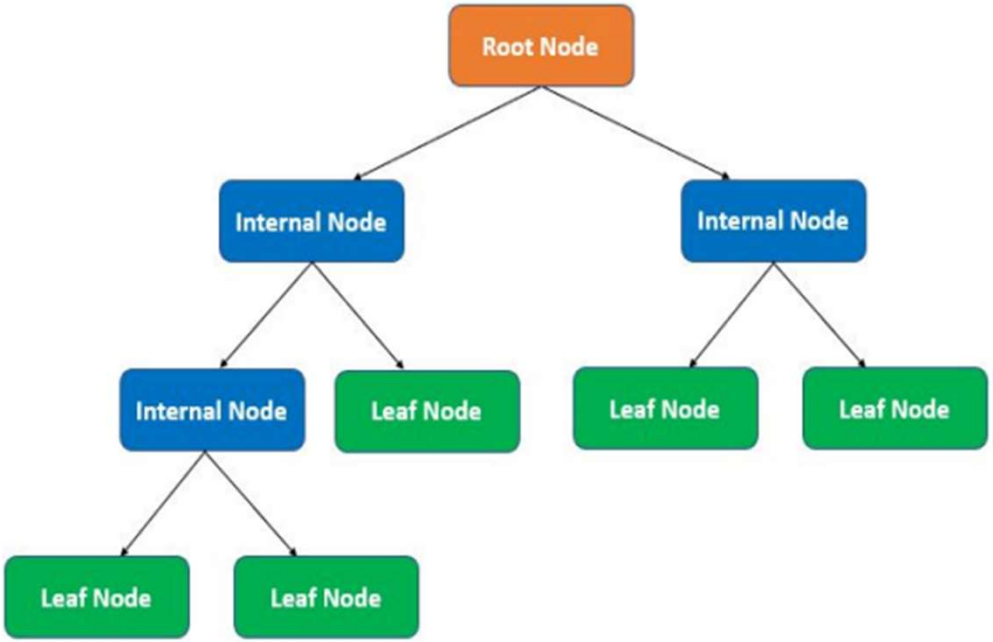
Random Forest

Decision Tree Classifier

Building the tree

- Recursive Process: Starts with the root and recursively splits data based on features.
- Stopping Criteria: Stops when a certain depth or impurity threshold is reached.



Random Forest

What is Ensemble Learning?

Definition:

- Combines multiple models to improve prediction accuracy and reduce errors.
- Works on the principle of **"wisdom of the crowd"**—multiple weak learners create a strong learner.

Why Ensemble Models?

- Single models (e.g., Decision Trees) may **overfit** or perform poorly on unseen data.
- Ensemble methods reduce variance, bias, and improve generalization.

Random Forest

- **Random Forest** is a powerful ensemble learning algorithm that
  - builds multiple decision trees and merges them to make more accurate and stable predictions.
  - It’s like a collection of individual trees (hence the "forest"), and each tree gets a say in the final decision.
  - It’s mainly used for **classification** and **regression** problems.

How Random Forest Works for Classification Problems

Here's a step-by-step breakdown:

1. **Bootstrap Sampling**: Random Forest creates different datasets by randomly selecting samples (with replacement) from the training data.
2. **Decision Tree Training**: For each bootstrap sample, a decision tree is built. However, during training, each split in the tree only considers a random subset of features.
3. **Majority Voting**: In classification, each tree in the forest makes its prediction, and the majority vote among the trees is taken as the final prediction.

Key Hyperparameters for Random Forest

- **n_estimators**: The number of trees in the forest. More trees usually lead to better performance but increase computation time.
- **max_depth**: The maximum depth of each tree. Deep trees can lead to overfitting, so you may want to control the depth.
- **max_features**: The number of features to consider when looking for the best split. You can set this to "auto" or "sqrt" for classification problems.
- **min_samples_split**: The minimum number of samples required to split a node.
- **class_weight**: Useful for handling imbalanced data by assigning more weight to the minority class.

Advantages and Limitations of Random Forest

Advantages:

- **Reduces Overfitting**: By averaging the results of multiple trees, Random Forest reduces overfitting that a single decision tree might face.
- **Handles Missing Data**: Random Forest can handle missing data fairly well by using averages from other trees.
- **Feature Importance**: It automatically computes feature importance, showing which features are most influential.

Limitations:

- **Slow for Large Datasets**: As the number of trees grows, training and prediction times increase.
- **Memory-Intensive**: Building many trees can consume a lot of memory.
- **Less Interpretable**: Compared to a single decision tree, Random Forest models are harder to interpret.

Code in Python

ML Interview Questions and Answers

https://medium.com/@vikashsinghy2k/top-interview-questions-and-answers-on-decision-trees-every-aspiring-data-scientist-should-know-1c40ffde6fc6

https://medium.com/@vikashsinghy2k/top-10-random-forest-interview-questions-and-answers-for-data-science-aspirants-9d2bfc688683

https://medium.com/@vikashsinghy2k/top-interview-questions-and-answers-on-bagging-algorithms-every-data-scientist-should-know-5bc65f637d91

https://medium.com/@vikashsinghy2k/frequently-asked-interview-questions-and-answers-on-linear-regression-d0e2e2339f58

https://medium.com/@vikashsinghy2k/top-time-series-forecasting-interview-questions-and-answers-to-master-your-data-science-skills-4363b940e8f1

Thank You!