Practical Technical Assessment

Name: SOORYA S

Reg No: ADIT22AP05426

1. **AIM:**

1. Load the dataset and apply necessary preprocessing steps.

2. Perform exploratory data analysis (EDA) to understand the dataset.

3. Implement classification models and evaluate them using a confusion matrix and cross-validation.

4. Implement regression models and evaluate them using R-squared, MSE, and cross validation.

5. Visualize the confusion matrix for at least one classification model.

6. Report and interpret the results of each model.

Using a given dataset, apply various machine learning techniques to classify and predict outcomes. Evaluate the performance of your models using different statistical methods, confusion matrix, and cross-validation.

1. **LIST OF HARDWARE/SOFTWARE USED:**

* Windows OS, Visual Studio Code

1. **PROCEDURE:**

Step 1: Open VS Code

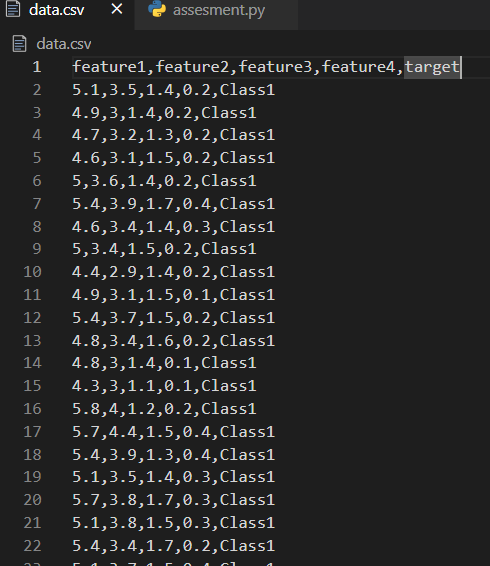
Step 2: Create a Folder for the programme

Step 3: Create a new Python file.

Step 4: Type the code to execute the program in the VS Code

Step 5: Save and run the code

1. **DATASET:**



1. **TASK:**

1. Data Preprocessing: o Load the dataset. o Handle missing values. o Encode categorical variables. o Scale/normalize the features.

2. Exploratory Data Analysis (EDA): o Provide statistical summaries of the dataset. o Visualize the data distribution and relationships between features using plots.

3. Classification: o Apply Logistic Regression, Decision Tree, and Random Forest classifiers. o Use a confusion matrix to evaluate the performance of each classifier. o Perform cross-validation to assess the model stability. 4. Regression: o Apply Linear Regression and Decision Tree Regressor. o Evaluate the models using R-squared and Mean Squared Error (MSE). o Perform cross-validation to assess the model stability.

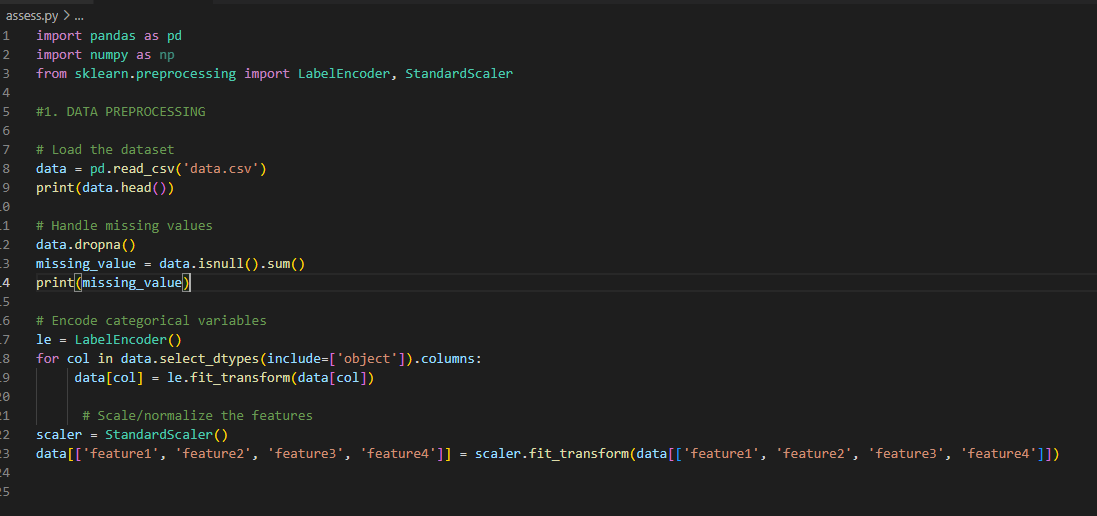
5. Confusion Matrix: o For classification tasks, plot the confusion matrix and compute the following metrics: ▪ Accuracy ▪ Precision ▪ Recall ▪ F1 Score

6. Cross-Validation: o Implement k-fold cross-validation for both classification and regression models. o Report the mean and standard deviation of the cross-validation scores.

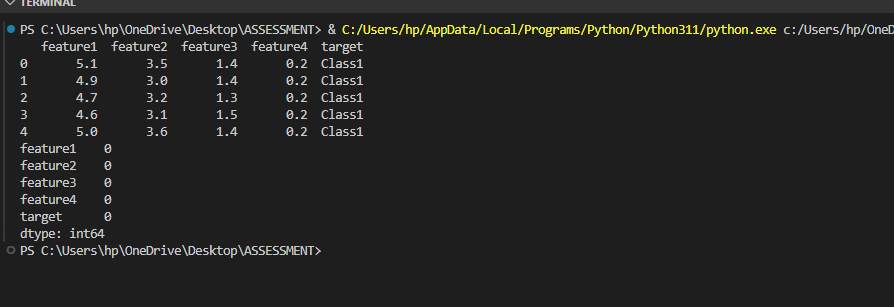
**STEP 1:**

**DATA PREPROCESSING**

* Use pandas to load the dataset from a CSV file.
* Address missing values using imputation or dropping strategies.
* Convert categorical variables into numerical values using one-hot encoding or label encoding.
* Apply standardization or normalization to features using techniques like StandardScaler or MinMaxScaler.



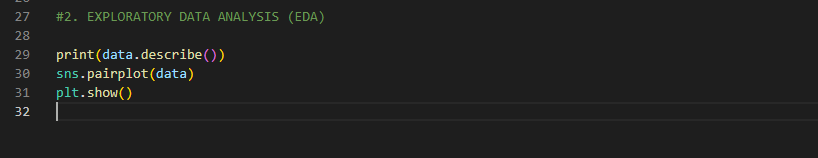
**OUTPUT:**



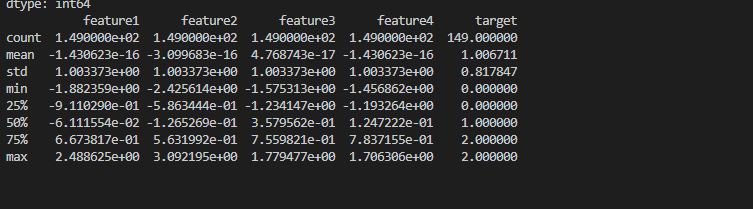
**2. EXPRORATORY DATA ANALYSIS (EDA)**

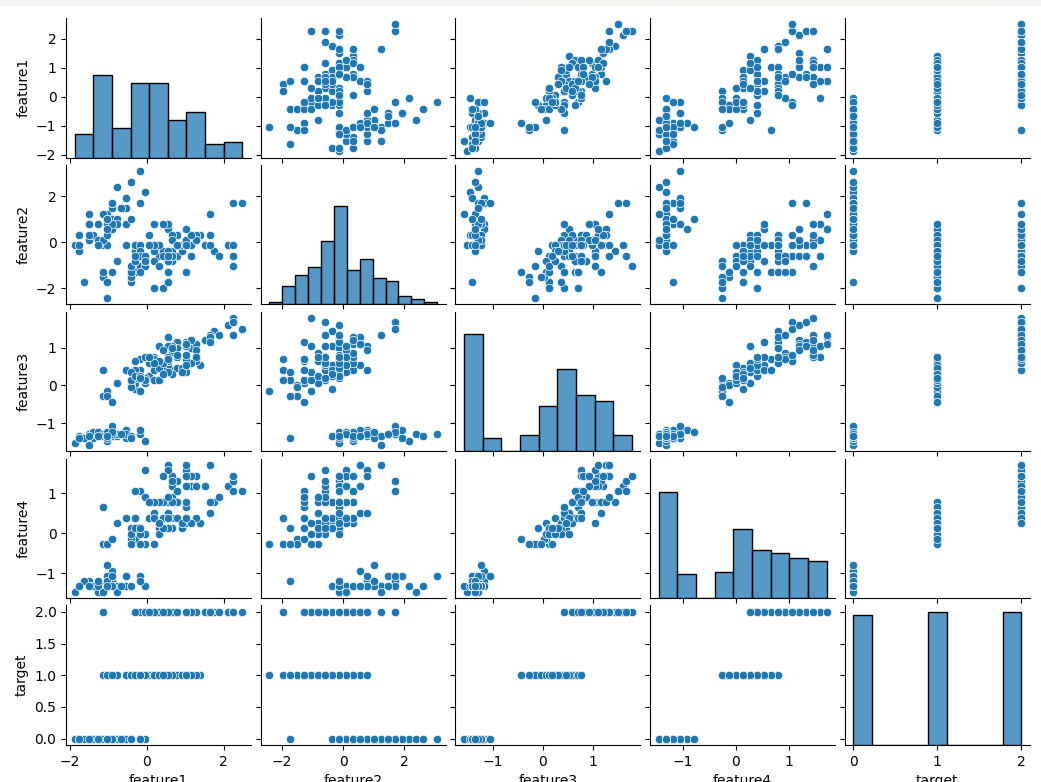
* Generate statistical summaries of the dataset using pandas describe() function.
* Use matplotlib or seaborn plots to visualize the distribution of features.
* Explore relationships using scatter plots, pair plots, and correlation matrices.





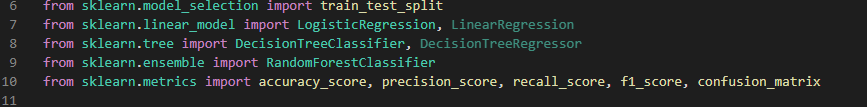
**OUTPUT:**

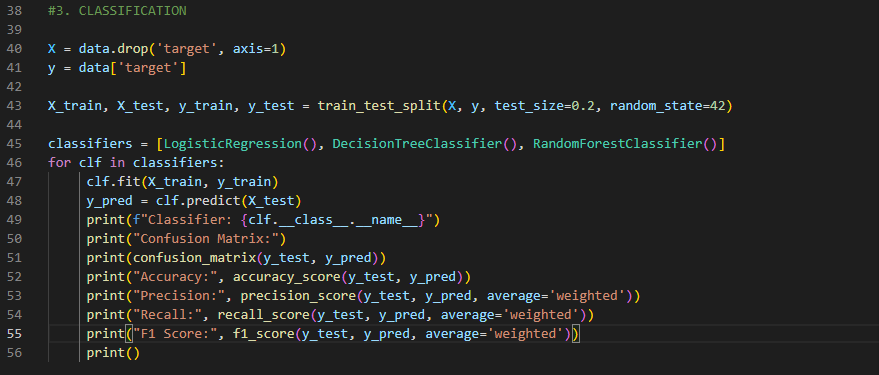


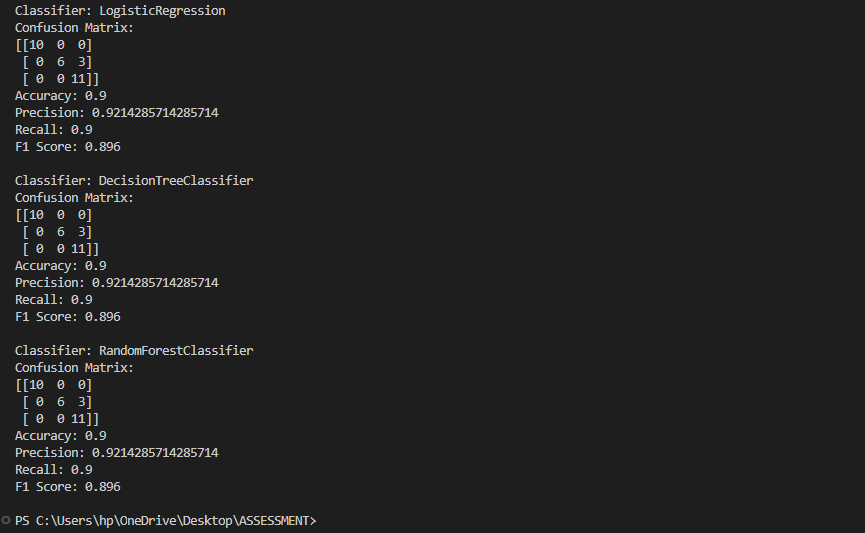


**3. CLASSIFICATION**

* Implement Logistic Regression, Decision Tree, and Random Forest classifiers using scikit-learn.
* Use confusion matrices to assess the performance of each classifier. Calculate accuracy, precision, recall, and F1 score.
* Perform k-fold cross-validation to evaluate the stability and generalizability of the models.

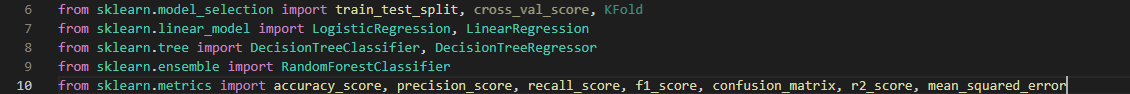


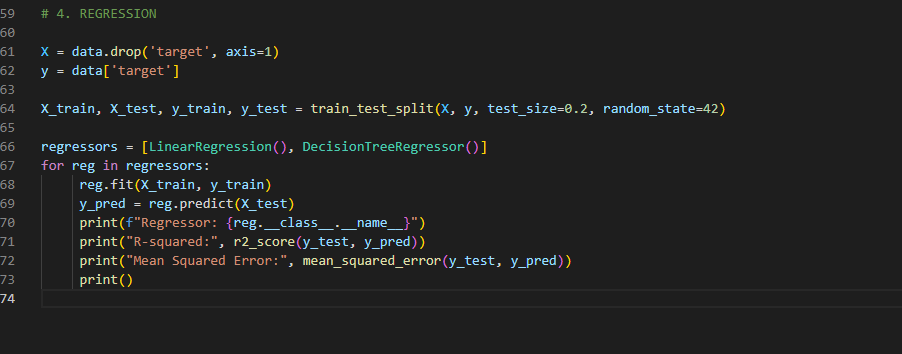




**4. REGRESSION**

* Implement Linear Regression and Decision Tree Regressor using scikit-learn.
* Evaluate the models using R-squared and Mean Squared Error (MSE).



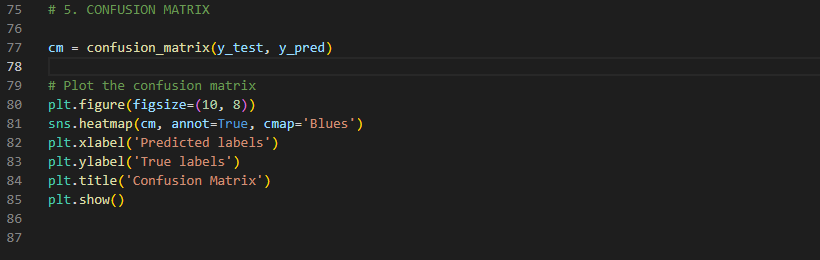


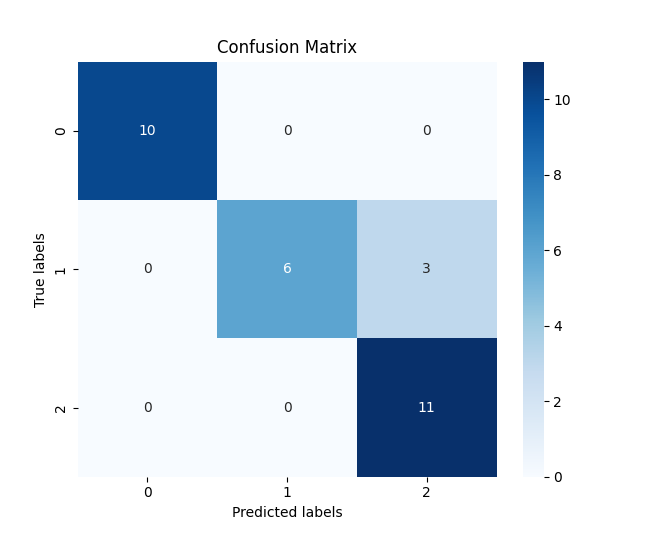


**6. CONFUSION METRICS**

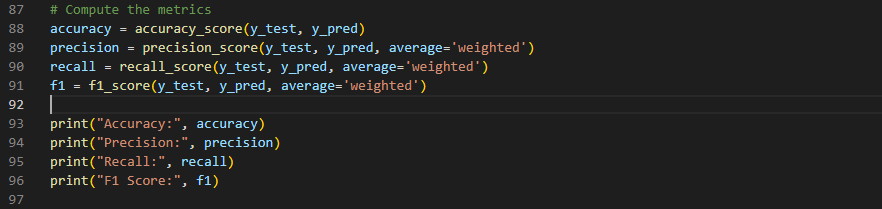
* Use heatmaps to plot the confusion matrix.
* Calculate accuracy, precision, recall, and F1 score based on the confusion matrix.

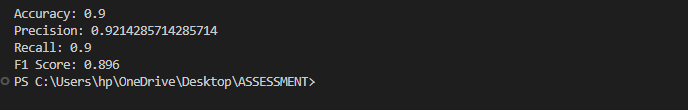






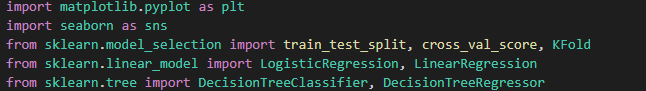
**COMPUTE ACCURACY, PRECISION, RECALL & F1 SCORE**

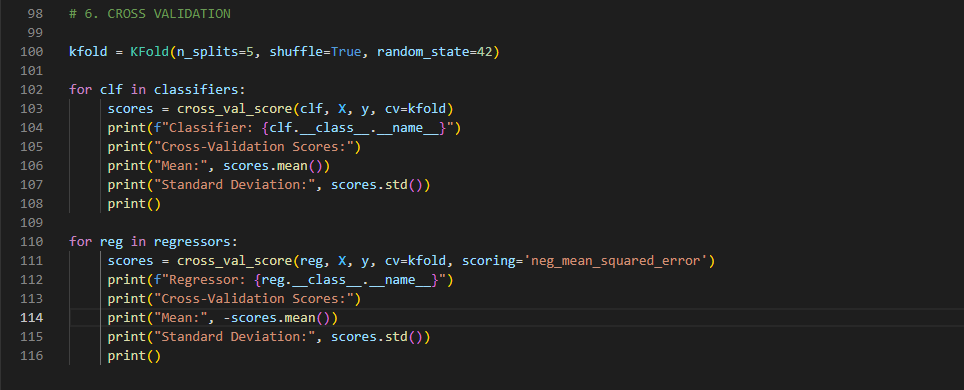


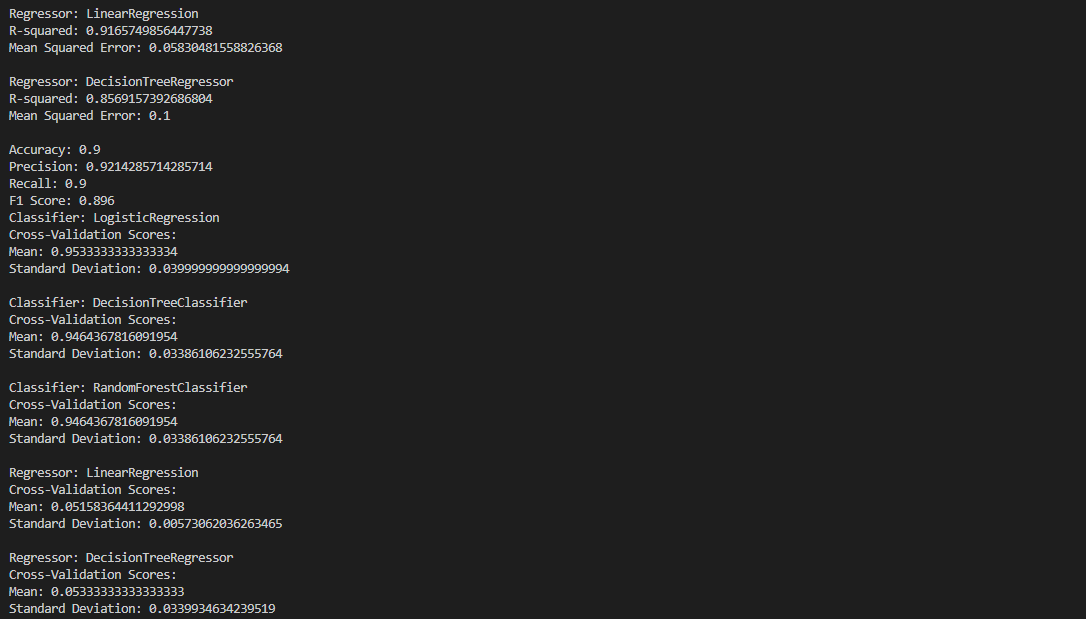


**6.CROSS VALIDATION**

* Use scikit-learn, KFold or cross\_val\_score for both classification and regression models.
* Report the mean and standard deviation of the cross-validation scores to understand the model, stability and variance in performance.







**RESULT:**

Programme executed successfully!