Transcript: Decision Trees for Classification

Introduction

1. What decision trees are and how they work.
2. Why Use Decision Trees for classification.
3. Demonstration of building a decision tree step by step in Python using Iris dataset.

What is a Decision Tree?

- A Decision Tree in Machine Learning is a supervised learning algorithm used for classification and regression.

Key components:

1. **Nodes:** Represented by a question or a decision.
2. **Branches:** Possible outcomes of the decision.
3. **Root Node:** This node represents the whole dataset and the first split.
4. **Leaf Nodes:** Terminal nodes that represent the final output by predicted class or value.

Example:

Suppose you're deciding whether to bring an umbrella. You check the weather:

- If it’s raining, take the umbrella.
- If it’s cloudy, check the forecast.
- If the forecast predicts rain, take it; otherwise, don’t.

We can visualize this decision process as a decision tree with conditions and outcomes.

Why Use Decision Trees?

Decision trees are popular for several reasons some of them are:

- **Ease of interpretation:** The model processes as a human decision-making.
- **Versatility:** Handles both the numerical and categorical data effectively.
- **No scaling required:** The preprocessing of data is simpler.

However, Decision Trees can still overfit when dealing with noisy data or have difficulty with complex relationships unless they're pruned or improved.

How Do Decision Trees Make Splits?

We can use some metrics to split the data by evaluating how "pure" each resulting subset becomes.
Two key metrics for this are:

1. **Gini Impurity:** It measures the probability of misclassification.

   Formula: Gini = $1 - \Sigma(p_i^2)$o

   Example: In a dataset with 40% apples and 60% oranges:

   Gini = $1 - (0.4^2 + 0.6^2) = 0.48$

   Lower Gini value means a better split.

2. **Information Gain:** This metric measures how much a split has improved the classification.

   Formula: Info Gain = Entropy(Parent) - Weighted Avg[Entropy(Children)]

   - It provides splits that results in purer and cleaner subsets.
   - We can use criterion='entropy' in scikit-learn to trains the model using Information Gain.

These metrics ensure the tree focuses on the features that are most important.

Building a Decision Tree in Python

Let’s implement a Decision Tree using Python and the Iris dataset.

Jupyter Notebook demonstration

Visualizing Decision Trees

Visualization is a huge advantage of Decision Trees it makes it so much easier to analyze and understand. We can use the plot_tree() function to see:

- How the model splits the data.
- The thresholds for each split.

Limitations and Overfitting

Decision Trees have some drawbacks too:

- They tend to **overfit** on small or noisy datasets.
- Sensitive to small changes in data.
- It require techniques like **pruning** or using ensemble methods like Random Forests for better performance.

Applications of Decision Trees

Decision Trees are widely used across industries some of the industries where it can be used are:

- **Healthcare:** Diagnosing diseases based on symptoms common is heart attack.
- **E-commerce:** Recommending products to users for customer retention and satisfaction.
- **Finance:** Approving or rejecting loan applications.