Various approaches to Machine Learning

Modeling

Definition: Modeling is the process of creating a simplified representation of a real-world system or phenomenon to understand, predict, or simulate its behavior. Models capture essential features of the system while abstracting away unnecessary details.

Purpose: Models are used to gain insights, make predictions, optimize decisions, and understand complex systems across various domains, including science, engineering, economics, and business.

Components of a Model:

Variables: Represent factors or attributes that influence the system's behavior.

Parameters: Values that define the model's structure and relationships between variables.

Equations/Functions: Mathematical expressions that describe how variables interact and evolve over time.

Types of Models: Models can be categorized into various types based on their complexity, purpose, and representation, such as mathematical models, statistical models, physical models, and computational models.

Machine Learning

Machine learning is a subset of artificial intelligence (AI) that focuses on developing algorithms and techniques that enable computers to learn from data and make predictions or decisions without being explicitly programmed.

***Key Components of Machine Learning*:**

- ***Data**:* Raw information used for training, testing, and evaluating machine learning models.
- ***Features**:* Input variables or attributes used to make predictions or classifications.
- ***Labels/Targets**:* Output variables representing the desired prediction or classification.
- ***Model**:* Algorithm or computational representation that learns patterns and relationships from data to make predictions or decisions.

Types of Machine Learning

Supervised Learning: Models learn from labeled data, making predictions or classifications based on input-output pairs.

Unsupervised Learning: Models learn patterns and structures from unlabeled data, identifying hidden relationships or clusters.

Reinforcement Learning: Models learn through trial and error by interacting with an environment and receiving feedback/rewards.

Steps in Machine Learning

- **Data Collection:** Gathering relevant data from various sources, ensuring data quality and integrity.
- **Data Preprocessing**: Cleaning, transforming, and preparing the data for analysis, including handling missing values, encoding categorical variables, and scaling numerical features.
- **Model Selection**: Choosing the appropriate machine learning algorithm or technique based on the problem type, data characteristics, and performance requirements.
- **Model Training**: Using the training data to fit the model parameters and learn patterns from the data.

- **Model Evaluation**: Assessing the model's performance on unseen data to measure its accuracy, precision, recall, F1-score, etc.
- **Model Deployment**: Deploying the trained model into production systems for making predictions or decisions on new, unseen data.
- **Model Monitoring and Maintenance**: Continuously monitoring the model's performance in real-world applications and updating it as needed to ensure its effectiveness and reliability.

Regression Analysis

Regression analysis is a statistical method used to model the relationship between one or more independent variables (features) and a dependent variable (target). It aims to predict the value of the dependent variable based on the values of the independent variables.

Key Concepts:

- **Dependent Variable (Target):** The variable we want to predict or explain.
- **Independent Variables (Features):** The variables used to predict the dependent variable.
- **Regression Line:** A line that best fits the data points and represents the relationship between the independent and dependent variables.

***Linear Regression*:**

Linear regression is a simple and widely-used regression technique that models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data.

The mathematical equation for Linear regression

$$Y = aX + b + \varepsilon$$

Here, **Y** = dependent variables (target variables),
X= Independent variables (predictor variables),
ε = Error term (residuals)
a and **b** are the linear coefficients.

Steps to Create a Simple Linear Regression Model

1. **Data Collection:** Gather a dataset with paired observations of the independent and dependent variables.
2. **Data Preprocessing:** Clean the data, handle missing values, and split the dataset into training and testing sets.
3. **Model Training:** Use the training data to fit the linear regression model by estimating the coefficients (a and b).
4. **Model Evaluation:** Evaluate the model's performance using metrics such as mean squared error (MSE) or R-squared (R^2) on the testing data.
5. **Prediction:** Make predictions on new data using the trained model.

***Applications and Examples of Regression*:**

1.Predictive Analytics:

Example: Predicting house prices based on features like size, number of bedrooms, and location using linear regression.

2.Financial Analysis:

Example: Forecasting stock prices based on historical data and market indicators using regression models.

3.Marketing and Sales:

Example: Estimating sales revenue based on advertising spend, promotions, and other marketing factors using regression analysis.

4.Healthcare:

Example: Predicting patient readmission rates based on demographics, medical history, and treatment outcomes using regression techniques.

5.Economics and Social Sciences:

Example: Analyzing the relationship between income and education level using linear regression to understand socioeconomic trends.

Model evaluation methods

1. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average absolute difference between the predicted values and the actual values. It gives us a sense of how close the predictions are to the actual values on average.

Formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- n is the number of samples.
- y_i is the actual value.
- $\hat{y}_i$ is the predicted value.

How it works:

1. Calculate the absolute differences between each predicted value and its corresponding actual value.

2. Take the average of these absolute differences to get the MAE.

Example:

Suppose we have the following actual and predicted values:

- Actual values: [10, 20, 30, 40, 50]
- Predicted values: [12, 18, 25, 35, 48]

Calculating MAE:

$$MAE = \frac{|10-12|+|20-18|+|30-25|+|40-35|+|50-48|}{5}$$

$$MAE = \frac{2+2+5+5+2}{5} = \frac{16}{5} = 3.2$$

2. Mean Squared Error (MSE)

Mean Squared Error (MSE) measures the average of the squares of the differences between predicted values and actual values. It penalizes larger errors more heavily than smaller ones.

Formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where the terms are the same as in MAE.

How it works:

1.Calculate the squared differences between each predicted value and its corresponding actual value.

2. Take the average of these squared differences to get the MSE.

Example:

Using the same actual and predicted values as before:

$$MSE = \frac{(10-12)^2+(20-18)^2+(30-25)^2+(40-35)^2+(50-48)^2}{5}$$

$$MSE = \frac{4+4+25+25+4}{5} = \frac{62}{5} = 12.4$$

3. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is the square root of the MSE. It is a popular evaluation metric as it represents the standard deviation of the residuals.

Formula:

$$RMSE = \sqrt{MSE}$$

Example:

Continuing with the same example:

$$RMSE = \sqrt{12.4} \approx 3.52$$

Notes

- MAE measures the average absolute difference between predicted and actual values.
- MSE measures the average squared difference between predicted and actual values.
- RMSE is the square root of the MSE, representing the standard deviation of the residuals.

R-squared (R^2)

The coefficient of determination, often referred to as R-squared (R^2), is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It indicates the goodness of fit of the model: how well the independent variables explain the variability in the dependent variable.

$$R^2 = 1 - \frac{Squares\ of\ residuals\ (SSR)}{Sum\ of\ squares\ (SST)} = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y})^2}$$

Where :-

- y_i represents the observed value for the i^{th} observation.
- $\bar{y}$ represents the mean of the dependent variable.
- $\hat{y}_i$ represents the predicted value for the i^{th} observation.

How it works:

- Calculate the total sum of squares (SST) by ***summing the squared differences between each actual value and the mean of actual values***.

- Calculate the sum of by ***summing the squared differences between each actual value and its corresponding predicted value***.
- Calculate R^2 using the formula.

Example:-

Step 1: Calculate the Mean of Observed Values ($\bar{y}$):

$$\bar{y} = \frac{10+20+30+40+50}{5} = \frac{150}{5} = 30$$

Step 2: Calculate SSR (Regression Sum of Squares):

$$SSR = \sum(\hat{y}_i - y)^2$$

$$SSR = (12-30)^2 + (18-30)^2 + (25-30)^2 + (35-30)^2 + (48-30)^2$$
$$SSR = 324 + 144 + 25 + 25 + 324 = 842$$

Step 3: Calculate SST (Total Sum of Squares):

$$SST = \sum(y_i - \bar{y})^2$$

$$SST = (10-30)^2 + (20-30)^2 + (30-30)^2 + (40-30)^2 + (50-30)^2$$
$$SST = 400 + 100 + 0 + 100 + 400 = 1000$$

Step 4: Calculate R^2 using SSR and SST:

$$R^2 = 1 - \frac{SSR}{SST}$$

$$R^2 = 1 - \frac{842}{1000} = 1 - 0.842 = 0.158$$

1.Mean Absolute Error (MAE):

Significance: MAE provides a measure of the average absolute difference between the predicted values and the actual values.

Use: MAE is straightforward to interpret and provides a clear indication of the model's accuracy. It is useful when the absolute error is critical and outliers should not be heavily penalized.

2.Mean Squared Error (MSE):

Significance: MSE measures the average squared difference between the predicted values and the actual values.

Use: MSE is commonly used in optimization algorithms as it is differentiable and provides a continuous loss function. It penalizes larger errors more heavily than smaller ones, making it suitable for applications where larger errors are more problematic.

3. Root Mean Squared Error (RMSE):

- **Significance:** RMSE is the square root of the MSE and provides a measure of the average magnitude of errors in the same units as the dependent variable.
- **Use:** RMSE is particularly useful when the absolute scale of errors matters, as it provides a more interpretable measure compared to MSE. It is commonly used to compare the performance of different models or to communicate the model's accuracy to stakeholders.

4. Coefficient of Determination (R^2):

- **Significance:** R^2 measures the proportion of the variance in the dependent variable that is explained by the regression model.
- **Use:** R^2 provides insights into how well the regression model fits the data. It ranges from 0 to 1, where 1 indicates a perfect fit and 0 indicates no relationship between the independent and dependent variables. R^2 is particularly useful for understanding the goodness of fit of the model and comparing different models.

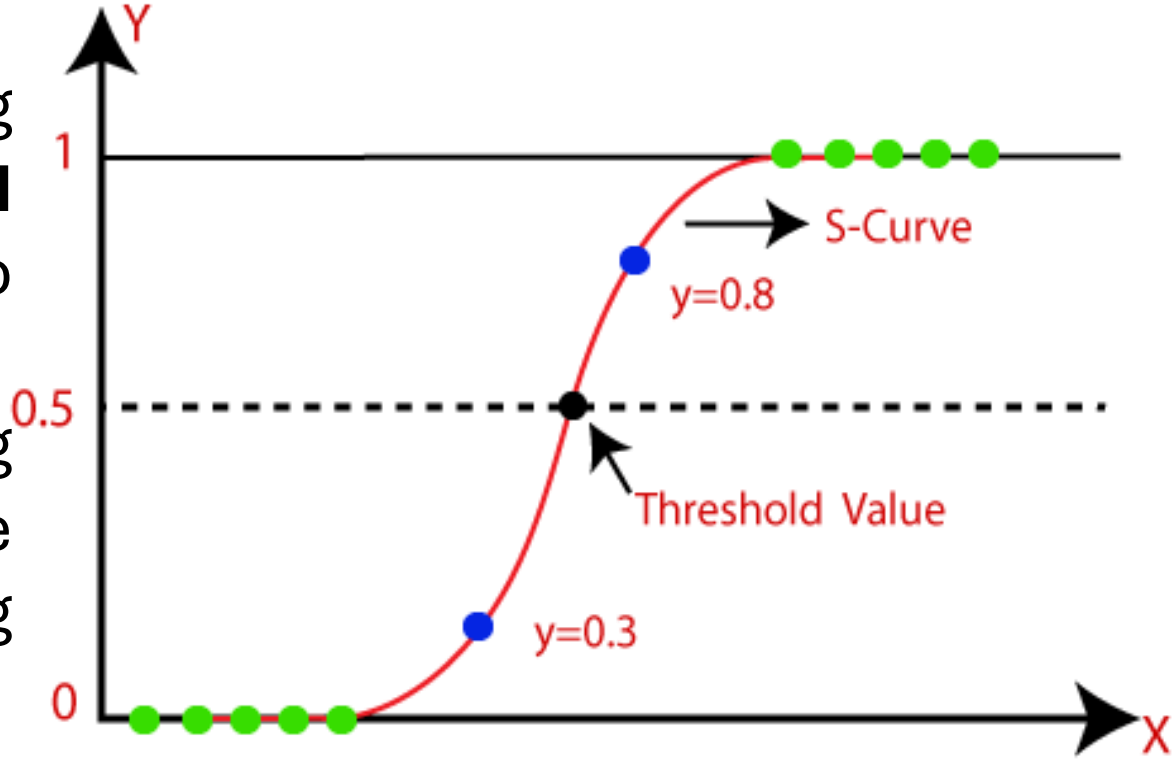
Logistic Regression:

Logistic regression is a Machine Learning algorithm, which comes under the **Supervised Learning technique**. It is used for predicting the **categorical dependent** variable using a given set of independent variables.

Logistic regression **predicts the output of a categorical** dependent variable. Therefore, 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).

It's a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using **continuous and discrete datasets**.



Equation for Logistic Regression

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

Type of Logistic Regression

Based on the categories, Logistic Regression can be classified into **three** types:

***Binomial*:** In binomial Logistic regression, there can be only two possible types of dependent variables, **such as 0 or 1, Pass or Fail, etc.**

Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as **"cat", "dogs", or "sheep"**

***Ordinal*:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as **"low", "Medium", or "High".**

Regression and Classification

Feature	Regression	Classification
Target Variable	Continuous numerical value	Categorical variable with discrete classes
Goal	Predict the magnitude of the target variable	Classify data points into predefined categories
Output	Continuous value on a spectrum	Discrete class label

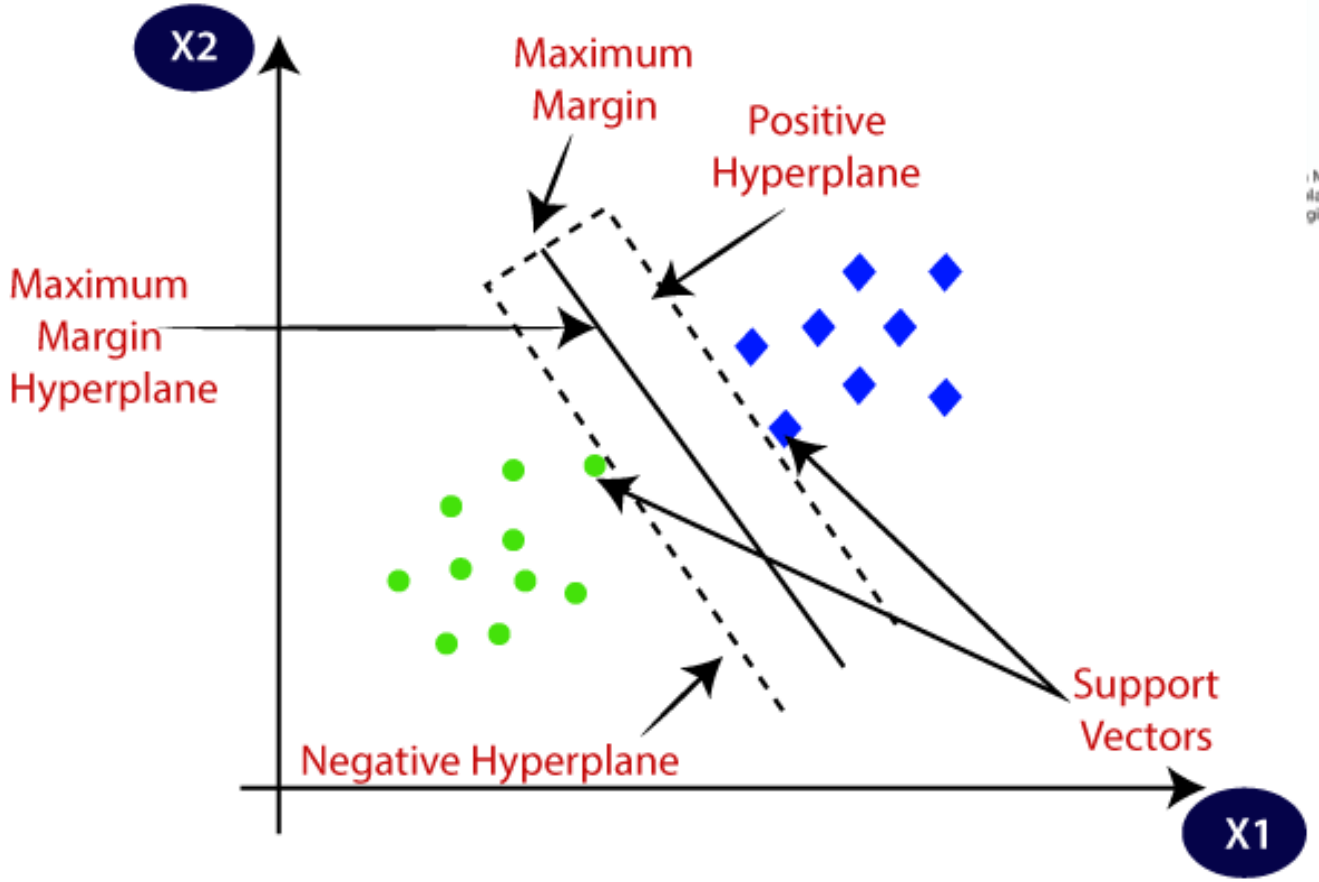
- Regression models often use techniques like least squares to minimize the difference between predicted and actual values.
- Classification models use algorithms like logistic regression, decision trees, or support vector machines to assign data points to classes.
- The choice between regression and classification depends on the nature of your prediction problem.

Support Vector Machine (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is **used for Classification as well as Regression problems**. However, primarily, it is **used for Classification problems** in Machine Learning.

The goal of the SVM algorithm is to create the **best line or decision boundary** that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a **hyperplane**.

SVM chooses the **extreme points/vectors** that help in creating the hyperplane. **These extreme cases are called support vectors**, and hence algorithm is termed a Support Vector Machine. Consider the diagram in which there are two different categories that are classified using a decision boundary or hyperplane.



Hyperplane: There can be **multiple lines/decision** boundaries to **segregate the classes** in n-dimensional space, but we need to find out the **best decision boundary** that helps to classify the data points. This **best boundary is known as the hyperplane of SVM**.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are **2 features**, then the hyperplane will be a **straight line**. And if there are **3 features**, then the hyperplane will be a **2-dimension plane**. We always create a hyperplane that has a **maximum margin**, which means the **maximum distance between the data points**.

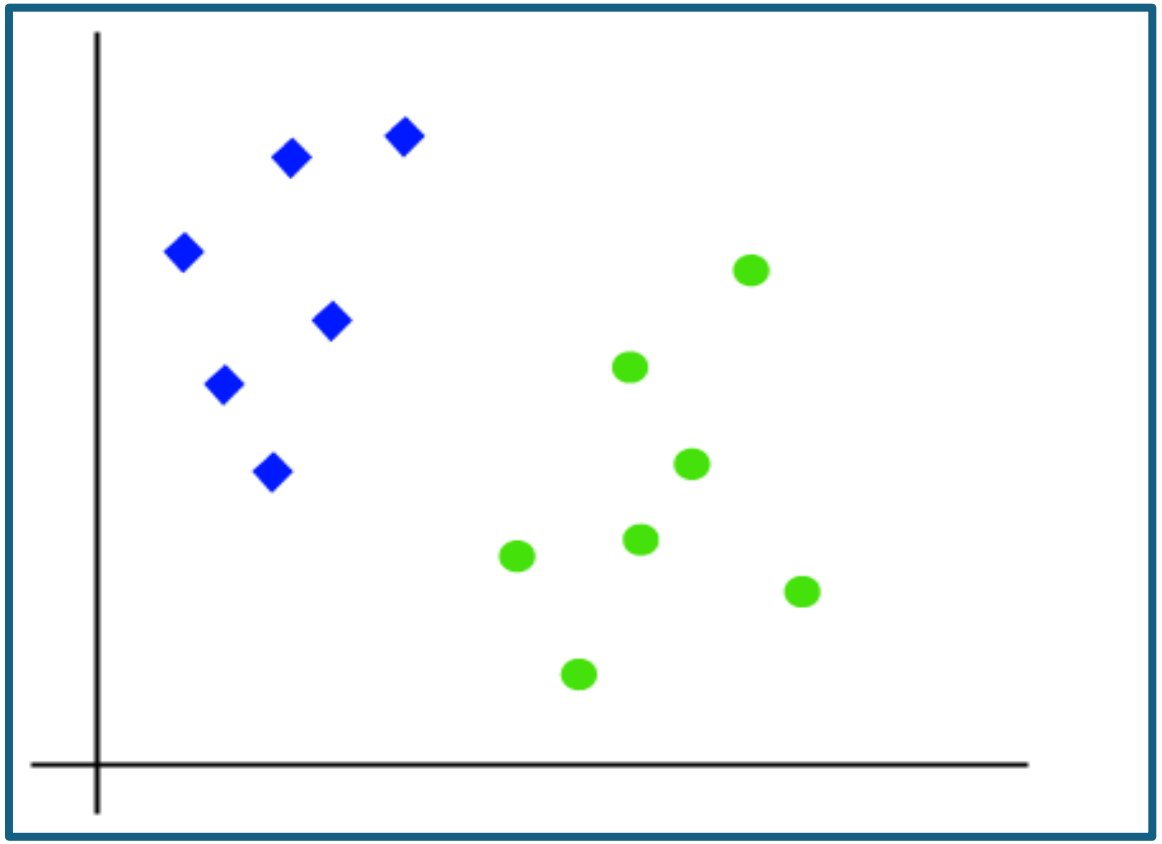
***Support Vectors*:** The data points or vectors that are the **closest to the hyperplane** and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

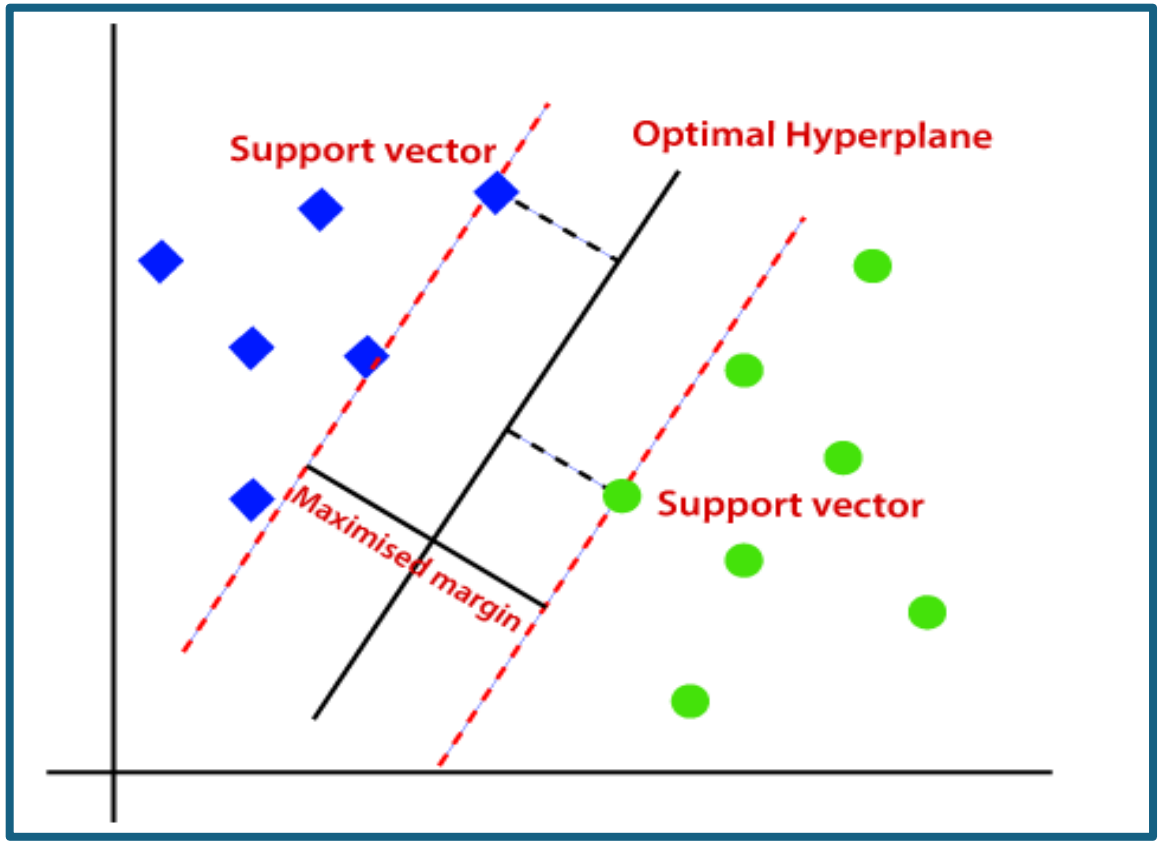
Role of Kernel in SVM

SVM algorithms use a **set of mathematical functions** that are defined as the **kernel**. The function of kernel is to **take data as input and transform it into the required form**. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example ***linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid*.**

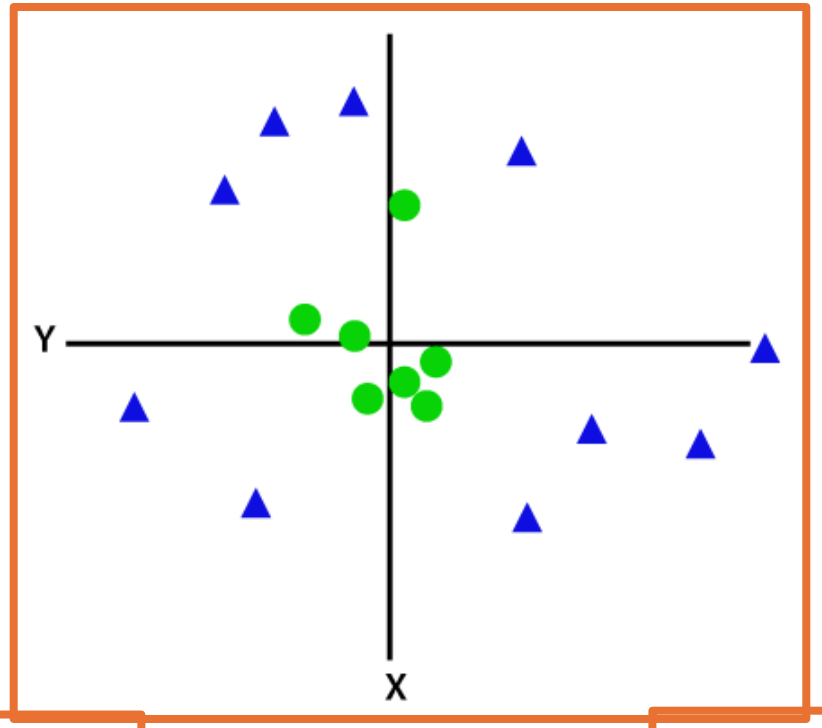
The kernel functions **return the inner product between two points in a suitable feature space**. Thus by defining a notion of similarity, with little computational cost even in very high-dimensional spaces.

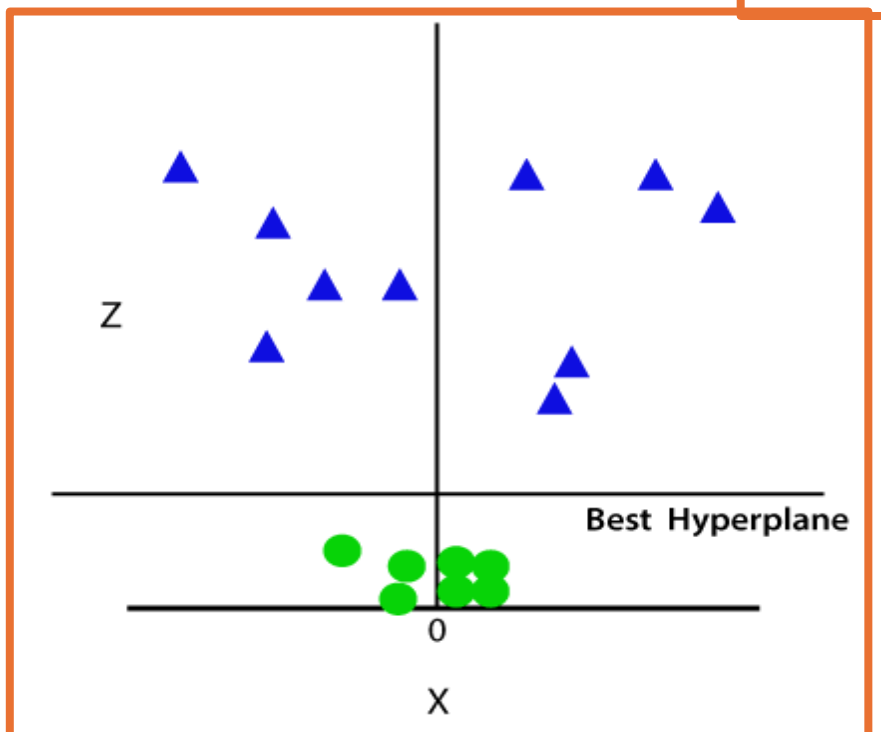
Linear SVM

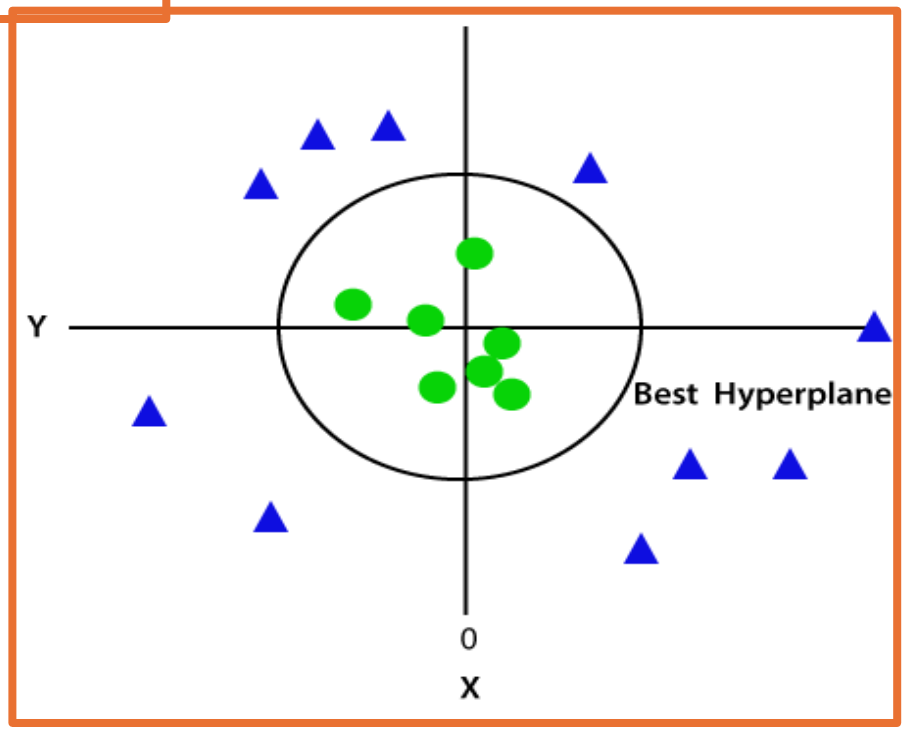



Support vector
Optimal Hyperplane
Maximised margin
Support vector

Non-Linear SVM



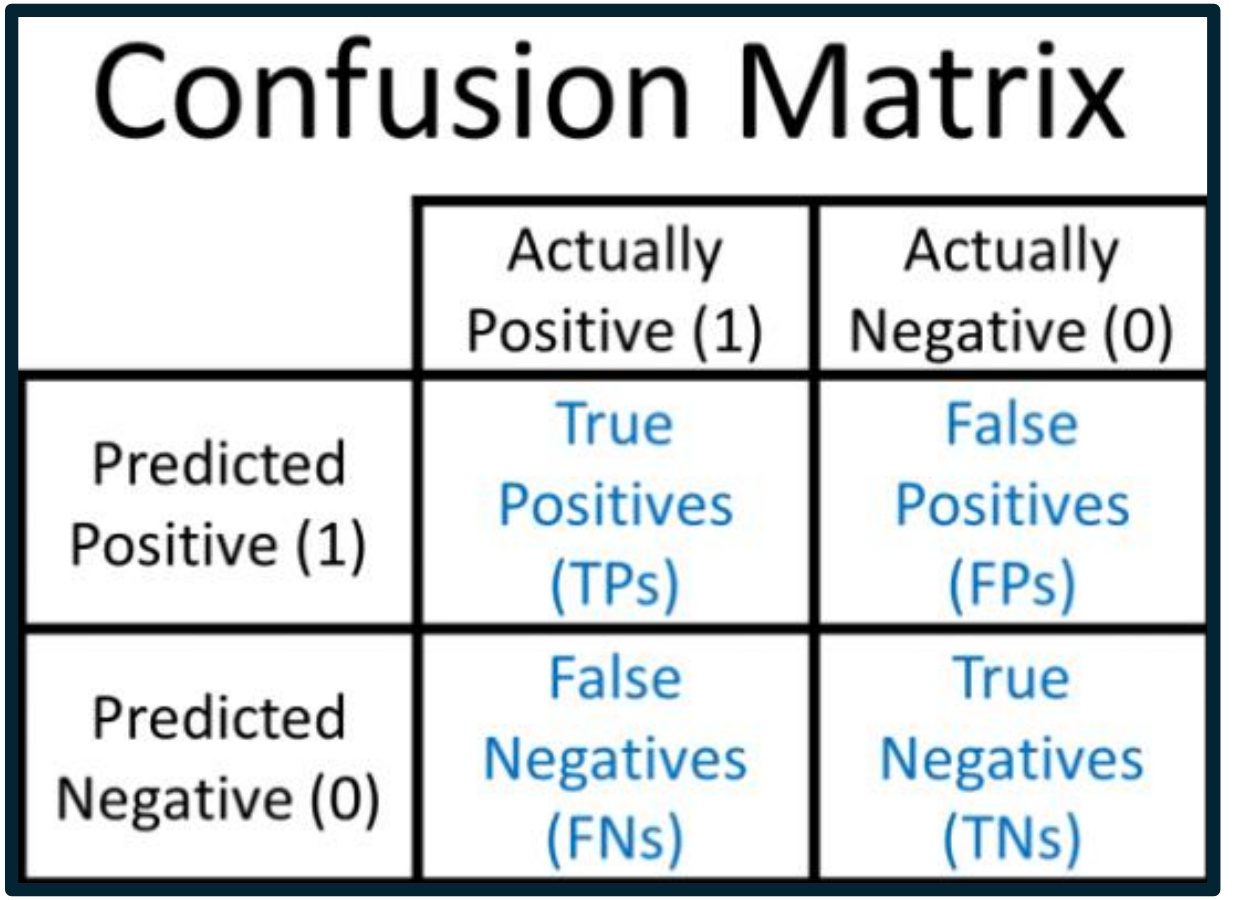




Confusion Matrix

The confusion matrix is a matrix used to determine the **performance of the classification models** for a given set of test data. It can only be determined if the **true values for test data are known**. Since it **shows the errors** in the model performance in the form of a matrix, hence also known as an **error matrix**.

The matrix is divided into **two dimensions**: **predicted values** and **actual values** along with a total number of predictions. Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.



True Positive (TP): These are the cases where the model correctly predicted a positive class for a data point that actually belongs to the positive class. (Correct positive classifications)

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

- ***False Positive (FP):*** These are the cases where the model incorrectly predicted a positive class for a data point that actually belongs to the negative class. (Type I error)
- ***False Negative (FN):*** These are the cases where the model incorrectly predicted a negative class for a data point that actually belongs to the positive class. (Type II error)
- ***True Negative (TN):*** These are the cases where the model correctly predicted a negative class for a data point that actually belongs to the negative class. (Correct negative classifications)

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

		Actual (True) Values	
		Cancer	No Cancer
Predicted Values	Cancer	45	18
	No Cancer	12	25

Classification Accuracy: It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given by:

Accuracy **= (TP + TN)/ (TP + TN + FP + FN)**

$$\frac{\text{True Positives + True Negatives}}{\text{True Positives + True Negatives + False Positives + False Negatives}} = \frac{\text{N. of Correct Predictions}}{\text{N. of all Predictions}} = \frac{\text{N. of Correct Predictions}}{\text{Size of Dataset}}$$

Precision: It can be defined as the number of **correct outputs** provided by the model or out of **all positive classes** that have been **predicted correctly by the model**, how many of them were true. It can be calculated using the below formula:

Precision = TP / (TP+FP)

$$\frac{\text{True Positives}}{\text{True Positives + False Positives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Predictions you Made}} = \frac{\text{N. of Correctly Predicted People with Cancer}}{\text{N. of People you Predicted to have Cancer}}$$

$$\frac{45}{45+18} = 0.714$$

Recall: *Recall* is a measure of how many of the positive cases the classifier correctly predicted, overall **actual positive** cases in the data. The recall must be **as high as possible.**

Recall = TP / (TP+FN)

$$\frac{\text{True Positives}}{\text{True Positives + False Negatives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Instances in the Dataset}} = \frac{\text{N. of Correctly Predicted People with Cancer}}{\text{N. of People with Cancer in the Dataset}}$$

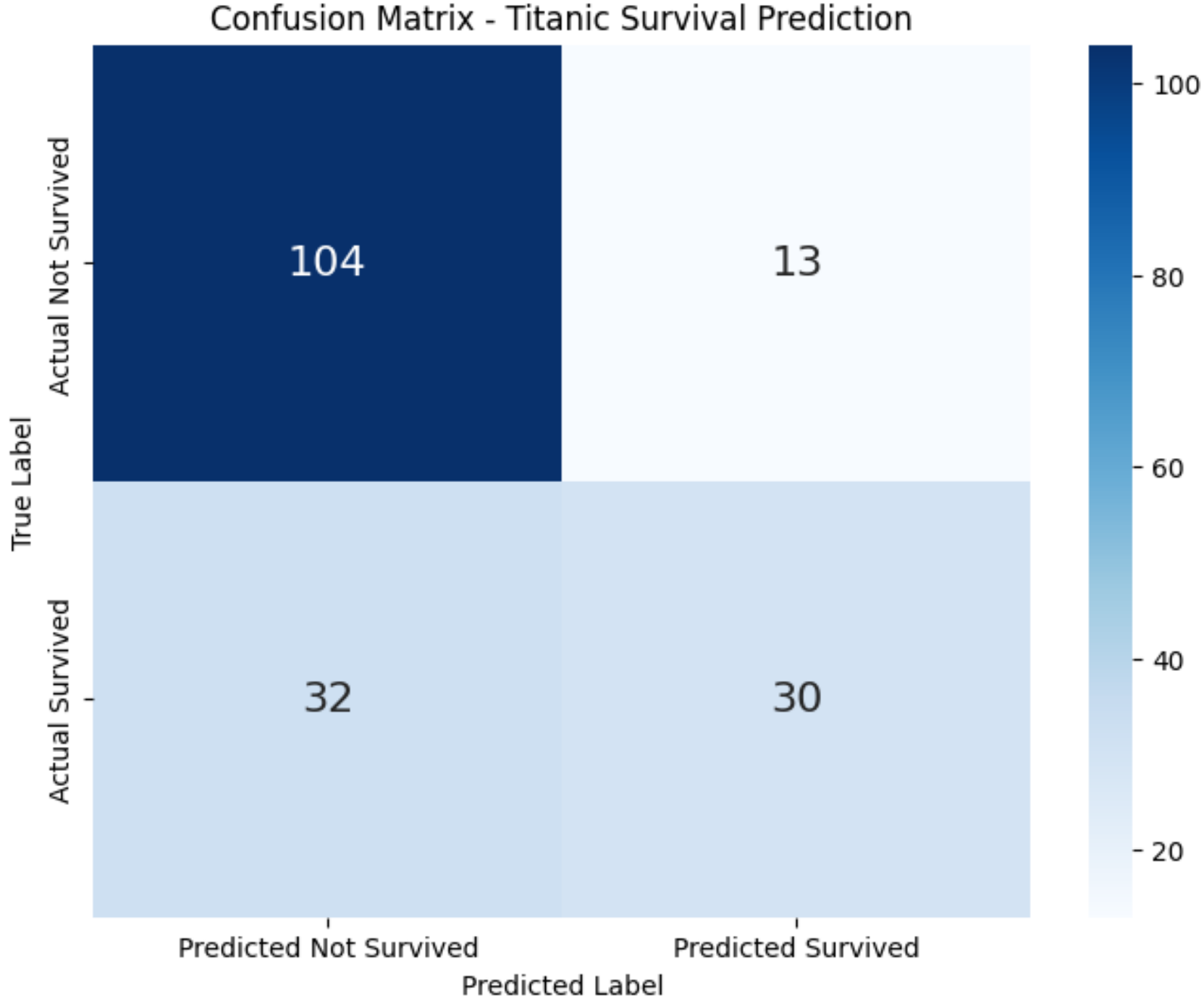
$$\frac{45}{45+12} = 0.79$$

- **F1-Score**

F1-Score is a measure combining both precision and recall. It is generally described as the ***harmonic mean*** of the two. The harmonic mean is just another way to calculate an "average" of values, generally described as more suitable for ratios (such as precision and recall) than the traditional arithmetic mean.

$$\textbf{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\frac{2 * 0.71 * 0.79}{0.71 + 0.79} = 0.747$$


Confusion Matrix - Titanic Survival Prediction
Actual Not Survived
Actual Survived
True Label
104
13
32
30
Predicted Not Survived
Predicted Survived
Predicted Label
100
80
60
40
20

Naïve Bayes

The naïve Bayes algorithm is a **supervised learning algorithm**, which is based on the **Bayes theorem** and used for solving classification problems. Naïve Bayes Classifier is one of the **simple and most effective Classification** algorithms which helps in building fast machine learning models that can make **quick predictions**. **It is a probabilistic classifier, which means it predicts based on the probability of an object**.

Naive Bayes algorithms are mostly used in **sentiment analysis**, **spam filtering, recommendation systems, etc.** They are fast and easy to implement but their **biggest disadvantage** is the **requirement for predictors to be independent**. In most of real-life cases, the predictors are dependent, this hinders the performance of the classifier.

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is **independent** of the occurrence of other features.

Bayes: It is called Bayes because it depends on the principle of **Bayes's theorem**.

Bayes' Theorem

Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Here,

P(A|B): Conditional probability of event A occurring, given the event B.

P(A): Probability of event A occurring.

P(B): Probability of event B occurring.

P(B|A): Conditional probability of event B occurring, given the event A.

Applications of Naïve Bayes Classifier

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment**

How Does Naive Bayes Work

1.Convert the given dataset into frequency tables.

2.Generate a Likelihood table by finding the probabilities of given features.

3.Now, use the Bayes theorem to calculate the posterior probability.

***Example*:-** Suppose, we have a training data set of 1200 fruits. The features in the data set are these: is the fruit yellow or not, is the fruit long or not, and is the fruit sweet or not? There are three different classes: mango, banana, and others.

Name	Yellow	Sweet	Long	Total
Banana	400	300	350	400
Mango	350	450	0	650
Others	50	100	50	150
Total	800	850	400	1200

- Out of 1200 fruits 400 are bananas, 650 are mangoes, and 150 are others.
- 350 of the total 650 mangoes are yellow and the rest are not and so on.
- 800 fruits are yellow, 850 are sweet and 400 are long from a total of 1200 fruits.

Question:- *Let's say you are given a fruit that is* ***yellow, sweet, and long and you must check the class to which it belongs.***

"Draw the likelihood table for the features against the classes."

Name	*Yellow*	*Sweet*	*Long*	*Total*
Banana	400/800=**P(Banana|Yellow)**	300/850=**P(B|S)**	350/400=**P(B|L)**	400/1200=P(Banana)
Mango	350/800=**P(Mango|Yellow)**	450/850=**P(M|S)**	0/400=**P(M|L)**	650/1200=P(Mango)
Others	50/800=**P(Other|Yellow)**	100/850=**P(O|S)**	50/400=**P(O|L)**	150/1200=P(Other)
Total	800/1200=**P(Yellow)**	850/1200=**P(Sweet)**	400/1200=**P(Long)**	1200

P(Banana\| Yellow, Sweet, Long) = P(Y\|B) * P(S\|B) * P(L\|B)

$$P(Y|B) = \frac{P(B|Y) * P(Y)}{P(B)} = \frac{\frac{400}{800} \times \frac{800}{1200}}{\frac{400}{1200}}$$

$$P(S|B) = \frac{P(B|S) * P(S)}{P(B)} = \frac{\frac{300}{\cancel{850}} \times \frac{\cancel{850}}{1200}}{400/1200}$$

$$P(L|B) = \frac{P(B|L) * P(L)}{P(B)} = \frac{\frac{350}{\cancel{400}} * \frac{\cancel{400}}{1200}}{400/1200}$$

P(Banana| Yellow, Sweet, Long) $= 1 * \frac{3}{4} * \frac{7}{8}$

Types of Naive Bayes Classifier:

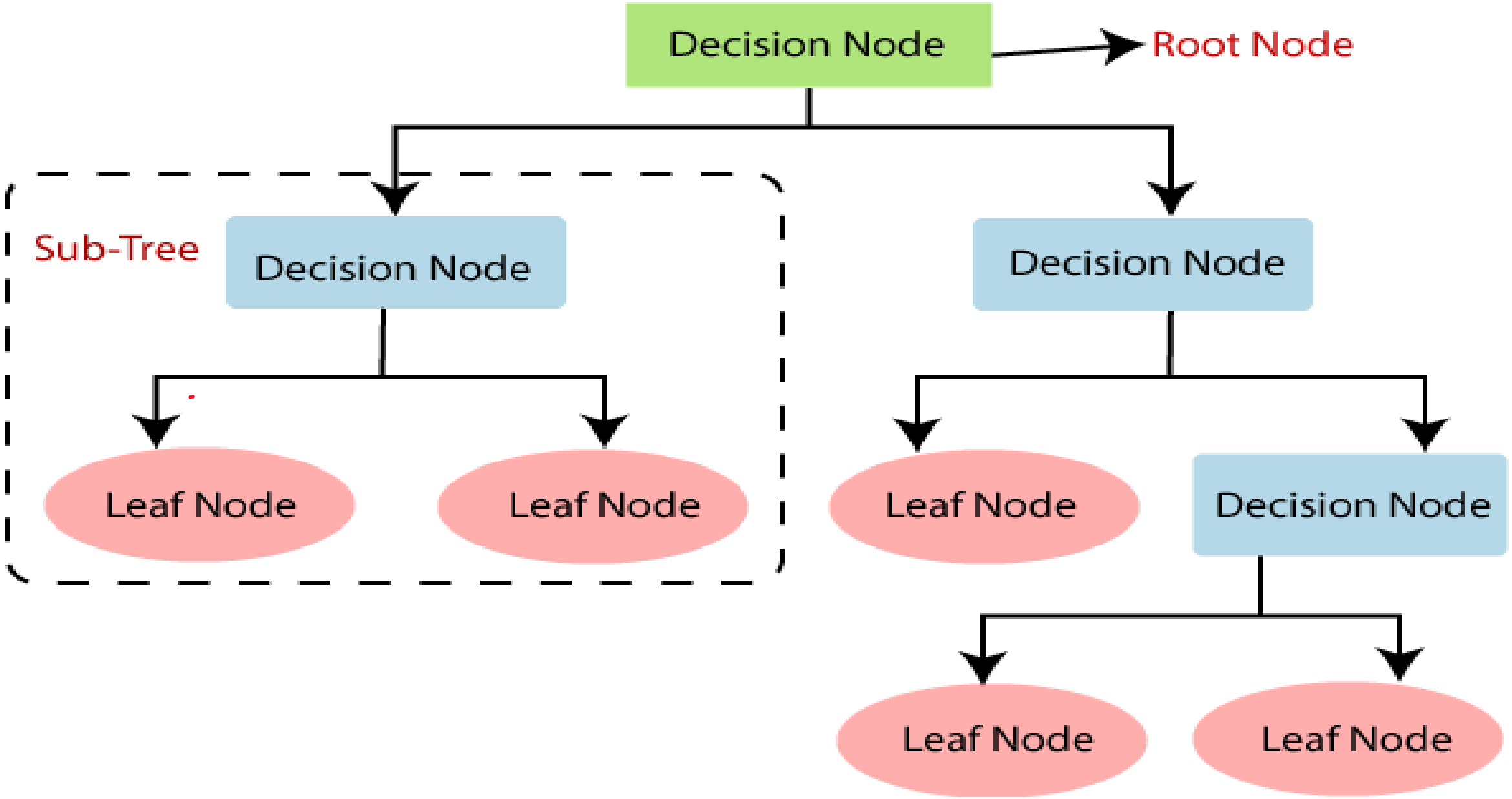
There are three types of Naive Bayes Models, which are given below:

- ***Gaussian distribution***: The Gaussian model assumes that features **follow a normal** This means if predictors take continuous values instead of discrete ones, then the model assumes that these values are sampled from the Gaussian distribution.
- ***Multinomial***: The Multinomial Naïve Bayes classifier is used when the data is **multinomial distributed**. It is primarily used for **document classification problems**, it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.
- ***Bernoulli***: The Bernoulli classifier works similarly to the Multinomial classifier, but the predictor variables are the **independent Booleans variables**. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

Decision Tree

A Decision Tree is a **Supervised learning technique** that can be used for both ***classification*** and ***Regression*** problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules,** and **each leaf node represents the outcome.** There are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***

It is called a decision tree because, ***like a tree***, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.** A decision tree simply asks a question and based on the answer (Yes/No), it further split the tree into subtrees.


Decision Node
Root Node
Sub-Tree
Decision Node
Decision Node
Leaf Node
Leaf Node
Leaf Node
Decision Node
Leaf Node
Leaf Node

Decision Tree Terminologies

***Root Node*:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

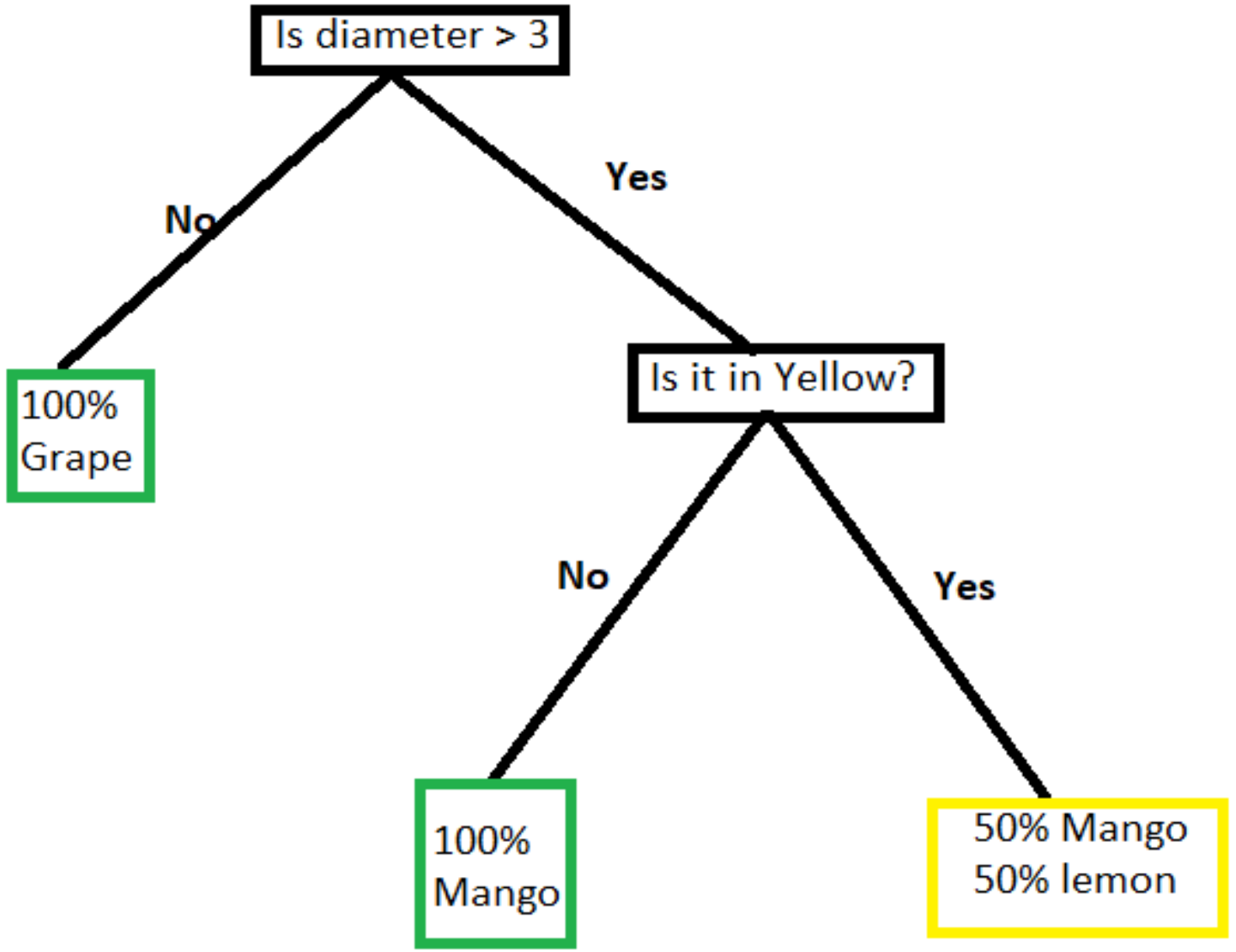
***Leaf Node*:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

***Splitting*:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

***Branch/Sub Tree*:** A tree formed by splitting the tree.

***Parent/Child node*:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

Colour	Diameter	Label
Green	3	Mango
Yellow	3	Mango
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon



Attribute Selection Measures

While implementing a Decision tree, the main issue arises regarding how to select the best attribute for the root node and sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.** By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

I. **Information Gain**
II. **Gini Index**

I. Information Gain

Information gain is the measurement of entropy changes after a dataset's segmentation based on an attribute. It calculates how much information a feature provides us about a class. According to the value of information gain, we split the node and build the decision tree. A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

$$\text{Entropy}(s) = -P(\text{yes})\log_2 P(\text{yes}) - P(\text{no})\log_2 P(\text{no})$$

Here,

- **S= Total number of samples**
- **P(yes)= probability of yes**
- **P(no)= probability of no**

II. Gini Index

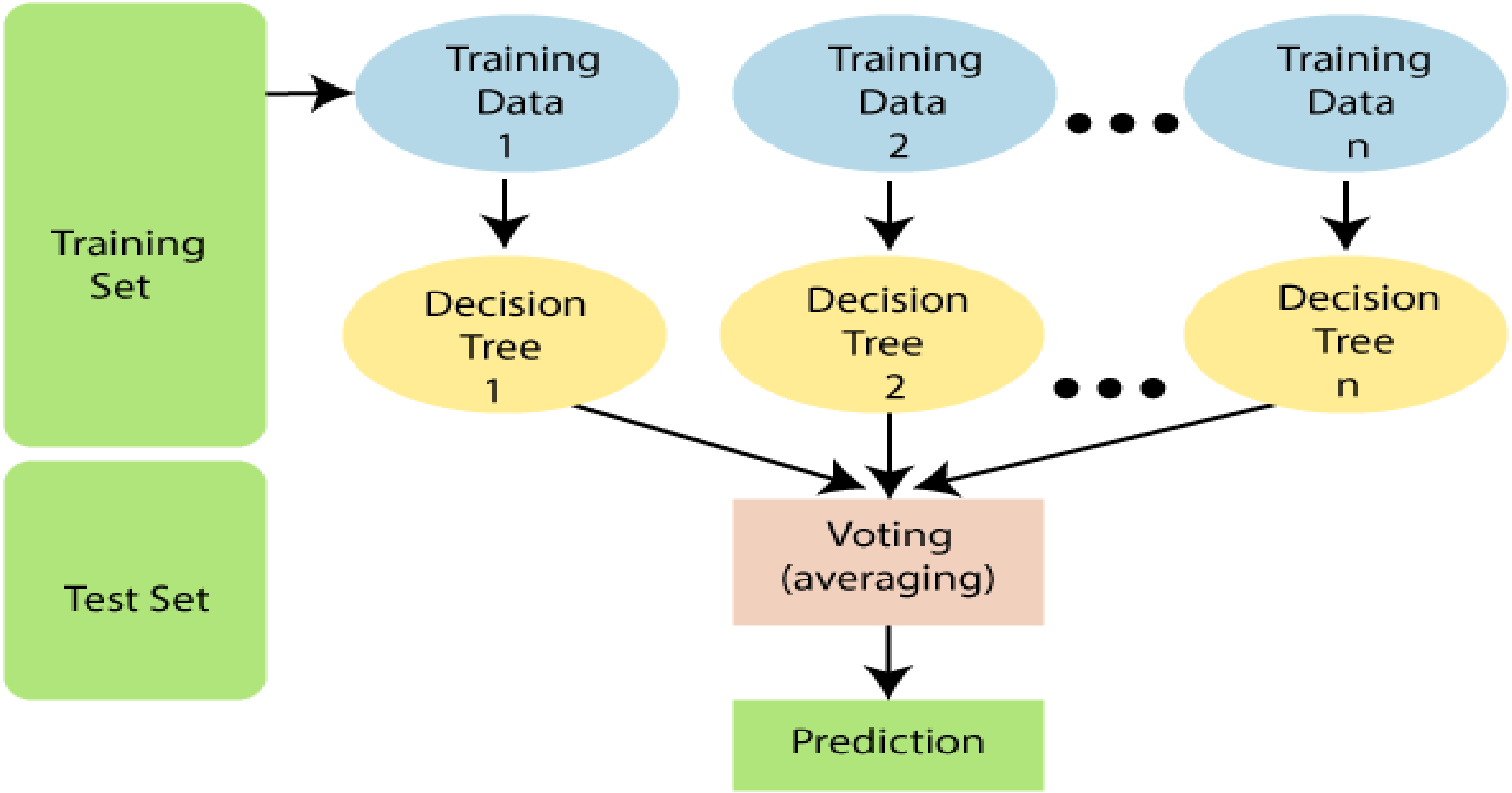
The Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm. An attribute with a low Gini index should be preferred as compared to the high Gini index. It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
Gini index can be calculated using the below formula:

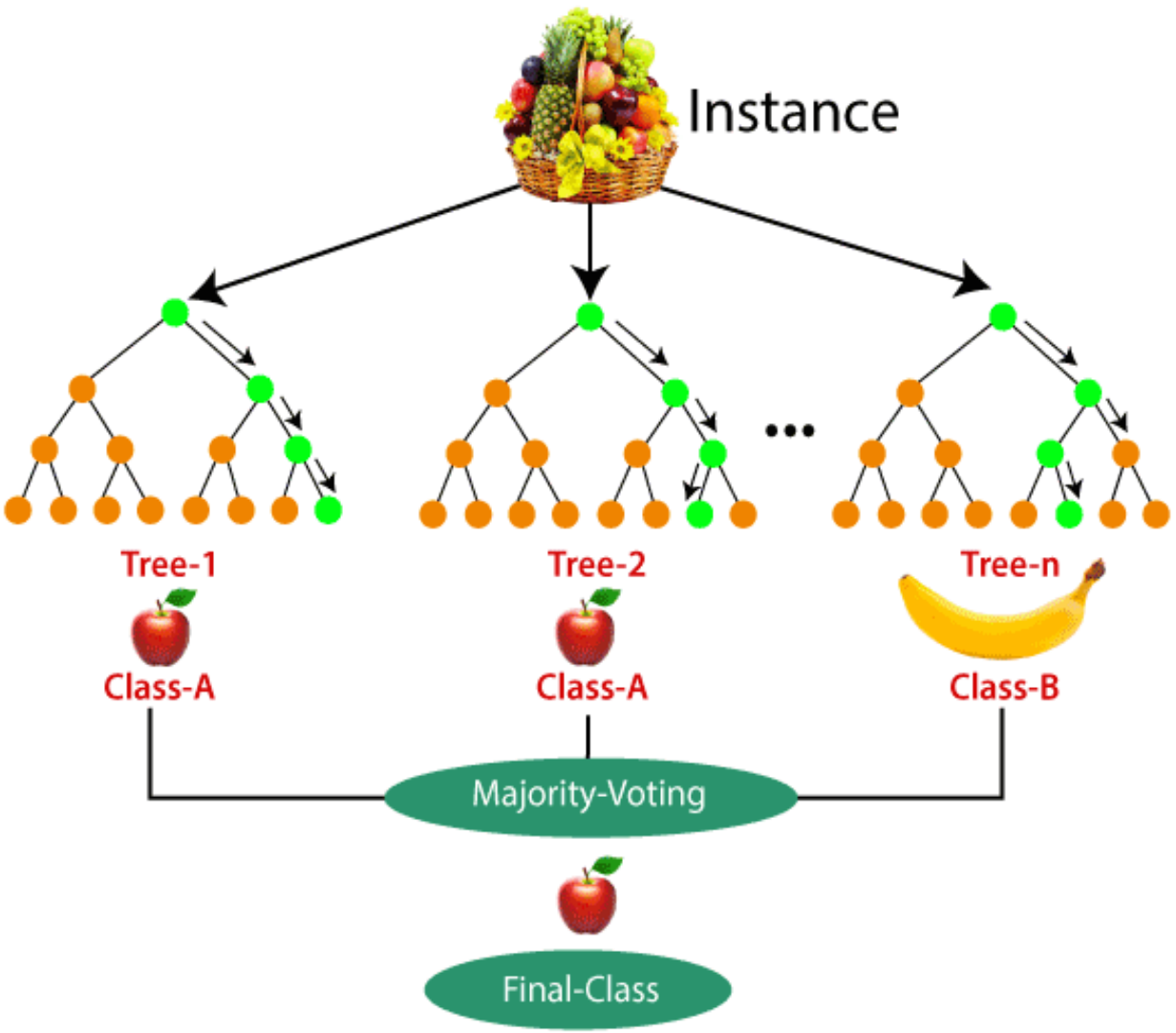
$$\text{Gini Index} = 1 - \sum_j P_j^2$$

Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and improve the performance of the model*. ***Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.


Training Set
Test Set
Training Data 1
Training Data 2
Training Data n
Decision Tree 1
Decision Tree 2
Decision Tree n
Voting (averaging)
Prediction



Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision.

Clustering

Clustering or cluster analysis is a machine learning technique, which groups the **unlabelled dataset**. It can be defined as "***A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group***.“ It does it by finding some similar patterns in the unlabelled dataset such as **shape, size, color, behavior, etc.**, and divides them as per the presence and absence of those similar patterns.

It is an unsupervised learning method; hence **no supervision** is provided to the algorithm, and it deals with the unlabeled dataset.

Clustering is **somewhere similar to the classification algorithm**, but the difference is the type of dataset that we are using. In classification, we work with the labeled data set, whereas in clustering, we work with the **unlabelled dataset.**

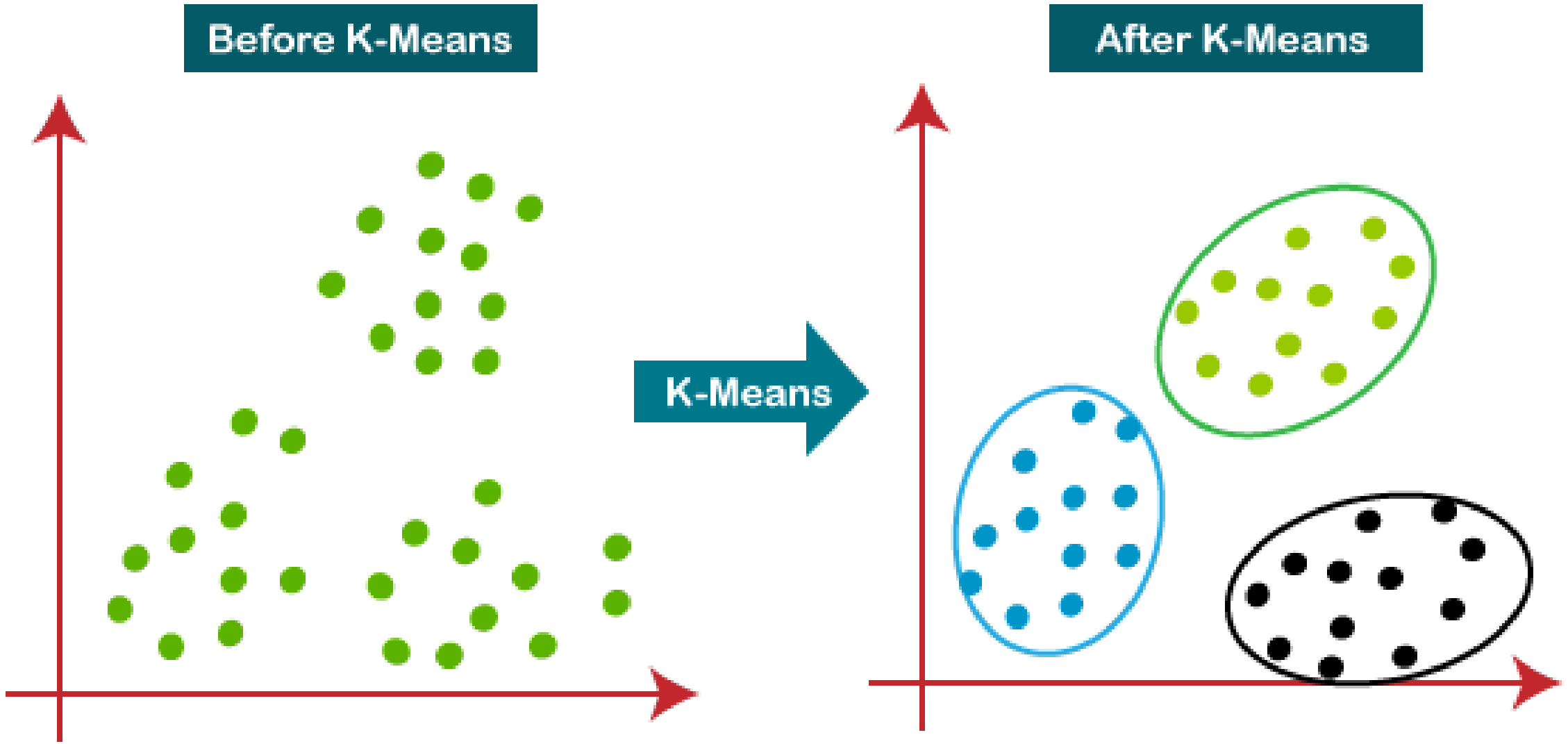
K-Means Clustering

K-Means Clustering is an **unsupervised learning algorithm** that is used to solve **clustering problems** in machine learning or data science. This algorithm groups the unlabeled dataset into different clusters. Here ***K defines the number of pre-defined clusters that need to be created in the process***, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

"It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties."

The k-means clustering algorithm mainly performs two tasks:

1. Determines the ***best value for K center points or centroids*** by an iterative process.
2. Assign each data point to **its closest k-center**. Those data points which are near the particular k-center, create a cluster.


Before K-Means
K-Means
After K-Means

Steps evolve K-Means Algorithm:

The working of the K-Means algorithm is explained in the below steps:

***Step 1*:** Select the number K to decide the number of clusters.

***Step 2*:** Select random K points or centroids. (It can be other than the input dataset).

***Step 3*:** Assign each data point to its closest centroid, which will form the predefined K clusters.

***Step 4*:** Calculate the variance and place a new centroid of each cluster.

***Step-5*:** Repeat the third steps, which means reassigning each datapoint to the new closest centroid of each cluster.

***Step-6*:** If any reassignment occurs, then go to step 4 else go to FINISH.

Step-7: The model is ready.

Find the value of "K number of clusters" in K-means Clustering

The performance of the K-means clustering algorithm depends upon the highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K.

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster.

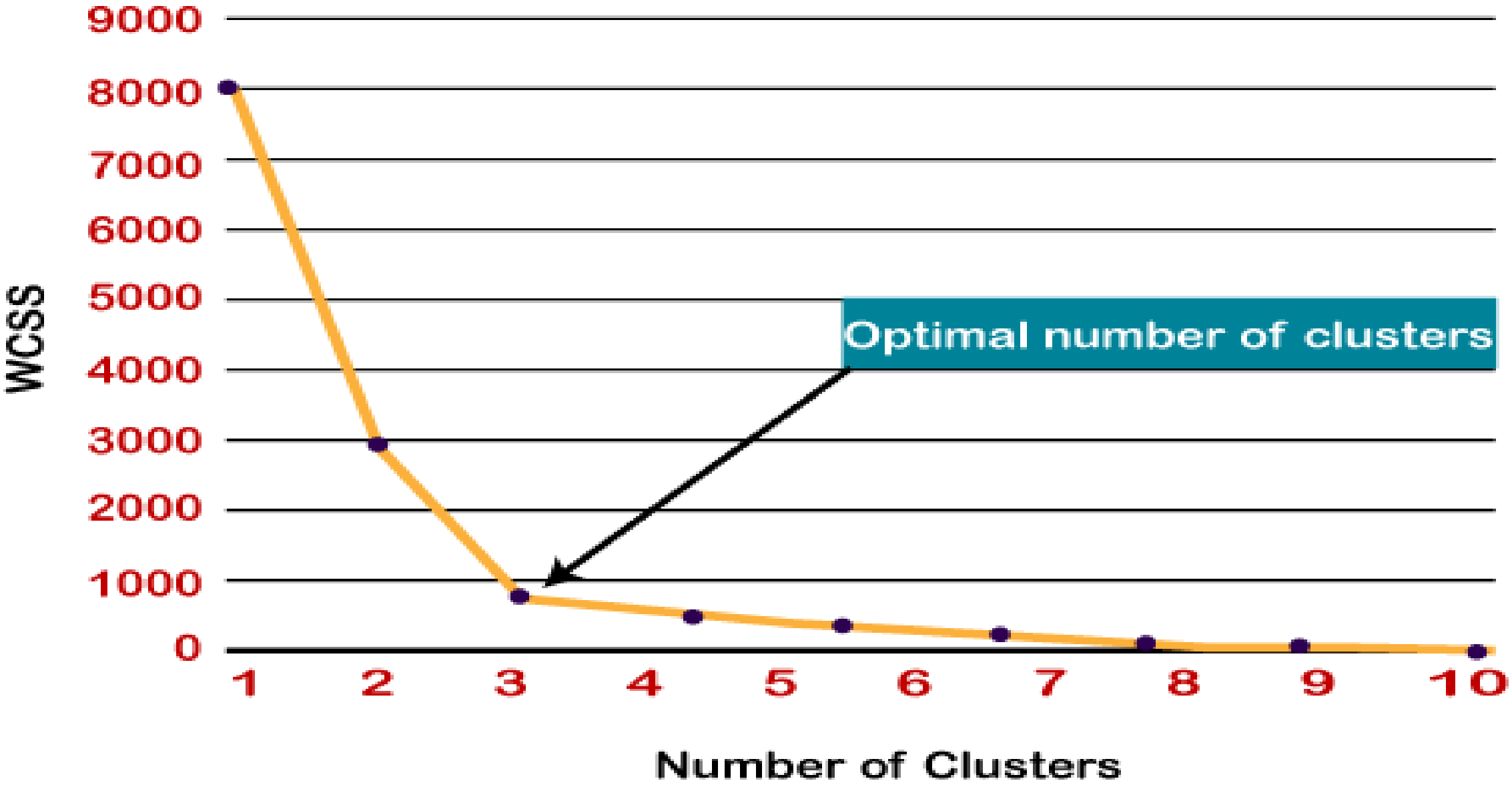
$$WCSS= \sum_{Pi\ in\ Cluster1} distance(P_i\ C_1)^2 + \sum_{Pi\ in\ Cluster2} distance(P_i\ C_2)^2 + \sum_{Pi\ in\ CLuster3} distance(P_i\ C_3)^2$$

$$WCSS = \sum_{Pi\ in\ Cluster1} distance(P_i\ C_1)^2 + \sum_{Pi\ in\ Cluster2} distance(P_i\ C_2)^2 + \sum_{Pi\ in\ CLuster3} distance(P_i\ C_3)^2$$

$\sum_{Pi\ in\ Cluster1}$ distance(P_i C_1)^2: It is the *sum of the square of the distances between each data point and its centroid within a cluster1* and the same for the other two terms. To measure the distance between data points and the centroid, we can use any method such as *Euclidean distance* or *Manhattan distance*.

Steps to find the optimal value of clusters

a) It executes the K-means clustering on a given dataset for different K values (***ranges from 1-10***).
b) For each value of K calculates the WCSS value.
c) Plots a curve between calculated WCSS values and the number of clusters K.
d) The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K


9000
8000
7000
6000
5000
4000
3000
2000
1000
0
WCSS
Optimal number of clusters
1
2
3
4
5
6
7
8
9
10
Number of Clusters

Hierarchical clustering

Hierarchical clustering is another unsupervised machine learning algorithm used to group the unlabeled datasets into a cluster and also known as **hierarchical cluster analysis** or HCA. In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**. Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

As we have seen in the K-means clustering that there are some challenges with this algorithm, which are a predetermined number of clusters, and it always tries to create the clusters of the same size. To solve these two challenges, we can opt for the hierarchical clustering algorithm because, in this algorithm, we don't need to have knowledge about the predefined number of clusters.

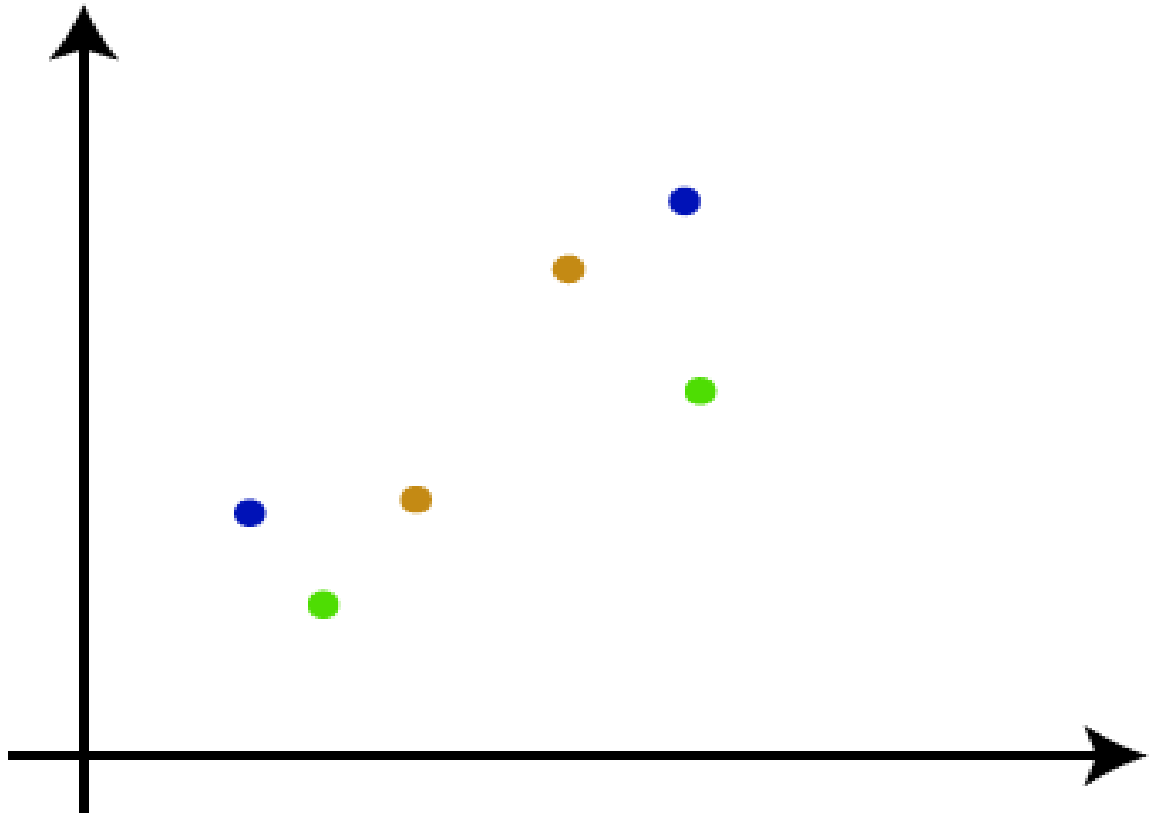
The hierarchical clustering technique has two approaches:

1. ***Agglomerative*:** Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
2. ***Divisive*:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**

How the Agglomerative Hierarchical clustering Work

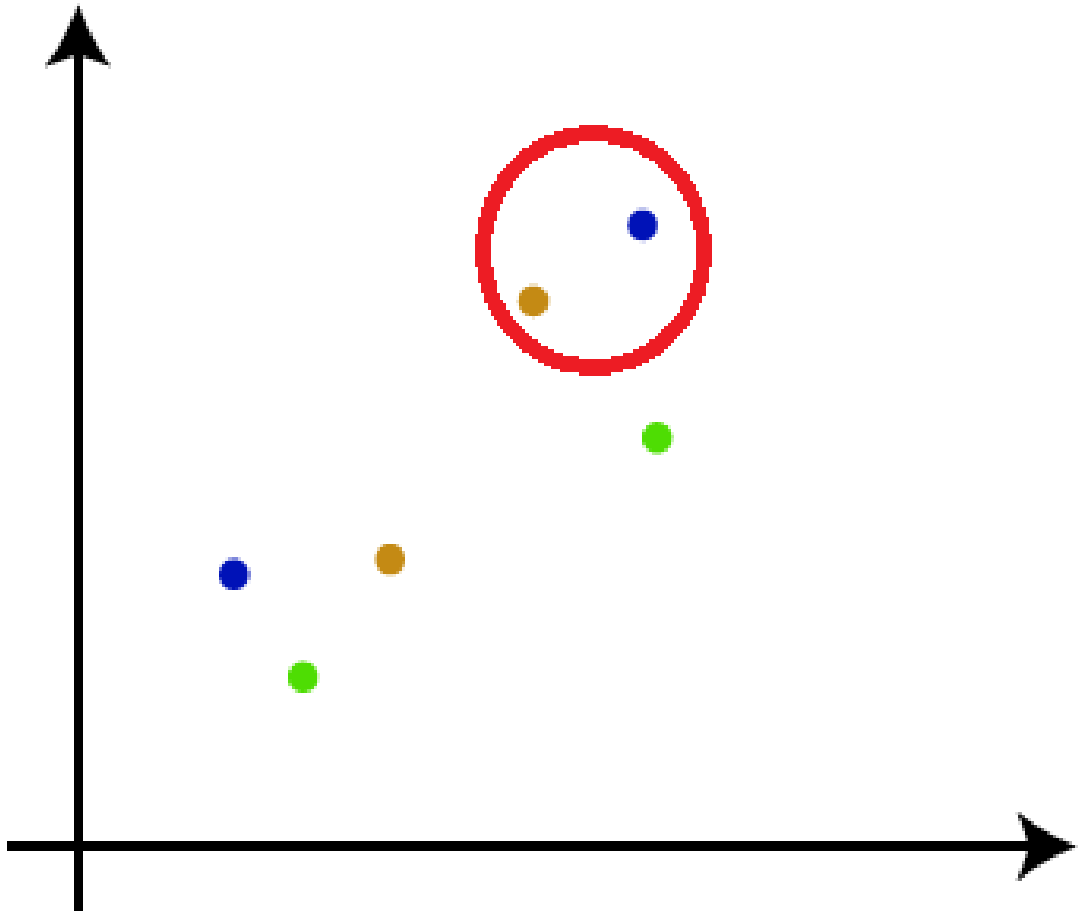
Step-1:

Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



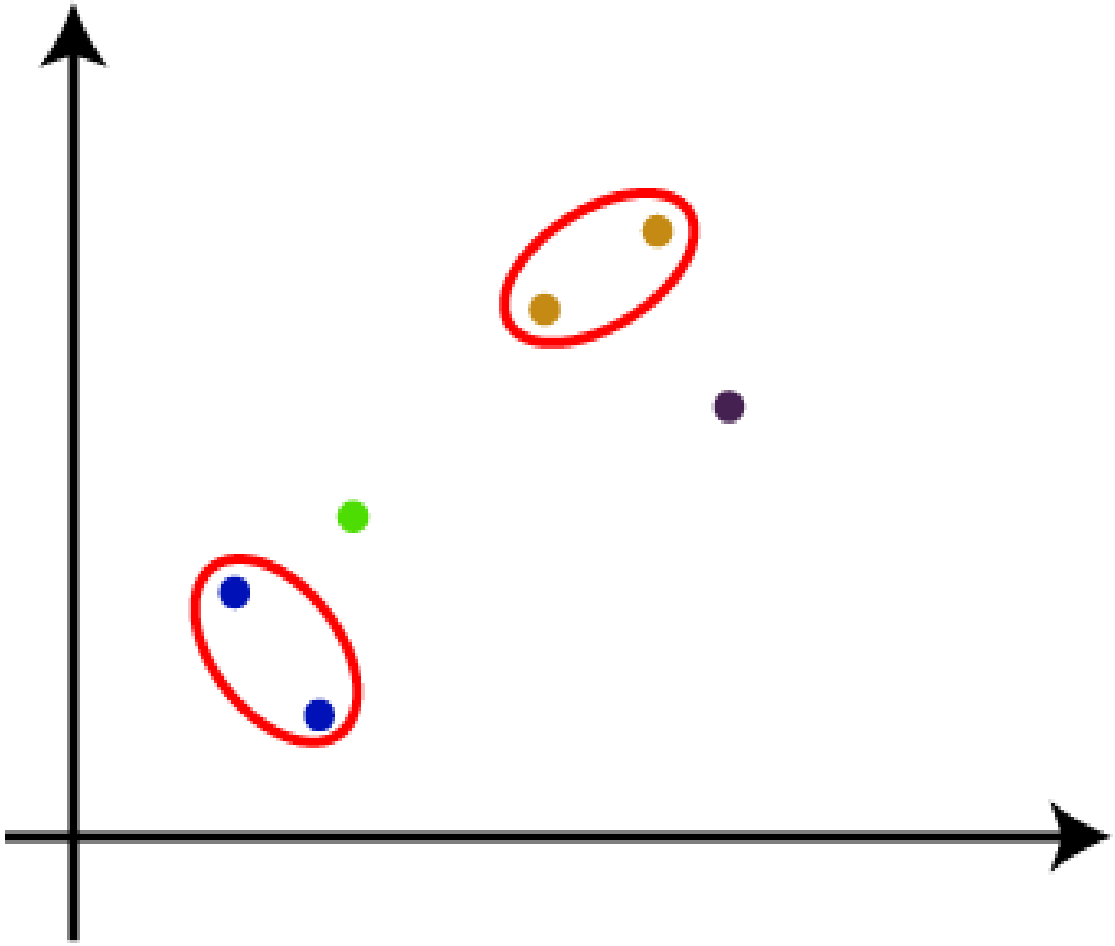
Step-2:

Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters



Step-3:

Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



Step-4:

Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images: