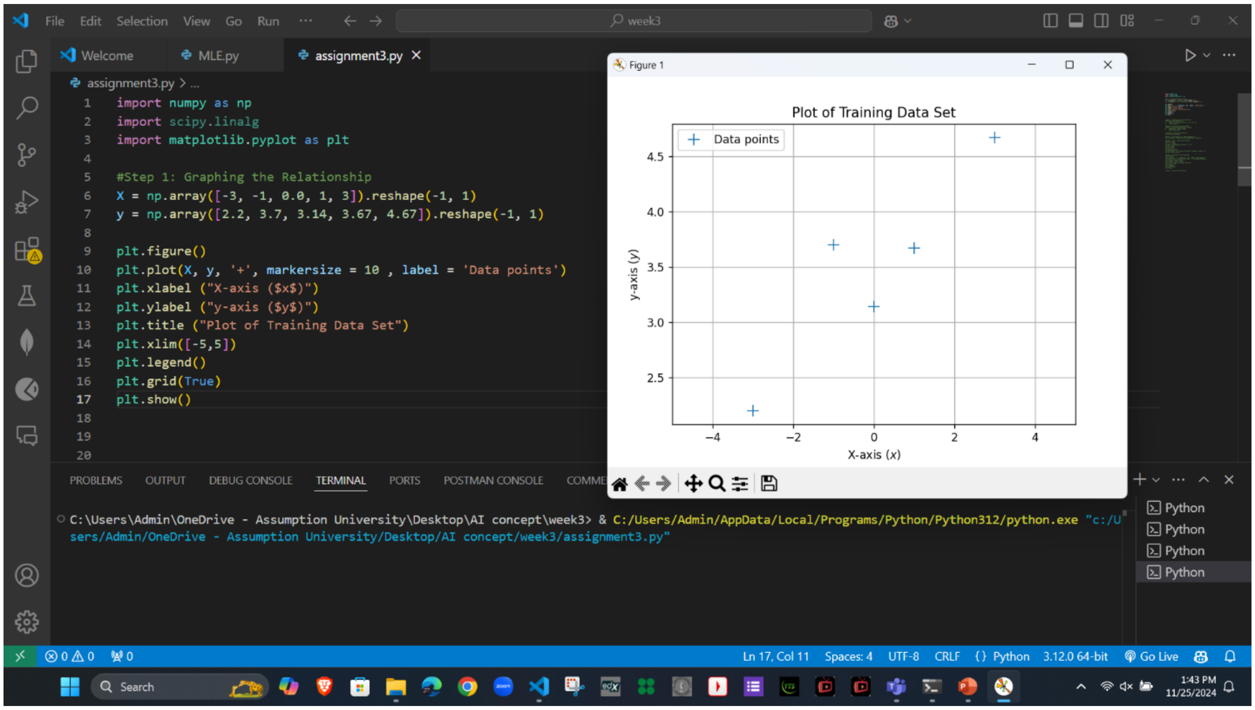
# Assignment 3: Maximum Likelihood Estimation

## Step 1: Graphing the Relationship

- Python code and relationship between x and y graph

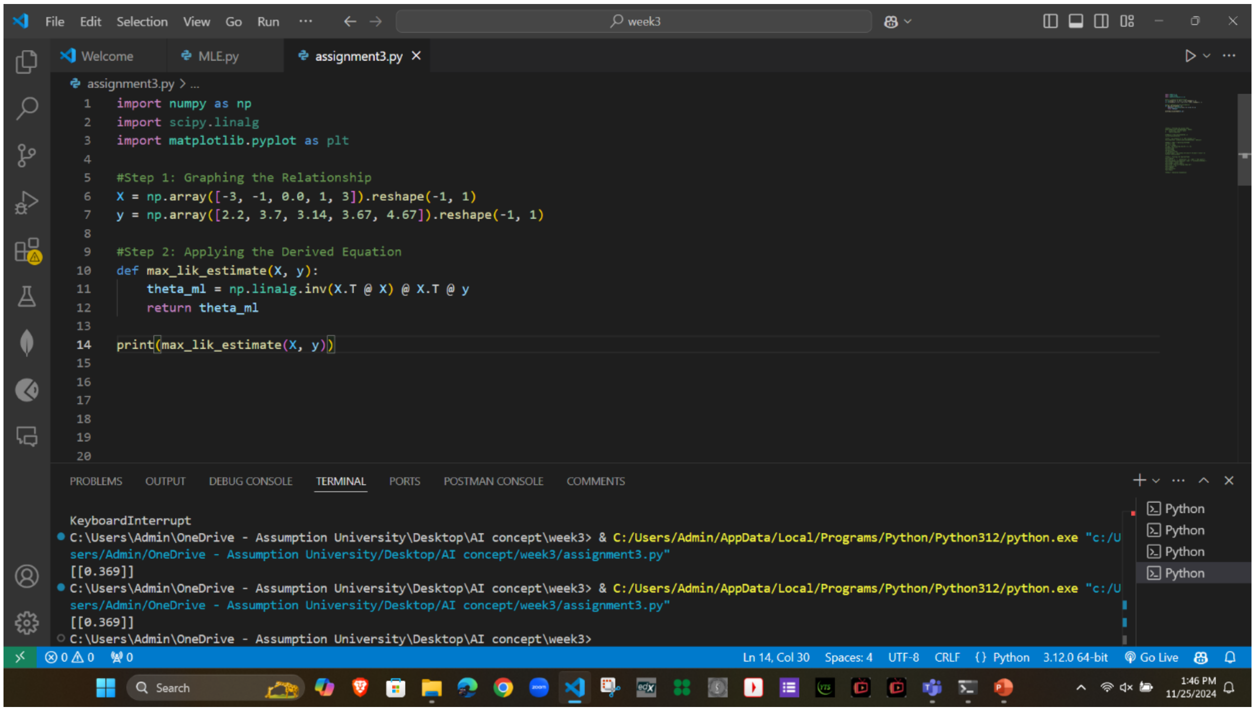


```python
import numpy as np
import scipy.linalg
import matplotlib.pyplot as plt

#Step 1: Graphing the Relationship
X = np.array([-3, -1, 0.0, 1, 3]).reshape(-1, 1)
y = np.array([2.2, 3.7, 3.14, 3.67, 4.67]).reshape(-1, 1)

plt.figure()
plt.plot(X, y, '+', markersize = 10 , label = 'Data points')
plt.xlabel ("X-axis ($x$)")
plt.ylabel ("y-axis ($y$)")
plt.title ("Plot of Training Data Set")
plt.xlim([-5,5])
plt.legend()
plt.grid(True)
plt.show()
```

```
C:\Users\Admin\OneDrive - Assumption University\Desktop\AI concept\week3> & C:/Users/Admin/AppData/Local/Programs/Python/Python312/python.exe "c:/U
sers/Admin/OneDrive - Assumption University/Desktop/AI concept/week3/assignment3.py"
```

File Edit Selection View Go Run
week3
Welcome
MLE.py
assignment3.py
Figure 1
Plot of Training Data Set
Data points
4.5
4.0
3.5
3.0
2.5
y-axis (y)
−4
−2
0
2
4
X-axis (x)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
Python
Python
Python
Python
Ln 17, Col 11 Spaces: 4 UTF-8 CRLF {} Python 3.12.0 64-bit Go Live
Search
1:43 PM
11/25/2024

## Step 2: Applying the Derived Equation



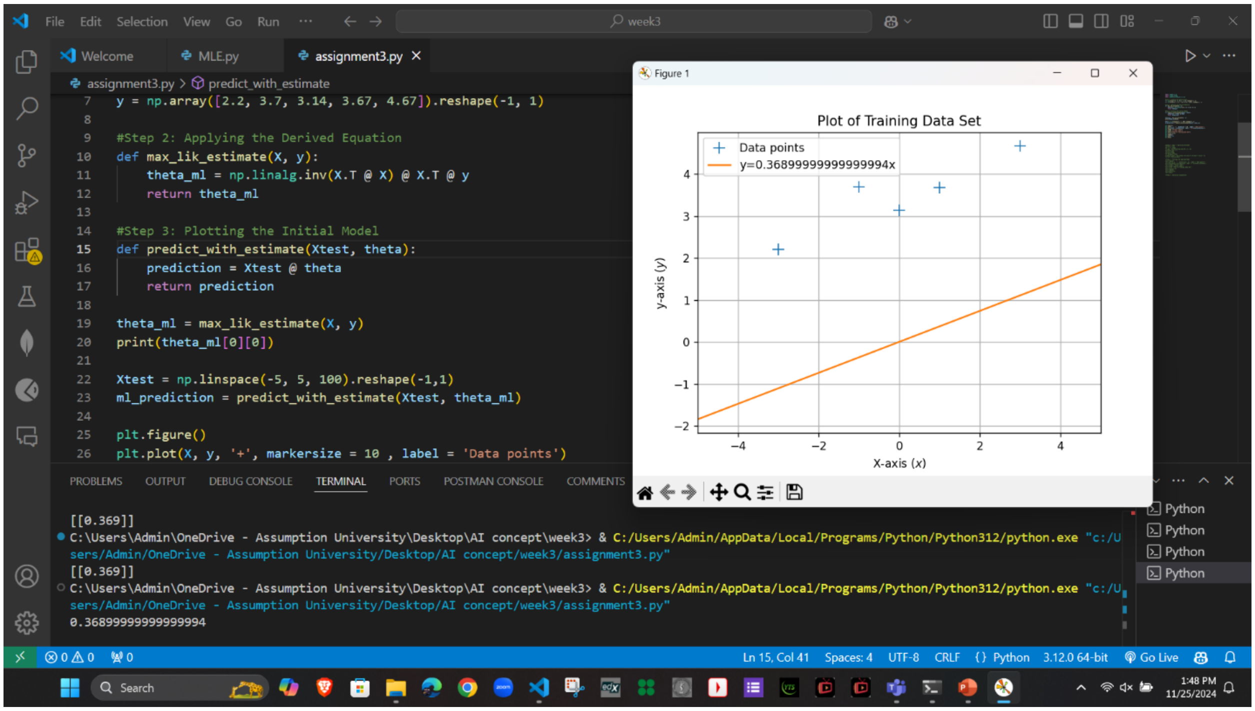
```python
import numpy as np
import scipy.linalg
import matplotlib.pyplot as plt

#Step 1: Graphing the Relationship
X = np.array([-3, -1, 0.0, 1, 3]).reshape(-1, 1)
y = np.array([2.2, 3.7, 3.14, 3.67, 4.67]).reshape(-1, 1)

#Step 2: Applying the Derived Equation
def max_lik_estimate(X, y):
    theta_ml = np.linalg.inv(X.T @ X) @ X.T @ y
    return theta_ml

print(max_lik_estimate(X, y))
```

```
KeyboardInterrupt
C:\Users\Admin\OneDrive - Assumption University\Desktop\AI concept\week3> & C:/Users/Admin/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Admin/OneDrive - Assumption University/Desktop/AI concept/week3/assignment3.py"
[[0.369]]
C:\Users\Admin\OneDrive - Assumption University\Desktop\AI concept\week3> & C:/Users/Admin/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Admin/OneDrive - Assumption University/Desktop/AI concept/week3/assignment3.py"
[[0.369]]
C:\Users\Admin\OneDrive - Assumption University\Desktop\AI concept\week3>
```

Theta value is 0.3689

## Step 3: Plotting the Initial Model

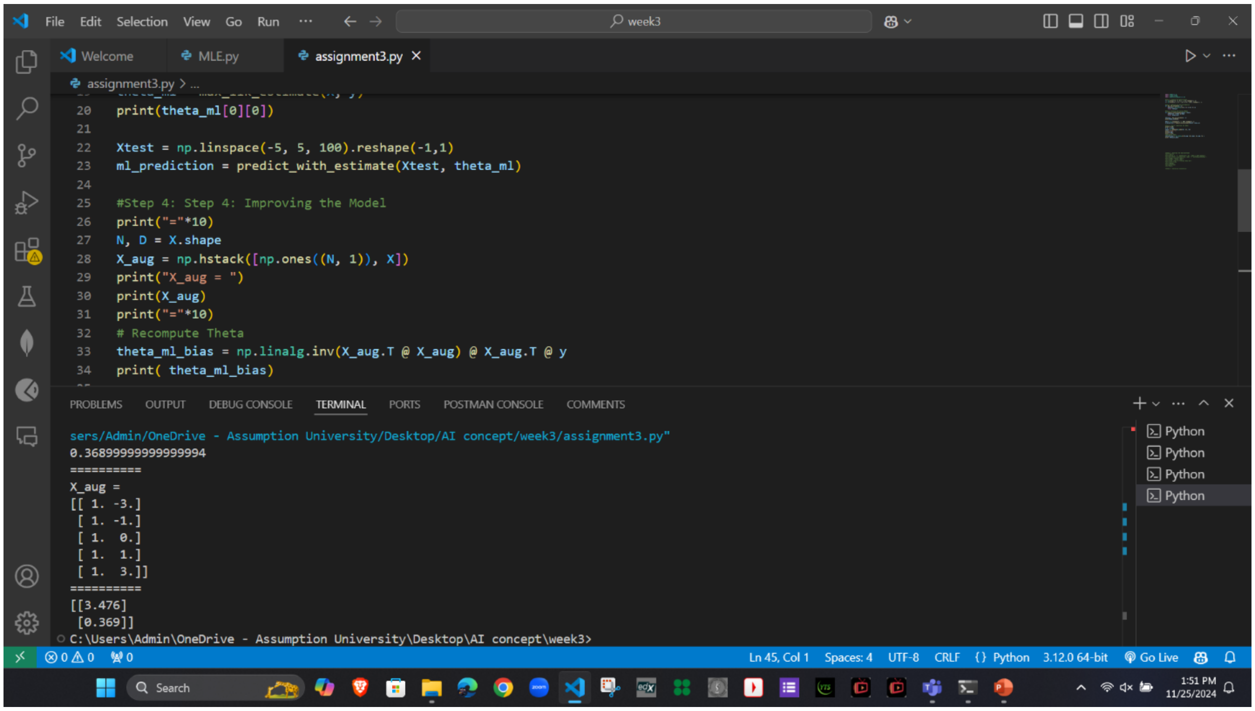
- Applying Theta obtained from Step2 in Step3



```python
y = np.array([2.2, 3.7, 3.14, 3.67, 4.67]).reshape(-1, 1)

#Step 2: Applying the Derived Equation
def max_lik_estimate(X, y):
    theta_ml = np.linalg.inv(X.T @ X) @ X.T @ y
    return theta_ml

#Step 3: Plotting the Initial Model
def predict_with_estimate(Xtest, theta):
    prediction = Xtest @ theta
    return prediction

theta_ml = max_lik_estimate(X, y)
print(theta_ml[0][0])

Xtest = np.linspace(-5, 5, 100).reshape(-1,1)
ml_prediction = predict_with_estimate(Xtest, theta_ml)

plt.figure()
plt.plot(X, y, '+', markersize = 10 , label = 'Data points')
```

```
[[0.369]]
C:\Users\Admin\OneDrive - Assumption University\Desktop\AI concept\week3> & C:/Users/Admin/AppData/Local/Programs/Python/Python312/python.exe "c:/U
sers/Admin/OneDrive - Assumption University/Desktop/AI concept/week3/assignment3.py"
[[0.369]]
C:\Users\Admin\OneDrive - Assumption University\Desktop\AI concept\week3> & C:/Users/Admin/AppData/Local/Programs/Python/Python312/python.exe "c:/U
sers/Admin/OneDrive - Assumption University/Desktop/AI concept/week3/assignment3.py"
0.36899999999999994
```

Step 4: Improving the Model



```python
print(theta_ml[0][0])

Xtest = np.linspace(-5, 5, 100).reshape(-1,1)
ml_prediction = predict_with_estimate(Xtest, theta_ml)

#Step 4: Step 4: Improving the Model
print("="*10)
N, D = X.shape
X_aug = np.hstack([np.ones((N, 1)), X])
print("X_aug = ")
print(X_aug)
print("="*10)
# Recompute Theta
theta_ml_bias = np.linalg.inv(X_aug.T @ X_aug) @ X_aug.T @ y
print( theta_ml_bias)
```

```
sers/Admin/OneDrive - Assumption University/Desktop/AI concept/week3/assignment3.py"
0.36899999999999994
==========
X_aug =
[[ 1. -3.]
 [ 1. -1.]
 [ 1.  0.]
 [ 1.  1.]
 [ 1.  3.]]
==========
[[3.476]
 [0.369]]
C:\Users\Admin\OneDrive - Assumption University\Desktop\AI concept\week3>
```

Theta values are 3.476 and 0.369

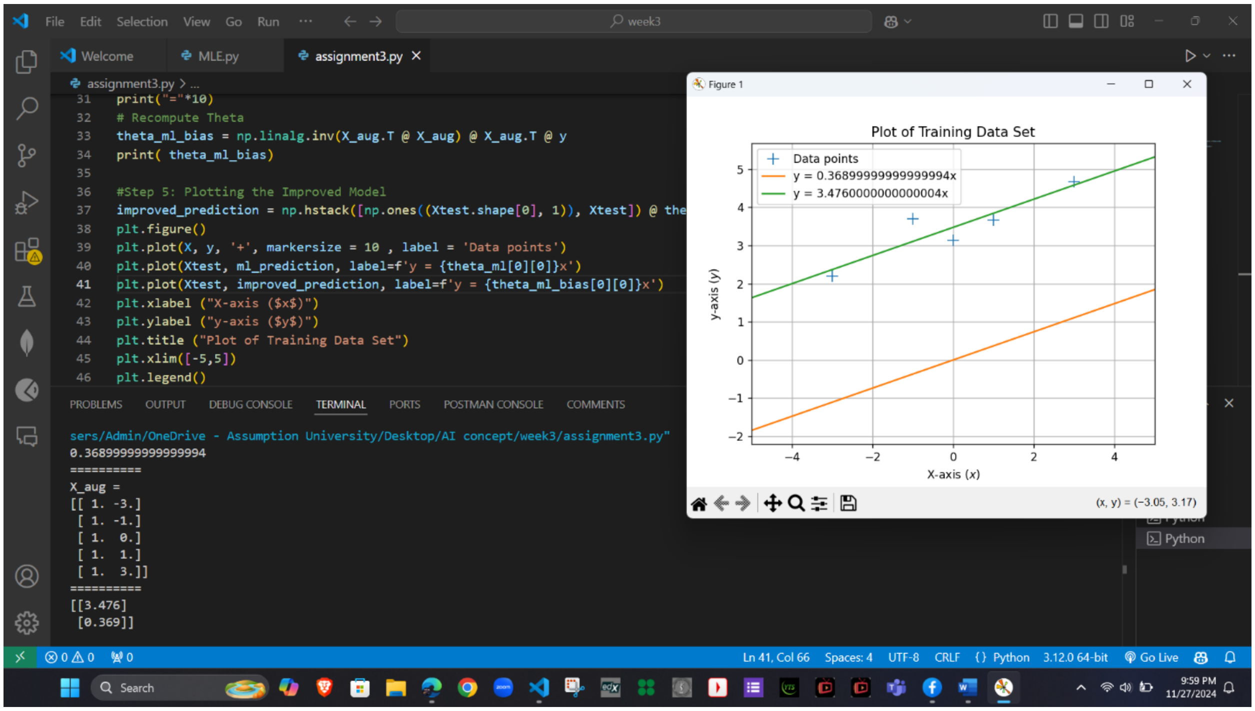
## Explanation

In Step 2, the model assumes the line passes through the origin, so there is no intercept term. The parameter θml is solely the slope, which might not fully capture the relationship if the line does not pass through the origin.

In Step 4, by adding a bias term, the model accounts for a possible offset in the data. The resulting θml-bias includes both the intercept and slope, potentially providing a more accurate fit to the data.

## Step 5: Plotting the Improved Model

- Use new Theta values from step4 in linear equation and plot the graph



```
print("="*10)
# Recompute Theta
theta_ml_bias = np.linalg.inv(X_aug.T @ X_aug) @ X_aug.T @ y
print( theta_ml_bias)

#Step 5: Plotting the Improved Model
improved_prediction = np.hstack([np.ones((Xtest.shape[0], 1)), Xtest]) @ the
plt.figure()
plt.plot(X, y, '+', markersize = 10 , label = 'Data points')
plt.plot(Xtest, ml_prediction, label=f'y = {theta_ml[0][0]}x')
plt.plot(Xtest, improved_prediction, label=f'y = {theta_ml_bias[0][0]}x')
plt.xlabel ("X-axis ($x$)")
plt.ylabel ("y-axis ($y$)")
plt.title ("Plot of Training Data Set")
plt.xlim([-5,5])
plt.legend()
```

```
sers/Admin/OneDrive - Assumption University/Desktop/AI concept/week3/assignment3.py"
0.36899999999999994
==========
X_aug =
[[ 1. -3.]
 [ 1. -1.]
 [ 1.  0.]
 [ 1.  1.]
 [ 1.  3.]]
==========
[[3.476]
 [0.369]]
```

## Step 6: Analytical Explanation

- **Step 2 Limitation:** Assumes the line passes through the origin (y=θx), which may not fit real-world data where y-intercept isn't zero.
- **Bias Term Role:** Adding ones enables the model to estimate a non-zero intercept (y=θ0+θ1x), making it more flexible.
- **Improved Fit:** Accounts for vertical shifts, reducing residual errors and better aligning the line with the data.
- **Broader Model Capability:** Supports relationships with both slope and intercept, fitting a wider range of linear patterns.
- **Mathematical Impact:** Minimizes errors by recalculating parameters (θ0 and θ1) for a closer match to data.