Module 9: Decision Trees: Part One

Learning outcomes

1. Analyse key metrics and concepts used to guide decision tree construction.
2. Determine effective splitting strategies and stopping criteria for decision trees.
3. Implement decision tree algorithms in Python for classification and prediction.
4. Identify pruning techniques that improve decision tree generalisation.
5. Apply decision tree methods to real-world problems.

Decision tree

- A decision tree is a hierarchical model that recursively splits data into subsets to make predictions.
- Root node is the topmost node where the initial split occurs.
- Leaf node is the terminal node representing a final prediction.
- Decision node is a node that splits the data into branches.
- Split is a decision rule based on a feature and threshold that divides data into groups.

Splitting criteria

- **Impurity:** a measure of how mixed a node is (i.e. how diverse the class labels are)
- **Entropy:** measures unpredictability: $\text{entropy} = -\sum \text{p(x)} \log_2 \text{p(x)}$
- **Gini index:** probability of misclassification when randomly assigning labels: $\text{Gini} = 1 - \sum \text{p(x)}^2$
- **Information gain:** reduction in impurity after a split: $\text{information gain} = \text{entropy(parent)} - \sum \frac{\text{n}_{\text{child}}}{\text{n}_{\text{tolal}}} \times \text{entropy(child)}$

Types of splits

- **Categorical features**
  - One node per category
  - Binary split: category vs others
  - Subset splits (e.g. {low, medium} vs {high})
- **Numerical features**
  - Binary splits at midpoints between consecutive values

Pruning methods

- **Pre-pruning** stops tree growth early using parameters such as maximum depth or minimum samples.

- **Post-pruning** trims a fully grown tree using validation data or cost-complexity pruning.
- **Common stopping criteria**
  - Maximum depth
  - Minimum samples per node or split
  - Minimum information gain

Predictions

- **Classification trees** predict using the majority class in the leaf node.
- **Regression trees** predict using the mean of target values in the leaf node.

Trade-offs

- **Overfitting:** the tree becomes too specific to training data, capturing noise.
- **Underfitting:** the tree is too shallow, missing underlying patterns.
- **Bias–variance trade-off:** pruning helps balance model complexity with generalisation ability.

Misclassification vs Gini vs entropy

Comparing the impurity measures

Criterion	Sensitivity	Common use
Entropy	High	Slower but more sensitive
Gini index	Moderate	Faster and widely used
Misclassification	Low	Too crude for splitting decisions

Tips for practice

- Use Gini index or entropy to evaluate splits.
- Apply pruning to prevent overfitting and simplify the model.
- Track model performance using metrics such as accuracy, precision, recall and the misclassification rate, not just training accuracy.