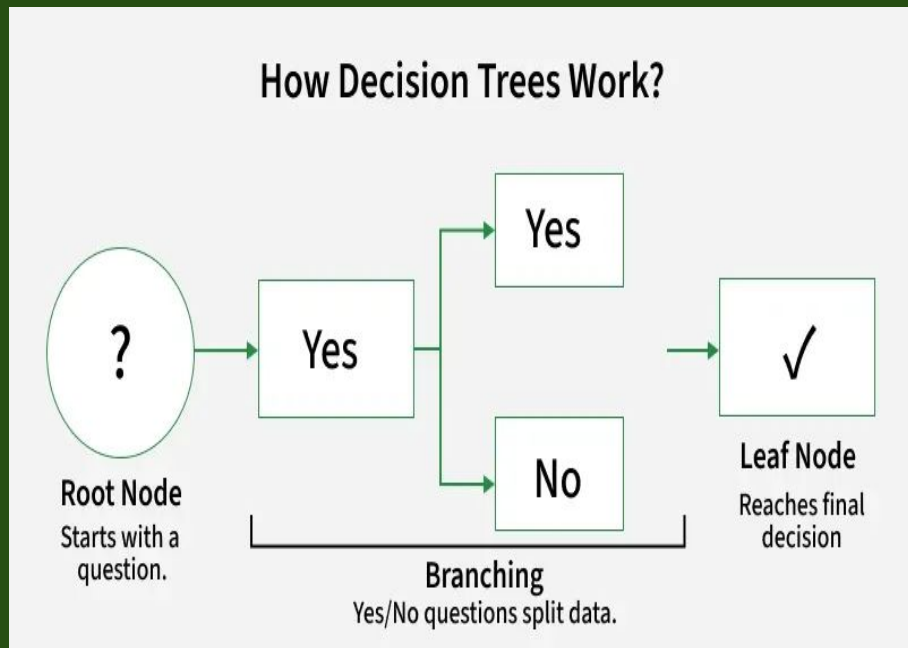


Random Forest Classification



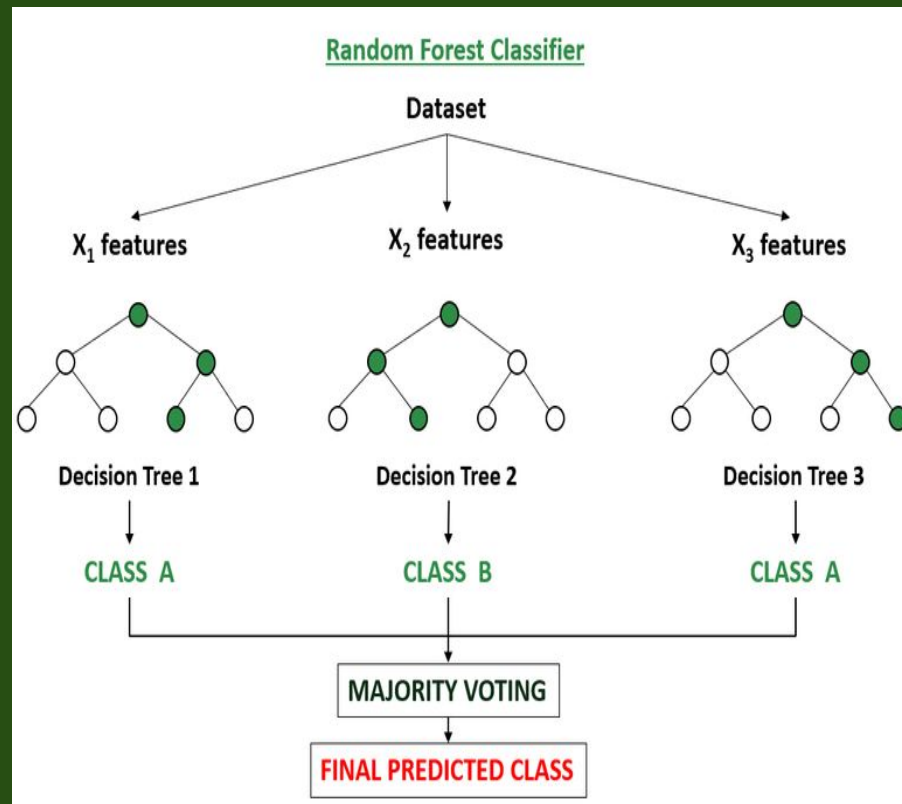
How do decision trees work?

- A Decision Tree is a flowchart-like model used to make predictions or decisions.
- Each node asks a question about a feature (e.g., "Is age > 30?").
- Each branch represents an answer (e.g., "Yes" or "No").
- Each leaf node represents a final decision or predicted class.
- It works by recursively splitting data into smaller, more uniform groups based on features.
- Goal: find splits that best separate the classes (minimize impurity, e.g., Gini index or entropy).



What is the Random Forest Algorithm in ML?

- Random Forest is a machine learning algorithm that uses many decision trees to make better predictions.
- Each tree looks at different random parts of the data and their results are combined by voting for classification or averaging for regression which makes it as ensemble learning technique.
- This helps in improving accuracy and reducing errors.



When to employ the Random Forest Algorithm

- When you want high accuracy with minimal parameter tuning.
- When your data has many features (both numerical and categorical).
- When you want to handle missing values and outliers gracefully.
- When working with a large datasets with many features without slowing down or losing accuracy
- Classification (spam detection, disease prediction, credit scoring)
- Regression (predicting prices, demand, etc.)

Limitations of the Random Forest Algorithm

- Can be slow when you have a lot of data or too many trees.
- Hard to understand exactly how it makes decisions — it's made up of many trees working together.
- Takes up more memory because it stores many trees at once.
- Can still overfit if the trees are too deep or if you use too many features.
- Not great for real-time predictions (like phone apps that need instant answers).

How to build a Random Forest Model

- Start with your training data — the examples your model will learn from.
- Use a method called bagging (short for Bootstrap Aggregating):
- Take random samples of your data with replacement (some rows may repeat).
- Build a different decision tree on each random sample.
- The tree only looks at a random subset of features at each split (not all of them).
- To get the final answer the model either looks at the majority vote (for classification) or average (for regression) is the final

The Process of Bagging (Bootstrap Aggregation)

