

## Практична робота № 3

### Варіант 13

#### Дослідження методів регресії та неконтрольованого навчання

**Мета:** 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. Побудувати регресійну модель на основі однієї змінної. Використовувати файл вхідних даних: data\_singlevar\_regr.txt

#### Лістинг файлу task-1.py

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

# Завантаження даних
input_file = 'data_singlevar_regr.txt'
data = np.loadtxt(input_file, delimiter=',')
X, Y = data[:, :-1], data[:, -1]

# Розділення даних на навчальні та тестові
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:]
```

					ДУ «Житомирська політехніка».22.121.13.000 – Лр03						
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи				Літ.	Арк.	Аркушів
Розроб.		Маковська О.Ю									
Перевір.		Пулеко І. В.								1	10
Керівник									ФІКТ Гр. ІПЗ-19-1[2]		
Н. контр.											
Зав. каф.											

```

# Створення лінійної регресії
linear_regressor = linear_model.LinearRegression()
linear_regressor.fit(X_train, Y_train)

# Прогнозування результатів
Y_test_pred = linear_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()

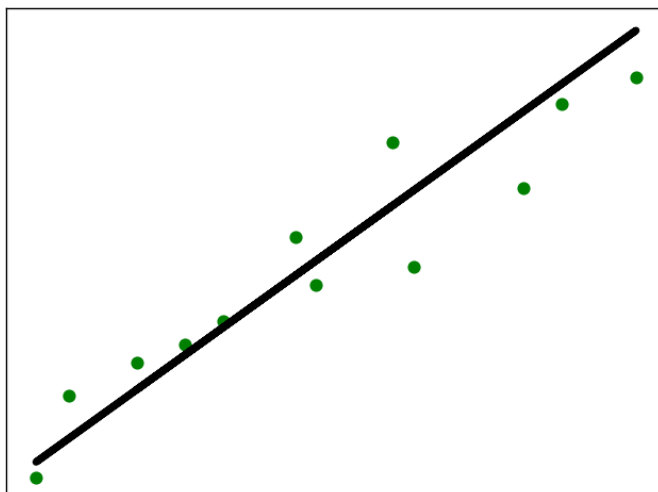
# Виведення результатів
print("Linear regressor performance:")
print(f"Mean absolute error = {round(sm.mean_absolute_error(Y_test, Y_test_pred), 2)}")
print(f"Mean squared error = {round(sm.mean_squared_error(Y_test, Y_test_pred), 2)}")
print(f"Median absolute error = {round(sm.median_absolute_error(Y_test, Y_test_pred), 2)}")
print(f"Explain variance score = {round(sm.explained_variance_score(Y_test, Y_test_pred), 2)}")
print(f"R2 score = {round(sm.r2_score(Y_test, Y_test_pred), 2)}")

# Збереження моделі
output_model_file = 'model.pkl'
with open(output_model_file, 'wb') as f:
    pickle.dump(linear_regressor, f)

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

Y_test_pred_new = model_linregr.predict(X_test)
print(f"\nNew mean absolute error = {round(sm.mean_absolute_error(Y_test, Y_test_pred_new), 2)}")

```



```

C:\Python311\python.exe C:/Users/Ola/
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.3.1. task-1.py

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулюко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

**Завдання 3.2:** Передбачення за допомогою регресії однієї змінної. Побудувати регресійну модель на основі однієї змінної. Використовувати вхідні дані відповідно свого варіанту.

**Варіант 3 файл: data\_regr\_3.txt**

№ за списком	13
№ варіанту	3

Лістинг файлу task-2.py

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

# Завантаження даних
input_file = 'data_regr_3.txt'
data = np.loadtxt(input_file, delimiter=',')
X, Y = data[:, :-1], data[:, -1]

# Розділення даних на навчальні та тестові
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:]

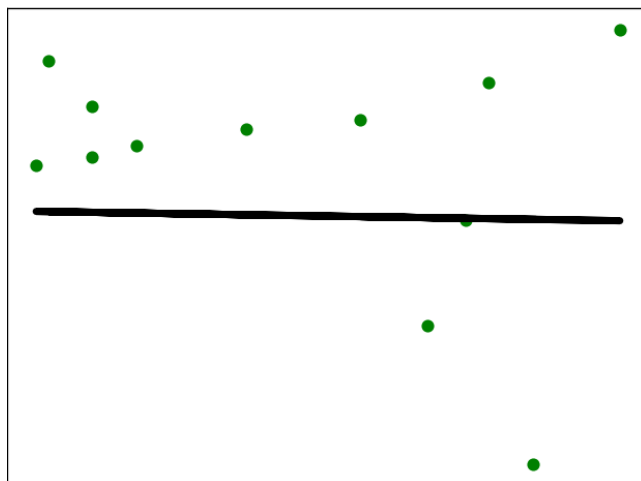
# Створення лінійної регресії
linear_regressor = linear_model.LinearRegression()
linear_regressor.fit(X_train, Y_train)

# Прогнозування результатів
Y_test_pred = linear_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()

# Виведення результатів
print("Linear regressor performance:")
print(f"Mean absolute error = {round(sm.mean_absolute_error(Y_test, Y_test_pred), 2)}")
print(f"Mean squared error = {round(sm.mean_squared_error(Y_test, Y_test_pred), 2)}")
print(f"Median absolute error = {round(sm.median_absolute_error(Y_test, Y_test_pred), 2)}")
print(f"Explain variance score = {round(sm.explained_variance_score(Y_test, Y_test_pred), 2)}")
print(f"R2 score = {round(sm.r2_score(Y_test, Y_test_pred), 2)}")
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		



```
C:\Python311\python.exe C:/Users/Ola,
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
```

Рис.3.2. task-2.py

**Завдання 3.3:** Створення багатовимірного регресора. Використовувати файл вхідних даних: data\_multivar\_regr.txt, побудувати регресійну модель на основі багатьох змінних.

Лістинг файлу task-3.py

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

# Завантаження даних
input_file = 'data_multivar_regr.txt'
data = np.loadtxt(input_file, delimiter=',')
X, Y = data[:, :-1], data[:, -1]

# Розділення даних на навчальні та тестові
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 лінійної регресії
linear_regressor = linear_model.LinearRegression()
linear_regressor.fit(X_train, Y_train)

# Прогнозування результатів
Y_test_pred = linear_regressor.predict(X_test)

# Виведення результатів
print("Linear regressor performance:")
print(f"Mean absolute error = {round(sm.mean_absolute_error(Y_test, Y_test_pred), 2)}")
print(f"Mean squared error = {round(sm.mean_squared_error(Y_test, Y_test_pred), 2)}")
print(f"Median absolute error = {round(sm.median_absolute_error(Y_test,
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 - Лр 02	Арк.
		Пулеко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```
Y_test_pred), 2)}")
print(f"Explain variance score = {round(sm.explained_variance_score(Y_test,
Y_test_pred), 2)}")
print(f"R2 score = {round(sm.r2_score(Y_test, Y_test_pred), 2)}")

# Створення поліноміальної регресії
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(f"Linear regression:\n{linear_regressor.predict(datapoint)}")
print(f"Polynomial regression:\n{poly_linear_model.predict(poly_datapoint)}")
```

```
C:\Python311\python.exe C:/Users/Ola/I
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:
[41.46007151]
```

Рис.3.3. task-3.py

**Завдання 3.4:** Регресія багатьох змінних. Розробіть лінійний регресор, використовуючи набір даних по діабету, який існує в sklearn.datasets.

Набір даних містить 10 вихідних змінних — вік, стать, індекс маси тіла, середній артеріальний тиск і шість вимірювань сироватки крові, отриманих у 442 пацієнтів із цукровим діабетом, а також реакцію, що цікавить, — кількісний показник прогресування захворювання через 1 рік після вихідного рівня. Отже, існує 442 екземпляри з 10 атрибутами. Колонка 11 є кількісною мірою прогресування захворювання через 1 рік після вихідного рівня. Кожен з цих 10 атрибутів був відцентрований по середньому та масштабований за часом

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

стандартного відхилення  $n\_samples$  (тобто сума квадратів кожного стовпця складає 1).

#### Лістинг файлу task-4.py

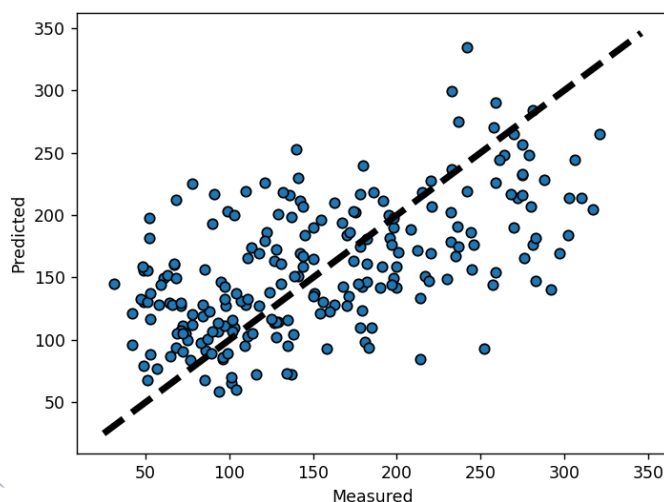
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model, datasets
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import train_test_split

diabetes = datasets.load_diabetes()
X = diabetes.data[:, np.newaxis, 2]
Y = diabetes.target

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.5,
                                                    random_state=0)
regressor = linear_model.LinearRegression()
regressor.fit(X_train, Y_train)
Y_pred = regressor.predict(X_test)

print(f"Mean absolute error = {round(mean_absolute_error(Y_test, Y_pred), 2)}")
print(f"Mean squared error = {round(mean_squared_error(Y_test, Y_pred), 2)}")
print(f"Regression coefficient = {round(regressor.coef_[0], 2)}")
print(f"Regression intercept = {round(regressor.intercept_, 2)}")
print(f"R2 score = {round(r2_score(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('Measured')
ax.set_ylabel('Predicted')
plt.show()
```



```
C:\Python311\python.exe C:/Users/Ola/D
Mean absolute error = 49.51
Mean squared error = 3736.39
Regression coefficient = 1057.06
Regression intercept = 154.13
R2 score = 0.32
```

Рис.3.4. task-4.py

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І. В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

**Завдання 3.5:** Згенеруйте свої випадкові дані обравши за списком відповідно свій варіант (згідно табл. 2.2) 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	13
№ варіанту	3

### Варіант 3

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

#### Лістинг файлу task-5.py

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

m = 100
X = 6 * np.random.rand(m, 1) - 4
Y = 0.5 * X ** 2 + X + 2 + np.random.randn(m, 1)

indices = np.argsort(X, axis=0)
X = X[indices].reshape(-1, 1)
Y = Y[indices].reshape(-1, 1)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.5,
                                                    random_state=0)
regressor = linear_model.LinearRegression()
regressor.fit(X_train, Y_train)
Y_pred = regressor.predict(X_test)

print(f"Mean absolute error = {round(mean_absolute_error(Y_test, Y_pred), 2)}")
print(f"Mean squared error = {round(mean_squared_error(Y_test, Y_pred), 2)}")
print(f"Regression coefficient = {round(regressor.coef_[0][0], 2)}")
print(f"Regression intercept = {round(regressor.intercept_[0], 2)}")
print(f"R2 score = {round(r2_score(Y_test, Y_pred), 2)}")

plt.scatter(X, Y, edgecolors=(0, 0, 0))
plt.plot(X_test, Y_pred, color="red")
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 - Лр 03	Арк.
		Пулеко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

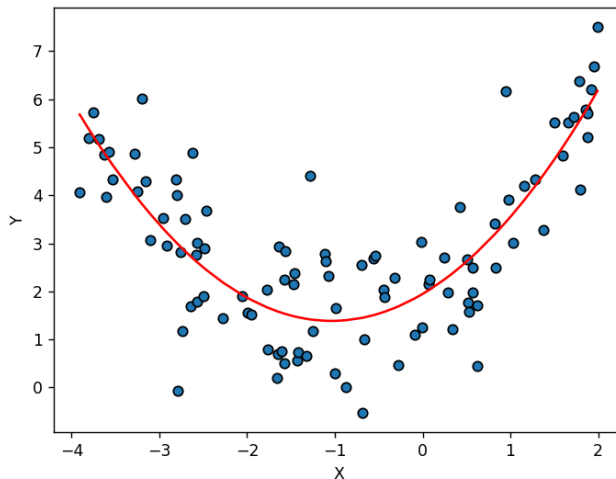
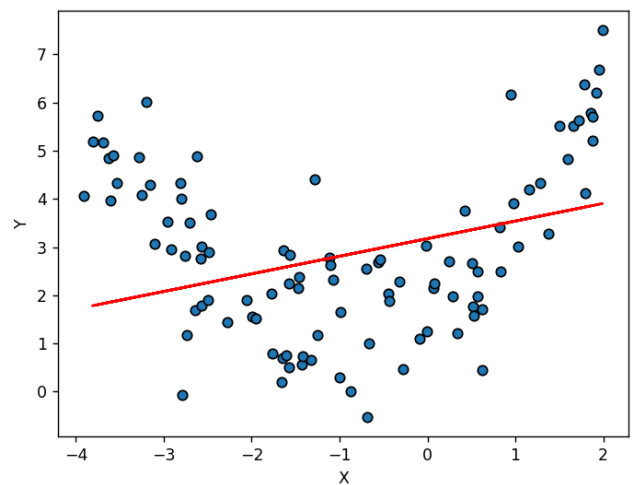
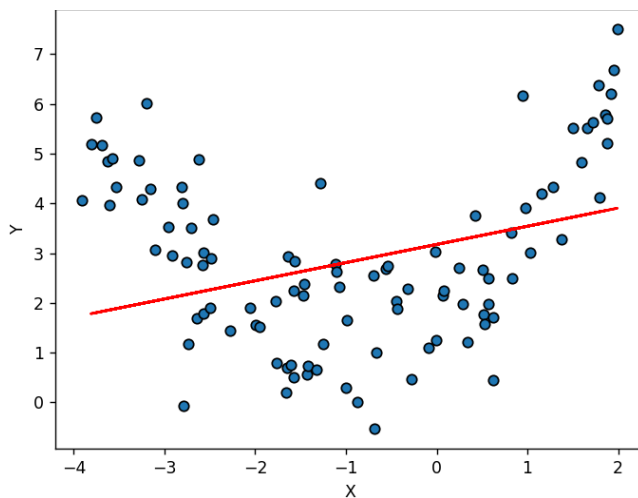
```

poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)
regressor = linear_model.LinearRegression()
regressor.fit(X_poly, Y)
Y_pred = regressor.predict(X_poly)

print(f"Mean absolute error = {round(mean_absolute_error(Y, Y_pred), 2)}")
print(f"Mean squared error = {round(mean_squared_error(Y, Y_pred), 2)}")
print(f"Regression coefficient = {round(regressor.coef_[0][0], 2)}")
print(f"Regression intercept = {round(regressor.intercept_[0], 2)}")
print(f"R2 score = {round(r2_score(Y, Y_pred), 2)}")

plt.scatter(X, Y, edgecolors=(0, 0, 0))
plt.plot(X, Y_pred, color="red")
plt.xlabel('X')
plt.ylabel('Y')
plt.show()

```



```

C:\Python311\python.exe C:/Users/Ola/D...
Mean absolute error = 1.64
Mean squared error = 3.87
Regression coefficient = 0.37
Regression intercept = 3.18
R2 score = -0.14
Mean absolute error = 0.81
Mean squared error = 1.06
Regression coefficient = 0.0
Regression intercept = 1.95
R2 score = 0.66

```

Рис.3.5. task-5.py

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		



**Завдання 3.6:** Побудова кривих навчання. Побудуйте криві навчання для ваших даних у попередньому завданні.

### Лістинг файлу task-6.py

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

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="Training set")
    plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label="Validation set")
    plt.legend(loc="upper right", fontsize=14)
    plt.xlabel("Training set size", fontsize=14)
    plt.ylabel("RMSE", fontsize=14)
    plt.show()

m = 100
X = 6 * np.random.rand(m, 1) - 5
Y = 0.5 * X ** 2 + X + 2 + np.random.randn(m, 1)

indices = np.argsort(X, axis=0)
X = X[indices].reshape(-1, 1)
Y = Y[indices].reshape(-1, 1)
linear_reg = LinearRegression()
plot_learning_curves(linear_reg, X, Y)

polynomial_regression = Pipeline([
    ("poly_features", PolynomialFeatures(degree=10, include_bias=False)),
    ("lin_reg", LinearRegression()),
])
plot_learning_curves(polynomial_regression, X, Y)

polynomial_regression = Pipeline([
    ("poly_features", PolynomialFeatures(degree=2, include_bias=False)),
    ("lin_reg", LinearRegression()),
])
plot_learning_curves(polynomial_regression, X, Y)
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

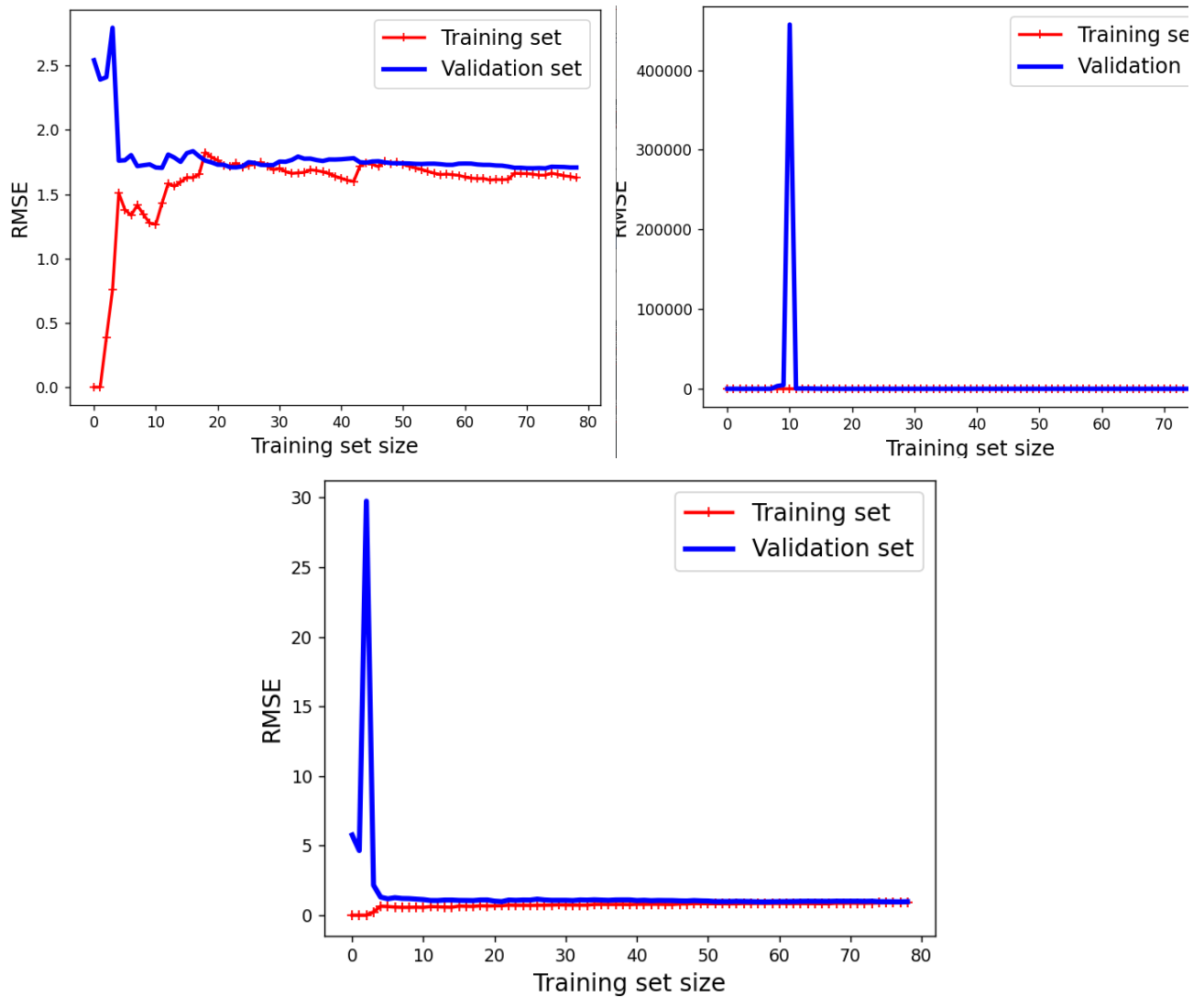


Рис.3.6. task-6.py

**Завдання 3.7:** Кластеризація даних за допомогою методу k-середніх. Провести кластеризацію даних методом k-середніх. Використовувати файл вхідних даних: data\_clustering.txt.

Лістинг файлу task-7.py

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

X = np.loadtxt('data_clustering.txt', delimiter=',')
num_clusters = 5

plt.figure()
plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none', edgecolors='k', s=30)
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')
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()

kmeans = KMeans(init='k-means++', n_clusters=num_clusters, n_init=10)
kmeans.fit(X)

step_size = 0.01
x_values, y_values = np.meshgrid(np.arange(x_min, x_max, step_size),
np.arange(y_min, y_max, step_size))
output = kmeans.predict(np.c_[x_values.ravel(), y_values.ravel()])

output = output.reshape(x_values.shape)
plt.figure()
plt.clf()
plt.imshow(output, interpolation='nearest',
extent=(x_values.min(), x_values.max(), y_values.min(),
y_values.max()),
cmap=plt.cm.Paired, aspect='auto', origin='lower')
plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none', edgecolors='k', s=30)
plt.title('Centroids and boundaries obtained using KMeans')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()
```

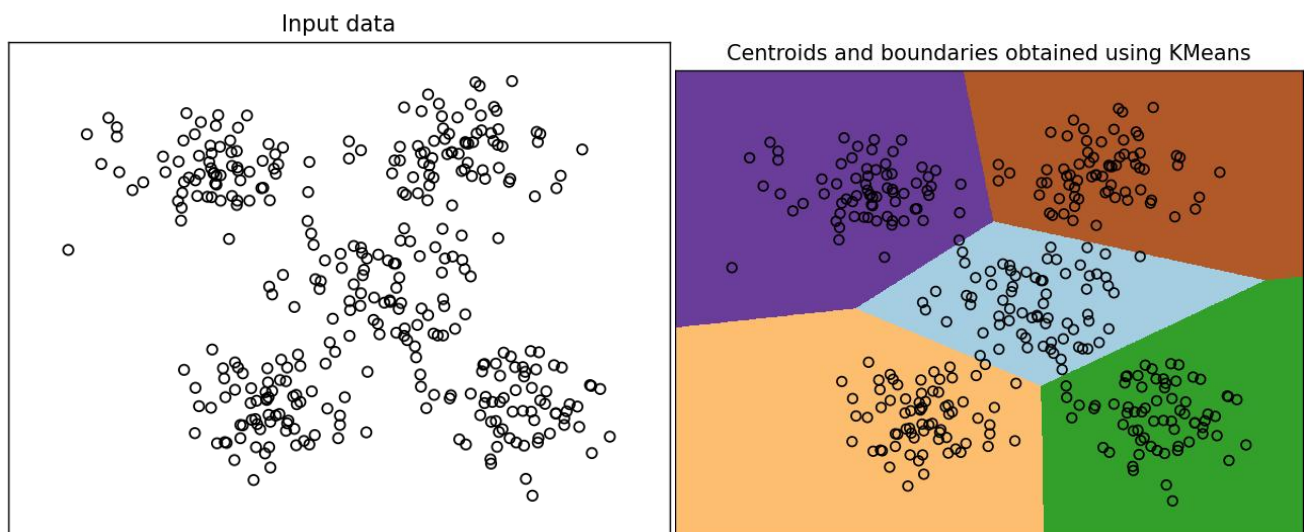


Рис.3.7. task-7.py

**Завдання 3.8:** Кластеризація К-середніх для набору даних Iris. Виконайте кластеризацію К-середніх для набору даних Iris, який включає три типи (класи) квітів ірису (Setosa, Versicolour і Virginica) з чотирма атрибутами: довжина чашолистка, ширина чашолистка, довжина пелюстки та ширина пелюстки.

Лістинг файлу task-8.py

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin
import numpy as np

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

kmeans = KMeans(n_clusters=Y.max() + 1, init='k-means++', n_init=10, max_iter=300,
                tol=0.0001, verbose=0, random_state=None, copy_x=True)
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()

def find_clusters(X, n_clusters, rseed=2):
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]

    while True:
        labels = pairwise_distances_argmin(X, centers)
        new_centers = np.array([X[labels == i].mean(0) for i in
range(n_clusters)])
        if np.all(centers == new_centers):
            break
        centers = new_centers
    return centers, labels

print("using find_clusters():")
centers, labels = find_clusters(X, 3)
print("n_clusters: 3, rseed: 2")
print(labels)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.show()

centers, labels = find_clusters(X, 3, rseed=0)
print("n_clusters: 3, rseed: 0")
print(labels)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.show()

labels = KMeans(3, random_state=0).fit_predict(X)
print("n_clusters: 3, rseed: 0")
print(labels)
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.show()

```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 - Лр 03	Арк.
		Пулюко І.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		



**Завдання 3.9:** Оцінка кількості кластерів з використанням методу зсуву середнього. Відповідно до рекомендацій, напишіть програму та оцініть максимальну кількість кластерів у заданому наборі даних за допомогою алгоритму зсуву середньою. Для аналізу використовуйте дані, які містяться у файлі data\_clustering.txt.

Лістинг файлу task-9.py

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import MeanShift, estimate_bandwidth
from itertools import cycle

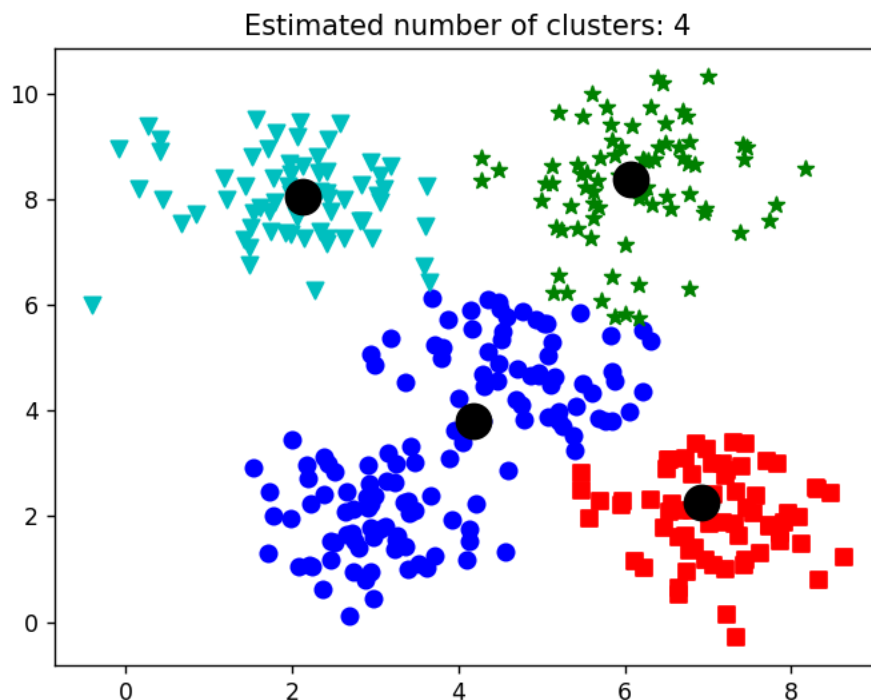
X = np.loadtxt('data_clustering.txt', delimiter=',')
bandwidth = estimate_bandwidth(X, quantile=0.2, n_samples=500)
ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
ms.fit(X)

cluster_centers = ms.cluster_centers_
labels = ms.labels_

print("cluster_centers:\n", cluster_centers)
print("labels:\n", labels)

plt.figure()
markers = cycle('o*sv')
colors = cycle('bgrcmyk')
for i, marker in zip(range(len(cluster_centers)), markers):
    plt.scatter(X[labels == i, 0], X[labels == i, 1], marker=marker,
                color=next(colors), s=50, label='cluster ' + str(i))
    cluster_center = cluster_centers[i]
    plt.plot(cluster_center[0], cluster_center[1], marker='o',
             markerfacecolor='k', markeredgecolor='k', markersize=15)
plt.title(f'Estimated number of clusters: {len(cluster_centers)}')
plt.show()
```

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І. В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```
C:\Python311\python.exe C:/Users/Ola/Documents/stud/4/AI_Python/Lr_3/lab-3/
cluster_centers:
[[4.17639175 3.79773196]
 [6.07428571 8.37909091]
 [6.92078947 2.25723684]
 [2.13478873 8.05690141]]
labels:
[0 3 0 2 1 0 3 0 0 1 0 3 0 2 1 0 3 1 2 1 0 3 0 0 1 0 3 0 2 1 0 3 0 2 1 0 3
0 2 1 0 3 0 2 1 0 3 1 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2
1 0 3 1 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0
3 0 2 1 0 3 3 2 1 0 3 0 2 1 0 3 1 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0
2 1 0 3 1 2 1 0 3 1 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1
0 3 0 2 1 0 3 0 2 1 0 3 1 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3
0 2 1 0 3 0 2 1 0 3 3 2 1 0 3 0 2 1 0 3 1 2 1 0 3 0 2 1 0 3 0 2
1 0 3 0 2 1 0 3 0 2 1 0 3 1 2 1 0 1 0 2 1 0 3 0 2 1 0 3 0 2 1 0
3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1 0 3 0
2 1 0 3 0 2 1 0 3 0 2 1 0 3 0 2 1]
```

Рис.3.9. task-9.py

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І. В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

[https://github.com/avrorilka/AI\\_Python](https://github.com/avrorilka/AI_Python)

**Висновки:** в ході виконання лабораторної роботи використовуючи спеціалізовані бібліотеки та мову програмування Python ми дослідити методи методи регресії та неконтрольованої класифікації даних у машинному навчанні. Також ми розглянули поняття регресії, розглянули правила побудови регресійних моделей.

		Маковська О.Ю.			ДУ«Житомирська політехніка».21.121.13.000 – Лр 03	Арк.
		Пулеко І. В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		