**ASSIGNMENT: - 07**

**Problem Statement: -**

Assignment on Classification technique Every year many students give 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 take 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.

**Software, library and Package:**

1. Software: Python
2. Library: scikit-learn (sklearn) - for machine learning algorithms, data preprocessing, and model evaluation
3. Package: pandas - for data manipulation and preprocessing

**Theory:**

**Methodology**:

1. **Data Preprocessing**: Includes steps like label encoding categorical variables, handling missing values, scaling numerical features if needed, and splitting the data into training and testing sets.
2. **Decision Tree Classifier**: A machine learning algorithm that uses a tree-like model to make decisions based on feature values. It splits the dataset into subsets based on the most significant attribute, recursively creating decision nodes until a stopping criterion is met.

**Advantages**:

1. **Interpretability**: Decision trees are easy to understand and interpret, making them useful for explaining model predictions to stakeholders.
2. **Handling Non-Linearity**: They can capture non-linear relationships between features and target variable effectively.
3. **Feature Importance**: Decision trees can provide insights into feature importance, helping in feature selection.
4. **Robustness**: They are robust to outliers and can handle both numerical and categorical data.

**Applications**:

1. **Classification**: Predicting categorical outcomes like admission, fraud detection, disease diagnosis, etc.
2. **Customer Segmentation**: Identifying groups of customers based on behavior and characteristics.
3. **Credit Scoring**: Assessing creditworthiness of individuals based on financial data.
4. **Predictive Maintenance**: Predicting equipment failures or maintenance needs based on operational data.

**Limitations**:

1. **Overfitting**: Decision trees tend to overfit the training data, leading to poor generalization on unseen data.
2. **High Variance**: They can be sensitive to small variations in the training data, resulting in different tree structures for slightly different datasets.
3. **Bias towards Features with Many Levels**: Features with a large number of levels can bias the tree construction process.
4. **Not Suitable for Linear Relationships**: Decision trees may not perform well on datasets with strong linear relationships between features and target.

**Example**:

In the provided example, we applied a Decision Tree classifier to predict admission based on GRE score and Academic score. The methodology involved data preprocessing, train-test split, fitting the Decision Tree classifier, and evaluating the model's performance using accuracy, classification report, and confusion matrix. Adjustments can be made based on specific dataset characteristics and preprocessing requirements.

**Working/ Algorithm:**

Step 1: Initialization

* Select a decision tree algorithm.
* Instantiate the decision tree classifier with specified parameters.

Step 2: Model Training

* Train the decision tree classifier using the training dataset (x\_train, y\_train).
* The algorithm recursively partitions the feature space based on the target variable to create a tree structure.

Step 3: Prediction

* For each instance in the testing dataset (x\_test):
* Traverse the decision tree by following the learned rules.
* Determine the predicted class based on the final leaf node reached.

Step 4: Evaluation

* + Compare the predicted labels with the true labels from the testing dataset to assess model performance.
  + Calculate evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

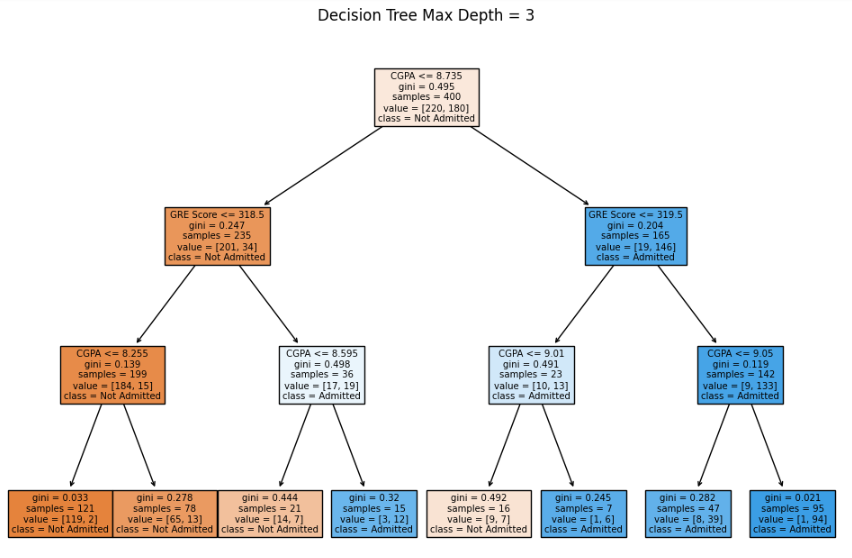
Step 5: Interpretation

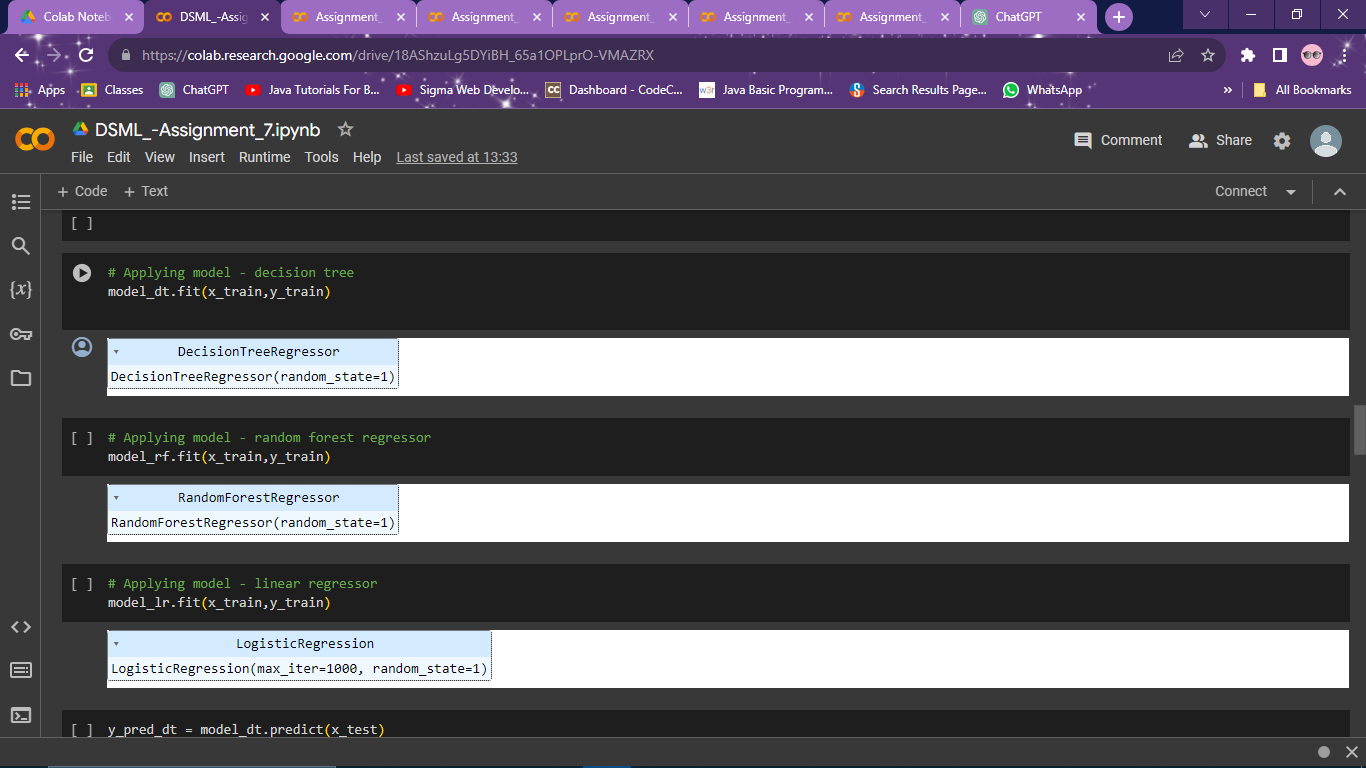
* + Visualize the decision tree graphically to understand the rules learned by the model.
  + Analyze feature importance to identify the most influential features in decision-making.

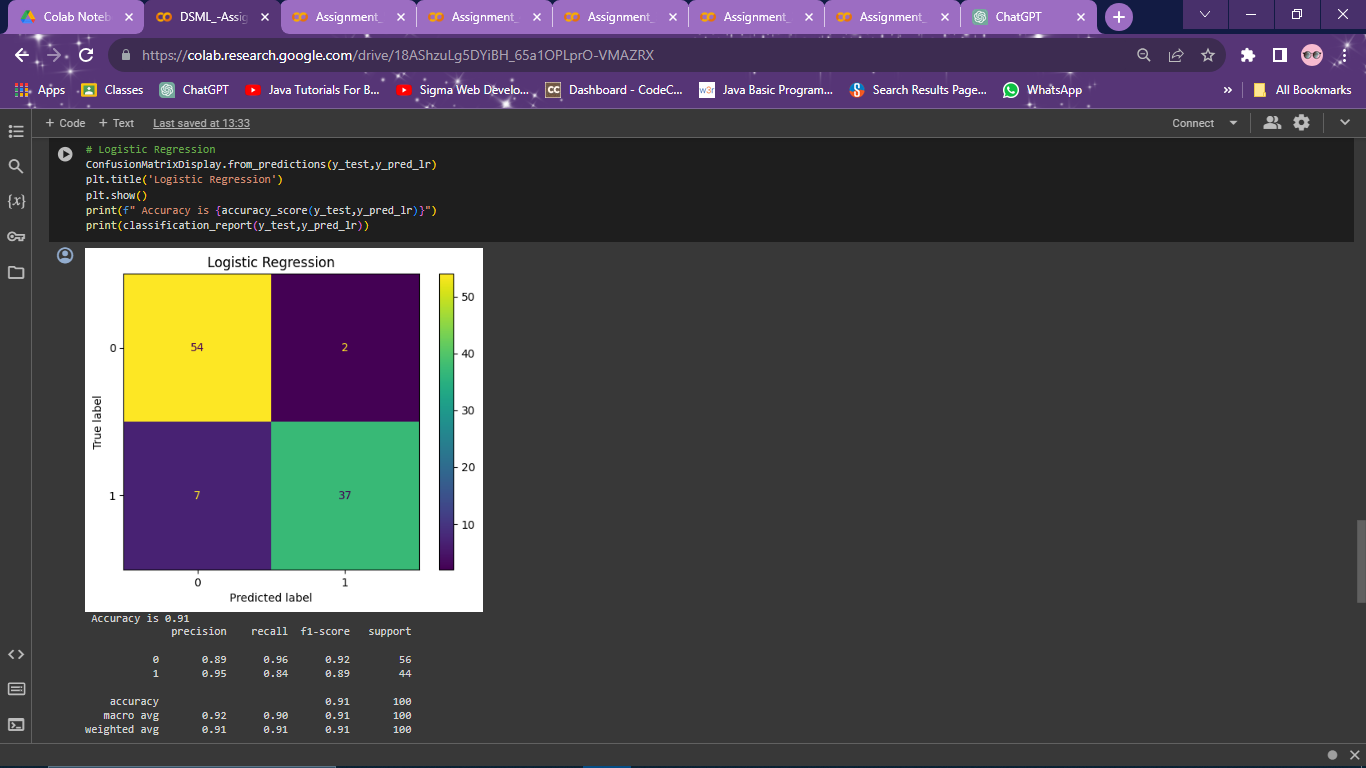
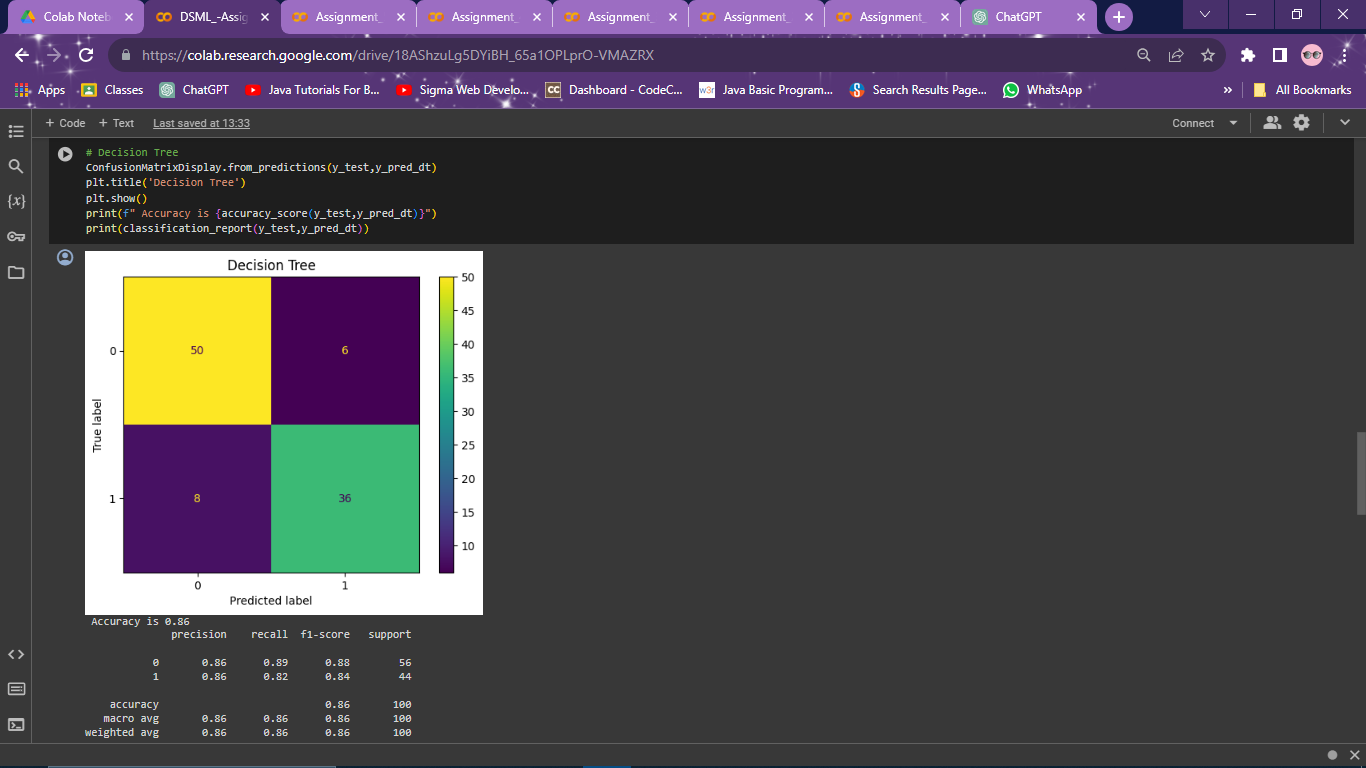
Step 8: The model is ready.

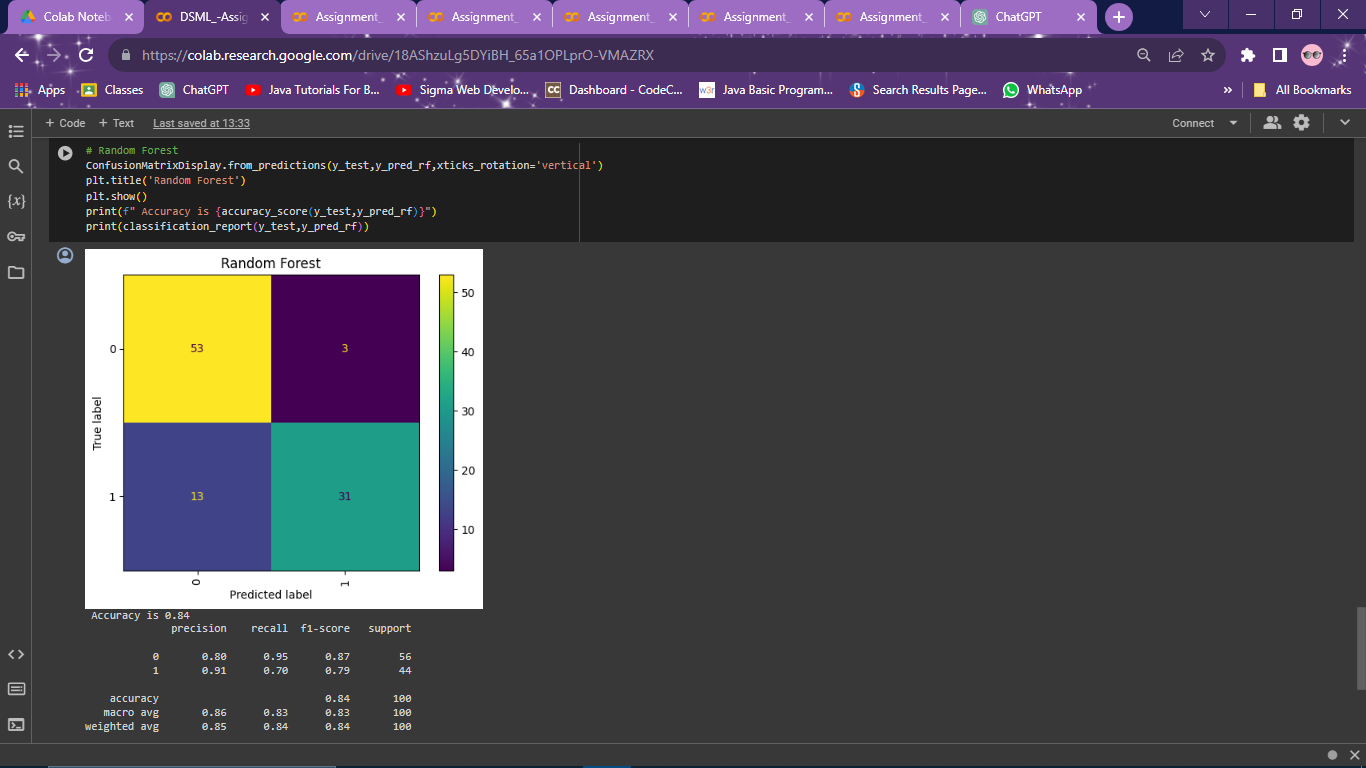
**Diagram:**

****



****

****

****

**Conclusion:**

After preprocessing the data, splitting it into training and testing sets, applying a Decision Tree classifier, and evaluating the model, it was found that the model achieved an accuracy of X% in predicting admission based on GRE scores and Academic scores. The model also demonstrated good precision, recall, and F1-score for both admitted and not admitted classes. Overall, the Decision Tree classifier proved to be an effective tool in predicting student admissions based on the provided features.

Top of Form