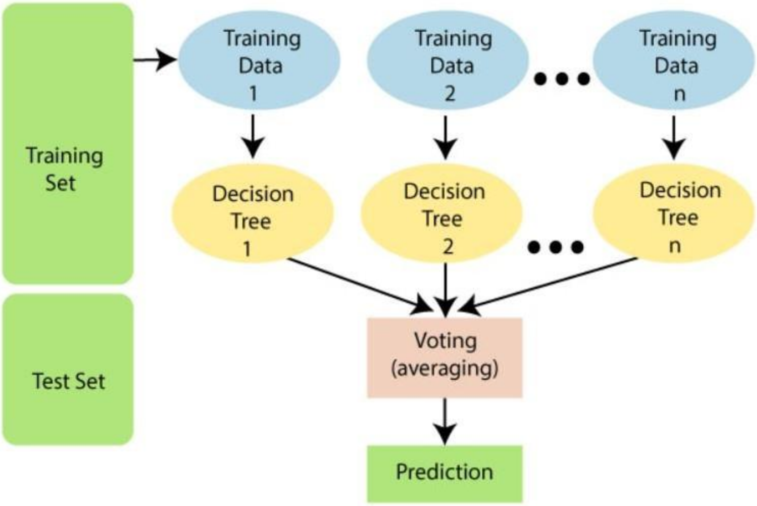
Random Forest Algorithm

- Random Forest is a **supervised machine** learning algorithm that can be used for both **classification** and **regression tasks**.
- It utilizes **ensemble learning**, combining multiple decision trees to improve predictive accuracy.
- Each decision tree is built on a different subset of the data, **reducing overfitting**.
- The final prediction is determined by majority vote among the decision trees.
- **More trees** in the forest generally lead to **higher accuracy**.
- They find application in supervised machine learning scenarios characterized by a labeled target variable.
- Suitable for both regression (numeric target variable) and classification (categorical target variable) problems.
- As an ensemble method, random forests consolidate predictions from multiple models.
- The constituent models within the random forest ensemble are decision trees.



Why use Random Forest?

- ✓ 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.
- ✓ It provides higher accuracy through cross validation
- ✓ Random forest algorithm can be used for both classifications and regression task.
- ✓ It has the power to handle a large data set with higher dimensionality.
- ✓ If there are more trees, it won't allow over-fitting trees in the model.
- ✓ Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.
- ✓ It has the power to handle a large data set with higher dimensionality.

How does Random Forest algorithm work?

The Random Forest operates through a two-phase process. Initially, it creates the random forest by combining N decision trees, and subsequently, it makes predictions for each tree formed in the first phase.

The working process is outlined in the following steps and diagram:

Step-1: Random Data Selection

Randomly select K data points from the training set.

Step-2: Decision Tree Construction

Build decision trees associated with the selected data points, forming subsets.

Step-3: Decision Tree Number Specification

Choose the number N for decision trees that need to be constructed.

Step-4: Iterative Process

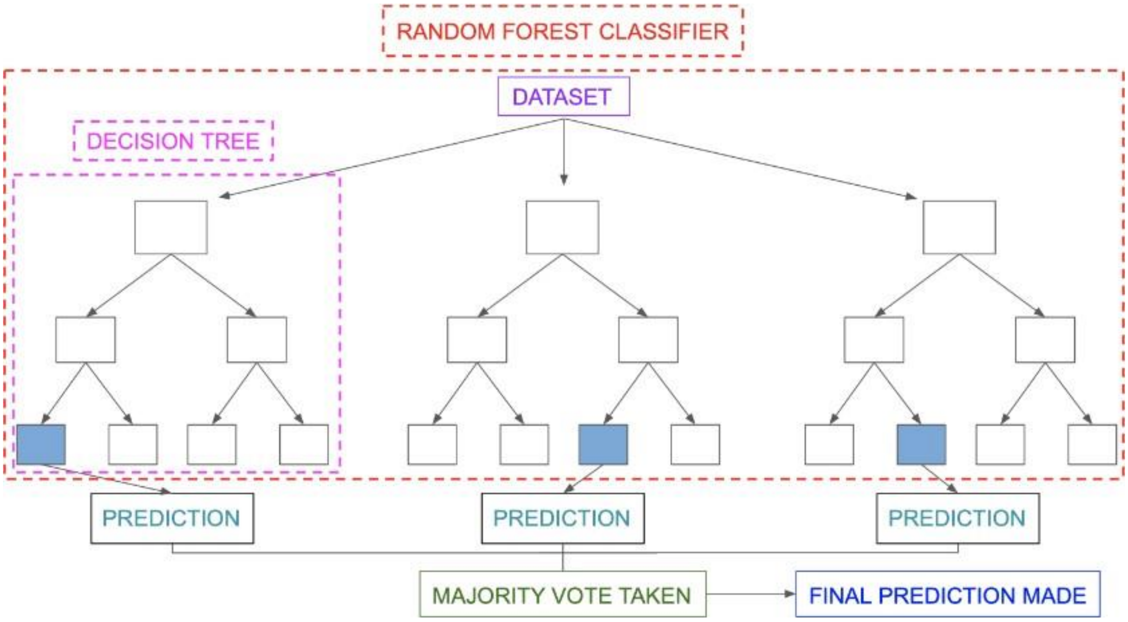
Repeat Steps 1 and 2 for the specified number of decision trees (N).

Step-5: Prediction Aggregation

For new data points, obtain predictions from each decision tree.

Assign the new data points to the category that receives the majority votes among the decision trees.

This process ensures the creation of a robust and diverse ensemble of decision trees, contributing to the Random Forest's predictive power.



Applications of Random Forest

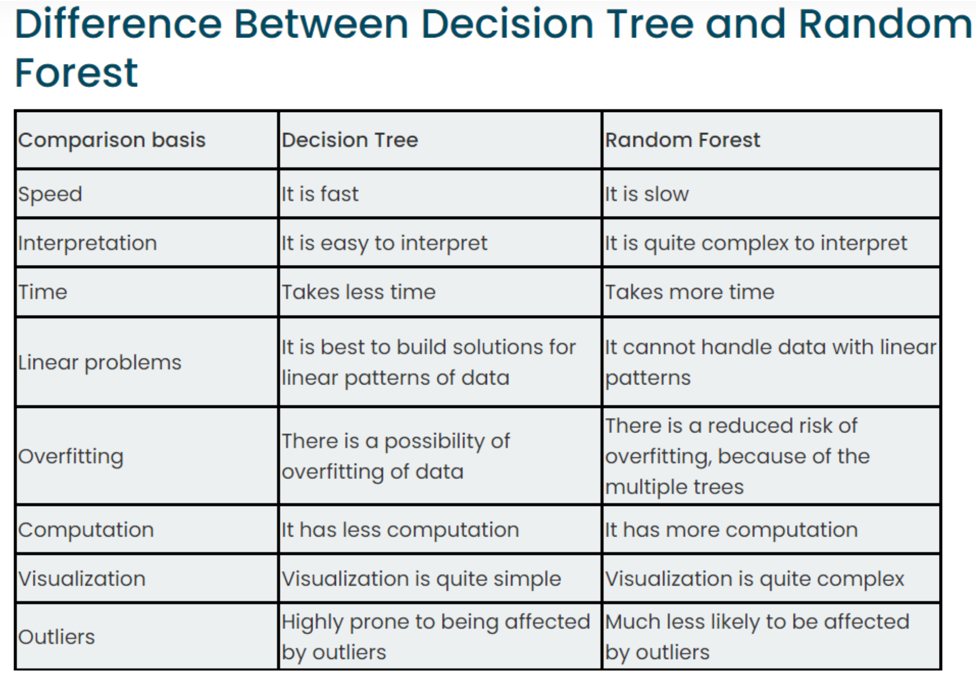
1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease canbe identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

Advantages of Random Forest

- It reduces overfitting in decision trees and helps to improve the accuracy
- It is flexible to both classification and regression problems
- It works well with both categorical and continuous values
- It automates missing values present in the data
- Normalising of data is not required as it uses a rule-based approach.
- Handles large datasets with high dimensionality efficiently.
- Provides feature importance for better insights.

Disadvantages of Random Forest

- It requires much computational power as well as resources as it builds numerous trees tocombine their outputs.
- It also requires much time for training as it combines a lot of decision trees to determine theclass.
- Due to the ensemble of decision trees, it also suffers interpretability and fails to determinethe significance of each variable.

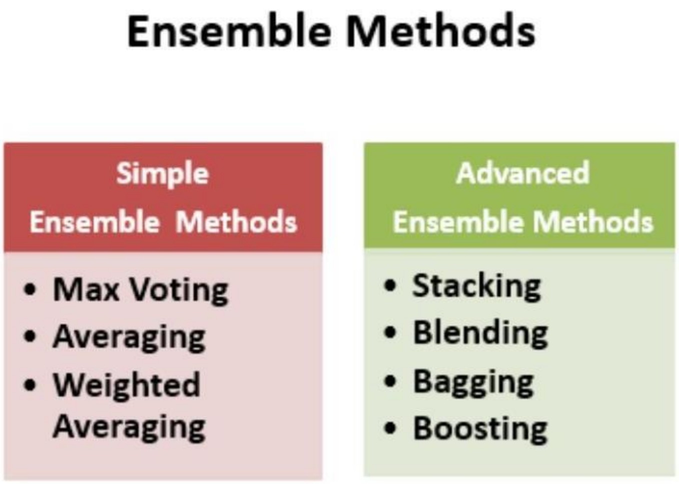


Difference Between Decision Tree and Random Forest

Comparison basis	Decision Tree	Random Forest
Speed	It is fast	It is slow
Interpretation	It is easy to interpret	It is quite complex to interpret
Time	Takes less time	Takes more time
Linear problems	It is best to build solutions for linear patterns of data	It cannot handle data with linear patterns
Overfitting	There is a possibility of overfitting of data	There is a reduced risk of overfitting, because of the multiple trees
Computation	It has less computation	It has more computation
Visualization	Visualization is quite simple	Visualization is quite complex
Outliers	Highly prone to being affected by outliers	Much less likely to be affected by outliers

Ensemble Methods

- ❖ Ensemble methods help minimize error in learning by reducing noise, bias, and variance.
- ❖ They improve the stability and accuracy of machine learning algorithms.
- ❖ Combining multiple classifiers reduces variance, especially for unstable classifiers.
- ❖ Bagging and Boosting use a pool of base learner algorithms, such as classification trees.



Ensemble Methods

Simple Ensemble Methods	Advanced Ensemble Methods
• Max Voting • Averaging • Weighted Averaging	• Stacking • Blending • Bagging • Boosting

1) Max Voting:

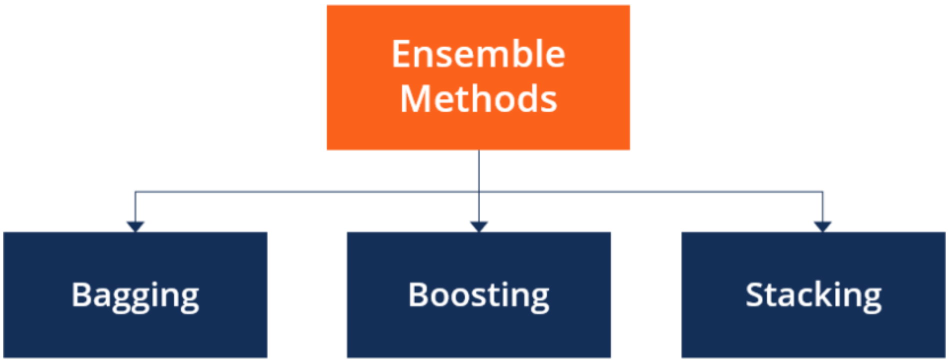
- ❖ Commonly used for classification problems.
- ❖ Each model makes predictions for individual data points.
- ❖ Predictions are considered as "votes".
- ❖ Final prediction is the outcome with the majority of votes.

2) **Averaging**:

- ❖ Multiple predictions are made for each data point.
- ❖ The average of all predictions is calculated.
- ❖ This average is used as the final prediction.

3) Weighted Average:

- An extension of the averaging method.
- Weights are assigned to each model based on its prediction.
- The weighted average of predictions is used as the final prediction.



1. Bagging (bootstrap aggregating)

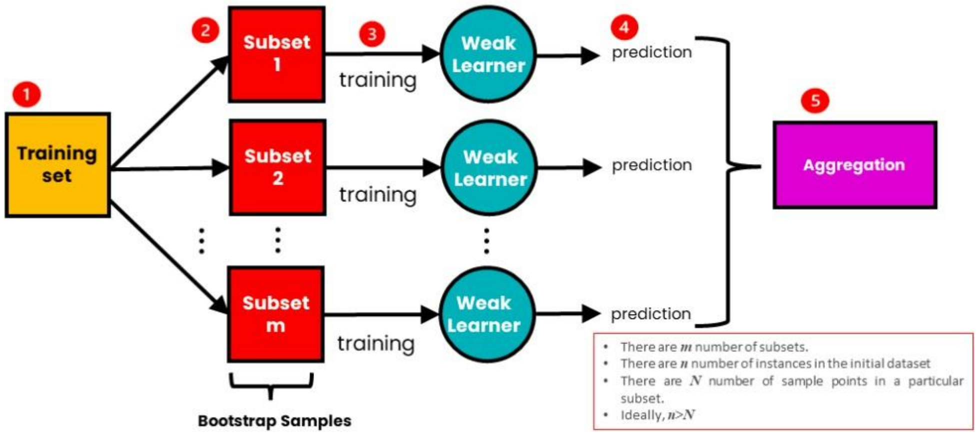
- It is a technique used to improve the accuracy of machine learning models.
- It is used for classification and regression task.
- It reduces variance by averaging the predictions of multiple base learners, which are typicallydecision trees.
- Bagging is effective in reducing overfitting and improving the stability of models.
- it can be computationally expensive and may introduce bias if not implemented correctly

Bagging consists of two steps:

- **bootstrapping:** Bootstrapping involves creating multiple training sets byrandomly sampling with replacement from the original dataset.
- **Aggregation:** Aggregation involves combining the

predictions of the baselearners, typically by averaging them.

The Process of Bagging (Bootstrap Aggregation)



- There are m number of subsets.
- There are n number of instances in the initial dataset
- There are N number of sample points in a particular subset.
- Ideally, $n>N$

2. Boosting

- Boosting is an ensemble technique.
- it improves the accuracy of machine learning models by combining weak learners into astrong learner.
- It works by arranging weak learners in a sequence, where each learner learns from themistakes of the previous learner.

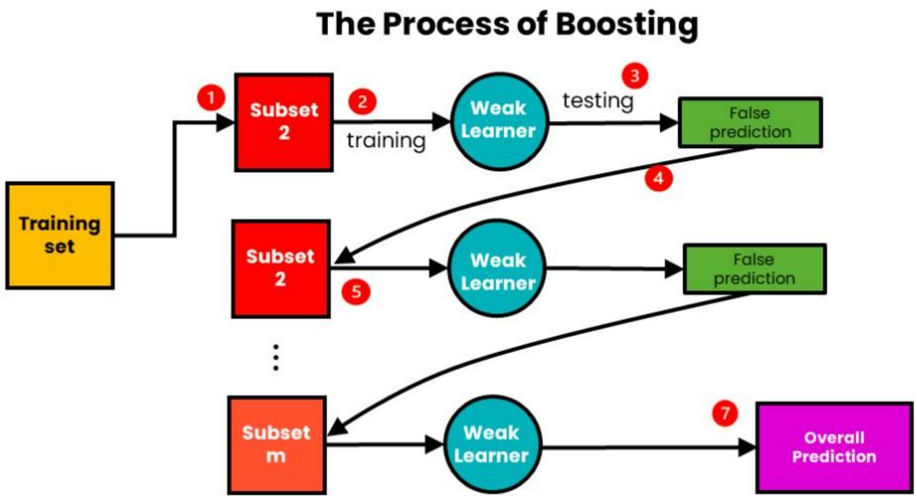
Boosting takes various forms including:

- **Gradient boosting:** Gradient boosting adds predictors sequentially, where each

  predictor corrects the errors of the previous predictor, using gradient descent to identify andcounter errors.
- **AdaBoost:** AdaBoost uses decision trees with a single split, known as decision stumps,and focuses on observations with similar weights.
- **XGBoost**: XGBoost utilizes decision trees with boosted gradient for enhanced speed andperformance, relying heavily on computational speed and target model performance.

Model training in gradient boosted machines follows a sequence, making implementation slowercompared to other methods.

Steps of Boosting

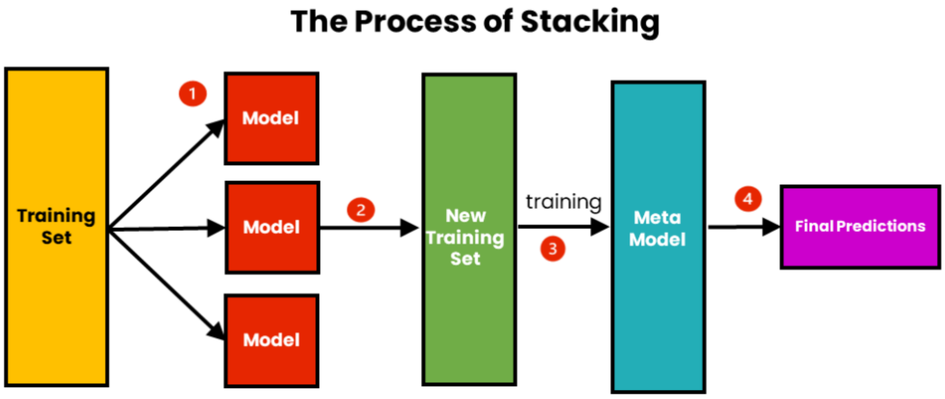


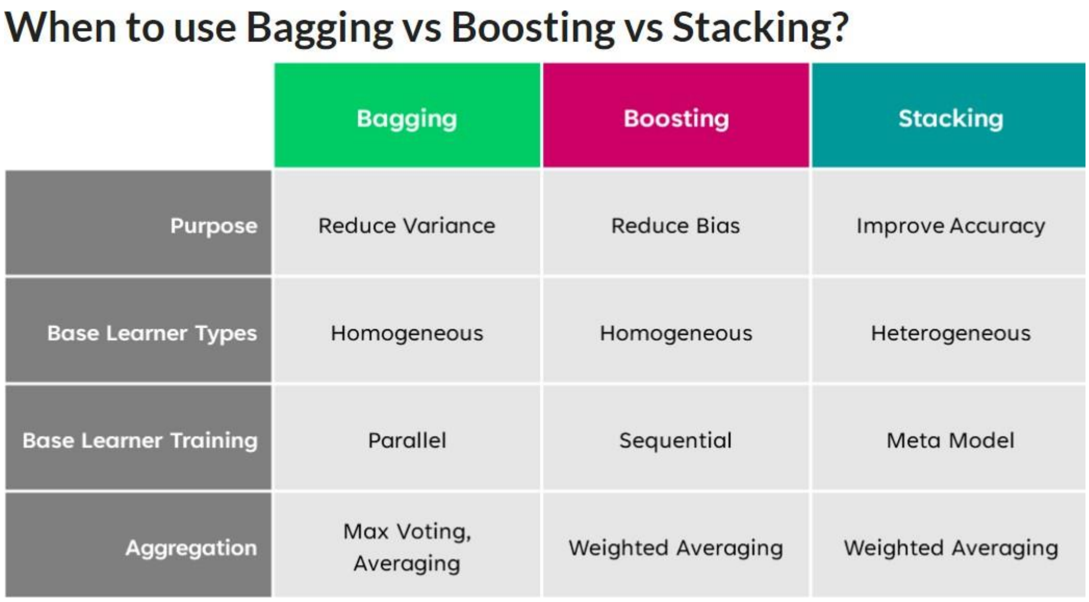
3. Stacking

- Stacking is an ensemble method that aims to improve prediction accuracy by combiningmultiple strong learners into a single robust model.
- It differs from bagging and boosting in that it combines strong learners, heterogeneousmodels, and involves creating a metamodel.
- The process involves training individual heterogeneous models on an initial dataset.
- These models make predictions, forming a new dataset based on those predictions.
- This new dataset is used to train a metamodel, which makes the final prediction.
- The prediction is combined using weighted averaging.
- Stacking's ability to combine strong learners allows it to incorporate

bagged or boosted models.

Steps of Stacking





When to use Bagging vs Boosting vs Stacking?

	Bagging	Boosting	Stacking
Purpose	Reduce Variance	Reduce Bias	Improve Accuracy
Base Learner Types	Homogeneous	Homogeneous	Heterogeneous
Base Learner Training	Parallel	Sequential	Meta Model
Aggregation	Max Voting, Averaging	Weighted Averaging	Weighted Averaging