

Лабораторна робота №4

ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ

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

Github - <https://github.com/Oleksi89/ai-systems>

Хід роботи

Завдання 2.1. 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

matplotlib.use('TkAgg')
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)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Васянович О.А.			Звіт з лабораторної роботи №4			Літ.		Арк.	Аркушів
Перевір.		Маєвський О.Є.								1	14
Реценз.								ФІКТ, гр. ІПЗ-22-3			
Н. Контр.											
Зав.каф.											

```

# 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()
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
with open(output_model_file, 'rb') as f:
    regressor_model = pickle.load(f)

# Прогнозування за допомогою моделі
y_test_pred_new = regressor_model.predict(X_test)
print("\nNew mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_pred_new), 2))

```

Результати:

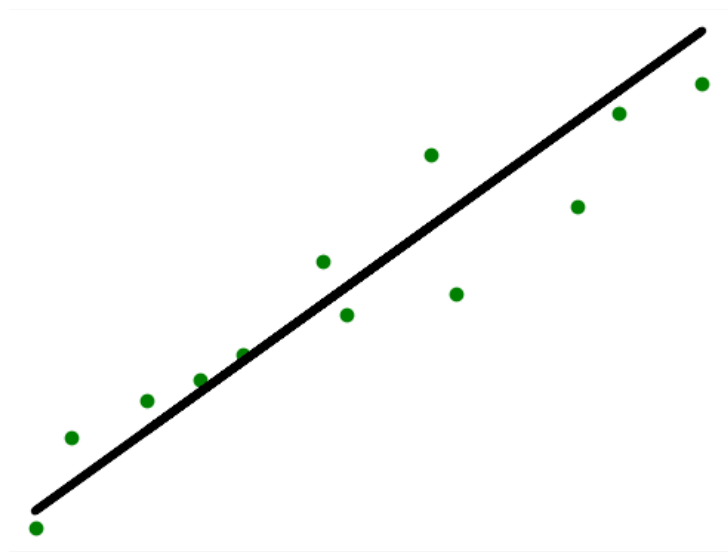


Рис.1 Графік функції

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		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.2 Результати оцінки якості

За результатами оцінки якості роботи регресора на тестових даних:

Коефіцієнт детермінації R^2 становить 0.86, що свідчить про високу точність апроксимації даних моделлю (пояснено 86% варіації)

Середня абсолютна похибка MAE дорівнює 0.59

Середньоквадратична похибка MSE дорівнює 0.49

Тестування збереження моделі за допомогою модуля pickle підтвердило коректність серіалізації, похибка відновленої моделі збігається з початковою MAE = 0.59. Побудований графік візуалізує лінійну залежність вхідних даних.

Завдання 2.2. Передбачення за допомогою регресії однієї змінної

Побудувати регресійну модель на основі однієї змінної. Використовувати вхідні дані відповідно свого 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

matplotlib.use('TkAgg')
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]

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

# 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()
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
with open(output_model_file, 'rb') as f:
    regressor_model = pickle.load(f)

# Прогнозування за допомогою моделі
y_test_pred_new = regressor_model.predict(X_test)
print("\nNew mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_pred_new), 2))

```

Результати:

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

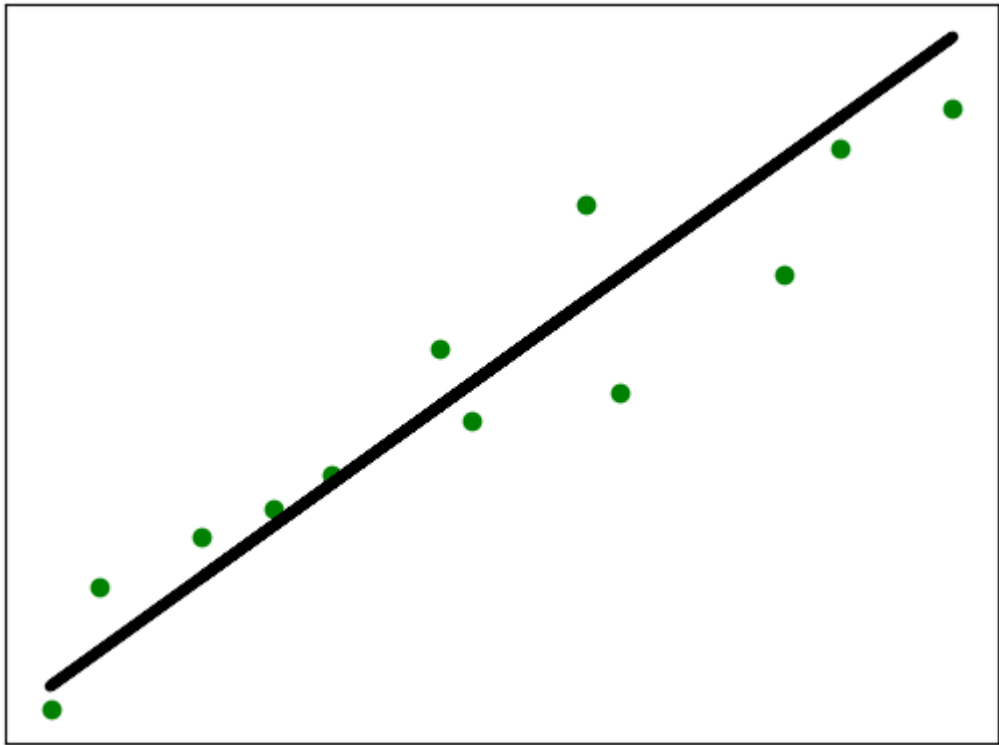


Рис.3 Графік функції

```
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.4 Результати оцінки якості

Оскільки файл вхідних даних варіанту ідентичний набору даних із завдання 2.1, отримані результати навчання та графіки повністю збігаються з попереднім пунктом.

Показники якості моделі:

Коефіцієнт детермінації R^2 : 0.86

Середня абсолютна похибка MAE: 0.59

Модель коректно виконує завдання прогнозування на заданому наборі даних.

Завдання 2.3. Створення багатовимірного регресора

Використовувати файл вхідних даних: data_multivar_regr.txt, побудувати регресійну модель на основі багатьох змінних

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

Лістинг програми:

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

matplotlib.use('TkAgg')
import matplotlib.pyplot as plt

# 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
linear_regressor = linear_model.LinearRegression()
linear_regressor.fit(X_train, y_train)

# Прогнозування результату
y_test_pred = linear_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)

# перетворення точки даних на поліном для подальшого прогнозу результату
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)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

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

Результати:

```
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:
[41.08240428]
```

Рис.5

У ході виконання завдання побудовано та порівняно моделі багатовимірної лінійної та поліноміальної регресій.

Показники якості лінійної моделі на тестовій вибірці:

Коефіцієнт детермінації R^2 : 0.86.

Середня абсолютна похибка MAE: 3.58.

Для перевірки точності було виконано прогнозування для контрольної точки (очікуване значення 41.35). Результати:

Лінійна регресія: 36.05.

Поліноміальна регресія (ступінь 10): 41.08.

Поліноміальний регресор забезпечив результат, значно ближчий до істинного, що свідчить про його вищу ефективність для даного набору даних.

Завдання 2.4. Регресія багатьох змінних

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

Лістинг програми:

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import matplotlib

matplotlib.use('TkAgg')
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

# Розділення на навчальні та тестові дані
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.5, random_state=0)

# Модель лінійної регресії і тренування
regr = linear_model.LinearRegression()
regr.fit(Xtrain, ytrain)

# прогноз по тестовій вибірці
ypred = regr.predict(Xtest)

# вивід коефіцієнтів регресії та показників
print("Коефіцієнти лінійної регресії: ", regr.coef_)
print("Зміщення (intercept) моделі: ", round(regr.intercept_, 2))
print("Показник R2: ", round(r2_score(ytest, ypred), 2))
print("Середня абсолютна помилка: ", round(mean_absolute_error(ytest, ypred), 2))
print("Середньоквадратична помилка: ", round(mean_squared_error(ytest, ypred), 2))

# графіки
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('Предбачено')
plt.show()

```

Результати:

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

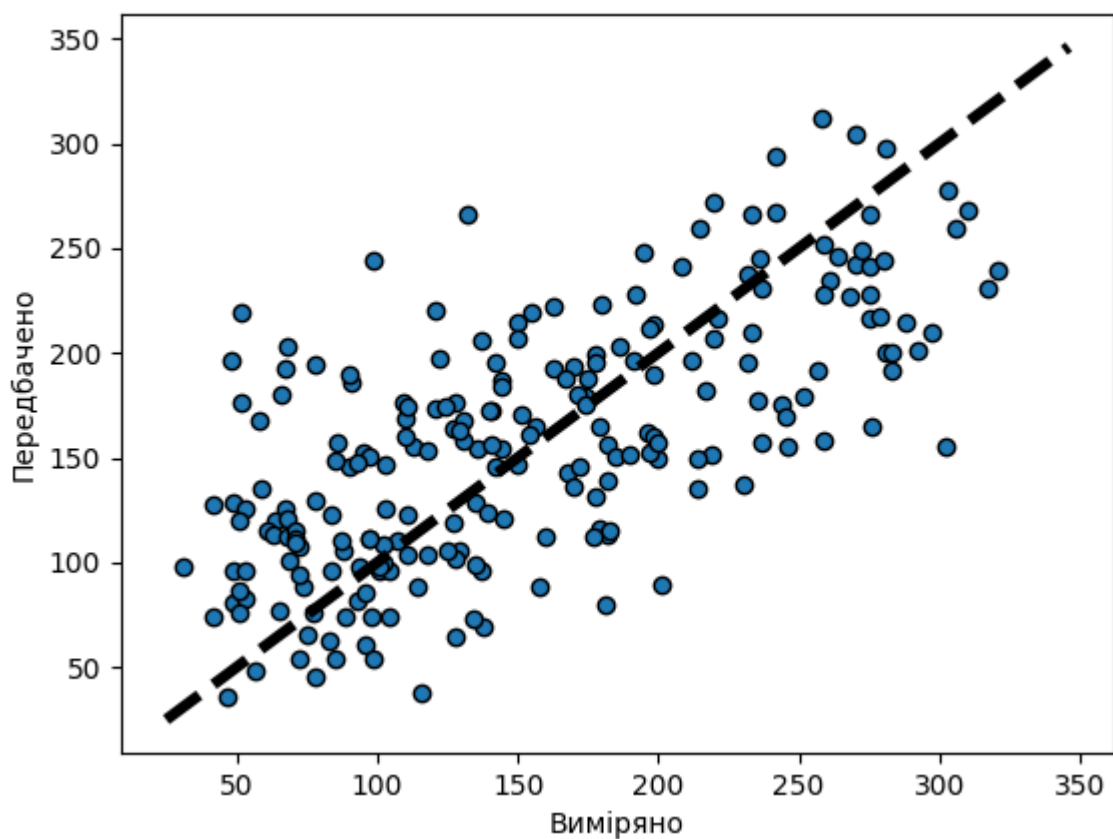


Рис.6 Графік

```
Коефіцієнти лінійної регресії: [ -20.4047621  -265.88518066  564.65086437  325.56226865  -692.16120333
 395.55720874  23.49659361  116.36402337  843.94613929  12.71856131]
Зміщення (intercept) моделі: 154.36
Показник R2: 0.44
Середня абсолютна помилка: 44.8
Середньоквадратична помилка: 3075.33
```

Рис.7 Результати оцінки якості

Завдання 2.5. Самостійна побудова регресії

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.

Варіант 6

```
m = 100
X = np.linspace(-3, 3, m)
y = 2 * np.sin(X) + np.random.uniform(-0.6, 0.6, m)
```

Лістинг програми:

```

import matplotlib
from sklearn.preprocessing import PolynomialFeatures

matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model

# 6 варіант
m = 100
X = np.linspace(-3, 3, m)
y = 2 * np.sin(X) + np.random.uniform(-0.6, 0.6, m)

# Приводимо X до форми двовим. масиву
X = X.reshape(-1, 1)

# Лінійна регресія
lin_model = linear_model.LinearRegression()
lin_model.fit(X, y)

# Поліноміальна
poly = PolynomialFeatures(degree=2, include_bias=False)
x_poly = poly.fit_transform(X)

poly_model = linear_model.LinearRegression()
poly_model.fit(x_poly, y)

# Вивід результатів
print('Linear ---')
print('Coef:', lin_model.coef_)
print('Intercept:', lin_model.intercept_)

print('\nPolynomial ---')
print('Coef:', poly_model.coef_)
print('Intercept:', poly_model.intercept_)

# Предикт для графіку
y_pred_lin = lin_model.predict(X)
y_pred_poly = poly_model.predict(x_poly)

# Малюємо графік
plt.figure(figsize=(9, 5))
plt.scatter(X, y, color='black', s=10, label="Початкові дані")
plt.plot(X, y_pred_lin, color='orange', label="Linear")
plt.plot(X, y_pred_poly, color='blue', label="Poly (deg=2)")

plt.xlabel("X")
plt.ylabel("Y")
plt.title("Порівняння регресій")
plt.legend()
plt.grid(alpha=0.5)
plt.show()

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Результати:

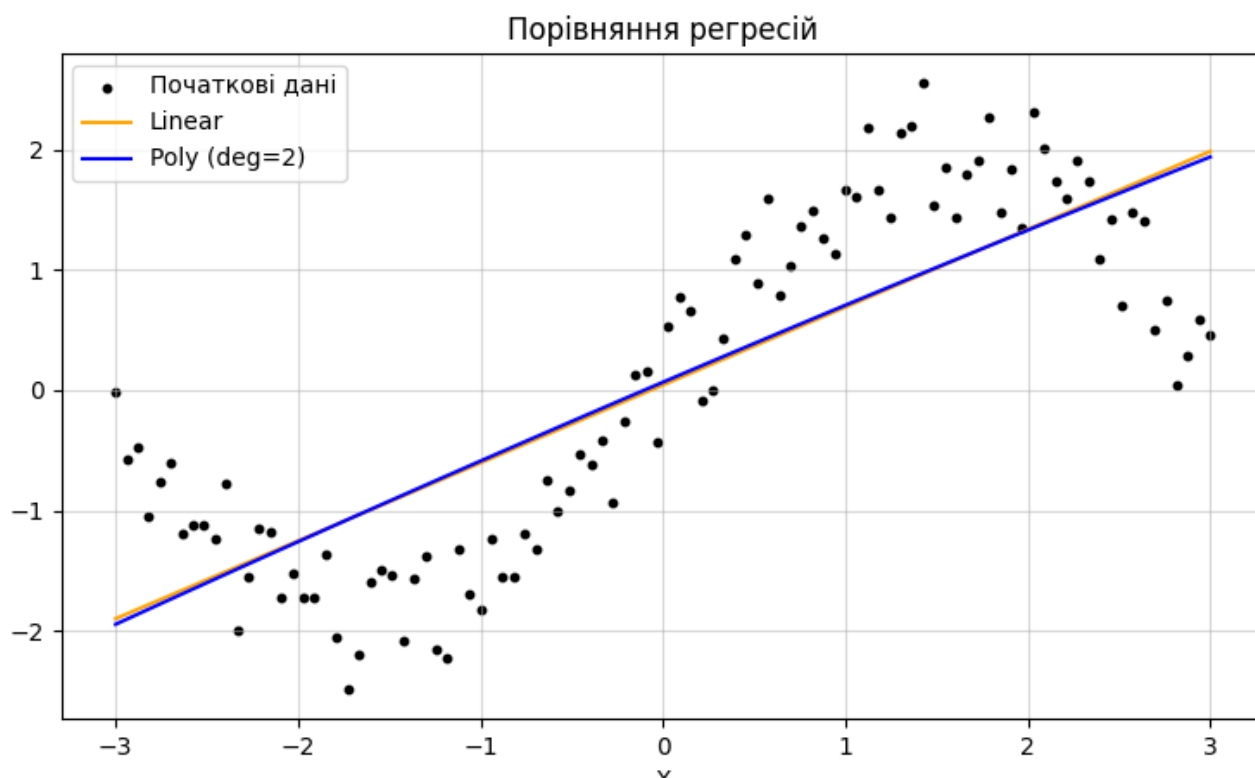


Рис.8 Графік

```
Linear ---
Coef: [0.64749796]
Intercept: 0.04187258114744384

Polynomial ---
Coef: [ 0.64749796 -0.00779215]
Intercept: 0.06572128599273658
```

Рис.9 Порівняння характеристик

За результатами навчання отримано такі рівняння залежності:

Лінійна модель:

$$y = 0.65x + 0.04$$

Поліноміальна модель (2-го ступеня):

$$y = -0.008x^2 + 0.65x + 0.07$$

Аналіз результатів:

У поліноміальній моделі коефіцієнт при x^2 є вкрай малим (-0.008), що вказує на незначний вплив кривизни. Основний внесок робить лінійна складова (0.65x),

яка збігається з коефіцієнтом лінійної регресії. Це підтверджує, що обрані дані мають лінійний характер розподілу.

Завдання 2.6. Побудова кривих навчання Побудуйте криві навчання для ваших даних у попередньому завданні.

Варіант 6

```
m = 100
X = np.linspace(-3, 3, m)
y = 2 * np.sin(X) + np.random.uniform(-0.6, 0.6, m)
```

Лістинг програми:

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

matplotlib.use("TkAgg")
import matplotlib.pyplot as plt
from sklearn import linear_model, preprocessing
from sklearn.pipeline import Pipeline

# будує криві навчання моделі для навчальних даних
def plot_learning_curves(model, X, y):
    # Розділення на навчальні та тестові дані
    X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2)
    train_errors, val_errors = [], []
    for i in range(1, len(X_train)):
        model.fit(X_train[:i], y_train[:i])
        # Прогнози
        y_train_predict = model.predict(X_train[:i])
        y_val_predict = model.predict(X_val)
        # Перехоплення помилок
        train_errors.append(mean_squared_error(y_train_predict, y_train[:i]))
        val_errors.append(mean_squared_error(y_val_predict, y_val))

    # Відображення кривих навчання
    plt.plot(np.sqrt(train_errors), "r-+", linewidth=2, label="Навчальний набір")
    plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label="Тестовий набір")
    plt.xlabel("Training set size")
    plt.ylabel("RMSE")
    plt.legend()
    plt.show()

# 6 варіант
m = 100
X = np.linspace(-3, 3, m)
y = 2 * np.sin(X) + np.random.uniform(-0.6, 0.6, m)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.6.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

```

# Приводимо X до форми двовим. масиву
X = X.reshape(-1, 1)

# Лінійна регресія
linear_regressor = linear_model.LinearRegression()

# Поліноміальні ознаки
polynomial_features = preprocessing.PolynomialFeatures(
    degree=2,
    include_bias=False,
)
X_poly_train = polynomial_features.fit_transform(X)

polynomial_regressor = Pipeline([
    ("poly_features", polynomial_features),
    ("lin_reg", linear_model.LinearRegression()),
])

# Навчання моделей на даних
linear_regressor.fit(X, y)
polynomial_regressor.fit(X_poly_train, y)

# Передбачення для обох моделей
y_linear = linear_regressor.predict(X)
y_polynomial = polynomial_regressor.predict(X_poly_train)

# Відображення кривих навчання для лінійної моделі та поліноміальної моделі
plot_learning_curves(linear_regressor, X, y)
plot_learning_curves(polynomial_regressor, X_poly_train, y)

```

Результати:

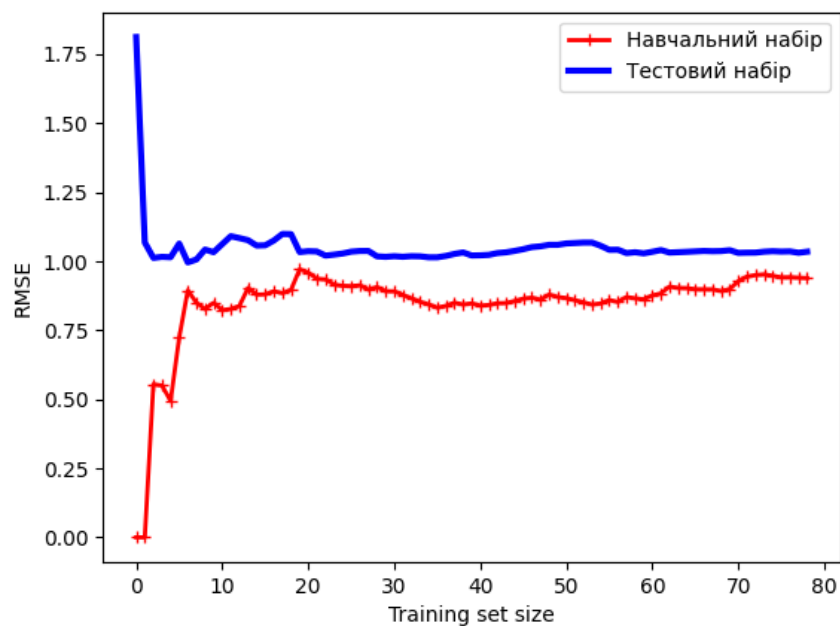


Рис.10 Криві навчання лінійної моделі

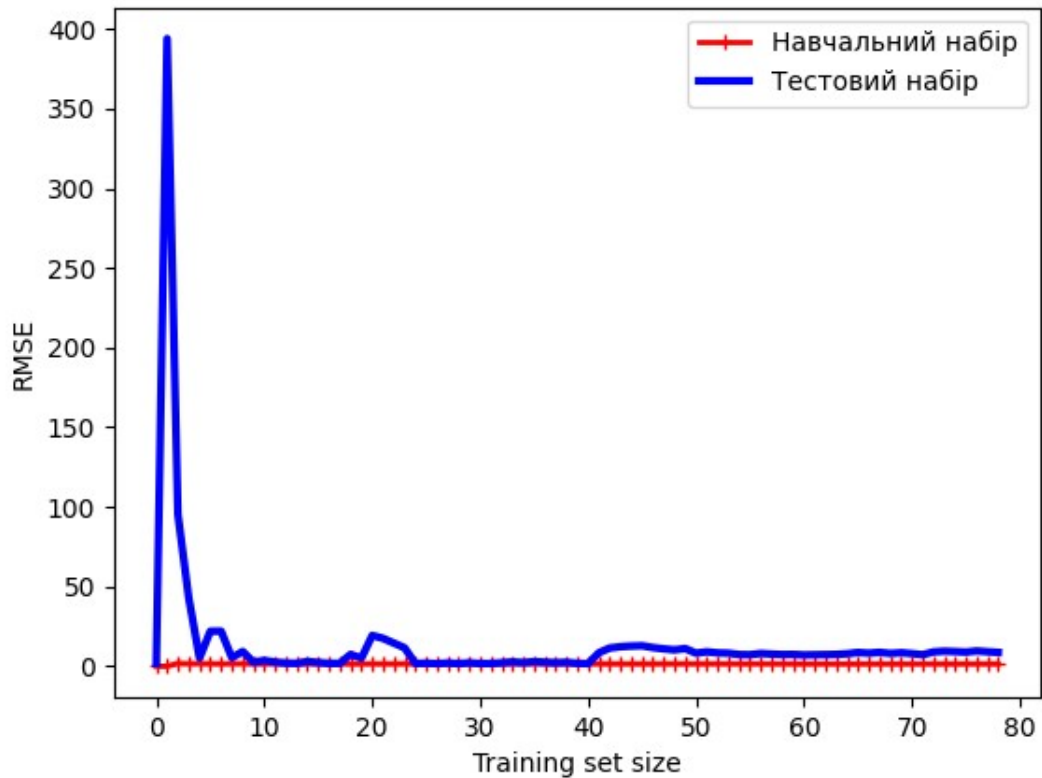


Рис.11 Криві навчання поліноміальної моделі(degree 10)

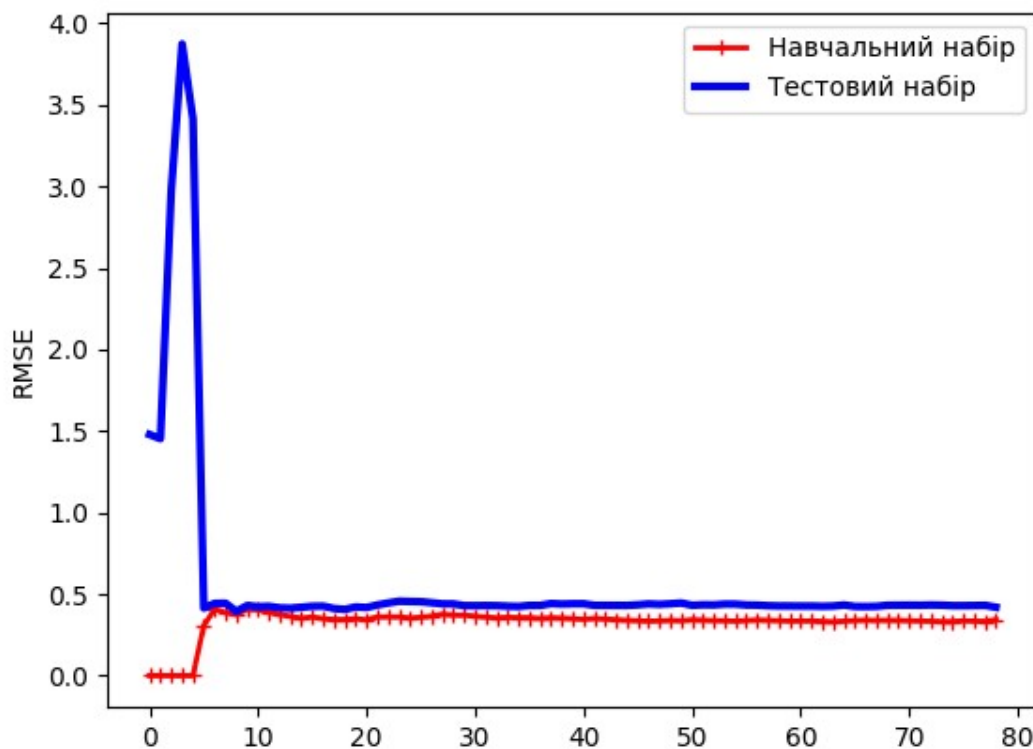


Рис.12 Криві навчання поліноміальної моделі(degree 2)

Висновок: у цій лабораторній роботі я використовуючи спеціалізовані бібліотеки та мову програмування Python дослідив методи регресії даних у машинному навчанні.