Tree Family of Algorithms

Agenda

- Decision Tree
- Random Forest
- Bagging Classifier
- Python Code
- Frequently asked Interview Question & Answers

Decision Tree Classifier

Decision Tree

1 Which of the following best describes a decision tree in machine learning?

A) A neural network-based algorithm used primarily for image recognition tasks.

B) An algorithm that clusters data into different groups based on similarity.

C) A machine learning algorithm used for both classification and regression tasks, which splits a dataset into smaller subsets.

D) A linear model used for time-series forecasting.

2 Decision tree is a non parametric ML algorithm?

A) True

B) False

Decision Tree

1 Which of the following best describes a decision tree in machine learning?

A) A neural network-based algorithm used primarily for image recognition tasks.

B) An algorithm that clusters data into different groups based on similarity.

C) A machine learning algorithm used for both classification and regression tasks, which splits a dataset into smaller subsets.

D) A linear model used for time-series forecasting.

2 Decision tree is a non parametric ML algorithm?

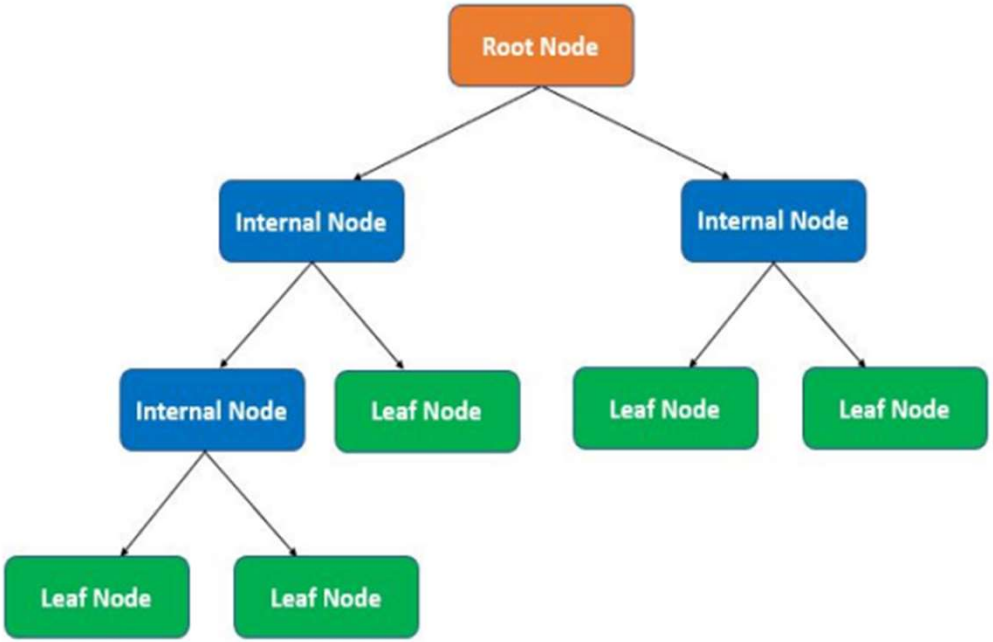
A) **True**

B) False

This means that decision trees have no assumptions about the space distribution and the structure of the classifier.

Building the tree

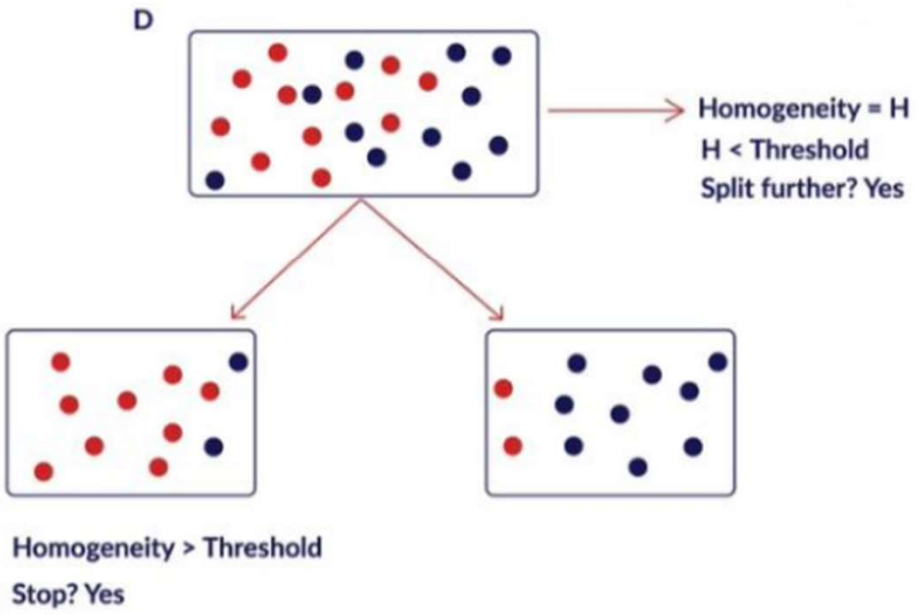
- Recursive Process: Starts with the root and recursively splits data based on features.
- Stopping Criteria: Stops when a certain depth or impurity threshold is reached.



Splitting and Homogeneity

Splitting in a decision tree involves dividing the dataset based on an attribute.

The goal is to achieve homogeneity in each node, meaning the samples in each node should be as similar as possible.



How to measure homogeneity?

The classification error is calculated as follows:

- $E = 1 - max(p_i)$

The Gini index is calculated as follows:

- $G = \sum_{i=1}^{k} p_i(1 - p_i)$

Entropy is calculated as follows:

- $D = -\sum_{i=1}^{k} p_i . log_2(p_i)$,

where p_i is the probability of finding a point with the label i, and k is the number of classes.

Demo in Python

https://medium.com/@vikashsinghy2k/complete-guide-to-decision-tree-classification-in-python-with-code-examples-6f69f408f880

Random Forest

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.

Bagging Classifier

What is Bagging Classifier?

Bagging Classifier (short for Bootstrap Aggregating) is another ensemble learning technique that, like Random Forest, creates multiple versions of a model and averages their results to reduce variance and improve stability.

The difference is that **Bagging** can use any base model (not just decision trees).

How Bagging Works for Classification Problems

The process for Bagging is quite similar to Random Forest, but here's how it works step by step:

1. **Bootstrap Sampling**: Like Random Forest, Bagging creates multiple training sets by sampling with replacement from the original dataset.
2. **Train Multiple Models**: Each dataset is used to train a base model (usually a decision tree, but it could be any model).
3. **Aggregation**: For classification, the Bagging Classifier aggregates the predictions from each model by majority vote.

Key Hyperparameters for Bagging Classifier

- **n_estimators**: The number of base models (trees) in the ensemble.
- **max_samples**: The proportion of the dataset to be sampled for each base model. It can be set to a percentage of the data.
- **base_estimator**: The base model to use. By default, this is a decision tree, but it can be any model.
- **bootstrap**: Whether to sample with replacement (True) or without replacement (False).
- **n_jobs**: The number of cores to use for parallel processing. This can speed up training significantly!

Advantages and Limitations of Bagging Classifier

Advantages:

- **Reduces Variance**: Bagging reduces variance and prevents overfitting by combining multiple models.
- **More Stable**: It produces more stable and robust models than a single base estimator.
- **Flexible**: You can use Bagging with various base models, giving it more flexibility.

Limitations:

- **Computationally Expensive**: Since Bagging requires training multiple models, it can be slow and resource-intensive.
- **Not Always Better**: If the base model is already performing well, Bagging may not always significantly improve the results.
- **Less Interpretability**: Like Random Forest, the resulting model is harder to interpret compared to a single decision tree.

Mathematical Formulas for Random Forest and Bagging

Gini Impurity: Measures how often a randomly chosen element would be incorrectly classified.

$$\text{Gini Impurity} = 1 - \sum_{i=1}^{C} p_i^2$$

Where p_i is the probability of class i.

Entropy: Measures the uncertainty of a dataset.

$$\text{Entropy} = -\sum_{i=1}^{C} p_i \log_2(p_i)$$

In both algorithms, trees are built by selecting splits that minimize these values, ensuring more "pure" groups at each node.

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!