

ЛАБОРАТОРНА РОБОТА № 3

ПОПЕРЕДНЯ ОБРОБКА ТА КОНТРОЛЬОВАНА КЛАСИФІКАЦІЯ ДАНИХ

Мета: використовуючи спеціалізовані бібліотеки та мову програмування Python, дослідити попередню обробку та класифікацію даних.

Хід роботи:

Посилання на репозиторій: <https://github.com/AlexanderSydorchuk/AI-Lab3>

Завдання 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.pyplot as plt

# Вхідний файл
input_file = 'data_singlevar_regr.txt'

# Завантаження даних
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)

# 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()
```

					ДУ «Житомирська політехніка». 19.121.25.000 - Лр1			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи	Лім.	Арк.	Аркушів
Розроб.		Сидорчук О.С.					1	5
Перевір.		Голенко М.Ю				ФІКТ Гр. ІПЗ-20-2[2]		
Керівник								
Н. контр.								
Зав. каф.								

```

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

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

```

Результат виконання програми:

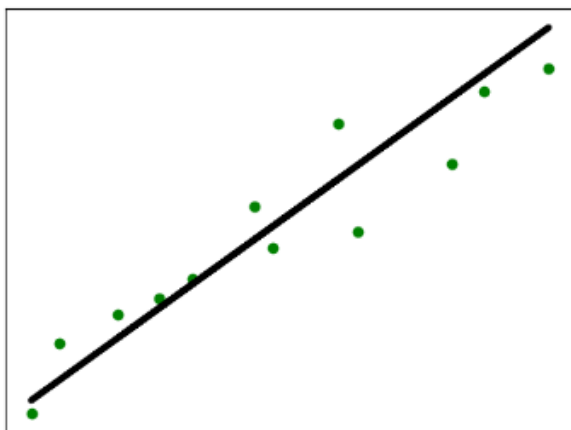
```

"D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\venv\Scripts\python.exe" "D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\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

```



Висновок: модель для вихідних даних побудована правильно. Показники MAE, MSE вказують на середню якість, а показник R2 видає добрий результат.

		Сидорчук О.С.			ДУ «Житомирська політехніка». 19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2. Побудувати регресійну модель на основі однієї змінної.

Використовувати вхідні дані відповідно свого варіанту, що визначається за списком групи у журналі (таблиця 2.1).

24
4

Лістинг програми:

```
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_4.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 = 'model2.pkl'
with open(output_model_file, 'wb') as f:
    pickle.dump(regressor, f)

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

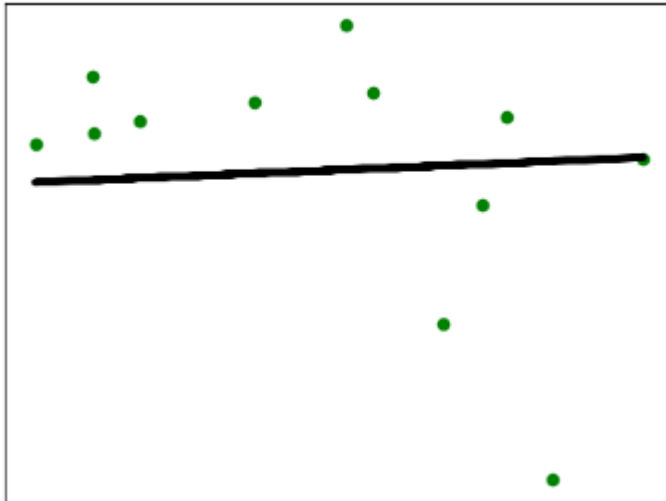
		Сидорчук О.С.			ДУ «Житомирська політехніка».19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання програми:

```
"D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\venv\Scripts\python.exe" "D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\LR_3_task_2.py"
Linear regressor performance:
Mean absolute error = 2.72
Mean squared error = 13.16
Median absolute error = 1.9
Explain variance score = -0.07
R2 score = -0.07

New mean absolute error = 2.72

Process finished with exit code 0
```



Завдання 3. Створення багатовимірного регресора. Використовувати файл вхідних даних: data_multivar_regr.txt, побудувати регресійну модель на основі багатьох змінних.

Лістинг програми:

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

input_file = 'data_multivar_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:]

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

		Сидорчук О.С.			ДУ «Житомирська політехніка».19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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))

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

```

Результат виконання програми:

```

"D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\venv\Scripts\python.exe" "D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\LR_3_task_3.py"
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:
[41.45901869]

Process finished with exit code 0

```

Висновок: У порівнянні із лінійним регресором, поліноміальний регресор демонструє кращі результати, в якого значення вище.

Завдання 4. Регресія багатьох змінних

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

Лістинг програми:

		Сидорчук О.С.			ДУ «Житомирська політехніка».19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

import matplotlib.pyplot as plt
import numpy as np
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

diabetes = datasets.load_diabetes()
X = diabetes.data
y = diabetes.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5,
random_state=0)

regression = linear_model.LinearRegression()
regression.fit(X_train, y_train)
y_pred = regression.predict(X_test)

print("Linear regressor performance:")
print("regression.coef_ =", regression.coef_)
print("regression.intercept_ =", regression.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()

```

Результат виконання програми:

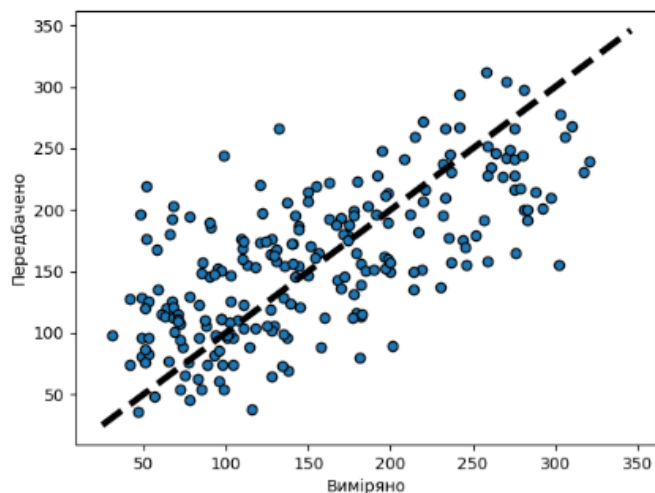
```

"D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\venv\Scripts\python.exe" "D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\LR_3_task_4.py"
Linear regressor performance:
regression.coef_ = [ -20.4047621  -265.88518066  564.65086437  325.56226865 -692.16120333
  395.55720874  23.49659361  116.36402337  843.94613929  12.71856131]
regression.intercept_ = 154.3589285280134
R2 score = 0.44
Mean absolute error = 44.8
Mean squared error = 3075.33

Process finished with exit code 0

```

		Сидорчук О.С.			ДУ «Житомирська політехніка».19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				
Змн.	Арк.	№ докум.	Підпис	Дата		6



Завдання 5. Самостійна побудова регресії

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.

24

4

Варіант 4

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

Лістинг програми:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.preprocessing import PolynomialFeatures

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

fig, ax = plt.subplots()
ax.scatter(X, y, edgecolors=(0, 0, 0))
plt.title("Лінійна регресія")
plt.show()

print(X[1], y[1])
```

		Сидорчук О.С.			ДУ «Житомирська політехніка». 19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

poly_features = PolynomialFeatures(degree=3, include_bias=False)
X_poly = poly_features.fit_transform(np.array(X).reshape(-1, 1))

linear_regression = linear_model.LinearRegression()
linear_regression.fit(X_poly, y)
print(linear_regression.intercept_, linear_regression.coef_)
y_pred = linear_regression.predict(X_poly)

fig, ax = plt.subplots()
ax.scatter(X, y, edgecolors=(0, 0, 0))
plt.plot(X, y_pred, color='red', linewidth=4)
plt.title("Поліноміальна регресія")
plt.show()

```

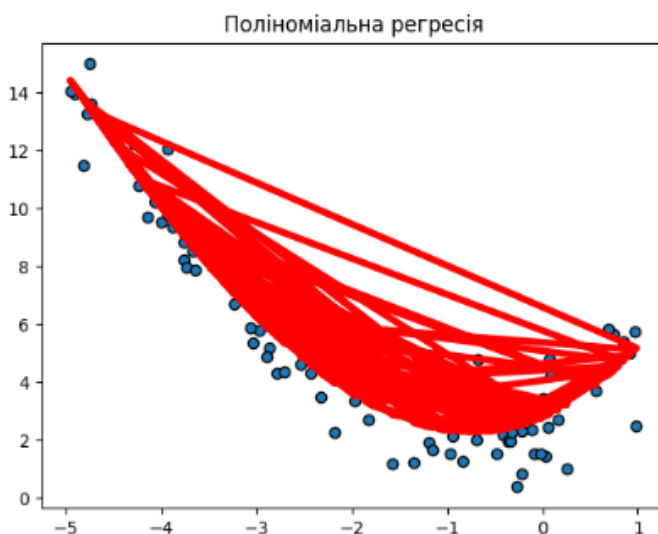
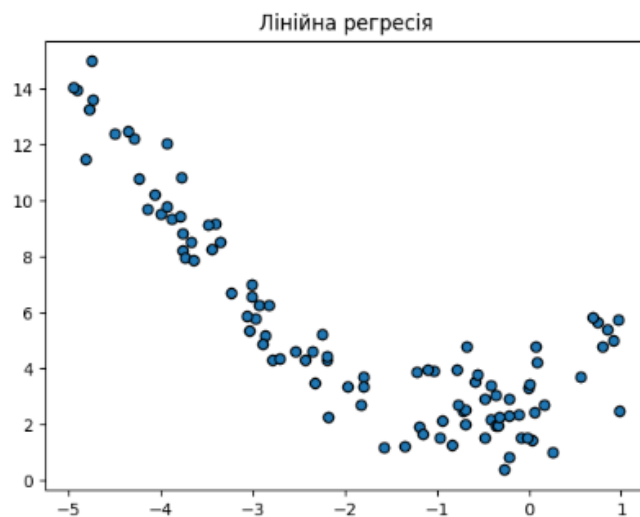
Результат виконання програми:

```

"D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\venv\Scripts\python.exe" "D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\LR_3_task_5.py"
[-0.67346673] [3.05984317]
[3.16461321] [[1.19449032 0.74251317 0.00383585]]

Process finished with exit code 0

```



		Сидорчук О.С.			ДУ «Житомирська політехніка». 19.121.25.000 - Лр1	Арк.
		Голенко М.Ю.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Модель: $0,7x^2 + x + 3$

Модель регресії з передбаченими коефіцієнтами:

$$0,8x^2 + 0.64 + 0.003$$

Завдання 6. Побудова кривих навчання

Побудуйте криві навчання для ваших даних у попередньому завданні.

Лістинг програми:

```
import matplotlib.pyplot as plt
import numpy as np
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
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import Pipeline

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

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))
    fig, ax = plt.subplots()
    plt.ylim(0, 2)
    ax.plot(np.sqrt(train_errors), "r-+", linewidth=2, label='train')
    ax.plot(np.sqrt(val_errors), "b-", linewidth=3, label='val')
    plt.show()

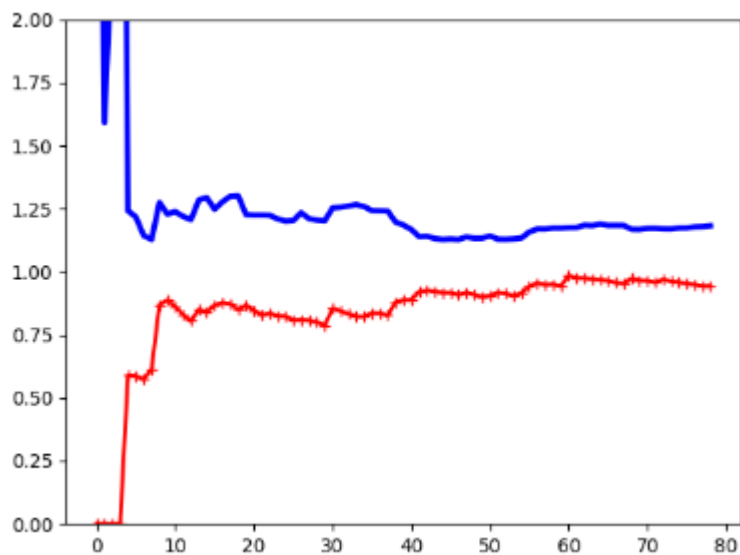
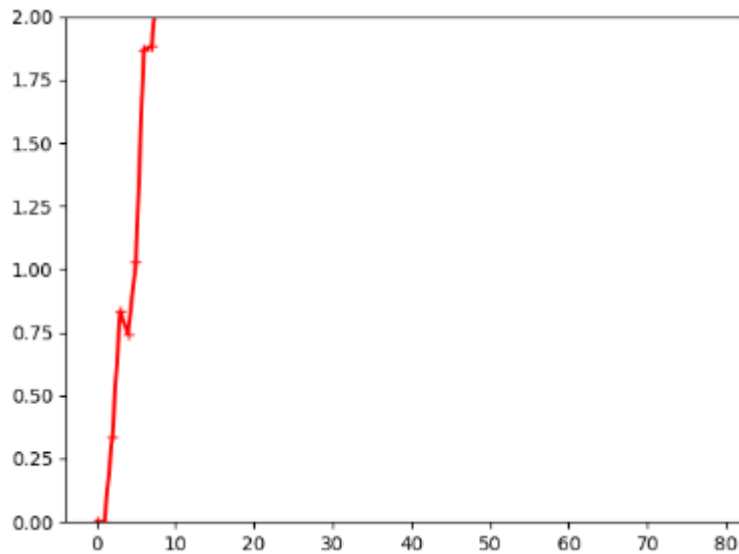
linear_regression = linear_model.LinearRegression()
plot_learning_curves(linear_regression, np.array(X).reshape(-1, 1), y)

polynomial_regression = Pipeline([
    ('poly_features', PolynomialFeatures(degree=3, include_bias=False)),
    ('lin_reg', linear_model.LinearRegression()),
])

plot_learning_curves(polynomial_regression, np.array(X).reshape(-1, 1), y)
```

		Сидорчук О.С.			ДУ «Житомирська політехніка».19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання програми:



Завдання 7. Кластеризація даних за допомогою методу k-середніх.

Провести кластеризацію даних методом k-середніх. Використовувати файл вхідних даних: data_clustering.txt.

Лістинг програми:

```
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='black', s=80)
```

		Сидорчук О.С.			ДУ «Житомирська політехніка». 19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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

step_size = 0.01
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))
output = kmeans.predict(np.c_[x_vals.ravel(), y_vals.ravel()])
output = output.reshape(x_vals.shape)

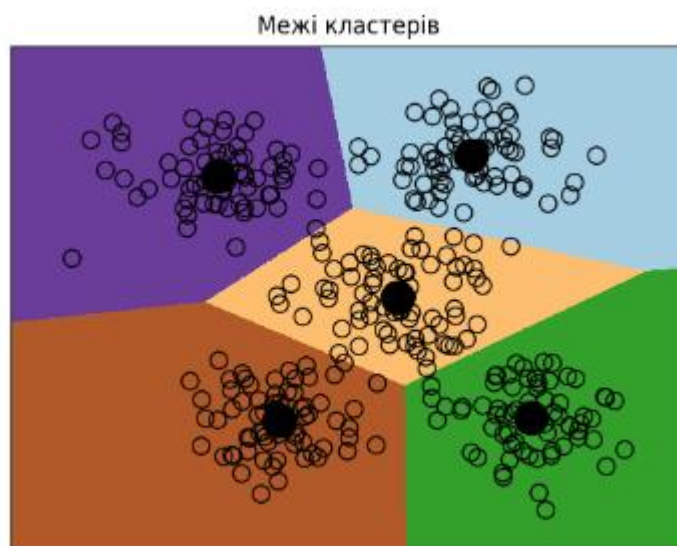
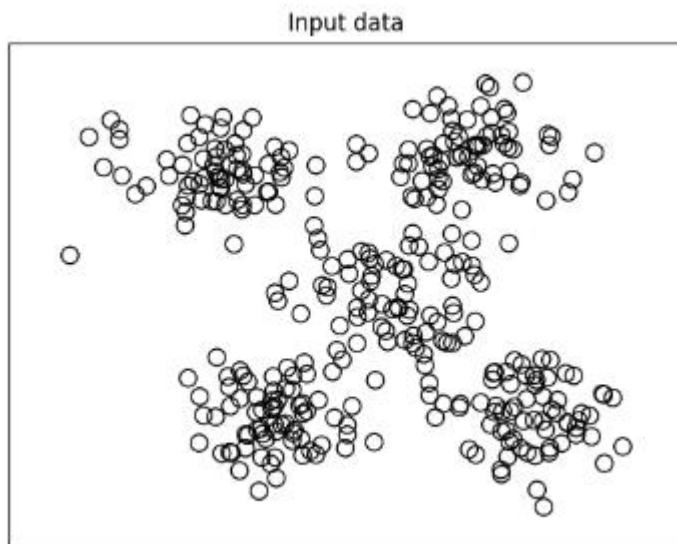
plt.figure()
plt.clf()
plt.imshow(output, interpolation='nearest',
           extent=(x_vals.min(), x_vals.max(),
                  y_vals.min(), y_vals.max()),
           cmap=plt.cm.Paired,
           aspect='auto',
           origin='lower')
plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none',
           edgecolors='black', s=80)
cluster_centers = kmeans.cluster_centers_
plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1],
           marker='o', s=210, linewidths=4, color='black',
           zorder=12, facecolors='black')

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('Межі кластерів')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()

```

Результат виконання програми:

		Сидорчук О.С.			ДУ «Житомирська політехніка». 19.121.25.000 - Лр1	Арк.
		Голенко М.Ю.				11
Змн.	Арк.	№ докум.	Підпис	Дата		



Висновок: метод k-середніх валідно працює, але лише за умови відомої кількості кластерів.

Завдання 8. Кластеризація K-середніх для набору даних Iris Виконайте кластеризацію K-середніх для набору даних Iris, який включає три типи (класи) квітів ірису (Setosa, Versicolour і Virginica) чотирма атрибутами: довжина чашолистка, ширина чашолистка, довжина пелюстки та ширина пелюстки. У цьому завданні використовуйте `sklearn.cluster.KMeans` для пошуку кластерів набору даних Iris.

Лістинг програми:

		Сидорчук О.С.			ДУ «Житомирська політехніка». 19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				12
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
import warnings
warnings.filterwarnings("ignore")

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, ran-
dom_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")
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.show()

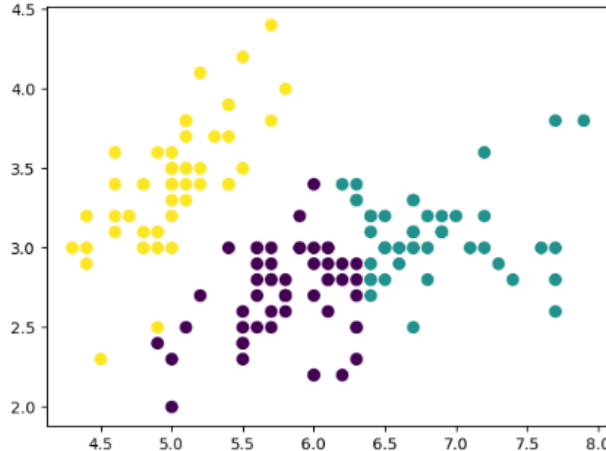
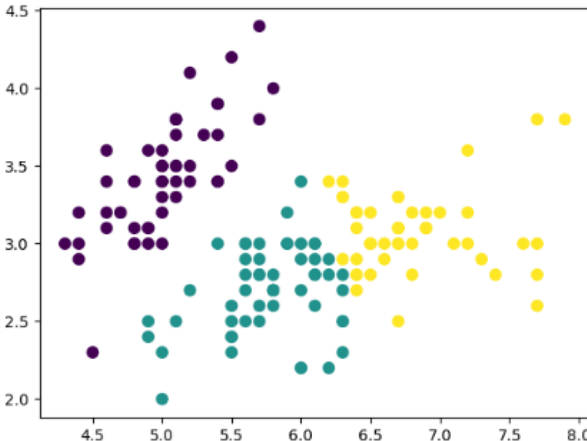
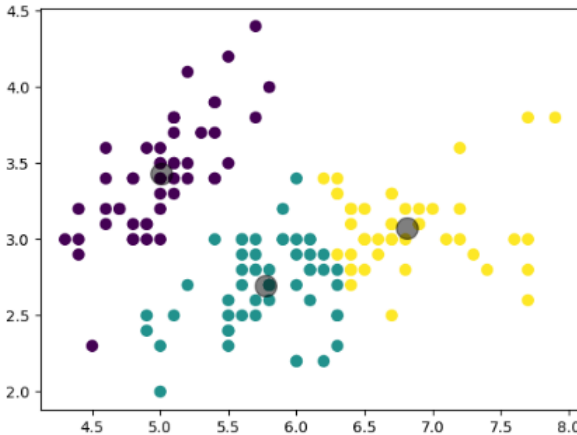
centers, labels = find_clusters(X, 3, rseed=0)
print("n_clusters: 3, rseed: 0")
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.show()

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

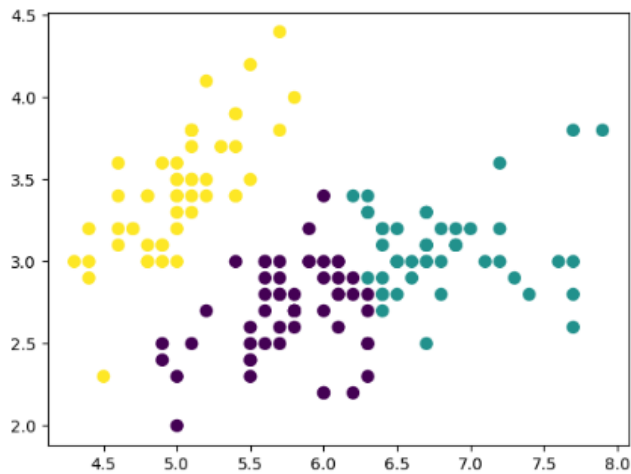
```

Результат виконання програми:

		Сидорчук О.С.			ДУ «Житомирська політехніка».19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				13
Змн.	Арк.	№ докум.	Підпис	Дата		

[illegible]

		Сидорчук О.С.			ДУ «Житомирська політехніка». 19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				14
Змн.	Арк.	№ докум.	Підпис	Дата		



Завдання 9. Оцінка кількості кластерів з використанням методу зсуву середнього

Лістинг програми:

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

# Завантаження даних
X = np.loadtxt('data_clustering.txt', delimiter=',')

# Оцінка ширини вікна для X
bandwidth_X = estimate_bandwidth(X, quantile=0.1, n_samples=len(X))

# Кластеризація даних методом зсуву середнього
meanshift_model = MeanShift(bandwidth=bandwidth_X, bin_seeding=True)
meanshift_model.fit(X)

# Витягування центрів кластерів
cluster_centers = meanshift_model.cluster_centers_
print('\nCenters of clusters:\n', cluster_centers)

# Оцінка кількості кластерів
labels = meanshift_model.labels_
num_clusters = len(np.unique(labels))
print("\nNumber of clusters in input data =", num_clusters)

# Відображення на графіку точок та центрів кластерів
plt.figure()
markers = 'o*xvs'
for i, marker in zip(range(num_clusters), markers):
    # Відображення на графіку точок, що належать поточному кластеру
    plt.scatter(X[labels == i, 0], X[labels == i, 1], marker=marker,
                color=np.random.rand(3,))
```

		Сидорчук О.С.			ДУ «Житомирська політехніка». 19.121.25.000 - Лр1	Арк.
		Голенко М.Ю.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

```
# Відображення на графіку центру кластера
cluster_center = cluster_centers[i]
plt.plot(cluster_center[0], cluster_center[1], marker='o',
         markerfacecolor='black', markeredgecolor='red',
         markersize=15)

plt.title('Кластери')
plt.show()
```

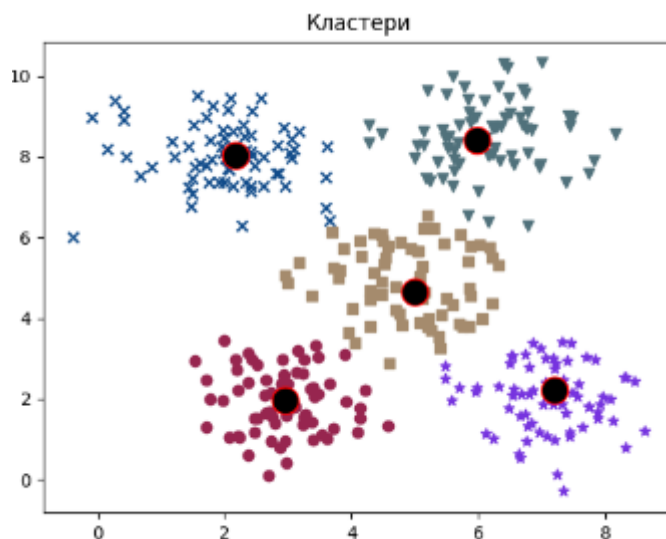
Результат виконання програми:

```
"D:\DFiles\LP\4 Karzen\1 Rihrtze\AI\Lab3\venv\Scripts\python.exe" "D:\DFiles\LP\4 Karzen\1 Rihrtze\AI\Lab3\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

Process finished with exit code 0
```



Завдання 10. Знаходження підгруп на фондовому ринку з використанням моделі поширення подібності

Лістинг програми:

```
import datetime
import json
import numpy as np
from sklearn import covariance, cluster
import yfinance as yf

input_file = 'company_symbol_mapping.json'
with open(input_file, 'r') as f:
```

		Сидорчук О.С.			ДУ «Житомирська політехніка».19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				16
Змн.	Арк.	№ докум.	Підпис	Дата		


```

company_symbols_map = json.loads(f.read())

symbols, names = np.array(list(company_symbols_map.items())).T

start_date = datetime.datetime(2003, 7, 3)
end_date = datetime.datetime(2007, 5, 4)

quotes = []
for symbol in symbols:
    quote = yf.Ticker(symbol).history(start=start_date, end=end_date)
    quotes.append(quote)

opening_quotes = np.array([quote['Open'].values for quote in
quotes]).astype(float)
closing_quotes = np.array([quote['Close'].values for quote in
quotes]).astype(float)

quotes_diff = closing_quotes - opening_quotes

X = quotes_diff.copy().T
X /= X.std(axis=0)

edge_model = covariance.GraphicalLassoCV()

with np.errstate(invalid='ignore'):
    edge_model.fit(X)

_, labels = cluster.affinity_propagation(edge_model.covariance_)
num_labels = labels.max()

for i in range(num_labels + 1):
    cluster_names = names[labels == i]
    if len(cluster_names) > 0:
        print("Cluster", i+1, "==>", ', '.join(cluster_names))

```

Результат виконання програми:

```

"D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\venv\Scripts\python.exe" "D:\DFiles\LP\4 Karzen\1 Rihtze\AI\Lab3\LR_3_task_10.py"
Cluster 1 ==> Microsoft, IBM, Amazon, Ford, 3M, Mc Donalds, Apple
Cluster 2 ==> Northrop Grumman, Boeing
Cluster 3 ==> Coca Cola, Pepsi, Kellogg, Procter Gamble

Process finished with exit code 0

```

Висновки: в ході лабораторної роботи, використовуючи спеціалізовані бібліотеки та мову програмування Python, було досліджено попередню обробку та класифікацію даних.

		Сидорчук О.С.			ДУ «Житомирська політехніка».19.121.25.000 - Лр1	Арк.
		Голенко М.Ю				17
Змн.	Арк.	№ докум.	Підпис	Дата		