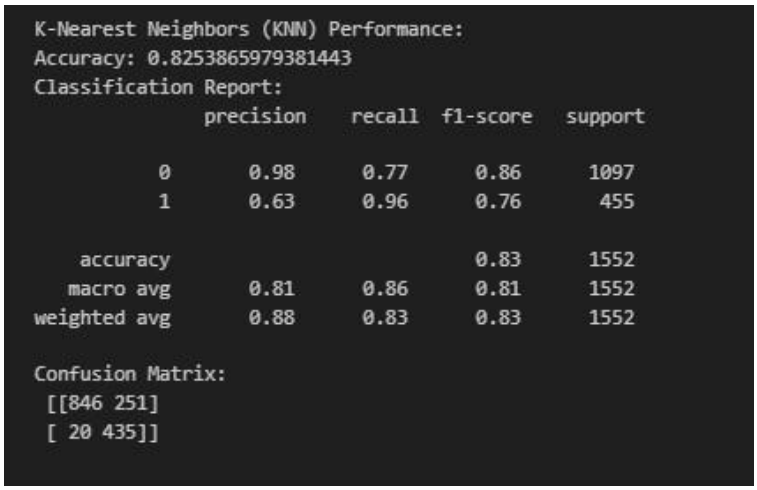
Assignment No : 2

```
import pandas as pd from sklearn.model_selection
 import train_test_split from sklearn.preprocessing
import StandardScaler from sklearn.neighbors
import KNeighborsClassifier from sklearn.svm
 import SVC from sklearn.metrics
 import classification_report, accuracy_score, confusion_matrix



df = pd.read_csv("emails.csv")
data = pd.read_csv("emails.csv")
X = data.iloc[:, 1:-1].values
y = data.iloc[:, -1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

scaler =StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

y_pred_knn = knn.predict(X_test)

svm = SVC(kernel='linear', random_state=42)
svm.fit(X_train, y_train)
```

```
y_pred_svm = svm.predict(X_test)
print("K-Nearest Neighbors (KNN) Performance:")
print(f"Accuracy: {accuracy_score(y_test, y_pred_knn)}")
print("Classification Report:\n", classification_report(y_test, y_pred_knn))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_knn))
```

output:




```
print("\nSupport Vector Machine (SVM) Performance:")

print(f"Accuracy: {accuracy_score(y_test, y_pred_svm)}")
print("Classification Report:\n", classification_report(y_test, y_pred_svm))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_svm)
```

output:

```
Support Vector Machine (SVM) Performance:
Accuracy: 0.9400773195876289
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.95      0.96      1097
           1       0.89      0.91      0.90       455

    accuracy                           0.94      1552
   macro avg       0.92      0.93      0.93      1552
weighted avg       0.94      0.94      0.94      1552

Confusion Matrix:
 [[1043   54]
 [  39  416]]
```

```
if accuracy_score(y_test, y_pred_knn) > accuracy_score(y_test,
y_pred_svm):
    print("\nKNN performed better.")
else:
    print("\nSVM performed better.")
```

output:

SVM performed better.