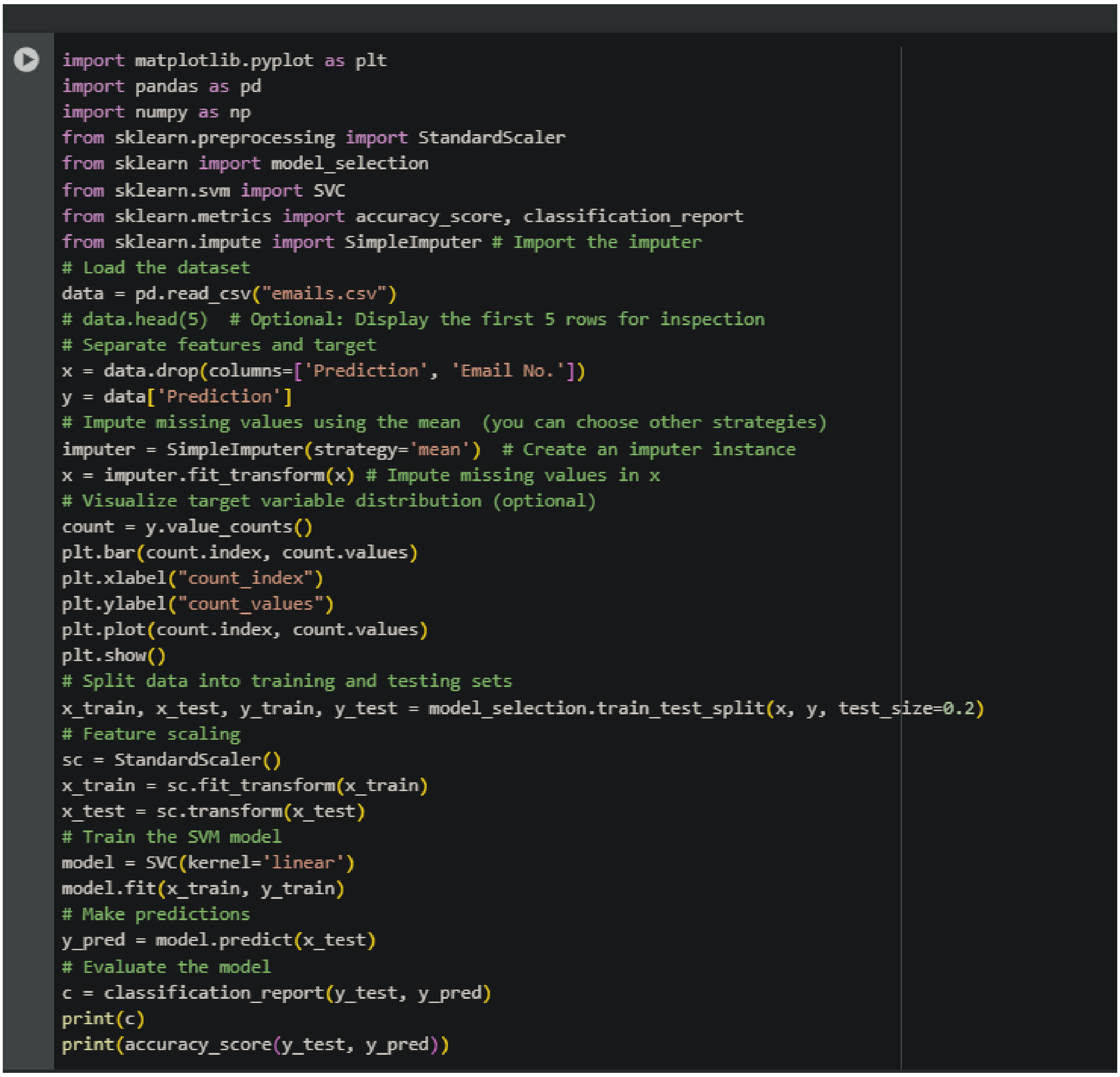
Machine Learning Lab -5

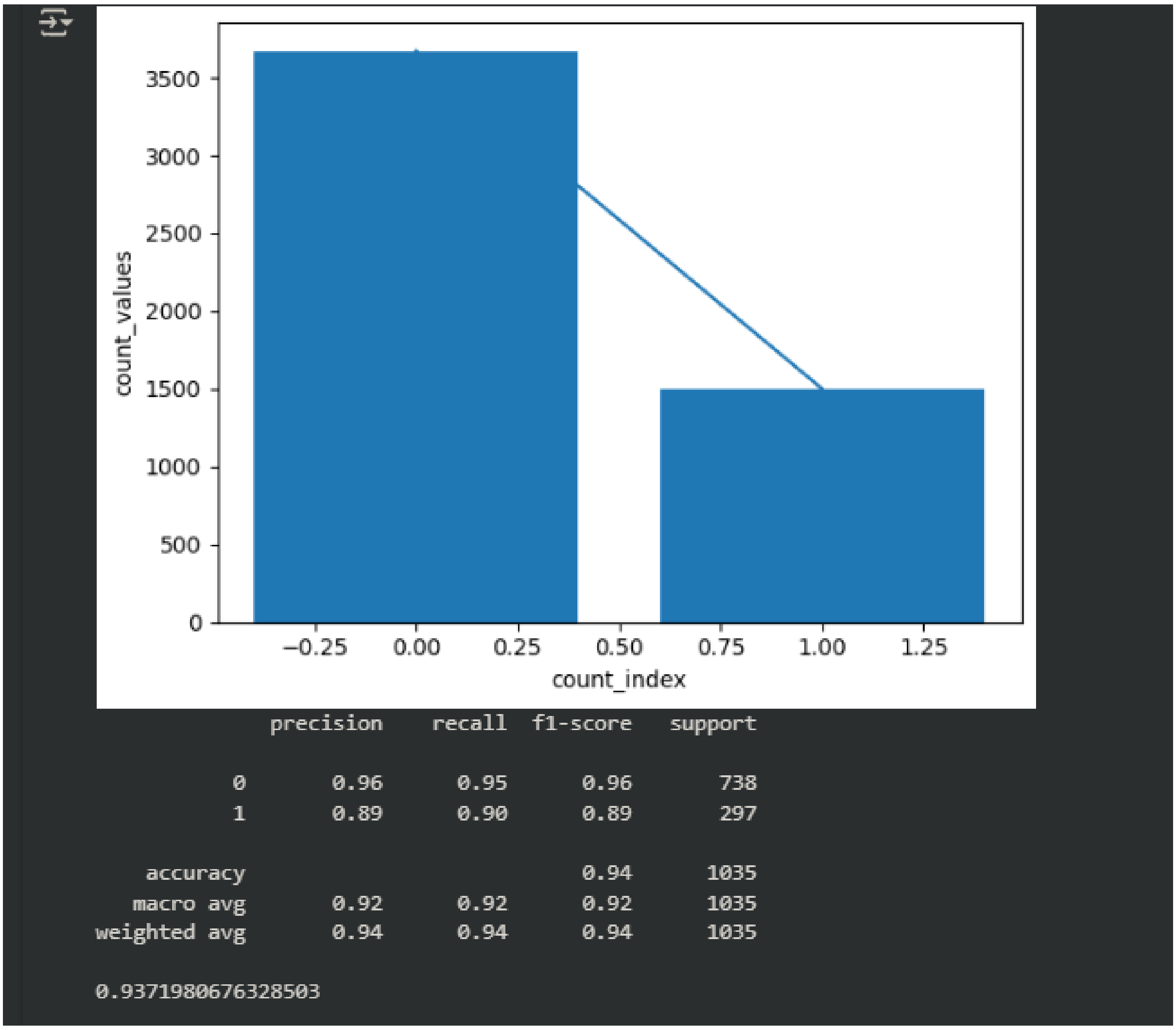
CB.SC.U4CYS23036

Praneesh R V



```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn import model_selection
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
from sklearn.impute import SimpleImputer # Import the imputer
# Load the dataset
data = pd.read_csv("emails.csv")
# data.head(5)  # Optional: Display the first 5 rows for inspection
# Separate features and target
x = data.drop(columns=['Prediction', 'Email No.'])
y = data['Prediction']
# Impute missing values using the mean  (you can choose other strategies)
imputer = SimpleImputer(strategy='mean')  # Create an imputer instance
x = imputer.fit_transform(x) # Impute missing values in x
# Visualize target variable distribution (optional)
count = y.value_counts()
plt.bar(count.index, count.values)
plt.xlabel("count_index")
plt.ylabel("count_values")
plt.plot(count.index, count.values)
plt.show()
# Split data into training and testing sets
x_train, x_test, y_train, y_test = model_selection.train_test_split(x, y, test_size=0.2)
# Feature scaling
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
# Train the SVM model
model = SVC(kernel='linear')
model.fit(x_train, y_train)
# Make predictions
y_pred = model.predict(x_test)
# Evaluate the model
c = classification_report(y_test, y_pred)
print(c)
print(accuracy_score(y_test, y_pred))
```

Output:



```
              precision    recall  f1-score   support

           0       0.96      0.95      0.96       738
           1       0.89      0.90      0.89       297

    accuracy                           0.94      1035
   macro avg       0.92      0.92      0.92      1035
weighted avg       0.94      0.94      0.94      1035

0.9371980676328503
```