Practical – 6

Aim: Write a program to implement K-Nearest Neighbors.

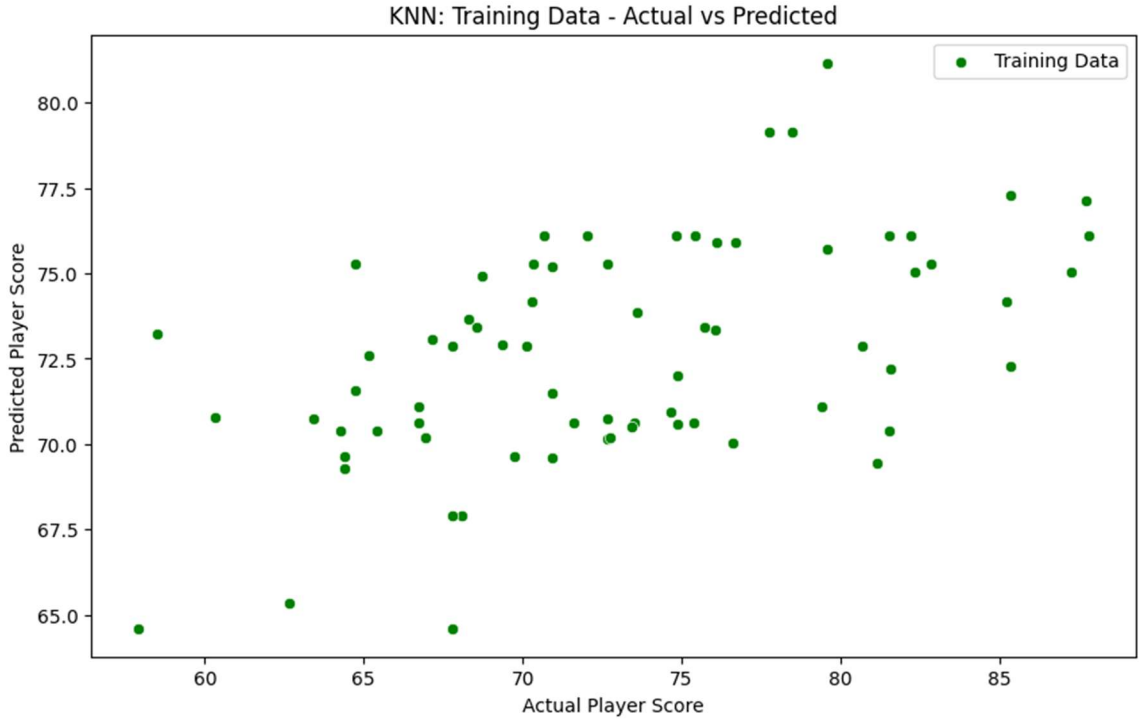
- Code:

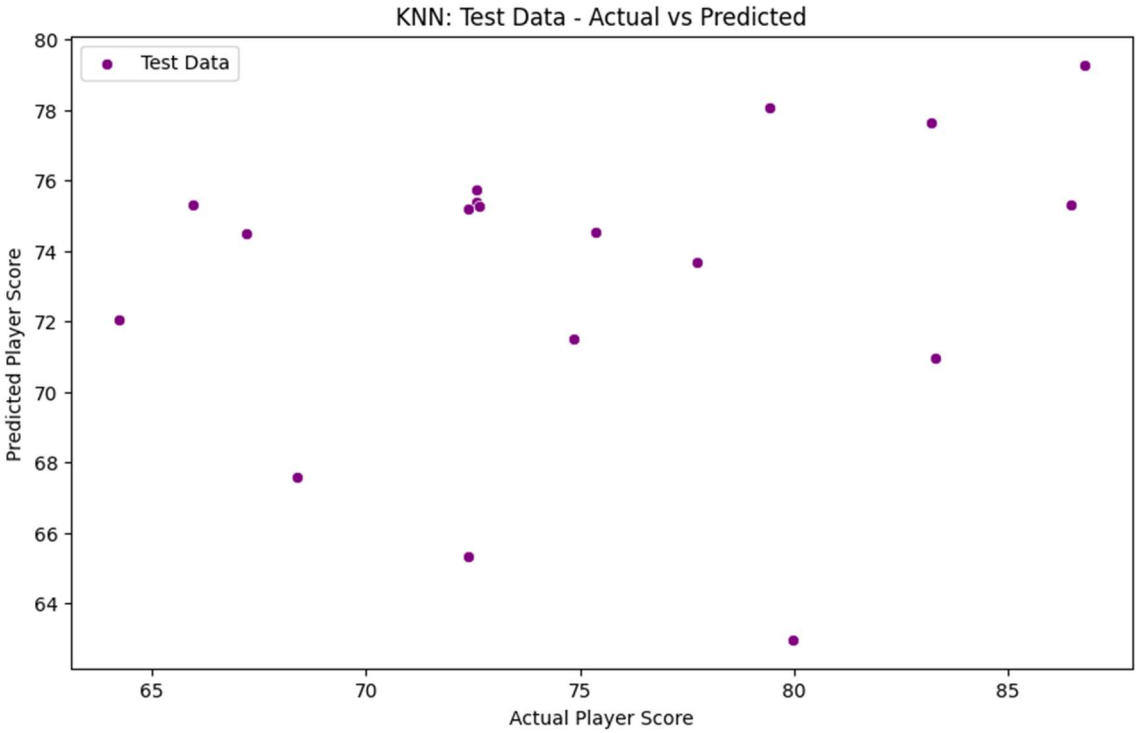
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, r2_score

data_set = pd.read_csv('/content/drive/MyDrive/temp/practical_4_2.csv')
print(data_set)

X = data_set[['Match_Duration', 'Loot_Collected', 'Enemies_Defeated']]
y = data_set['Player_Score']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

knn_model = KNeighborsRegressor(n_neighbors=3)
knn_model.fit(X_train, y_train)
y_pred = knn_model.predict(X_test)

print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))

plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_train, y=knn_model.predict(X_train), color='green', label='Training Data')
plt.xlabel('Actual Player Score')
plt.ylabel('Predicted Player Score')
plt.title('KNN: Training Data - Actual vs Predicted')
plt.legend()
plt.show()

plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred, color='purple', label='Test Data')
plt.xlabel('Actual Player Score')
plt.ylabel('Predicted Player Score')
plt.title('KNN: Test Data - Actual vs Predicted')
plt.legend()
plt.show()
```

- Output

	Player_Score	Match_Duration	Loot_Collected	Enemies_Defeated
0	79.967142	4.404851	0.683368	6
1	73.617357	10.213946	5.097842	10
2	78.476885	2.121969	1.616006	3
3	87.230299	1.274597	0.538544	17
4	72.658466	1.381719	2.487649	19
..	...	...	...	...
85	67.803925	6.324675	3.405740	6
86	72.393613	3.927831	2.612654	7
87	66.728510	2.152900	3.038517	8
88	66.937696	1.583439	3.139229	4
89	81.515035	4.283136	3.290001	13

[90 rows x 4 columns]
Mean Squared Error: 53.78721784298837
R^2 Score: -0.22632677170050797





Faculty Signature: _____ Date: _____