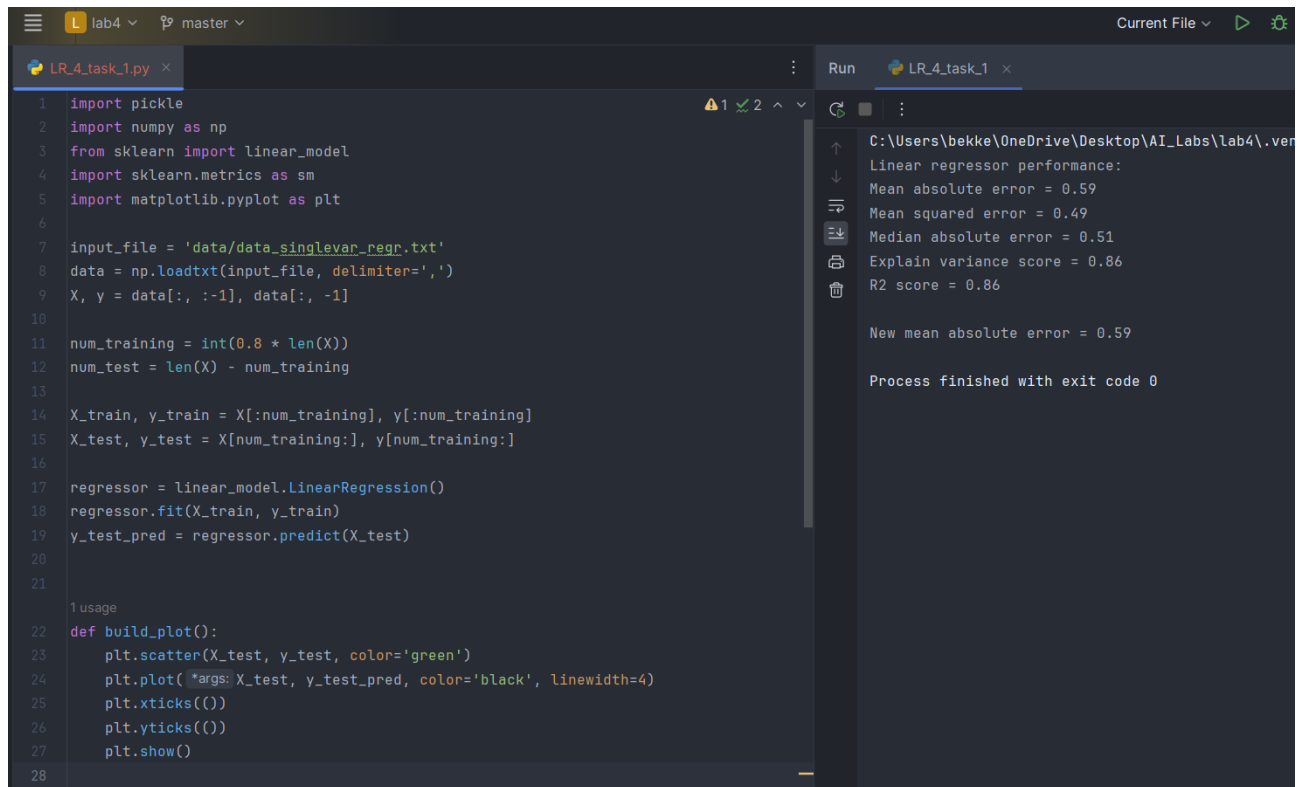


## ЛАБОРАТОРНА РОБОТА № 4

**Мета роботи:** використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити методи регресії даних у машинному навчанні.

**Завдання 1.** Створення регресора однієї змінної



```
1 import pickle
2 import numpy as np
3 from sklearn import linear_model
4 import sklearn.metrics as sm
5 import matplotlib.pyplot as plt
6
7 input_file = 'data/data_singlevar_regr.txt'
8 data = np.loadtxt(input_file, delimiter=',')
9 X, y = data[:, :-1], data[:, -1]
10
11 num_training = int(0.8 * len(X))
12 num_test = len(X) - num_training
13
14 X_train, y_train = X[:num_training], y[:num_training]
15 X_test, y_test = X[num_training:], y[num_training:]
16
17 regressor = linear_model.LinearRegression()
18 regressor.fit(X_train, y_train)
19 y_test_pred = regressor.predict(X_test)
20
21
22 def build_plot():
23     plt.scatter(X_test, y_test, color='green')
24     plt.plot(*args: X_test, y_test_pred, color='black', linewidth=4)
25     plt.xticks(())
26     plt.yticks(())
27     plt.show()
28
```

Run LR\_task\_1

C:\Users\bekke\OneDrive\Desktop\AI\_Labs\lab4\ver

Linear regressor performance:

- Mean absolute error = 0.59
- Mean squared error = 0.49
- Median absolute error = 0.51
- Explain variance score = 0.86
- R2 score = 0.86

New mean absolute error = 0.59

Process finished with exit code 0

Рис 1. Оцінки якості

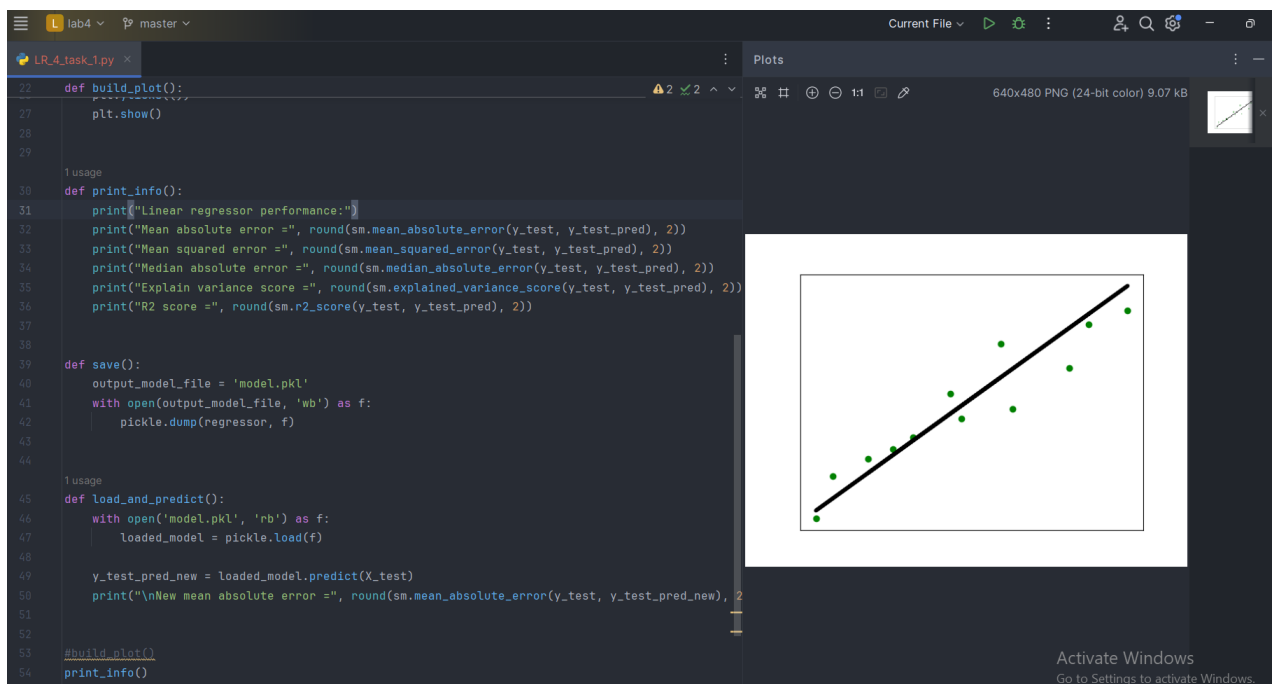


Рис 2. Графік функції

**Висновок:** оскільки на графіку відстань від лінії до більшості точок не велика (відхилення не велике), можна зробити висновок, що модель, в багатьох випадках, працює нормально. Також показники похибок невеликі, а  $R^2$  & Variance score = 86% високі, що підтверджує попередні твердження.

## Завдання 2. Передбачення за допомогою регресії однієї змінної

Побудувати регресійну модель на основі однієї змінної. Використовувати вхідні дані відповідно свого варіанту, що визначається за списком групи у журналі (таблиця 2.1).

`N%2, file = data_regr_2.txt`

Код той самий, змінилась тільки назва файлу.

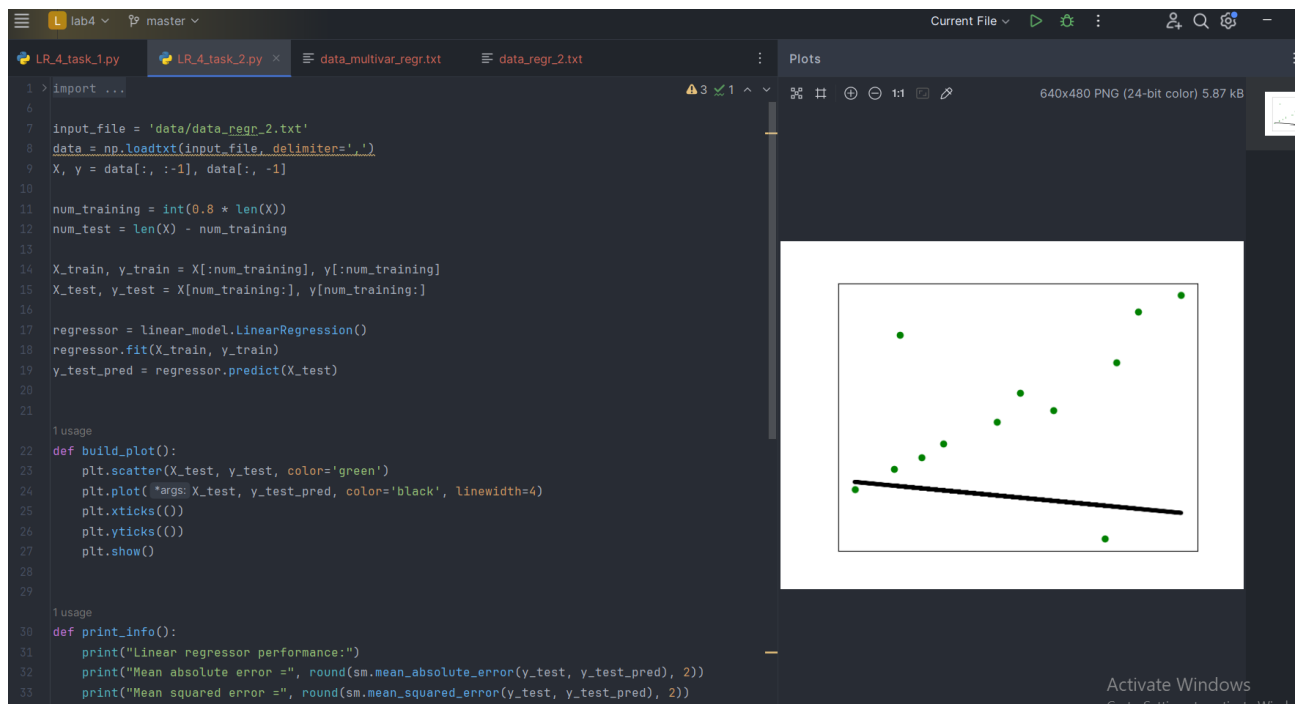


Рис 3. Графік функції

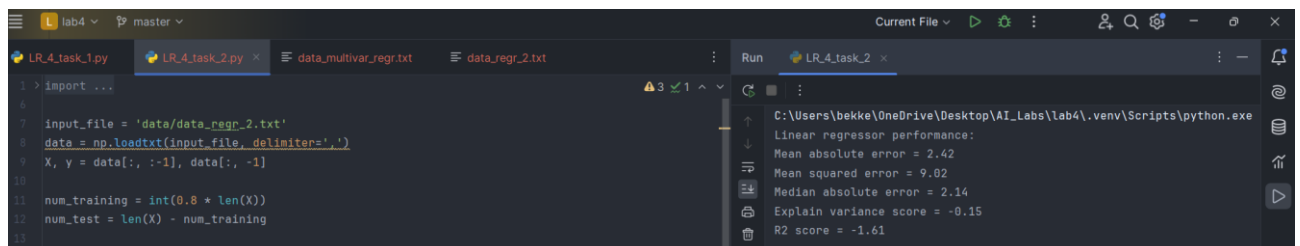
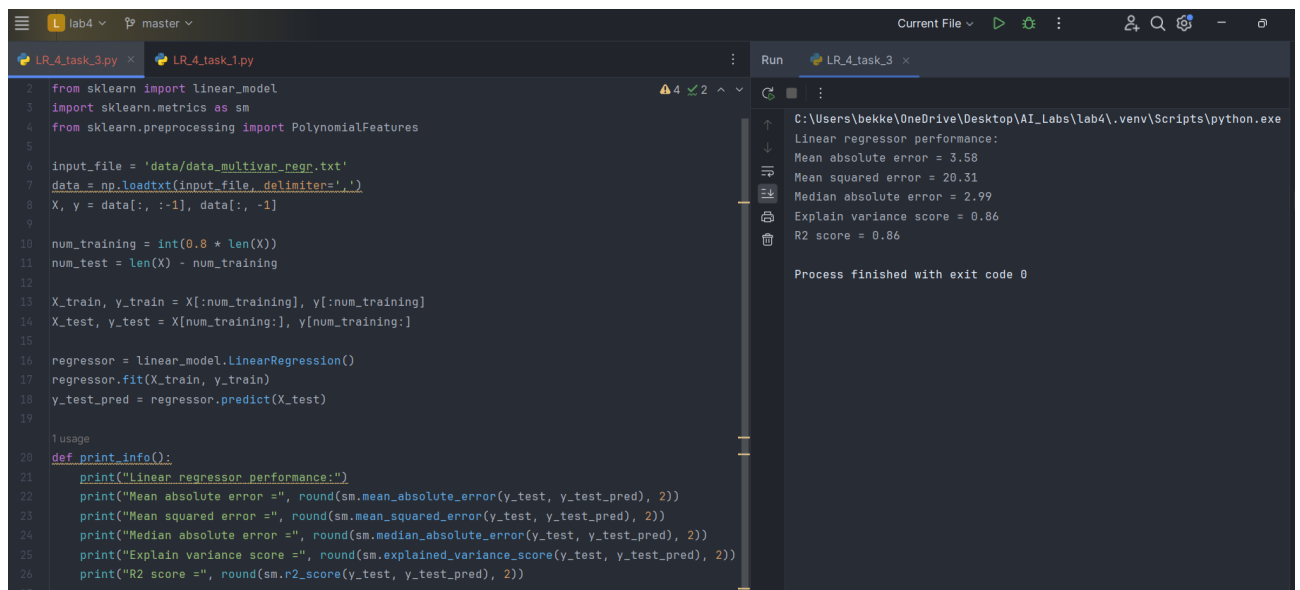


Рис 4. Показники

**Висновок:** на графіку видно дуже великі відхилення від лінії, тому модель працює погано, що підтверджують і показники, особливо Variance score & absolute score, вони від'ємні, що може свідчити про невірні прогнози моделі (гірші ніж просто константний прогноз середнього значення). Також, squared error, призначена для виявлення великих відхилень, у цьому випадку, це 9, що є також великим показником (негативно великим). Можна також припустити, що лінійна регресія погано працює на малих наборах даних.

**Завдання 3.** Використовувати файл вхідних даних: data\_multivar\_regr.txt, побудувати регресійну модель на основі багатьох змінних.



```
2 from sklearn import linear_model
3 import sklearn.metrics as sm
4 from sklearn.preprocessing import PolynomialFeatures
5
6 input_file = 'data/data_multivar_regr.txt'
7 data = np.loadtxt(input_file, delimiter=',')
8 X, y = data[:, :-1], data[:, -1]
9
10 num_training = int(0.8 * len(X))
11 num_test = len(X) - num_training
12
13 X_train, y_train = X[:num_training], y[:num_training]
14 X_test, y_test = X[num_training:], y[num_training:]
15
16 regressor = linear_model.LinearRegression()
17 regressor.fit(X_train, y_train)
18 y_test_pred = regressor.predict(X_test)
19
20 usage
21 def print_info():
22     print("Linear regressor performance:")
23     print("Mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_pred), 2))
24     print("Mean squared error =", round(sm.mean_squared_error(y_test, y_test_pred), 2))
25     print("Median absolute error =", round(sm.median_absolute_error(y_test, y_test_pred), 2))
26     print("Explain variance score =", round(sm.explained_variance_score(y_test, y_test_pred), 2))
27     print("R2 score =", round(sm.r2_score(y_test, y_test_pred), 2))
```

Run LR\_4\_task\_3

C:\Users\bekke\OneDrive\Desktop\AI\_Labs\lab4\.venv\Scripts\python.exe

Linear regressor performance:

- Mean absolute error = 3.58
- Mean squared error = 20.31
- Median absolute error = 2.99
- Explain variance score = 0.86
- R2 score = 0.86

Process finished with exit code 0

Рис 5. Показники моделі лінійної регресії

```
13 X_train, y_train = X[:num_training], y[:num_training]
14 X_test, y_test = X[num_training:], y[num_training:]
15
16 regressor = linear_model.LinearRegression()
17 regressor.fit(X_train, y_train)
18 y_test_pred = regressor.predict(X_test)
19
20 def print_info():
21     print("Linear regressor performance:")
22     print("Mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_pred), 2))
23     print("Mean squared error =", round(sm.mean_squared_error(y_test, y_test_pred), 2))
24     print("Median absolute error =", round(sm.median_absolute_error(y_test, y_test_pred), 2))
25     print("Explained variance score =", round(sm.explained_variance_score(y_test, y_test_pred), 2))
26     print("R2 score =", round(sm.r2_score(y_test, y_test_pred), 2))
27
28
29 #print_info()
30
31 polynomial = PolynomialFeatures(degree=10)
32 X_train_transformed = polynomial.fit_transform(X_train)
33 datapoint = [[7.75, 6.35, 5.56]]
34 poly_datapoint = polynomial.fit_transform(datapoint)
35 poly_linear_model = linear_model.LinearRegression()
36 poly_linear_model.fit(X_train_transformed, y_train)
37 print("\nLinear regression:\n", regressor.predict(datapoint))
38 print("\nPolynomial regression:\n", poly_linear_model.predict(poly_datapoint))
```

Run LR\_4\_task\_3

C:\Users\bekke\OneDrive\Desktop\AI\_Labs\la

Linear regression:  
[36.05286276]

Polynomial regression:  
[41.46543157]

Process finished with exit code 0

Рис 6. Результати поліноміального регресора та лінійного

На 11 рядку немає даних, які зазначені в лабораторній, а також немає й жодного рядка з такими значеннями, але, у будь якому випадку, якщо навіть взяти реальні дані, то різниця залишається помітною.

>	Q	↺	Cc	W	.*	0 resu
1	2.06,3.48,7.21,15.69					
2	6.37,3.01,7.27,15.34					
3	1.18,1.2,5.42,0.66					
4	7.37,3.81,-1.95,38.37					
5	6.16,1.39,7.39,9.96					
6	4.49,5.74,6.05,23.4					
7	7.35,2.49,5.4,24.15					
8	7.7,3.39,6.62,24.89					
9	3.87,3.67,1.81,30.23					
10	5.61,6.14,1.42,41.2					
11	7.66,6.29,5.66,41.35					

Рис 7. Актуальні дані

**Висновок:** у даному випадку, поліноміальний регресор передбачає краще, тому що передбачене значення ближче до реального. Лінійний регресор можна підігнати до поліноміального шляхом трансформації даних

## Завдання 4. Регресія багатьох змінних

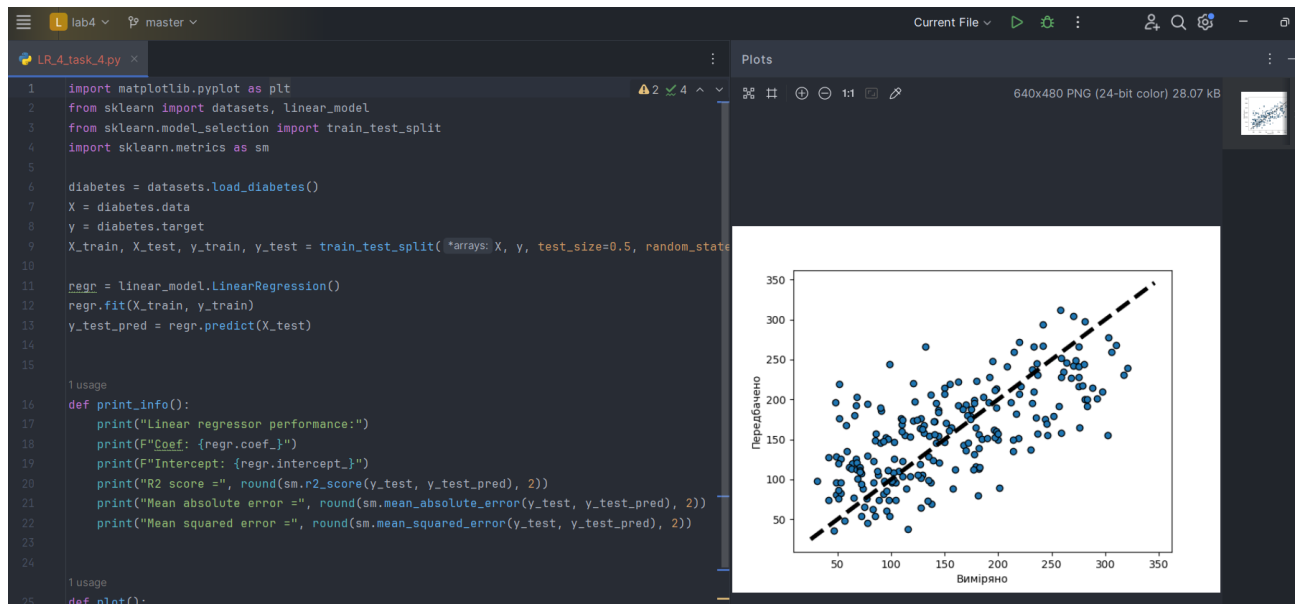


Рис 8. Графік функції

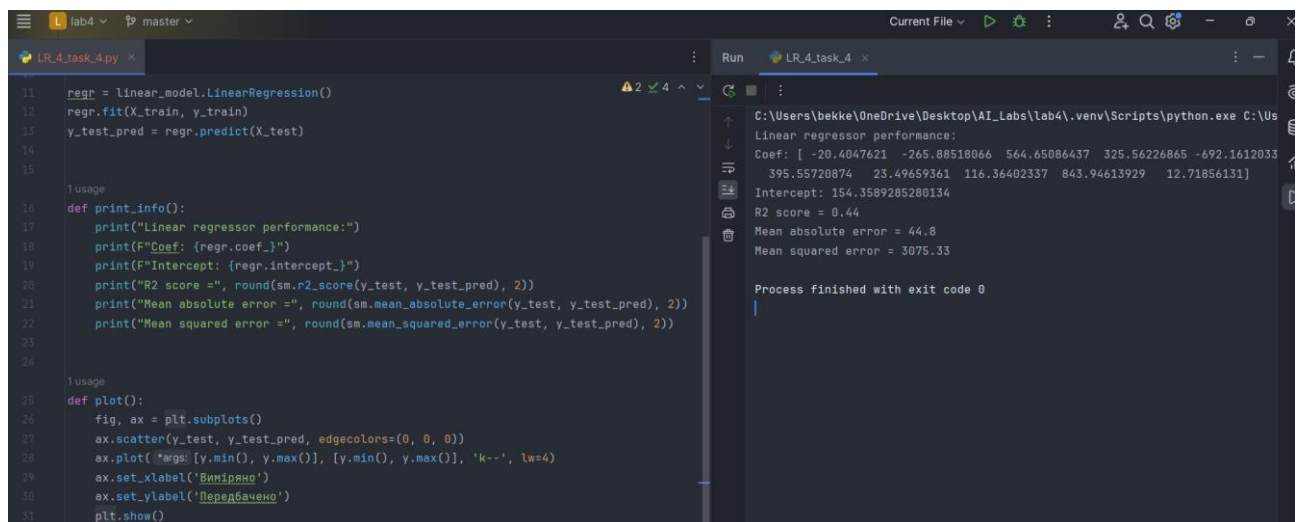


Рис 9. Показники

**Висновок:** на графіку видно, що є точки, які лежать близько до лінії, а також точки, що далеко, це може свідчити про не дуже гарну роботу моделі, про що також свідчать і показники, наприклад  $R^2$  (0.44) є прийнятним, але не ідеальним.

## Завдання 5. Самостійна побудова регресії

N% 2, Варіант 2

Згенеруйте свої випадкові дані обравши за списком відповідно свій варіант (згідно табл. 2.2) та виведіть їх на графік

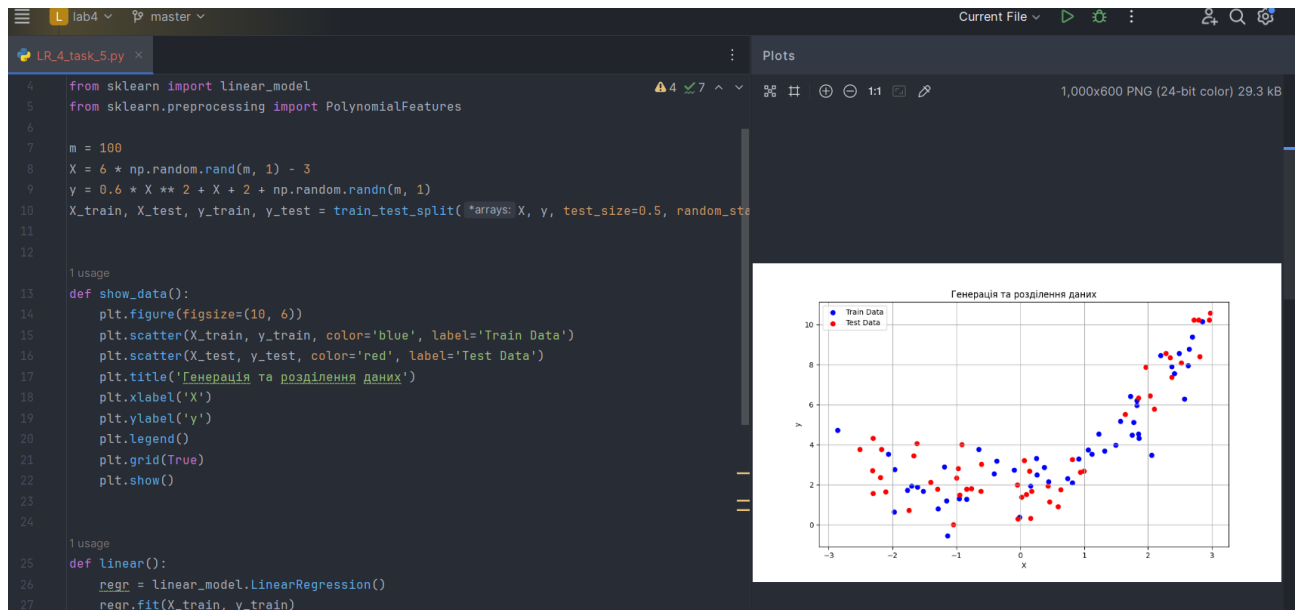


Рис 10. Дані на графіку

Побудуйте по них модель лінійної регресії, виведіть на графік

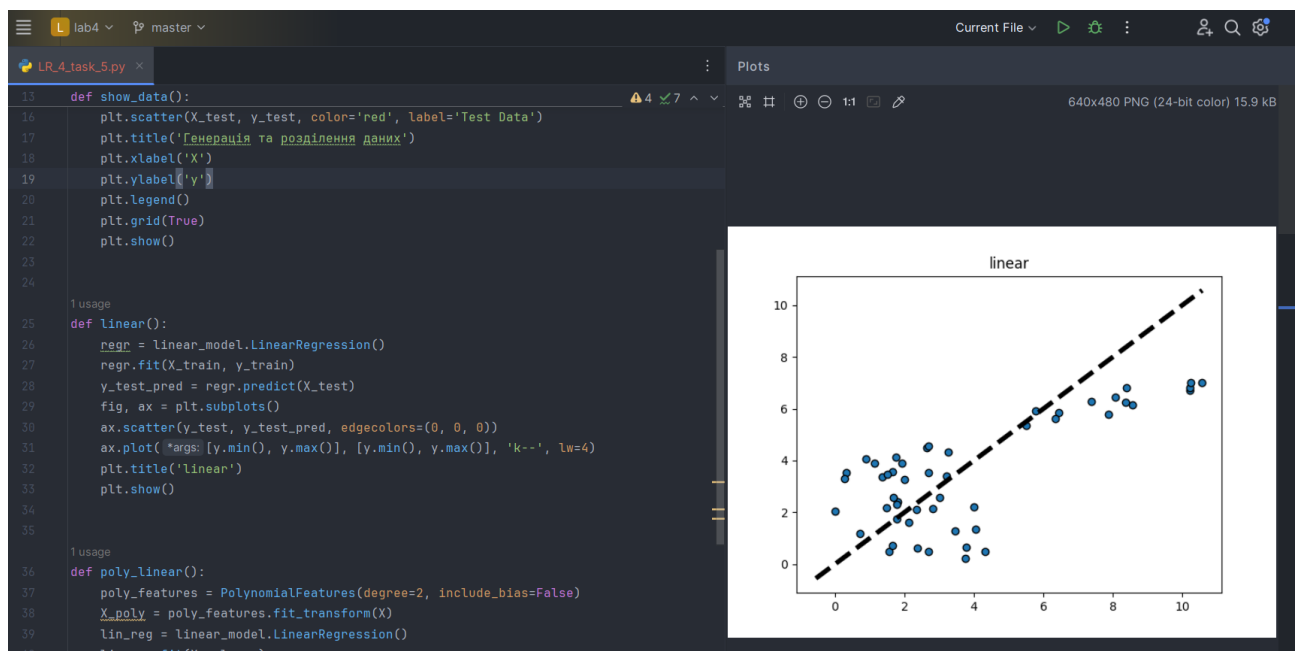


Рис 11. Модель лінійної регресії

Побудуйте по них модель поліноміальної регресії, виведіть на графік. Оцініть її якість.

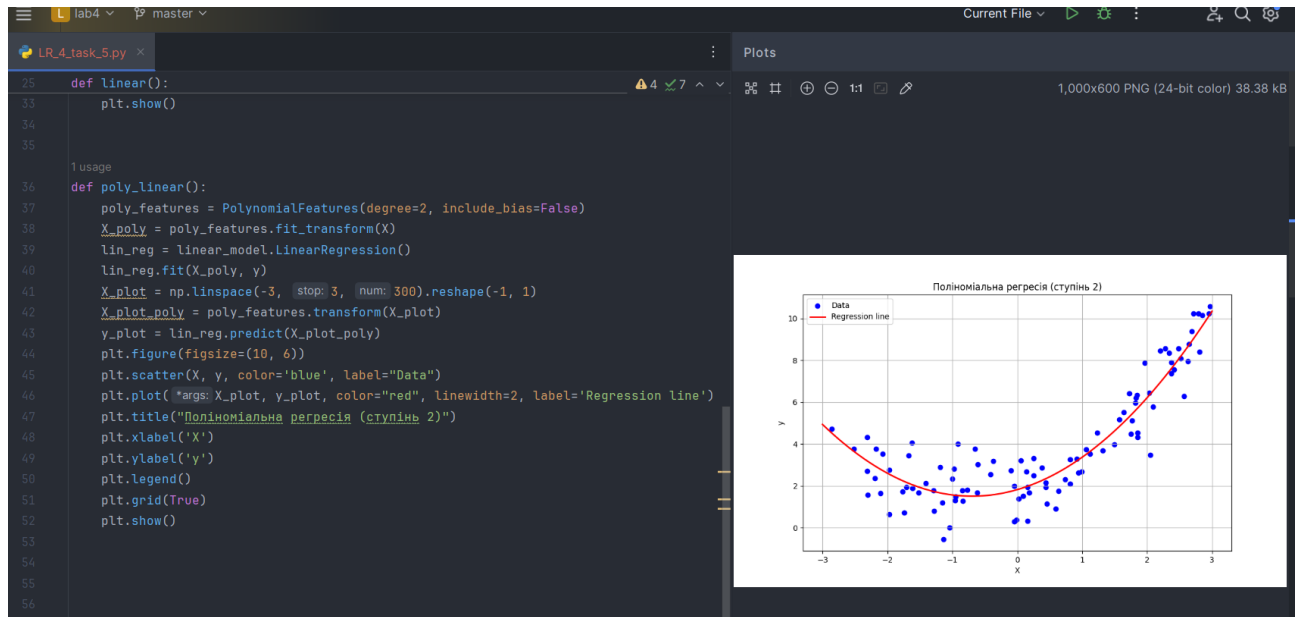


Рис 12. Модель поліноміальної регресії

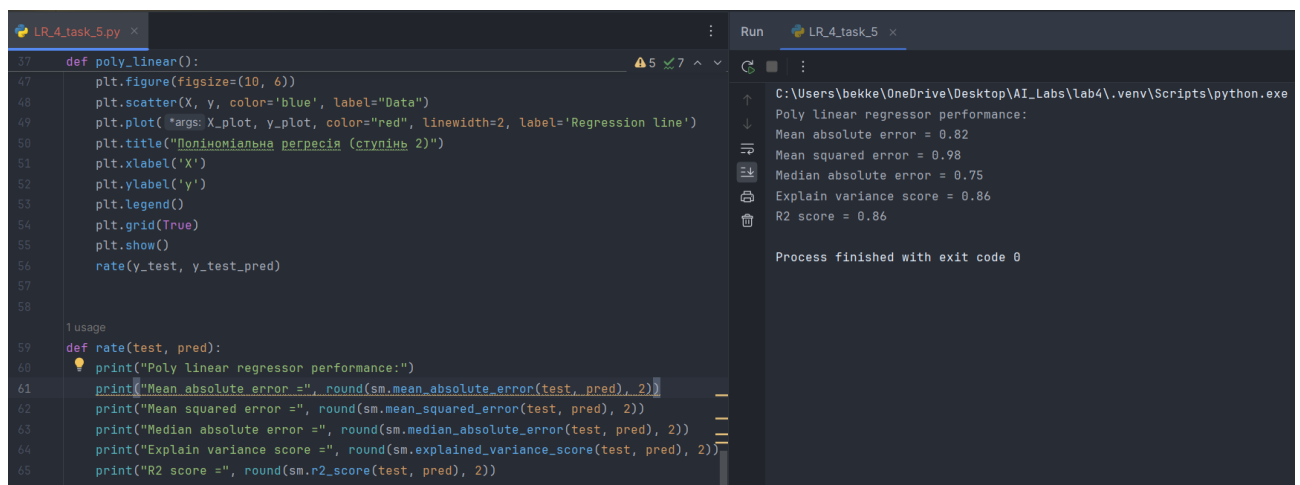


Рис 13. Оцінка якості

Запишіть модель вашого варіанта у вигляді математичного рівняння:

$$y = 0.6 * X^2 + X + 2 + np.rand$$

Запишіть отриману вами модель регресії з передбаченими коефіцієнтами

$$y = 0.58 * x^2 + 1.08 * x + 2.13$$

Коефіцієнти отримав ось так:

```
coefficients = lin_reg.coef_  
intercept = lin_reg.intercept_  
print(f"y = {coefficients[0][1]:.2f} * x^2 + {coefficients[0][0]:.2f} * x + {intercept[0]:.2f}")
```

Рис 14. Коефіцієнти

**Висновок:** якщо дані складніші за звичайну пряму, то використовувати звичайну лінійну модель недоцільно, краще використати поліноміальну модель, щоб отримати лінію зображену належним чином (пряма лінія ніколи не зможе бути підігнаною під нелінійні дані). Що до оцінок поліноміальної моделі: вони досить високі, тобто модель працює добре.

### Завдання 6. Побудова кривих навчання

*Побудуйте криві навчання для звичайної лінійної регресійної моделі із попереднього завдання (ваш варіант)*

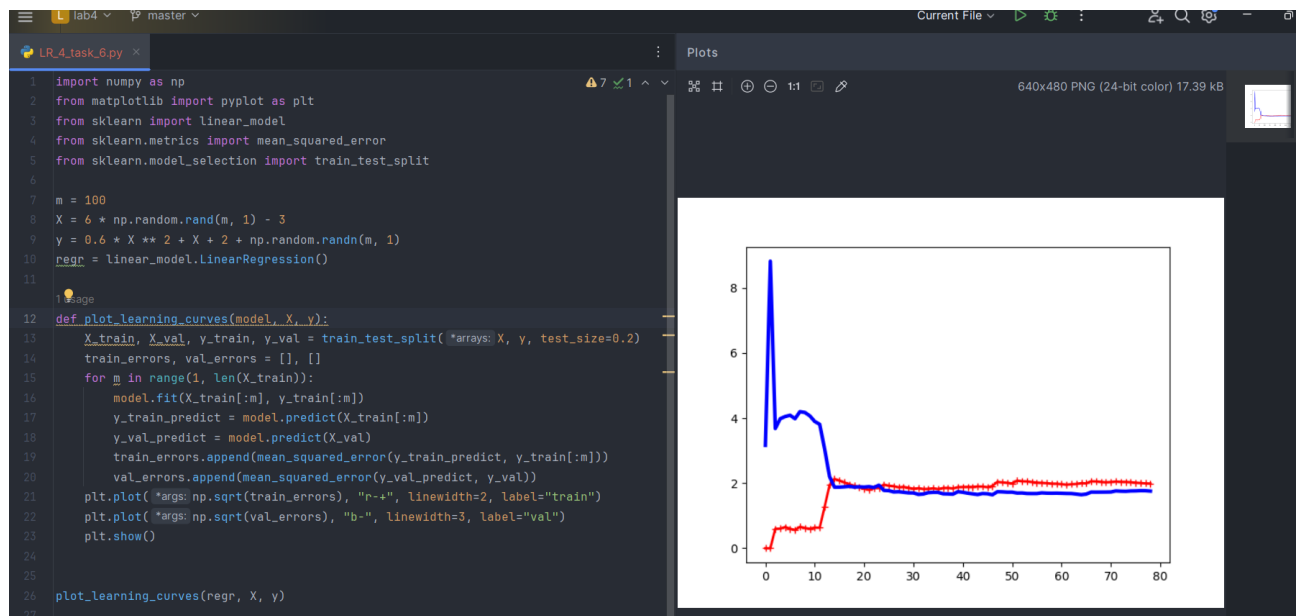


Рис 15. Криві навчання для лінійної регресійної моделі



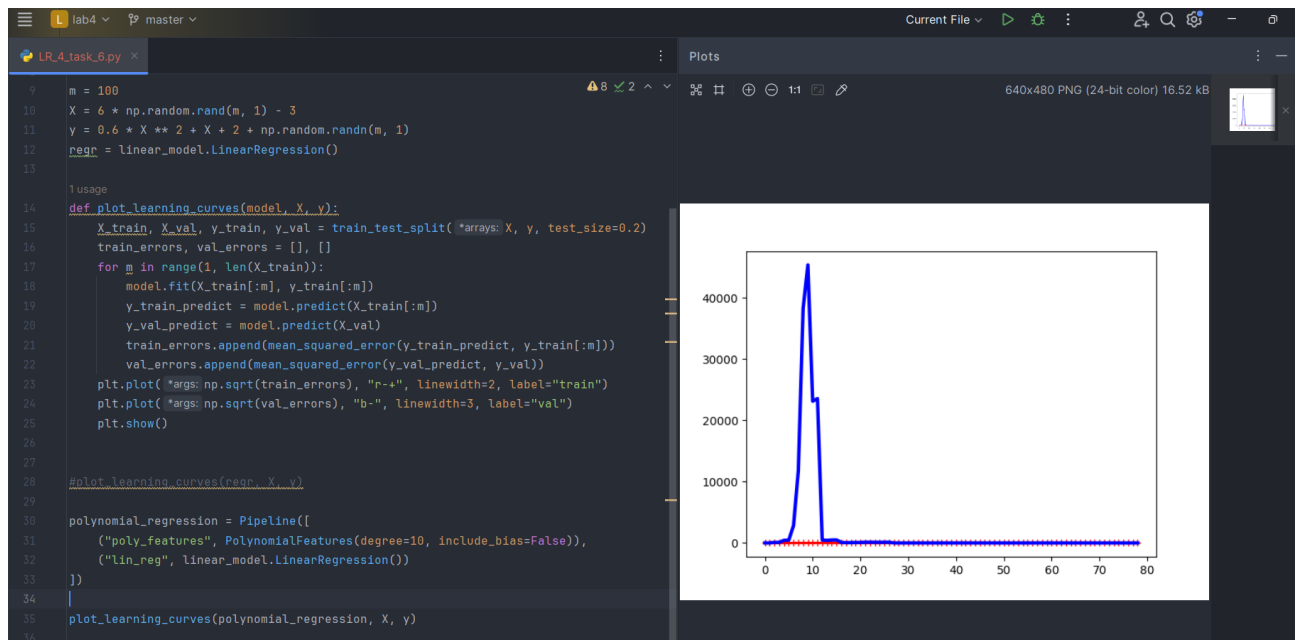


Рис 16. Криві навчання для поліноміальної регресії (degree=10)



Рис 17. Криві навчання для поліноміальної регресії (degree=2)

Github: [link](#)

**Висновок:** на даній лабораторній роботі я, використовуючи спеціалізовані бібліотеки та мову програмування Python, дослідив методи регресії даних у машинному навчанні, а також навчився оцінювати їх та дізнався в яких випадках їх потрібно застосовувати. Навчився за допомогою кривих навчання аналізувати моделі.