

# ЛАБОРАТОРНА РОБОТА №3

## ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ

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

**Хід роботи:**

### Завдання 1. Створення регресора однієї змінної

Побудувати регресійну модель на основі однієї змінної. Використовувати файл вхідних даних: data\_singlevar\_regr.txt.

Лістинг програми:

```
import pickle
import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
import matplotlib.pyplot as plt

# Вхідний файл, який містить дані
input_file = 'data_singlevar_regr.txt'

# Завантаження даних
data = np.loadtxt(input_file, delimiter=',')
X, y = data[:, :-1], data[:, -1]

# Розбивка даних на навчальний та тестовий набори
num_training = int(0.8 * len(X))
num_test = len(X) - num_training

# Тренувальні дані
X_train, y_train = X[:num_training], y[:num_training]
# Тестові дані
X_test, y_test = X[num_training:], y[num_training:]

# Створення об'єкта лінійного регресора
regressor = linear_model.LinearRegression()
regressor.fit(X_train, y_train)

# Прогнозування результату
y_test_pred = regressor.predict(X_test)

# Побудова графіка
plt.scatter(X_test, y_test, color='green')
plt.plot(X_test, y_test_pred, color='black', linewidth=4)
plt.xticks(())
plt.yticks(())
plt.show()
```

					Житомирська Політехніка.22.121.15.000 – Лр03				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Сівченко О. О.			Звіт з лабораторної роботи	Літ.	Арк.	Аркушів	
Перевір.		Філіпов В. О.					1	12	
Керівник						ФІКТ Гр. ПІ-60[2]			
Н. контр.									
Зав. каф.									

```

print("Linear regressor performance:")
print("Mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_pred),
2))
print("Mean squared error =", round(sm.mean_squared_error(y_test, y_test_pred),
2))
print("Median absolute error =", round(sm.median_absolute_error(y_test,
y_test_pred), 2))
print("Explain variance score =", round(sm.explained_variance_score(y_test,
y_test_pred), 2))
print("R2 score =", round(sm.r2_score(y_test, y_test_pred), 2))

# Файл для збереження моделі
output_model_file = 'model.pkl'
# Збереження моделі
with open(output_model_file, 'wb') as f:
    pickle.dump(regressor, f)

# Завантаження моделі
y_test_pred_new = regressor.predict(X_test)
print("\nNew mean absolute error =", round(sm.mean_absolute_error(y_test,
y_test_pred_new), 2))

```

```

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

```

Рисунок 1.1 – Результат роботи програми

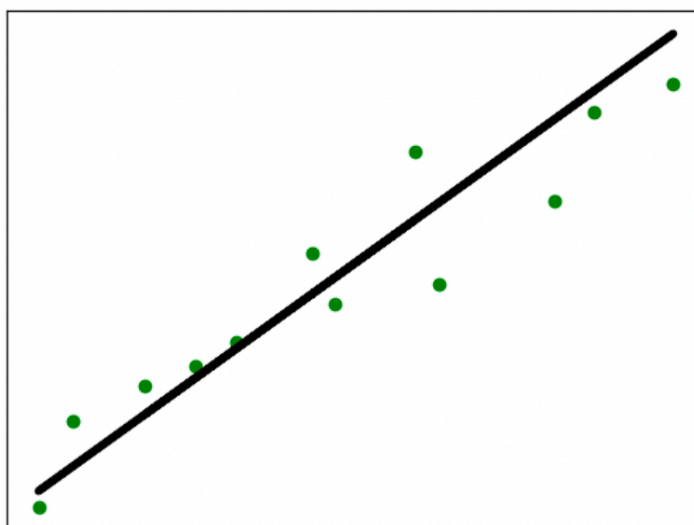


Рисунок 1.2 – Результат роботи програми у вигляді графіку

		Сівченко О. О.			Житомирська Політехніка.22.121.15.000 – Пр03	Арк.
		Філіпов В. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

За результатами роботи програми та судячи з графіку на рисунку 1.2 можна зробити висновок, що у деяких випадках відхилення від справжніх значень надто відрізняється.

## 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 1.1).

Таблиця 1.1

№ за списком	6
№ варіанту	Варіант 1 файл: data_regr_1.txt

Лістинг програми:

```
import pickle
import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
import matplotlib.pyplot as plt

# Вхідний файл, який містить дані
input_file = 'data_regr_1.txt'

# Завантаження даних
data = np.loadtxt(input_file, delimiter=',')
X, y = data[:, :-1], data[:, -1]

# Розбивка даних на навчальний та тестовий набори
num_training = int(0.8 * len(X))
num_test = len(X) - num_training

# Тренувальні дані
X_train, y_train = X[:num_training], y[:num_training]
# Тестові дані
X_test, y_test = X[num_training:], y[num_training:]

# Створення об'єкта лінійного регресора
regressor = linear_model.LinearRegression()
regressor.fit(X_train, y_train)

# Прогнозування результату
y_test_pred = regressor.predict(X_test)

# Побудова графіка
plt.scatter(X_test, y_test, color='green')
plt.plot(X_test, y_test_pred, color='black', linewidth=4)
plt.xticks(())
plt.yticks(())
```

		Сівченко О. О.			Житомирська Політехніка.22.121.15.000 – Лр03	Арк.
		Філіпов В. О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```
plt.show()

print("Linear regressor performance:")
print("Mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_pred),
2))
print("Mean squared error =", round(sm.mean_squared_error(y_test, y_test_pred),
2))
print("Median absolute error =", round(sm.median_absolute_error(y_test,
y_test_pred), 2))
print("Explain variance score =", round(sm.explained_variance_score(y_test,
y_test_pred), 2))
print("R2 score =", round(sm.r2_score(y_test, y_test_pred), 2))

# Файл для збереження моделі
output_model_file = 'model_1.pkl'
# Збереження моделі
with open(output_model_file, 'wb') as f:
    pickle.dump(regressor, f)

# Завантаження моделі
y_test_pred_new = regressor.predict(X_test)
print("\nNew mean absolute error =", round(sm.mean_absolute_error(y_test,
y_test_pred_new), 2))
```

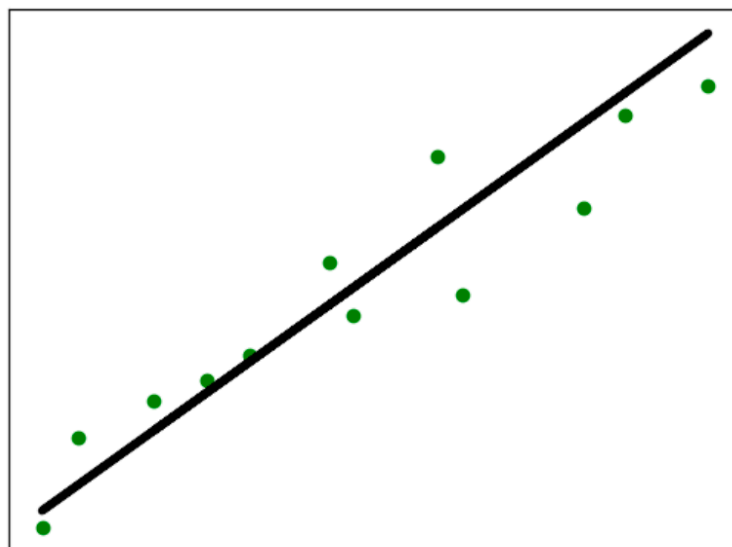


Рисунок 1.3 – Результат роботи програми у вигляді графіку

```
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
```

Рисунок 1.4 – Результат роботи програми

		Сівченко О. О.			Житомирська Політехніка.22.121.15.000 – Лр03	Арк.
		Філіпов В. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Порівнюючи із результатами завдання 2.1 вони вийшли такими самими, тому що дані у файлах абсолютно однакові.

### Завдання 3. Створення багатовимірного регресора

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

Лістинг програми:

```
import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
from sklearn.preprocessing import PolynomialFeatures

# Вхідний файл, який містить дані
input_file = 'data_multivar_regr.txt'

# Завантаження даних
data = np.loadtxt(input_file, delimiter=',')
X, y = data[:, :-1], data[:, -1]

# Розбивка даних на навчальний та тестовий набори
num_training = int(0.8 * len(X))
num_test = len(X) - num_training

# Тренувальні дані
X_train, y_train = X[:num_training], y[:num_training]
# Тестові дані
X_test, y_test = X[num_training:], y[num_training:]

# Створення об'єкта лінійного регресора
regressor = linear_model.LinearRegression()
regressor.fit(X_train, y_train)

# Прогнозування результату
y_test_pred = regressor.predict(X_test)

print("Linear regressor performance:")
print("Mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_pred), 2))
print("Mean squared error =", round(sm.mean_squared_error(y_test, y_test_pred), 2))
print("Median absolute error =", round(sm.median_absolute_error(y_test, y_test_pred), 2))
print("Explain variance score =", round(sm.explained_variance_score(y_test, y_test_pred), 2))
print("R2 score =", round(sm.r2_score(y_test, y_test_pred), 2))

# Поліноміальна регресія
polynomial = PolynomialFeatures(degree=10)
X_train_transformed = polynomial.fit_transform(X_train)
datapoint = [[7.75, 6.35, 5.56]]
poly_datapoint = polynomial.fit_transform(datapoint)

poly_linear_model = linear_model.LinearRegression()
poly_linear_model.fit(X_train_transformed, y_train)
```

		Сівченко О. О.			Житомирська Політехніка.22.121.15.000 – Лр03	Арк.
		Філіпов В. О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```
print("\nLinear regression:\n", regressor.predict(datapoint))
print("\nPolynomial regression:\n", poly_linear_model.predict(poly_datapoint))
```

```
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

Linear regression:
[36.05286276]

Polynomial regression:
[41.46072631]
```

Рисунок 1.5 – Результат роботи програми

Порівняно з лінійним регресором поліноміальний регресор забезпечує отримання результату, ближчого до значення 41.35, а саме було отримано 41.46 – тобто дає кращі результати.

#### Завдання 4. Регресія багатьох змінних

Розробіть лінійний регресор, використовуючи набір даних по діабету, який існує в sklearn.datasets.

Лістинг програми:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split

diabetes = datasets.load_diabetes()
X = diabetes.data
y = diabetes.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5,
                                                    random_state=0)
```

		Сівченко О. О.			Житомирська Політехніка.22.121.15.000 – Лр03	Арк.
		Філіпов В. О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

regr = linear_model.LinearRegression()
regr.fit(X_train, y_train)

y_pred = regr.predict(X_test)

print("Linear regressor performance:")
print("Mean absolute error =", round(mean_absolute_error(y_test, y_pred), 2))
print("Mean squared error =", round(mean_squared_error(y_test, y_pred), 2))
print("Independent term = \n", regr.coef_)
print("Estimated coefficients for regression problem =", regr.intercept_)
print("R2 score =", round(r2_score(y_test, y_pred), 2))

fig, ax = plt.subplots()
ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4)
ax.set_xlabel('Виміряно')
ax.set_ylabel('Передбачено')
plt.show()

```

Linear regressor performance:

Mean absolute error = 44.8

Mean squared error = 3075.33

Independent term =

[ -20.41129305 -265.88594023 564.64844662 325.55650029 -692.23796104  
395.62249978 23.52910434 116.37102129 843.98257585 12.71981044]

Estimated coefficients for regression problem = 154.3589882135515

R2 score = 0.44

Рисунок 1.6 – Результат роботи програми

		Сівченко О. О.			Житомирська Політехніка.22.121.15.000 – Лр03	Арк.
		Філіпов В. О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

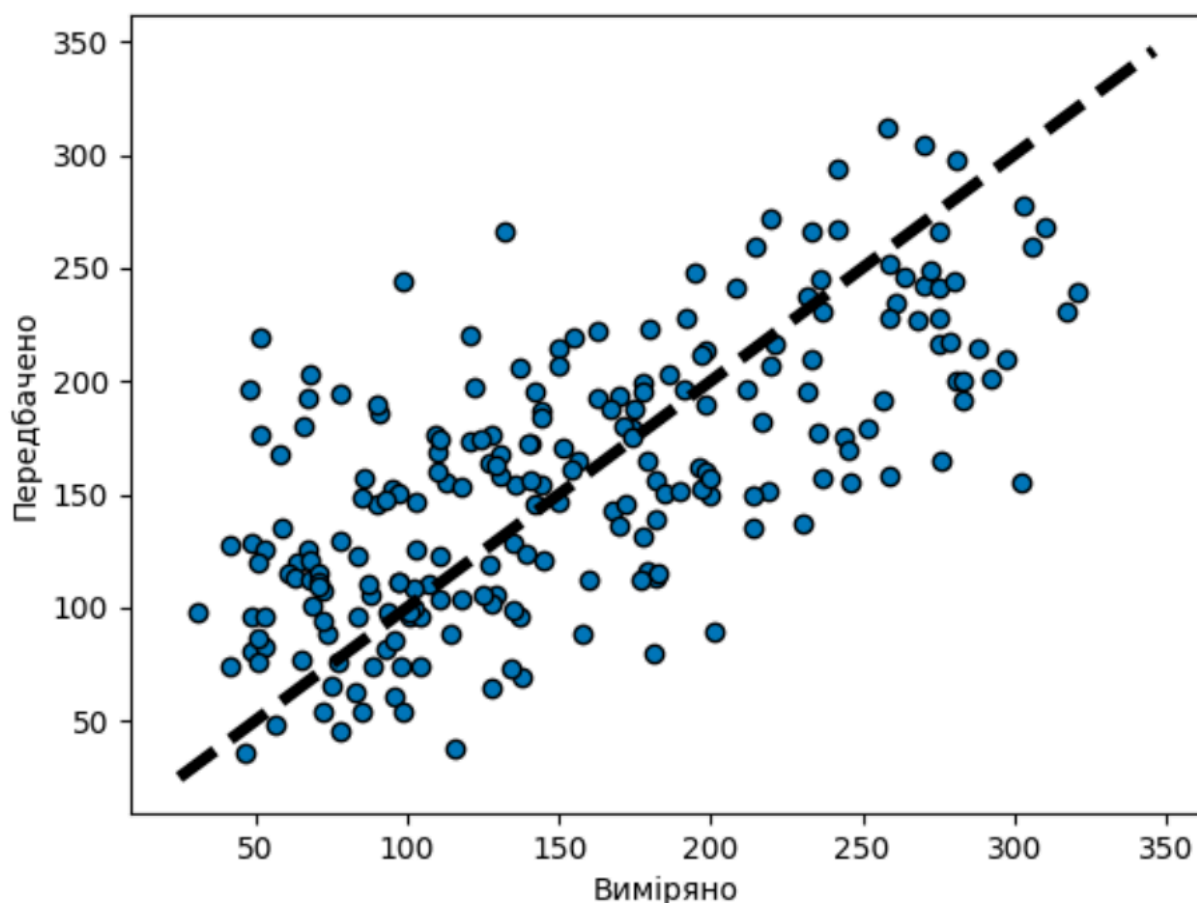


Рисунок 1.7 – Результат роботи програми у вигляді графіку

### Завдання 5. Самостійна побудова регресії

Згенеруйте свої випадкові дані обравши за списком відповідно свій варіант (згідно табл. 1.2) та виведіть їх на графік. Побудуйте по них модель лінійної регресії, виведіть на графік. Побудуйте по них модель поліноміальної регресії, виведіть на графік. Оцініть її якість.

Таблиця 1.2

№ за списком	15
№ варіанту	5

Варіант 5

$m = 100$

$X = 6 * \text{np.random.rand}(m, 1) - 3$

$y = 0.4 * X ** 2 + X + 4 + \text{np.random.randn}(m, 1)$



## Лістинг програми:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model
from sklearn.preprocessing import PolynomialFeatures

m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.4 * X ** 2 + X + 4 + np.random.randn(m, 1)

print(X[1], y[1])

X = X.reshape(-1, 1)
poly_features = PolynomialFeatures(degree=3, include_bias=False)
X_poly = poly_features.fit_transform(X)

lin_reg = linear_model.LinearRegression()
lin_reg.fit(X_poly, y)
print("Independent term =", lin_reg.coef_)
print("Estimated coefficients for regression problem =", lin_reg.intercept_)
y_pred = lin_reg.predict(X_poly)

fig, ax = plt.subplots()
ax.scatter(X, y, edgecolors=(0, 0, 0))
plt.plot(X, y_pred, color='black', linewidth=1)
ax.set_xlabel('x1')
ax.set_ylabel('y')
plt.show()
```

```
[-0.12150327] [2.66679436]
Independent term = [[ 1.07195116  0.36928166 -0.01970371]]
Estimated coefficients for regression problem = [4.10997883]
```

Рисунок 2.9 – Результат роботи програми

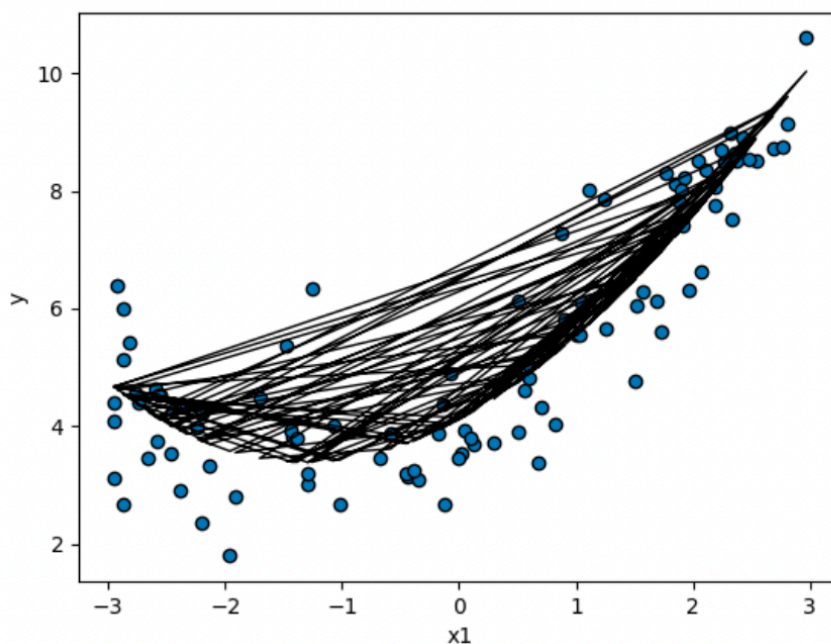


Рисунок 3.1 – Результат роботи програми у вигляді графіку

		Сівченко О. О.			Житомирська Політехніка.22.121.15.000 – Пр03	Арк.
		Філіпов В. О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

$$y = 0.4x^2 + x + 4 + \text{гаусовий шум}$$

$$y = 0.36x^2 + 1.07x + 4.10$$

## Завдання 6. Побудова кривих навчання

Побудуйте криві навчання для ваших даних у попередньому завданні.

Лістинг програми:

```
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model

m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.4 * X ** 2 + X + 4 + np.random.randn(m, 1)

X = X.reshape(-1, 1)

def plot_learning_curves(model, X, y):
    X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2)
    train_errors, val_errors = [], []
    for m in range(1, len(X_train)):
        model.fit(X_train[:m], y_train[:m])
        y_train_predict = model.predict(X_train[:m])
        y_val_predict = model.predict(X_val)
        train_errors.append(mean_squared_error(y_train_predict, y_train[:m]))
        val_errors.append(mean_squared_error(y_val_predict, y_val))
    fig, ax = plt.subplots()
    plt.ylim(0, 2)
    plt.plot(np.sqrt(train_errors), "r-+", linewidth=2, label="train")
    plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label="val")
    plt.show()

lin_reg = linear_model.LinearRegression()
plot_learning_curves(lin_reg, X, y)

from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import Pipeline
polynomial_regression = Pipeline([("poly_features",
                                   PolynomialFeatures(degree=10,
                                                       include_bias=False)), # degree=2
                                   ("lin_reg", linear_model.LinearRegression())
                                   ])
plot_learning_curves(polynomial_regression, X, y)
```

		Сівченко О. О.			Житомирська Політехніка. 22.121.15.000 – Лр03	Арк.
		Філіпов В. О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

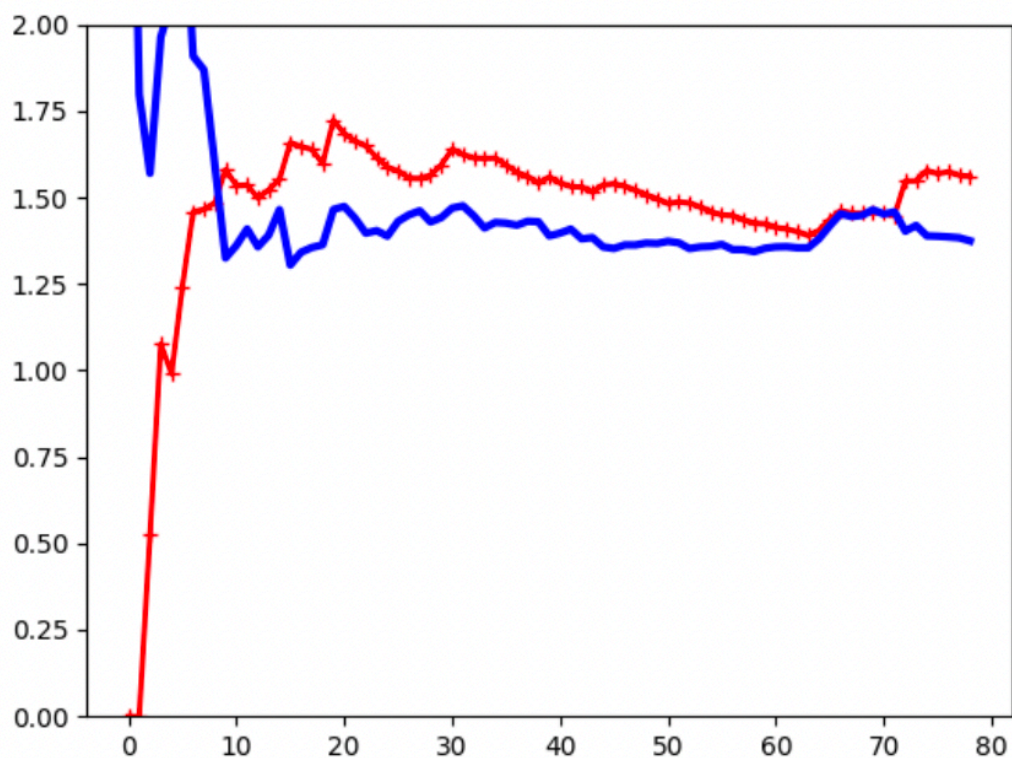


Рисунок 3.2 – Результат роботи програми у вигляді графіку

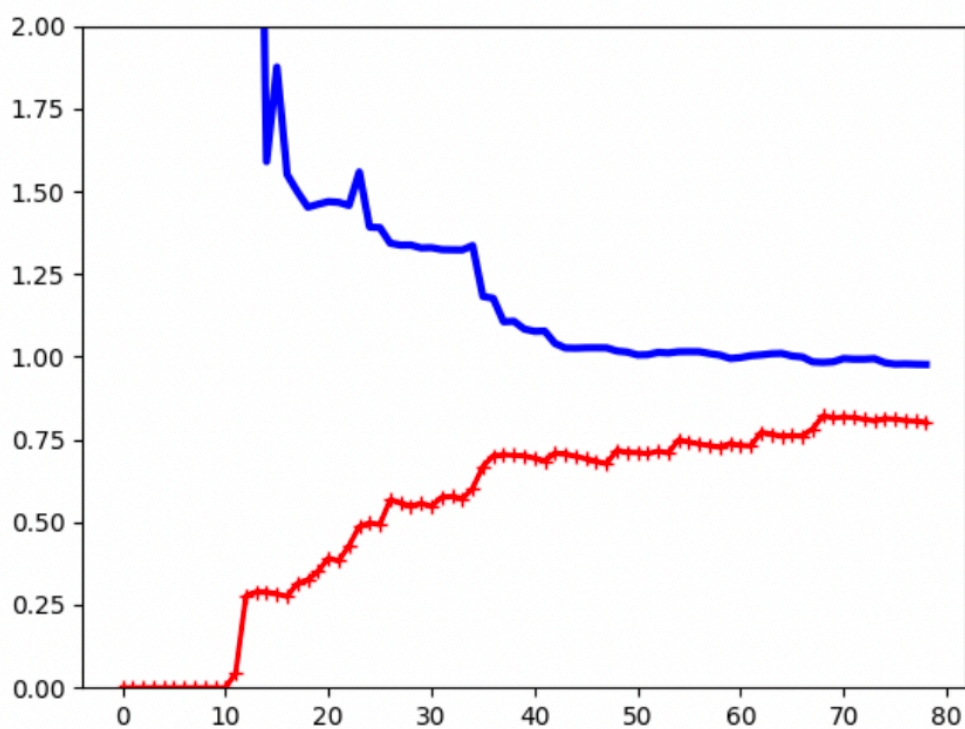


Рисунок 3.3 – Результат роботи програми у вигляді графіку

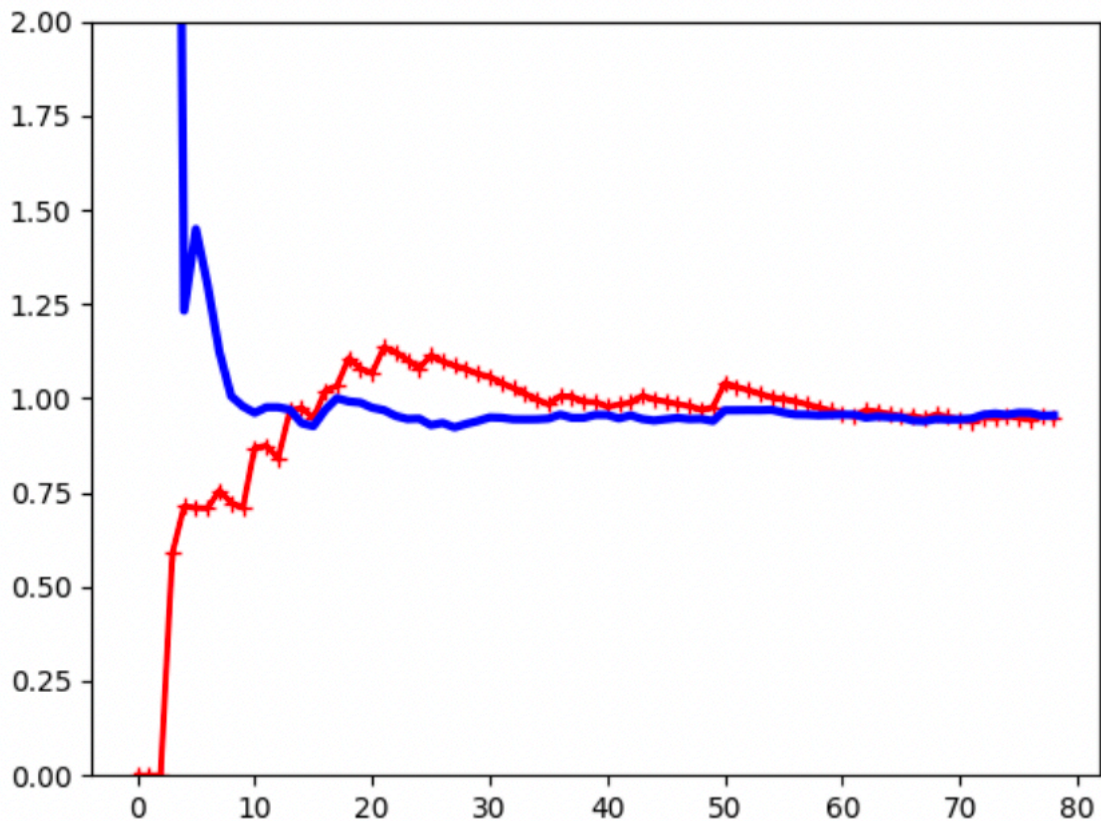


Рисунок 3.4 – Результат роботи програми у вигляді графіку

Посилання на GitHub: <https://github.com/SoylerProfile/SHI>

**Висновки:** в ході виконання лабораторної роботи було досліджено методи регресії даних у машинному навчанні, використовуючи спеціалізовані бібліотеки та мову програмування Python.

		Сівченко О. О.			Житомирська Політехніка.22.121.15.000 – Лр03	Арк.
		Філіпов В. О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		