

ЛАБОРАТОРНА РОБОТА № 3 ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ

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

Хід роботи:

Завдання 3.1. Створення регресора однієї змінної

```
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()
# Тренування моделі
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()
# Обрахування метрик
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_
```

Рис. 1. Неповний код програми

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 – ЛрЗ						
Зм	Арк.	№ докум.	Підпис	Дата							
Розроб.		Куліш М.В.			Звіт з лабораторної роботи №3			Літ.	Арк.	Акрушів	
Перевір.		Пулеко І.В.								1	18
Реценз.								ФІКТ, гр. КБм-22-1			
Н. Контр.											
Зав.каф.											

```

/home/xtr99/labs/ai/lab03/venv/bin/python /home/xtr99/labs/ai/lab03/LR_3_task_1.py
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

Process finished with exit code 0

```

Рис. 2. Результат виконання програми.

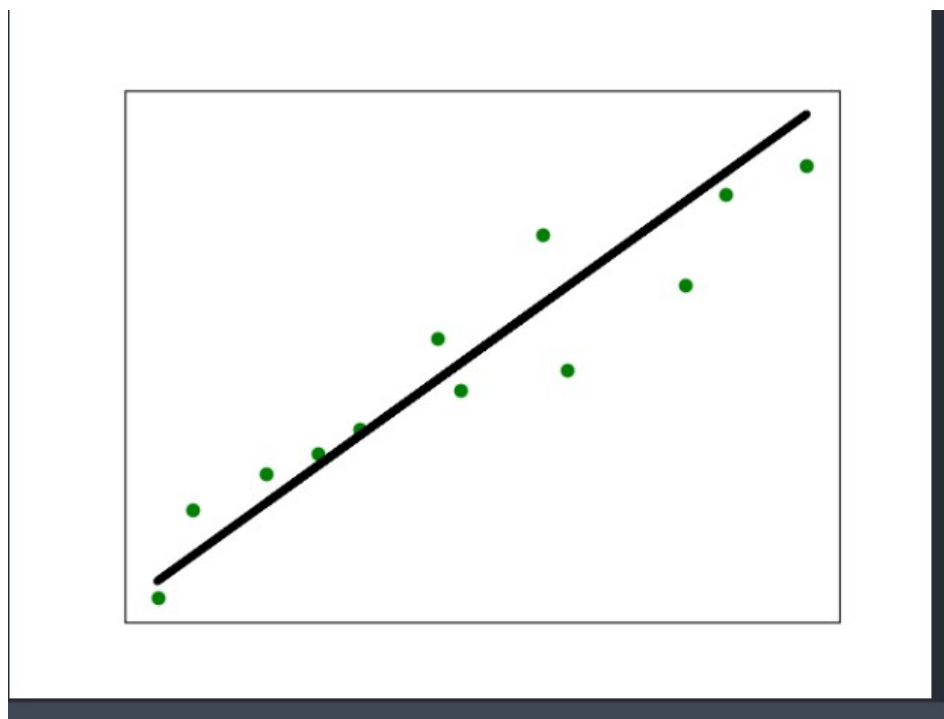


Рис. 3. Відображення графіка.

Показники ефективності лінійного регресора є наступними:

1. Середня абсолютна похибка (MAE) = 0,59: MAE представляє середню абсолютну різницю між прогнозованими та фактичними значеннями. У цьому випадку середня абсолютна різниця становить 0,59, що свідчить про те, що в середньому прогнози не відповідають дійсності на 0,59 одиниць.
2. Середньоквадратична похибка (MSE) = 0,49: MSE вимірює середню квадратичну різницю між прогнозованими та фактичними значеннями. Нижчий показник MSE свідчить про кращу продуктивність, і в цьому випадку середньоквадратична різниця становить 0,49.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Медіанна абсолютна похибка = 0,51: Медіанна абсолютна похибка являє собою медіанне значення абсолютної різниці між прогнозованими та фактичними значеннями. У цьому випадку медіанна абсолютна похибка становить 0,51.
4. Показник поясненої дисперсії = 0,86: Показник поясненої дисперсії є мірою того, наскільки добре модель пояснює дисперсію даних. Він показує частку дисперсії цільової змінної, яку можна пояснити за допомогою моделі. Оцінка 0,86 означає, що модель лінійної регресії може пояснити приблизно 86% дисперсії даних.
5. Оцінка $R^2 = 0,86$: Показник R^2 , також відомий як коефіцієнт детермінації, відображає частку дисперсії цільової змінної, яку можна пояснити за допомогою лінійної регресійної моделі. Показник R^2 0,86 означає, що модель пояснює приблизно 86% дисперсії даних.
6. Нова середня абсолютна похибка дорівнює 0,59, що дорівнює початковій середній абсолютній похибці, згаданій вище.

Завдання 2.2. Передбачення за допомогою регресії однієї змінної.

```
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'
# 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:]
```

Рис. 4. Неповний код програми.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

/home/xtr99/labs/ai/Lab03/venv/bin/python /home/xtr99/labs/ai/Lab03/LR_3_task_2.py
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. 5. Результат виконання програми.

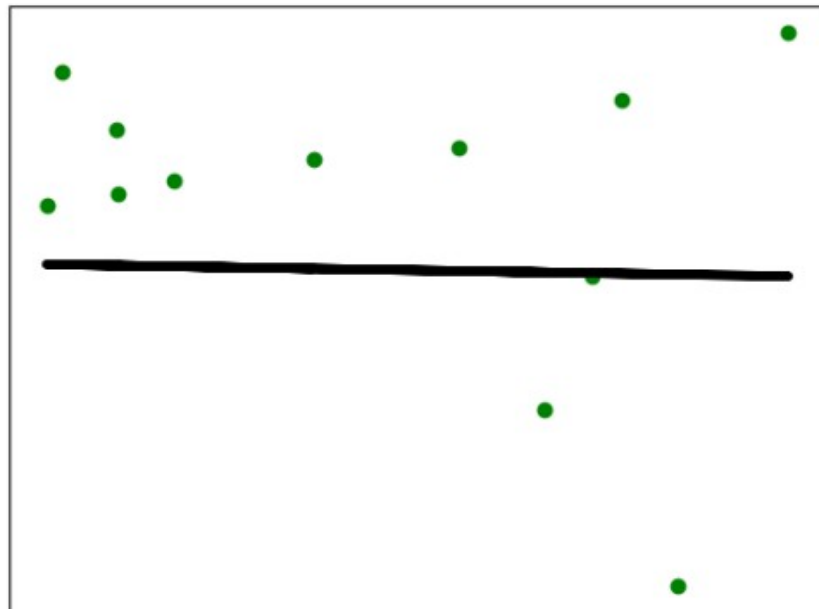


Рис. 6. Відображення графіка.

Середня абсолютна похибка (MAE) 3,59 представляє середню абсолютну різницю між прогнозованими значеннями та істинними значеннями. Це означає, що в середньому прогнози лінійного регресора відхиляються від істинних значень приблизно на 3,59 одиниці.

Середньоквадратична похибка (MSE), що становить 17,39, вимірює середню квадратичну різницю між прогнозованими та істинними значеннями. Вона дає уявлення про середню квадратичну величину помилок.

Медіана абсолютної похибки 3,39 - це медіана абсолютної різниці між прогнозованими та істинними значеннями. Вона вказує на те, що половина абсолютних похибок є нижчими за 3,39, а інша половина - вищими за 3,39.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

Показник поясненої дисперсії 0,02 вказує на те, що лише невелика частка (2%) дисперсії цільової змінної пояснюється лінійною регресійною моделлю. Це свідчить про те, що модель не дуже добре відображає мінливість даних.

Показник R^2 , що дорівнює -0,16, є коефіцієнтом детермінації і відображає частку дисперсії цільової змінної, яка пояснюється моделлю. Від'ємне значення R^2 свідчить про те, що лінійна регресійна модель погано підходить до даних і працює гірше, ніж горизонтальна лінія. Це означає, що прогнози моделі не кращі, ніж просто прогнозування середнього значення цільової змінної.

Нарешті, нова середня абсолютна похибка 3,59 залишається такою ж, як і початкова MAE, що свідчить про те, що не відбулося жодних змін у роботі лінійного регресора.

Завдання 2.3. Створення багатовимірного регресора

```
# 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()

# Тренування моделі
linear_regressor.fit(X_train, y_train)

# Прогнозування результату
y_test_pred = linear_regressor.predict(X_test)

# Обрахування метрик
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))
```

Рис. 7. Неповний код програми.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		


```

/home/xtr99/labs/ai/lab03/venv/bin/python /home/xtr99/labs/ai/lab03/LR_3_task_3.py
Linear Regressor performance:
Mean absolute error = 3.58
Mean squared error = 20.31
Median absolute error = 2.99
Explained variance score = 0.86
R2 score = 0.86

Linear regression:
[36.05286276]

Polynomial regression:
[41.46234773]

Process finished with exit code 0

```

Рис. 8. Результат виконання програми.

Модель лінійної регресії демонструє добру продуктивність з високим коефіцієнтом детермінації (R^2) і високим поясненим коефіцієнтом варіації. Середня абсолютна похибка (MAE) і середня квадратична похибка (MSE) також є прийнятними. Це означає, що модель добре апроксимує дані і має високу точність у прогнозуванні цільової змінної. У порівнянні з поліноміальною регресією, лінійна регресія показує кращі результати з точки зору точності прогнозування.

Завдання 2.4. Регресія багатьох змінних

```

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

# Завантаження даних
diabetes = datasets.load_diabetes()
X = diabetes.data
y = diabetes.target

# Поділ на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)

# Створення та тренування
regr = linear_model.LinearRegression()
regr.fit(X_train, y_train)

# Прогноз
y_pred = regr.predict(X_test)

print("Regression coefficient", regr.coef_)
print("Regression intercept", regr.intercept_)
print("R2 score =", round(r2_score(y_test, y_pred), 2))
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))

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()

```

Рис. 9. Код програми.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

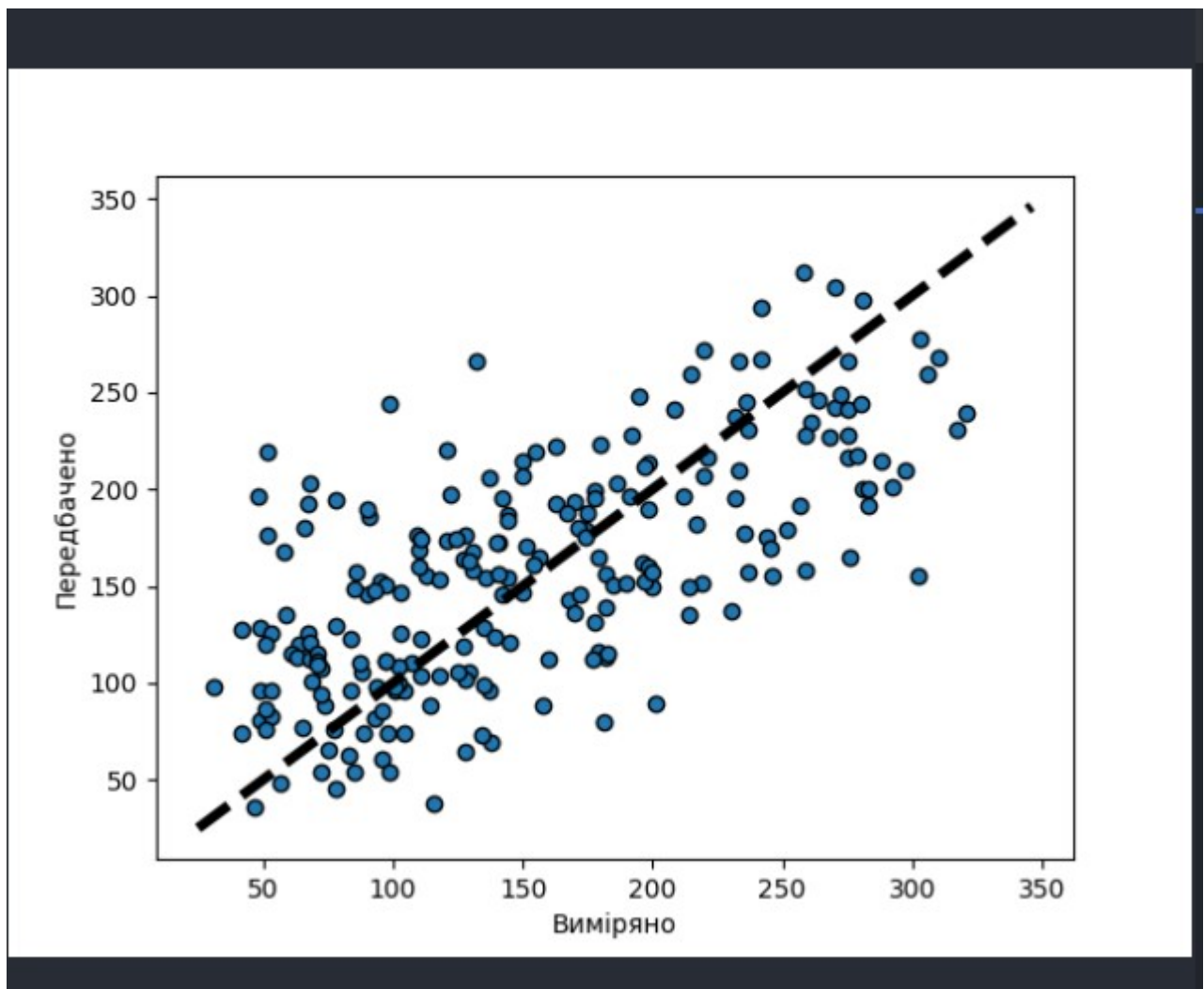


Рис. 10. Відображення графіка.

```
/home/xtr99/labs/ai/lab03/venv/bin/python /home/xtr99/labs/ai/lab03/LR_3_task_4.py
Regression coefficient [ -20.4047621 -265.88518066 564.65086437 325.56226865 -692.16120333
 395.55720874 23.49659361 116.36402337 843.94613929 12.71856131]
Regression intercept 154.35892852801342
R2 score = 0.44
Mean absolute error = 44.8
Mean squared error = 3075.33
```

Рис. 11. Результат виконання програми.

На підставі наданих показників продуктивності моделі регресії можна зробити наступні висновки: коефіцієнт детермінації R^2 становить 0.44, що означає, що лише 44% варіації вихідної змінної можна пояснити за допомогою використаної моделі. Це може свідчити про те, що модель не є достатньо точною у прогнозуванні цільової змінної. Середня абсолютна похибка (MAE) складає 44.8, що означає, що прогнози в середньому відхиляються на 44.8 одиниць від реальних значень. Це може бути великою похибкою, особливо якщо врахувати масштаб даних і контекст задачі. Середня квадратична похибка (MSE) має значення 3075.33, що є високим значенням. Це вказує на велику різницю між прогнозованими значеннями та реальними значеннями, що може бути небажаним для точного прогнозування.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

У загальному, на основі цих показників можна сказати, що модель регресії має обмежену точність і може не надавати задовільних результатів у прогнозуванні цільової змінної. Варто розглянути інші моделі або вдосконалити поточну модель для отримання кращих результатів.

Завдання 2.5. Самостійна побудова регресії.

```
# Генерація даних
m = 100
X = np.linspace(-3, 3, m)
y = 2 * np.sin(X) + np.random.uniform(-0.5, 0.5, m)
X = X.reshape(-1, 1)
y = y.reshape(-1, 1)

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

# Поліноміальна регресія
polynomial = PolynomialFeatures(degree=2, include_bias=False)
X_poly = polynomial.fit_transform(X)
polynomial.fit(X_poly, y)
poly_linear_model = linear_model.LinearRegression()
poly_linear_model.fit(X_poly, y)
y_pred = poly_linear_model.predict(X_poly)
print("\nr2: ", sm.r2_score(y, y_pred))

# Лінійна регресія
plt.scatter(X, y, color='red')
plt.plot(X, linear_regressor.predict(X), color='blue', linewidth=1)
plt.title("Лінійна регресія")
plt.show()

# Поліноміальна регресія
plt.scatter(X, y, color='red')
plt.plot(X, y_pred, "+", color='blue', linewidth=2)
plt.title("Поліноміальна регресія")
plt.show()
```

Рис. 12. Код програми.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

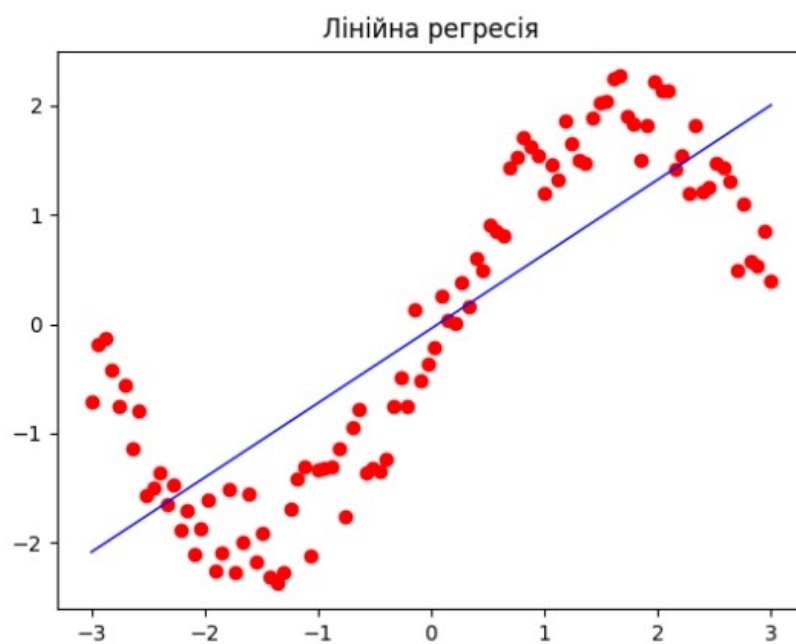


Рис. 13. Графік лінійної регресії.

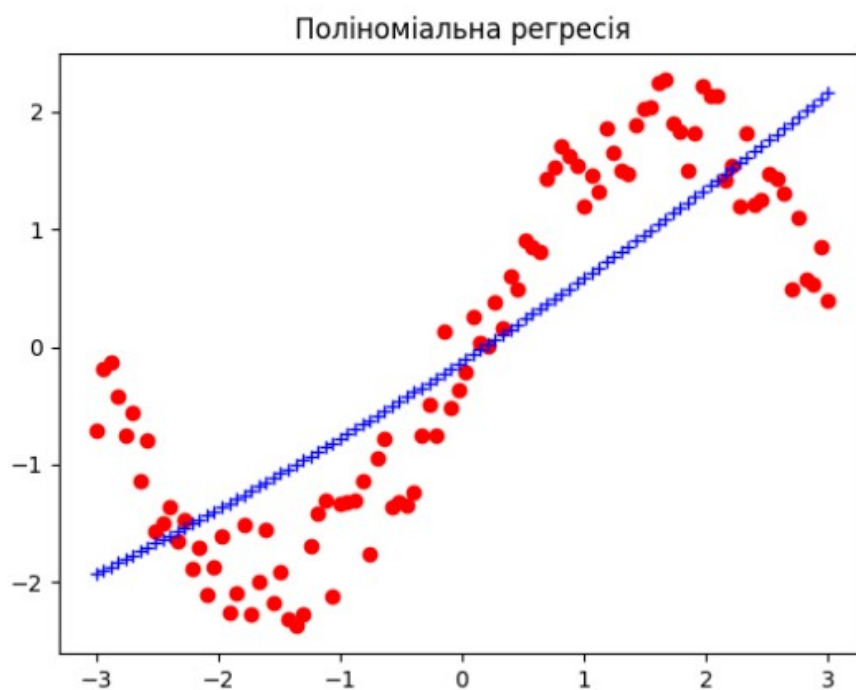


Рис. 14. Графік поліноміальної регресії.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.6. Побудова кривих навчання.

```

3 usages
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))
    plt.plot(np.sqrt(train_errors), "r-+", linewidth=2, label="train")
    plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label="val")

# Генерація даних
m = 100
X = np.linspace(-3, 3, m)
y = 2 * np.sin(X) + np.random.uniform(-0.5, 0.5, m)
X = X.reshape(-1, 1)
y = y.reshape(-1, 1)

# Створення об'єкта лінійного регресора
linear_regression = LinearRegression()
linear_regression.fit(X, y)

# Побудова графіка лінійної регресії
plot_learning_curves(linear_regression, X, y)
plt.xticks()
plt.yticks()
plt.show()

polynomial_regression = Pipeline(
    [("poly_features", PolynomialFeatures(degree=10, include_bias=False)), ("linear_regression", LinearRegression())]
)
plot_learning_curves(polynomial_regression, X, y)
plt.show()

polynomial_regression = Pipeline(
    [("poly_features", PolynomialFeatures(degree=2, include_bias=False)), ("linear_regression", LinearRegression())]
)

```

Рис. 15. Неповний код програми.

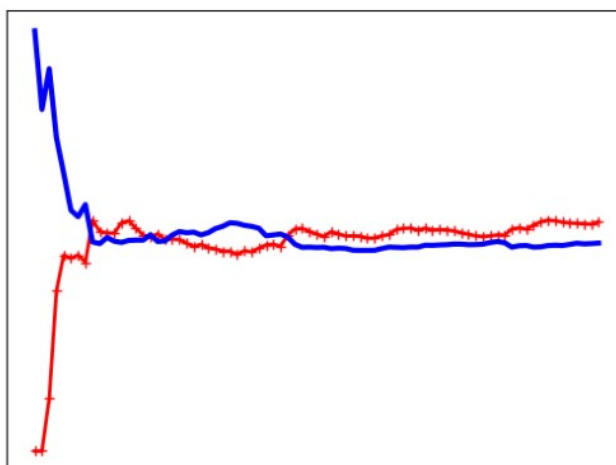


Рис. 16. Криві навчання для лінійної моделі.

f

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

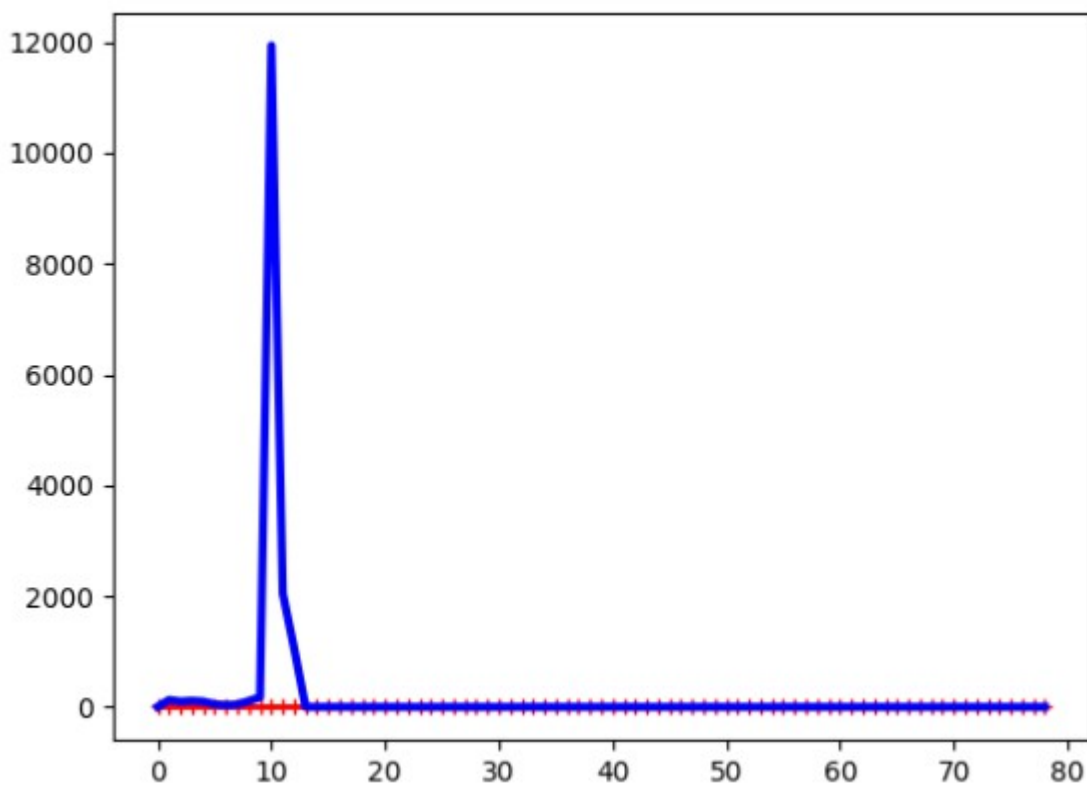


Рис. 17. Криві навчання для поліноміальної моделі 10 ступеня.

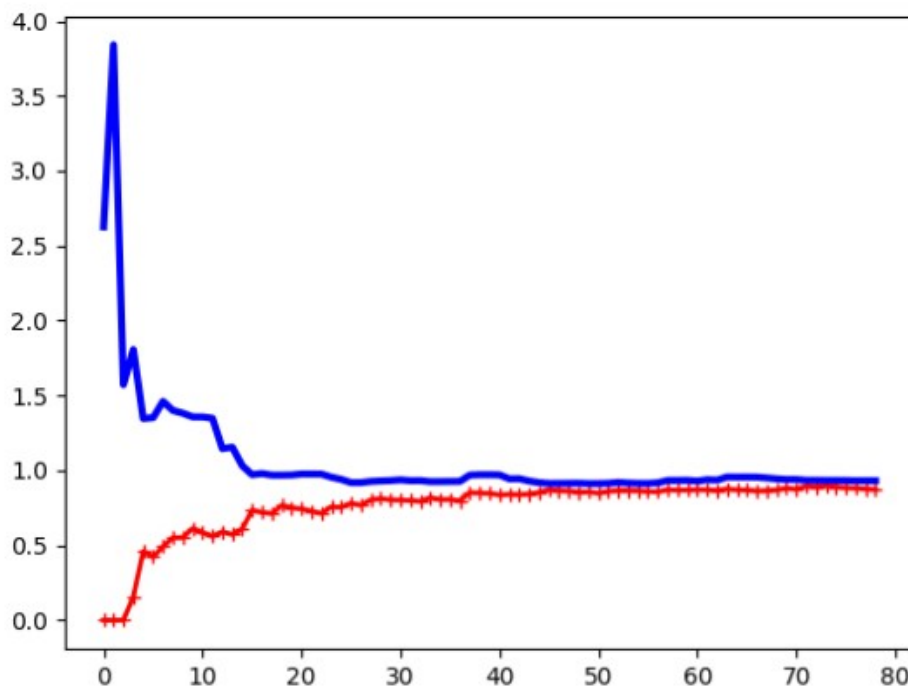


Рис. 18. Криві навчання для поліноміальної моделі 2-го ступеня.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.7. Кластеризація даних за допомогою методу k-середніх

```
# Включення вхідних даних до графіка
plt.figure()
plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none', edgecolors='black', s=80)
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
plt.title('Input data')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())

# Створення об'єкту KMeans
kmeans = KMeans(init='k-means++', n_clusters=num_clusters, n_init=10)

# Навчання моделі кластеризації KMeans
kmeans.fit(X)

# Визначення кроку сітки
step_size = 0.01

# Відображення точок сітки
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
x_vals, y_vals = np.meshgrid(np.arange(x_min, x_max, step_size),
                              np.arange(y_min, y_max, step_size))

# Передбачення вихідних міток для всіх точок сітки
output = kmeans.predict(np.c_[x_vals.ravel(), y_vals.ravel()])
```

Рис. 19. Неповний код програми.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

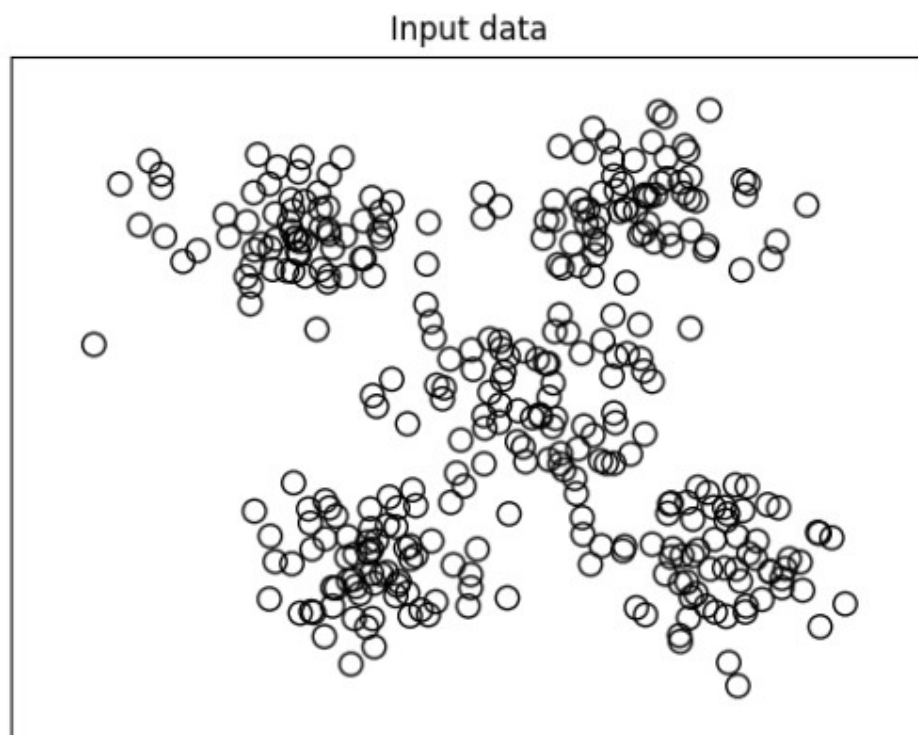


Рис. 20. Графік вхідних даних

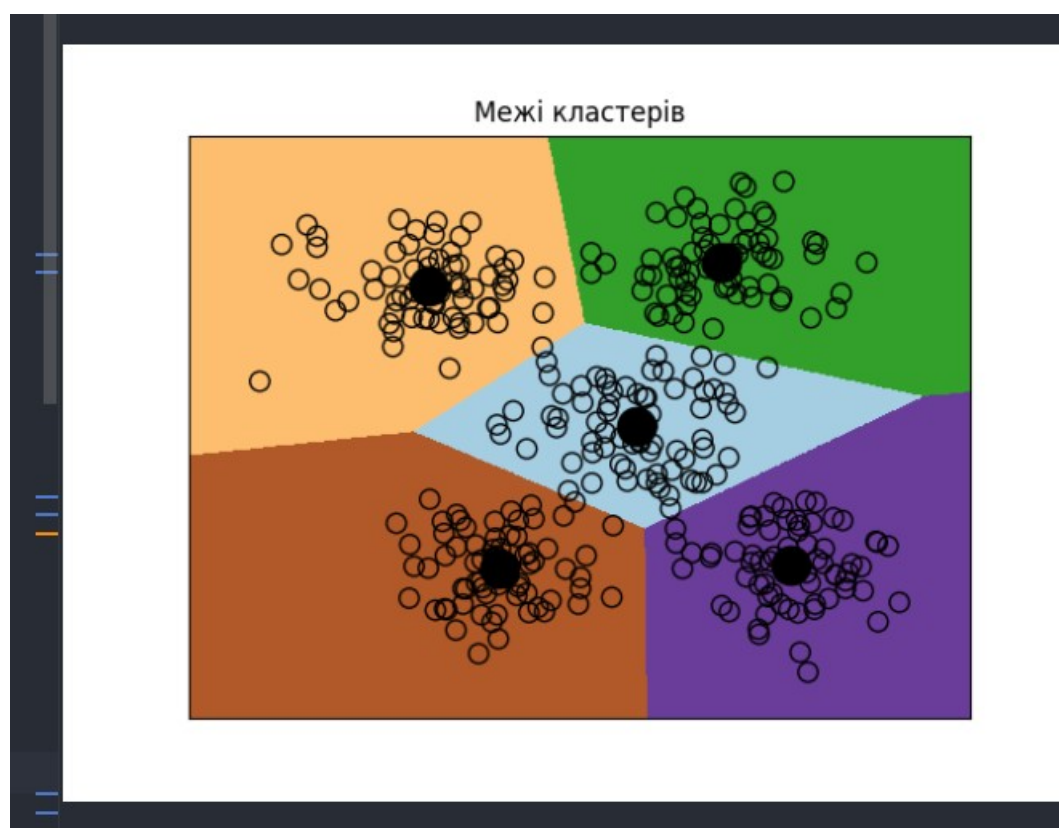


Рис. 21. Графік меж кластерів.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Метод к-середніх є ефективним алгоритмом кластеризації, який добре працює, коли кількість кластерів відома. Він дозволяє розділити набір даних на кластери шляхом групування схожих об'єктів разом. Однак, метод к-середніх має деякі обмеження. Одним з найбільших обмежень є необхідність заздалегідь знати кількість кластерів, на які розподілити дані. Це може бути проблемою в ситуаціях, коли кількість кластерів невідома або може змінюватися. Крім того, метод к-середніх може бути чутливим до початкового вибору центроїдів кластерів і може давати різні результати при кожному запуску.

Отже, хоча метод к-середніх є ефективним і широко використовується, він має свої обмеження, особливо коли кількість кластерів невідома. В таких випадках можуть бути більш прогресивні алгоритми кластеризації, які можуть автоматично визначати кількість кластерів або працювати з змінною кількістю кластерів.

Завдання 2.8. Кластеризація К-середніх для набору даних Iris

```
# Отримуємо дані
iris = datasets.load_iris()
X = iris.data[:, :2]
Y = iris.target

# Визначаємо початкові кластери
kmeans = KMeans(n_clusters=Y.max() + 1, init='k-means++', n_init=10, max_iter=300, tol=0.0001, verbose=0, random_state=None)
kmeans.fit(X)
y_pred = kmeans.predict(X)
print("n_clusters: 3, n_init: 10, max_iter: 300, tol: 0.0001, verbose: 0, random_state: None, copy_x: True")
print(y_pred)
plt.figure()
plt.scatter(X[:, 0], X[:, 1], c=y_pred, s=50, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)
plt.show()

@sages
def find_clusters(X, n_clusters, rseed=2):
    # Випадково обираємо кластери
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]
```

Рис. 22. Неповний код програми.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

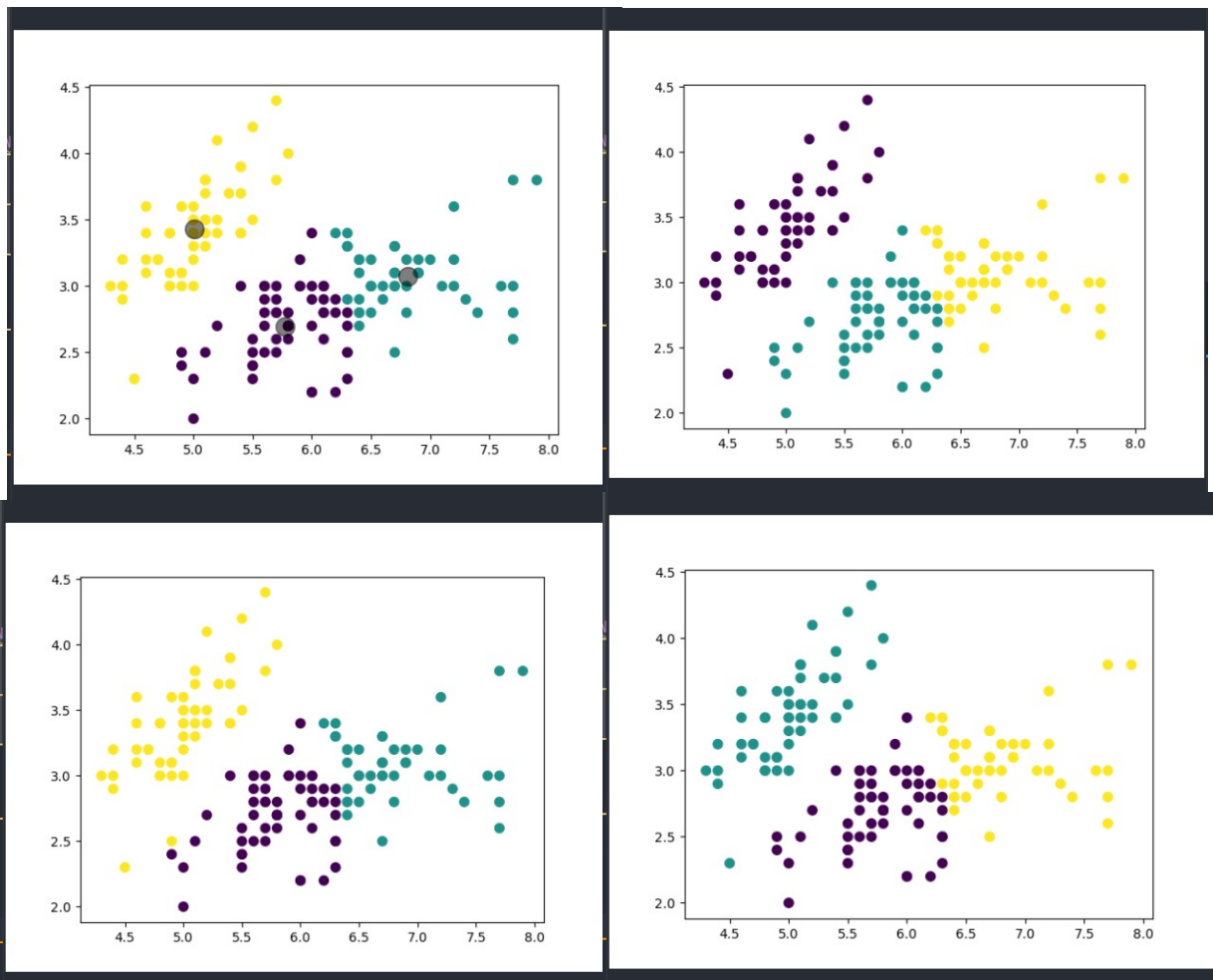


Рис. 23. Кластеризація для набору даних Iris.

Завдання 2.9. Оцінка кількості кластерів з використанням методу зсув середнього.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

```

1  import ...
5
6  # Завантаження даних
7  X = np.loadtxt('data_clustering.txt', delimiter=',')
8  # Оцінка ширини вікна для X
9  bandwidth_X = estimate_bandwidth(X, quantile=0.1, n_samples=len(X))
10 # Класифікація даних методом зсуву середнього
11 meanshift_model = MeanShift(bandwidth=bandwidth_X, bin_seeding=True)
12 meanshift_model.fit(X)
13 # Витягування центрів кластерів
14 cluster_centers = meanshift_model.cluster_centers_
15 print('\nCenters of clusters:\n', cluster_centers)
16 # Оцінка кількості кластерів
17 labels = meanshift_model.labels_
18 num_clusters = len(np.unique(labels))
19 print("\nNumber of clusters in input data =", num_clusters)
20 # Відображення на графіку точок та центрів кластерів
21 plt.figure()
22 markers = 'o*xvs'
23 for i, marker in zip(range(num_clusters), markers):
24     # Відображення на графіку точок, що належать поточному кластеру
25     plt.scatter(X[labels == i, 0], X[labels == i, 1], marker=marker,
26               color=np.random.rand(3,))
27
28     # Відображення на графіку центру кластера
29     cluster_center = cluster_centers[i]
30     plt.plot(cluster_center[0], cluster_center[1], marker='o',
31             markerfacecolor='black', markeredgecolor='red',
32             markersize=15)
33
34 plt.title('Кластери')
35 plt.show()

```

Рис. 24. Код програми.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

```
/home/xtr99/labs/ai/lab03/venv/bin/python /home/xtr99/labs/ai/lab03/LR_3_task_9.py
```

```
Centers of clusters:
```

```
[[2.95568966 1.95775862]
```

```
[7.20690909 2.20836364]
```

```
[2.17603774 8.03283019]
```

```
[5.97960784 8.39078431]
```

```
[4.99466667 4.65844444]]
```

```
Number of clusters in input data = 5
```

Рис. 25. Результат центрів кластерів.

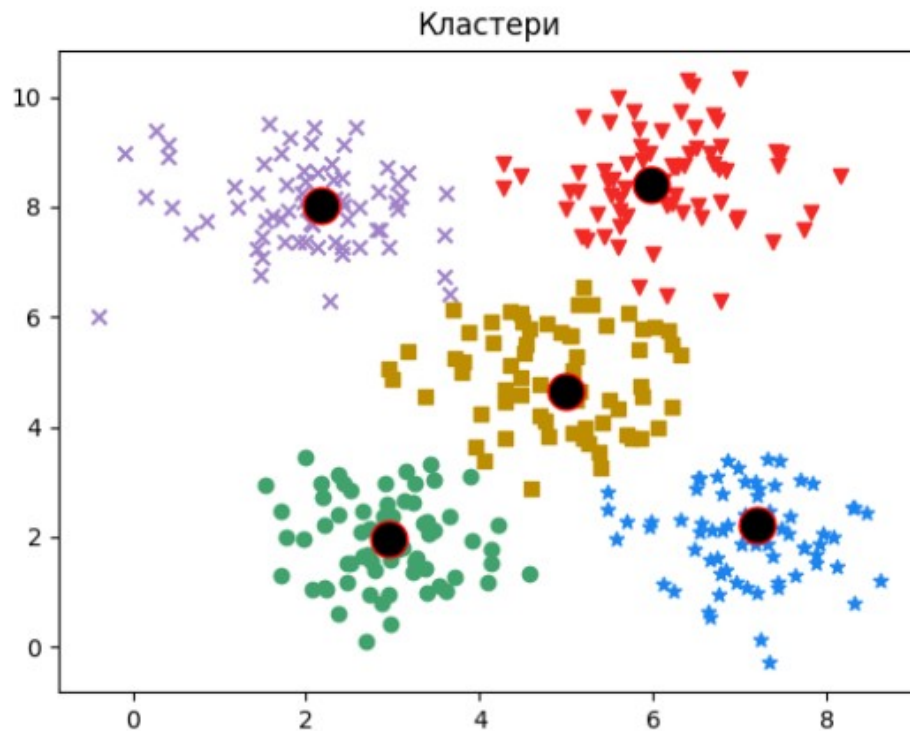


Рис. 26. Графік кластерів.

Застосування методу зсуву середнього дозволяє зменшити шум та випадкові варіації у даних, зберігаючи при цьому загальну тенденцію. Цей метод може бути корисним для виявлення трендів, циклів або сезонності в часових рядах. Він також може використовуватися для вирішення задачі прогнозування шляхом використання отриманих значень зсуву середнього. Отже, метод зсуву середнього є корисним інструментом для аналізу часових рядів, згладжування даних та виявлення загальних тенденцій. Його застосування допомагає покращити якість даних і спростити подальший аналіз часових рядів.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

Покликання на github:

<https://github.com/mrkulish/ai-labs/tree/bd19a56a03d6e4fd2b2ea83d218a88e530b02f83/lab03>

Висновок: під час виконання завдань лабораторної роботи навчилися працювати з використанням спеціалізованих бібліотек та мови програмування Python було досліджено методи регресії та неконтрольованої класифікації даних у машинному навчанні.

		Куліш М.В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.125.08.000 - ЛрЗ	Арк.
		Пулеко І.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		