# **Practical – 5**

**Aim: Write a program to implement Linear Regression**

* Code:

from os import XATTR\_SIZE\_MAX

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

import seaborn as sns

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']

print(X, y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print(X\_train, X\_test, y\_train, y\_test)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

plt.figure(figsize=(10, 6))

sns.scatterplot(x=y\_train, y=model.predict(X\_train), color='blue', label='Training Data')

plt.xlabel('Actual Player Score')

plt.ylabel('Predicted Player Score')

plt.title('Training Data: Actual vs Predicted Player Score')

plt.legend()

plt.show()

plt.figure(figsize=(10, 6))

sns.scatterplot(x=y\_test, y=y\_pred, color='orange', label='Test Data Predictions')

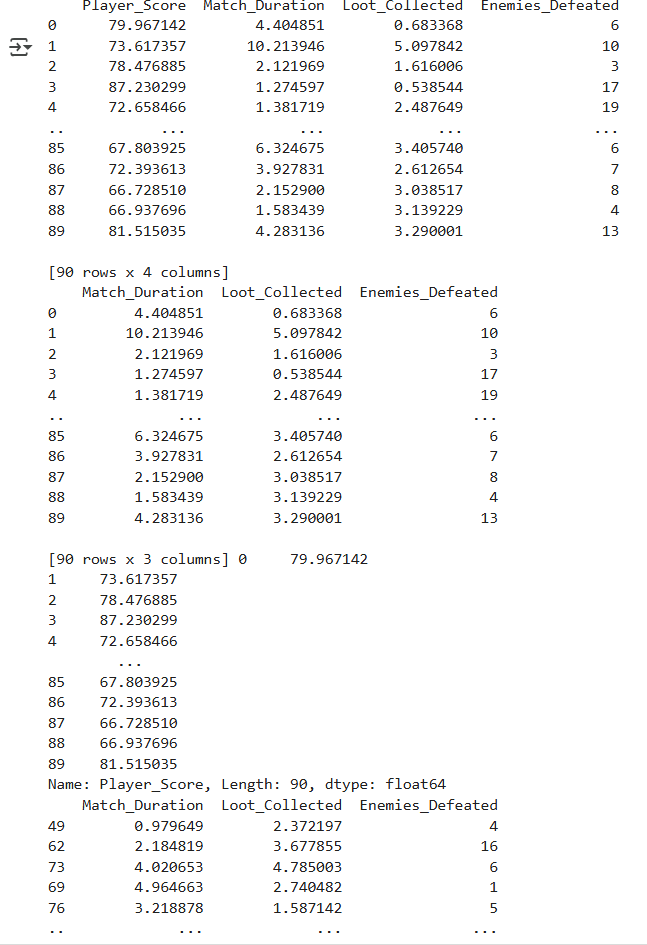
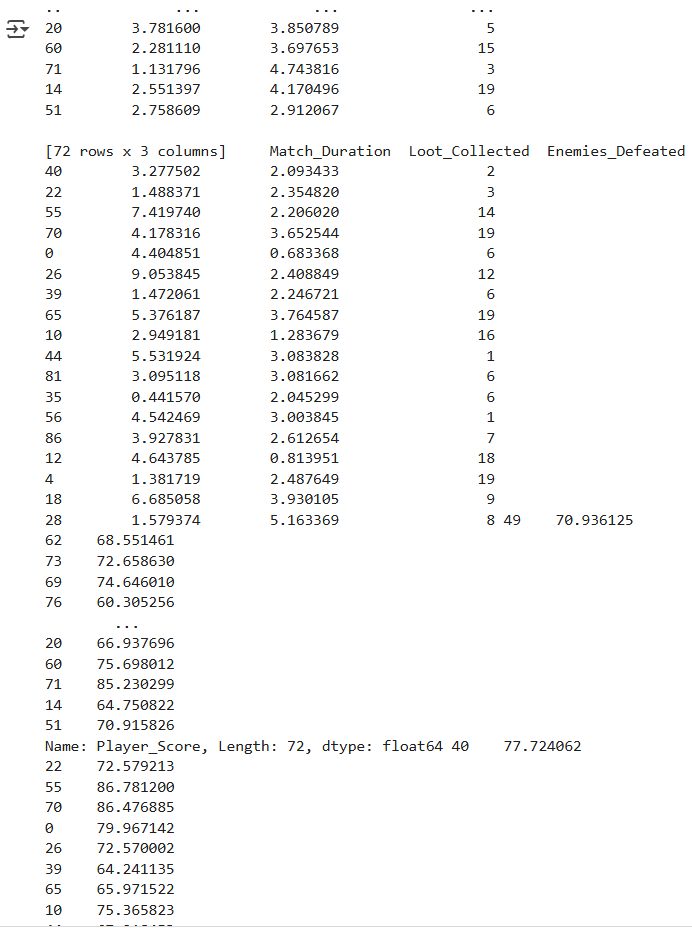
plt.xlabel('Actual Player Score')

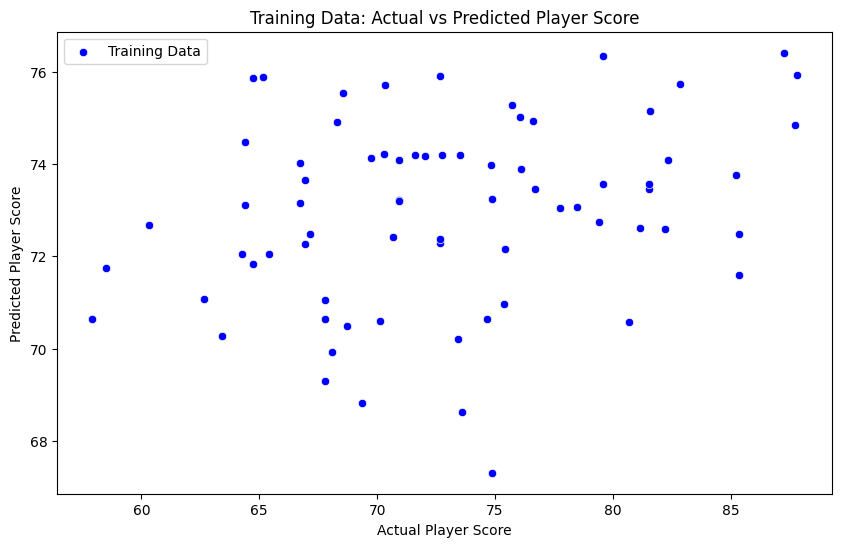
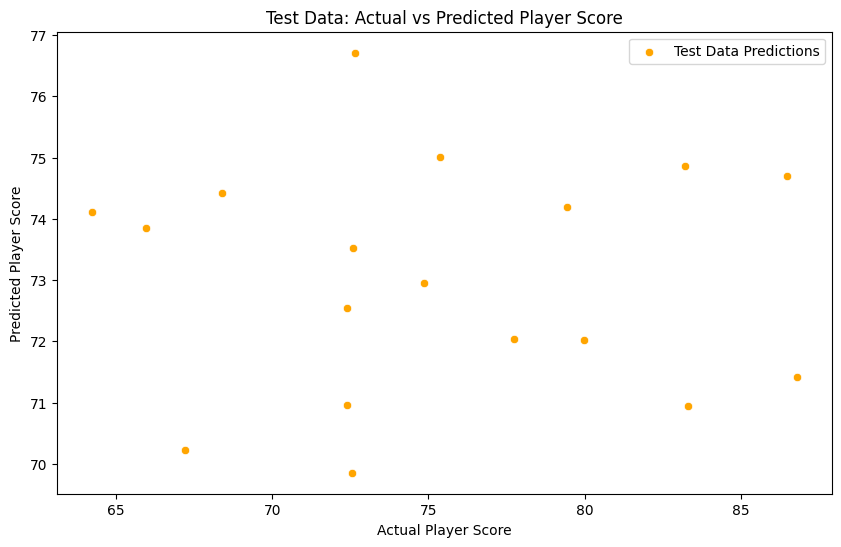
plt.ylabel('Predicted Player Score')

plt.title('Test Data: Actual vs Predicted Player Score')

plt.legend()

plt.show()

* Output:



Faculty Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_