

## ЛАБОРАТОРНА РОБОТА № 4

### Дослідження методів регресії

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

### Хід роботи:

#### Завдання 1:

#### Лістинг коду:

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

input_file = "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),
)
```

					ДУ «Житомирська політехніка».24.121.8.000 – Лр4						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Гейна В. С.			Звіт з лабораторної роботи			Літ.	Арк.	Аркушів	
Перевір.		Іванов Д. А.							1	11	
Керівник								ФІКТ Гр. ППЗ-21-5			
Н. контр.											
Зав. каф.											

```
print("R2 score =", round(sm.r2_score(y_test, y_test_pred), 2))

output_model_file = "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),
)
```

```
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
```

Рис. 1. Результат виконання програми

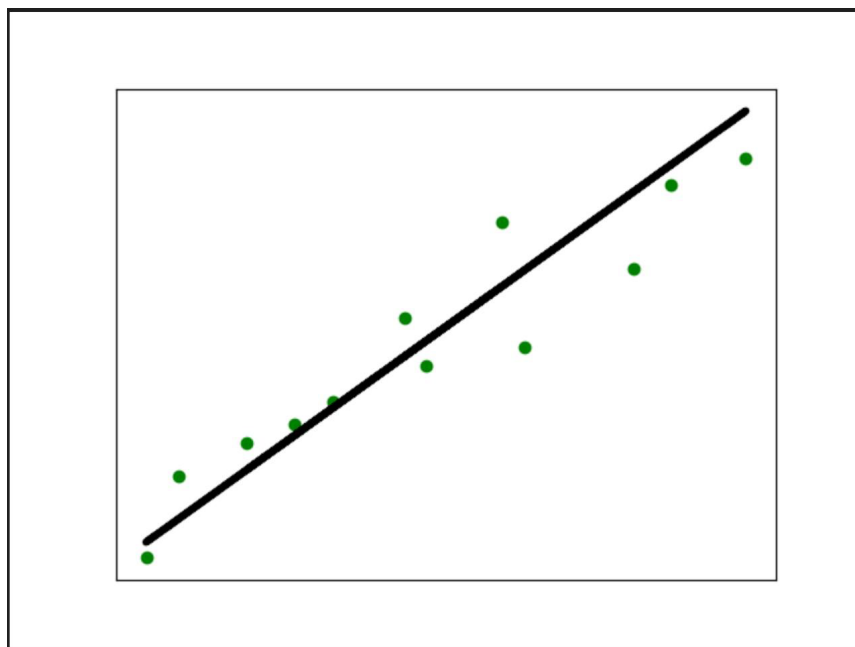


Рис. 2. Результат виконання програми

Лінійний регресор ефективно моделює залежність на основі однієї змінної, хоча окремі відхилення від лінії можуть свідчити про наявність шуму в даних.

		Гейна В. С.			ДУ «Житомирська політехніка». 24.121.8.000 – Лр4	Арк.
		Іванов Д. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

## Завдання 2:

### Лістинг коду:

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

input_file = "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("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 = "model_2.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),
)
```

```
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
```

Рис. 3. Результат виконання програми

		Гейна В. С.			ДУ «Житомирська політехніка».24.121.8.000 – Лр4	Арк.
		Іванов Д. А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

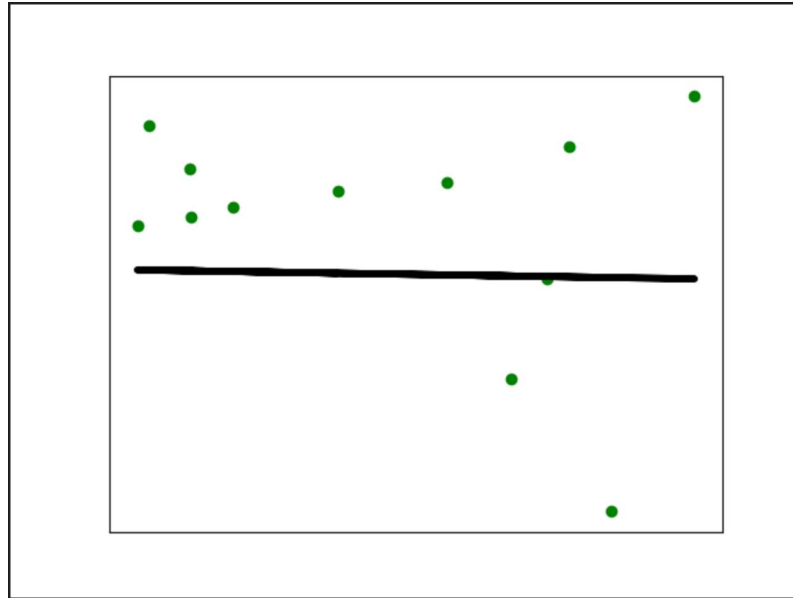


Рис. 4. Результат виконання програми

Модель значно не відповідає даним, що найбільш очевидно демонструє негативний показник  $R^2$ . Це вказує на те, що точність передбачень моделі є гіршою, ніж у випадкових прогнозів.

### Завдання 3:

Лістинг коду:

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

data = np.loadtxt("data_multivar_regr.txt", 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)
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),
)
```

		Гейна В. С.			ДУ «Житомирська політехніка». 24.121.8.000 – Лр4	Арк.
		Іванов Д. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

)
print("R2 score =", round(sm.r2_score(y_test, y_test_pred), 2))
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)
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:
[41.45615457]

```

Рис. 5. Результат виконання програми

Обидві моделі добре відповідають цим даним, однак поліноміальна регресія демонструє кращу точність прогнозування, що вказує на наявність складних трендів у наборі даних.

#### Завдання 4:

Лістинг коду:

```

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)

```

		Гейна В. С.			ДУ «Житомирська політехніка». 24.121.8.000 – Лр4	Арк.
		Іванов Д. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

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

```

```

Regr coef = [ -20.4047621  -265.88518066  564.65086437  325.56226865 -692.16120333
  395.55720874  23.49659361  116.36402337  843.94613929  12.71856131]
Regr intercept = 154.35892852801342
Mean absolute error = 81.37
Mean squared error = 9521.15
R2 score = -0.49

```

Рис. 6. Результат виконання програми

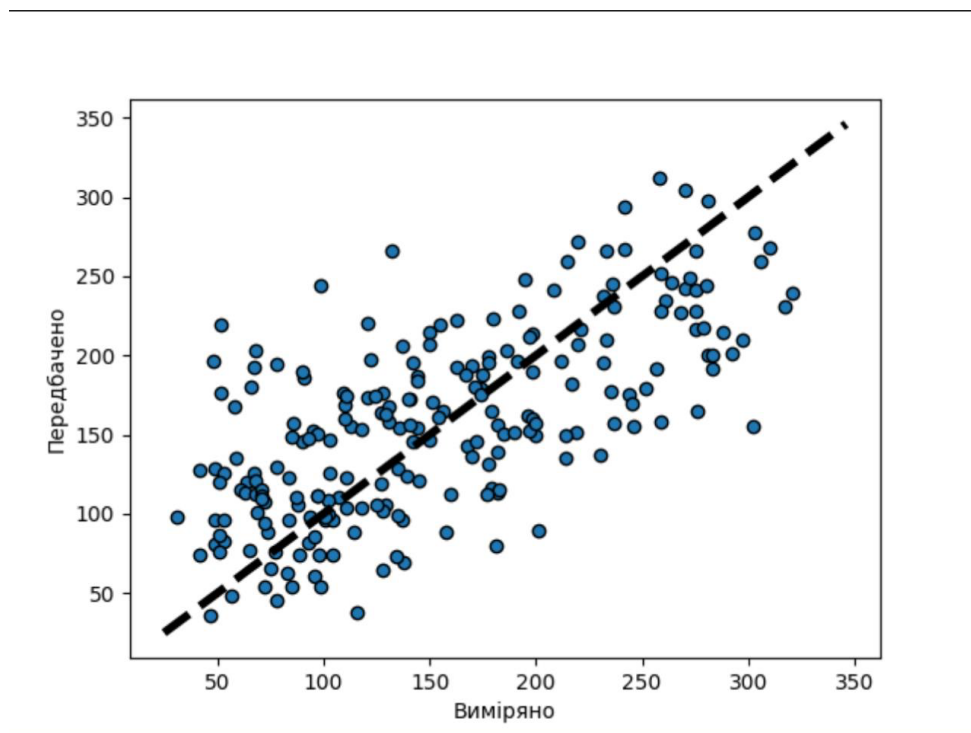


Рис. 7. Результат виконання програми

Модель виявляється непридатною для цих даних.

		Гейна В. С.			ДУ «Житомирська політехніка».24.121.8.000 – Лр4	Арк.
		Іванов Д. А.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

## Завдання 5:

### Лістинг коду:

```
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

m = 100
X = 6 * np.random.rand(m, 1) - 4
y = 0.5 * X**2 + X + 2 + np.random.randn(m, 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
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:")
print("Mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_predict), 2))
print("Mean squared error =", round(sm.mean_squared_error(y_test, y_test_predict), 2))
print(
    "Median absolute error =",
    round(sm.median_absolute_error(y_test, y_test_predict), 2),
)
print("Explain variance score =", round(sm.explained_variance_score(y_test, y_test_predict), 2))
print("R2 score =", round(sm.r2_score(y_test, y_test_predict), 2))
fig, ax = plt.subplots()
ax.scatter(y_test, y_test_predict, edgecolors=(0, 0, 0))
ax.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], "k--", lw=4)
ax.set_xlabel("Виміряно")
ax.set_ylabel("Передбачено")
plt.show()
```

```
Polynomial coefficient:
[[1.10992721 0.49224886]] [2.15279682]

Polynomial regressor performance:
Mean absolute error = 0.79
Mean squared error = 0.97
Median absolute error = 0.66
Explain variance score = 0.57
R2 score = 0.55
```

Рис. 8. Результат виконання програми

		Гейна В. С.			ДУ «Житомирська політехніка». 24.121.8.000 – Лр4	Арк.
		Іванов Д. А.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

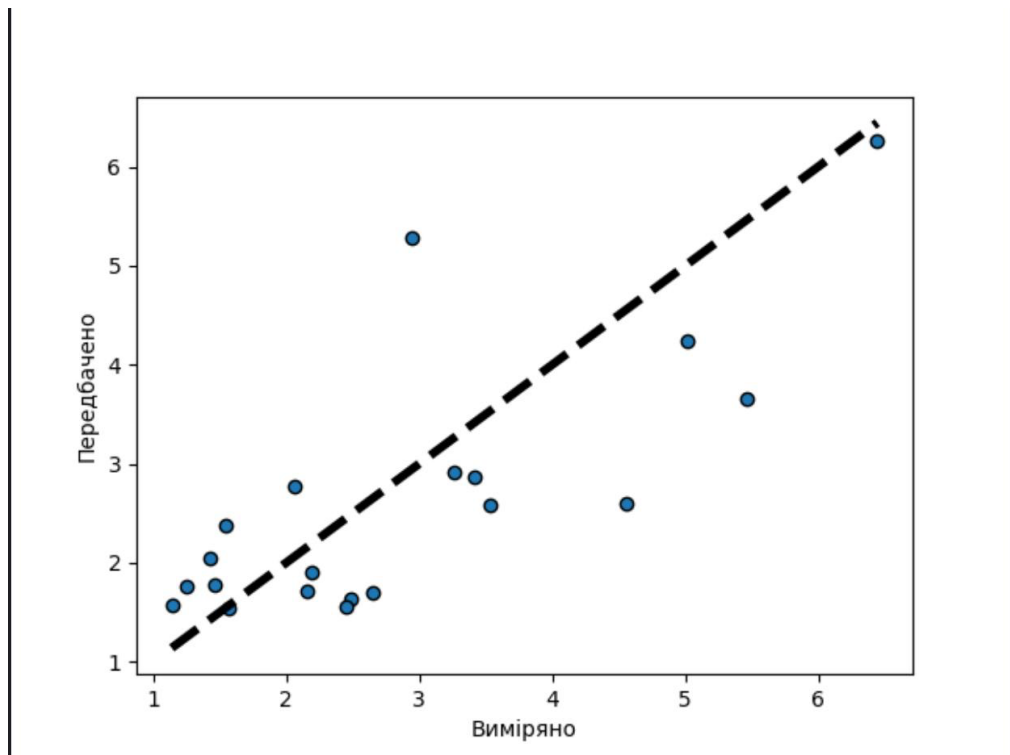


Рис. 9. Результат виконання програми

Модель у вигляді математичного рівняння:  $y = 0.5x_1^2 + x_2 + 2 + \text{гаусовий шум}$ .

Коефіцієнти нашої моделі:  $y = 0.536x_1^2 + 1.072x_2 + 1.78$

### Завдання 6:

Лістинг коду:

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

def generate_data(m, random_state=None):
    if random_state is not None:
        np.random.seed(random_state)
    X = 6 * np.random.rand(m, 1) - 4
    y = 0.5 * X ** 2 + X + 2 + np.random.randn(m, 1)
    return X, y

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(mean_squared_error(y_train_predict, y_train[:m]))
test_errors.append(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.xlabel("Training set size")
plt.ylabel("RMSE")
plt.legend()
plt.title("Learning Curves")
plt.show()

def create_polynomial_pipeline(degree):
    return Pipeline([
        ("poly_features", PolynomialFeatures(degree=degree, include_bias=False)),
        ("lin_reg", LinearRegression()),
    ])

X, y = generate_data(100, random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)

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

degrees = [10, 2]
for degree in degrees:
    poly_pipeline = create_polynomial_pipeline(degree)
    plot_learning_curves(poly_pipeline, X_train, y_train, X_test, y_test)

```

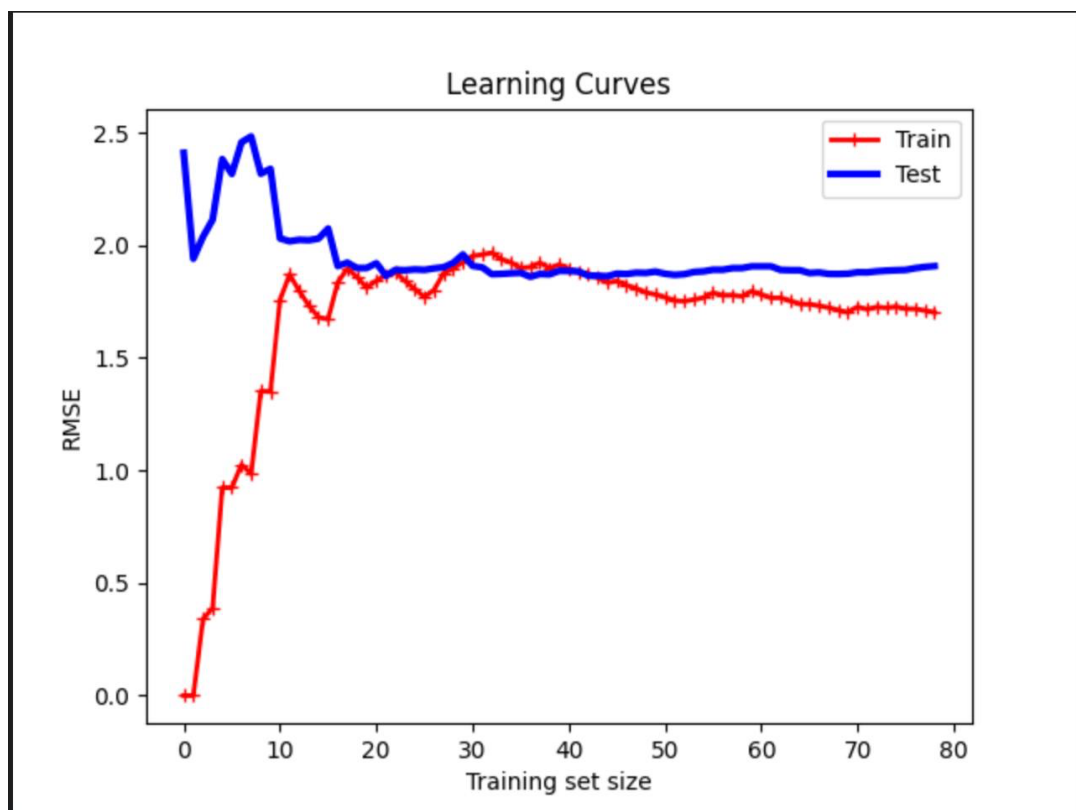


Рис. 10. Результат виконання програми

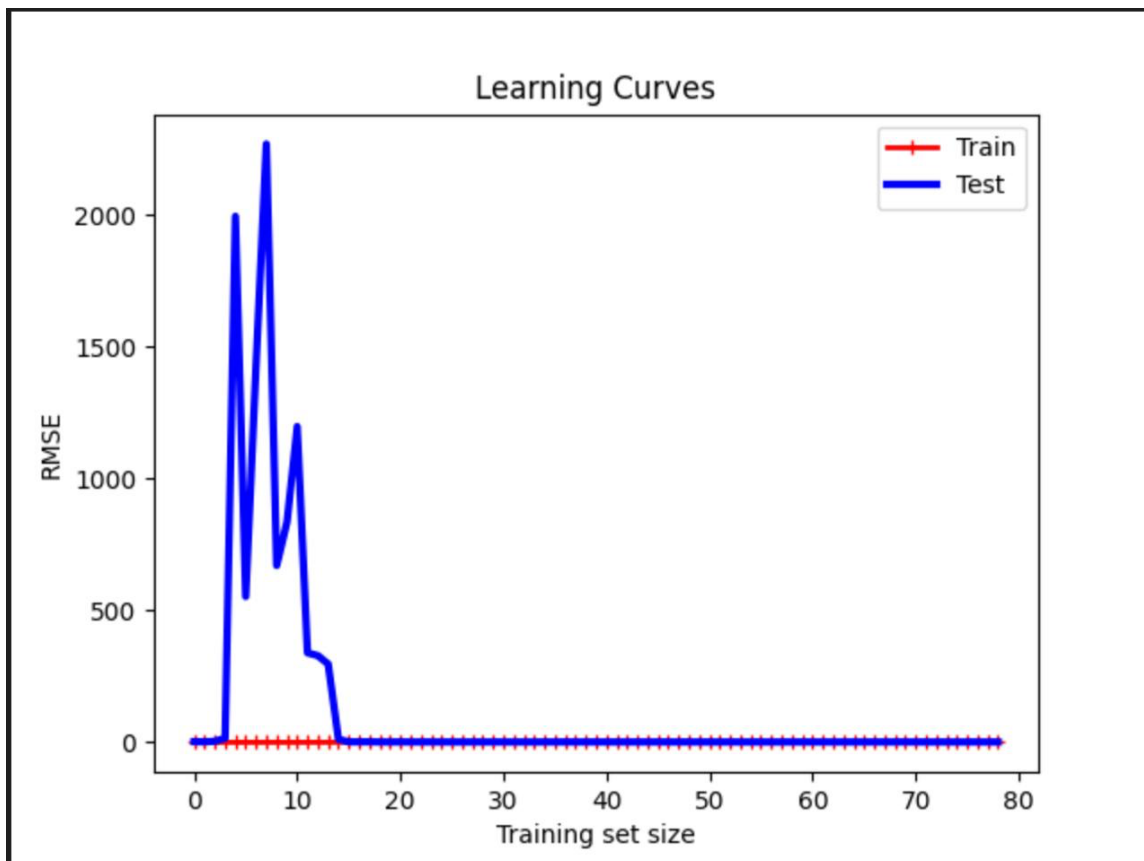


Рис. 11. Результат виконання програми

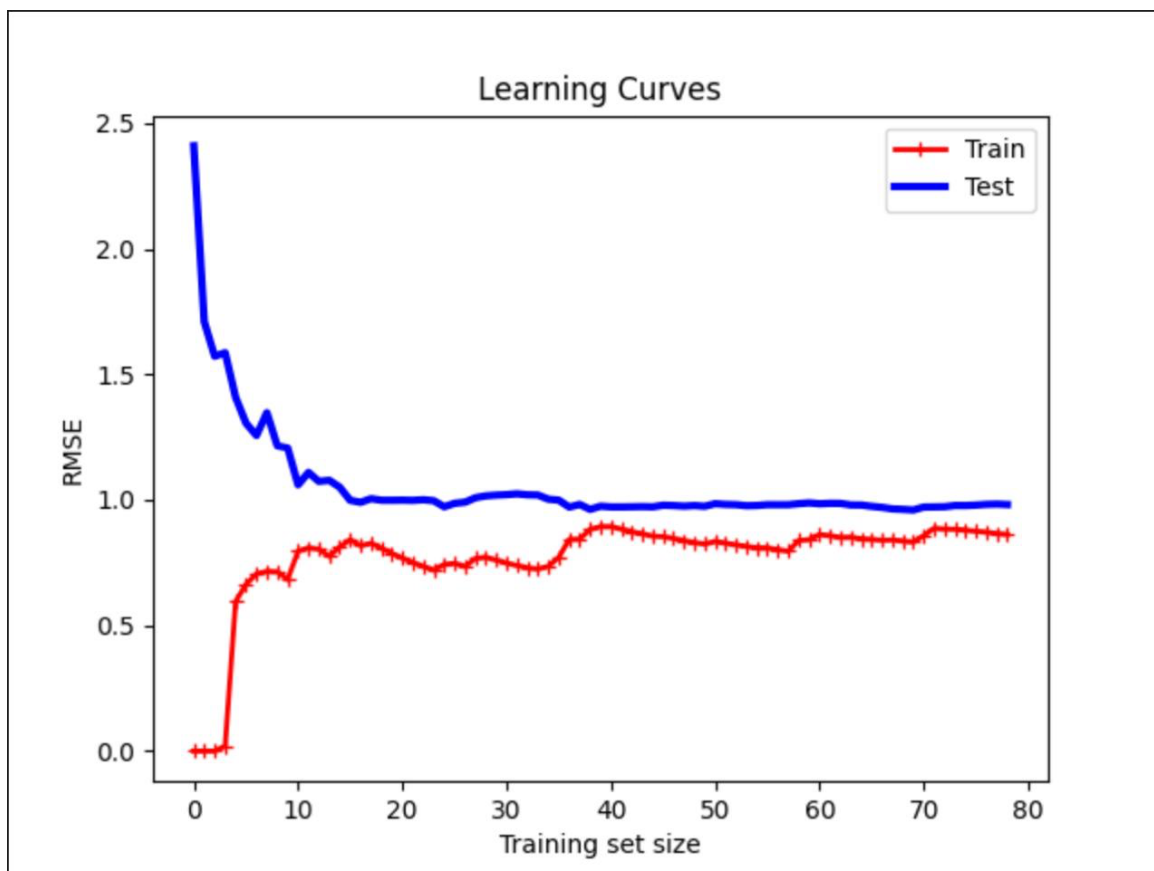


Рис. 12. Результат виконання програми

		Гейна В. С.			ДУ «Житомирська політехніка». 24.121.8.000 – Лр4	Арк.
		Іванов Д. А.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Посилання на репозиторій на GitHub:  
<https://github.com/vladyslavgeyna/artificial-intelligence-systems/tree/main/lab3>.

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

		Гейна В. С.			ДУ «Житомирська політехніка».24.121.8.000 – Лр4	Арк.
		Іванов Д. А.				11
Змн.	Арк.	№ докум.	Підпис	Дата		