

# ЛАБОРАТОРНА РОБОТА № 4 ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ

Варіант 13

Хід роботи:

**Завдання 1:** Побудувати регресійну модель на основі однієї змінної:

**Код програми:**

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

input_file = "Лабораторна робота 4/data_singlevar_regr.txt"
data = np.loadtxt(input_file, delimiter=",")

X, y = data[:, :-1], data[:, -1]
num_training = int(0.8 * len(X))
num_test = len(X) - num_training

X_train, y_train = X[:num_training], y[:num_training]
X_test, y_test = X[num_training:], y[num_training:]

regressor = linear_model.LinearRegression()
regressor.fit(X_train, y_train)
y_test_pred = regressor.predict(X_test)

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()

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))
```

					ДУ «Житомирська політехніка».21.121.5.000 - Лр1				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Корнійчук В. В.			Звіт з лабораторної роботи		Лім.	Арк.	Аркушів
Перевір.		Іванов Д. А.						1	
Керівник							ФІКТ Гр. ІПЗ-21-5[2]		
Н. контр.									
Зав. каф.									

```

output_model_file = "Лабораторна робота 4/model.pkl"
with open(output_model_file, "wb") as f:
    pickle.dump(regressor, f)

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),
)

```

### Виконання:

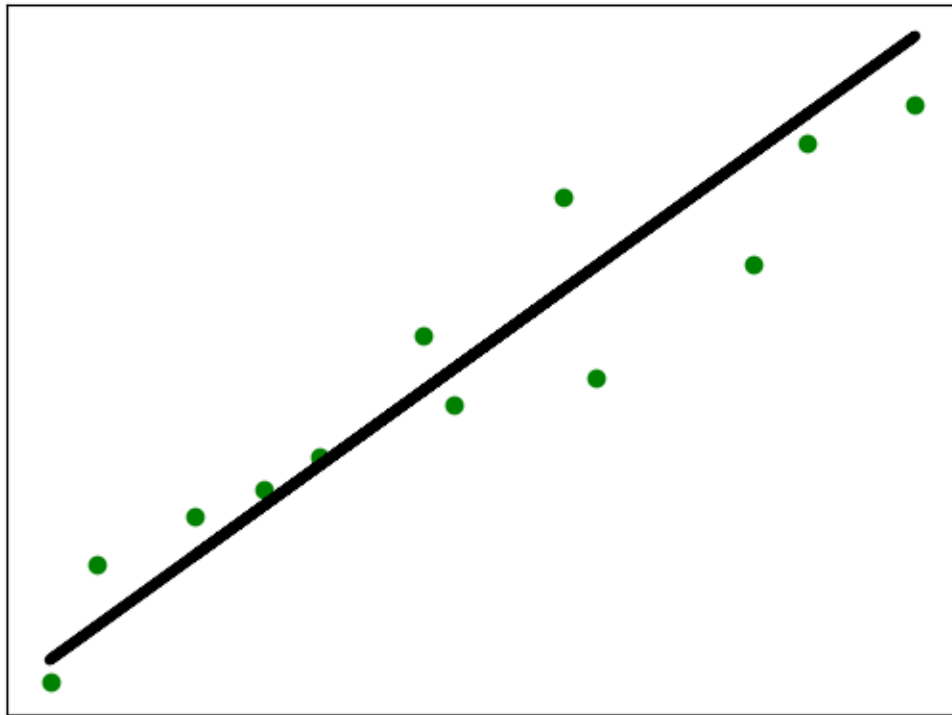


Рисунок 1.1 – Графік функції

```

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 1.2 – Виконання програми

**Висновок:** Лінійний регресор добре працює на основі однієї змінної, незважаючи на поодинокі відхилення від лінії – проблема може бути в наявності шуму в даних.

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

## Завдання 2: Побудувати регресійну модель на основі однієї змінної:

№ за списком	1	2	3	4	5	6	7	8	9	10
№ варіанту	1	2	3	4	5	1	2	3	4	5

№ за списком	11	12	13	14	15	16	17	18	19	20
№ варіанту	1	2	3	4	5	1	2	3	4	5

№ за списком	21	22	23	24	25	26	27	28	29	30
№ варіанту	1	2	3	4	5	1	2	3	4	5

Варіант 1 файл: data\_regr\_1.txt

Варіант 2 файл: data\_regr\_2.txt

Варіант 3 файл: data\_regr\_3.txt

Варіант 4 файл: data\_regr\_4.txt

Варіант 5 файл: data\_regr\_5.txt

### Код програми:

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

input_file = "Лабораторна робота 4/data_regr_3.txt"
data = np.loadtxt(input_file, delimiter=",")

X, y = data[:, :-1], data[:, -1]
num_training = int(0.8 * len(X))
num_test = len(X) - num_training

X_train, y_train = X[:num_training], y[:num_training]
X_test, y_test = X[num_training:], y[num_training:]

regressor = linear_model.LinearRegression()
regressor.fit(X_train, y_train)
y_test_pred = regressor.predict(X_test)

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()

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(
```

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

    "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))

output_model_file = "Лабораторна робота 4/model.pkl"
with open(output_model_file, "wb") as f:
    pickle.dump(regressor, f)

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),
)

```

### Виконання:

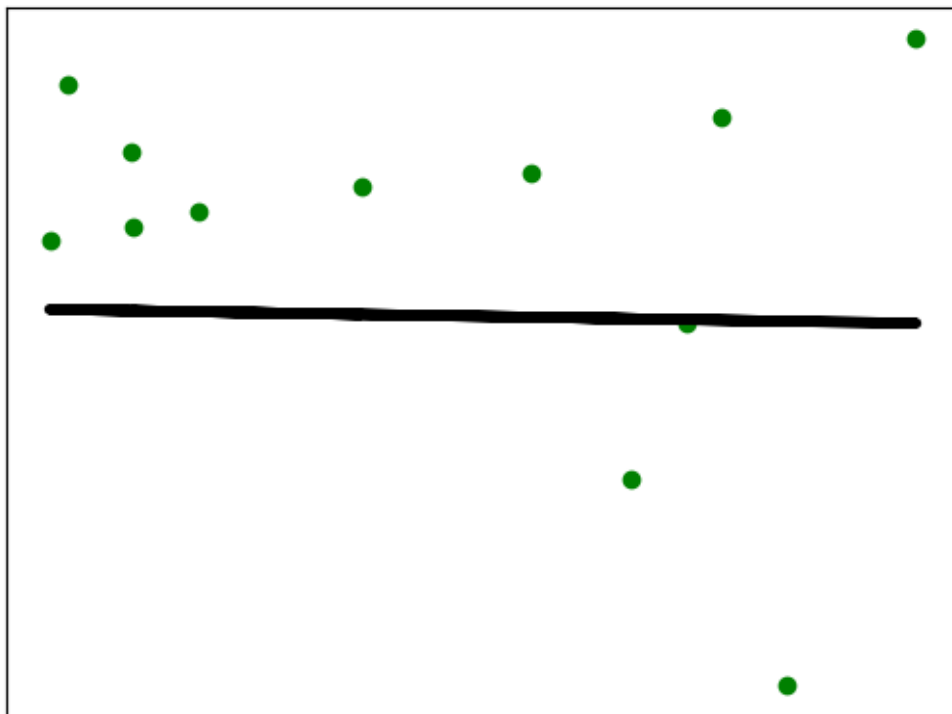


Рисунок 2.1 – Графік функції

```

Linear regressor performance:
Mean absolute error = 3.59
Mean squared error = 17.39
Median absolute error = 3.39
Explain variance score = 0.02
R2 score = -0.16

New mean absolute error = 3.59

```

Рисунок 2.2 – Виконання програми

**Висновок:** Модель сильно не підходить для даних – про це найочивидніше свідчить показник R2, що є негативним. Це означає, що модель передбачує гірше ніж будь-які випадкові прогнози.

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

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

**Код програми:**

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

def parse_data(filename):
    data = np.loadtxt(filename, delimiter=",")

    X, y = data[:, :-1], data[:, -1]
    num_training = int(0.8 * len(X))
    num_test = len(X) - num_training

    X_train, y_train = X[:num_training], y[:num_training]
    X_test, y_test = X[num_training:], y[num_training:]

    return X_train, y_train, X_test, y_test

def evaluate_performance(y, y_pred):
    print("Mean absolute error =", round(sm.mean_absolute_error(y, y_pred), 2))
    print("Mean squared error =", round(sm.mean_squared_error(y, y_pred), 2))
    print(
        "Median absolute error =",
        round(sm.median_absolute_error(y, y_pred), 2),
    )
    print(
        "Explain variance score =",
        round(sm.explained_variance_score(y, y_pred), 2),
    )
    print("R2 score =", round(sm.r2_score(y, y_pred), 2))

X_train, y_train, X_test, y_test = parse_data(
    "Лабораторна робота 4/data_multivar_regr.txt"
)

regressor = linear_model.LinearRegression()
regressor.fit(X_train, y_train)
y_test_pred = regressor.predict(X_test)

print("Linear regressor performance:")
evaluate_performance(y_test, y_test_pred)

polynomial = PolynomialFeatures(degree=10)
X_train_transformed = polynomial.fit_transform(X_train)
datapoint = [[7.75, 6.35, 5.56]]
```

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```
poly_datapoint = polynomial.fit_transform(datapoint)
poly_linear_model = linear_model.LinearRegression()
poly_linear_model.fit(X_train_transformed, y_train)

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

### Виконання:

```
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:
[41.46543157]
```

Рисунок 3.1 – Виконання програми

**Висновок:** Обидві моделі добре підходять для цих даних, проте поліноміальна регресія прогнозує краще – ознака складних трендів у наборі даних.

**Завдання 4:** Розробіть лінійний регресор, використовуючи набір даних по діабету.

### Код програми:

```
import matplotlib.pyplot as plt
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
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.5, random_state=0)

regr = linear_model.LinearRegression()
regr.fit(Xtrain, ytrain)
ypred = regr.predict(Xtest)

print("Regr coef =", regr.coef_)
print("Regr intercept =", regr.intercept_)
print("Mean absolute error =", round(mean_absolute_error(ytrain, ypred), 2))
print("Mean squared error =", round(mean_squared_error(ytrain, ypred), 2))
print("R2 score =", round(r2_score(ytrain, ypred), 2))

fig, ax = plt.subplots()
ax.scatter(ytest, ypred, edgecolors=(0, 0, 0))
```

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```
ax.plot([y.min(), y.max()], [y.min(), y.max()], "k--", lw=4)
ax.set_xlabel("Виміряно")
ax.set_ylabel("Передбачено")
plt.show()
```

**Виконання:**

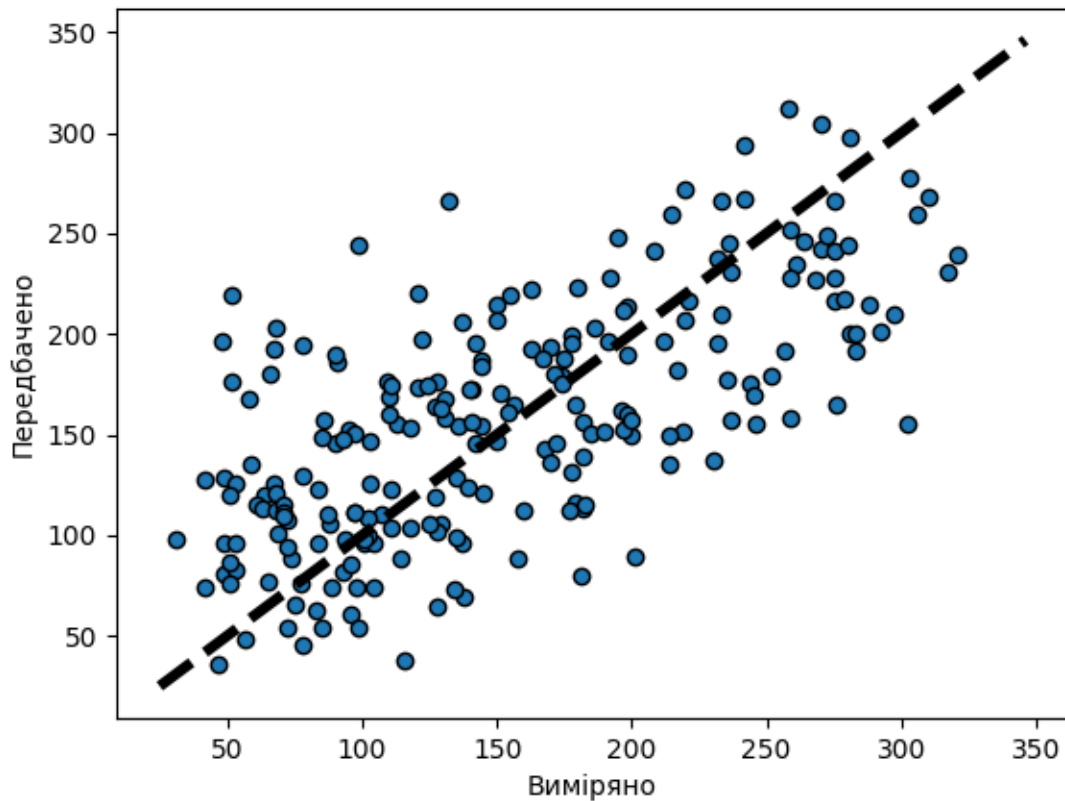


Рисунок 4.1 – Графік функції

```
Regr coef = [ -20.4047621 -265.88518066 564.65086437 325.56226865 -692.16120333
395.55720874 23.49659361 116.36402337 843.94613929 12.71856131]
Regr intercept = 154.3589285280134
Mean absolute error = 81.37
Mean squared error = 9521.15
R2 score = -0.49
```

Рисунок 4.2 – Виконання програми

**Висновок:** Модель сильно не підходить для цих даних.

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

**Завдання 5:** Згенеруйте свої випадкові дані обравши за списком відповідно свій варіант 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.

№ за списком	11	12	13	14	15	16	17	18	19	20
№ варіанту	1	2	3	4	5	6	7	8	9	10

### Код програми:

```
import numpy as np
import matplotlib.pyplot as plt
import sklearn.metrics as sm
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures

def evaluate_performance(y, y_pred):
    print("Mean absolute error =", round(sm.mean_absolute_error(y, y_pred), 2))
    print("Mean squared error =", round(sm.mean_squared_error(y, y_pred), 2))
    print(
        "Median absolute error =",
        round(sm.median_absolute_error(y, y_pred), 2),
    )
    print(
        "Explain variance score =",
        round(sm.explained_variance_score(y, y_pred), 2),
    )
    print("R2 score =", round(sm.r2_score(y, y_pred), 2))

def show_plot(y, y_pred):
    fig, ax = plt.subplots()
    ax.scatter(y, 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()

def generate_data():
    m = 100
    X = 6 * np.random.rand(m, 1) - 4
    y = 0.5 * X**2 + X + 2 + np.random.randn(m, 1)
    return X, y

if __name__ == "__main__":
    X, y = generate_data()
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=5
    )
```

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

polynomial = PolynomialFeatures(degree=2, include_bias=False)
X_train_transformed = polynomial.fit_transform(X_train)

model = linear_model.LinearRegression()
model.fit(X_train_transformed, y_train)

X_test_transformed = polynomial.fit_transform(X_test)
y_test_predict = model.predict(X_test_transformed)

print("Polynomial coefficient:\n", model.coef_, model.intercept_)
print("\nPolynomial regressor performance:")
evaluate_performance(y_test, y_test_predict)
show_plot(y_test, y_test_predict)

```

### Виконання:

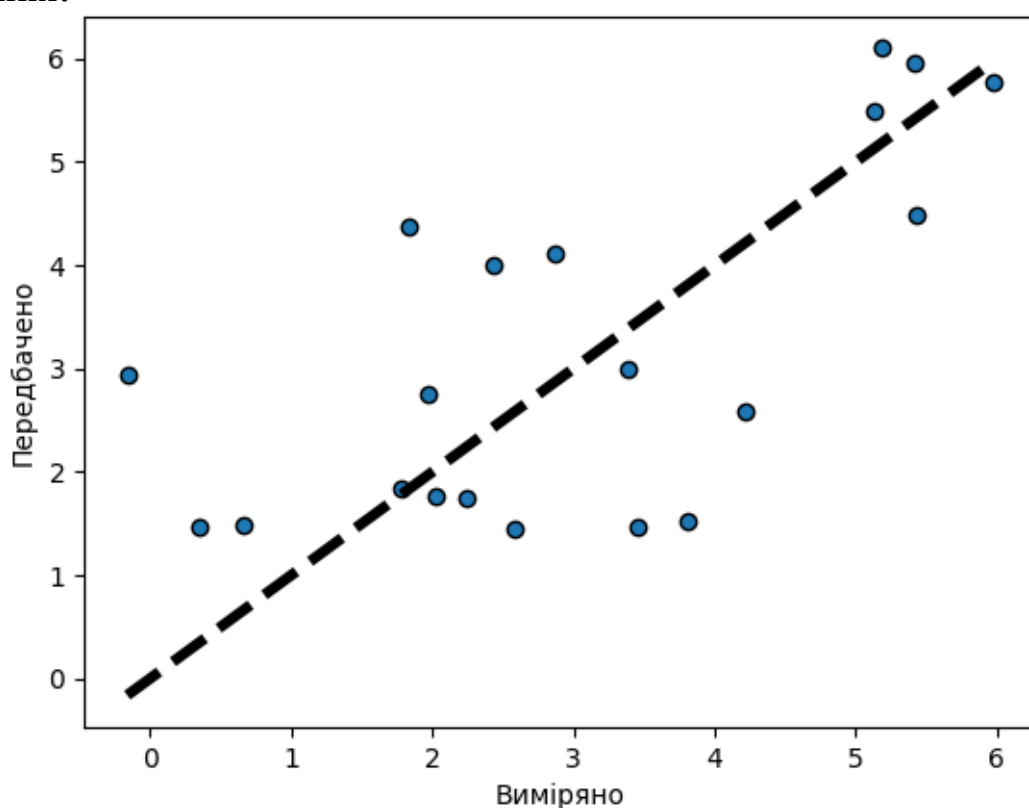


Рисунок 5.1 – Графік функції

```

Polynomial coefficient:
[1.78295097] [[1.07204626 0.53689412]]

Polynomial regressor performance:
Mean absolute error = 0.91
Mean squared error = 1.28
Median absolute error = 0.77
Explain variance score = 0.58
R2 score = 0.53

```

Рисунок 5.2 – Виконання програми

Модель у вигляді математичного рівняння:  $y = 0.5x_1^2 + x_2 + 2 + \text{гаусовий шум}$ . Ко-ефіцієнти нашої моделі:  $y = 0.536x_1^2 + 1.072x_2 + 1.78$

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

**Завдання 6:** Побудуйте криві навчання для ваших даних у попередньому завданні.

**Код програми:**

```
import numpy as np
import matplotlib.pyplot as plt
import sklearn.metrics as sm
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import Pipeline

from LR_4_task_5 import generate_data

def plot_learning_curves(model, X_train, y_train, X_test, Y_test):
    train_errors = []
    test_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_test_predict = model.predict(X_test)
        train_errors.append(sm.mean_squared_error(y_train_predict, y_train[:m]))
        test_errors.append(sm.mean_squared_error(y_test_predict, y_test))

    plt.plot(np.sqrt(train_errors), "r-+", linewidth=2, label="train")
    plt.plot(np.sqrt(test_errors), "b-", linewidth=3, label="test")
    plt.show()

X, y = generate_data()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)

lin_reg = linear_model.LinearRegression()
plot_learning_curves(lin_reg, X_train, y_train, X_test, y_test)

polynomial_10_degree_regression = Pipeline(
    [
        ("poly_features", PolynomialFeatures(degree=10, include_bias=False)),
        ("lin_reg", linear_model.LinearRegression()),
    ]
)
plot_learning_curves(polynomial_10_degree_regression, X_train, y_train, X_test, y_test)

polynomial_2_degree_regression = Pipeline(
    [
        ("poly_features", PolynomialFeatures(degree=2, include_bias=False)),
        ("lin_reg", linear_model.LinearRegression()),
    ]
)
```

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```
)
plot_learning_curves(polynomial_2_degree_regression, X_train, y_train, X_test,
y_test)
```

**Виконання:**

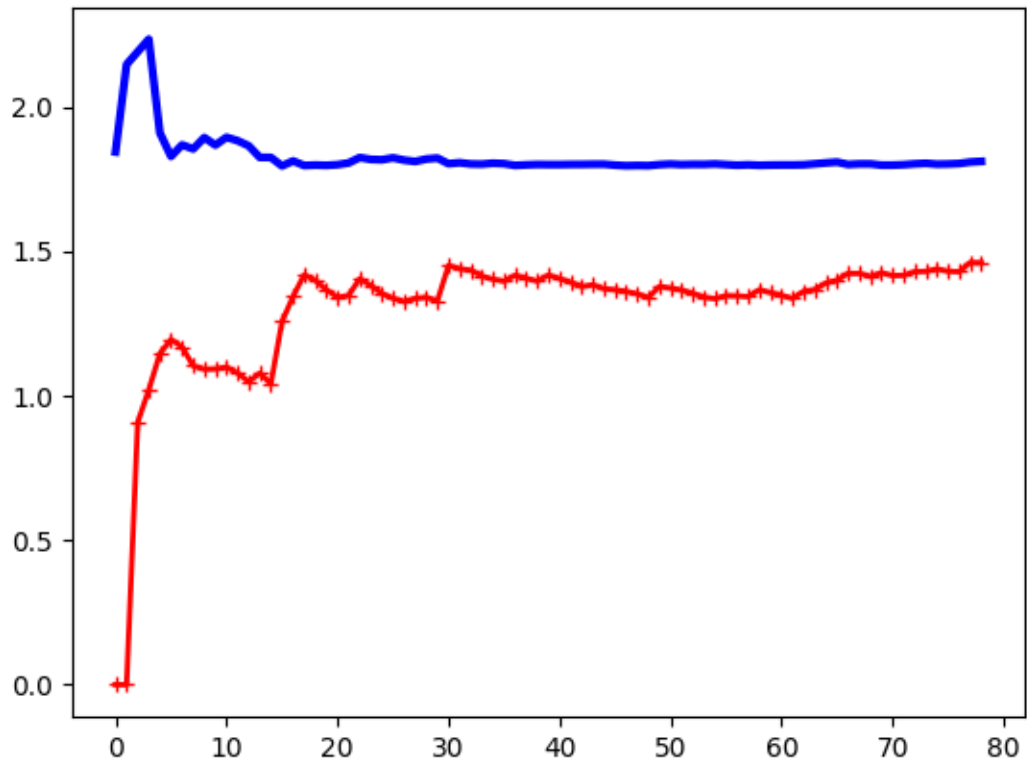


Рисунок 6.1 – Крива навчання лінійної моделі

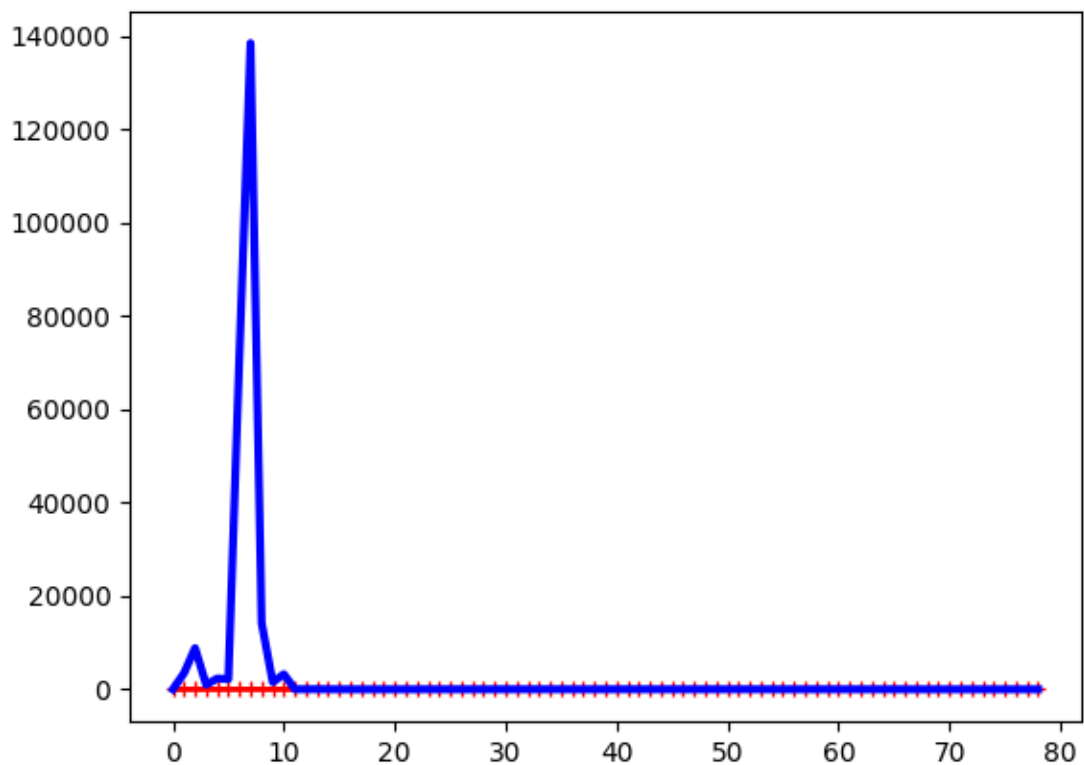


Рисунок 6.2 – Крива навчання для поліноміальної моделі

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

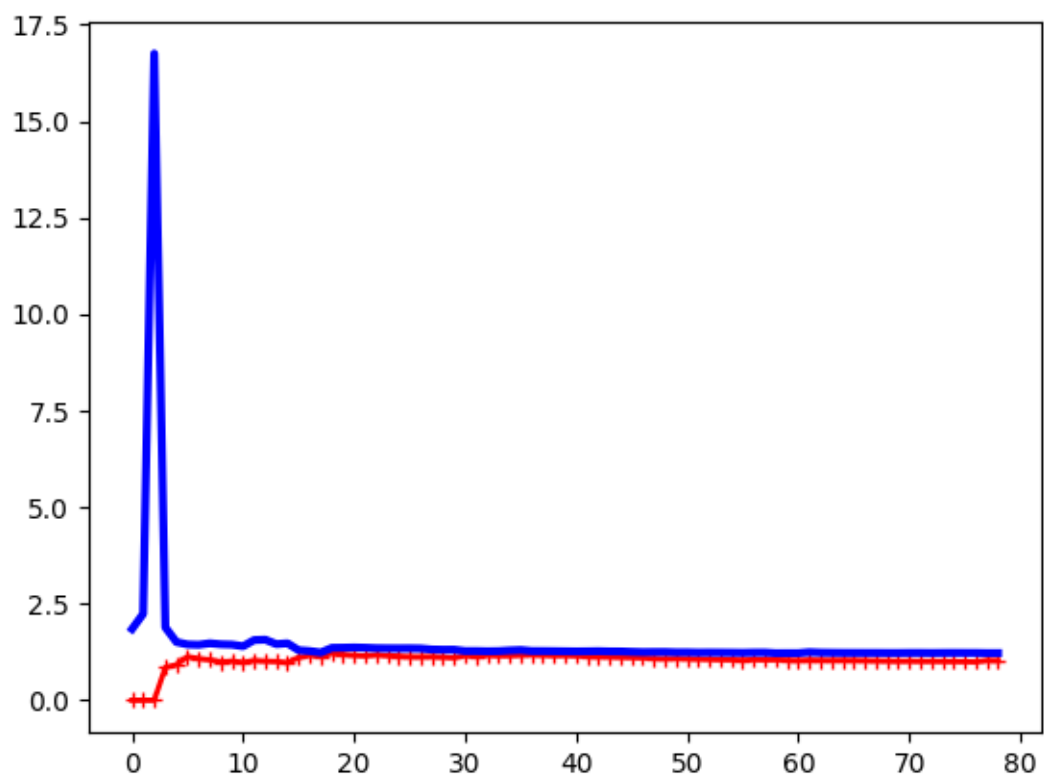


Рисунок 6.2 – Крива навчання для моделі 2-го ступеня

**Посилання на GitHub:** <https://github.com/ipz215kvv/artificial-intelligence-systems>

		Корнійчук В. В.			ДУ «Житомирська політехніка».21.121.5.000 - Лр1	Арк.
		Іванов Д. А.				12
Змн.	Арк.	№ докум.	Підпис	Дата		