Random Forest

What is a Random Forest Algorithm?

Random Forest algorithm is a powerful tree learning technique in Machine Learning to make predictions and then we do the voting of all the tress to make prediction. They are widely used for classification and regression task.

- It is a type of classifier that uses many decision trees to make predictions.
- It takes different random parts of the dataset to train each tree and then it combines the results by averaging them. This approach helps improve the accuracy of predictions. Random Forest is based on ensemble learning.

What is ensemble learning?

Ensemble learning combines the predictions of multiple models (called "weak learners" or "base models") to make a stronger, more reliable prediction. The goal is to reduce errors and improve performance.

It is like asking a group of experts for their opinions instead of relying on just one person. Each expert might make mistakes, but when you combine their knowledge, the final decision is often better and more accurate.

There are two main types of ensemble methods:

- Bagging (Bootstrap Aggregating): Models are trained independently on different subsets of the data, and their results are averaged or voted on.
- Boosting: Models are trained sequentially, with each one learning from the mistakes of the previous model.

Explain Boosting Algorithm.

Boosting is an ensemble technique that combines multiple weak learners to create a strong learner. Weak models are trained in series such that each next model tries to correct errors of the previous model until the entire training dataset is predicted correctly. One of the most well-known boosting algorithms is AdaBoost (Adaptive Boosting). Below is an overview of Boosting algorithm:

- Begin with a single weak learner and assign equal weights to all training examples.
- Train weak learners on these datasets.
- Boosting works by training models sequentially where each model focuses on correcting the errors of its predecessor. Boosting typically uses a single type of weak learner like decision trees.
- Boosting assigns weights to training datapoints. Misclassified examples receive higher weights in the next iteration so that next models pay more attention to them.

Explain Bagging Algorithm.

Bagging classifier can be used for both regression and classification tasks. Below is an overview of Bagging classifier algorithm:

- Divides the original training data into 'N' subsets and randomly selects a subset with replacement in some rows from other subsets. This step ensures that the base models are trained on diverse subsets of the data and there is no class imbalance.
- For each bootstrapped sample we train a base model independently on that subset of data. These weak models are trained in parallel to increase computational efficiency and reduce time consumption.
- We can use different base learners i.e different ML models as base learners to bring variety and robustness.
- To make a prediction on testing data combine the predictions of all base models.
- For classification tasks it can include majority voting or weighted majority while for regression it involves averaging the predictions.
- Some samples are excluded from the training subset of particular base models during the bootstrapping method. These "out-of-bag" samples can be used to estimate the model's performance without the need for cross-validation.
- After aggregating the predictions from all the base models, Bagging produces a final prediction for each instance.

What are the key features of Random Forest?

- Handles Missing Data: Automatically handles missing values during training, eliminating the need for manual imputation.
- Algorithm ranks features based on their importance in making predictions offering valuable insights for feature selection and interpretability.
- Scales Well with Large and Complex Data without significant performance degradation.
- Algorithm is versatile and can be applied to both classification tasks (e.g., predicting categories) and regression tasks (e.g., predicting continuous values).

Assumptions of Random Forest

- Each tree makes its own decisions: Every tree in the forest makes its own predictions without relying on others.
- Random parts of the data are used: Each tree is built using random samples and features to reduce mistakes.
- Enough data is needed: Sufficient data ensures the trees are different and learn unique patterns and variety.
- Different predictions improve accuracy: Combining the predictions from different trees leads to more accurate final results.

How does Random Forest Algorithm Works?

The random Forest algorithm works in several steps:

- Random Forest builds multiple decision trees using random samples of the data. Each tree is trained on a different subset of the data which makes each tree unique.
- When creating each tree the algorithm randomly selects a subset of features or variables to split the data rather than using all available features at a time. This adds diversity to the trees.
- Each decision tree in the forest makes a prediction based on the data it was trained on. When making final prediction random forest combines the results from all the trees.
    - For classification tasks the final prediction is decided by a majority vote. This means that the category predicted by most trees is the final prediction.
    - For regression tasks the final prediction is the average of the predictions from all the trees.
- The randomness in data samples and feature selection helps to prevent the model from overfitting making the predictions more accurate and reliable.

Advantages of Random Forest

- Decision trees run the risk of overfitting as they tend to tightly fit all the samples within training data. However, when there's a robust number of decision trees in a random forest, the classifier won't overfit the model since the averaging of uncorrelated trees lowers the overall variance and prediction error.
- Since random forest can handle both regression and classification tasks with a high degree of accuracy, it is a popular method among data scientists.
- Feature bagging also makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing.
- Random forest makes it easy to evaluate variable importance, or contribution, to the model. There are a few ways to evaluate feature importance. Gini importance and mean decrease in impurity (MDI) are usually used to measure how much the model's accuracy decreases when a given variable is excluded.

Disadvantages of Random Forest

- Since random forest algorithms can handle large data sets, they can provide more accurate predictions, but can be slow to process data as they are computing data for each individual decision tree.
- Since random forests process larger data sets, they'll require more resources to store that data.
- The prediction of a single decision tree is easier to interpret when compared to a forest of them.