Assignment Code: DA-AG-012

Decision Tree | **Assignment**

Instructions: Carefully read each question. Use Google Docs, Microsoft Word, or a similar tool to create a document where you type out each question along with its answer. Save the document as a PDF, and then upload it to the LMS. Please do not zip or archive the files before uploading them. Each question carries 20 marks.

Total Marks: 100

Question 1: What is a Decision Tree, and how does it work in the context of classification?

Answer:

A **Decision Tree** is a popular supervised machine learning algorithm used for both classification and regression problems. In classification tasks, it helps in predicting the class label of input data by learning simple decision rules from the features of the data.

It is structured like a tree, where:

- The **root node** represents the entire dataset, which gets split into two or more homogeneous sets based on a specific feature.
- **Internal nodes** represent decisions based on feature values.
- **Branches** represent the outcomes of those decisions.
- **Leaf nodes** represent the final class label or prediction.

The decision tree works by recursively partitioning the dataset using algorithms such as **ID3**, **CART**, or **C4.5**, which use metrics like:

- **Information Gain** (based on Entropy),
- **Gini Index**, or

- **Gain Ratio**

These metrics measure how well a feature separates the data into target classes.

For example, in a binary classification problem (e.g., "Yes" or "No"), the decision tree evaluates feature values (like age, income, etc.) step by step, choosing splits that best reduce impurity at each level. This continues until the tree reaches a stopping criterion (e.g., maximum depth, minimum samples per leaf, or perfect classification).

Advantages in Classification:

- **Easy to visualize and interpret**
- **No need for feature scaling**
- **Handles both numerical and categorical data**

Question 2: Explain the concepts of Gini Impurity and Entropy as impurity measures. How do they impact the splits in a Decision Tree?

Answer:

Gini Impurity and Entropy are two commonly used impurity measures in Decision Trees that help determine the best way to split data at each node. Their goal is to increase the "purity" of nodes — meaning how homogeneous the classes are after a split.

---

1. Gini Impurity:

- **Definition:** Gini Impurity calculates the probability of incorrectly classifying a randomly chosen element if it were randomly labeled according to the class distribution.
- **Formula:**
  $\text{Gini} = 1 - \Sigma(p_i^2)$
  where p_i is the probability of class *i* in the node.

- **Range:** 0 (pure node) to 0.5 (most impure for binary classification).
- **Properties:** Gini tends to isolate the most frequent class and is faster to compute. It is used in algorithms like CART (Classification and Regression Trees).

---

2. Entropy and Information Gain:

- **Definition:** Entropy is a measure from information theory that quantifies the disorder or uncertainty in the dataset. In Decision Trees, we use Information Gain to decide the best split.
- **Formula for Entropy:**
  $\text{Entropy} = -\Sigma(p_i \times \log_2(p_i))$
- **Information Gain (IG):**
  IG = Entropy(Parent) − Weighted average Entropy(Children)
- **Range:** 0 (pure) to 1 (maximally impure for binary classification).
- **Properties:** Entropy is more theoretical and tends to build more balanced trees, but it is computationally heavier than Gini.

---

3. Impact on Decision Tree Splits:

- Both measures are used to evaluate how good a split is.
- At each node, the Decision Tree algorithm checks all possible features and thresholds, and chooses the one that results in:
  - The **lowest Gini Impurity**, or
  - The **highest Information Gain**.
- The selected split helps the model create purer child nodes, leading to better classification accuracy.

Question 3: What is the difference between Pre-Pruning and Post-Pruning in Decision Trees? Give one practical advantage of using each.

Answer:

In Decision Trees, pruning is a technique used to reduce the size of the tree and prevent overfitting. There are two types of pruning: Pre-Pruning and Post-Pruning.

1. Pre-Pruning (Early Stopping):

- **Definition:** Pre-pruning stops the tree from growing once a certain condition is met, even before it becomes fully grown.
- **Conditions for Stopping:**
    - Maximum depth of tree reached
    - Minimum number of samples at a node
    - Gini or Entropy does not improve significantly
- **Practical Advantage:**
  ➤ *Saves time and memory during training by preventing the creation of an overly complex tree.*

2. Post-Pruning (Cost Complexity Pruning):

- **Definition:** Post-pruning allows the tree to grow fully and then removes unnecessary branches that do not contribute much to accuracy.
- **How it works:**
    - The complete tree is built
    - Cross-validation or a validation set is used to prune the branches that reduce performance
- **Practical Advantage:**
  ➤ *Improves model's ability to generalize by removing overfitted sections of the tree after it's fully built.*

3. Key Differences between Pre-Pruning and Post-Pruning:

- **Timing:**
    - ***Pre-Pruning:*** Stops the tree from growing further during training.
    - ***Post-Pruning:*** Grows the full tree first, then removes unnecessary branches.
- **Control:**
    - ***Pre-Pruning:*** Applies stopping rules like max depth or minimum samples.
    - ***Post-Pruning:*** Uses a validation set or cross-validation to decide which branches to prune.
- **Speed:**
    - ***Pre-Pruning:*** Faster because it builds a smaller tree upfront.
    - ***Post-Pruning:*** Slower since it grows the entire tree before pruning.
- **Accuracy:**
    - ***Pre-Pruning:*** Might stop too early and miss useful patterns.
    - ***Post-Pruning:*** More accurate because it evaluates complete patterns before trimming.
- **Overfitting Handling:**
    - ***Pre-Pruning:*** Prevents overfitting from the beginning.
    - ***Post-Pruning:*** Fixes overfitting after it occurs.

Question 4: What is Information Gain in Decision Trees, and why is it important for choosing the best split?

Answer:

Information Gain (IG) is a key concept used in Decision Trees to measure how well a particular feature separates the data into distinct classes. It is based on the concept of **Entropy**, which measures the impurity or disorder in a dataset.

---

1. What is Information Gain?

- **Definition:**
  Information Gain is the **reduction in entropy** after a dataset is split on a particular feature. It tells us how much "information" a feature gives us about the class label.

- **Formula:**
  Information Gain (IG) = Entropy (Parent) − Weighted Average Entropy (Children)

  That is:
  IG = Entropy(parent) − [(n_1 / N) × Entropy(child_1) + (n_2 / N) × Entropy(child_2) + ...]

  where:

  - n_1, n_2 = number of samples in child nodes
  - N = total number of samples in the parent node

2. Importance in Choosing the Best Split:

- At every node in the tree, the algorithm calculates Information Gain for each feature.
- The feature with the **highest Information Gain** is selected for the split.
- This leads to the **purest possible division** of data and helps the tree learn useful patterns.

3. Simple Example:

Suppose we are classifying whether a customer will buy a product.
If the feature `Age` gives more Information Gain than `Income`, then the tree will split based on `Age` — because it provides more useful information to classify the data.

Question 5: What are some common real-world applications of Decision Trees, and what are their main advantages and limitations?

Answer:

Decision Trees are widely used in various real-world applications because they are easy to understand, interpret, and implement. Below are some **popular applications**, along with their **advantages** and **limitations**.

1. Real-World Applications of Decision Trees:

- **Medical Diagnosis:**
  Used to predict diseases based on symptoms, test results, and patient history.

- **Credit Scoring:**
  Banks use decision trees to approve or reject loan applications based on income, credit score, and other factors.

- **Customer Churn Prediction:**
  Companies use decision trees to identify customers likely to stop using a service.

- **Fraud Detection:**
  Helps in identifying unusual transaction patterns that may indicate fraud.

- **Marketing and Targeting:**
  Used to segment customers and personalize advertisements based on behavior.

- **Risk Assessment:**
  Insurance companies use decision trees to evaluate risk and decide premiums.

2. Main Advantages of Decision Trees:

- **Easy to Understand and Visualize:**
  The tree structure is simple to explain and interpret.

- **No Need for Data Normalization or Scaling:**
  Works well with both numerical and categorical data.

- **Fast Training Speed:**
  Especially effective for small to medium-sized datasets.

- **Handles Non-Linear Relationships:**
  Decision Trees can model complex patterns in the data.

3. Main Limitations of Decision Trees:

- **Overfitting:**
  Trees can become too complex and fit the training data too closely, reducing accuracy on new data.

- **Unstable:**
  Small changes in the data can result in a completely different tree structure.

- **Biased with Imbalanced Data:**
  Decision Trees may favor features with more levels or classes.

- **Less Accurate Compared to Ensembles:**
  Alone, they may perform worse than models like Random Forests or Gradient Boosted Trees.

Dataset Info:

- **Iris Dataset** for classification tasks (sklearn.datasets.load_iris() or provided CSV).

- **Boston Housing Dataset** for regression tasks (sklearn.datasets.load_boston() or provided CSV).

Question 6: Write a Python program to:

- Load the Iris Dataset
- Train a Decision Tree Classifier using the Gini criterion
- Print the model’s accuracy and feature importances

(Include your Python code and output in the code box below.)

Answer: code

Question 7: Write a Python program to:

- Load the Iris Dataset
- Train a Decision Tree Classifier with max_depth=3 and compare its accuracy to a fully-grown tree.

(Include your Python code and output in the code box below.)

Answer: code

Question 8: Write a Python program to:

- Load the Boston Housing Dataset
- Train a Decision Tree Regressor
- Print the Mean Squared Error (MSE) and feature importances

Answer: code

Question 9: Write a Python program to:

- Load the Iris Dataset
- Tune the Decision Tree’s max_depth and min_samples_split using GridSearchCV
- Print the best parameters and the resulting model accuracy

Answer: code

Question 10: Imagine you're working as a data scientist for a healthcare company that wants to predict whether a patient has a certain disease. You have a large dataset with mixed data types and some missing values.

Explain the step-by-step process you would follow to:

- Handle the missing values
- Encode the categorical features
- Train a Decision Tree model
- Tune its hyperparameters
- Evaluate its performance And describe what business value this model could provide in the real-world setting.

Answer: code