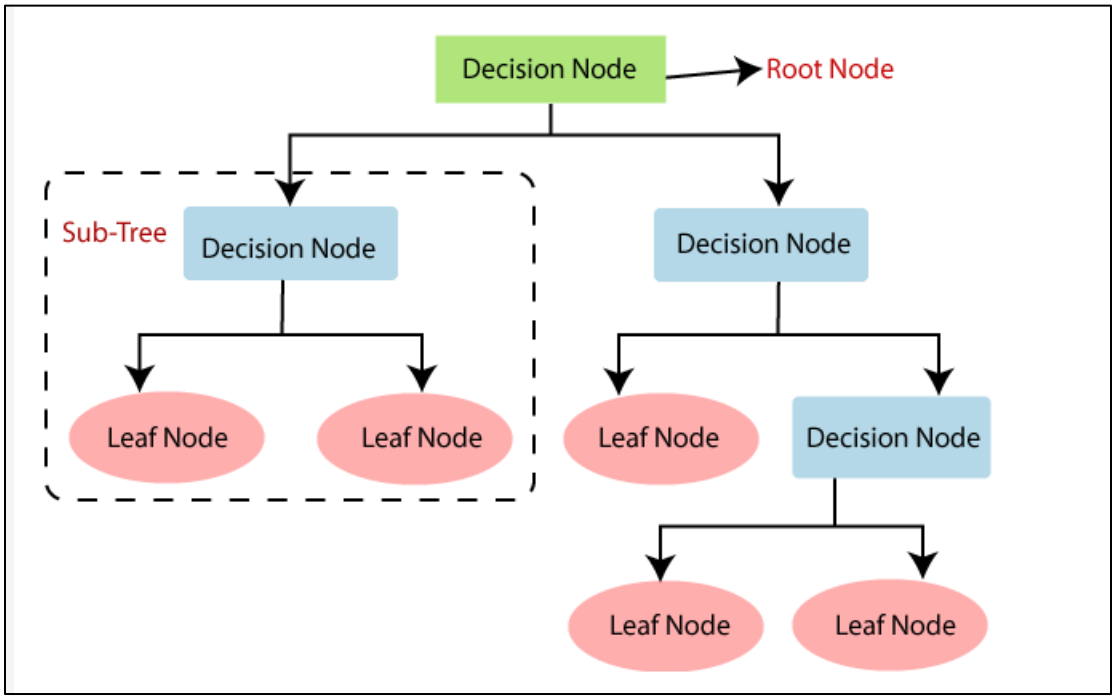
Decision Tree

- 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.
- In a Decision tree, there are two nodes-
    - **Decision Nodes** – They are used to make any decision and have multiple branches
    - **Leaf Nodes** – They 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
- In order to build a tree, we use the **CART** algorithm, which stands for **Classification and Regression Tree algorithm**.



Pruning- Pruning is the process of removing the unwanted branches from the tree

Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM

- **Entropy**
  - Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data
- **Gini**
  - 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 the low Gini index should be preferred as compared to the high Gini index.
- **Information Gain**
  - Information gain is the measurement of changes in entropy after the segmentation of a dataset 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

Steps to make decision tree

1. Begin the tree with the root node [**S**] which contains the complete dataset.
2. Find the best attribute in the dataset using **Attribute Selection Measure (ASM)**
3. Divide the S into subsets that contains possible values for the best attributes
4. Generate the decision tree node, which contains the best attribute.
5. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node

Advantages of Decision Tree

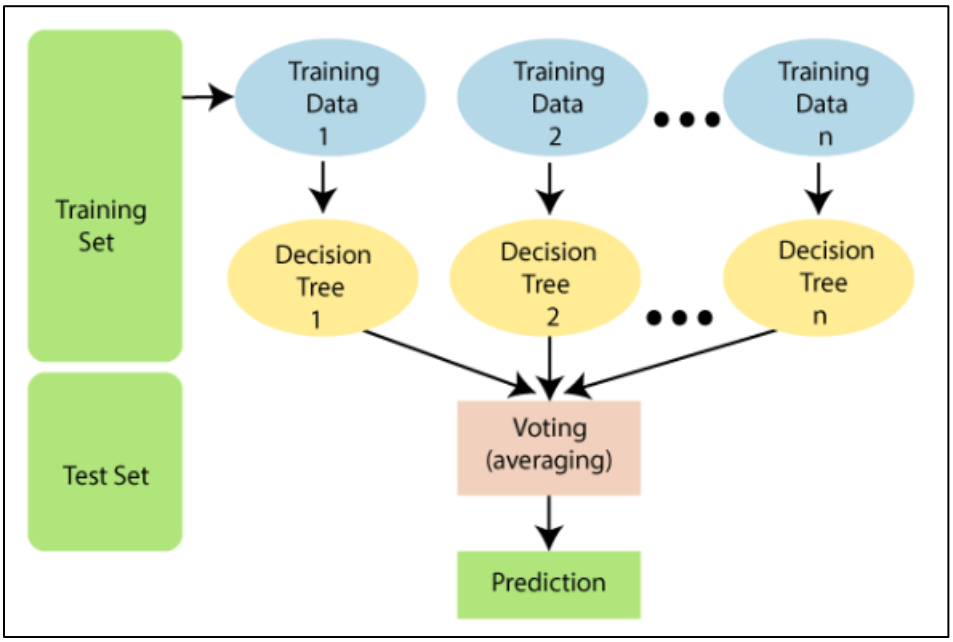
- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems
- It helps to think about all the possible outcomes for a problem

Disadvantages of Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm

Random Forest

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique
- It can be used for both Classification and Regression problems in ML.
- Mainly used for classification problem
- It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to 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.



Steps involved in Random Forest

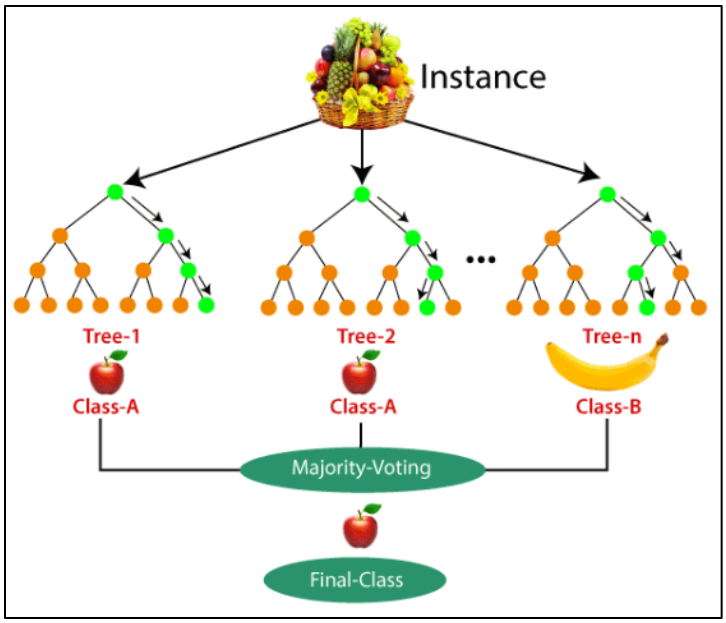
1. Select random K data points from the training set
2. Build the decision trees associated with the selected data points (Subsets).
3. Choose the number N for decision trees that you want to build.
4. Repeat Step 1 & 2.
5. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes

Advantages

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently
- It can also maintain accuracy when a large proportion of data is missing.

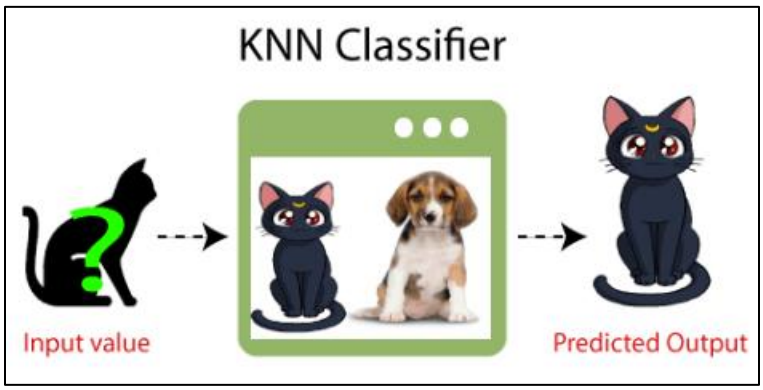
Disadvantages

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks



K-Nearest Neighbour

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique
- 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.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.



How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Advantages of KNN

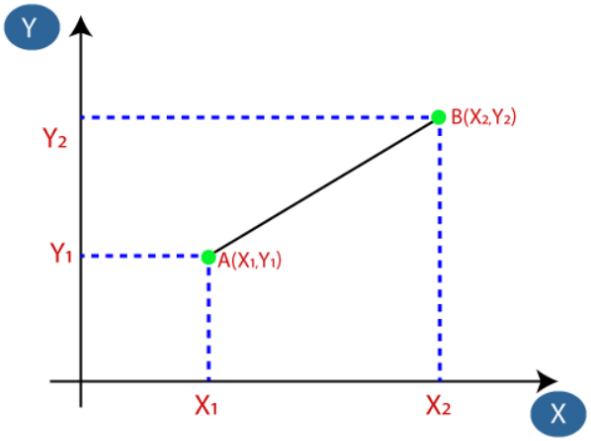
- It is simple to implement
- It is robust to the noisy training data
- It can be more effective if the training data is large

Disadvantages of KNN

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

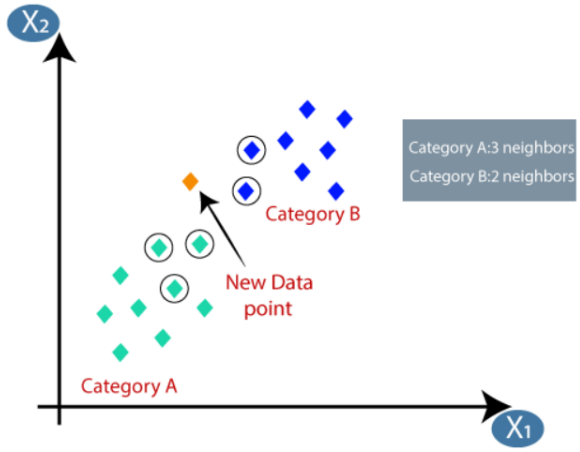
Example

- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



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

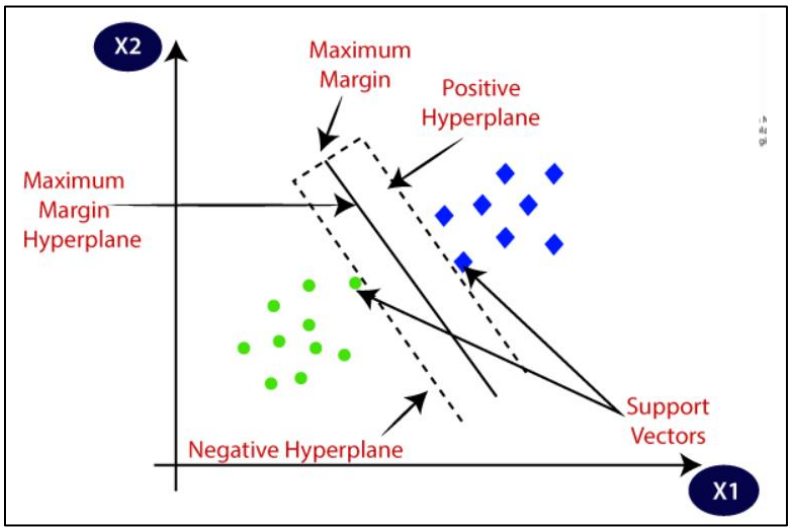
- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



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

Support Vector Machine

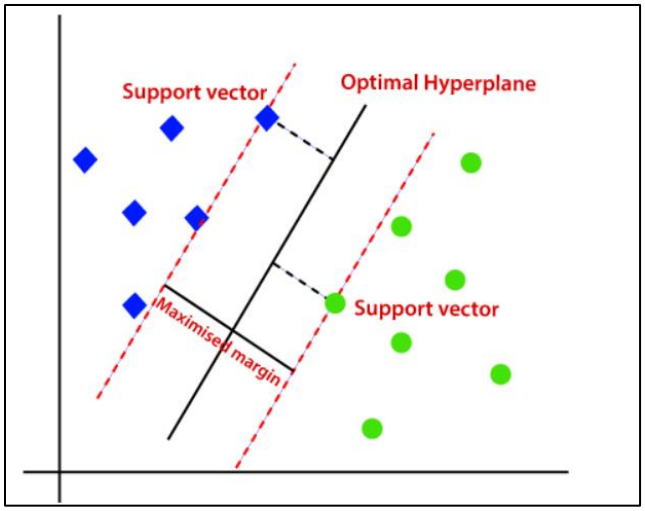
- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms
- It is used for Classification as well as Regression problems and primarily, it is used for Classification problems
- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate 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 as support vectors, and hence algorithm is termed as Support Vector Machine.



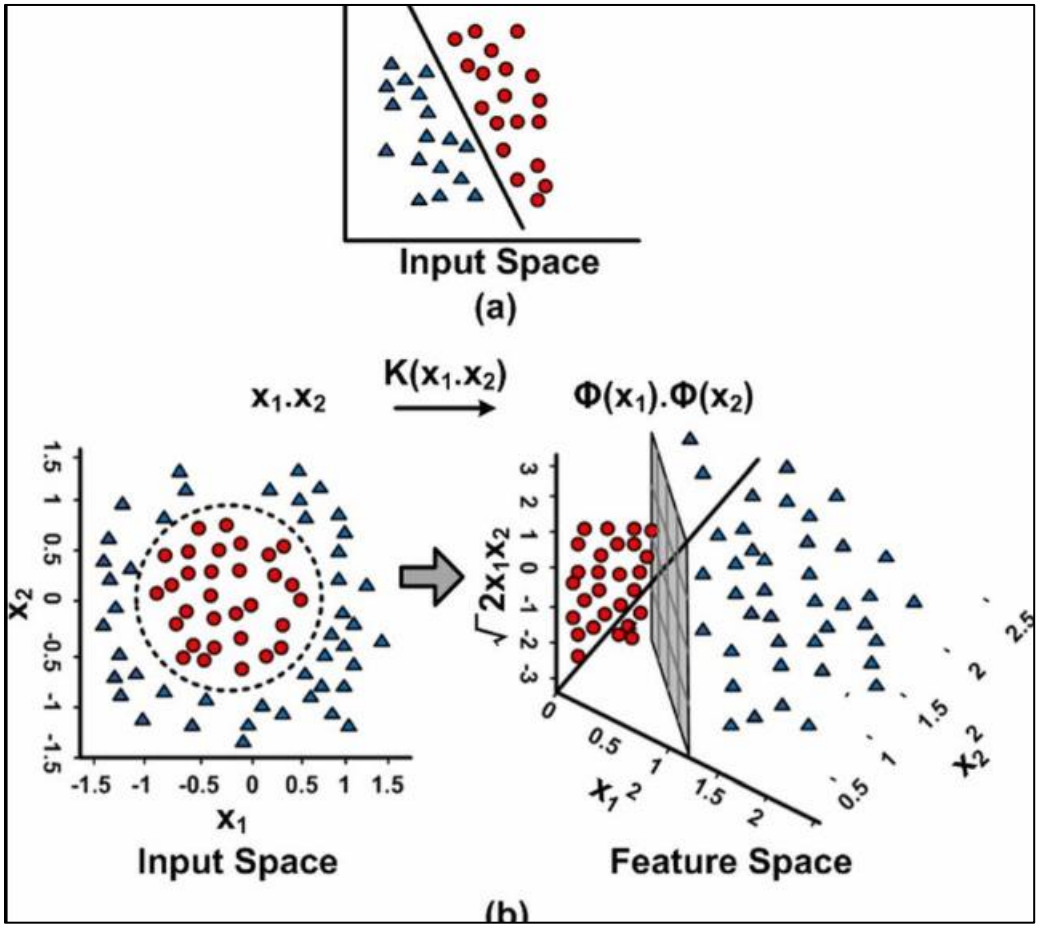
Types of SVM

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.



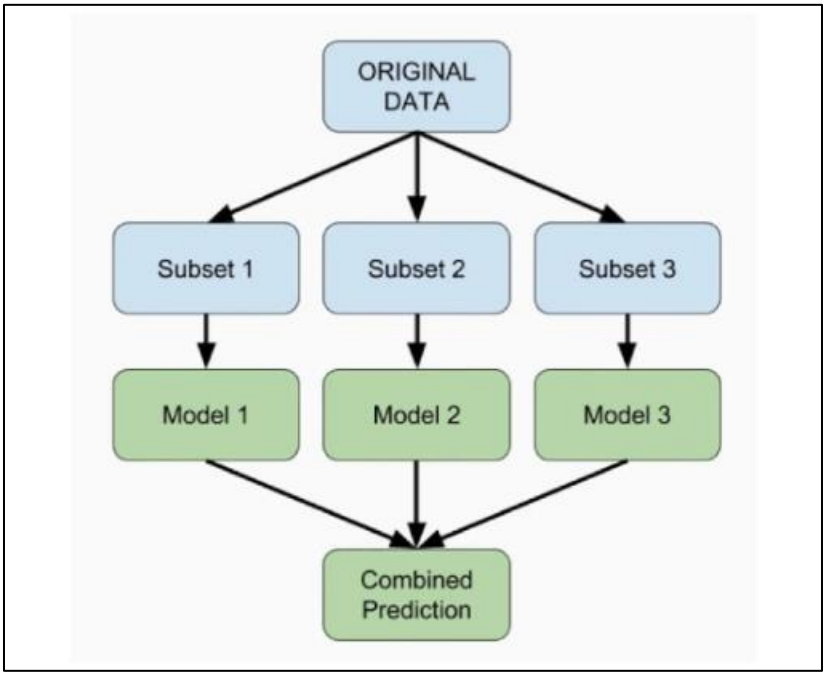
(Linear SVM)



(Non Linear SVM)

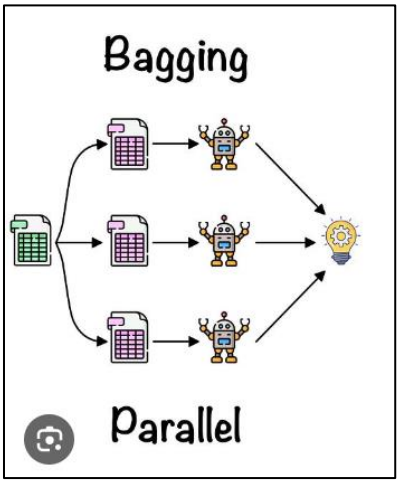
Ensemble Learning

- Ensemble learning is a machine learning technique that involves combining the predictions of multiple models to improve the overall performance and accuracy of a predictive model
- The basic idea behind ensemble learning is that by combining the predictions of multiple models, the weaknesses of individual models can be reduced, and their strengths can be improved to make more accurate and robust predictions.



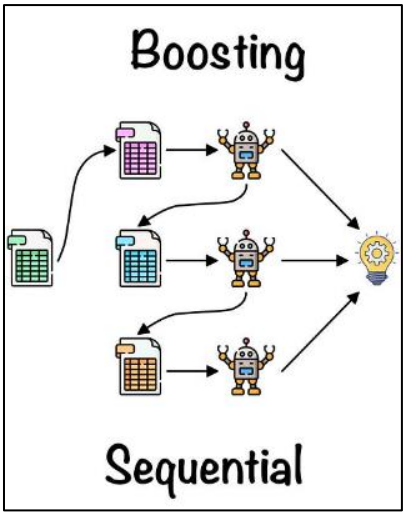
Bagging

- Bagging involves training multiple instances of the same model on different subsets of the training data
- Each model is trained independently, and the final prediction is made by averaging or voting on the predictions of all the models.
- One of the most famous bagging algorithms is the Random Forest, which is an ensemble of decision trees



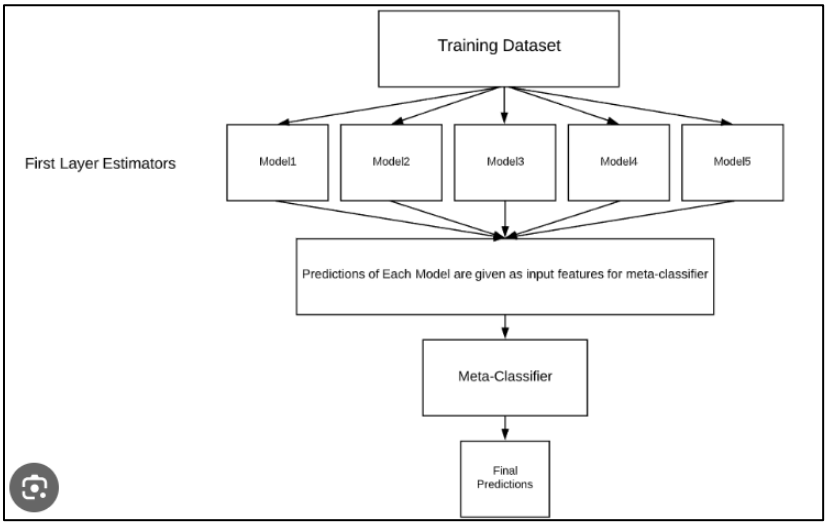
Boosting

- Boosting focuses on training a sequence of models where each model corrects the errors of the previous one
- Common boosting algorithms include AdaBoost, Gradient Boosting



Stacking

- Stacking, or stacked generalization, involves training multiple models, called base models, and then training a higher-level model, often called a meta-learner, that combines the predictions of the base models.
- Stacking is a more advanced ensemble technique that can capture complex relationships among the base models.



Voting

- Voting ensembles combine the predictions of multiple models by either taking a majority vote (hard voting) or averaging the predicted probabilities (soft voting) to make the final prediction.