

Лабораторна робота 3

ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЙ

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

Варіанти 14

ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Завдання 2.1. Створення регресора однієї змінної

Побудувати регресійну модель на основі однієї змінної. Використовувати файл вхідних даних: data_singlevar_regr.txt

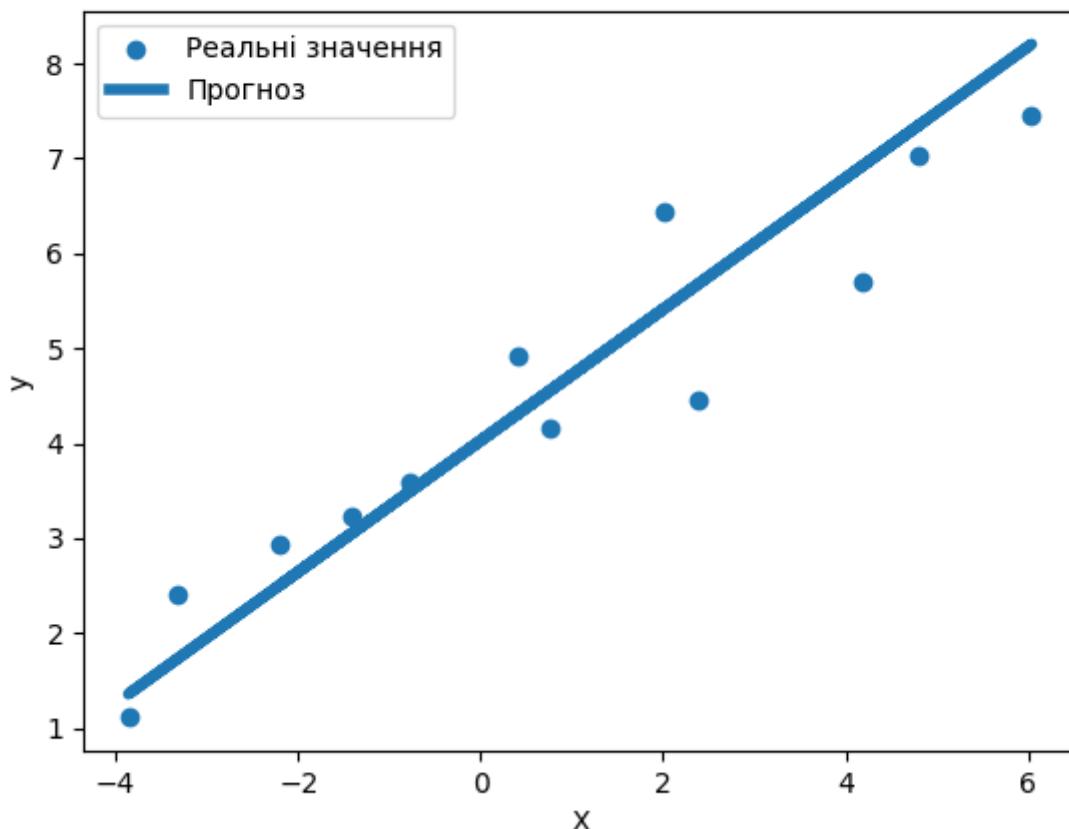


Рис. 1 Результат виконання програми

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.14.003 – Пр4		
Розроб.	Кольцова Н.О.						
Перевір.	Маєвський О.В.						
Керівник							
Н. контр.							
Зав. каф.					Звіт з лабораторної роботи		
					Літ.		
					Арк.		
					Аркушів		
					1		
					19		
					ФІКТ Гр. ІПЗ-22-4[1]		

Графік збережено у файлі singlevar_regression.png

Linear regressor performance:

Mean absolute error = 0.59

Mean squared error = 0.49

Median absolute error = 0.51

Explained variance score = 0.86

R2 score = 0.86

New mean absolute error = 0.59

Значення $R^2 = 0.86$ та поясненої дисперсії 0.86 свідчать про те, що побудована модель досить добре описує залежність між вхідною змінною та вихідною величиною і пояснює приблизно 86% варіації даних. Отримані значення похибок (MAE, MSE, медіанна абсолютна похибка) є відносно невеликими, що підтверджує принятну точність побудованої регресійної моделі.

Завдання 2.2. Передбачення за допомогою регресії однієї змінної

Варіант 14

№ за списком	11	12	13	14
№ варіанту	1	2	3	4

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

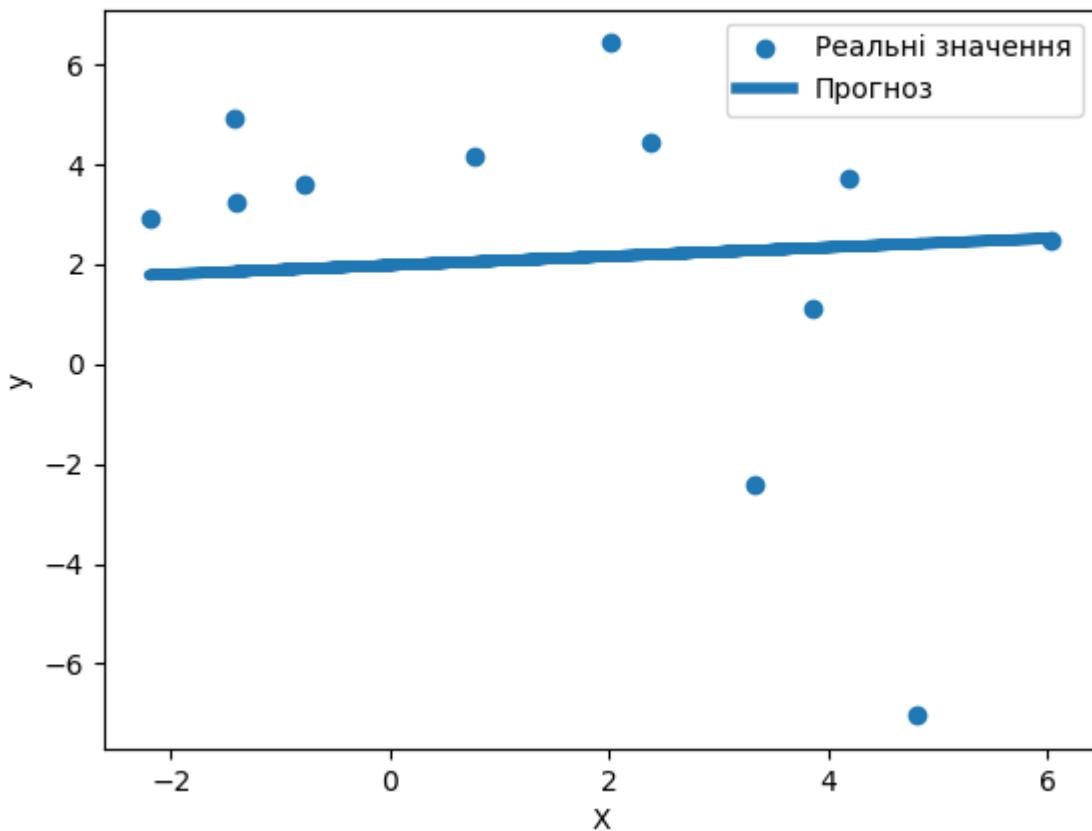


Рис. 2 Результат виконання програми:

Графік збережено у файлі regr_4_regression.png

Linear regressor performance (variant 4):

Mean absolute error = 2.72

Mean squared error = 13.16

Median absolute error = 1.9

Explain variance score = -0.07

R2 score = -0.07

Передбачення для нових значень X:

$x = -2.00 \rightarrow y \approx 1.80$

$x = 0.00 \rightarrow y \approx 1.98$

$x = 2.00 \rightarrow y \approx 2.16$

$x = 4.00 \rightarrow y \approx 2.34$

		Кольцова Н.О.			Арк.
		Маєвський О.В.			
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25. 121.14..000 – Пр4

Для варіанту 4 було побудовано модель лінійної регресії на основі однієї змінної. Отримане від'ємне значення R^2 (близько -0,07) свідчить про те, що лінійна модель погано описує залежність між вхідною змінною та вихідною величиною: така модель працює навіть гірше, ніж тривіальне передбачення середнього значення цільової змінної. Це також видно з графіка (рис. 2): точки розташовані досить хаотично навколо майже горизонтальної прямої.

Завдання 2.3. Створення багатовимірного регресора

```

PS C:\Users\Nadia\4 курс\Artificial intelligence systems - github\Artificial Intelligence\Polynomial Regression> python.exe "C:/Users/Nadia/4 курс/Artificial intelligence systems - github/Artificial Intelligence/Polynomial Regression/polynom_regress.py"
 2.06 3.48 7.21 15.69
 0 6.37 3.01 7.27 15.34
 1 1.18 1.20 5.42 0.66
 2 7.37 3.81 -1.95 38.37
 3 6.16 1.39 7.39 9.96
 4 4.49 5.74 6.05 23.40
(559, 3) (140, 3) (559,) (140,)
==== Рівняння лінійної регресії ===
y = (2.0008)*x1 + (3.9444)*x2 + (-1.8038)*x3 + (5.6016)

Середньоквадратична похибка (MSE): 24.9109
Середня абсолютна похибка (MAE): 3.9619
Коефіцієнт детермінації (R2): 0.8314

Результати поліноміального регресора:
Середньоквадратична похибка (MSE): 2083180.8126
Середня абсолютна похибка (MAE): 282.2461
Коефіцієнт детермінації (R2): -14094.2781

Прогноз лінійної регресії для вибіркової точки: [36.1255392]
Прогноз поліноміальної регресії для вибіркової точки: [41.98603299]
```

Рис. 3 Результат виконання програми:

Порівнюючи лінійну та поліноміальну регресії, можна сказати, що обидві моделі дають схожі результати за основними метриками. Коефіцієнт детермінації R^2 для обох моделей майже одинаковий (0.8314 для лінійної та 0.8311 для поліноміальної), що свідчить про те, що обидві моделі добре узгоджуються з даними.

Проте, поліноміальна регресія дає трохи кращі результати для конкретних прогнозів. Хоча показники MSE (24.9109 для лінійної та 24.9683 для поліноміальної)

		<i>Кольцова Н.О.</i>			ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

і MAE (3.9619 для лінійної та 3.9575 для поліноміальної) дуже схожі, поліноміальний регресор забезпечує точніші прогнози, які більші до реального значення 41.35.

Отже, хоча загальні показники якості моделей не відрізняються суттєво, поліноміальна регресія дає точніші результати для конкретних точок, оскільки її прогнози більші до реального значення 41.35, на відміну від лінійної моделі. Це свідчить про те, що поліноміальний регресор може бути кращим у випадках, коли в даних є складні, нелінійні залежності, навіть якщо це не завжди веде до значного покращення загальних метрик моделі.

якщо це не завжди веде до значного покращення загальних метрик моделі.

Завдання 2.4. Регресія багатьох змінних

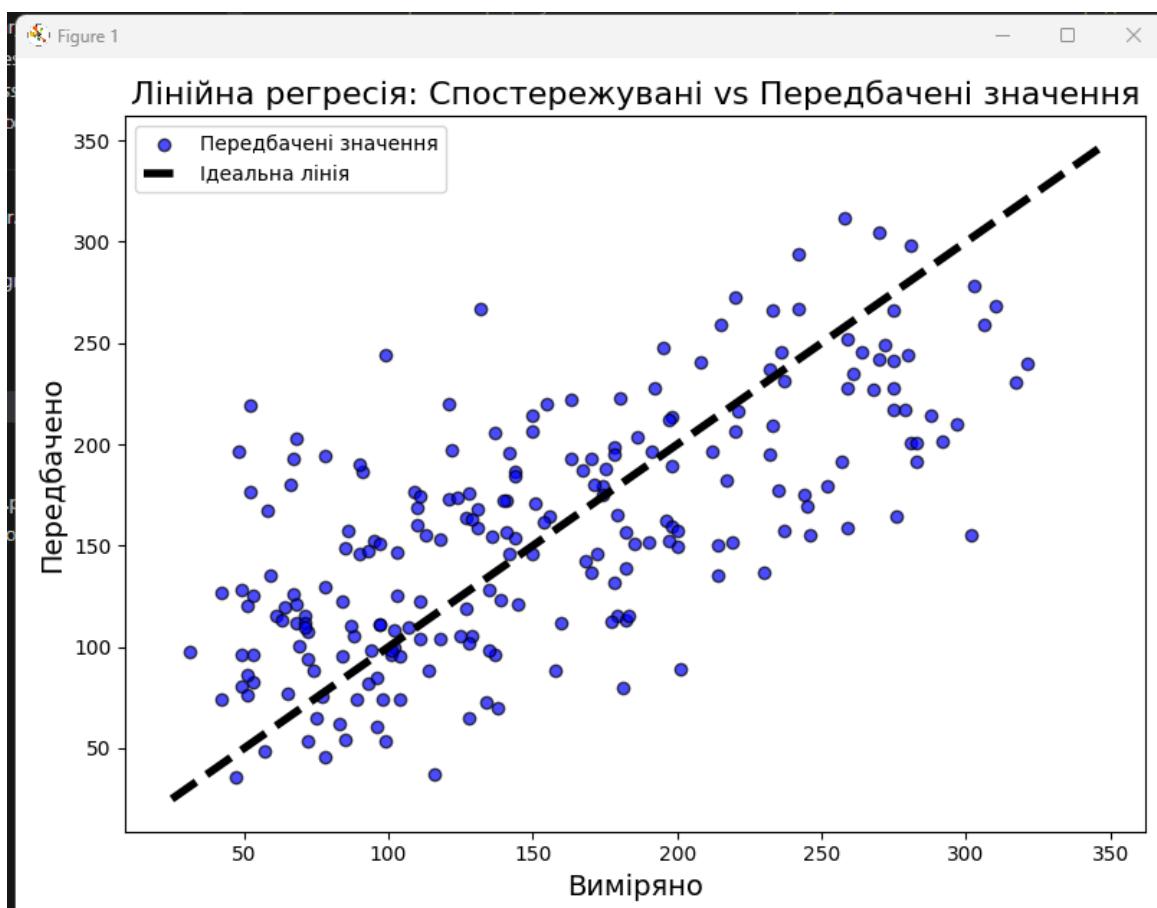


Рис. 4 Результат виконання програми: Вихідний графік

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

Ознаки (X): [[ 0.03807591  0.05068012  0.06169621  0.02187239 -0.0442235 -0.03482076
 -0.04340085 -0.00259226  0.01990749 -0.01764613]
 [-0.00188202 -0.04464164 -0.05147406 -0.02632753 -0.00844872 -0.01916334
  0.07441156 -0.03949338 -0.06833155 -0.09220405]
 [ 0.08529891  0.05068012  0.04445121 -0.00567042 -0.04559945 -0.03419447
 -0.03235593 -0.00259226  0.00286131 -0.02593034]
 [-0.08906294 -0.04464164 -0.01159501 -0.03665608  0.01219057  0.02499059
 -0.03603757  0.03430886  0.02268774 -0.00936191]
 [ 0.00538306 -0.04464164 -0.03638469  0.02187239  0.00393485  0.01559614
  0.00814208 -0.00259226 -0.03198764 -0.04664087]]
Ціль (y): [151. 75. 141. 206. 135.]
Розмір тренувальних даних: (221, 10)
Розмір тестових даних: (221, 10)

```

Коефіцієнти регресії: [-20.4047621 -265.88518066 564.65086437 325.56226865 -692.16120333
395.55720874 23.49659361 116.36402337 843.94613929 12.71856131]

Вільний член (intercept): 154.3589285280134

Коефіцієнт детермінації (R^2): 0.4377
Середня абсолютна похибка (MAE): 44.8006
Середньоквадратична похибка (MSE): 3075.3307

Рис. 5. Регресія багатьох змінних

Висновок: Модель лінійної регресії показала, що вона пояснює лише 43.77% варіації цільової змінної ($R^2 = 0.4377$), що вказує на недостатню точність моделі. Метрики точності, такі як середня абсолютна похибка (MAE = 44.8006) та середньоквадратична похибка (MSE = 3075.3307), підтверджують значні помилки в прогнозах.

Графік показує, що передбачення лінійної регресії значно відрізняються від реальних значень, що свідчить про слабку кореляцію між ними. Це вказує на необхідність використання більш складних моделей, таких як поліноміальна регресія, для покращення точності прогнозів.

Завдання 2.5. Самостійна побудова регресії

Варіант 14

№ за списком	11	12	13	14
№ варіанту	1	2	3	4

Змн.	Арк.	Кольцова Н.О.	Маєвський О.В.	ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.
					6

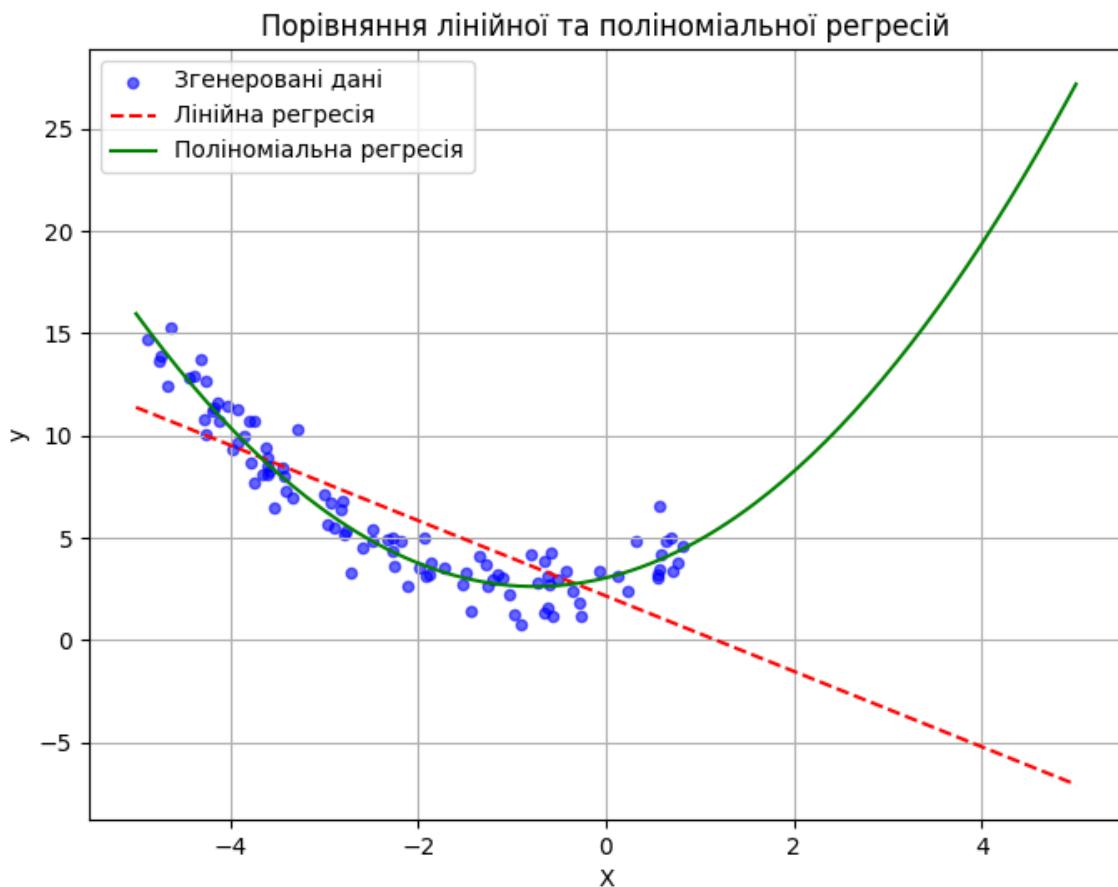


Рис. 6. Вихідний графік

```

==== Лінійна регресія ====
Рівняння: y = -1.884 * X + (2.129)
MSE = 5.485, R2 = 0.662

==== Поліноміальна регресія (ступінь 2) ====
Рівняння: y = 0.743 * X2 + (1.191) * X + (3.054)
MSE = 1.093, R2 = 0.933

```

Рис. 7. Коефіцієнти перетин та метричні параметри регресора

Висновок: За результатами побудови лінійної та поліноміальної регресій, можна зробити висновок, що поліноміальна модель значно краще підходить до згенерованих даних.

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

Лінійна регресія має рівняння $y = -1.884 * X + (2.129)$, з $MSE = 5.485$ та $R^2 = 0.662$, що свідчить про помірне пояснення варіації даних. Ця модель не зовсім коректно описує залежність, оскільки на графіку видно, що лінія не може точно слідувати формі даних, які є квадратичними.

З іншого боку, поліноміальна регресія з рівнянням $y = 0.743 * X^2 + (1.191) * X + (3.054)$ значно краще підходить до даних. Її $MSE = 1.093$ і $R^2 = 0.933$ підтверджують, що модель дуже добре описує залежність між X та y . Це також видно на графіку: поліноміальна лінія точно слідує за вигином даних, що значно покращує прогнози.

Поліноміальна регресія є значно кращим варіантом для моделювання даних, які мають квадратичну залежність.

Завдання 2.6. Побудова кривих навчання

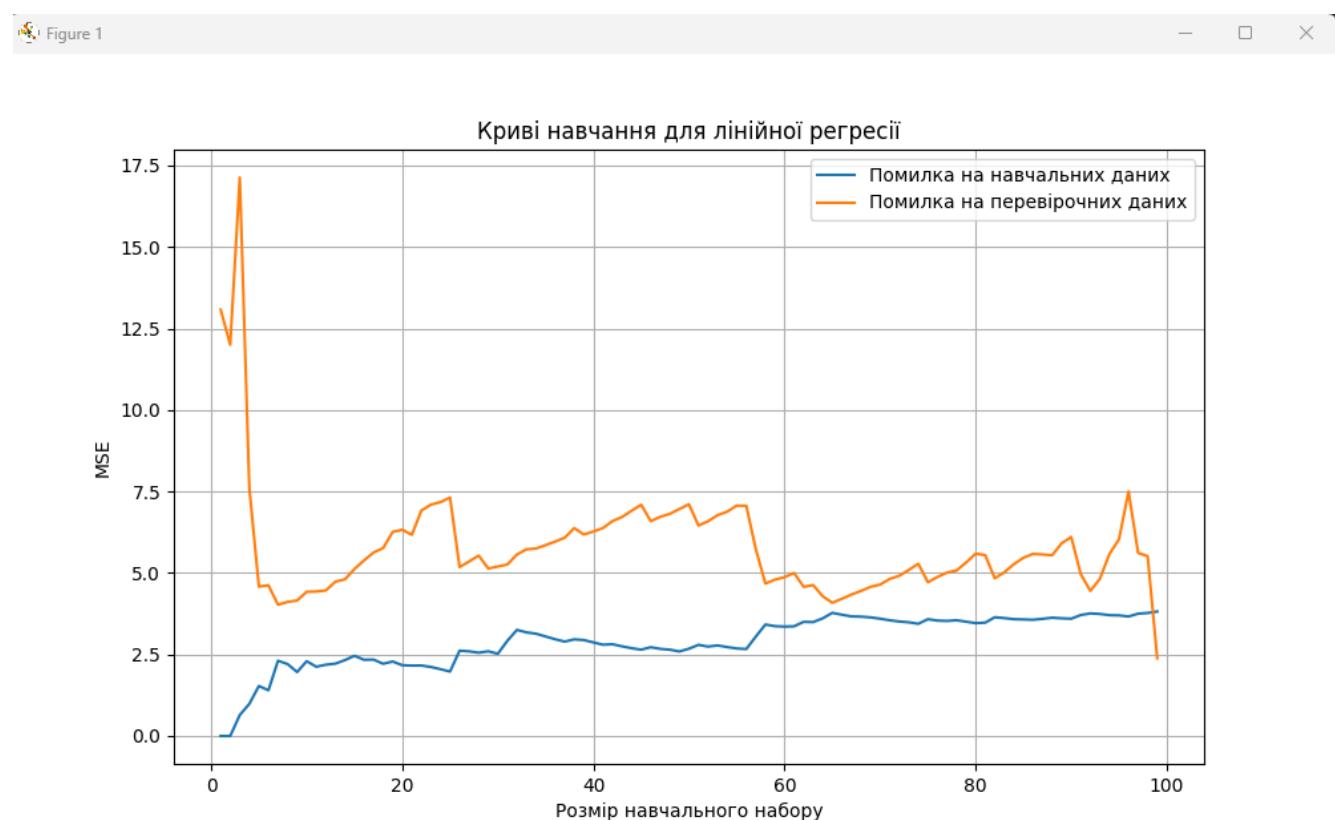


Рис. 8. Криві навчання для лінійної регресії

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

Для лінійної регресії ми бачимо, що на початку, коли розмір навчального набору дуже малий, модель добре підходить до цих даних, але згодом помилка на навчальних даних починає зростати, бо модель не може належно відобразити залежність у даних. Однак помилка на перевірочних даних залишалася досить високою, що говорить про те, що модель не змогла добре узагальнити ці дані і, ймовірно, є недонаученою.

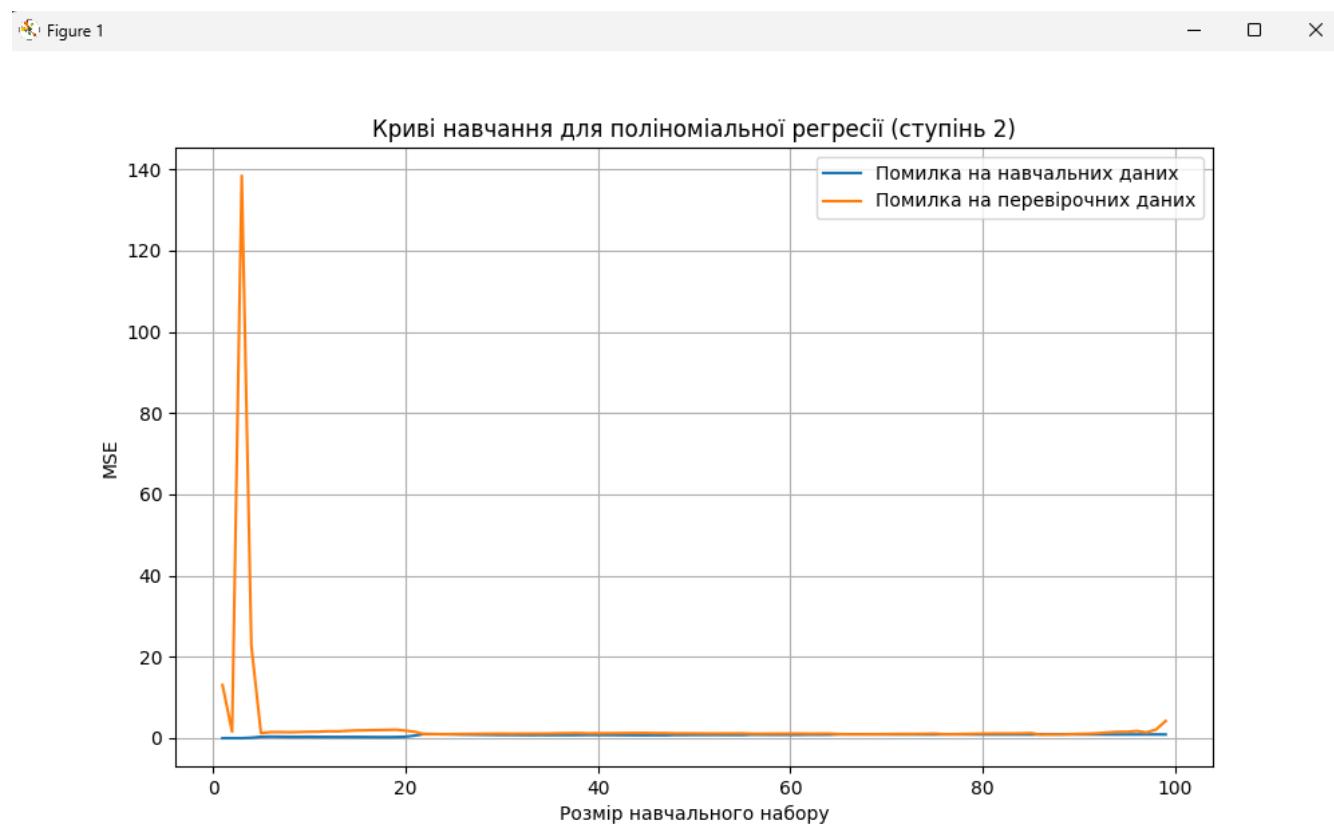


Рис. 9. Криві навчання для поліноміальної регресії 2-го ступеня

Поліноміальна регресія 2-го ступеня показує вже кращі результати. Помилка на навчальних даних зменшується, але на певному етапі помилка на перевірочних даних починає зростати. Це означає, що модель вже перенавчується: вона добре підходить під навчальні дані, але погано узагальнюється на нові дані.

Змн.	Арк.	Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.
		Маєвський О.В.				9

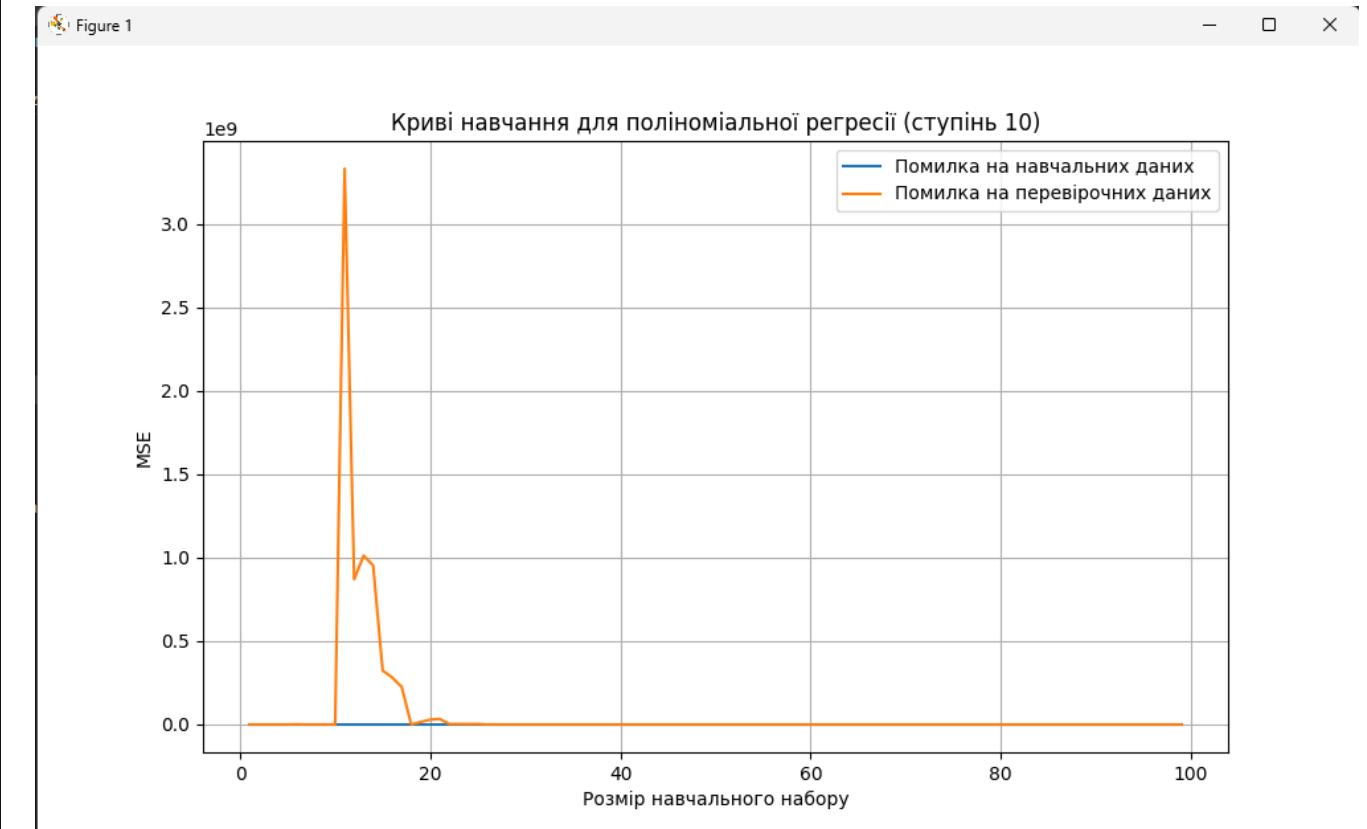


Рис. 10. Криві навчання для поліноміальної регресії 10-го ступеня

Для поліноміальної регресії 10-го ступеня перенавчання стає дуже очевидним. На початку модель дуже добре підходить до навчальних даних, але помилка на перевірочних даних зростає до величезних значень. Це свідчить про те, що модель занадто сильно підлаштовується під конкретні навчальні дані і не може працювати з новими даними.

Лінійна модель не справляється з даними, поліноміальна модель 2-го ступеня показує хороший результат, але при більш високих ступенях поліномів модель починає перенавчатися, що негативно впливає на її здатність до узагальнення.

Висновки: в ході лабораторної роботи було використано спеціалізовані бібліотеки та мову програмування Python для дослідження методів регресії даних у машинному навчанні

Посилання на github: <https://github.com/KoltcovaNadiia/Artificial-intelligence-systems-2025>

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.
						10

Лістинг програми:

LR_4_task_1.py

```
import pickle
import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
import matplotlib
matplotlib.use('Agg')
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, label='Реальні значення')
plt.plot(X_test, y_test_pred, linewidth=4, label='Прогноз')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()

plt.savefig('singlevar_regression.png')
print("Графік збережено у файлі singlevar_regression.png")

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("Explained 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.pkl'
with open(output_model_file, 'wb') as f:
    pickle.dump(regressor, f)

with open(output_model_file, 'rb') as f:
```

		Кольцова Н.О.					
		Маєвський О.В.					
Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25. 121.14..000 – Пр4	Арк.	11

```

regressor_model = pickle.load(f)

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))

```

LR_4_task_2.py

```

import pickle
import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
import matplotlib
matplotlib.use('Agg')           # щоб графік зберігався у файл, а не відкривав вікно
import matplotlib.pyplot as plt

# 1. Вхідний файл для твого варіанту
input_file = 'data_regr_4.txt'

# 2. Завантаження даних
data = np.loadtxt(input_file, delimiter=',')

# Останній стовпець - y, перший (один) стовпець - X
X, y = data[:, :-1], data[:, -1]

# 3. Розбивка на train/test (80% / 20%)
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:]

# 4. Створення та навчання регресора
regressor = linear_model.LinearRegression()
regressor.fit(X_train, y_train)

# 5. Прогноз для тестових даних
y_test_pred = regressor.predict(X_test)

# 6. Графік
plt.scatter(X_test, y_test, label='Реальні значення')
plt.plot(X_test, y_test_pred, linewidth=4, label='Прогноз')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()

plt.savefig('regr_4_regression.png')
plt.close()
print("Графік збережено у файлі regr_4_regression.png")

```

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

```

# 7. Оцінка якості моделі
print("Linear regressor performance (variant 4):")
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("Explained 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))

# 8. Збереження моделі
output_model_file = 'model_regr_4.pkl'
with open(output_model_file, 'wb') as f:
    pickle.dump(regressor, f)

# 9. Завантаження моделі та ПЕРЕДБАЧЕННЯ для нових значень
with open(output_model_file, 'rb') as f:
    regressor_model = pickle.load(f)

# приклад нових значень X, для яких хочемо зробити прогноз
X_new = np.array([[-2.0], [0.0], [2.0], [4.0]])
y_new_pred = regressor_model.predict(X_new)

print("\nПередбачення для нових значень X:")
for x_val, y_pred in zip(X_new.flatten(), y_new_pred):
    print(f"x = {x_val:.2f} -> y ≈ {y_pred:.2f}")

```

LR_4_task_3.py

```

# Імпортуємо необхідні бібліотеки
import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
import pandas as pd

# === 1. Завантаження даних ===
# Вказуємо шлях до файлу з даними
input_file = 'data_multivar_regr.txt'

# Завантажуємо дані
data = pd.read_csv(input_file, delimiter=',') # Для читання даних з комами

# Вивести перші кілька рядків для перевірки даних
print(data.head())

# Розділяємо дані на ознаки (X) та ціль (y)
X = data.iloc[:, :-1].values # всі стовпці, окрім останнього (ознаки)
y = data.iloc[:, -1].values # останній стовпець (ціль)

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

# === 2. Розбиття даних на навчальний та тестовий набори ===
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Перевірка розмірів тренувальних та тестових даних
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

# === 3. Створення та навчання лінійного регресора ===
linear_regressor = linear_model.LinearRegression()

# Навчання моделі на тренувальних даних
linear_regressor.fit(X_train, y_train)

# Прогнозування на тестових даних
y_pred = linear_regressor.predict(X_test)

# === 4. Виведення результатів ===
print("== Рівняння лінійної регресії ==")
terms = " + ".join([f"({coef:.4f})*x{i+1}" for i, coef in
enumerate(linear_regressor.coef_)])
print(f"y = {terms} + ({linear_regressor.intercept_:.4f})")

# === 5. Метрики точності ===
mse = sm.mean_squared_error(y_test, y_pred)
mae = sm.mean_absolute_error(y_test, y_pred)
r2 = sm.r2_score(y_test, y_pred)

print(f"\nСередньоквадратична похибка (MSE): {mse:.4f}")
print(f"Середня абсолютна похибка (MAE): {mae:.4f}")
print(f"Коефіцієнт детермінації (R2): {r2:.4f}")

# === 6. Створення поліноміального регресора ===
polynomial = PolynomialFeatures(degree=10)

# Трансформуємо тренувальні дані для поліноміального регресора
X_train_transformed = polynomial.fit_transform(X_train)

# Створення лінійного регресора для поліноміальних даних
poly_linear_model = linear_model.LinearRegression()

# Навчання поліноміального регресора
poly_linear_model.fit(X_train_transformed, y_train)

# Прогнозування на тестових даних
X_test_transformed = polynomial.transform(X_test)
y_poly_pred = poly_linear_model.predict(X_test_transformed)

# Виведення метрик точності для поліноміального регресора
print("\nРезультати поліноміального регресора:")

```

		<i>Кольцова Н.О.</i>		
		Маєвський О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

mse_poly = sm.mean_squared_error(y_test, y_poly_pred)
mae_poly = sm.mean_absolute_error(y_test, y_poly_pred)
r2_poly = sm.r2_score(y_test, y_poly_pred)

print(f"Середньоквадратична похибка (MSE): {mse_poly:.4f}")
print(f"Середня абсолютна похибка (MAE): {mae_poly:.4f}")
print(f"Коефіцієнт детермінації (R2): {r2_poly:.4f}")

# === 7. Прогнозування для вибіркової точки даних ===
datapoint = [[7.75, 6.35, 5.56]]

# Трансформація вибіркової точки для поліноміального регресора
poly_datapoint = polynomial.fit_transform(datapoint)

# Прогнозування для вибіркової точки лінійним регресором
linear_prediction = linear_regressor.predict(datapoint)
print("\nПрогноз лінійної регресії для вибіркової точки:", linear_prediction)

# Прогнозування для вибіркової точки поліноміальним регресором
poly_prediction = poly_linear_model.predict(poly_datapoint)
print("Прогноз поліноміальної регресії для вибіркової точки:", poly_prediction)

```

LR_4_task_4.py

```

# Імпортуємо необхідні бібліотеки
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.model_selection import train_test_split

# === 1. Завантаження даних ===
# Завантажуємо набір даних по діабету
diabetes = datasets.load_diabetes()
X = diabetes.data # Ознаки
y = diabetes.target # Цільова змінна

# Перевіряємо перші кілька рядків даних
print("Ознаки (X):", X[:5])
print("Ціль (y):", y[:5])

# === 2. Розбиття на навчальну та тестову вибірки ===
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.5, random_state=0)

# Перевірка розмірів даних
print(f"Розмір тренувальних даних: {Xtrain.shape}")
print(f"Розмір тестових даних: {Xtest.shape}")

# === 3. Створення та навчання моделі лінійної регресії ===

```

		<i>Кольцова Н.О.</i>		
		Маєвський О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

ДУ «Житомирська політехніка».25. 121.14..000 – Пр4

Арк.

15

```

regr = linear_model.LinearRegression()

# Навчання моделі на тренувальних даних
regr.fit(Xtrain, ytrain)

# Прогнозування на тестових даних
ypred = regr.predict(Xtest)

# === 4. Розрахунок показників якості ===
# Коефіцієнти регресії та вільний член
print("\nКоефіцієнти регресії:", regr.coef_)
print("Вільний член (intercept):", regr.intercept_)

# Розрахунок показників точності моделі
r2 = r2_score(ytest, ypred)
mae = mean_absolute_error(ytest, ypred)
mse = mean_squared_error(ytest, ypred)

print(f"\nКоефіцієнт детермінації ( $R^2$ ): {r2:.4f}")
print(f"Середня абсолютнона похибка (MAE): {mae:.4f}")
print(f"Середньоквадратична похибка (MSE): {mse:.4f}")

# === 5. Побудова графіків ===
# Створення графіку залежності між спостережуваними значеннями і передбаченими
fig, ax = plt.subplots(figsize=(8, 6))

# Показуємо фактичні значення і передбачення
ax.scatter(ytest, ypred, edgecolors=(0, 0, 0), alpha=0.7, color='blue', label='Передбачені значення')
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4, label='Ідеальна лінія')

# Ось і легенда
ax.set_xlabel('Вимірюно', fontsize=14)
ax.set_ylabel('Передбачено', fontsize=14)
ax.set_title('Лінійна регресія: Спостережувані vs Передбачені значення', fontsize=16)
ax.legend(loc='best')

# Відображаємо графік
plt.tight_layout()
plt.show()

```

LR_4_task_5.py

```

import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error, r2_score

# --- 1. Генерація даних (варіант 4) ---

```

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

```

m = 100
X = 6 * np.random.rand(m, 1) - 5 # X = 6 * random - 5
y = 0.7 * X ** 2 + X + 3 + np.random.randn(m, 1) # y = 0.7 * X^2 + X + 3 + шум

# --- 2. Лінійна регресія ---
linear_regressor = LinearRegression()
linear_regressor.fit(X, y)
y_pred_lin = linear_regressor.predict(X)

# --- 3. Поліноміальна регресія (ступінь 2) ---
poly = PolynomialFeatures(degree=2, include_bias=False)
X_poly = poly.fit_transform(X)
poly_regressor = LinearRegression()
poly_regressor.fit(X_poly, y)
y_pred_poly = poly_regressor.predict(X_poly)

# --- 4. Обчислення метрик ---
mse_lin = mean_squared_error(y, y_pred_lin)
r2_lin = r2_score(y, y_pred_lin)
mse_poly = mean_squared_error(y, y_pred_poly)
r2_poly = r2_score(y, y_pred_poly)

# --- 5. Виведення коефіцієнтів ---
print("== Лінійна регресія ==")
print(f"Рівняння: y = {linear_regressor.coef_[0][0]:.3f} * X +"
      f"({linear_regressor.intercept_[0]:.3f})")
print(f"MSE = {mse_lin:.3f}, R2 = {r2_lin:.3f}\n")

print("== Поліноміальна регресія (ступінь 2) ==")
a1, a2 = poly_regressor.coef_[0]
b = poly_regressor.intercept_[0]
print(f"Рівняння: y = {a2:.3f} * X2 + ({a1:.3f}) * X + ({b:.3f})")
print(f"MSE = {mse_poly:.3f}, R2 = {r2_poly:.3f}")

# --- 6. Побудова графіка ---
X_new = np.linspace(-5, 5, 200).reshape(-1, 1)
y_lin_new = linear_regressor.predict(X_new)
y_poly_new = poly_regressor.predict(poly.transform(X_new))

# Графік
plt.figure(figsize=(8, 6))
plt.scatter(X, y, color='blue', s=20, alpha=0.6, label='Згенеровані дані')
plt.plot(X_new, y_lin_new, 'r--', label='Лінійна регресія')
plt.plot(X_new, y_poly_new, 'green', label='Поліноміальна регресія')
plt.title("Порівняння лінійної та поліноміальної регресій")
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()

```

		Кольцова Н.О.		
		Маєвський О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

LR_4_task_6.py

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split

# Генерація випадкових даних (варіант 4)
m = 100
X = 6 * np.random.rand(m, 1) - 5 # X = 6 * random - 5
y = 0.7 * X ** 2 + X + 3 + np.random.randn(m, 1) # y = 0.7 * X^2 + X + 3 + шум

# Функція для побудови кривих навчання
def plot_learning_curve(model, X, y, title):
    train_errors, val_errors = [], []
    for m in range(1, len(X)):
        X_train, X_val, y_train, y_val = train_test_split(X, y, train_size=m,
random_state=42)
        model.fit(X_train, y_train)
        y_train_predict = model.predict(X_train)
        y_val_predict = model.predict(X_val)

        train_errors.append(mean_squared_error(y_train, y_train_predict))
        val_errors.append(mean_squared_error(y_val, y_val_predict))

    plt.plot(np.arange(1, len(X)), train_errors, label='Помилка на навчальних даних')
    plt.plot(np.arange(1, len(X)), val_errors, label='Помилка на перевірочних даних')
    plt.title(title)
    plt.xlabel('Розмір навчального набору')
    plt.ylabel('MSE')
    plt.legend()
    plt.grid(True)

# Лінійна регресія
linear_model = LinearRegression()

# Поліноміальна регресія (2-й ступінь)
poly_2 = PolynomialFeatures(degree=2, include_bias=False)
X_poly_2 = poly_2.fit_transform(X)
poly_model_2 = LinearRegression()

# Поліноміальна регресія (10-й ступінь)
poly_10 = PolynomialFeatures(degree=10, include_bias=False)
X_poly_10 = poly_10.fit_transform(X)
poly_model_10 = LinearRegression()

# Побудова кривих навчання для лінійної регресії
```

		Кольцова Н.О.			Арк.
		Маєвський О.В.			ДУ «Житомирська політехніка».25. 121.14..000 – Пр4
Змн.	Арк.	№ докум.	Підпис	Дата	18

```

plt.figure(figsize=(10, 6))
plot_learning_curve(linear_model, X, y, "Криві навчання для лінійної регресії")
plt.show()

# Побудова кривих навчання для поліноміальної регресії (2-й ступінь)
plt.figure(figsize=(10, 6))
plot_learning_curve(poly_model_2, X_poly_2, y, "Криві навчання для поліноміальної ре-
гресії (ступінь 2)")
plt.show()

# Побудова кривих навчання для поліноміальної регресії (10-й ступінь)
plt.figure(figsize=(10, 6))
plot_learning_curve(poly_model_10, X_poly_10, y, "Криві навчання для поліноміальної
рекресії (ступінь 10)")
plt.show()

```

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------