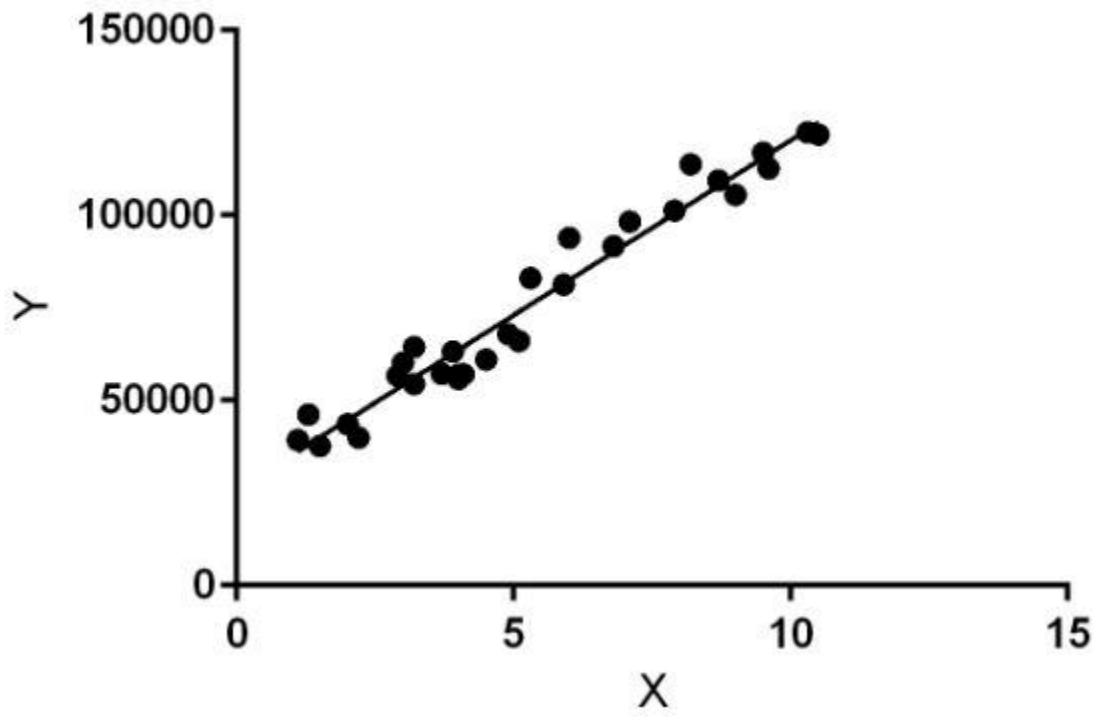
UNIT 2

SUPERVISED LEARNING

Linear Regression

- Linear Regression is a machine learning algorithm based on supervised learning.
- It performs a regression task.
- Regression models a target prediction value based on independent variables.
- It is mostly used for finding out the relationship between variables and forecasting.
- Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.
- There are many names for a regression's dependent variable.
- It may be called an outcome variable, criterion variable, endogenous variable, or regressand.
- The independent variables can be called exogenous variables, predictor variables, or regressors.
- Linear regression is used in many different fields, including finance, economics, and psychology, to understand and predict the behaviour of a particular variable.
- For example, in finance, linear regression might be used to understand the relationship between a company's stock price and its earnings, or to predict the future value of a currency based on its past performance.
- One of the most important supervised learning tanks is regression.
- In regression set of records are present with X and Y values and this values are used to learn a function, so that if you want to predict Y from an unknown X this learn function can be used.
- In regression we have to find value of Y, So, a function is required which predicts Y given XY is continuous in case of regression.

- Here Y is called as criterion variable and X is called as predictor variable.
- There are many types of functions or modules which can be used for regression. Linear function is the simplest type of function.
- Here, X may be a single feature or multiple features representing the problem.



- Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)).
- Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person.
- The regression line is the best fit line for our model.

Hypothesis function for Linear Regression :

$$y = \theta_1 + \theta_2.x$$

Linear Models for Classification

Classification:

- Classification is a process of finding a function which helps in dividing the dataset into classes based on different parameters.
- In Classification, a computer program is trained on the training dataset and based on that training, it categorizes the data into different classes.

Example:

- The best example to understand the Classification problem is Email Spam Detection.
- The model is trained on the basis of millions of emails on different parameters, and whenever it receives a new email, it identifies whether the email is spam or not.
- If the email is spam, then it is moved to the Spam folder.

Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the following types:

- Logistic Regression
- K-Nearest Neighbours
- Support Vector Machines
- Kernel SVM
- Naïve Bayes
- Decision Tree Classification
- Random Forest Classification

Regression:

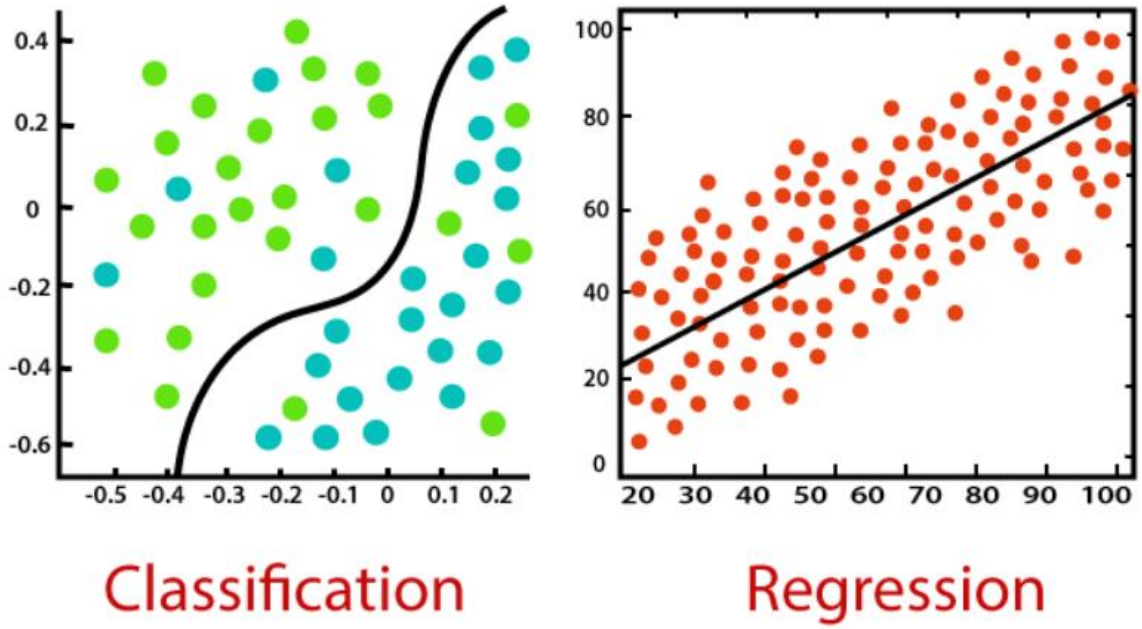
- Regression is a process of finding the correlations between dependent and independent variables.
- It helps in predicting the continuous variables such as prediction of Market Trends, prediction of House prices, etc.

- The task of the Regression algorithm is to find the mapping function to map the input variable(x) to the continuous output variable(y).

Example: Suppose we want to do weather forecasting, so for this, we will use the Regression algorithm. In weather prediction, the model is trained on the past data, and once the training is completed, it can easily predict the weather for future days.

Types of Regression Algorithm:

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression



Linear Discriminant Analysis

1. Linear Discriminant Analysis (LDA) is a supervised learning algorithm used for classification tasks in machine learning. It is a technique used to find a linear combination of features that best separates the classes in a dataset.
2. LDA works by projecting the data onto a lower-dimensional space that maximizes the separation between the classes. It does this by finding a set of linear discriminants that maximize the ratio of between-class variance to within-class variance. In other words, it finds the directions in the feature space that best separate the different classes of data.
3. LDA assumes that the data has a Gaussian distribution and that the covariance matrices of the different classes are equal. It also assumes that the data is linearly separable, meaning that a linear decision boundary can accurately classify the different classes.

LDA has several advantages, including:

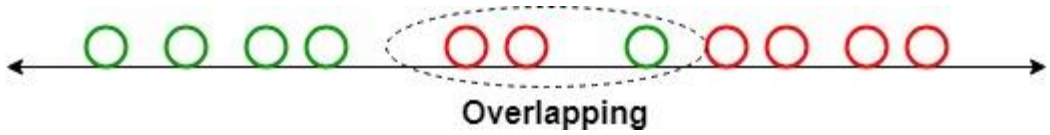
- It is a simple and computationally efficient algorithm.
- It can work well even when the number of features is much larger than the number of training samples.
- It can handle multicollinearity (correlation between features) in the data.

However, LDA also has some limitations, including:

- It assumes that the data has a Gaussian distribution, which may not always be the case.
- It assumes that the covariance matrices of the different classes are equal, which may not be true in some datasets.
- It assumes that the data is linearly separable, which may not be the case for some datasets.
- It may not perform well in high-dimensional feature spaces.

- Linear Discriminant Analysis or Normal Discriminant Analysis or Discriminant Function Analysis is a dimensionality reduction technique that is commonly used for supervised classification problems.
- It is used for modelling differences in groups i.e. separating two or more classes.
- It is used to project the features in higher dimension space into a lower dimension space.

  For example, we have two classes and we need to separate them efficiently.
- Classes can have multiple features. Using only a single feature to classify them may result in some overlapping as shown in the below figure.
- So, we will keep on increasing the number of features for proper classification.



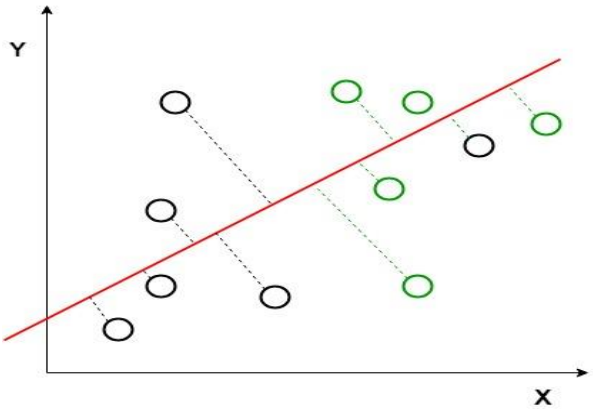
Example:

Suppose we have two sets of data points belonging to two different classes that we want to classify. As shown in the given 2D graph, when the data points are plotted on the 2D plane, there's no straight line that can separate the two classes of the data points completely. Hence, in this case, LDA (Linear Discriminant Analysis) is used which reduces the 2D graph into a 1D graph in order to maximize the separability between the two classes.

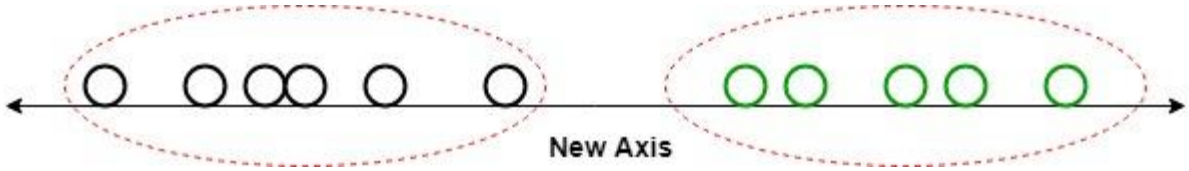
Here, Linear Discriminant Analysis uses both the axes (X and Y) to create a new axis and projects data onto a new axis in a way to maximize the separation of the two categories and hence, reducing the 2D graph into a 1D graph.

Two criteria are used by LDA to create a new axis:

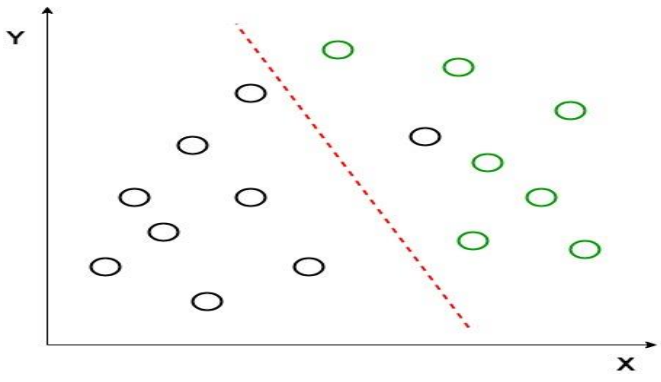
1. Maximize the distance between means of the two classes.
2. Minimize the variation within each class.



In the above graph, it can be seen that a new axis (in red) is generated and plotted in the 2D graph such that it maximizes the distance between the means of the two classes and minimizes the variation within each class. In simple terms, this newly generated axis increases the separation between the data points of the two classes. After generating this new axis using the above-mentioned criteria, all the data points of the classes are plotted on this new axis and are shown in the figure given below.



But Linear Discriminant Analysis fails when the mean of the distributions are shared, as it becomes impossible for LDA to find a new axis that makes both the classes linearly separable. In such cases, we use non-linear discriminant analysis.



Probabilistic Model in Machine Learning

- A Probabilistic model in machine learning is a mathematical representation of a real-world process that incorporates uncertain or random variables.
- The goal of probabilistic modelling is to estimate the probabilities of the possible outcomes of a system based on data or prior knowledge.
- Probabilistic models are used in a variety of machine learning tasks such as classification, regression, clustering, and dimensionality reduction.

Some popular probabilistic models include:

- Gaussian Mixture Models (GMMs)
- Hidden Markov Models (HMMs)
- Bayesian Networks
- Markov Random Fields (MRFs)

Probabilistic models allow for the expression of uncertainty, making them particularly well-suited for real-world applications where data is often noisy or incomplete. Additionally, these models can often be updated as new data becomes available, which is useful in many dynamic and evolving systems.

For better understanding, we will implement the probabilistic model on the OSIC Pulmonary Fibrosis problem on the kaggle.

Problem Statement: "In this competition, you'll predict a patient's severity of decline in lung function based on a CT scan of their lungs. You'll determine lung function based on output from a spirometer, which measures the volume of air inhaled and exhaled.

The challenge is to use machine learning techniques to make a prediction with the image, metadata, and baseline FVC as input."

Importing Libraries

```
1. import pandas as pd
2. import numpy as np
```

```
3.  import seaborn as sns
4.  import matplotlib.pyplot as plt
```

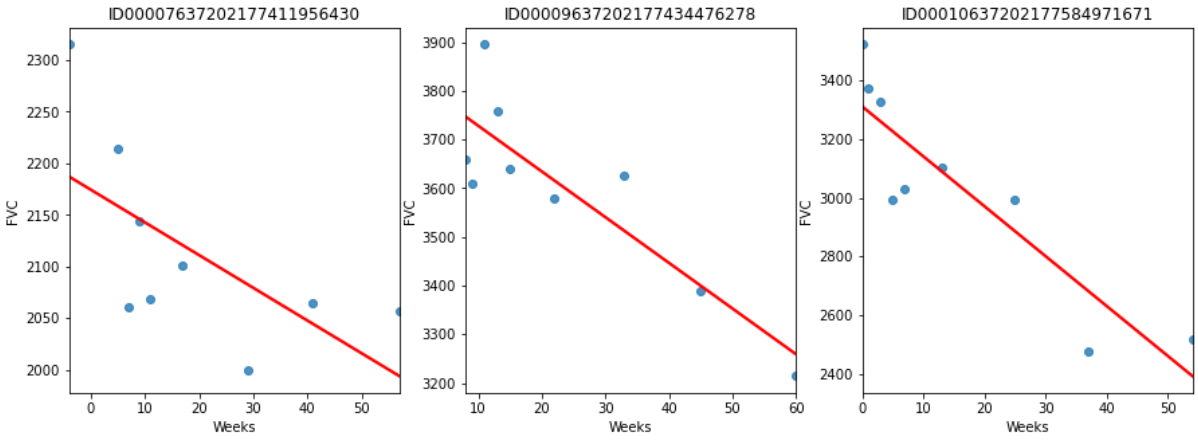
EDA

```
1.  train_Datafame = pd.read_csv('train.csv')
2.  test_Dataframe = pd.read_csv('test.csv')
```

Let's see this decline in lung function for three different patients.

```
1.  def chart_builder(patient_id, ax):
2.      d = train_Datafame[train_Datafame['Patient'] == patient_id]
3.      x = d['Weeks']
4.      y = d['FVC']
5.      ax.set_title(patient_id)
6.      ax = sns.regplot(x, y, ax=ax, ci=None, line_kws={'color':'red'})
7.
8.
9.  f, axes = plt.subplots(1, 3, figsize=(15, 5))
10. chart_builder('ID00007637202177411956430', axes[0])
11. chart_builder('ID00009637202177434476278', axes[1])
12. chart_builder('ID00010637202177584971671', axes[2])
```

Output:



$$
\begin{aligned}
FVC_{ij} &\sim \mathcal{N}(\alpha_i + j\beta_i, \sigma_i) \\
\sigma_i &\sim |\mathcal{N}(0, 200)| \\
\alpha_i &\sim \mathcal{N}(FVC_i^b + w_i^b \beta^{int}, \sigma^{int}) \\
\beta_i &\sim \mathcal{N}(\alpha^s + A_i \beta_c^s, \sigma^s) \\
\beta^{int} &\sim \mathcal{N}(0, 100) \\
\sigma^{int} &\sim |\mathcal{N}(0, 100)| \\
\beta_c^s &\sim \mathcal{N}(0, 100) \\
\alpha^s &\sim \mathcal{N}(0, 100) \\
\sigma^s &\sim |\mathcal{N}(0, 100)|
\end{aligned}
$$

Support Vector Machines (SVM)

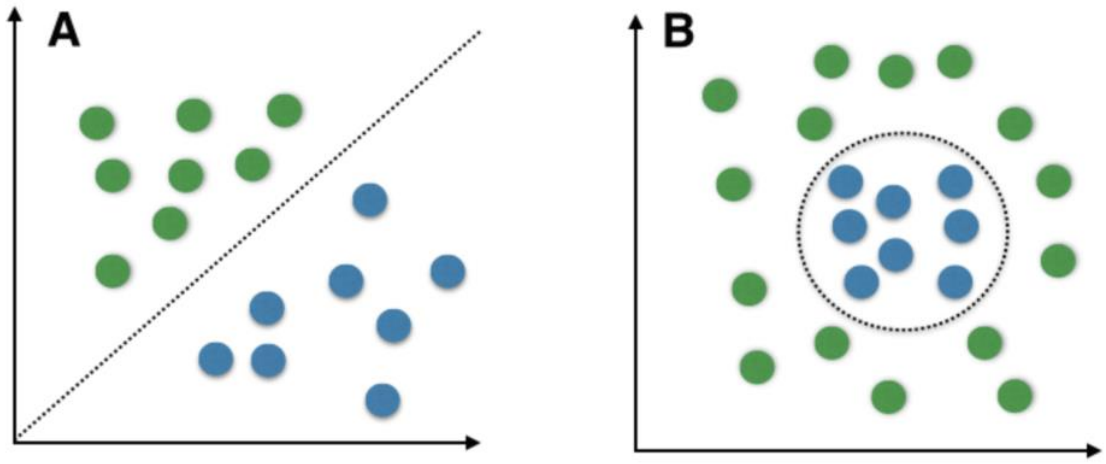
- Support Vector Machines (SVMs) are a type of supervised learning algorithm that can be used for classification or regression tasks.
- The main idea behind SVMs is to find a hyperplane that maximally separates the different classes in the training data.
- This is done by finding the hyperplane that has the largest margin, which is defined as the distance between the hyperplane and the closest data points from each class.
- Once the hyperplane is determined, new data can be classified by determining on which side of the hyperplane it falls.
- SVMs are particularly useful when the data has many features, and/or when there is a clear margin of separation in the data.
- What are Support Vector Machines? Support Vector Machine (SVM) is a relatively simple Supervised Machine Learning Algorithm used for classification and/or regression.
- It is more preferred for classification but is sometimes very useful for regression as well. Basically, SVM finds a hyper-plane that creates a boundary between the types of data.
- In 2-dimensional space, this hyper-plane is nothing but a line.
- In SVM, we plot each data item in the dataset in an N-dimensional space, where N is the number of features/attributes in the data.
- Next, find the optimal hyperplane to separate the data.

- So by this, you must have understood that inherently, SVM can only perform binary classification (i.e., choose between two classes). However, there are various techniques to use for multi-class problems.
- Support Vector Machine for Multi-CLass Problems To perform SVM on multi-class problems, we can create a binary classifier for each class of the data.

The two results of each classifier will be :

- The data point belongs to that class OR
- The data point does not belong to that class.

For example, in a class of fruits, to perform multi-class classification, we can create a binary classifier for each fruit. For say, the 'mango' class, there will be a binary classifier to predict if it IS a mango OR it is NOT a mango. The classifier with the highest score is chosen as the output of the SVM. SVM for complex (Non Linearly Separable) SVM works very well without any modifications for linearly separable data. Linearly Separable Data is any data that can be plotted in a graph and can be separated into classes using a straight line.



- We use Kernelized SVM for non-linearly separable data. Say, we have some non-linearly separable data in one dimension.
- We can transform this data into two dimensions and the data will become linearly separable in two dimensions.
- This is done by mapping each 1-D data point to a corresponding 2-D ordered pair.
- So for any non-linearly separable data in any dimension, we can just map the data to a higher dimension and then make it linearly separable.

- This is a very powerful and general transformation. A kernel is nothing but a measure of similarity between data points.
- The kernel function in a kernelized SVM tells you, that given two data points in the original feature space, what the similarity is between the points in the newly transformed feature space.

There are various kernel functions available, but two are very popular :

Radial Basis Function Kernel (RBF):

- The similarity between two points in the transformed feature space is an exponentially decaying function of the distance between the vectors and the original input space as shown below.
- RBF is the default kernel used in SVM.

Polynomial Kernel:

- The Polynomial kernel takes an additional parameter, 'degree' that controls the model's complexity and computational cost of the transformation.

The Kernel Trick:

- A very interesting fact is that SVM does not actually have to perform this actual transformation on the data points to the new high dimensional feature space.
- This is called the kernel trick.
- Internally, the kernelized SVM can compute these complex transformations just in terms of similarity calculations between pairs of points in the higher dimensional feature space where the transformed feature representation is implicit.
- This similarity function, which is mathematically a kind of complex dot product is actually the kernel of a kernelized SVM.
- This makes it practical to apply SVM when the underlying feature space is complex or even infinite-dimensional.
- The kernel trick itself is quite complex and is beyond the scope of this article.

Important Parameters in Kernelized SVC (Support Vector Classifier)

1. **The Kernel:** The kernel, is selected based on the type of data and also the type of transformation. By default, the kernel is Radial Basis Function Kernel (RBF).
2. **Gamma :** This parameter decides how far the influence of a single training example reaches during transformation, which in turn affects how tightly the decision boundaries

end up surrounding points in the input space. If there is a small value of gamma, points farther apart are considered similar. So more points are grouped together and have smoother decision boundaries (maybe less accurate). Larger values of gamma cause points to be closer together (may cause overfitting).

3. **The 'C' parameter:** This parameter controls the amount of regularization applied to the data. Large values of C mean low regularization which in turn causes the training data to fit very well (may cause overfitting). Lower values of C mean higher regularization which causes the model to be more tolerant of errors (may lead to lower accuracy).

Pros of Kernelized SVM:

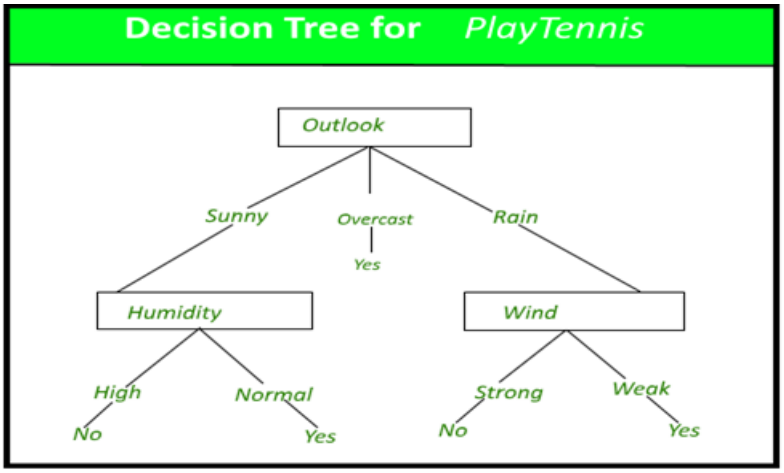
1. They perform very well on a range of datasets.
2. They are versatile: different kernel functions can be specified, or custom kernels can also be defined for specific datatypes.
3. They work well for both high and low dimensional data.

Cons of Kernelized SVM:

1. Efficiency (running time and memory usage) decreases as the size of the training set increases.
2. Needs careful normalization of input data and parameter tuning.
3. Does not provide a direct probability estimator.
4. Difficult to interpret why a prediction was made.

Decision Tree

- Decision Tree is the most powerful and popular tool for classification and prediction.
- A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.



A decision tree for the concept PlayTennis.

Construction of Decision Tree:

- A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning.
- The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.
- The construction of a decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.
- Decision trees can handle high-dimensional data.
- In general decision tree, classifier has good accuracy.
- Decision tree induction is a typical inductive approach to learn knowledge on classification.

Short note on Decision Tree:-

- A decision tree which is also known as prediction tree refers a tree structure to mention the sequences of decisions as well as consequences.
- Considering the input X = (X1, X2,… Xn), the aim is to predict a response or output variable Y.
- Each element in the set (X1, X2,…Xn) is known as input variable.It is possible to achieve the prediction by the process of building a decision tree which has test points as well as branches.
- At each test point, it is decided to select a particular branch and traverse down the tree.

- Ultimately, a final point is reached, and it will be easy to make prediction.
- In a decision tree, all the test points exhibit testing specific input variables (or attributes), and the developed decision tree is represented by the branches.
- Because of flexibility as well as simple visualization, decision trees are mostly probably deployed in data mining applications for the purpose of classification.
- In the decision tree, the input values are considered as categorical or continuous.
- A structure of test points (known as nodes) and branches is established by the decision tree by which the decision being made will be represented.
- Leaf node is the one which do not have further branches. The returning value of leaf nodes is class labels while in some cases they return the probability scores.
- It is possible to convert decision tree into a set of decision rules.
- There are two types of Decision trees: classification trees and regression trees
- Classification trees are generally applied to output variables which are categorical and mostly binary in nature, for example yes or no, sale or not, and so on.
- Whereas regression trees are applied to output variables which are numeric or continuous, for example predicted price of a consumer good.
- In variety of situations, it is possible to apply decision tree. It is easy to represent them in a visual way, and the analogous straightforward.
- Also as the result is a sequence of logical if-then statements, there is no any presence of underlying assumption regarding a linear or nonlinear relationship between the input variables and the response variable.

Decision Tree Representation:

- Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance.
- An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure.
- This process is then repeated for the subtree rooted at the new node. The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returns the classification associated with the particular leaf.(in this case Yes or No).

For example, the instance

(Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong)

- would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance.
- In other words, we can say that the decision tree represents a disjunction of conjunctions of constraints on the attribute values of instances.

(Outlook = Sunny ^ Humidity = Normal) v (Outlook = Overcast) v (Outlook = Rain ^ Wind = Weak)

Gini Index:

- Gini Index is a score that evaluates how accurate a split is among the classified groups.
- Gini index evaluates a score in the range between 0 and 1, where 0 is when all observations belong to one class, and 1 is a random distribution of the elements within classes.
- In this case, we want to have a Gini index score as low as possible.
- Gini Index is the evaluation metrics we shall use to evaluate our Decision Tree Model.

Naïve Bayes Classifier Algorithm

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

- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Why is it called Naïve Bayes

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve:

- It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features.
- Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple.
- Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes:

- It is called Bayes because it depends on the principle of Bayes' Theorem.

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

- Suppose we have a dataset of weather conditions and corresponding target variable "Play".
- So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions.

So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

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

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

Problem: If the weather is sunny, then the Player should play or not?

Solution: To solve this, first consider the below dataset:

Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Applying Bayes'theorem:

P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)

P(Sunny|Yes)= 3/10= 0.3

P(Sunny)= 0.35

P(Yes)=0.71

So P(Yes|Sunny) = 0.3*0.71/0.35= 0.60

P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)

P(Sunny|NO)= 2/4=0.5

P(No)= 0.29

P(Sunny)= 0.35

So P(No|Sunny)= 0.5*0.29/0.35 = 0.41

So as we can see from the above calculation that P(Yes|Sunny)>P(No|Sunny)

Hence on a Sunny day, Player can play the game.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

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 analysis.

Bayes Theorem in Machine learning

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.

The formula for Bayes' theorem is given as:

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

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

- Machine Learning is one of the most emerging technology of Artificial Intelligence.
- We are living in the 21th century which is completely driven by new technologies and gadgets in which some are yet to be used and few are on its full potential.
- Similarly, Machine Learning is also a technology that is still in its developing phase.
- There are lots of concepts that make machine learning a better technology such as supervised learning, unsupervised learning, reinforcement learning, perceptron models, Neural networks, etc.

- In this article "Bayes Theorem in Machine Learning", we will discuss another most important concept of Machine Learning theorem i.e., Bayes Theorem.
- But before starting this topic you should have essential understanding of this theorem such as what exactly is Bayes theorem, why it is used in Machine Learning, examples of Bayes theorem in Machine Learning and much more.
- So, let's start the brief introduction of Bayes theorem.

Introduction to Bayes Theorem in Machine Learning

- Bayes theorem is given by an English statistician, philosopher, and Presbyterian minister named Mr. Thomas Bayes in 17^{th} century.
- Bayes provides their thoughts in decision theory which is extensively used in important mathematics concepts as Probability.
- Bayes theorem is also widely used in Machine Learning where we need to predict classes precisely and accurately.
- An important concept of Bayes theorem named Bayesian method is used to calculate conditional probability in Machine Learning application that includes classification tasks.
- Further, a simplified version of Bayes theorem (Naïve Bayes classification) is also used to reduce computation time and average cost of the projects.
- Bayes theorem is also known with some other name such as Bayes rule or Bayes Law.
- Bayes theorem helps to determine the probability of an event with random knowledge.
- It is used to calculate the probability of occurring one event while other one already occurred.
- It is a best method to relate the condition probability and marginal probability.
- Bayes Theorem is used to estimate the precision of values and provides a method for calculating the conditional probability.
- However, it is hypocritically a simple calculation but it is used to easily calculate the conditional probability of events where intuition often fails.
- Some of the data scientist assumes that Bayes theorem is most widely used in financial industries but it is not like that.
- Other than financial, Bayes theorem is also extensively applied in health and medical, research and survey industry, aeronautical sector, etc.

What is Bayes Theorem?

- Bayes theorem is one of the most popular machine learning concepts that helps to calculate the probability of occurring one event with uncertain knowledge while other one has already occurred.

Bayes' theorem can be derived using product rule and conditional probability of event X with known event Y:

- According to the product rule we can express as the probability of event X with known event Y as follows;

1. P(X ? Y)= P(X|Y) P(Y) {equation 1}
    - Further, the probability of event Y with known event X:

1. P(X ? Y)= P(Y|X) P(X) {equation 2}

Mathematically, Bayes theorem can be expressed by combining both equations on right hand side. We will get:

$$P(X|Y) = \frac{P(Y|X).P(X)}{P(Y)}$$

Here, both events X and Y are independent events which means probability of outcome of both events does not depends one another.

The above equation is called as Bayes Rule or Bayes Theorem.

- P(X|Y) is called as posterior, which we need to calculate. It is defined as updated probability after considering the evidence.
- P(Y|X) is called the likelihood. It is the probability of evidence when hypothesis is true.

- P(X) is called the prior probability, probability of hypothesis before considering the evidence
- P(Y) is called marginal probability. It is defined as the probability of evidence under any consideration.

Hence, Bayes Theorem can be written as:

posterior = likelihood * prior / evidence

Prerequisites for Bayes Theorem

- While studying the Bayes theorem, we need to understand few important concepts.

These are as follows:

1. Experiment

- An experiment is defined as the planned operation carried out under controlled condition such as tossing a coin, drawing a card and rolling a dice, etc.

2. Sample Space

- During an experiment what we get as a result is called as possible outcomes and the set of all possible outcome of an event is known as sample space.

  For example, if we are rolling a dice, sample space will be:

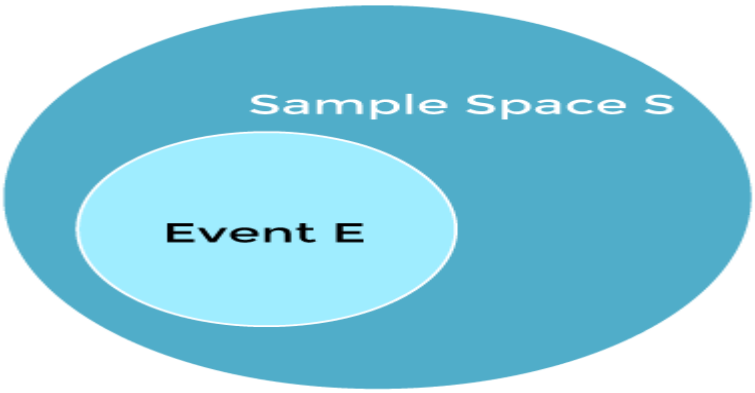
  S1 = {1, 2, 3, 4, 5, 6}

  Similarly, if our experiment is related to toss a coin and recording its outcomes, then sample space will be:

  S2 = {Head, Tail}

3. Event

- Event is defined as subset of sample space in an experiment.
- Further, it is also called as set of outcomes.



Assume in our experiment of rolling a dice, there are two event A and B such that;

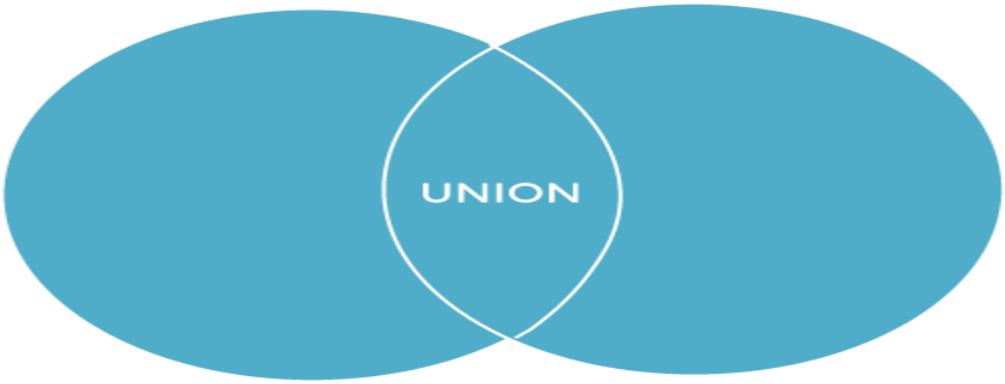
A = Event when an even number is obtained = {2, 4, 6}

B = Event when a number is greater than 4 = {5, 6}

- Probability of the event A "P(A)"= Number of favourable outcomes / Total number of possible outcomes
  P(E) = 3/6 =1/2 =0.5
- Similarly, Probability of the event B "P(B)"= Number of favourable outcomes / Total number of possible outcomes
  =2/6
  =1/3
  =0.333

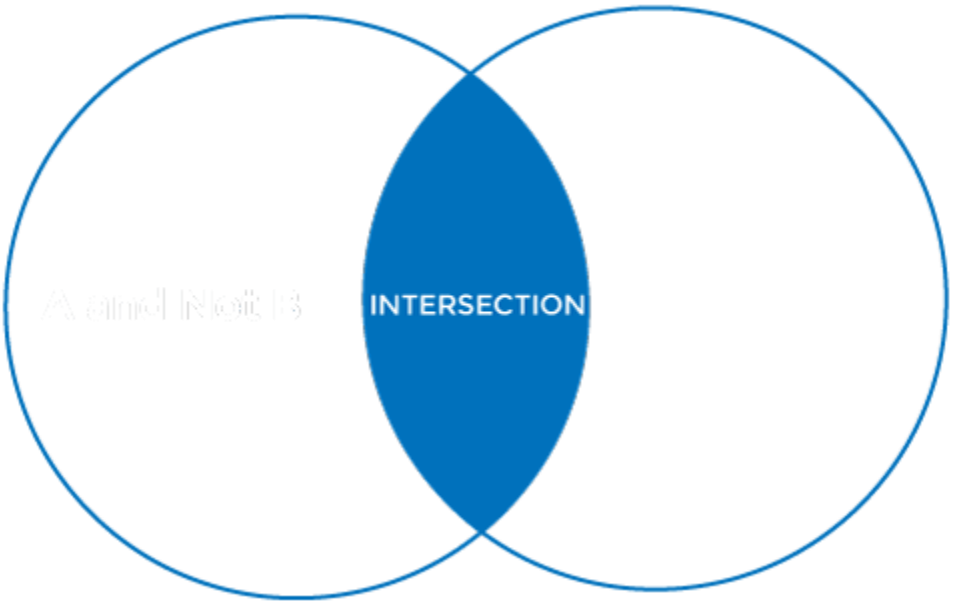
Union of event A and B:

A∪B = {2, 4, 5, 6}

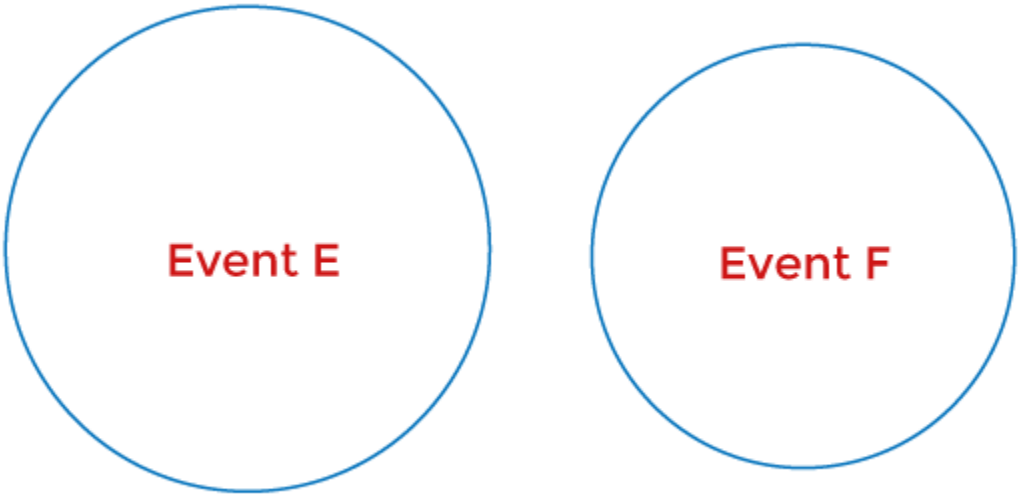


Intersection of event A and B:

A∩B= {6}



- Disjoint Event: If the intersection of the event A and B is an empty set or null then such events are known as disjoint event or mutually exclusive events also.



DISJOINT EVENTS

4. Random Variable:

- It is a real value function which helps mapping between sample space and a real line of an experiment.
- A random variable is taken on some random values and each value having some probability.
- However, it is neither random nor a variable but it behaves as a function which can either be discrete, continuous or combination of both.

5. Exhaustive Event:

- As per the name suggests, a set of events where at least one event occurs at a time, called exhaustive event of an experiment.
- Thus, two events A and B are said to be exhaustive if either A or B definitely occur at a time and both are mutually exclusive for e.g., while tossing a coin, either it will be a Head or may be a Tail.

6. Independent Event:

- Two events are said to be independent when occurrence of one event does not affect the occurrence of another event.

- In simple words we can say that the probability of outcome of both events does not depends one another.

Mathematically, two events A and B are said to be independent if:

$P(A \cap B) = P(AB) = P(A)*P(B)$

7. Conditional Probability:

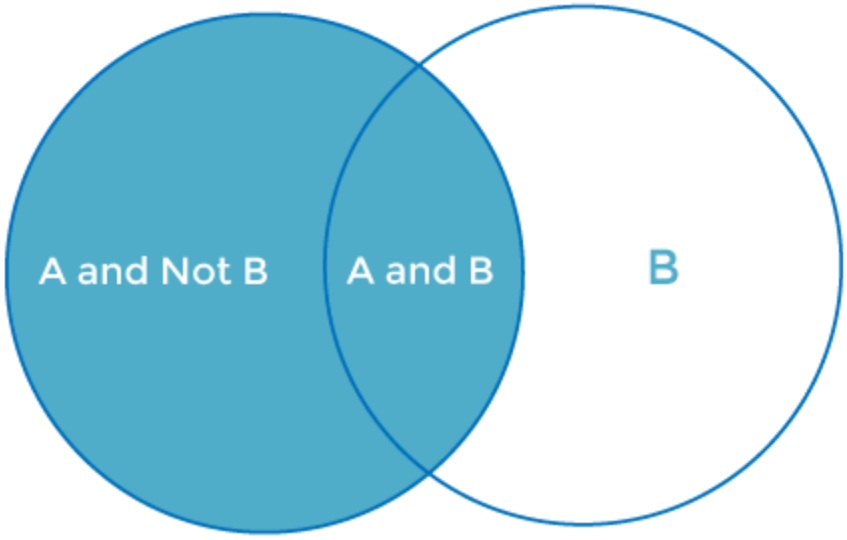
- Conditional probability is defined as the probability of an event A, given that another event B has already occurred (i.e. A conditional B). This is represented by P(A|B) and we can define it as:

$$P(A|B) = P(A \cap B) / P(B)$$

8. Marginal Probability:

- Marginal probability is defined as the probability of an event A occurring independent of any other event B.
- Further, it is considered as the probability of evidence under any consideration.

$$P(A) = P(A|B)*P(B) + P(A|\sim B)*P(\sim B)$$



Here ~B represents the event that B does not occur.

Ensemble Methods

Stacking in Machine Learning

- There are many ways to ensemble models in machine learning, such as Bagging, Boosting, and stacking.
- Stacking is one of the most popular ensemble machine learning techniques used to predict multiple nodes to build a new model and improve model performance.
- Stacking enables us to train multiple models to solve similar problems, and based on their combined output, it builds a new model with improved performance.
- In this topic, "Stacking in Machine Learning", we will discuss a few important concepts related to stacking, the general architecture of stacking, important key points to implement stacking, and how stacking differs from bagging and boosting in machine learning.
- Before starting this topic, first, understand the concepts of the ensemble in machine learning. So, let's start with the definition of ensemble learning in machine learning.

What is Ensemble learning in Machine Learning?

- Ensemble learning is one of the most powerful machine learning techniques that use the combined output of two or more models/weak learners and solve a particular computational intelligence problem.
- E.g., a Random Forest algorithm is an ensemble of various decision trees combined.
- Ensemble learning is primarily used to improve the model performance, such as classification, prediction, function approximation, etc.
- In simple words, we can summarise the ensemble learning as follows:
- An ensembled model is a machine learning model that combines the predictions from two or more models.”

There are 3 most common ensemble learning methods in machine learning. These are as follows:

- Bagging
- Boosting
- Stacking

However, we will mainly discuss Stacking on this topic.

1. Bagging:

Bagging is a method of ensemble modeling, which is primarily used to solve supervised machine learning problems. It is generally completed in two steps as follows:

- **Bootstrapping**: It is a random sampling method that is used to derive samples from the data using the replacement procedure. In this method, first, random data samples are fed to the primary model, and then a base learning algorithm is run on the samples to complete the learning process.
- **Aggregation:** This is a step that involves the process of combining the output of all base models and, based on their output, predicting an aggregate result with greater accuracy and reduced variance.

Example: In the Random Forest method, predictions from multiple decision trees are ensembled parallelly. Further, in regression problems, we use an average of these predictions to get the final output, whereas, in classification problems, the model is selected as the predicted class.

Bagging

- Bagging is used when our objective is to reduce the variance of a decision tree.
- Here the concept is to create a few subsets of data from the training sample, which is chosen randomly with replacement.
- Now each collection of subset data is used to prepare their decision trees thus, we end up with an ensemble of various models.

- The average of all the assumptions from numerous tress is used, which is more powerful than a single decision tree.
- Random Forest is an expansion over bagging.
- It takes one additional step to predict a random subset of data.
- It also makes the random selection of features rather than using all features to develop trees. When we have numerous random trees, it is called the Random Forest.

These are the following steps which are taken to implement a Random forest:

- Let us consider X observations Y features in the training data set. First, a model from the training data set is taken randomly with substitution.
- The tree is developed to the largest.
- The given steps are repeated, and prediction is given, which is based on the collection of predictions from n number of trees.

Advantages of using Random Forest technique:

- It manages a higher dimension data set very well.
- It manages missing quantities and keeps accuracy for missing data.

Disadvantages of using Random Forest technique:

Since the last prediction depends on the mean predictions from subset trees, it won't give precise value for the regression model.

2. Boosting

- Boosting is an ensemble method that enables each member to learn from the preceding member's mistakes and make better predictions for the future.
- Unlike the bagging method, in boosting, all base learners (weak) are arranged in a sequential format so that they can learn from the mistakes of their preceding learner.
- Hence, in this way, all weak learners get turned into strong learners and make a better predictive model with significantly improved performance.

- We have a basic understanding of ensemble techniques in machine learning and their two common methods, i.e., bagging and boosting.
- Now, let's discuss a different paradigm of ensemble learning, i.e., Stacking.

Boosting:

- Boosting is another ensemble procedure to make a collection of predictors.
- In other words, we fit consecutive trees, usually random samples, and at each step, the objective is to solve net error from the prior trees.
- If a given input is misclassified by theory, then its weight is increased so that the upcoming hypothesis is more likely to classify it correctly by consolidating the entire set at last converts weak learners into better performing models.
- Gradient Boosting is an expansion of the boosting procedure.

1. Gradient Boosting = Gradient Descent + Boosting

It utilizes a gradient descent algorithm that can optimize any differentiable loss function. An ensemble of trees is constructed individually, and individual trees are summed successively. The next tree tries to restore the loss (It is the difference between actual and predicted values).

Advantages of using Gradient Boosting methods:

- It supports different loss functions.
- It works well with interactions.

Disadvantages of using a Gradient Boosting methods:

It requires cautious tuning of different hyper-parameters.

Bagging Vs Boosting

- We all use the Decision Tree Technique on day to day life to make the decision.
- Organizations use these supervised machine learning techniques like Decision trees to make a better decision and to generate more surplus and profit.
- Ensemble methods combine different decision trees to deliver better predictive results, afterward utilizing a single decision tree.
- The primary principle behind the ensemble model is that a group of weak learners come together to form an active learner.
- There are two techniques given below that are used to perform ensemble decision tree.

Bagging	Boosting
Various training data subsets are randomly drawn with replacement from the whole training dataset.	Each new subset contains the components that were misclassified by previous models.
Bagging attempts to tackle the over-fitting issue.	Boosting tries to reduce bias.
If the classifier is unstable (high variance), then we need to apply bagging.	If the classifier is steady and straightforward (high bias), then we need to apply boosting.
Every model receives an equal weight.	Models are weighted by their performance.
Objective to decrease variance, not bias.	Objective to decrease bias, not variance.
It is the easiest way of connecting predictions that belong to the same type.	It is a way of connecting predictions that belong to the different types.
Every model is constructed independently.	New models are affected by the performance of the previously developed model.