Name: Jiya Sharma
Roll no: 281056
PRN: 22311741

Assignment 7

Problem Statement:

Every year many students take the GRE exam to get admission in foreign Universities. The data set contains GRE Scores (out of 340), TOEFL Scores (out of 120), University Rating (out of 5), Statement of Purpose strength (out of 5), Letter of Recommendation strength (out of 5), Undergraduate GPA (out of 10), Research Experience (0=no, 1=yes), Admitted (0=no, 1=yes). Admitted is the target variable.

Data Set: https://www.kaggle.com/mohansacharya/graduate-admissions

The counselor of the firm is supposed to check whether the student will get an admission or not based on his/her GRE score and Academic Score. So to help the counselor to make appropriate decisions build a machine learning model classifier using a Decision tree to predict whether a student will get admission or not.

a) Apply Data pre-processing (Label Encoding, Data Transformation….) techniques if necessary.
b) Perform data preparation (Train-Test Split)
c) Apply Machine Learning Algorithm
d) Evaluate Model.

Objective:

- To preprocess and clean the dataset for improved model performance.
- To split the dataset for effective model training and evaluation.
- To build and apply a Decision Tree classifier for admission prediction.
- To assess the model's accuracy and effectiveness.

Resources used:

- Software used: Visual Studio Code
- Libraries used: Pandas, Matplotlib, sklearn, Seaborn

Theory:

1. **Decision Tree Classifier:**

   A Decision Tree is a supervised learning algorithm used for classification and regression tasks. It splits the dataset into smaller subsets using conditions based on feature values, forming a tree-like structure. The main components of a Decision Tree are:

- **Root Node:** Represents the entire dataset and selects the best feature for splitting.
- **Internal Nodes:** Represent decision points based on attribute values.
- **Leaf Nodes:** Represent final classification labels (Admission = Yes/No).

The model works by recursively splitting the data based on features that result in the highest information gain or lowest Gini impurity.

2. **Summary Statistics:**

   Summary statistics provide key insights into the dataset, including:

- **Minimum & Maximum Values:** Identify data range.
- **Mean & Median:** Represent central tendency.
- **Standard Deviation & Variance:** Measure data dispersion.
- **Percentiles:** Show data distribution.

3. **Data Preprocessing:**

   Data preprocessing ensures the dataset is clean and ready for analysis. Steps include:

- Handling missing values (e.g., imputation, removal).
- Encoding categorical variables (e.g., Label Encoding, One-Hot Encoding).
- Normalization and scaling for numerical stability.

4. **Model Evaluation Metrics:**

   To assess model performance, we use:

- **Accuracy:** Measures correct predictions over total instances.
- **Precision & Recall:** Evaluate positive class performance.
- **F1-score:** Balances precision and recall.
- **Confusion Matrix:** Displays true/false positives and negatives.

Methodology:

1. **Data Preprocessing:**

- Handle missing values, if any, using imputation techniques.
- Perform feature scaling or normalization, if required.
- Encode categorical features if necessary

2. **Data Preparation:**

- Split the dataset into training (80%) and testing (20%) subsets using Scikit-Learn's train_test_split() function.

3. **Model Implementation:**

- Implement a Decision Tree Classifier using Scikit-Learn's DecisionTreeClassifier.
- Train the model using the training dataset.
- Predict admission outcomes using the test dataset.

4. **Model Evaluation:**

- Evaluate model performance using metrics such as Accuracy, Precision, Recall, and F1-score.
- Use a Confusion Matrix to analyse model predictions.

Conclusion:

- Preprocessing ensures that the dataset is clean and suitable for modelling.
- A Decision Tree classifier can effectively classify students based on their academic scores.
- Evaluating performance metrics helps in understanding model accuracy and effectiveness.