

**Project Report**

**Course:** Programming in Python

**Section:** B

Submitted By,

|  |  |
| --- | --- |
| Name | ID |
| Anik Sen | 20-42138-1 |
| Tanvir Chowdhury | 20-42699-1 |
| Afrida Mahrin Chowdhury | 20-42729-1 |
| Sanjida Esha | 20-42705-1 |

**Date of Submission: 3rd May, 2023**

**Table of Contents**

Page

* Section 1:

*Project Overview* ------------------------------------------------------------- 03

* Section 2:

*Dataset Overview* ------------------------------------------------------------ 04

* Section 3:

*Data Preprocessing and Exploratory Data Analysis* --------------------- 05

* Section 4:

*Model Development* --------------------------------------------------------- 07

* Section 5:

*Comparison of Models* ----------------------------------------------------- 18

*Conclusion* ------------------------------------------------------------------- 21

* **SECTION 1**

***Project Overview:***

The goal of this project is to compare the performance of different classification models on a real dataset. The dataset has 9471 instances and requires data preprocessing and exploratory data analysis.

The project will be implemented in Python and will make use of the scikit-learn library for machine learning. The following classifiers will be used in this project:

1. Naive Bayes
2. K-Nearest Neighbors (KNN)
3. Decision Tree
4. Logistic Regression
5. Support Vector Machine (SVM)

The data cleaning process will involve dropping unnecessary columns and handling missing values using forward fill and backward fill techniques.

Exploratory data analysis will be carried out using histogram plots to analyze the distribution of each feature in the dataset. The correlation matrix will also be plotted as a heatmap to show the correlation between the features.

The dataset will be split into training and testing sets using the train\_test\_split method from scikit-learn. The models will be trained on the training set and tested on the testing set to evaluate their performance.

The performance of the models will be evaluated based on their predictive accuracy, which will be computed using the accuracy\_score method from scikit-learn. Classification reports and confusion matrices will also be used to evaluate the performance of the models.

Finally, the results of the model comparison will be summarized and conclusions will be drawn based on the performance of the models. The project will demonstrate the effectiveness of different classification models and their suitability for different types of datasets.

* **SECTION 2**

***Data Set overview:***

The dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. The device was located on the field in a significantly polluted area, at road level, within an Italian city. Data were recorded from March 2004 to February 2005 (one year)representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Ground Truth hourly averaged concentrations for CO, Non Metanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx) and

Nitrogen Dioxide (NO2) and were provided by a co-located reference certified analyzer. Evidences of cross-sensitivities as well as both concept and sensor drifts are present as described in De Vito et al., Sens. And Act. B, Vol. 129,2,2008 (citation required) eventually affecting sensors concentration estimation capabilities. Missing values are tagged with -200 value. This dataset can be used exclusively for research purposes. Commercial purposes are fully excluded.

Attribute Information:

0 Date (DD/MM/YYYY)

1 Time (HH.MM.SS)

2 True hourly averaged concentration CO in mg/m^3 (reference analyzer)

3 PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)

4 True hourly averaged overall Non Metanic HydroCarbons concentration in microg/m^3 (reference analyzer)

5 True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)

6 PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)

7 True hourly averaged NOx concentration in ppb (reference analyzer)

8 PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)

9 True hourly averaged NO2 concentration in microg/m^3 (reference analyzer)

10 PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)

11 PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)

12 Temperature in Â°C

13 Relative Humidity (%)

14 AH Absolute Humidity

***Dataset reference link:***

*https://archive.ics.uci.edu/ml/datasets/air+quality?fbclid=IwAR1swsn\_wCInfHjjBlr26ETclr0hZ\_IFeBIfMiT929tpeYQlsuffFG0HYgk*

* **SECTION 3**

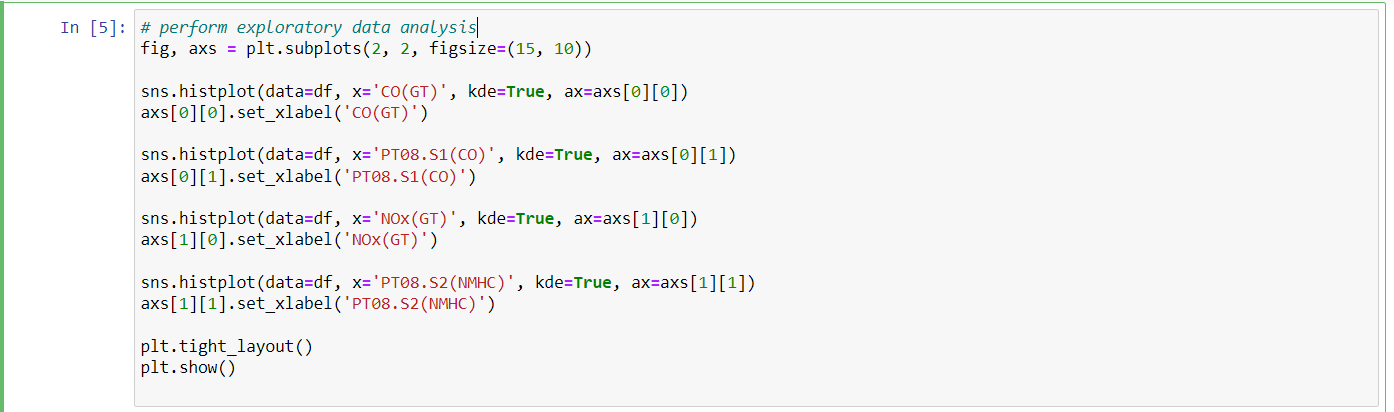
***Data Preprocessing:***

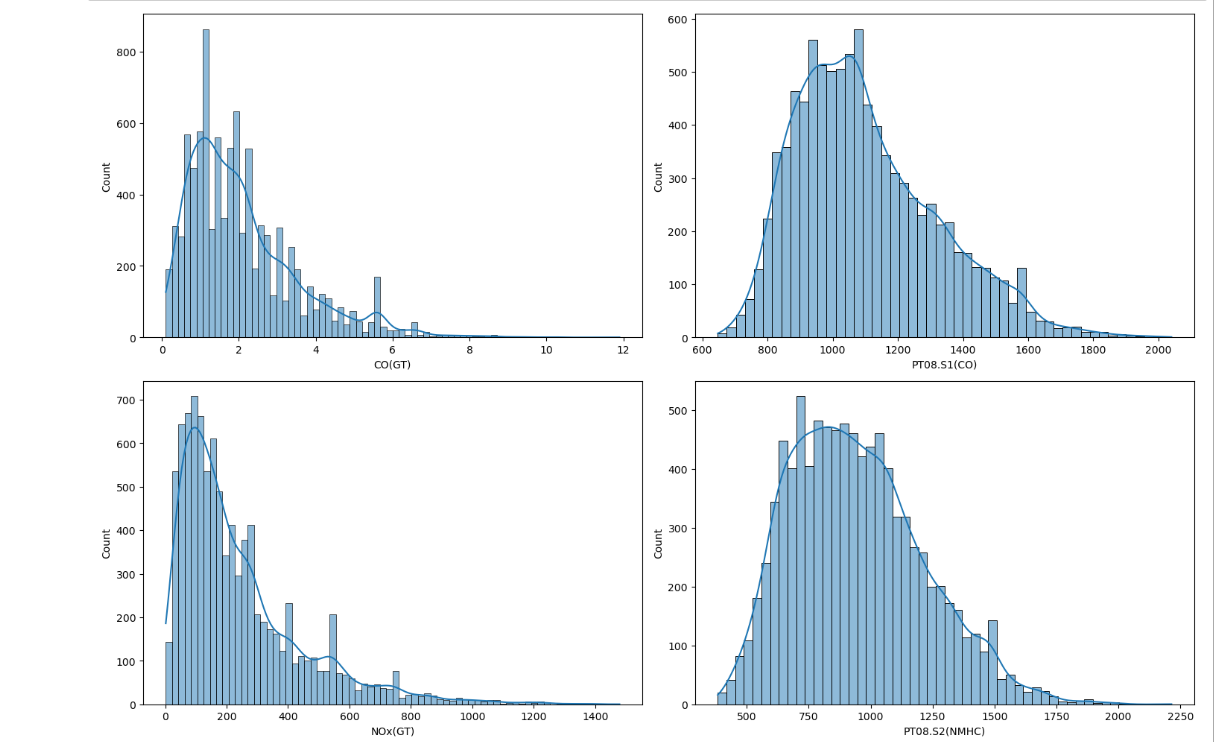
In the code, unnecessary columns are dropped and missing values are handled using forward fill and backward fill methods, which are common techniques in data preprocessing to ensure that the dataset is cleaned and ready for analysis.



***Exploratory Data Analysis for Air Quality Dataset:***

The code generates four histograms for four different variables to visualize their distributions in the dataset. These variables are CO(GT), PT08.S1(CO), NOx(GT), and PT08.S2(NMHC). By analyzing these histograms, one can get an idea of the range, skewness, and shape of the distribution of each variable.or Air Quality Dataset

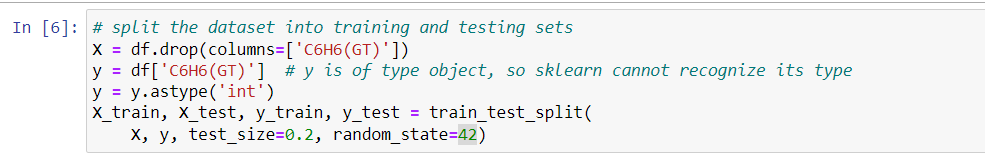




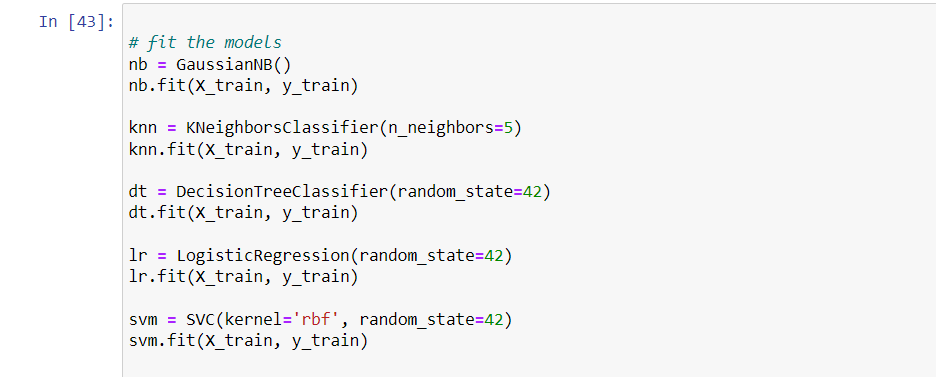
The code performs exploratory data analysis on a dataset using the Seaborn library. It creates a 2x2 grid of subplots using matplotlib, with each subplot containing a histogram plot of a different variable from the dataset. The variables plotted are CO(GT), PT08.S1(CO), NOx(GT), and PT08.S2(NMHC). The histograms have kernel density estimation (KDE) plots overlaid, which show the estimated probability density function of the data. Finally, the plots are displayed using plt.show() and plt.tight\_layout() is used to adjust the spacing between the subplots.

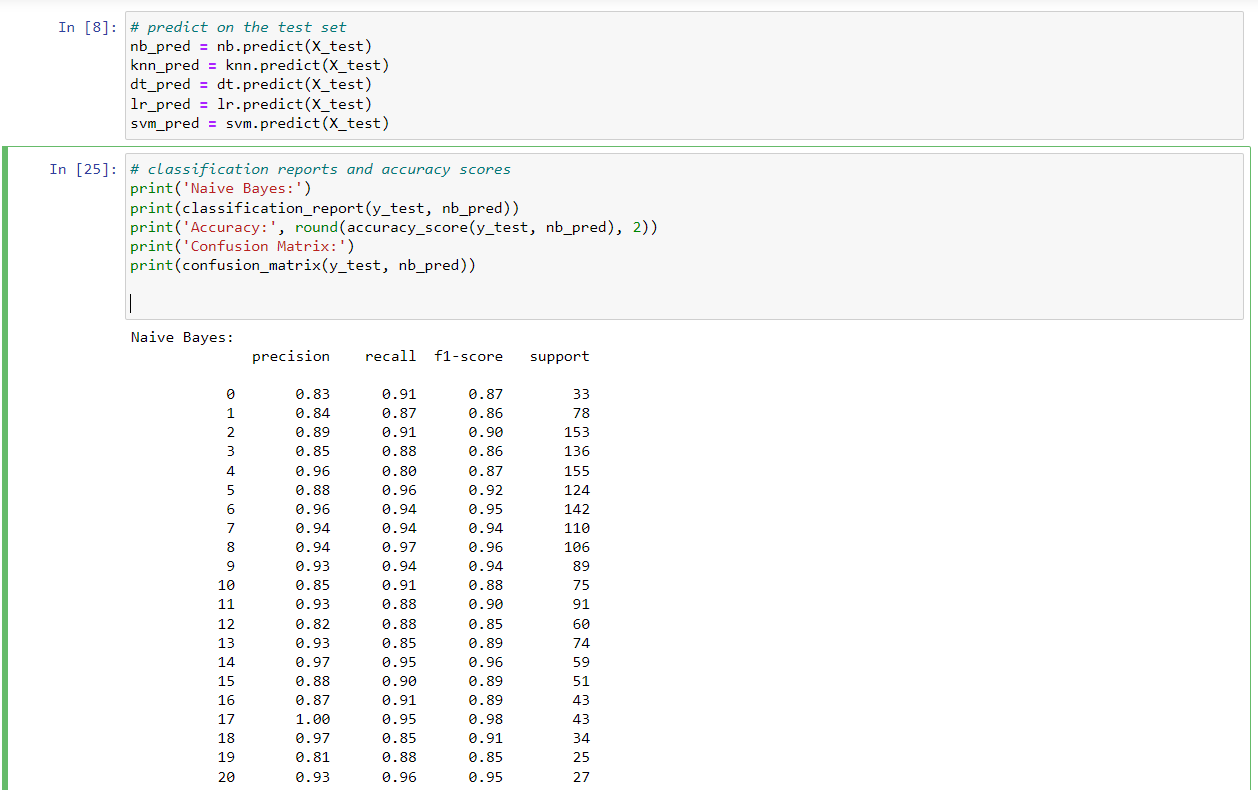
* **SECTION 4:**

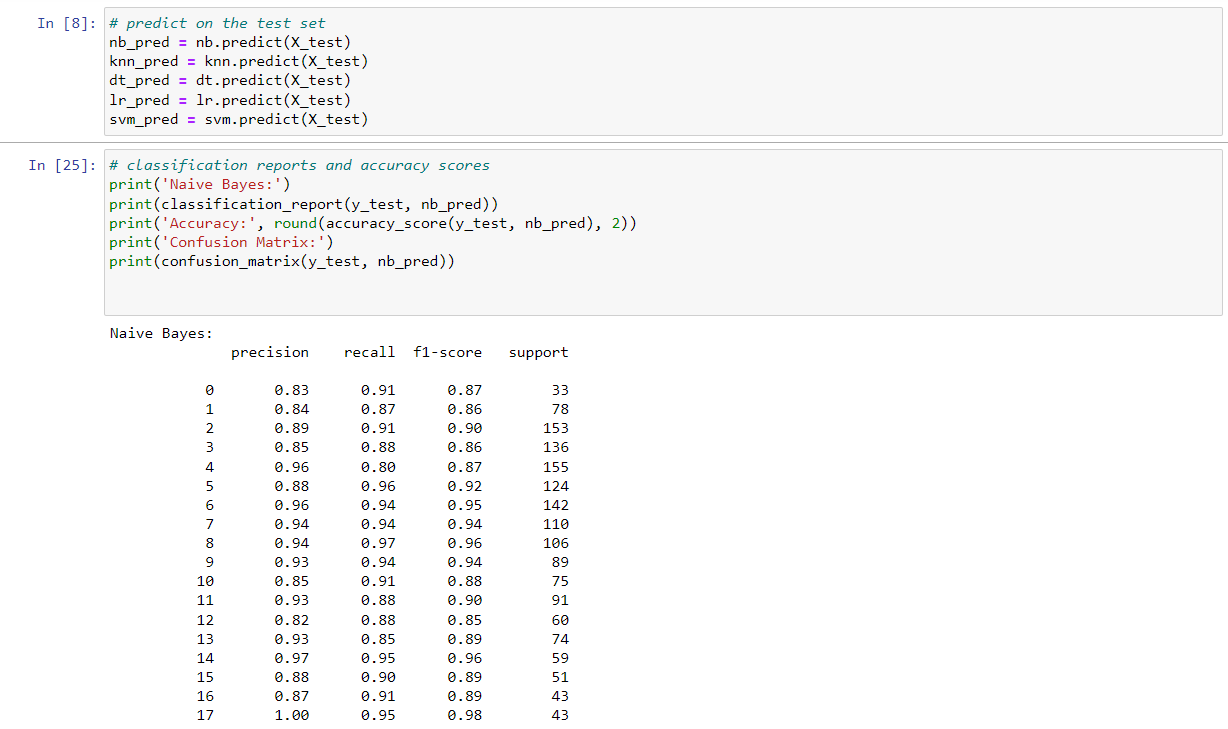
***Model development:***

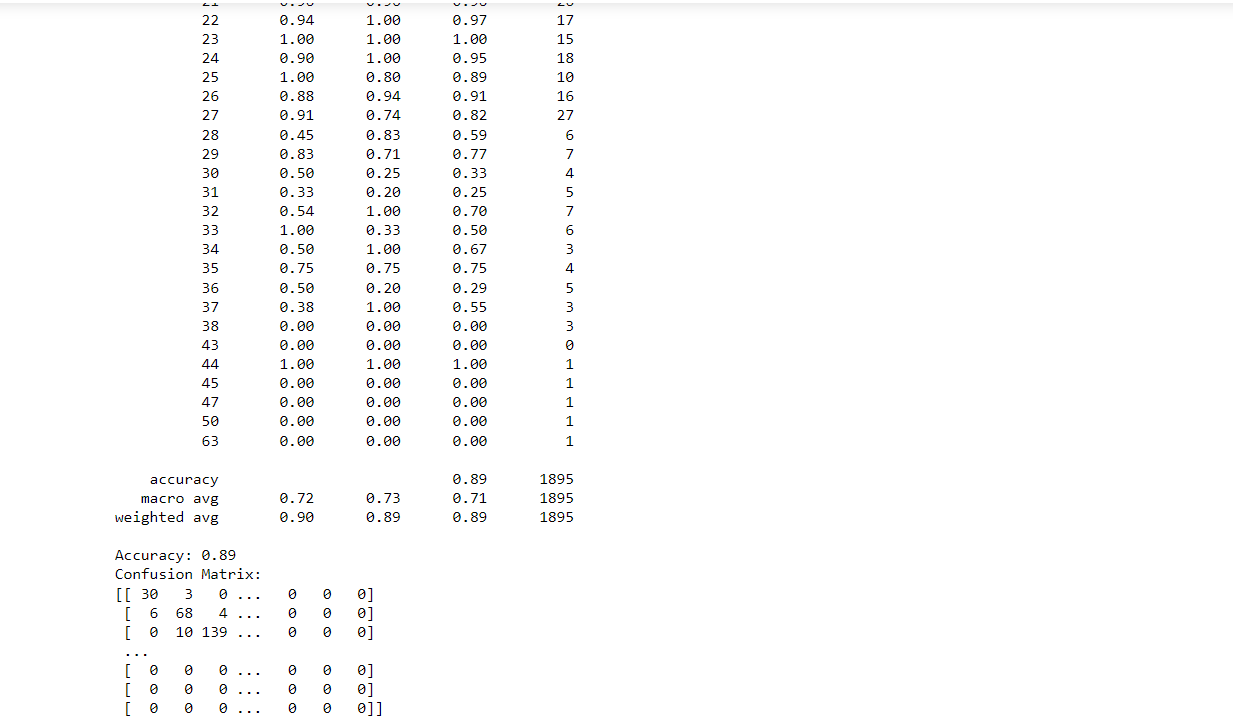
**Splitting the dataset into training and testing sets for regression analysis:**

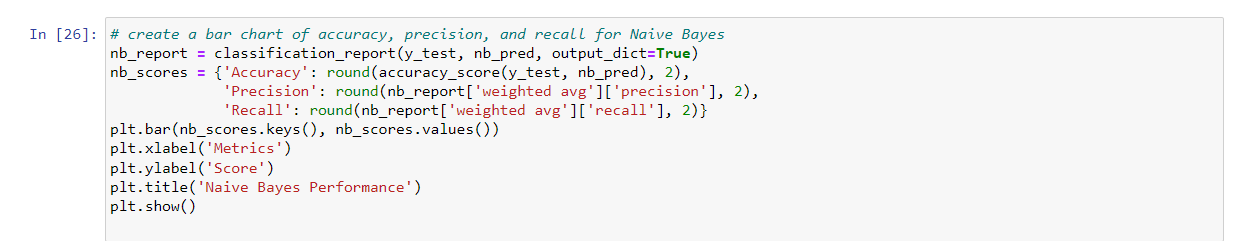
**Training Multiple Classification Models on a Dataset:**

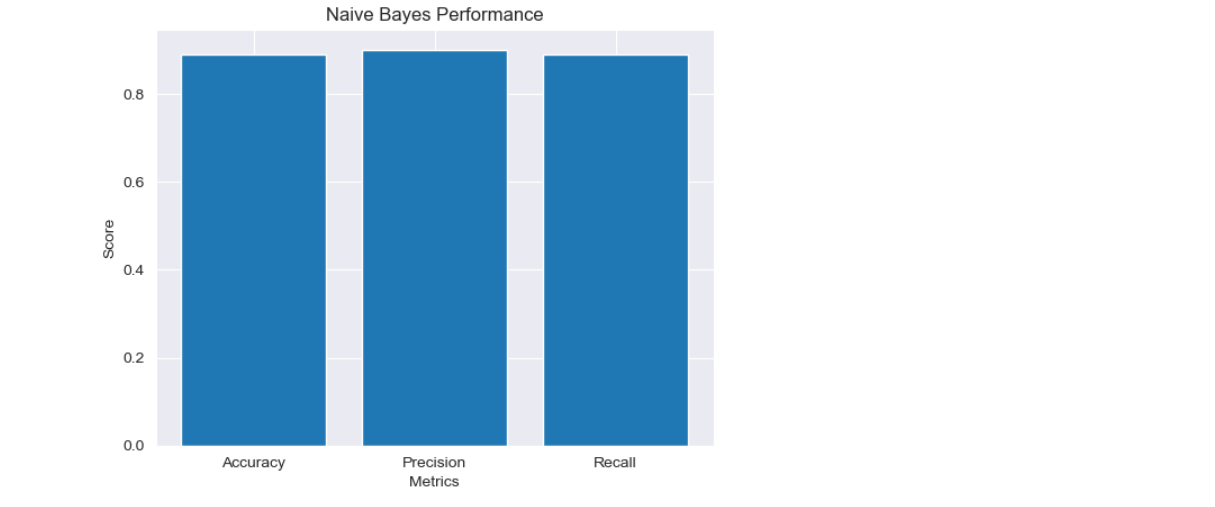
The code creates instances of five different classification models (Gaussian Naive Bayes, K-Nearest Neighbors, Decision Tree, Logistic Regression, and Support Vector Machine) and trains them on a given dataset using the fit() method.

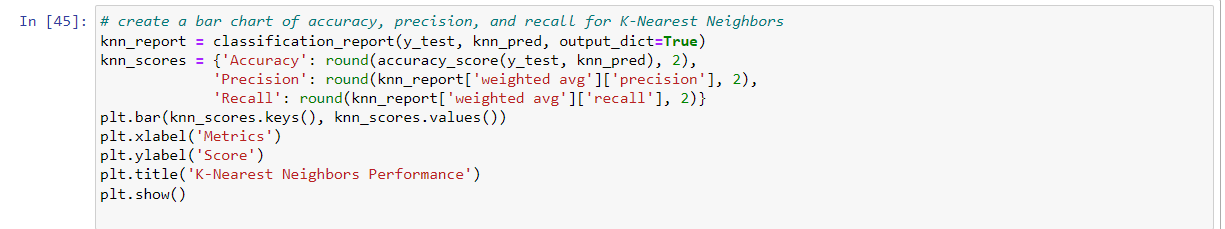


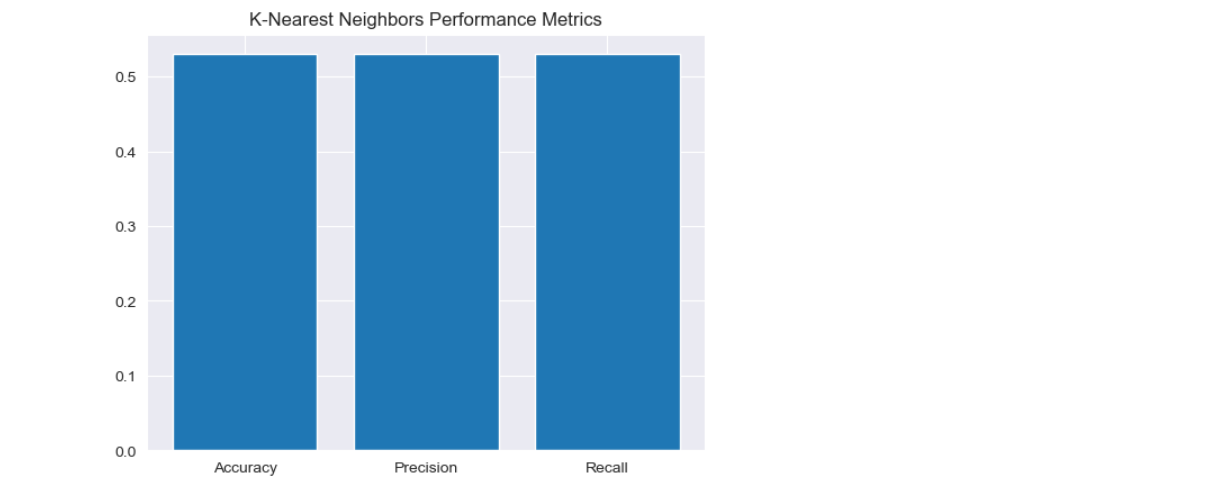


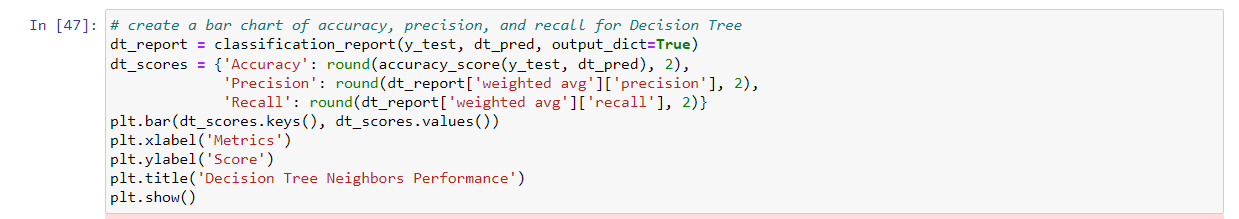


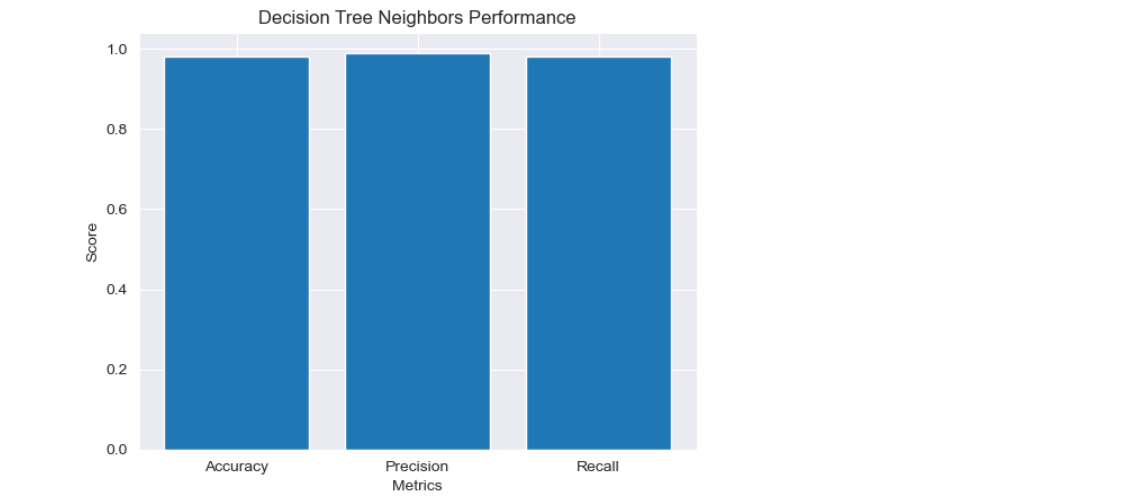


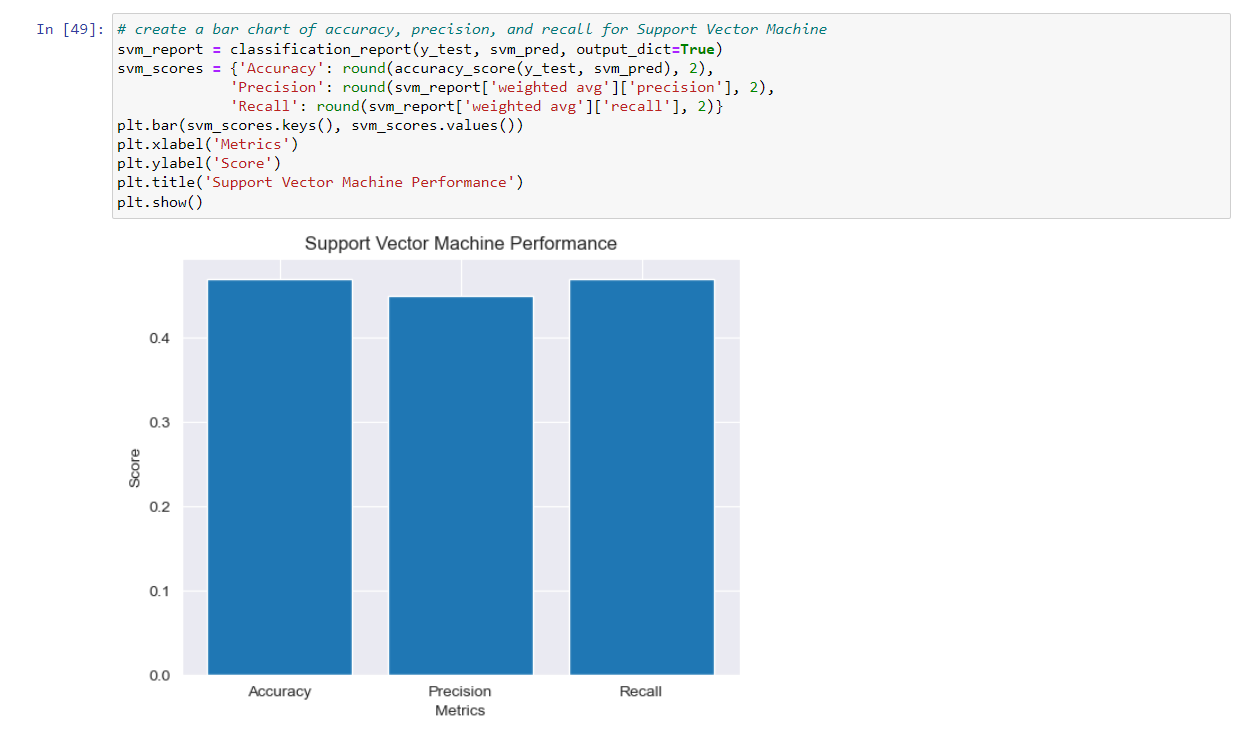


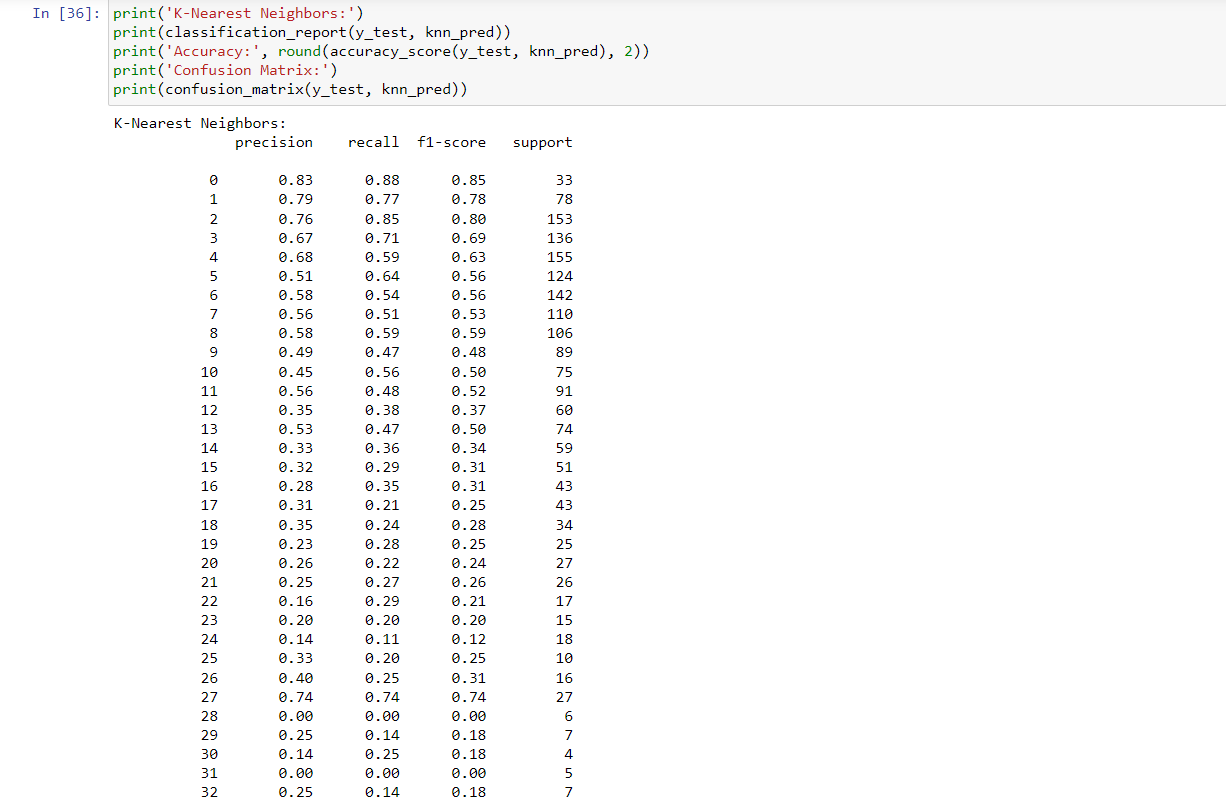


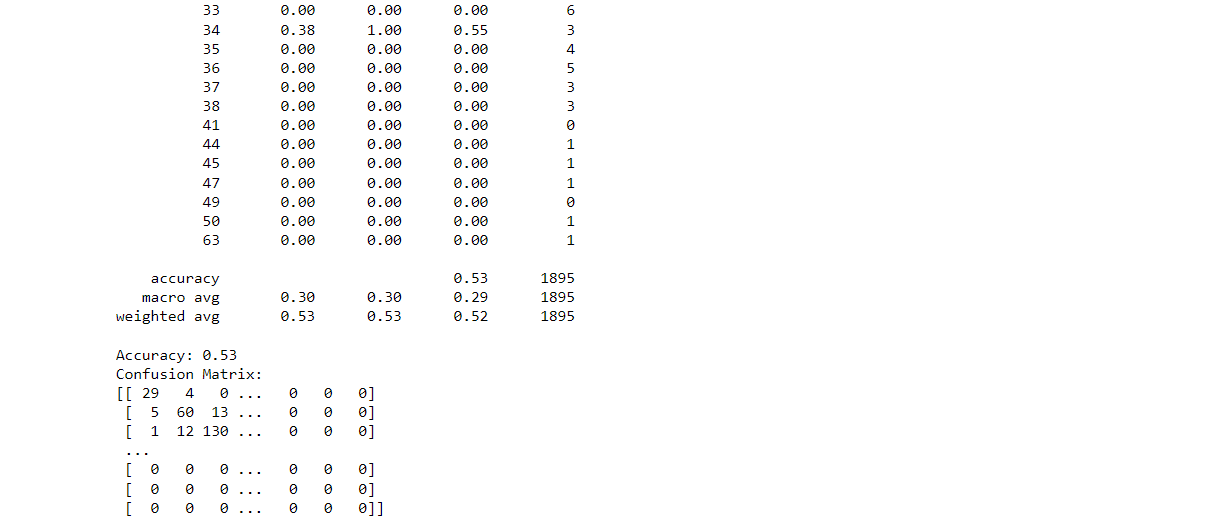


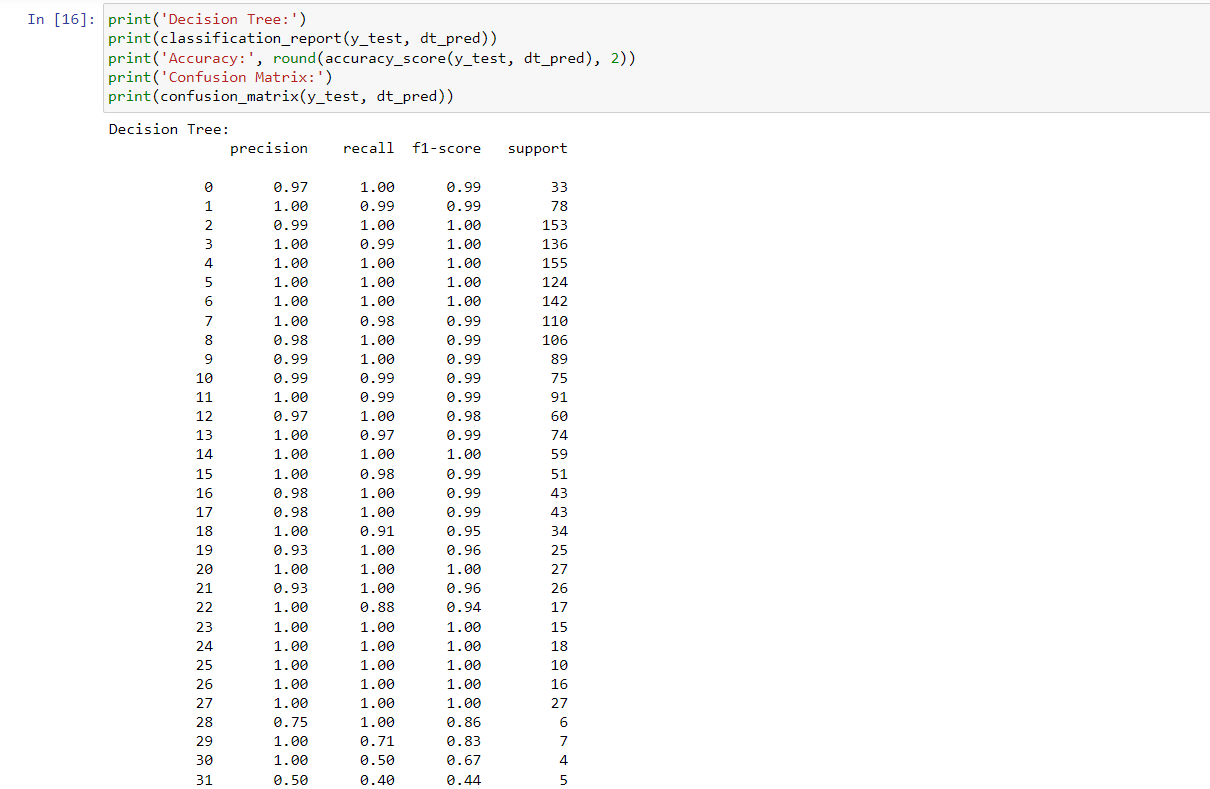


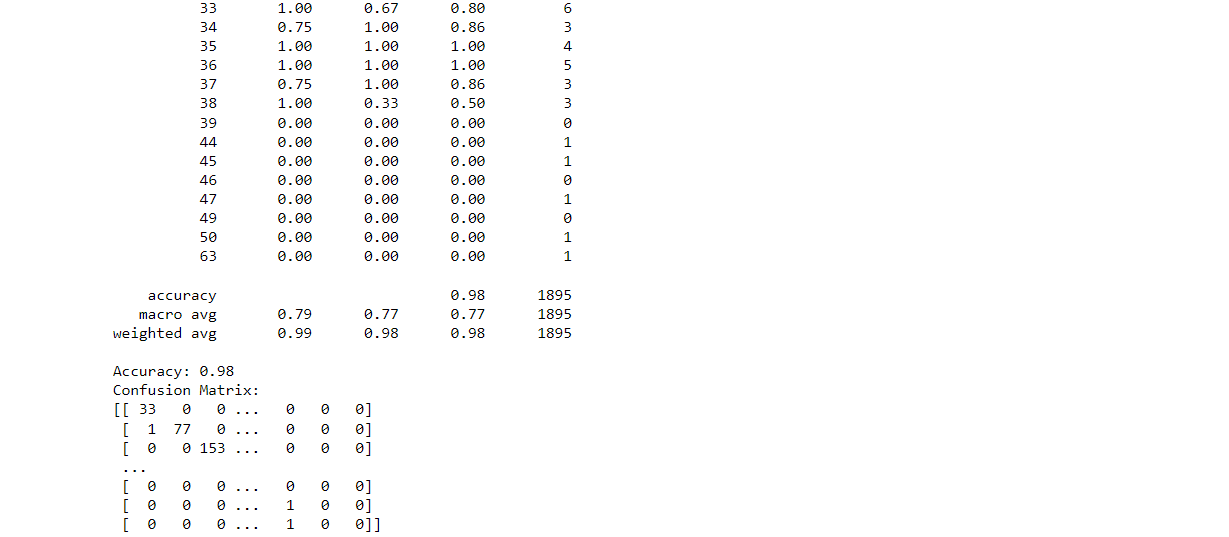


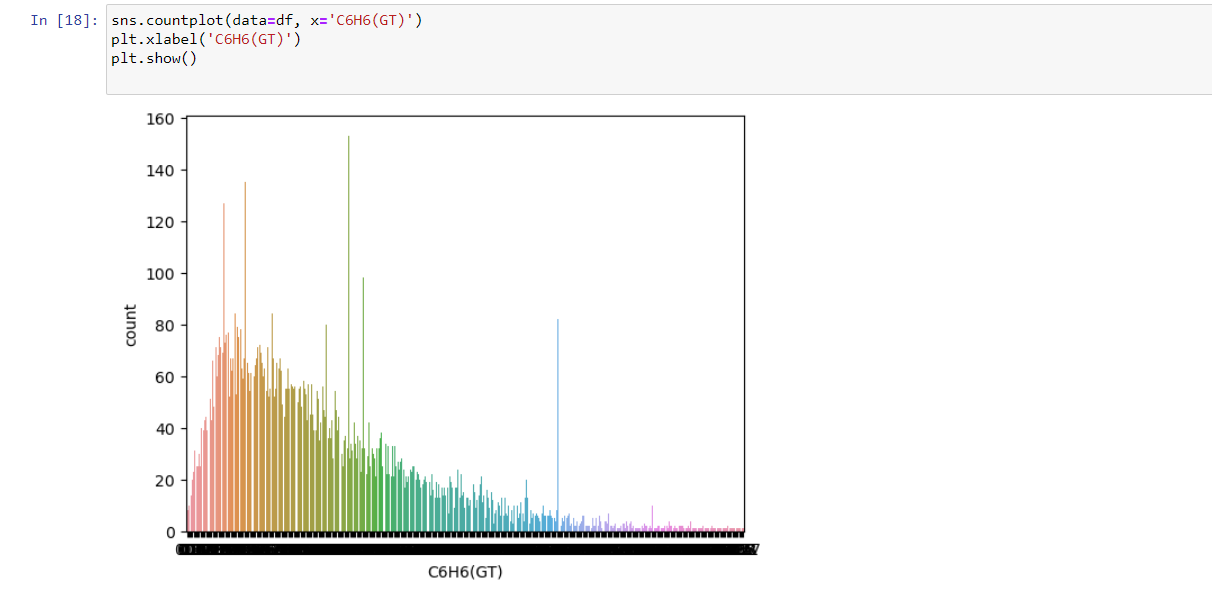
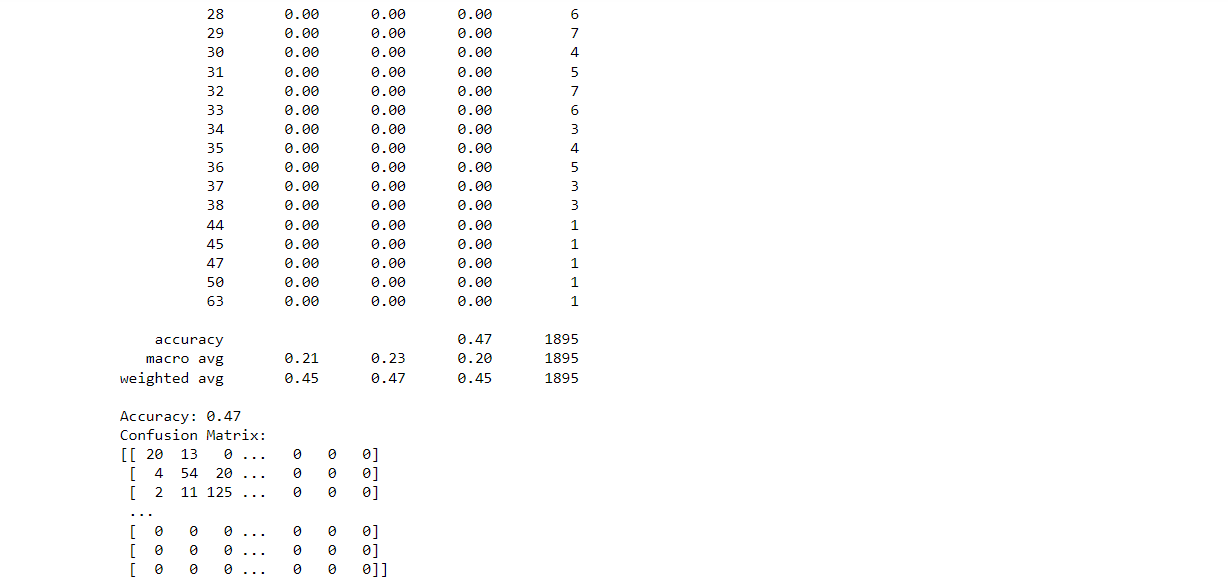
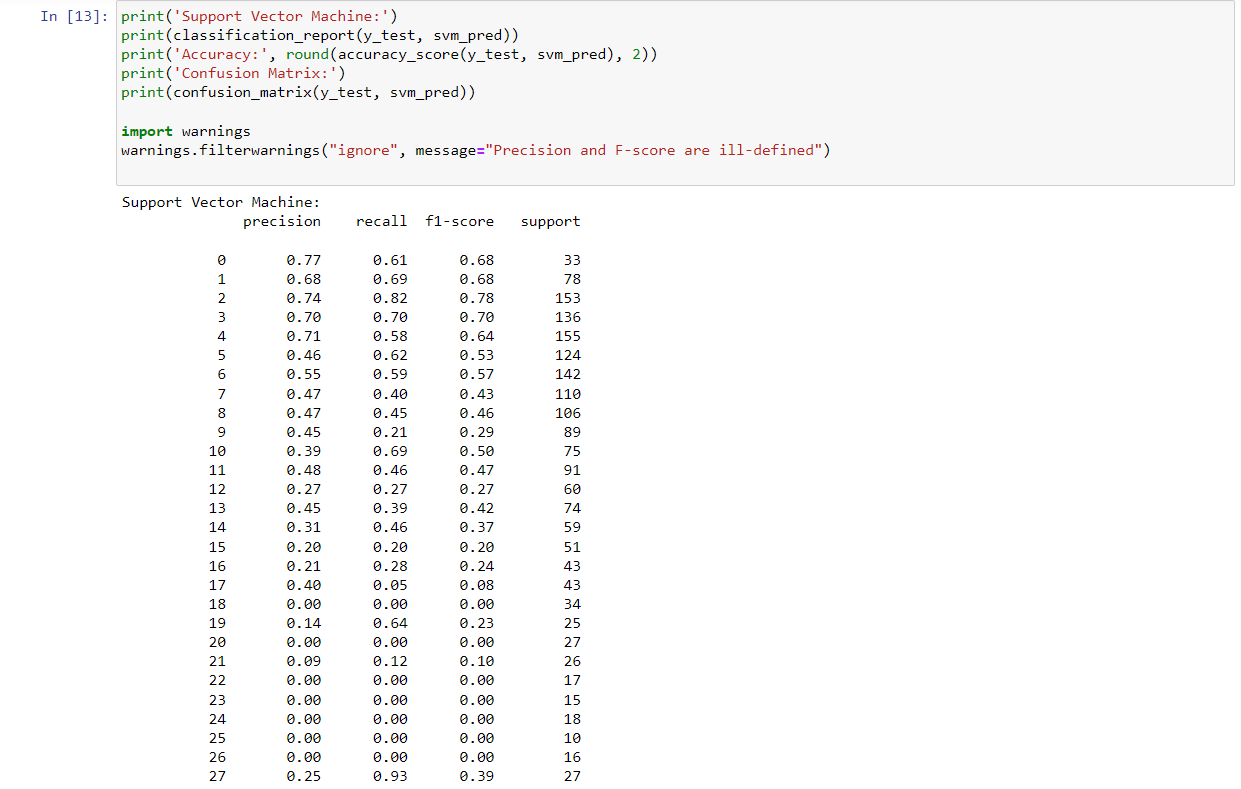
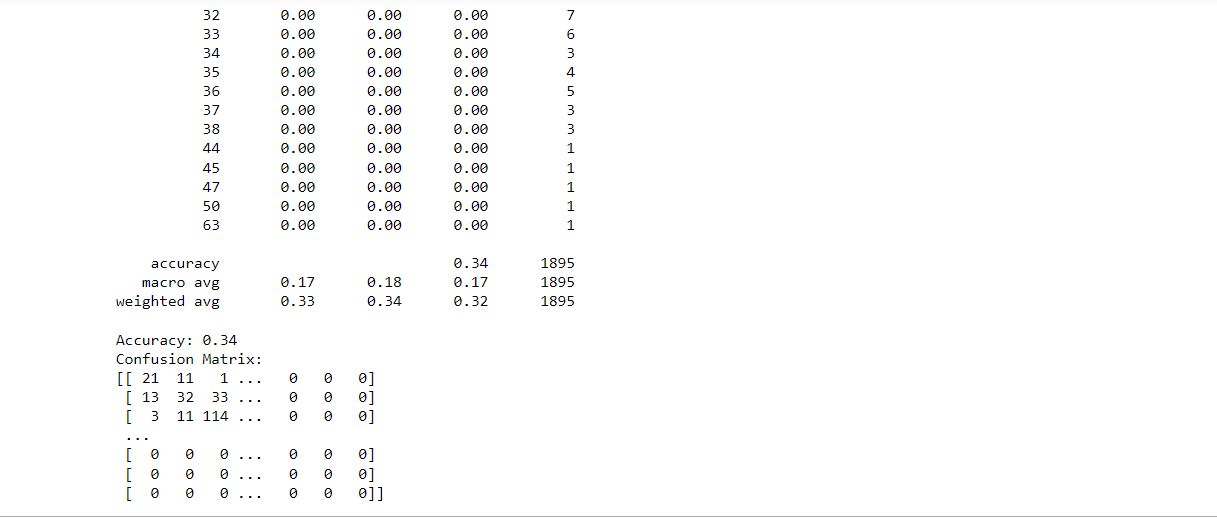
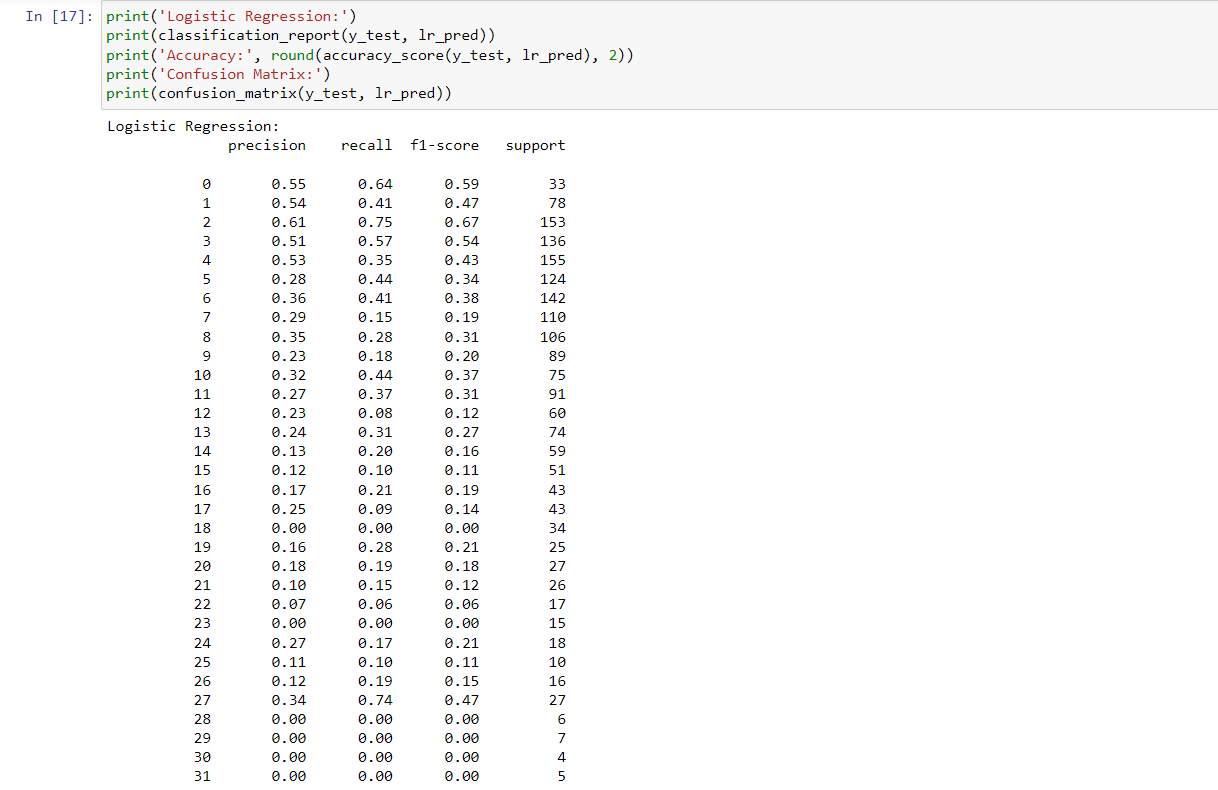
 

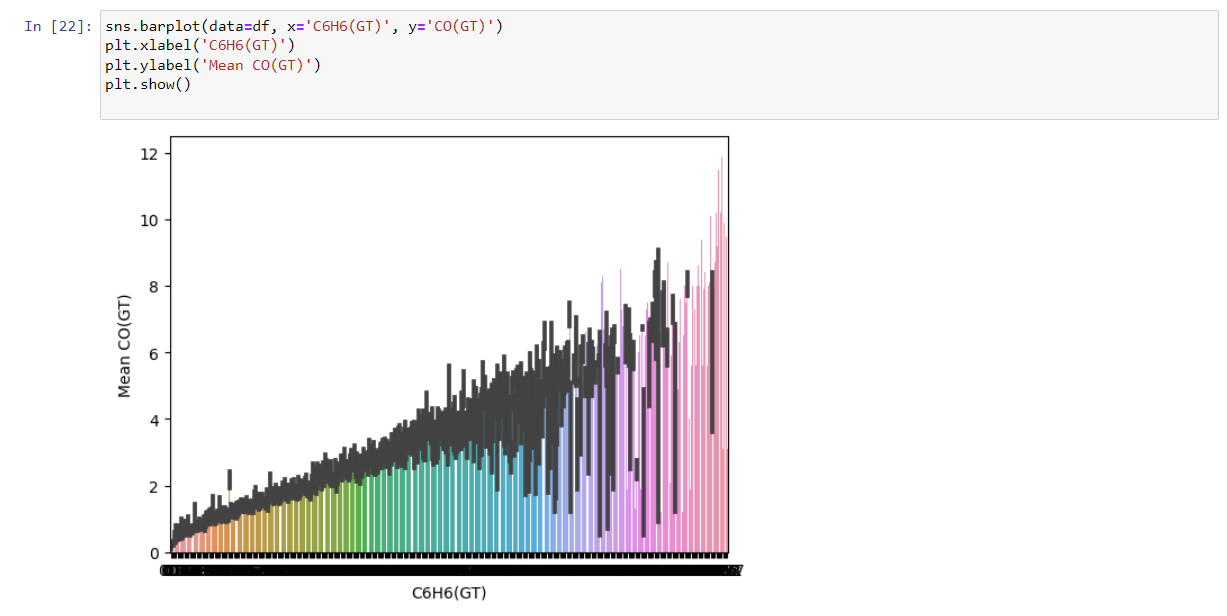


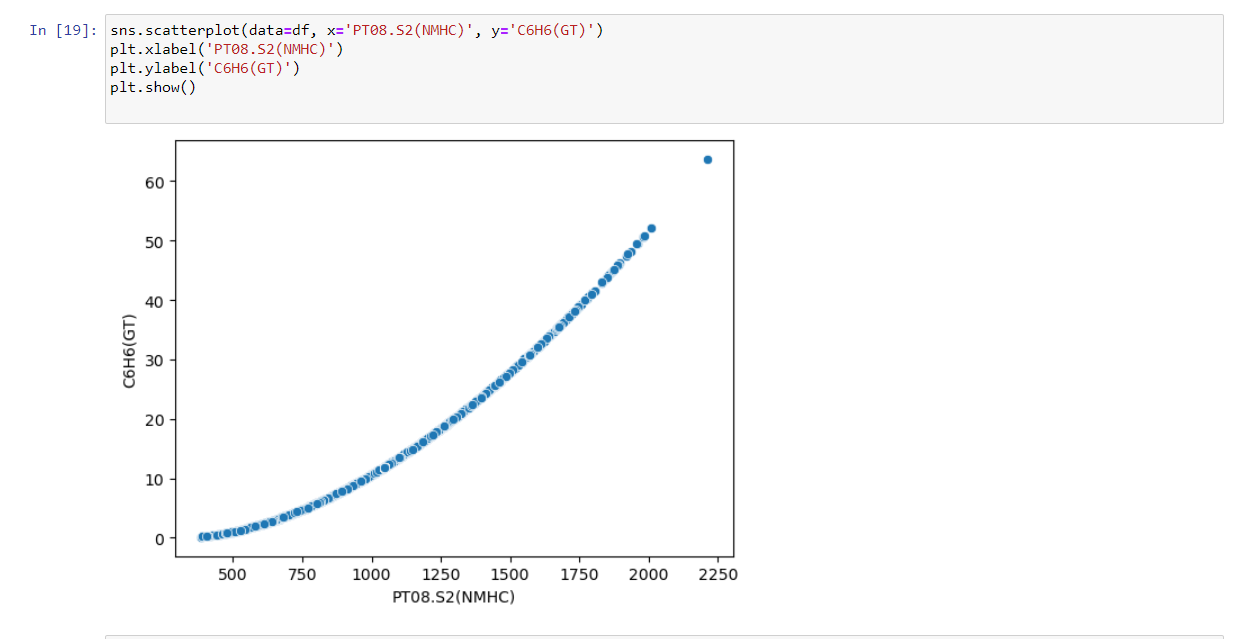


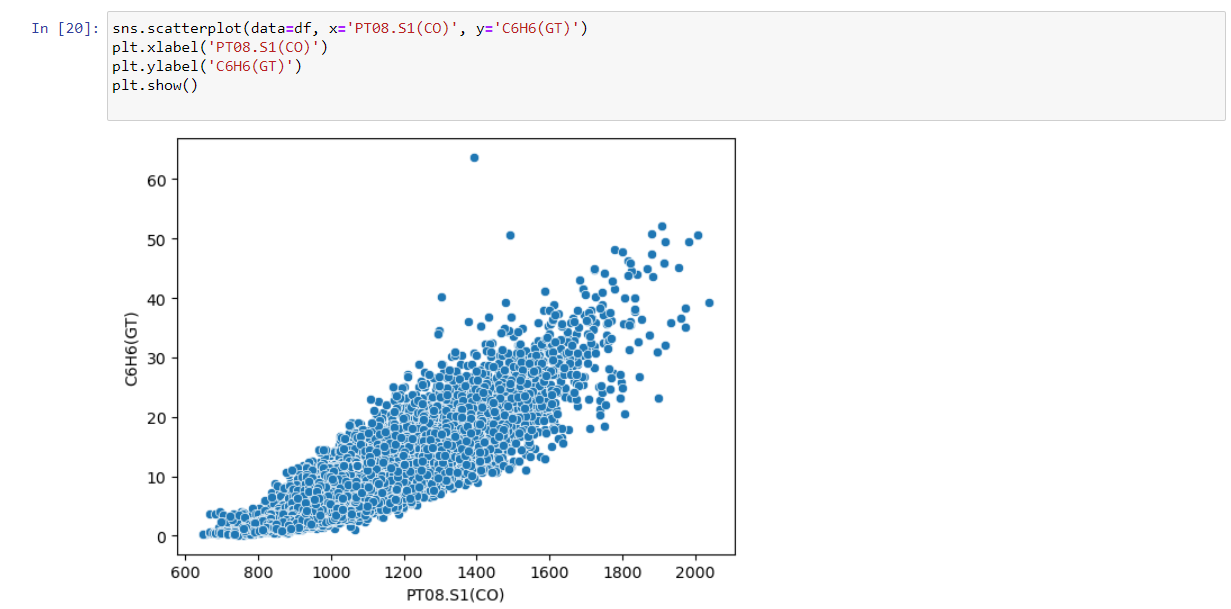










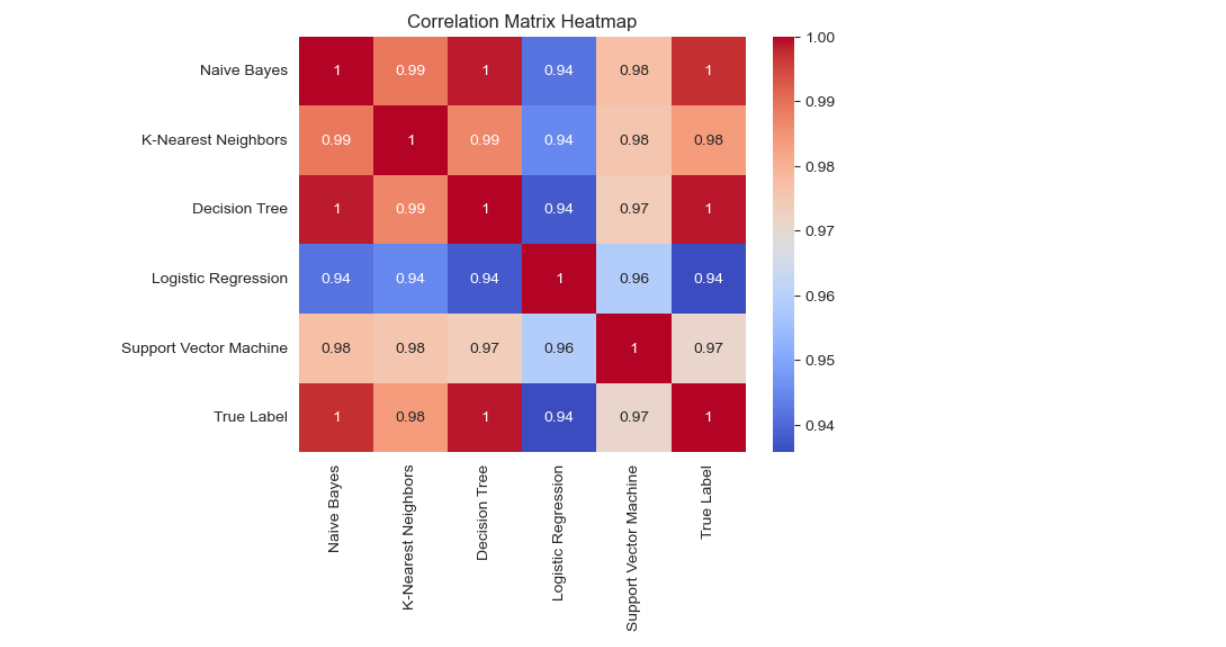


* **SECTION 5:**

***Comparison of Predicted Labels from Multiple Machine Learning Models using Correlation Matrix Heatmap :***

This code takes the predicted labels from Naive Bayes, K-Nearest Neighbors, Decision Tree, Logistic Regression, and Support Vector Machine models and creates a correlation matrix heatmap to visualize the correlation between these predicted labels and the true labels. The heatmap provides an overview of how the models are performing and how they are correlated with each other.

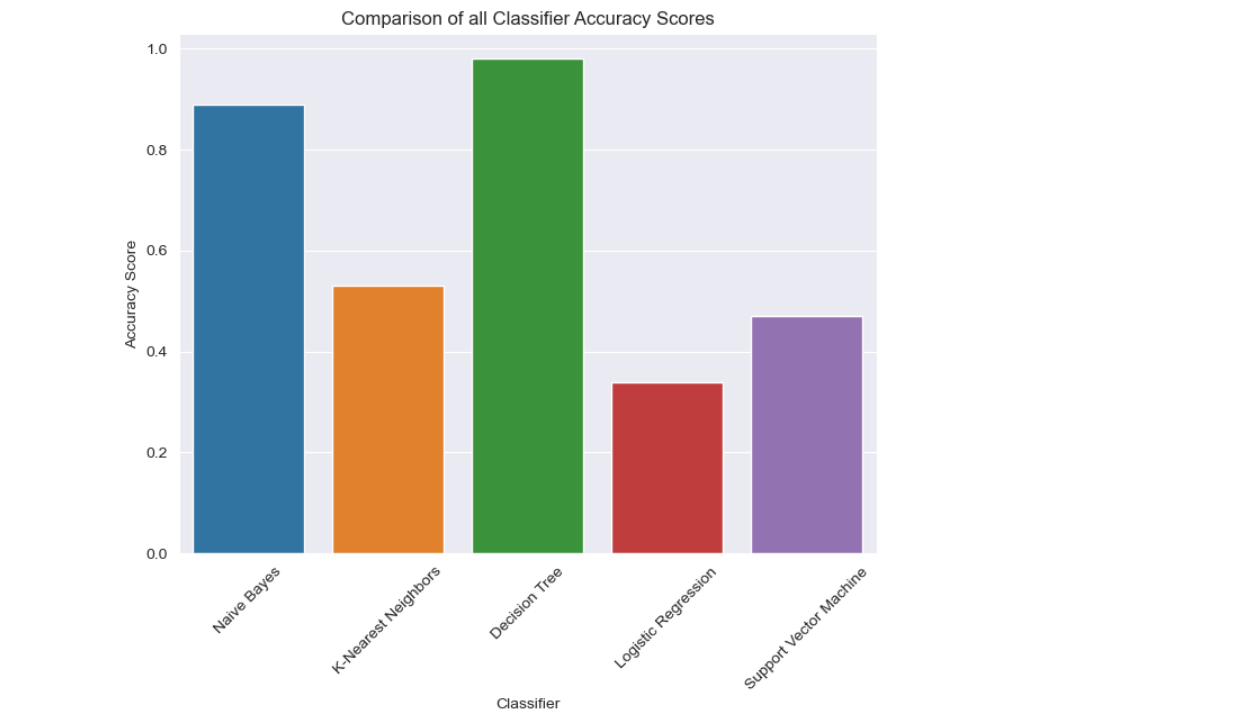




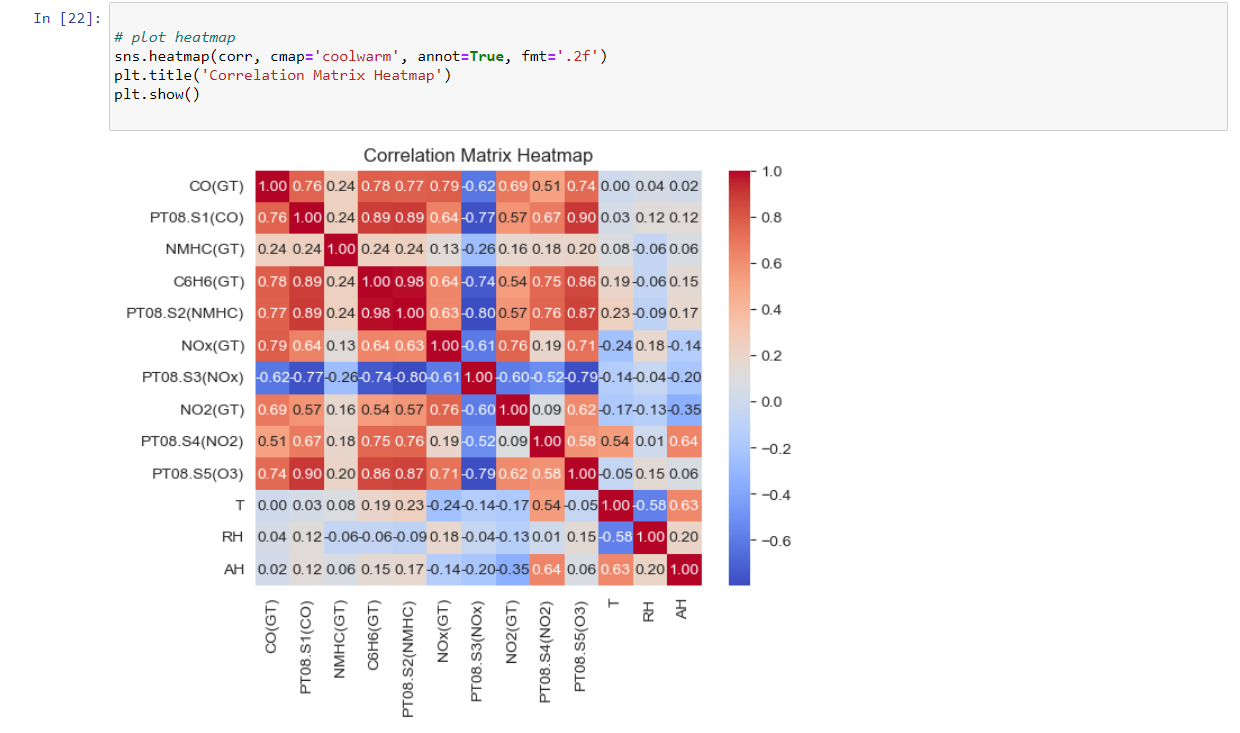
***Comparison of all Classifier Accuracy Scores***

This following code generates a bar chart showing the accuracy scores of various classification models including Naive Bayes, K-Nearest Neighbors, Decision Tree, Logistic Regression, and Support Vector Machine. The chart helps to visualize and compare the performance of these models based on their accuracy scores.





***Correlation Matrix Heatmap***



In conclusion, this project had aimed to explore the performance of different classification models on a real dataset using Python. The project involved data preprocessing, exploratory data analysis, and the implementation of five different classification models including Naive Bayes, KNN, Decision Tree, Logistic Regression, and SVM.

The project had focused on evaluating the predictive accuracy of the models on the testing set, and further analysis had been done using classification reports and confusion matrices. Through this project, we had aimed to demonstrate the suitability of different classification models for different types of datasets and provide a comparison of their performance.

By carrying out this project, we had hoped to provide valuable insights into the effectiveness of different classification models and their potential applications.

---------------------------------------------------------------------------------