Decision Tree

**Objective:**

The objective of this assignment is to apply Decision Tree Classification to a given dataset, analyse the performance of the model, and interpret the results.

**Tasks:**

1. Data Preparation:

Load the dataset into your preferred data analysis environment (e.g., Python with libraries like Pandas and NumPy).

**2. Exploratory Data Analysis (EDA):**

Perform exploratory data analysis to understand the structure of the dataset.

Check for missing values, outliers, and inconsistencies in the data.

Visualize the distribution of features, including histograms, box plots, and correlation matrices.

**3. Feature Engineering:**

If necessary, perform feature engineering techniques such as encoding categorical variables, scaling numerical features, or handling missing values.

**4. Decision Tree Classification:**

Split the dataset into training and testing sets (e.g., using an 80-20 split).

Implement a Decision Tree Classification model using a library like scikit-learn.

Train the model on the training set and evaluate its performance on the testing set using appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score, ROC-AUC).

**5. Hyperparameter Tuning:**

Perform hyperparameter tuning to optimize the Decision Tree model. Experiment with different hyperparameters such as maximum depth, minimum samples split, and criterion.

**6. Model Evaluation and Analysis:**

Analyse the performance of the Decision Tree model using the evaluation metrics obtained.

Visualize the decision tree structure to understand the rules learned by the model and identify important features

**Interview Questions:**

1. What are some common hyperparameters of decision tree models, and how do they affect the model's performance?

2. What is the difference between the Label encoding and One-hot encoding?

Answers:

<https://colab.research.google.com/drive/12wxrO9gQNbnYAagCqtOQnUhRgwb3H8SB?usp=sharing>

Ans 1) In the context of decision tree models, some common hyper-parameters include:

1. Maximum Depth: This hyper-parameter controls the maximum depth of the tree. A deeper tree can lead to overfitting, while a shallower tree may result in under-fitting.

2. Minimum Samples Split: This hyper-parameter determines the minimum number of samples required to split a node. A lower value can lead to overfitting, while a higher value can result in under-fitting.

3. Minimum Samples Leaf: This hyper-parameter sets the minimum number of samples required to be at a leaf node. Similar to minimum samples split, it helps control overfitting and under-fitting.

4. Maximum Features: This hyper-parameter specifies the maximum number of features to consider when looking for the best split. It can help prevent overfitting by limiting the number of features used.

These hyper-parameters play a crucial role in determining the complexity and generalization ability of a decision tree model. By tuning these hyper-parameters effectively, one can optimize the model's performance and prevent issues such as overfitting or under-fitting.

Ans: 2) Label encoding involves converting each category in a categorical variable into a numerical label. This is done by assigning a unique integer to each category, which can be useful for algorithms that require numerical inputs.

The key difference between label encoding and one-hot encoding lies in how they represent categorical variables numerically. Label encoding assigns unique integers to categories, while one-hot encoding creates binary columns for each category. The choice between the two techniques depends on the nature of the data and the requirements of the model being used.