**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

Program: B.Tech\MBA.Tech

**Course: Machine Learning**

**Experiment No.07**

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**A.1 Aim:** To implement Naïve Bayes algorithm for classification.

**A.2 Prerequisite:**

Python Programming, Pandas library, Numpy Library, MatplotLib, Seaborn Library

**A.3 Outcome:**

**After successful completion of this experiment students will be able to:**

1. Implement Naïve Bayes technique for the classification
2. Compare results of Naïve Bayes
3. Understand and infer results of different classification metrics

**A.4 Theory:**

**A.4.1 Naïve Bayes Classifier:**

* Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
* It is mainly used in *text classification* that includes a high-dimensional training dataset.
* Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
* **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
* Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

**A.4.2 Bayes’ Theorem:**

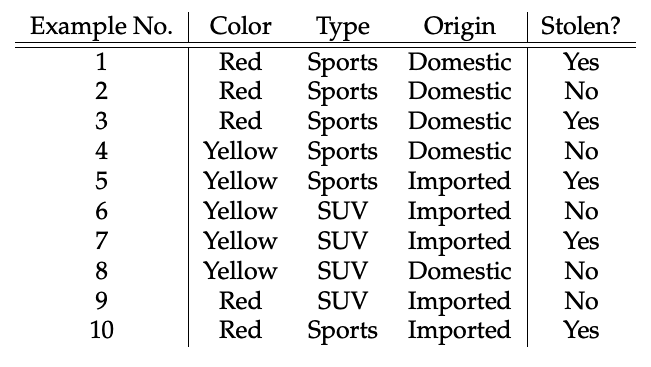
* Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
* The formula for Bayes' theorem is given as:
* Naïve Bayes Classifier Algorithm
* Where,
* P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
* P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
* P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
* P(B) is Marginal Probability: Probability of Evidence.

**A.4.3 Working of Naïve Bayes Classifier:**

* Convert the given dataset into frequency tables.
* Generate Likelihood table by finding the probabilities of given features.
* Now, use Bayes theorem to calculate the posterior probability.

**Tasks:**

**Task 1: For the given car data set, apply Naïve Bayes algorithm for the classification. Show all the steps of training phase and identify the class for test data point (color =Yellow, Type=Sports, Origin=Domestic). Solve answer on paper and upload the image.**



**Task 2:** **For the given adult.csv dataset, perform below operations.**

* **Upload data set into the dataframe**
* **Check the shape of the data set.**
* **Find out all the categorical columns from the data set**
* **Check if null values exist in all the categorical columns**
* **Identify the problems with “workclass”, “Occupation”,”native\_country” columns and rectify it.**
* **Explore numeric columns and check any null values exist for the numeric columns.**
* **Create a feature vector with x= all the columns except income and y=income**
* **Implement feature engineering for the train, test split data set:**
  + **Check the data types of columns of the input features of training data set**
  + **Identify categorical columns that has null values and fill them with most probable value in the data set**
  + **Repeat above step for the input features of test data set**
  + **Apply one hot encoding on all the categorical columns**
  + **Apply feature scaling using robust scaler**

**Task 4: Implement Naïve Bayes on the given data set.**

**Task 5: Print confusion matrix**

**Task 6: Check accuracy score of the classifier.**

**Task 7: Print classification report using sklearn.metrics**

**Task 7: Draw ROC curve for the model.**

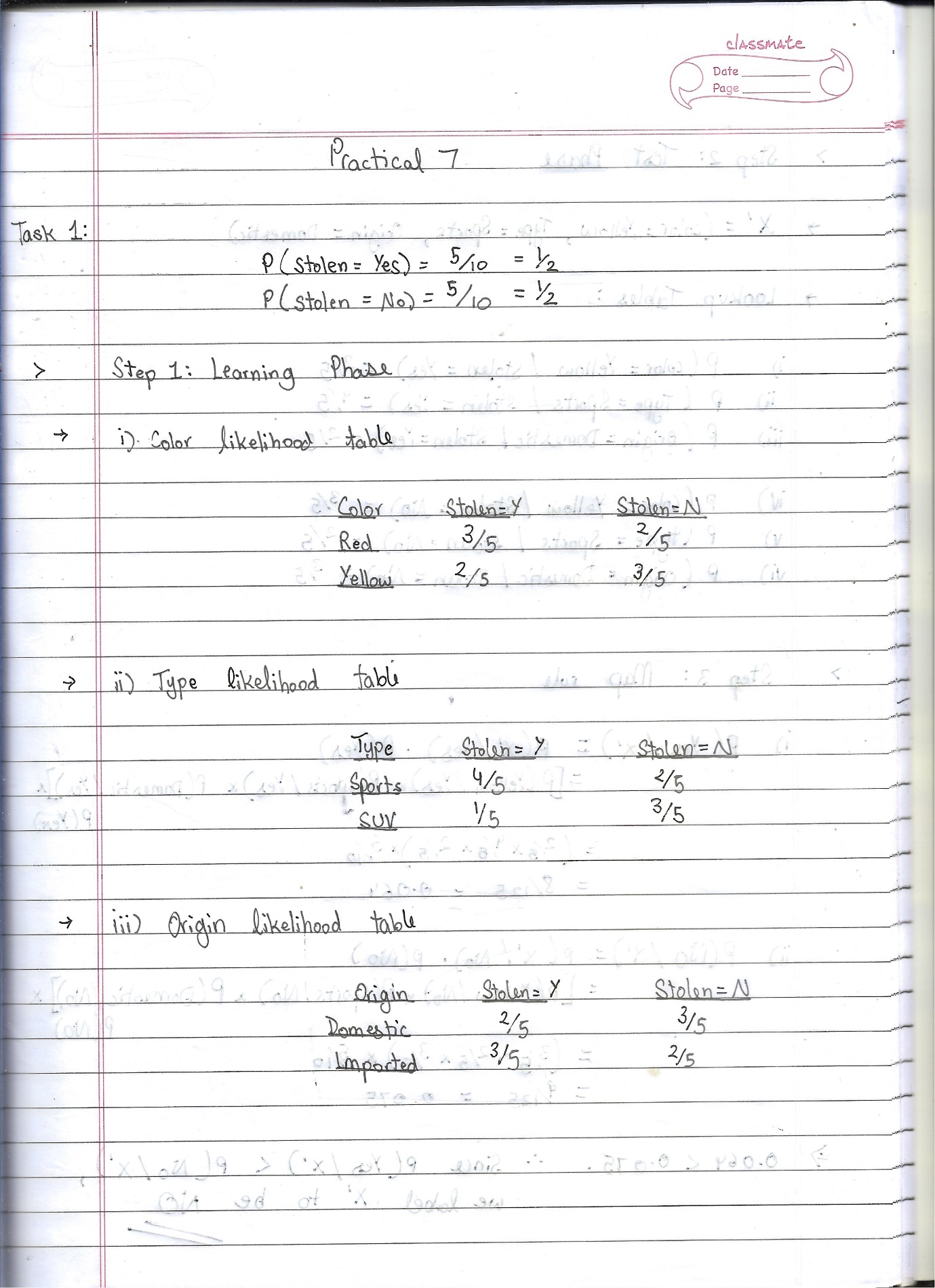
PART B

(PART B : TO BE COMPLETED BY STUDENTS)

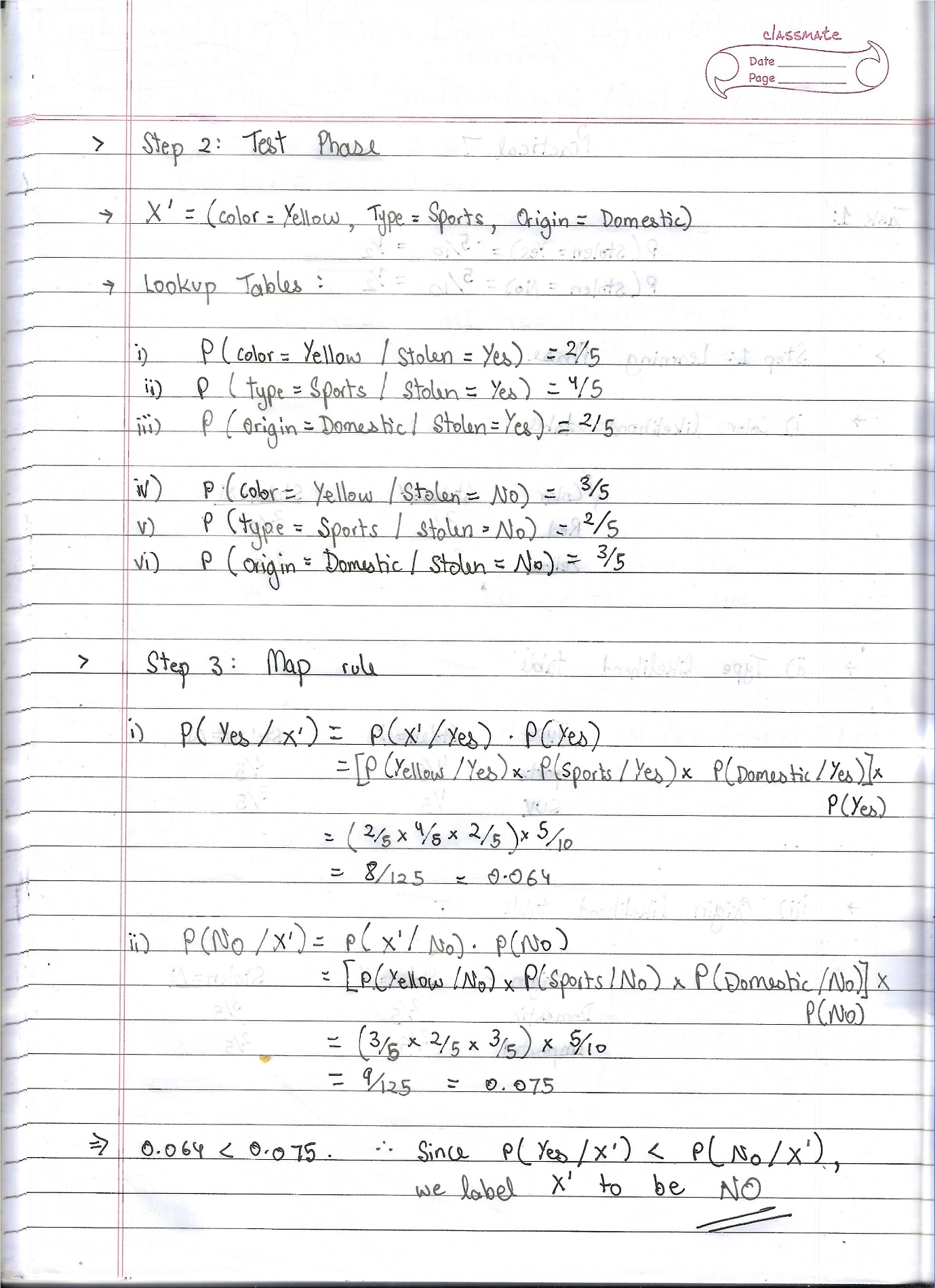
***(Students must submit the soft copy as per following segments within two hours of the practical.)***

|  |  |
| --- | --- |
| Roll No. N052 | Name: Pratyush Kumar |
| Class : MBA Tech CE (div. D) | Batch : B2 |
| Date of Experiment: 24-02-2024 | Date of Submission: 25-02-2024 |
| Grade : |  |

**B.1 Task 1**



*pratyush*



*pratyush*

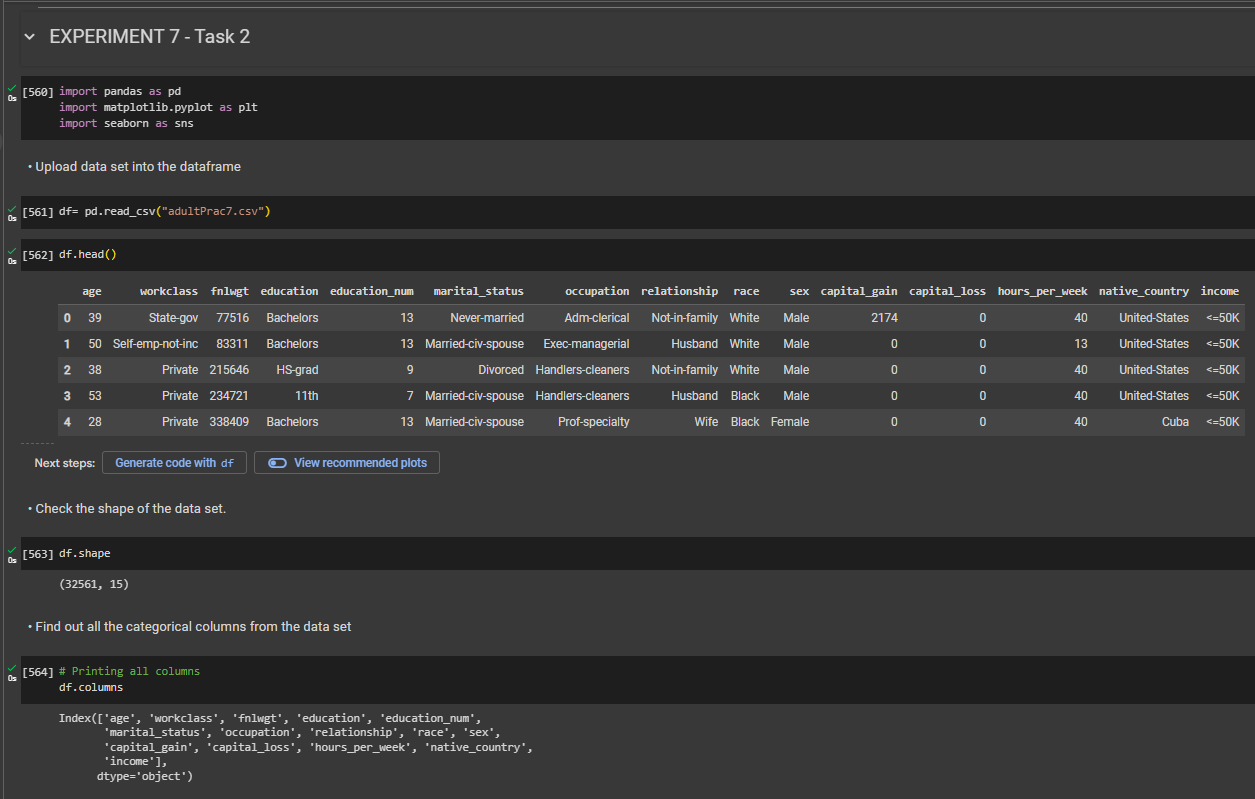
**B.2 Task 2**

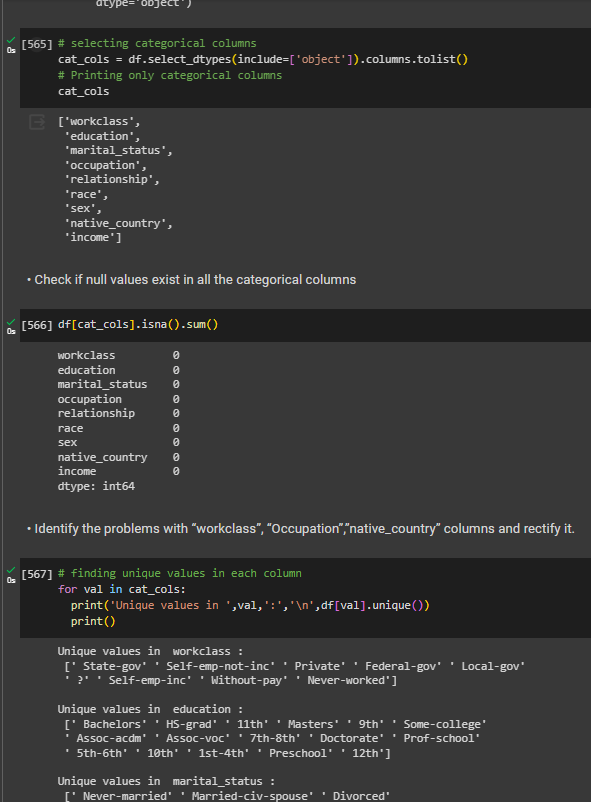
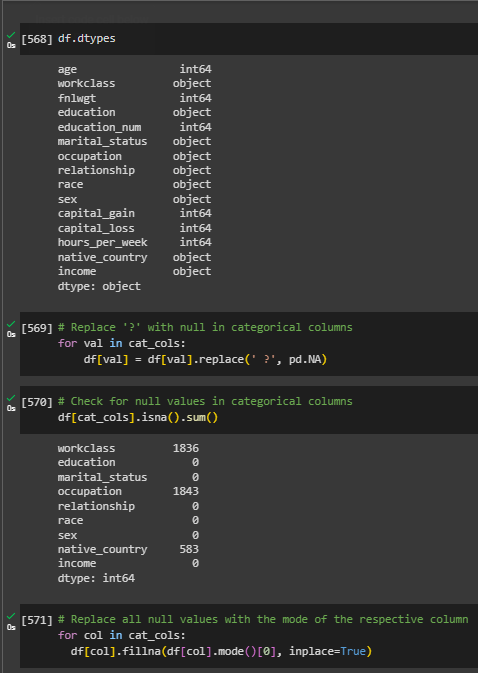
**Colab link:** https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY?usp=sharing

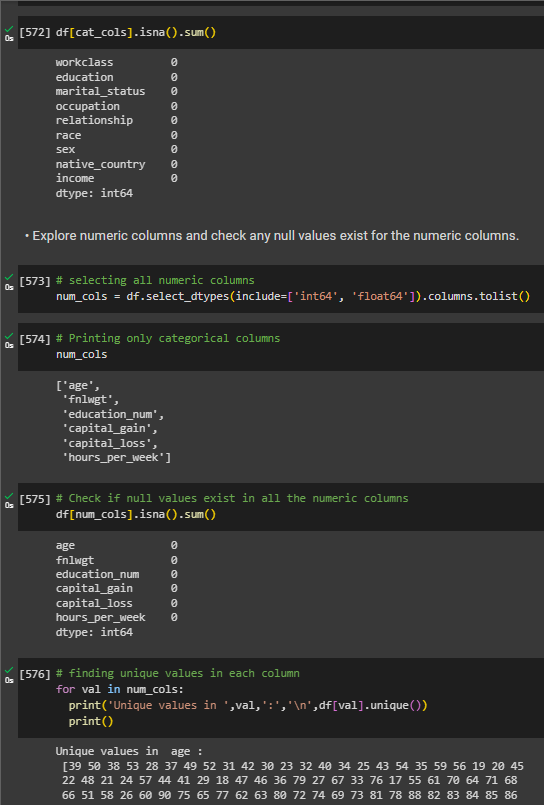
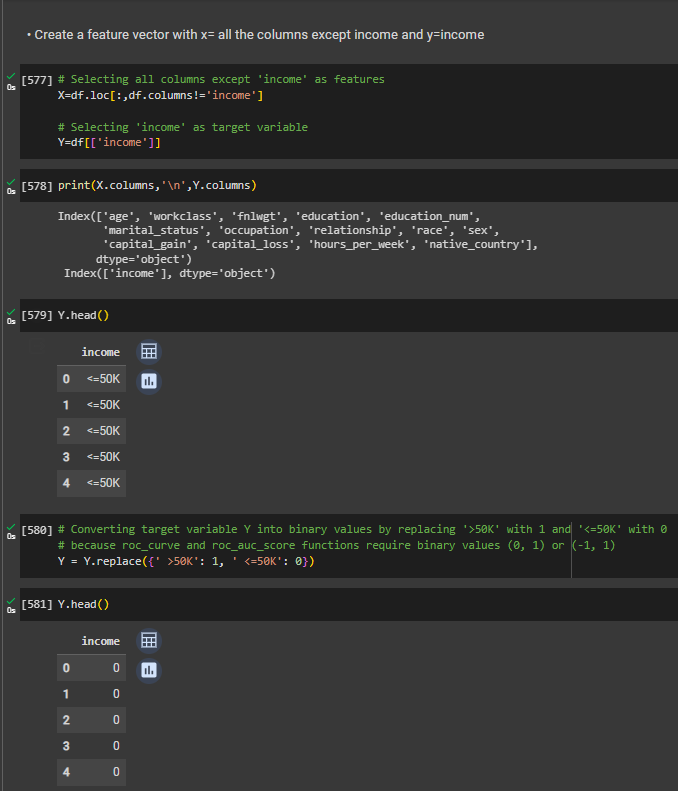
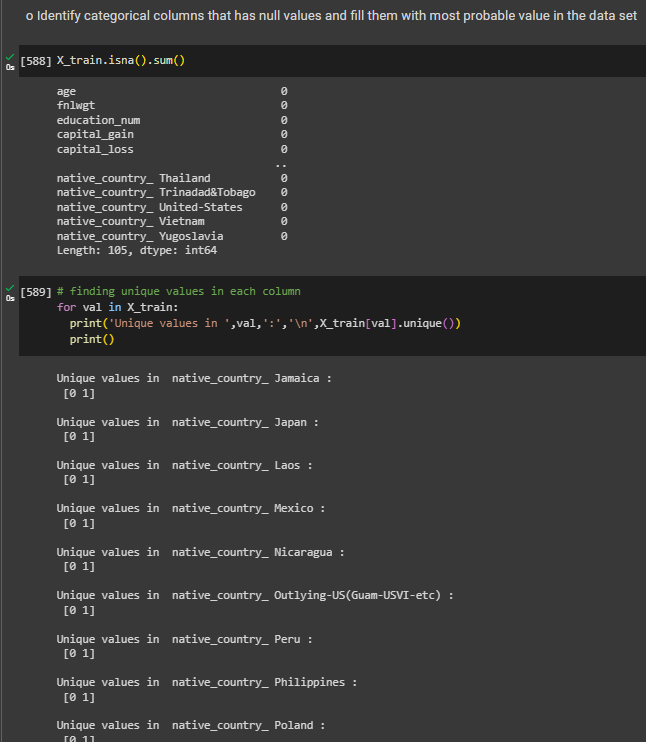
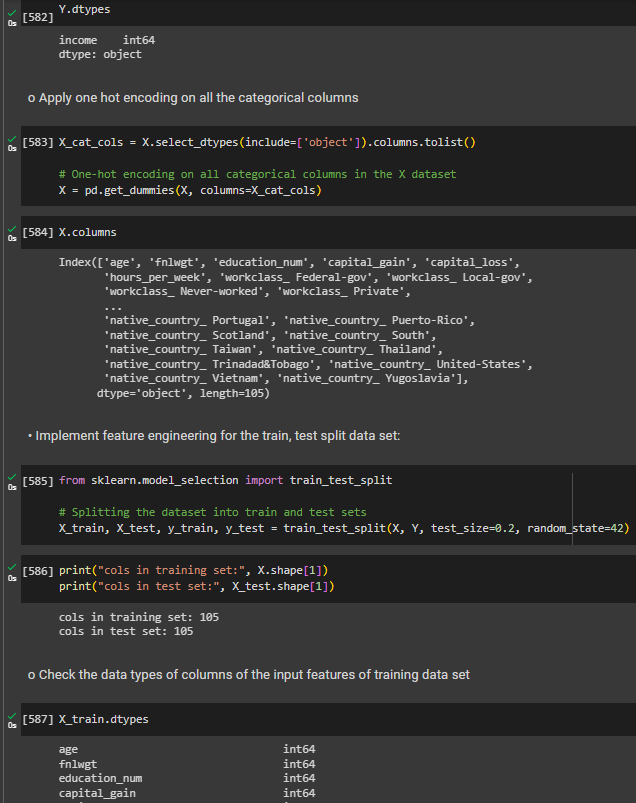
* **Source Code**

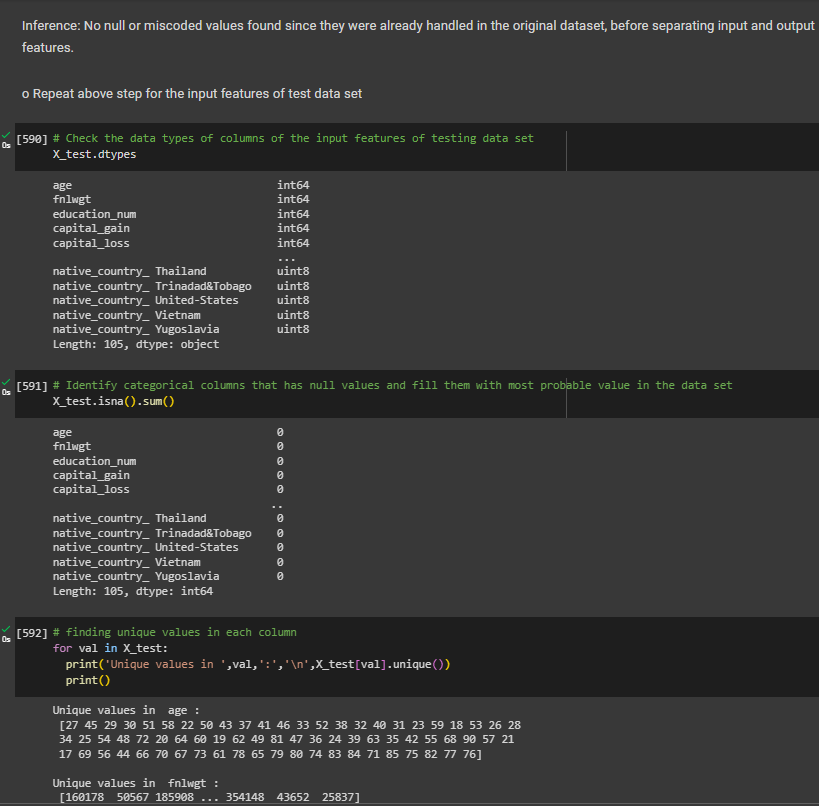
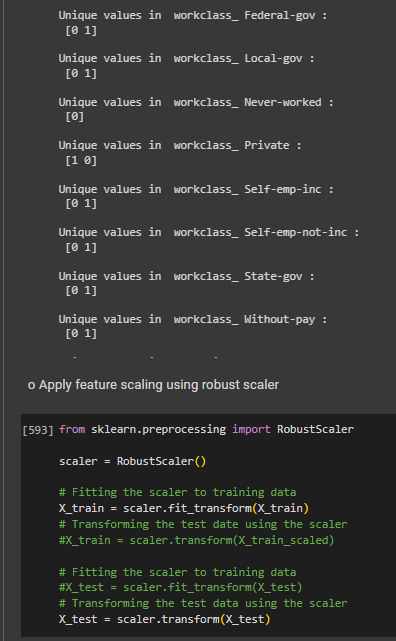
*"""  
 \* This file contains code snippets to performing dataset cleaning operations on adultPrac7.csv  
 \* ML-E7-Task2  
 \*  
 \* Original file is located at: https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY  
 \* @author Pratyush Kumar (github.com/pratyushgta)  
"""*import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
"""• Upload data set into the dataframe"""  
  
df = pd.read\_csv("adultPrac7.csv")  
  
df.head()  
  
"""• Check the shape of the data set."""  
  
df.shape  
  
"""• Find out all the categorical columns from the data set"""  
  
# Printing all columns  
df.columns  
  
# selecting categorical columns  
cat\_cols = df.select\_dtypes(include=['object']).columns.tolist()  
# Printing only categorical columns  
cat\_cols  
  
"""• Check if null values exist in all the categorical columns"""  
  
df[cat\_cols].isna().sum()  
  
"""• Identify the problems with “workclass”, “Occupation”,”native\_country” columns and rectify it."""  
  
# finding unique values in each column  
for val in cat\_cols:  
 print('Unique values in ', val, ':', '\n', df[val].unique())  
 print()  
  
df.dtypes  
  
# Replace '?' with null in categorical columns  
for val in cat\_cols:  
 df[val] = df[val].replace(' ?', pd.NA)  
  
# Check for null values in categorical columns  
df[cat\_cols].isna().sum()  
  
# Replace all null values with the mode of the respective column  
for col in cat\_cols:  
 df[col].fillna(df[col].mode()[0], inplace=True)  
  
df[cat\_cols].isna().sum()  
  
"""• Explore numeric columns and check any null values exist for the numeric columns."""  
  
# selecting all numeric columns  
num\_cols = df.select\_dtypes(include=['int64', 'float64']).columns.tolist()  
  
# Printing only categorical columns  
num\_cols  
  
# Check if null values exist in all the numeric columns  
df[num\_cols].isna().sum()  
  
# finding unique values in each column  
for val in num\_cols:  
 print('Unique values in ', val, ':', '\n', df[val].unique())  
 print()  
  
"""• Create a feature vector with x= all the columns except income and y=income"""  
  
# Selecting all columns except 'income' as features  
X = df.loc[:, df.columns != 'income']  
  
# Selecting 'income' as target variable  
Y = df[['income']]  
  
print(X.columns, '\n', Y.columns)  
  
Y.head()  
  
# Converting target variable Y into binary values by replacing '>50K' with 1 and '<=50K' with 0  
# because roc\_curve and roc\_auc\_score functions require binary values (0, 1) or (-1, 1)  
Y = Y.replace({' >50K': 1, ' <=50K': 0})  
  
Y.head()  
  
Y.dtypes  
  
"""o Apply one hot encoding on all the categorical columns"""  
  
X\_cat\_cols = X.select\_dtypes(include=['object']).columns.tolist()  
  
# One-hot encoding on all categorical columns in the X dataset  
X = pd.get\_dummies(X, columns=X\_cat\_cols)  
  
X.columns  
  
"""• Implement feature engineering for the train, test split data set:  
  
"""  
  
from sklearn.model\_selection import train\_test\_split  
  
# Splitting the dataset into train and test sets  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=42) # 0.2 42 or 0.33 125  
  
print("cols in training set:", X.shape[1])  
print("cols in test set:", X\_test.shape[1])  
  
"""o Check the data types of columns of the input features of training data set"""  
  
X\_train.dtypes  
  
"""o Identify categorical columns that has null values and fill them with most probable value in the data set"""  
  
X\_train.isna().sum()  
  
# finding unique values in each column  
for val in X\_train:  
 print('Unique values in ', val, ':', '\n', X\_train[val].unique())  
 print()  
  
"""Inference: No null or miscoded values found since they were already handled in the original dataset, before separating input and output features.  
  
o Repeat above step for the input features of test data set  
"""  
  
# Check the data types of columns of the input features of testing data set  
X\_test.dtypes  
  
# Identify categorical columns that has null values and fill them with most probable value in the data set  
X\_test.isna().sum()  
  
# finding unique values in each column  
for val in X\_test:  
 print('Unique values in ', val, ':', '\n', X\_test[val].unique())  
 print()  
  
"""o Apply feature scaling using robust scaler"""  
  
from sklearn.preprocessing import RobustScaler  
  
scaler = RobustScaler()  
  
# Fitting the scaler to training data  
X\_train = scaler.fit\_transform(X\_train)  
  
# Transforming the test data using the scaler  
X\_test = scaler.transform(X\_test)

* **Input/ Output**

****

****

****

****

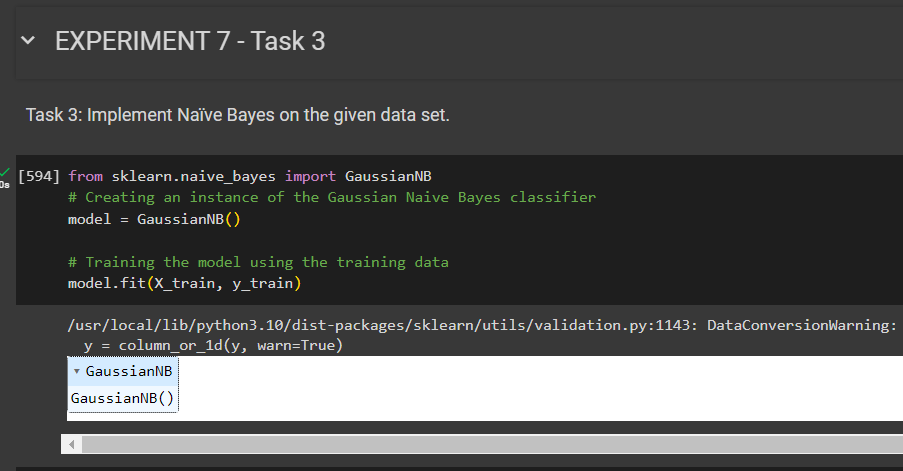
**B.3 Task 3**

**Colab link:** https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY?usp=sharing

* **Source Code**

*"""  
 \* This file contains code snippets to Implement Naïve Bayes on adultPrac7.csv  
 \* ML-E7-Task3  
 \*  
 \* Original file is located at: https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY  
 \* @author Pratyush Kumar (github.com/pratyushgta)  
"""*"""  
Task 3: Implement Naïve Bayes on the given data set.  
"""  
  
from sklearn.naive\_bayes import GaussianNB  
  
# Creating an instance of the Gaussian Naive Bayes classifier  
model = GaussianNB()  
  
# Training the model using the training data  
model.fit(X\_train, y\_train)  
  
# Predicting the output  
predicted = model.predict(X\_test)

* **Input/ Output**

****

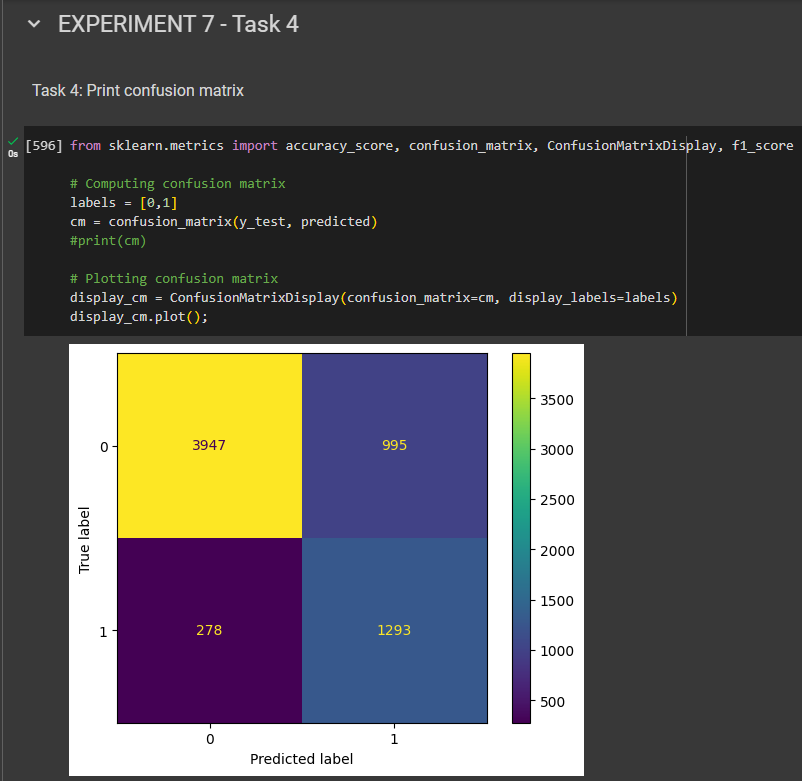
**B.4 Task 4**

**Colab link:** https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY?usp=sharing

* **Source Code**

*"""  
 \* This file contains code snippets to print confusion matrix (dataset used: adultPrac7.csv)  
 \* ML-E7-Task4  
 \*  
 \* Original file is located at: https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY  
 \* @author Pratyush Kumar (github.com/pratyushgta)  
"""*"""  
  
Task 4: Print confusion matrix  
"""  
  
from sklearn.metrics import accuracy\_score, confusion\_matrix, ConfusionMatrixDisplay, f1\_score  
  
# Computing confusion matrix  
labels = [0, 1]  
cm = confusion\_matrix(y\_test, predicted)  
# print(cm)  
  
# Plotting confusion matrix  
display\_cm = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=labels)  
display\_cm.plot();

* **Input/ Output**

****

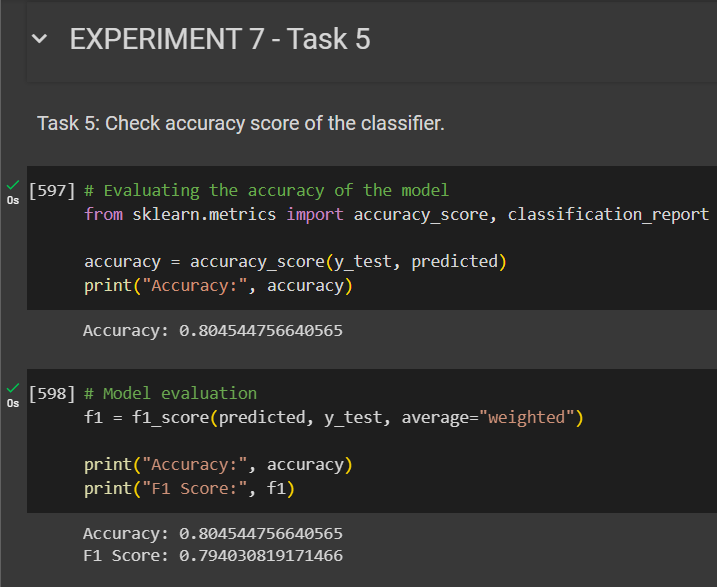
**B.5 Task 5**

**Colab link:** https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY?usp=sharing

* **Source Code**

*"""  
 \* This file contains code snippets to check accuracy score of Naïve Bayes classifier (adultPrac7.csv)  
 \* ML-E7-Task5  
 \*  
 \* Original file is located at: https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY  
 \* @author Pratyush Kumar (github.com/pratyushgta)  
"""*"""  
Task 5: Check accuracy score of the classifier.  
"""  
  
# Evaluating the accuracy of the model  
from sklearn.metrics import accuracy\_score, classification\_report  
  
accuracy = accuracy\_score(y\_test, predicted)  
print("Accuracy:", accuracy)  
  
# Model evaluation  
f1 = f1\_score(predicted, y\_test, average="weighted")  
  
print("Accuracy:", accuracy)  
print("F1 Score:", f1)

* **Input/ Output**

****

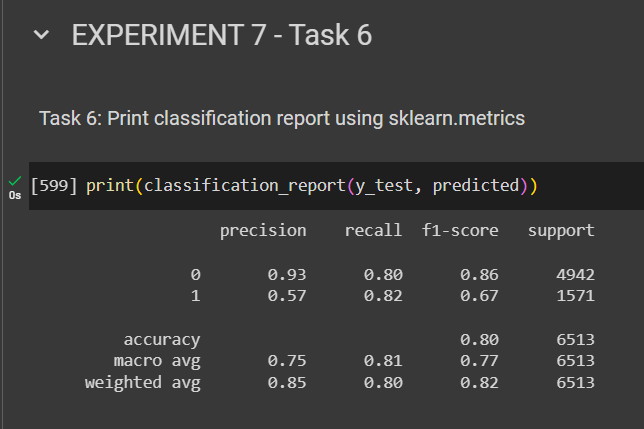
**B.6 Task 6**

**Colab link:** https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY?usp=sharing

* **Source Code**

*"""  
 \* This file contains code snippets to check accuracy score of Naïve Bayes classifier (adultPrac7.csv)  
 \* ML-E7-Task5  
 \*  
 \* Original file is located at: https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY  
 \* @author Pratyush Kumar (github.com/pratyushgta)  
"""*"""  
Task 5: Check accuracy score of the classifier.  
"""  
  
# Evaluating the accuracy of the model  
from sklearn.metrics import accuracy\_score, classification\_report  
  
accuracy = accuracy\_score(y\_test, predicted)  
print("Accuracy:", accuracy)  
  
# Model evaluation  
f1 = f1\_score(predicted, y\_test, average="weighted")  
  
print("Accuracy:", accuracy)  
print("F1 Score:", f1)

* **Input/ Output**

****

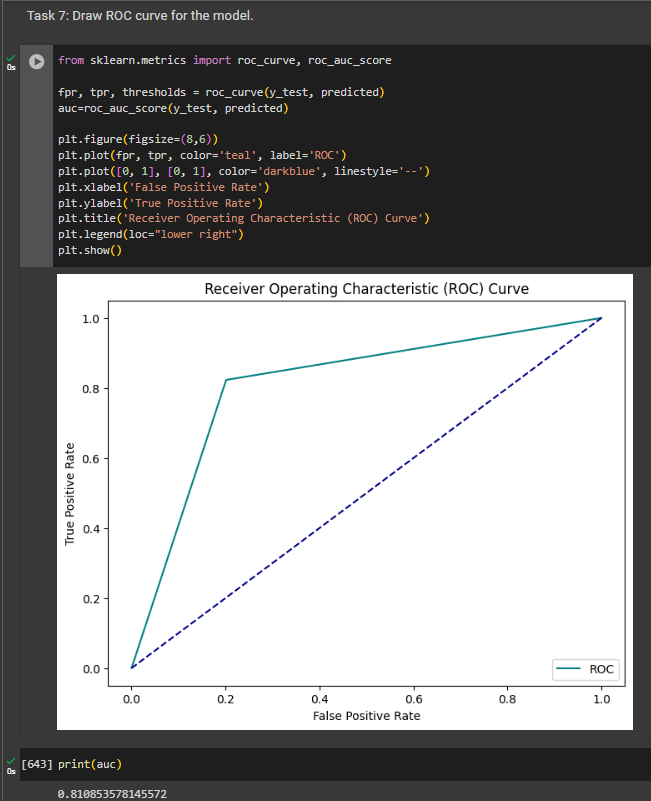
**B.7 Task 7**

**Colab link:** https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY?usp=sharing

* **Source Code**

*"""  
 \* This file contains code snippets to draw ROC curve for the Naïve Bayes classifier (adultPrac7.csv)  
 \* ML-E7-Task7  
 \*  
 \* Original file is located at: https://colab.research.google.com/drive/1R-YzFUNjbV6g85CR2NC0XHQ7FipvD7pY  
 \* @author Pratyush Kumar (github.com/pratyushgta)  
"""*"""  
Task 7: Draw ROC curve for the model.  
"""  
  
from sklearn.metrics import roc\_curve, roc\_auc\_score  
  
fpr, tpr, thresholds = roc\_curve(y\_test, predicted)  
auc = roc\_auc\_score(y\_test, predicted)  
  
plt.figure(figsize=(8, 6))  
plt.plot(fpr, tpr, color='teal', label='ROC')  
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')  
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
plt.title('Receiver Operating Characteristic (ROC) Curve')  
plt.legend(loc="lower right")  
plt.show()  
  
print(auc)

* **Input/ Output**

****

**B.8 Conclusion:**

*(Students must write the conclusion in their own words.)*

Performed data cleaning and implemented Naïve Bayes algorithm for classification and evaluated its performance through comparative analysis and interpretation of various classification metrics.