ML Lab 6

Praneesh R V
CB.SC.U4CYS23036

Procedure:
Load and read the data: The dataset is loaded using pandas for easy data manipulation.
Split the data: The data is split into training and testing sets, ensuring that the model is trained on one portion of the data and evaluated on another.
Normalize the data: StandardScaler is used to normalize the feature values, bringing them to a common scale with mean 0 and standard deviation 1.
Train models: Three machine learning models—SVM (Support Vector Machine), KNN (K-Nearest Neighbors), and Naive Bayes—are trained on the normalized training data.
Evaluate models: The models are evaluated on the test data using several performance metrics:
Accuracy
F1 Score
Precision
Recall
Compare models: A plot is created to visually compare the performance of the models across the evaluation metrics (Accuracy, F1 Score, Precision, and Recall).

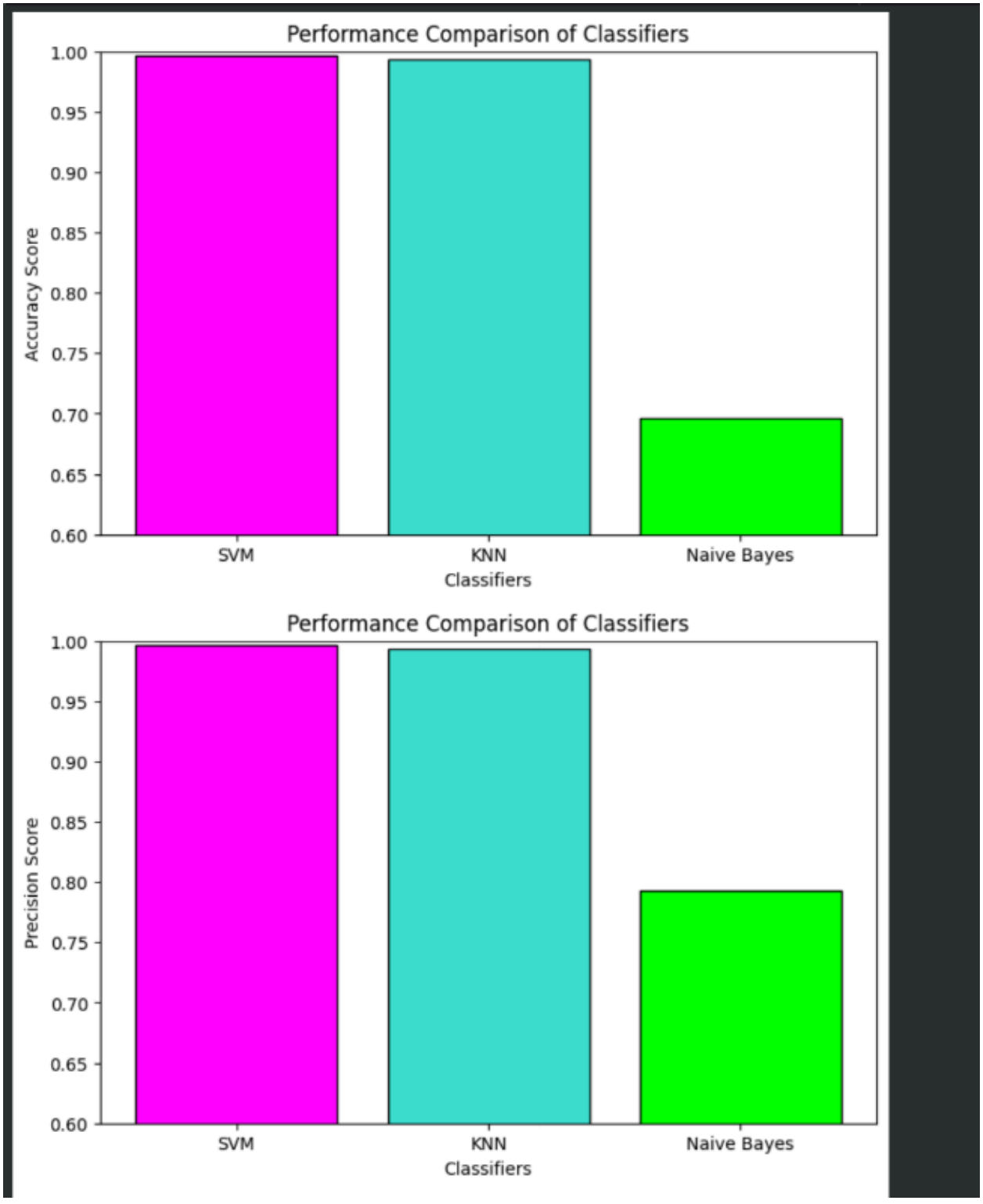
```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler as ss
from sklearn.metrics import accuracy_score, classification_report,precision_score,f1_score,recall_score
from sklearn.model_selection import train_test_split as tt
from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier as knn
from sklearn.naive_bayes import GaussianNB as gnb

data = pd.read_csv('all_data_3_.csv')
x = data.drop('class', axis=1)
y = data['class']
sc = ss()
x_scaled = sc.fit_transform(x)
x = pd.DataFrame(x_scaled, columns=x.columns)
x_train, x_test, y_train, y_test = tt(x, y, test_size=0.2, random_state=42)
model = SVC(kernel='linear')
model.fit(x_train, y_train)
y_pred = model.predict(x_test)

print("SVM :")
svm_accuracy = accuracy_score(y_test, y_pred)
svm_precision = precision_score(y_test, y_pred,average='weighted')
svm_recall = recall_score(y_test, y_pred,average='weighted')
svm_f1 = f1_score(y_test, y_pred,average='weighted')
print(classification_report(y_test, y_pred))


print("K nearest neighbors:")
knn=knn(n_neighbors=5)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
knn_accuracy = accuracy_score(y_test, y_pred)
knn_precision = precision_score(y_test, y_pred,average='weighted')
knn_recall = recall_score(y_test, y_pred,average='weighted')
knn_f1 = f1_score(y_test, y_pred,average='weighted')
print(classification_report(y_test,y_pred))
```

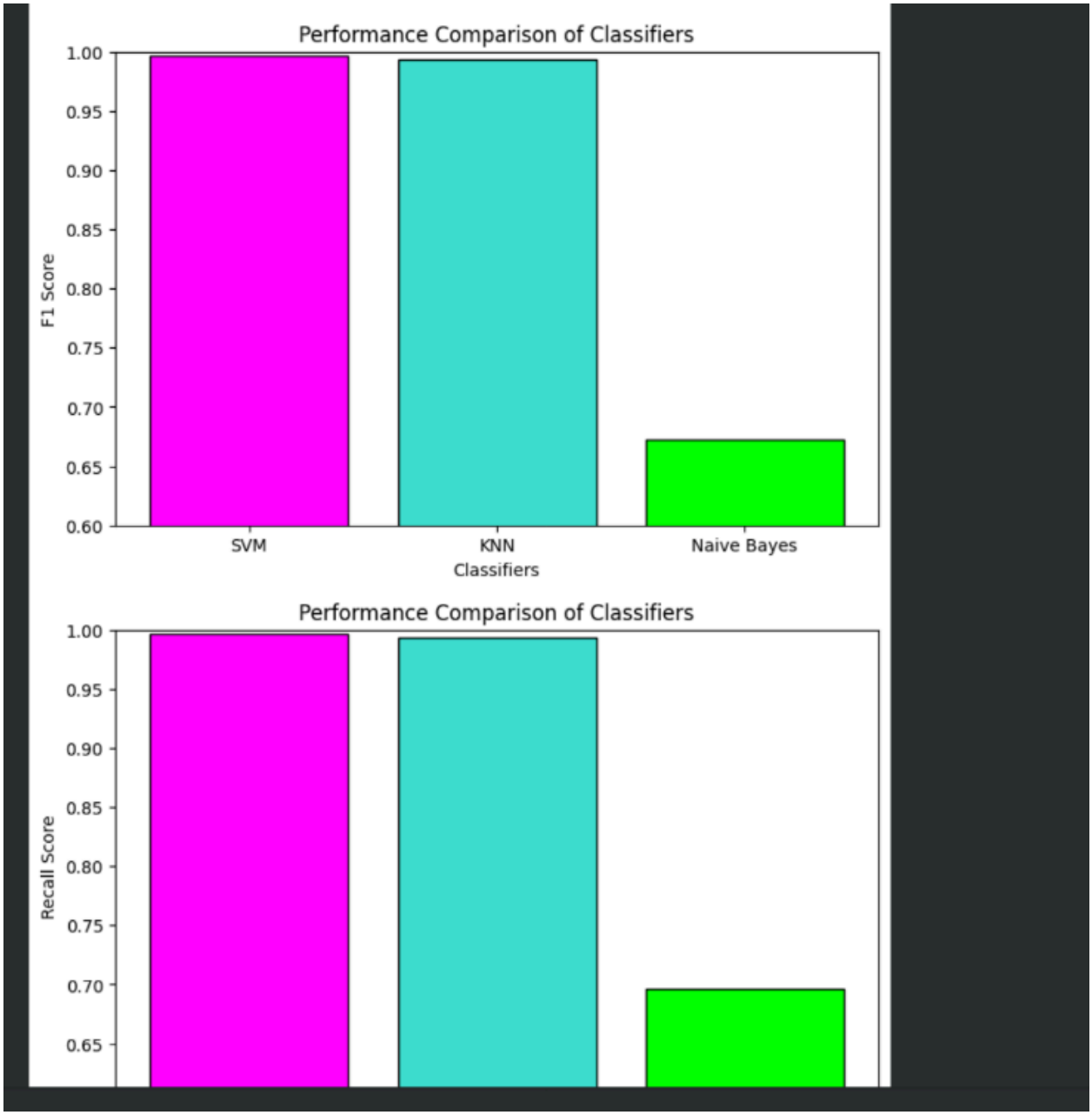
```
print("Navie Bayes:")
nb=gnb()
nb.fit(x_train,y_train)
y_pred=nb.predict(x_test)
nb_accuracy = accuracy_score(y_test, y_pred)
nb_precision = precision_score(y_test, y_pred,average='weighted')
nb_recall = recall_score(y_test, y_pred,average='weighted')
nb_f1 = f1_score(y_test, y_pred,average='weighted')
print(classification_report(y_test,y_pred))

# Plotting the accuracy comparison
models = ['SVM', 'KNN', 'Naive Bayes']
accuracies = [svm_accuracy, knn_accuracy, nb_accuracy]
plt.figure(figsize=(8, 5))
plt.bar(models, accuracies, color=['magenta', 'turquoise', 'lime'],edgecolor='black', linewidth=1)
plt.xlabel("Classifiers")
plt.ylabel("Accuracy Score")
plt.ylim(0.6, 1)
plt.title("Performance Comparison of Classifiers")
plt.show()

#precision comparison
models = ['SVM', 'KNN', 'Naive Bayes']
accuracies = [svm_precision, knn_precision, nb_precision]
plt.figure(figsize=(8, 5))
plt.bar(models, accuracies, color=['magenta', 'turquoise', 'lime'],edgecolor='black', linewidth=1)
plt.xlabel("Classifiers")
plt.ylabel("Precision Score")
plt.ylim(0.6, 1)
plt.title("Performance Comparison of Classifiers")
plt.show()
```

```
#f1 comparison
models = ['SVM', 'KNN', 'Naive Bayes']
accuracies = [svm_f1, knn_f1, nb_f1]
plt.figure(figsize=(8, 5))
plt.bar(models, accuracies, color=['magenta', 'turquoise', 'lime'],edgecolor='black', linewidth=1)
plt.xlabel("Classifiers")
plt.ylabel("F1 Score")
plt.ylim(0.6, 1)
plt.title("Performance Comparison of Classifiers")
plt.show()

#recall comparison
models = ['SVM', 'KNN', 'Naive Bayes']
accuracies = [svm_recall, knn_recall, nb_recall]
plt.figure(figsize=(8, 5))
plt.bar(models, accuracies, color=['magenta', 'turquoise', 'lime'],edgecolor='black', linewidth=1)
plt.xlabel("Classifiers")
plt.ylabel("Recall Score")
plt.ylim(0.6, 1)
plt.title("Performance Comparison of Classifiers")
plt.show()
```

Output:

```
SVM :
              precision    recall  f1-score   support

  bruteForce       1.00      1.00      1.00        37
   httpFlood       1.00      0.98      0.99        98
   icmp-echo       0.99      1.00      1.00       130
      normal       1.00      1.00      1.00       134
   slowloris       1.00      1.00      1.00       160
    slowpost       1.00      1.00      1.00        99
     tcp-syn       0.99      0.99      0.99       190
   udp-flood       1.00      1.00      1.00       152

    accuracy                           1.00      1000
   macro avg       1.00      1.00      1.00      1000
weighted avg       1.00      1.00      1.00      1000

K nearest neighbors:
              precision    recall  f1-score   support

  bruteForce       1.00      1.00      1.00        37
   httpFlood       1.00      0.97      0.98        98
   icmp-echo       0.98      1.00      0.99       130
      normal       1.00      0.99      0.99       134
   slowloris       0.98      1.00      0.99       160
    slowpost       1.00      1.00      1.00        99
     tcp-syn       1.00      1.00      1.00       190
   udp-flood       1.00      0.99      0.99       152

    accuracy                           0.99      1000
   macro avg       0.99      0.99      0.99      1000
weighted avg       0.99      0.99      0.99      1000

Navie Bayes:
              precision    recall  f1-score   support

  bruteForce       0.35      1.00      0.52        37
   httpFlood       1.00      0.87      0.93        98
   icmp-echo       0.61      0.74      0.67       130
      normal       0.88      0.79      0.83       134
   slowloris       0.62      1.00      0.77       160
    slowpost       0.65      1.00      0.79        99
     tcp-syn       0.94      0.43      0.59       190
   udp-flood       0.94      0.20      0.34       152

    accuracy                           0.70      1000
   macro avg       0.75      0.75      0.68      1000
weighted avg       0.79      0.70      0.67      1000
```


Performance Comparison of Classifiers
Accuracy Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
SVM
KNN
Naive Bayes
Classifiers
Performance Comparison of Classifiers
Precision Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
SVM
KNN
Naive Bayes
Classifiers


Performance Comparison of Classifiers
F1 Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
SVM
KNN
Naive Bayes
Classifiers
Performance Comparison of Classifiers
Recall Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65