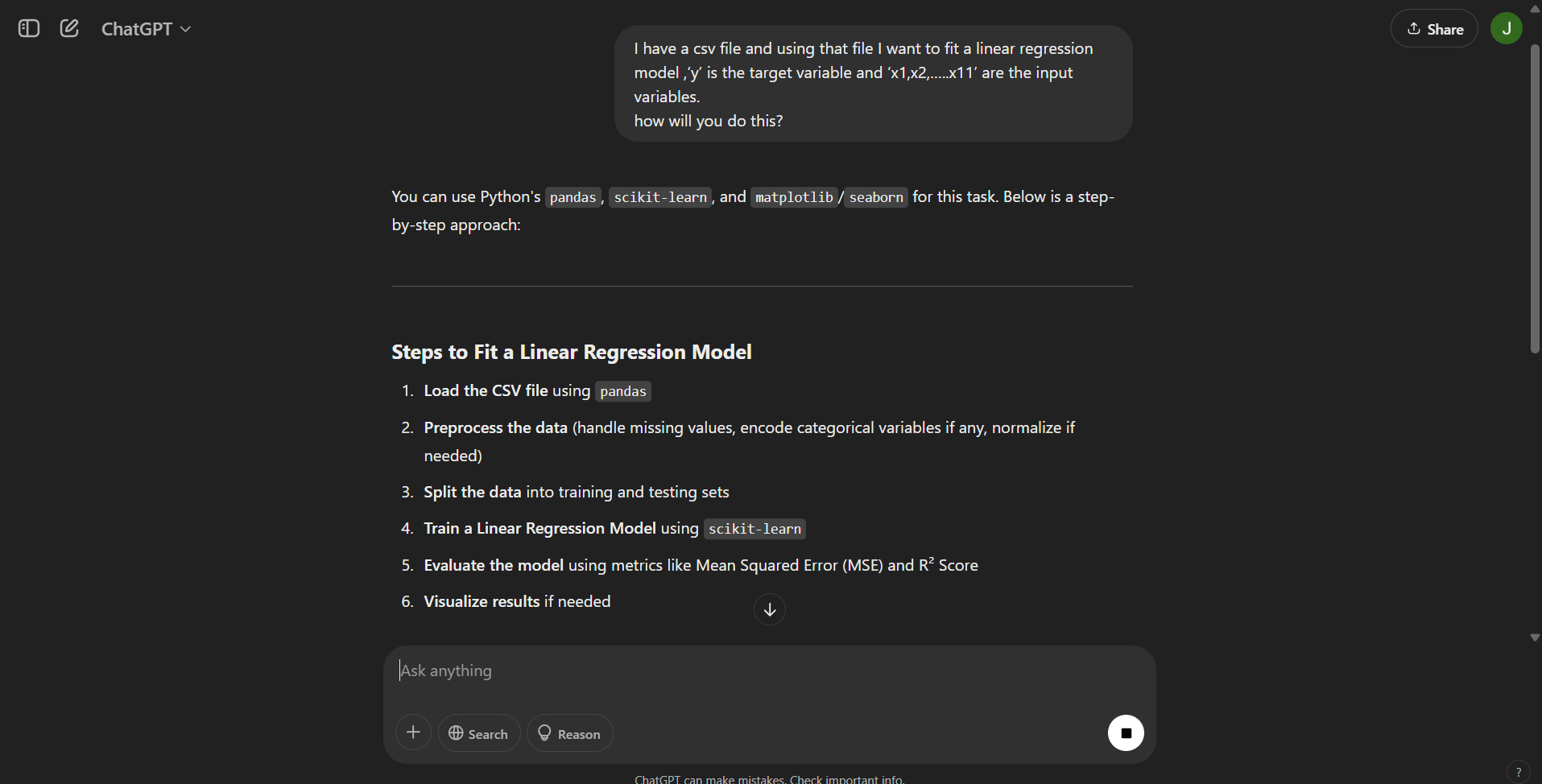
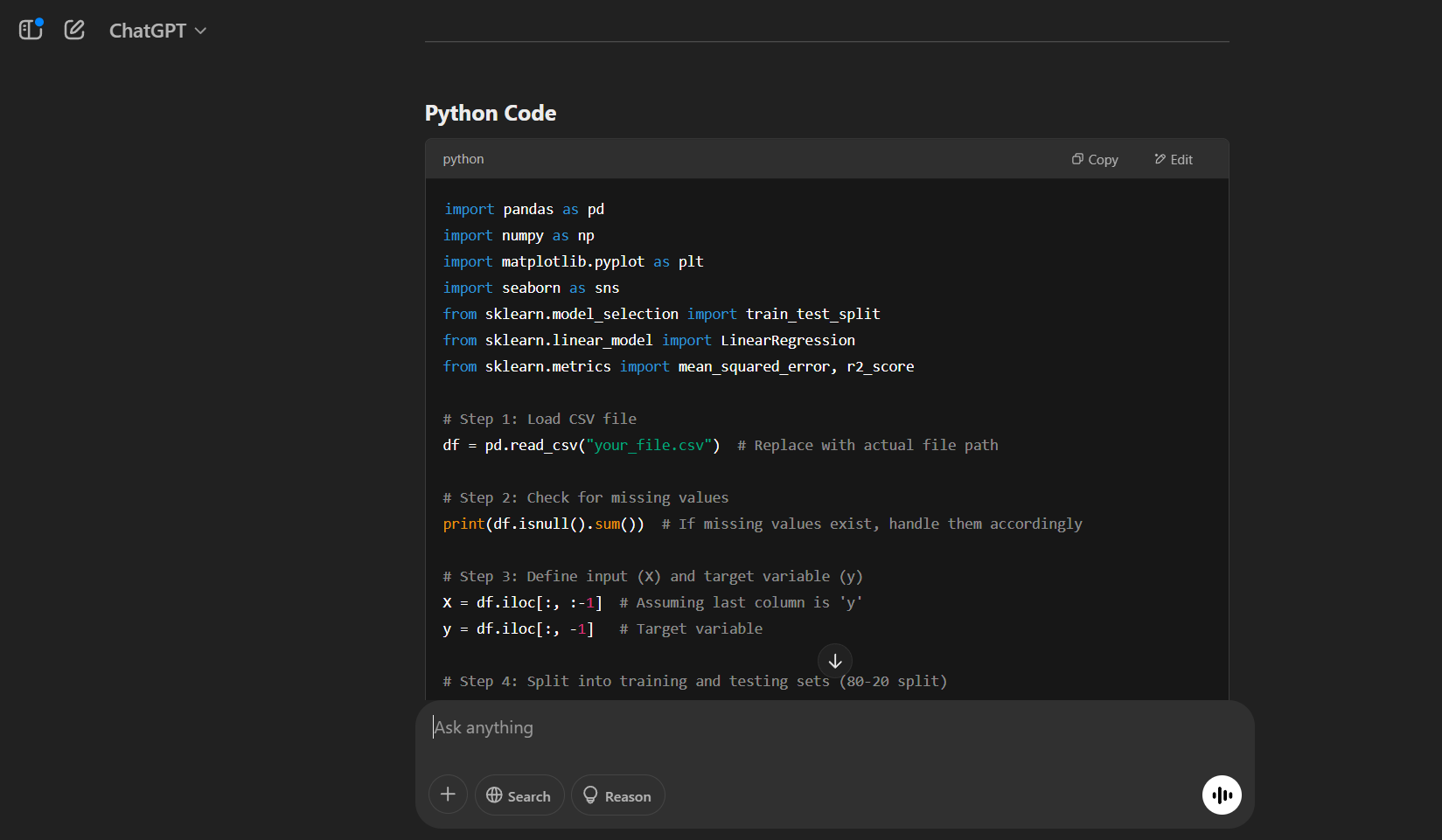
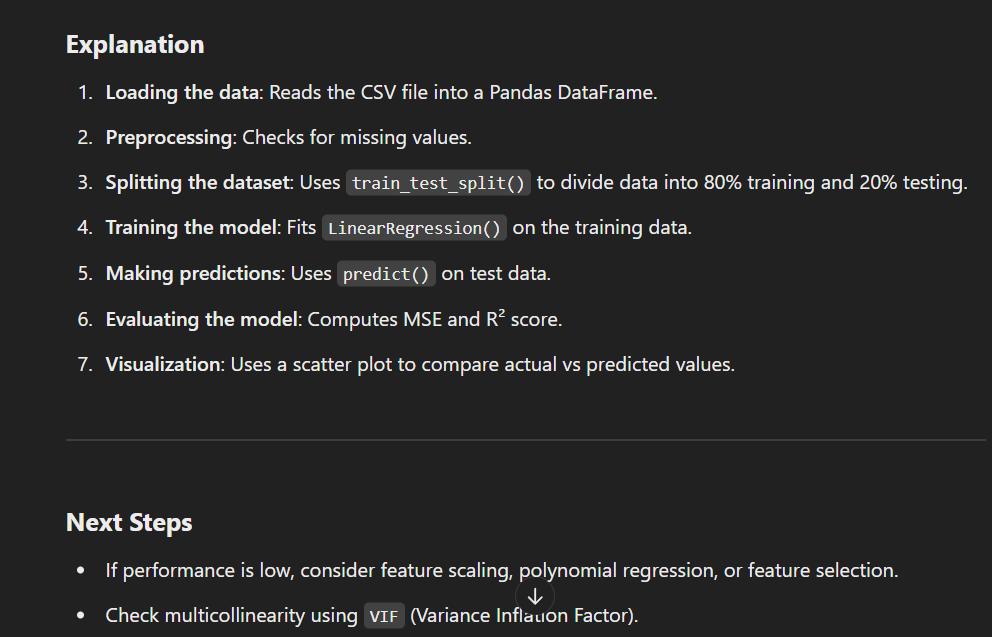
**Name: Jasmine Sayyad**

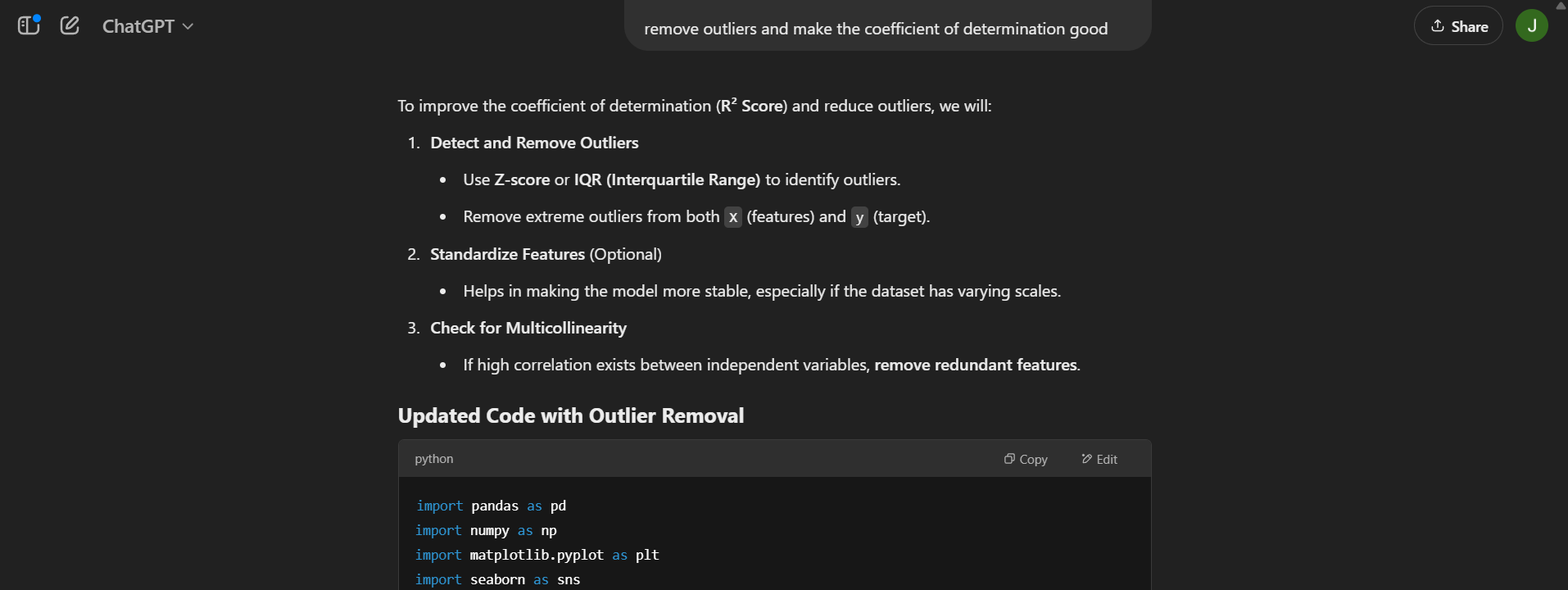
**PRN: 22510033**

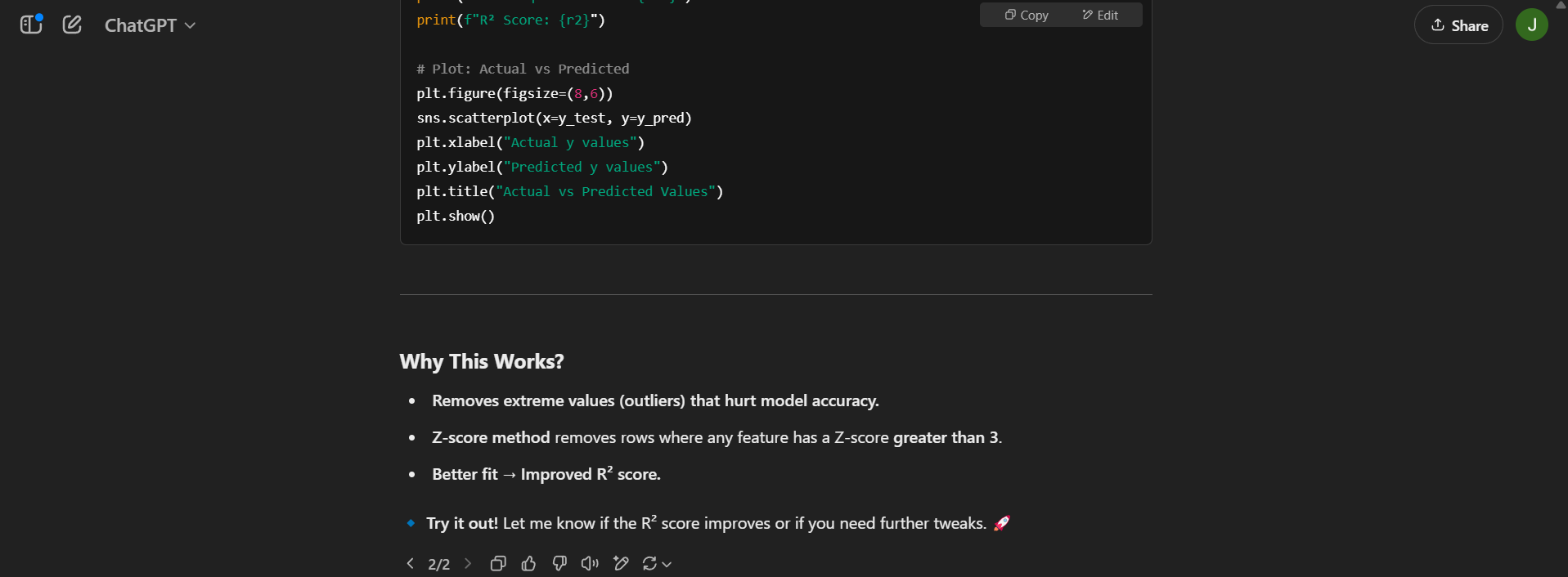
You have been provided data in the CSV file. Use this file to fit a linear regression model using ChatGPT alone. ‘y’ is the target variable and ‘x1,x2,…..x11’ are the input variables. Please take a screenshot of your prompts and chatGPT’s responses and submit in a word document or ppt. 10\*coefficient of determination will be your score. Copying of prompts will lead to assignment disqualification.









  
  
code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from scipy.stats import zscore  # Import Z-score function

# Load CSV file

df = pd.read\_csv("linear\_regression\_3.csv")  # Replace with actual file path

# Check for missing values

print(df.isnull().sum())

# Ensure X is a DataFrame and y is a Series

y = df.iloc[:, 0]        # First column is 'y' (Series)

X = df.iloc[:, 1:].copy()  # Rest are features (DataFrame)

# ---- Step 1: Remove Outliers using Z-score ----

df\_clean = df[(np.abs(zscore(df)) < 3).all(axis=1)]  # Keep only rows with Z < 3

# Extract cleaned X and y

y = df\_clean.iloc[:, 0]

X = df\_clean.iloc[:, 1:].copy()

# Train-test split (80-20)

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

# Train Linear Regression Model

model = LinearRegression()

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

# Make Predictions

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

# Model Evaluation

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}")

print(f"R² Score: {r2}")

# Plot: Actual vs Predicted

plt.figure(figsize=(8,6))

sns.scatterplot(x=y\_test, y=y\_pred)

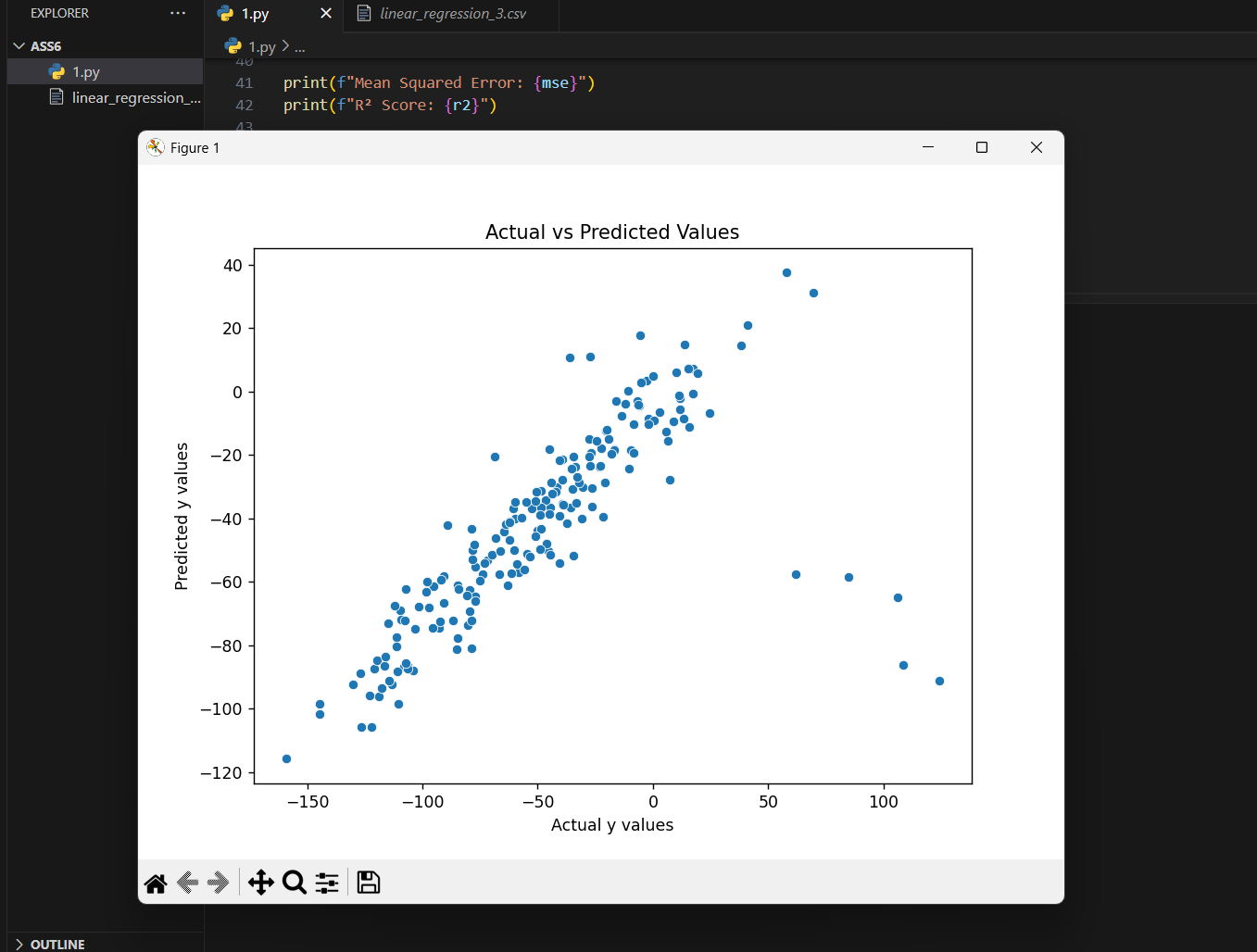
plt.xlabel("Actual y values")

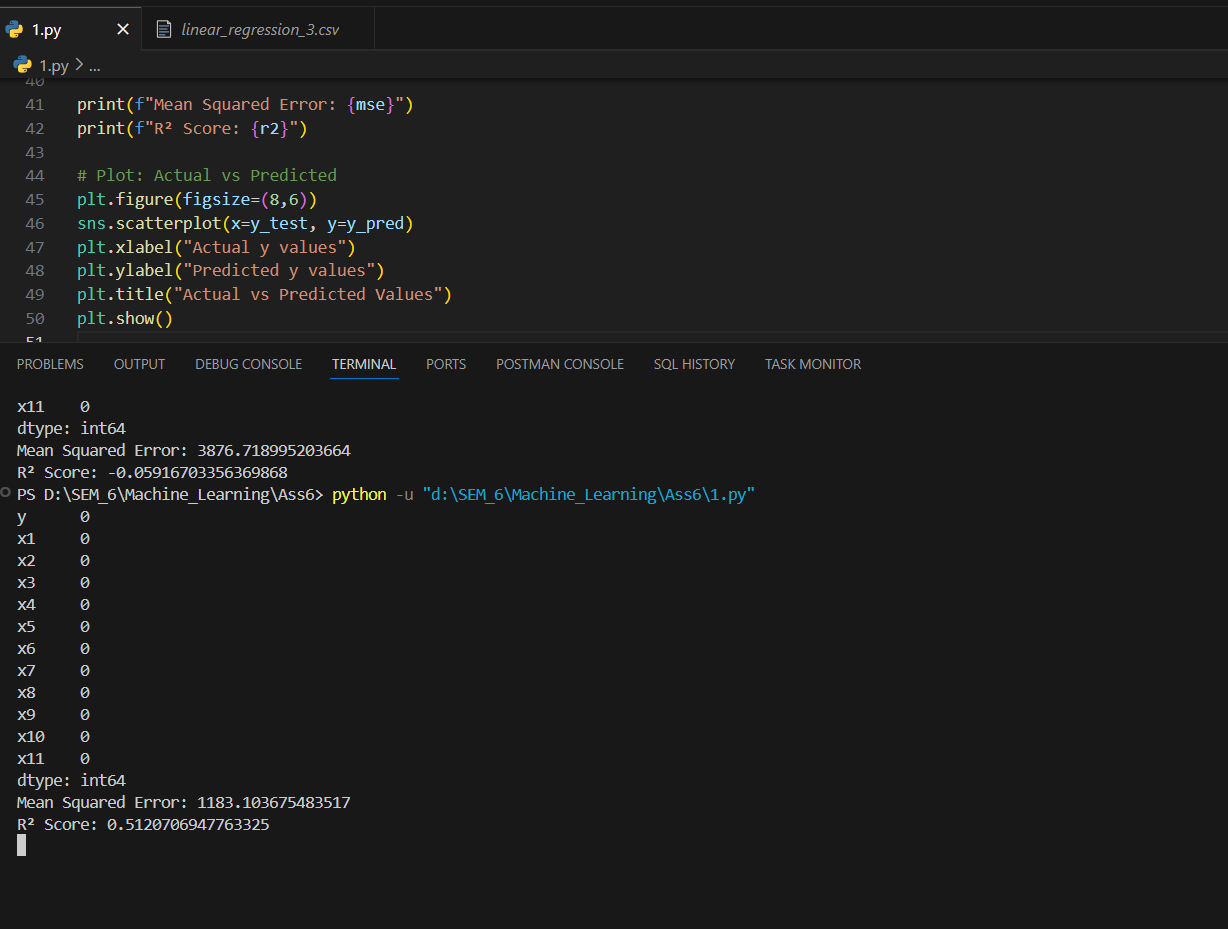
plt.ylabel("Predicted y values")

plt.title("Actual vs Predicted Values")

plt.show()

Output of code





coefficient of determination is => 0.5121670