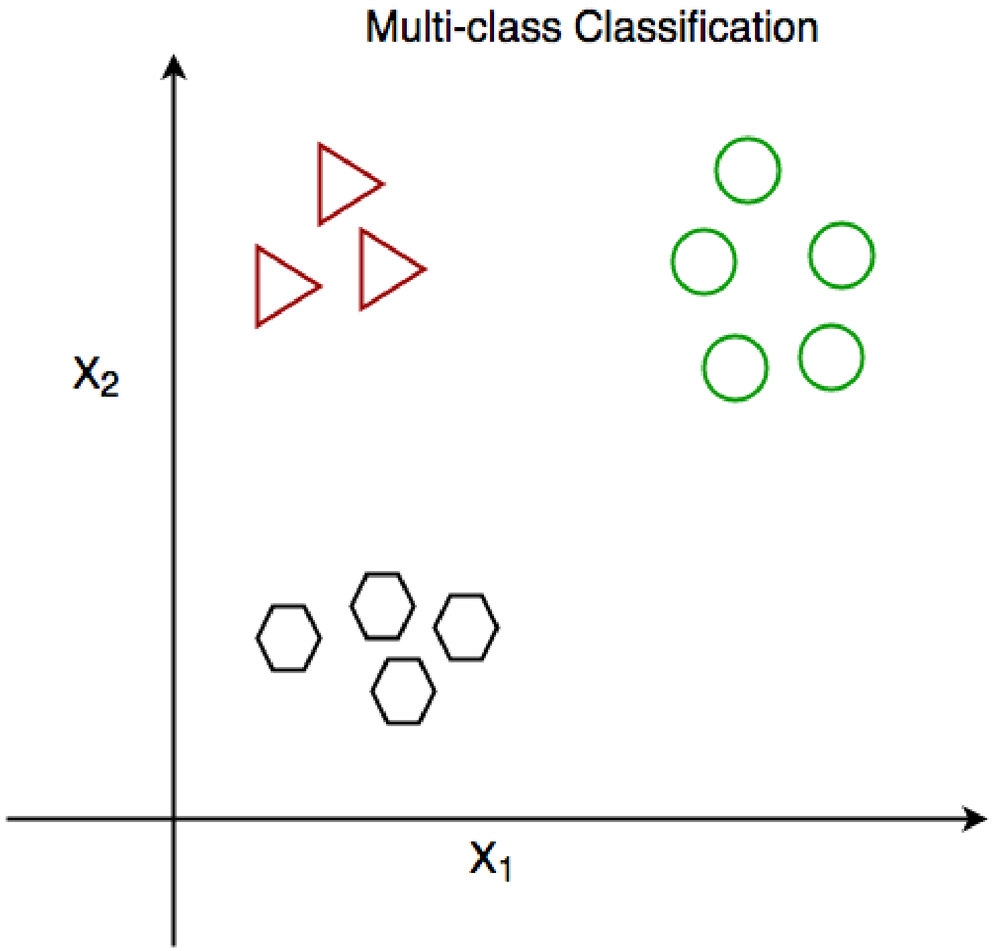
The algorithm which implements the classification on a dataset is known as a **classifier**.

There are **two types of Classifications**:

- **Binary Classifier**: If the classification problem has only two possible outcomes, then it is called a Binary Classifier.

Examples: **YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG**, etc.

Types of Classification Algorithm



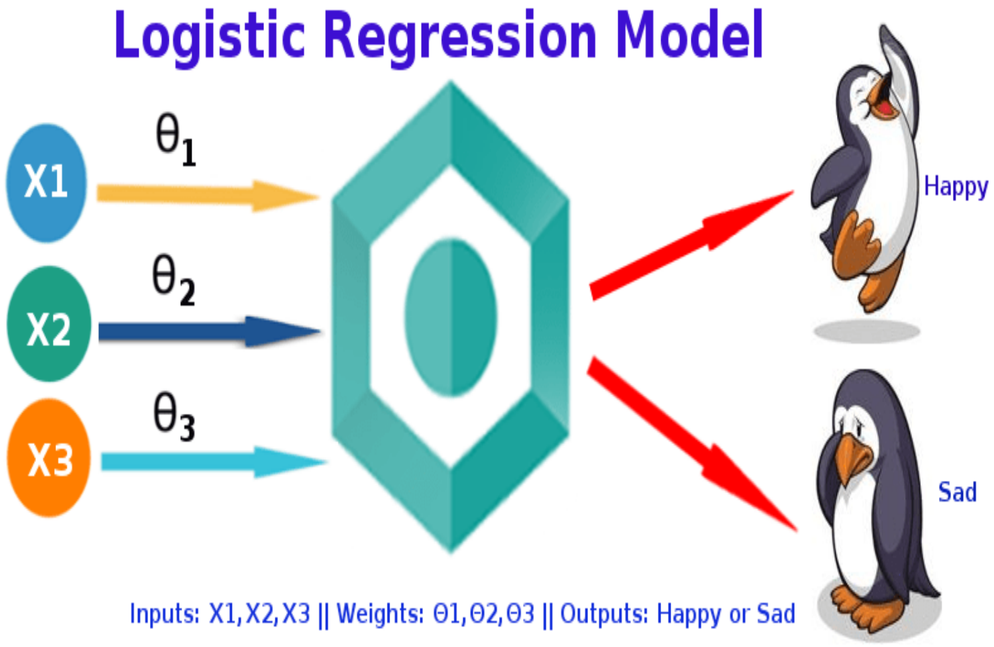
Multi-class Classifier: If a classification problem has **more than two outcomes**, then it is called a Multi-class Classifier.

Example:

Classifications of **types of crops** - **Rice, Wheat, Pulses, Tea**

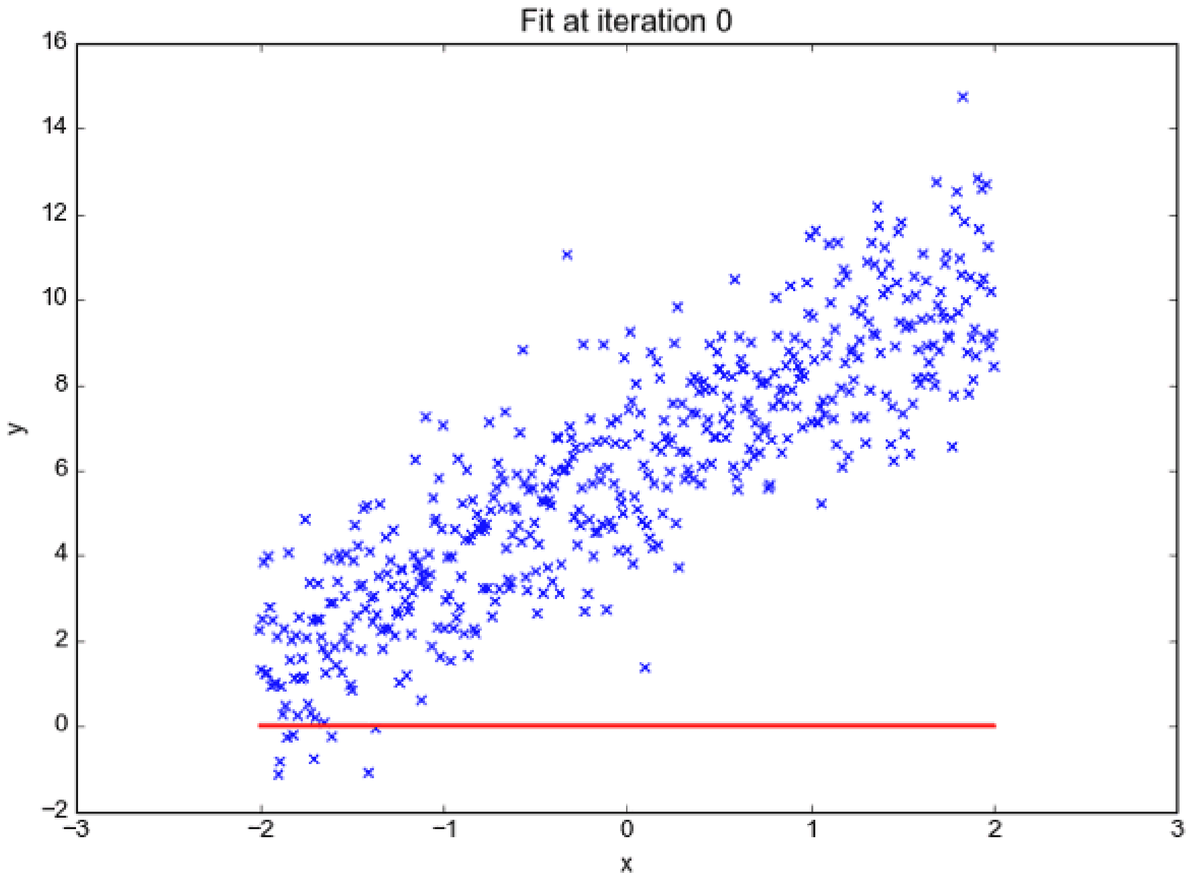
Classification of **types of music**- **Punjabi, Classical, Folk, Indian Rock**

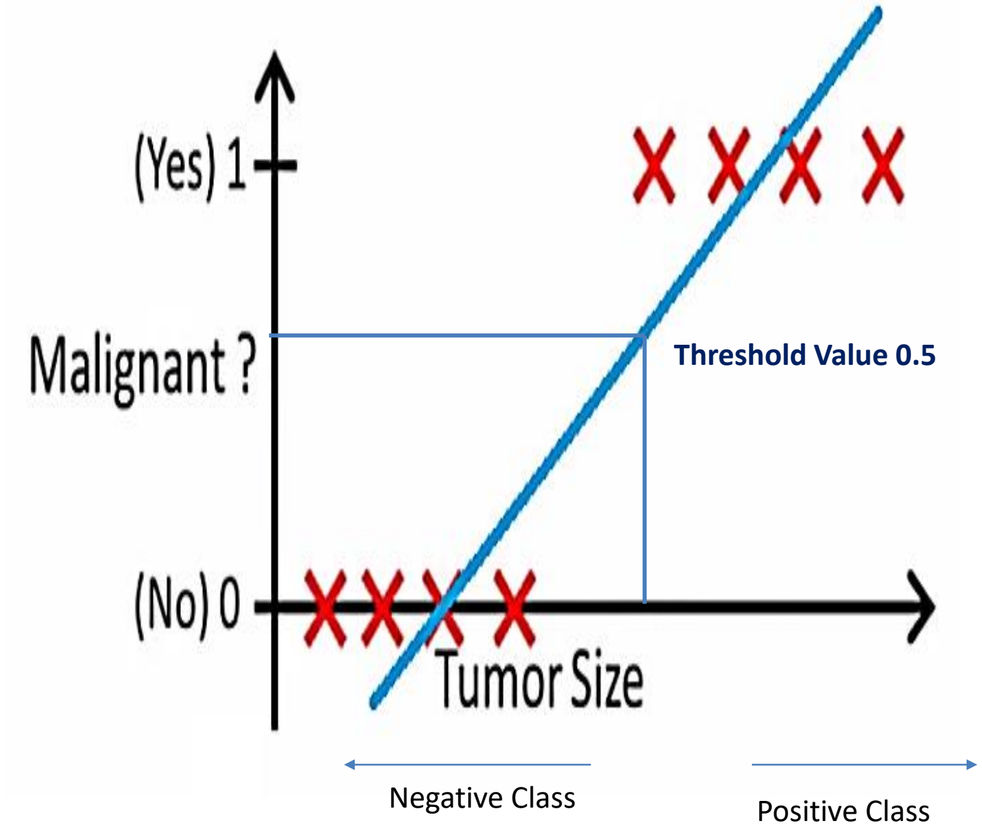
Logistic Regression in Machine Learning



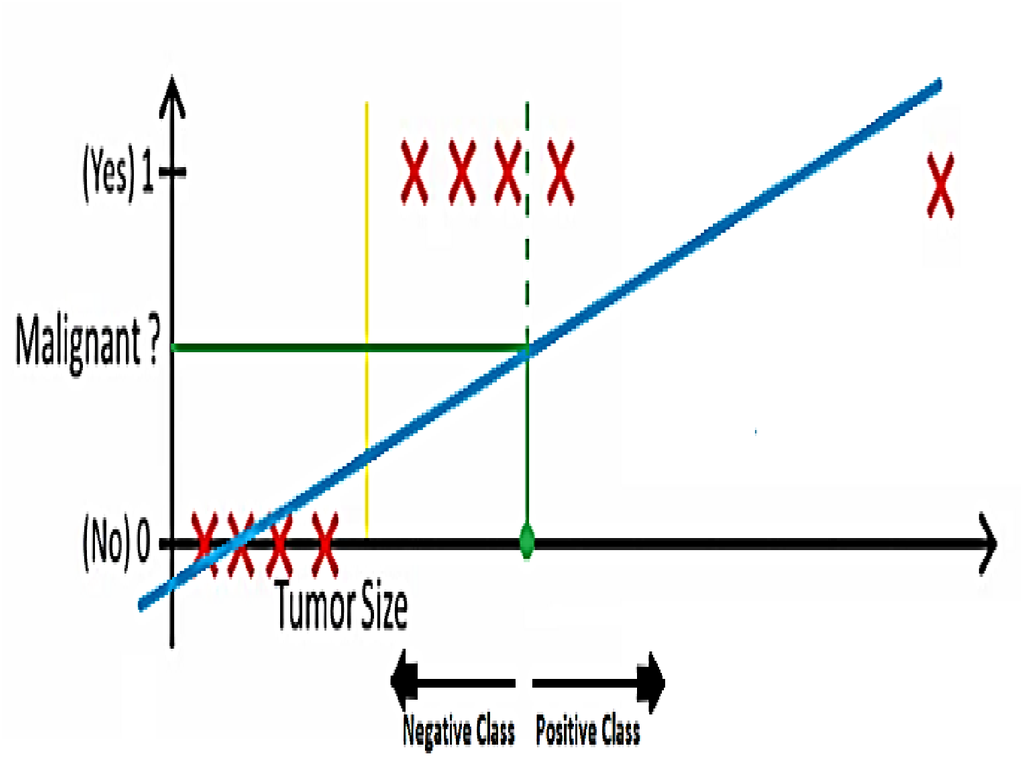
- **Logistic regression** is the most popular supervised learning classification algorithm used to **predict the probability of a target variable**.
- The nature of the target or dependent variable is **Categorical or Discrete** in nature.
- Categorical or discrete variables mean either **Happy or Sad**, **Yes or No, 0 or 1, true or fals**e, etc.
- It is one of the simplest ML algorithms that can be used for various classification problems such as **Spam Detection, Diabetes Prediction, Cancer Detection** etc.

Logistic Regression in Machine Learning



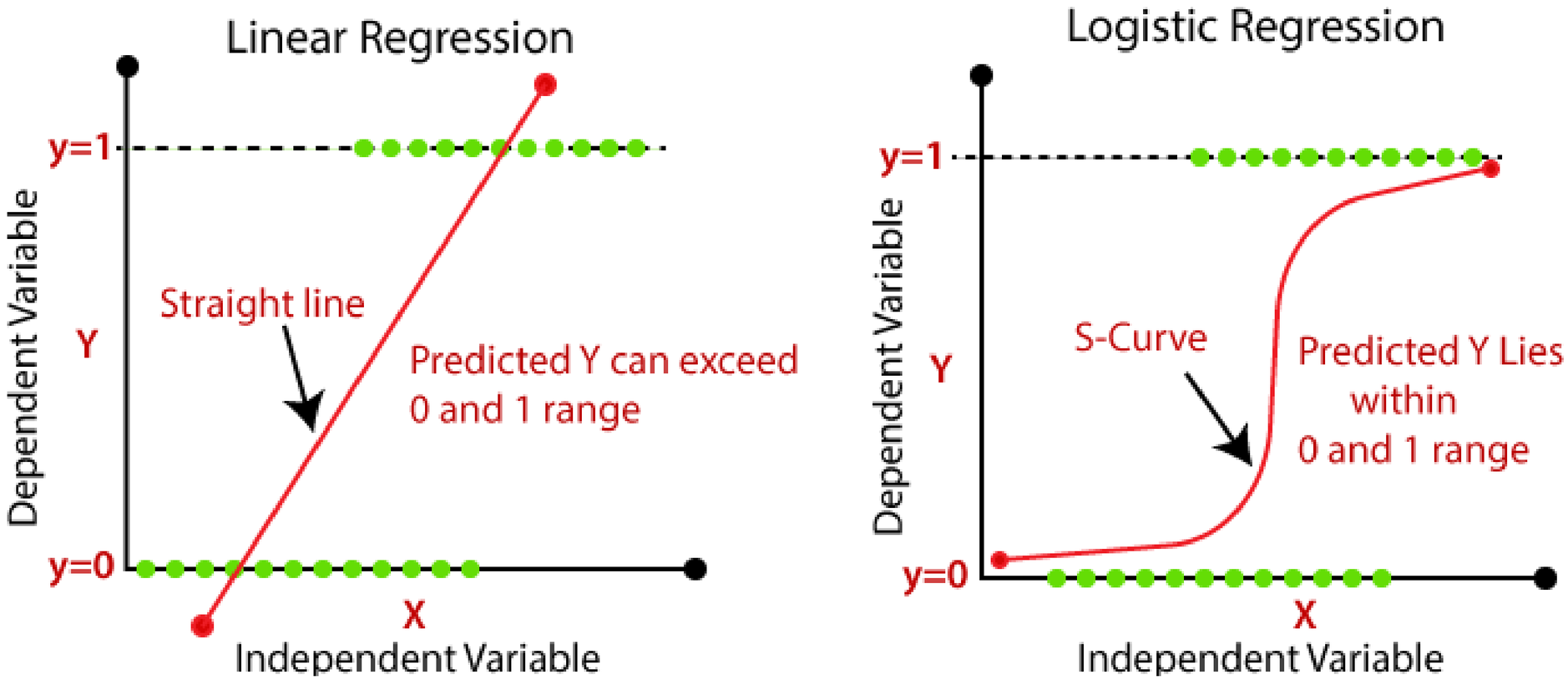


Logistic Regression in Machine Learning



- There is problem when data contain outliers.
- We wont be able to easily decide threshold value for segregation of classes.
- That’s the reason, Logistic Regression Comes into picture to sort such type of issues.

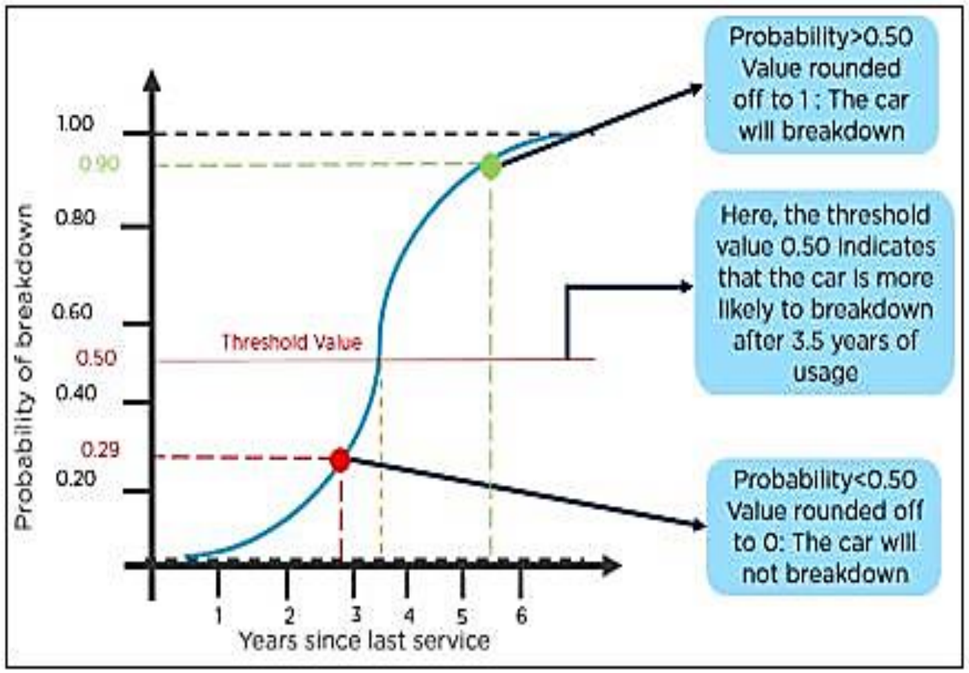
Logistic Regression in Machine Learning



- In Logistic regression, instead of fitting a regression line, we fit an **"S" shaped Curve using Logistic Function (Sigmoid Function)**, which predicts two maximum values (0 or 1).

Logistic Regression in Machine Learning

Sigmoid Function



- The sigmoid function is a mathematical function used to map the predicted values to probabilities.

$$f(x) = \frac{1}{1 + e^{-(x)}}$$

- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression cannot go beyond this limit (must be between 0 and 1), so it forms a **curve like the "S" form**. The S-form curve is called the **Sigmoid function or the logistic function**.
- We use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Since this is a binary classification, logistic regression can be used to build the model.

Dataset source:
https://www.kaggle.com/c/titanic/data

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

#Importing the libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import seaborn as sns
```

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

#Reading the dataset

dataset = pd.read_csv("titanic.csv")

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

#Reading the dataset

```
dataset = pd.read_csv("titanic.csv")
```

Dataset Column Description

- **PassenderId:** PassengerId is the Id given to all the passengers to identify each individual uniquely.
- **Survived:** Survived indicates whether the passenger survived or not (***0 for not survived and 1 for survived***).
- **Pclass:** Passenger class indicates the class a passenger belongs to (1 for 1st class, 2 for 2nd class, and 3 for 3rd class).
- **Name:** Name is the name of the passenger.
- **Sex:** Sex indicates the gender of the passenger.
- **Age:** Age indicates the age of the passenger.
- **SibSp:** SibSp indicates the number of siblings/spouses aboard.
- **Parch:** Parch indicates the number of parents/children aboard.
- **Ticket:** Ticket indicates the ticket number.
- **Fare:** Fare is the passenger fare in pounds.
- **Cabin:** The cabin indicates the cabin number.
- **Embarked:** Embarked indicates port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton).

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Data Pre-Processing

1.Checking for missing values in the dataset

#Checking for missing values

dataset.isnull().sum()

Or

dataset.info()

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

The columns Age, Cabin, and Embarked have missing values.

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Data Pre-Processing

2. Filling missing values in the dataset

#Filling Age column by median

```
dataset["Age"].fillna(dataset["Age"].median(skipna=True), inplace=True)
```

#Fillimg Embarked column by the most common port of embarkation

```
dataset['Embarked'].fillna(dataset['Embarked'].mode()[0], inplace=True)
```

#Dropping the cabin columns

```
dataset.drop('Cabin', axis=1, inplace=True)
```

The columns Age, Cabin, and Embarked have missing values.

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Data Pre-Processing

3. Checking missing values in the dataset

#Checking for missing values

dataset.isnull().sum()

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Data Pre-Processing

4. Dropping unnecessary columns

#Dropping unnecessary columns

```
dataset.drop('PassengerId', axis=1, inplace=True)

dataset.drop('Name', axis=1, inplace=True)

dataset.drop('Ticket', axis=1, inplace=True)
```

The columns PassengerId, Name, and Ticked are unnecessary as they do not affect the target variable, i.e., Survived. Therefore, we can drop those columns from the dataset.

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Data Pre-Processing

4. Dropping unnecessary columns

#Dropping unnecessary columns

```
dataset.drop('PassengerId', axis=1, inplace=True)

dataset.drop('Name', axis=1, inplace=True)

dataset.drop('Ticket', axis=1, inplace=True)
```

The columns PassengerId, Name, and Ticked are unnecessary as they do not affect the target variable, i.e., Survived. Therefore, we can drop those columns from the dataset.

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Data Pre-Processing

Transformation into a categorical column

```
dataset.replace({'Sex':{'male':0,'female':1}, 'Embarked':{'S':0,'C':1,'Q':2}},
inplace=True)
```

Now run the

dataset.head() command again,

we find that the values have been replaced successfully.

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Let’s split the data into the target and feature variables

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, confusion_matrix
```

```
X = dataser.drop(columns = ['PassengerId','Name','Ticket','Survived'],axis=1)

Y = dataser['Survived']
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2, random_state=2)
```

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

Logistic Regression

```
model = LogisticRegression()

model.fit(X_train, Y_train)

Y_pred = log_reg.predict(X_test)

Y_pred
```

Logistic Regression in Machine Learning

Problem statement: The aim is to make predictions on the survival outcome of passengers.

accuracy_score(Y_pred, Y_test)

>> 0.8067796610169492

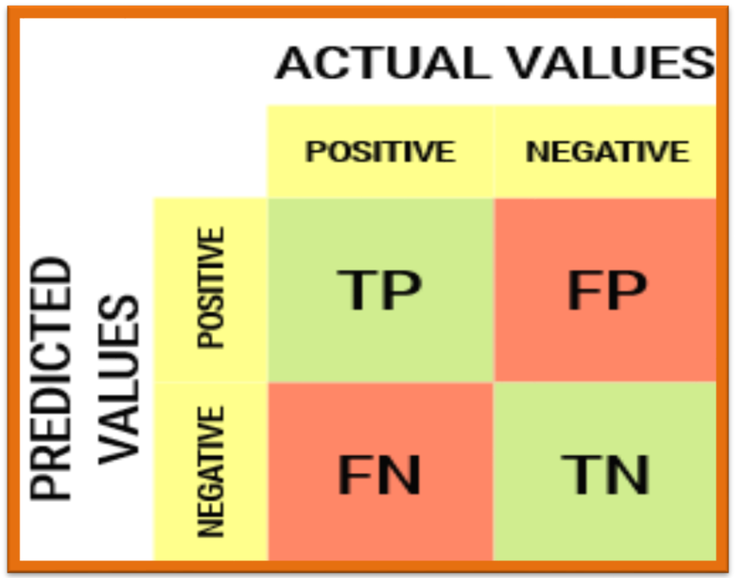
confusion_matrix(Y_pred, Y_test)

>> array([[158, 31],

[26, 80]])

 # 31 + 26 = 57 wrong prediction

2. Confusion Matrix



- **Confusion Matrix (Error Matrix)** is used to measure the **performance of the classification model**.
- The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is represented by confusion matrix.
- It is represented in N x N matrix form where N is the number of target classes. The matrix compares the actual target values with those predicted by the proposed machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

2. Confusion Matrix

The matrix looks like as below table

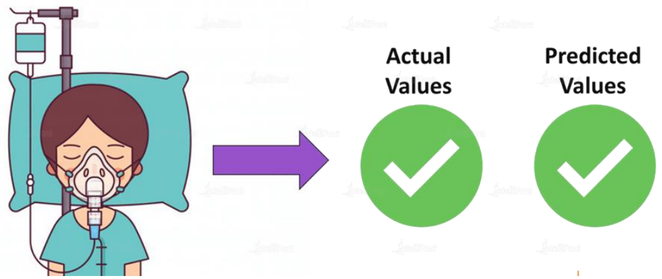
	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP) (Type 1 Error)
Predicted Negative	False Negative (FN) (Type 2 Error)	True Negative (TN)

- **True Positive (TP) -** The actual value was positive and the model predicted a positive value
- **True Negative (TN)** - The actual value was negative and the model predicted a negative value
- **False Positive (FP) – (Type 1 error)** The predicted value was falsely predicted, the actual value was negative but the model predicted a positive value. Also known as the **Type 1 error**
- **False Negative (FN) – (Type 2 error)** The predicted value was falsely predicted, the actual value was positive but the model predicted a negative value. Also known as the **Type 2 error**

2. Confusion Matrix

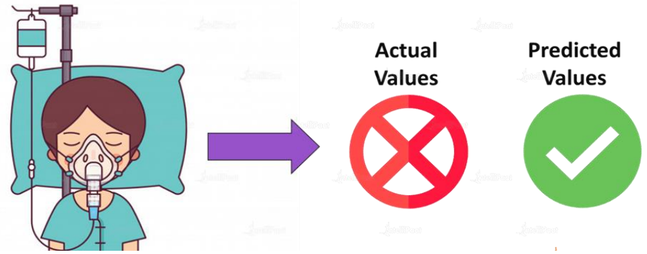
The matrix looks like as below table

	Actual Positive		Actual Negative	
Predicted Positive	True Positive	(TP)	False Positive	(FP)
Predicted Negative	False Negative	(FN)	True Negative	(TN)

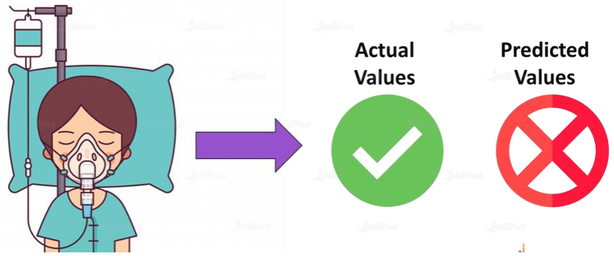


True Positive (TP)

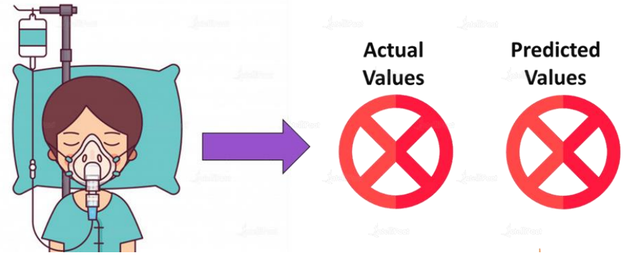
(The actual value was positive and the model predicted a positive value)



False Positive (FP) – Type 1 error
(The predicted value was falsely predicted)



False Negative (FN) (Type 2 error)
The predicted value was falsely predicted.



True Negative (TN)
The actual value was negative and the model also predicted a negative value

2. Confusion Matrix

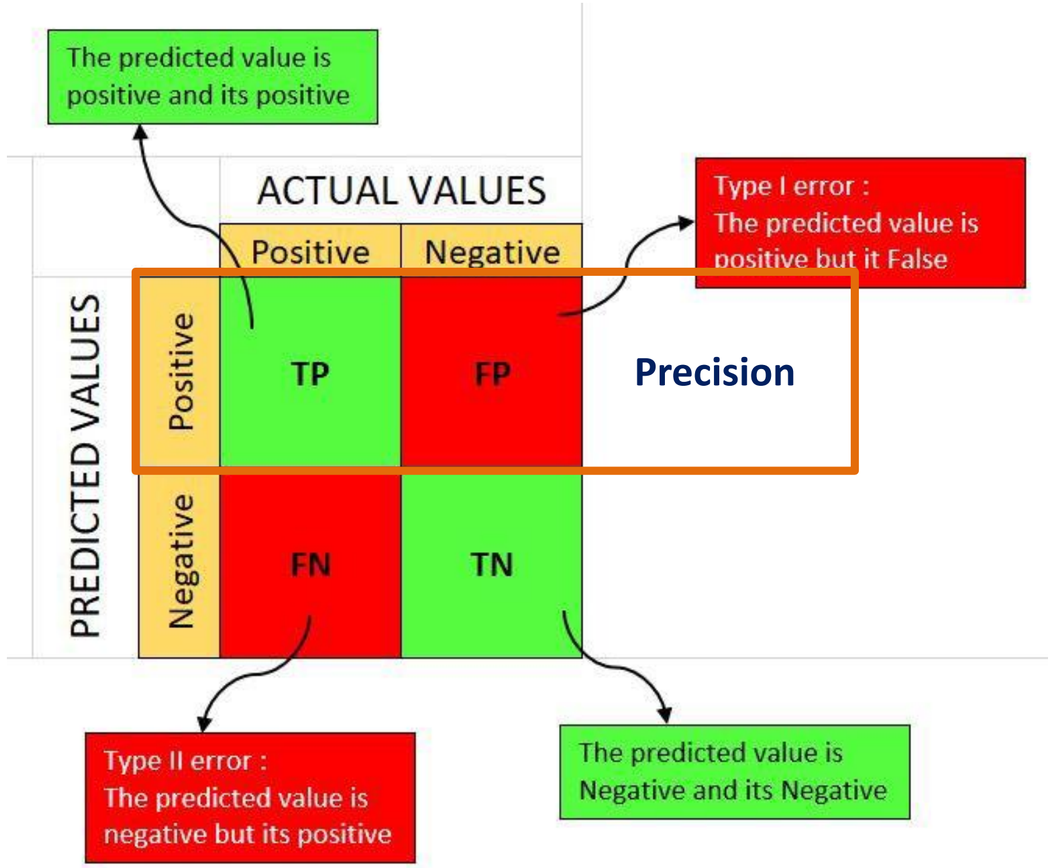
The matrix looks like as below table

	Actual Positive		Actual Negative	
Predicted Positive	True Positive	(TP)	False Positive	(FP)
Predicted Negative	False Negative	(FN)	True Negative	(TN)

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}$$

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

2. Confusion Matrix



Precision: **A**lso called **positive predictive value**, is the fraction of relevant instances among the retrieved (predicted) instances.

Or

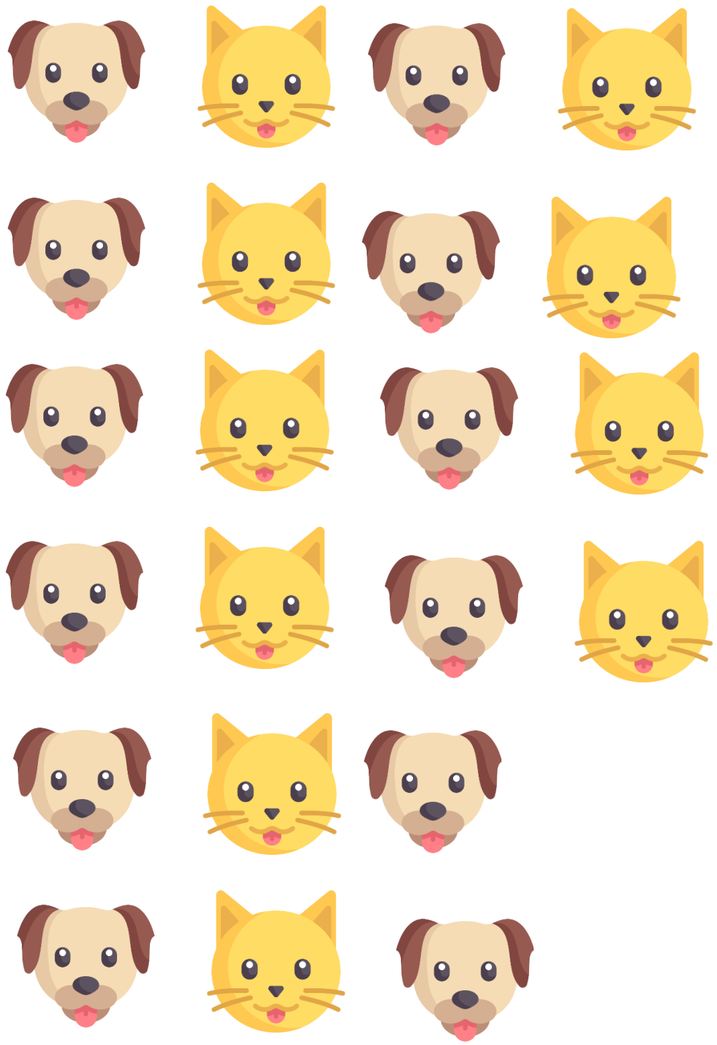
- The proportion of positive cases that were correctly identified or What percent of your predictions were correct?

(This would determine whether our model is reliable or not)

$$\textbf{Precision} = \frac{TP\ (Actual\ relevant\ Instances)}{TP+FP\ (Total\ retrieved\ predicted\ elements)}$$

(It tells how many retrieved (predicted) items are relevant)

2. Confusion Matrix (Understanding Precision)



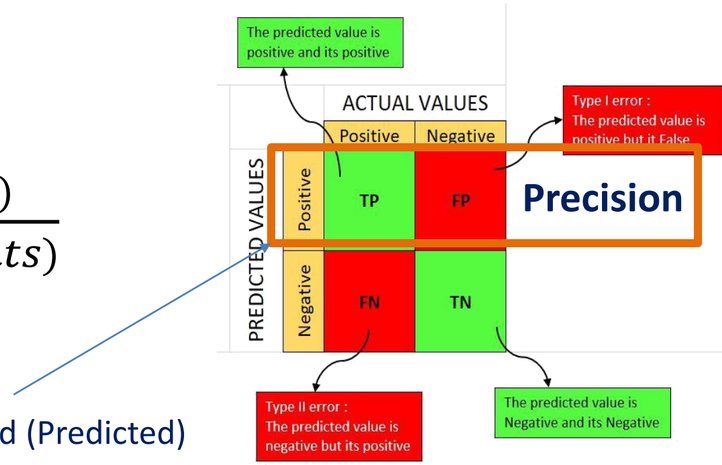
Precision: is the fraction of relevant instances among the retrieved instances.

Example

Consider a Machine Learning Model for **recognizing dogs** (the **relevant** element) in a digital photograph. It contains **ten cats and twelve dogs**.

After processing the Machine Learning Model, **it identifies eight dogs**. Of the eight elements identified as dogs, **only five actually are dogs** (true positives or relevant instances), then what is Precision?

$$\textbf{Precision} = \frac{5}{8} = \frac{TP\ (relevant\ Instances)}{TP+FP\ (retrieved\ elements)}$$



Total Retrieved (Predicted) Elements

Note- Precision TALKS about VALIDITY of the Model

Assessment

10

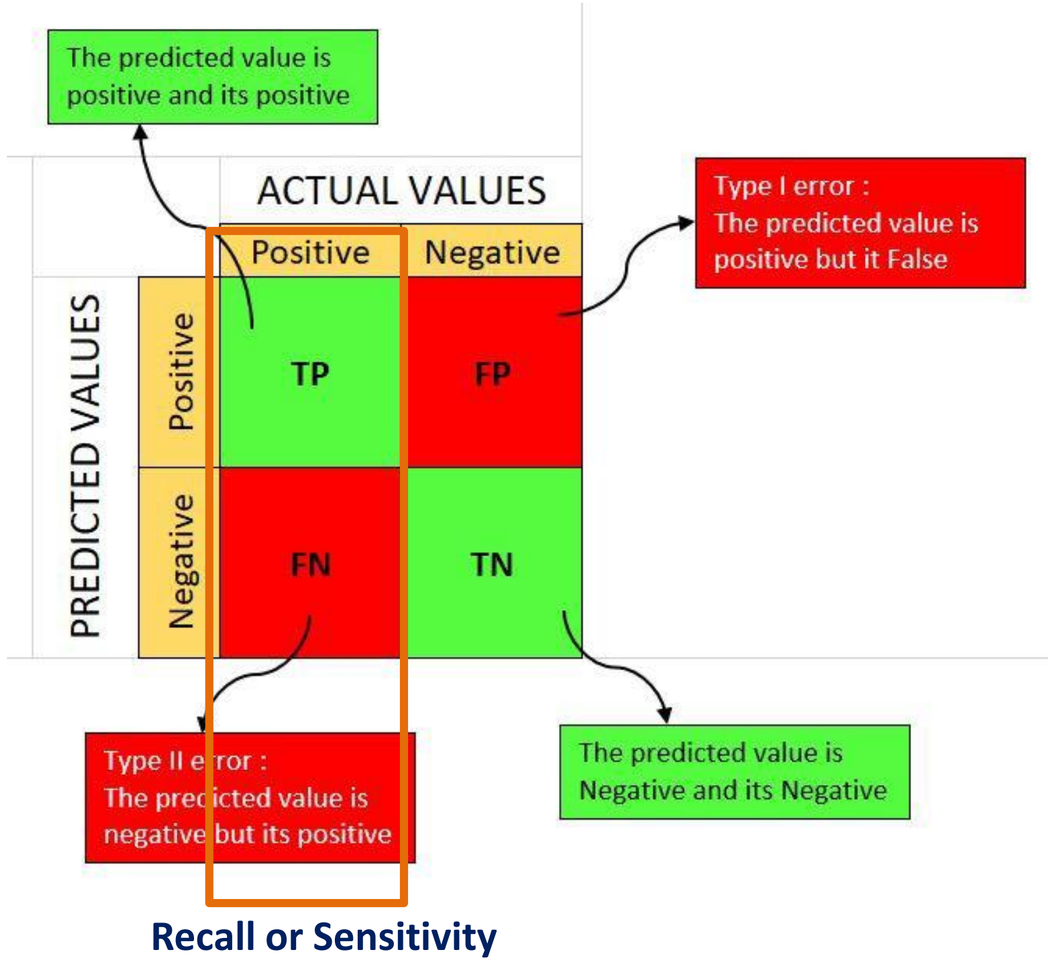
When you type a particular Queries on Google Search Engine, it returns 30 pages in total, in which only 20 of pages are relevant, then what is the precision of the model?

1. 3/2
2. 2/3
3. 6/9
4. B & C are Correct

Answer- 4

~Dr Gaurav Kumar, Asst. Prof, CEA, GLA University

2. Confusion Matrix



Recall: **recall** (also known as **sensitivity**) is the fraction of relevant instances that were retrieved

Or

- It tells us how many of the actual positive cases we were able to predict correctly with our model. (**Also known as True Positive Rate**)

  or

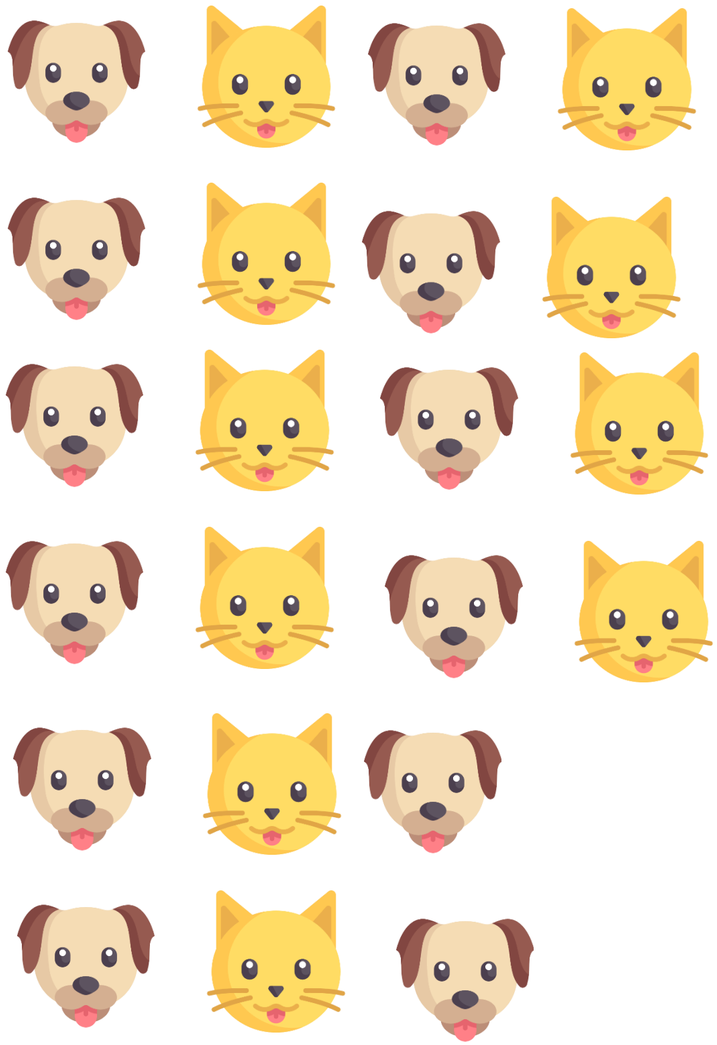
- What percent of the positive cases did you catch?

  (The proportion of actual positive cases which are correctly identified)

$$\textbf{Recall} = \frac{TP\ (Actual\ relevant\ predicted\ Instances)}{TP+FN\ (Total\ Relevant\ elements)}$$

(It tells how many relevant items are retrieved)

2. Confusion Matrix (Understanding Recall)

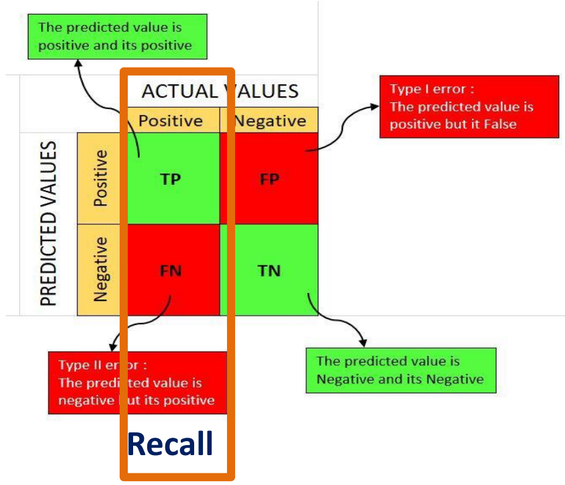


Example

Consider a Machine Learning Model for recognizing dogs (the **relevant** element) in a digital photograph. It contains **ten cats and twelve dogs**.

After processing the Machine Learning Model, **it identifies eight dogs**. Of the eight elements identified as dogs, **only five actually are dogs** (true positives or relevant instances), while the **other three are cats (false positives). Seven dogs were missed (false negatives)**, and **seven cats were correctly excluded (true negatives).** then what is Recall?

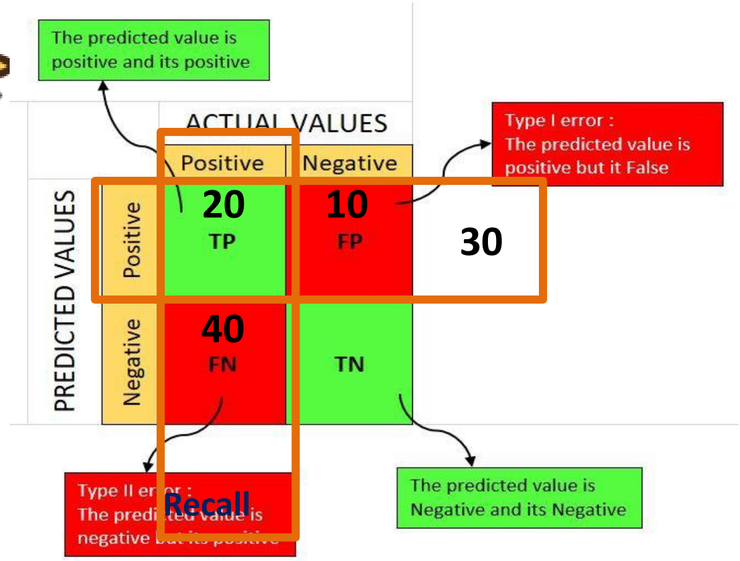
$$\textbf{Recall} = \frac{5}{12} = \frac{true\ positives\ (TP)}{TP+FN\ (Total\ Relevant\ elements)}$$



Note- Recall TALKS about COMPLETENESS or COVERAGE of the Model

Assessment

10



When you type a particular Queries on Google Search Engine, it returns 30 pages in total, in which only 20 of pages are relevant, and failed to return 40 additional relevant pages, then what is the Recall of the model?

1. 2/3
2. 2/7
3. 1/3
4. None of the Above

Answer- 3

~Dr Gaurav Kumar, Asst. Prof, CEA, GLA University

2. Confusion Matrix

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

- **Negative Predicative Value (False Negative Rate):** The proportion of negative cases that were correctly identified.
- **Specificity:** The proportion of actual negative cases which are correctly identified. **(True Negative Rate)**

Understanding the Need of Confusion Matrix

Let’s say you want to predict how many people are infected with a **Covid-19 virus** in times before they show the symptoms, and isolate them from the healthy population (ringing any bells, yet?). The two values for our target variable would be: **Sick (1) and (Healthy) Not Sick(0).**

Understanding the Need of Confusion Matrix

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

There are 947 data points for the negative class (Not Sick) and 53 data points for the positive class (Sick).

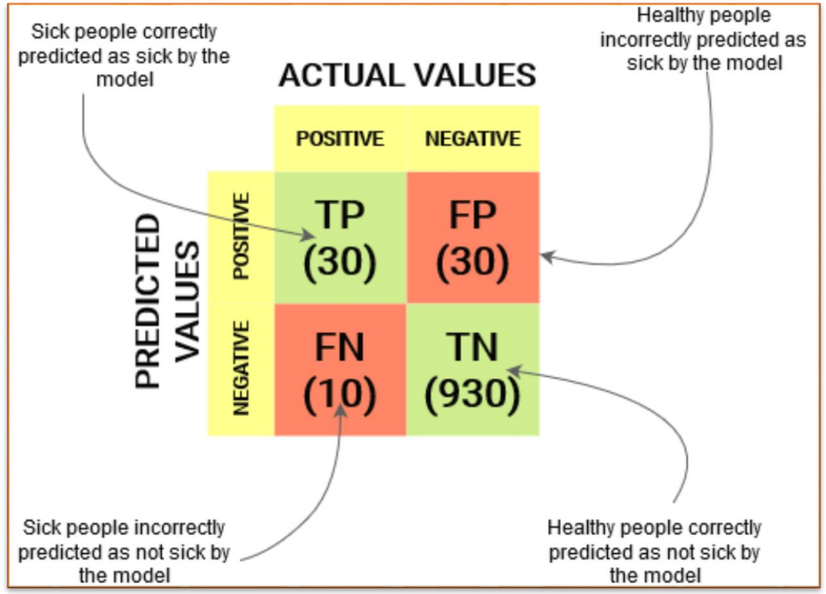
Now, you must be wondering – why do we need a confusion matrix when we have our all-weather friend – **Accuracy?**

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}$$

Well, let's see where accuracy falters.

Understanding the Need of Confusion Matrix

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN



		Actual Value	
		Sick (1)	Not Sick (0)
Predicted Value	Sick (1)	30	30
	Not Sick (0)	10	930

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

$$Accuracy = \frac{30 + 930}{30 + 30 + 930 + 10} = 0.96$$

Understanding the Need of Confusion Matrix

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

So, the accuracy for our model turns out to be: 96%

Not Bad!

		Actual Value	
		Sick (1)	Not Sick (0)
Predicted Value	Sick (1)	30	30
	Not Sick (0)	10	930

Q1- Can we rely on this accuracy? Do you think this is a correct metric for our model given the seriousness of the issue?

Q2- Does Accuracy tell any information about Covid Spread?

Understanding the Need of Confusion Matrix

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

So, the accuracy for our model turns out to be: 96%

Not Bad!

		Actual Value	
		Sick (1)	Not Sick (0)
Predicted Value	Sick (1)	30	30
	Not Sick (0)	10	930

- Shouldn't we be measuring how many positive cases we can predict correctly to arrest the spread of the covid-19 virus?

i.e. **sick people** should not be predicted **not sick**, accuracy?

Understanding the Need of Confusion Matrix

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

So, the accuracy for our model turns out to be: 96%

		Actual Value	
		Sick (1)	Not Sick (0)
Predicted Value	Sick (1)	30 (TP)	30 (FP)
	Not Sick (0)	10 (FN)	930 (TN)

- **Reliability of the model (Precision)** = 30/60 * 100 = 50%
- **Sensitivity of the model (Recall, Accuracy of Sick)** = 30/40 * 100= 75%

Understanding the Need of Confusion Matrix

ID	Actual Sick?	Predicted Sick?	Outcome
1	1	1	TP
2	0	0	TN
3	0	0	TN
4	1	1	TP
5	0	0	TN
6	0	0	TN
7	1	0	FP
8	0	1	FN
9	0	0	TN
10	1	0	FP
:	:	:	:
1000	0	0	FN

So, the accuracy for our model turns out to be: 96%

- Recall is important in medical cases where it doesn't matter whether we raise a false alarm but the actual positive cases should not go undetected.
- In our example, Recall would be a better metric because we don't want to accidentally discharge an infected person and let them mix with the healthy population thereby spreading the Covid-19 virus.
- Now you can understand why accuracy was a bad metric for our model.
- But there will be cases where there is no clear distinction between whether Precision is more important or Recall. What should we do in those cases?

We combine them!

F1 SCORE

$$F1 - score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

- F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is.

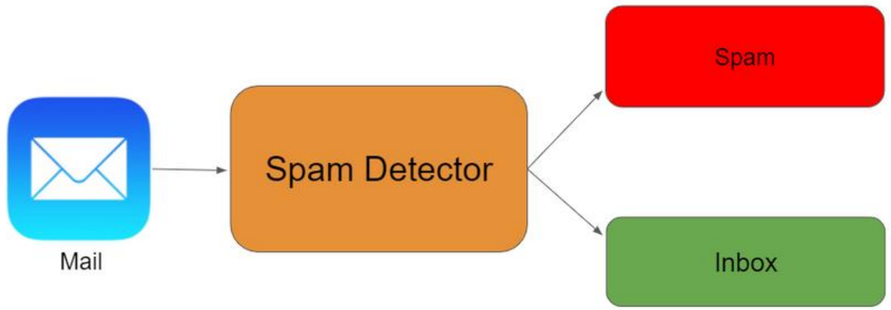
- In Short, it tells What percent of positive predictions were correct?

  F1 Score = 2*(Recall * Precision) / (Recall + Precision)

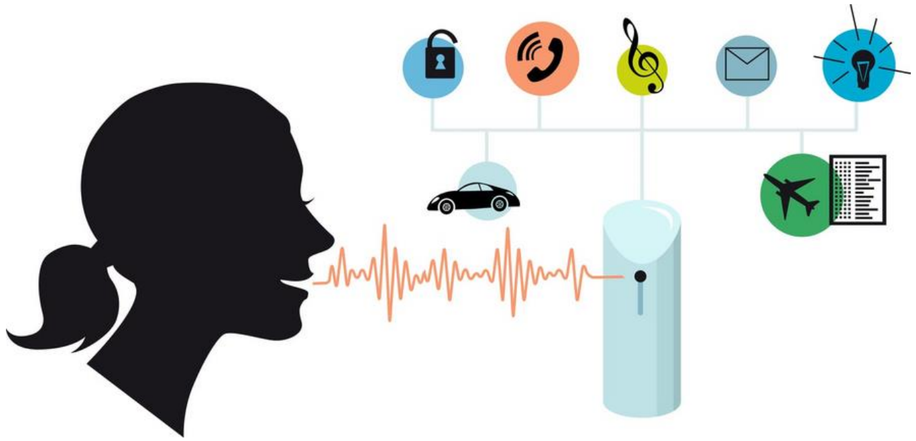
- The greater the F1 Score, the better is the performance of our model.

- F1 should be used **to compare classifier models, not global accuracy**

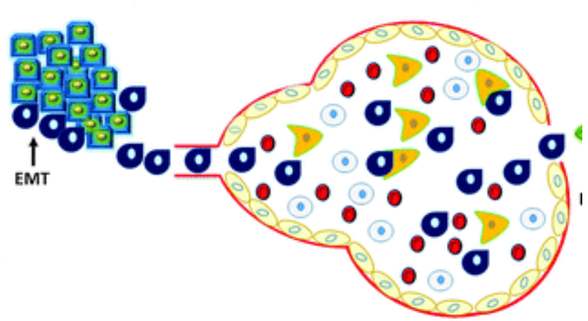
Some Popular Use cases of Classification Algorithms



Email Spam Detection



Speech Recognition



Identifications of Cancer tumor cells

Drugs Classification

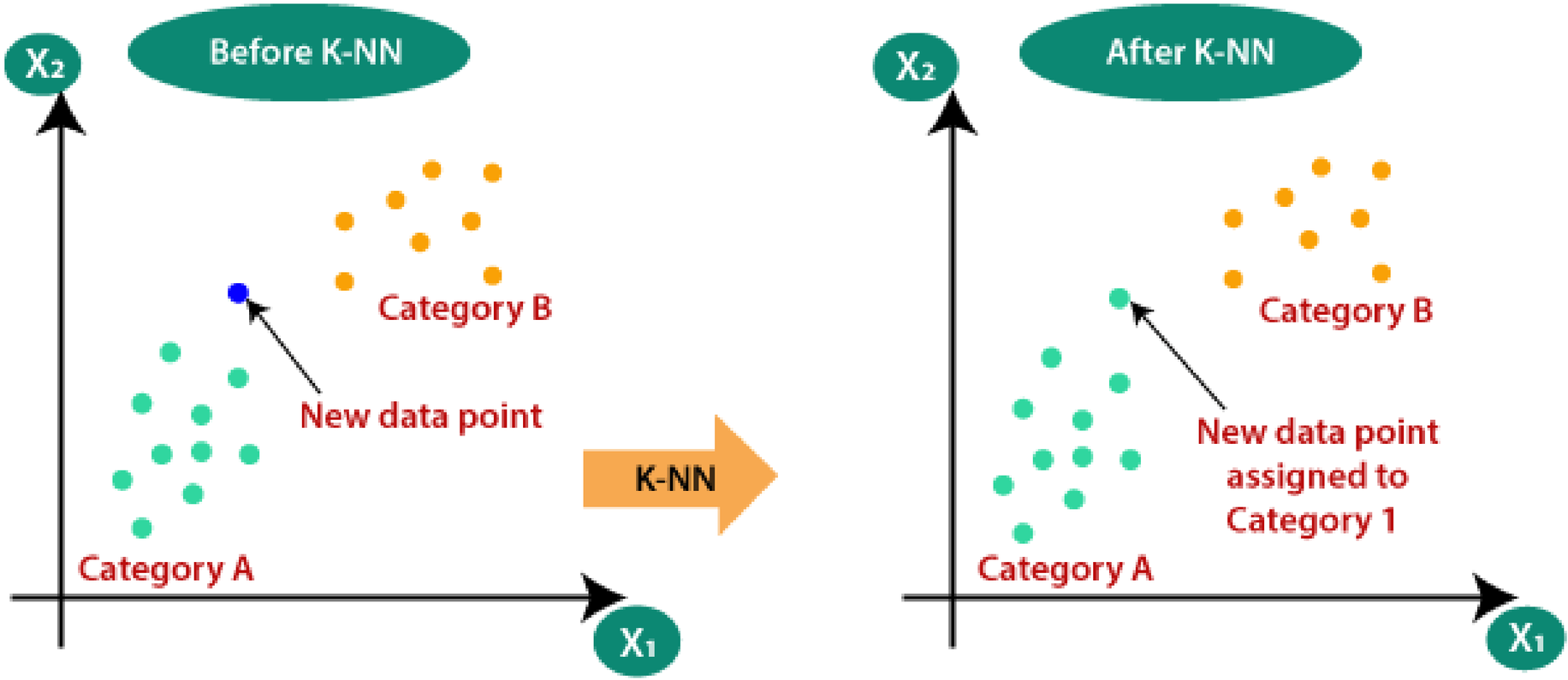
Biometric Identification,.

~Dr Gaurav Kumar, Asst. Prof, CEA, GLA University

K-Nearest Neighbor (KNN) Algorithm for Machine Learning

Another Simple Supervised Machine Learning algorithms to solve classification problems.

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.



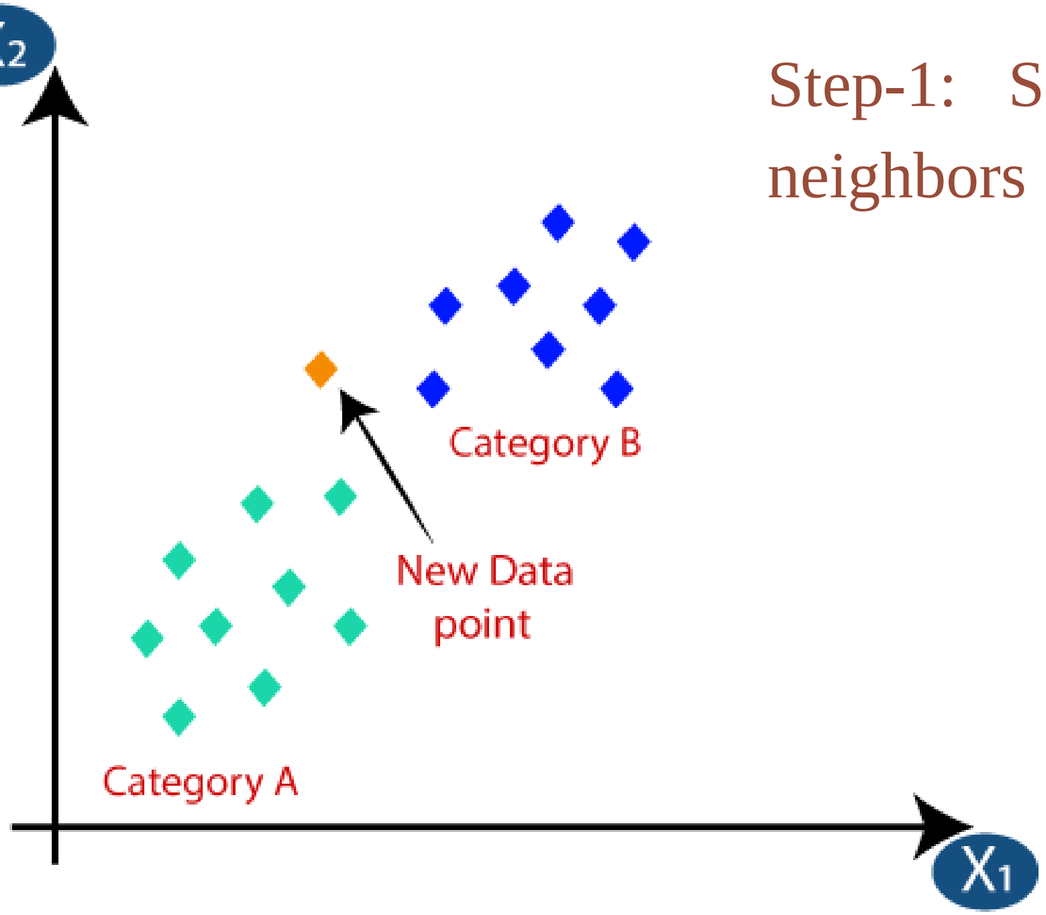
~Dr Gaurav Kumar, Asst. Prof, CEA, GLA University

Example of K-Nearest Neighbor (KNN) Algorithm

Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog.

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.

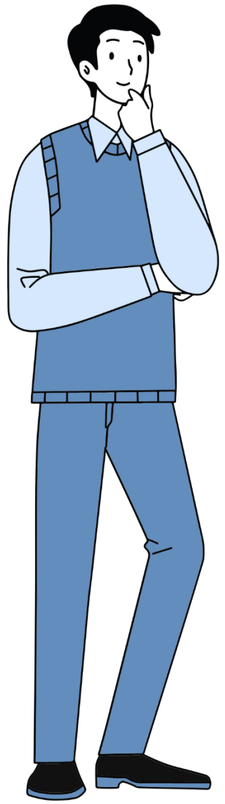
How does K-NN work?



Step-1: Select the number K of the neighbors

How to select the value of K in the K-NN Algorithm?

Some points to be remembered while selecting the value of K in the K-NN algorithm:



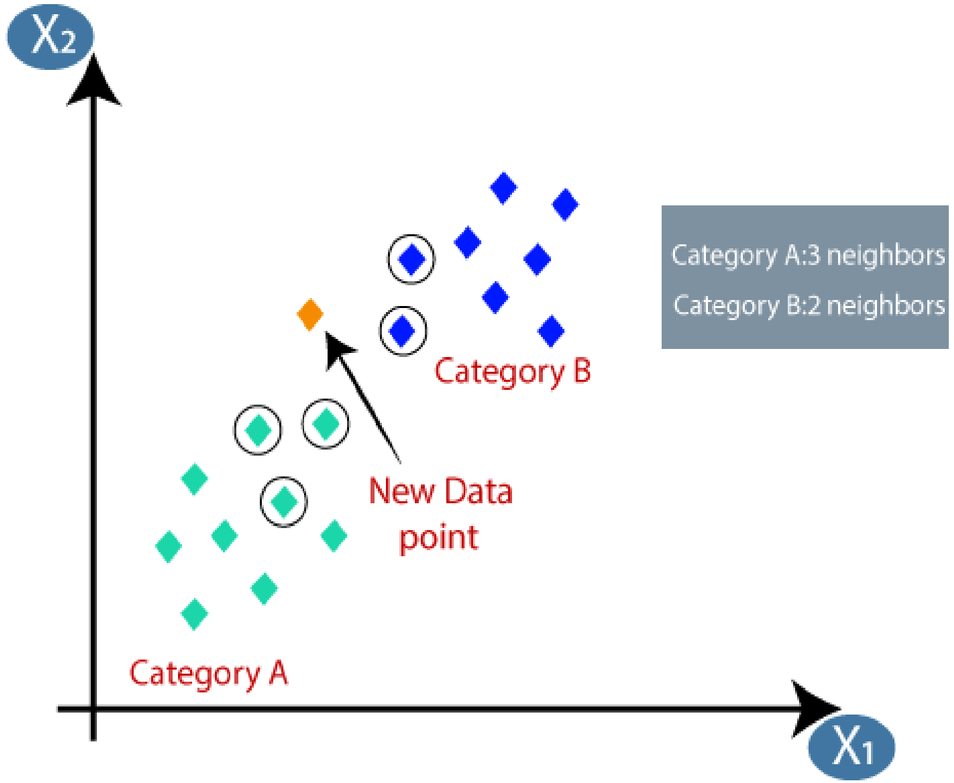
1. There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.

2. 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.

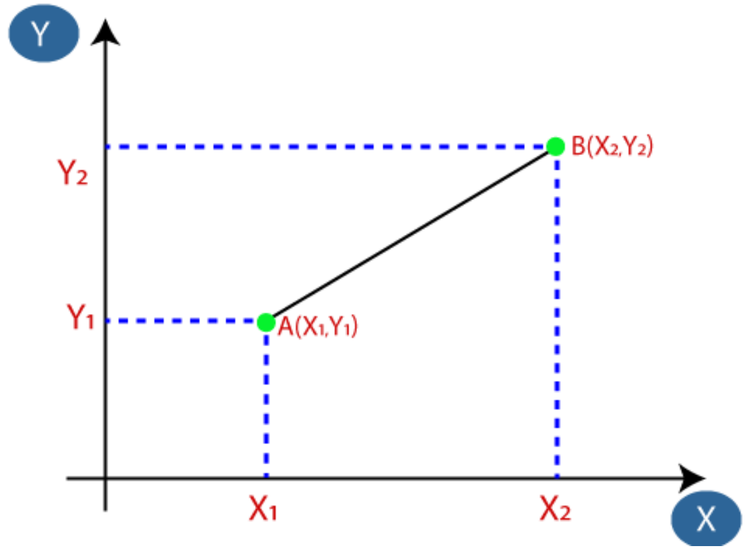
3. Large values for K are good, but it may find some difficulties in terms of time and resource cosumptions

How does K-NN work?

Step-2: Consider some neighbors and calculate Euclidean distance.



Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$



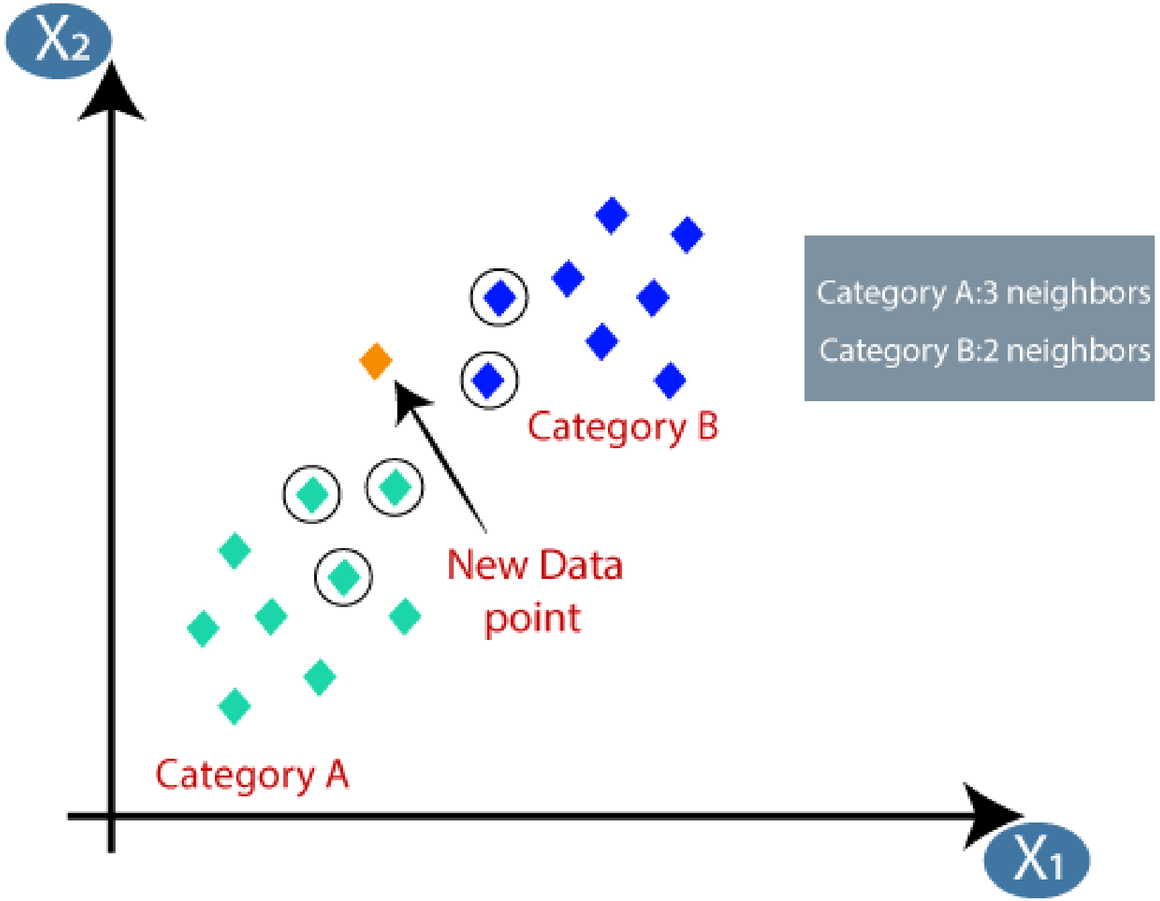
Step-3: Among these k neighbors, count the number of the data points in each category.

How does K-NN work?

Step-4: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready

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



Advantages and Disadvantages of KNN

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 sometimes.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

GLA UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)
Accredited with A Grade by NAAC

THANKYOU