

ЛАБОРАТОРНА РОБОТА № 3

ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ ТА НЕКОНТРОЛЬОВАНОГО НАВЧАННЯ

Посилання на GitHub: <https://github.com/enot1k666/labsOAI.git>

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

2. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ ТА МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ЙОГО ВИКОНАННЯ

Завдання 2.1. Створення регресора однієї змінної

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

Лістинг програми:

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

# Вхідний файл, який містить дані
input_file = 'data_singlevar_regr.txt'

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

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()

print("Linear regressor performance:")
print("Mean absolute error =",
      round(sm.mean_absolute_error(y_test, y_test_pred), 2))
print("Mean squared error =",
```

					ДУ «Житомирська політехніка».22.121.18.000 – Лр3			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи	Лім.	Арк.	Аркушів
Розроб.		Скоківський В.						
Перевір.		Голенко М.Ю.					1	
Керівник						ФІКТ Гр. ІПЗ-20-3		
Н. контр.								
Зав. каф.								

```

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:

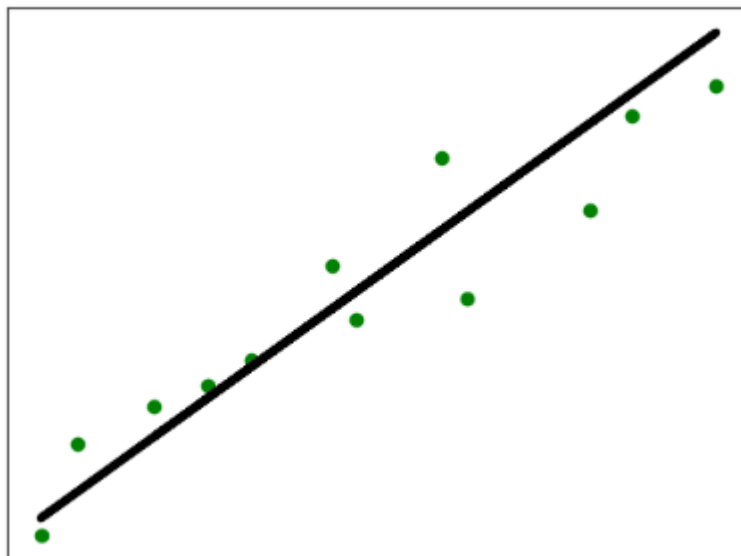
```

C:\Users\skoki\PycharmProjects\labs0AI\.venv\Scripts\python.exe C:\Users\skoki\PycharmProjects\labs0AI\lab3\task1.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

```



Регресійний аналіз за допомогою лінійної регресії

Висновок: Модель лінійної регресії була піддана тренуванню на 80% даних і оцінена на залишкових 20%. У результаті отримані такі метрики: середня абсолютна помилка – 0.59, середня квадратична різниця – 0.49, медіана абсолютних поми-

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – ЛрЗ	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

лок – 0.51, оцінка поясненої дисперсії – 0.86. При відновленні моделі з використанням збереженого файлу нова абсолютна помилка також дорівнює 0.59, що свідчить про те, що збережена та завантажена модель демонструють однакові результати.

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

Завдання 2.2. Передбачення за допомогою регресії однієї змінної

Побудувати регресійну модель на основі однієї змінної. Використовувати вхідні дані відповідно свого варіанту, що визначається за списком групи у журналі (таблиця 2.1).

Таблиця 2.1

№ за списком	11	12	13	14	15	16	17	18	19	20
№ варіанту	1	2	3	4	5	1	2	3	4	5

Варіант 3 файл: data_regr_3.txt

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_3.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))
```

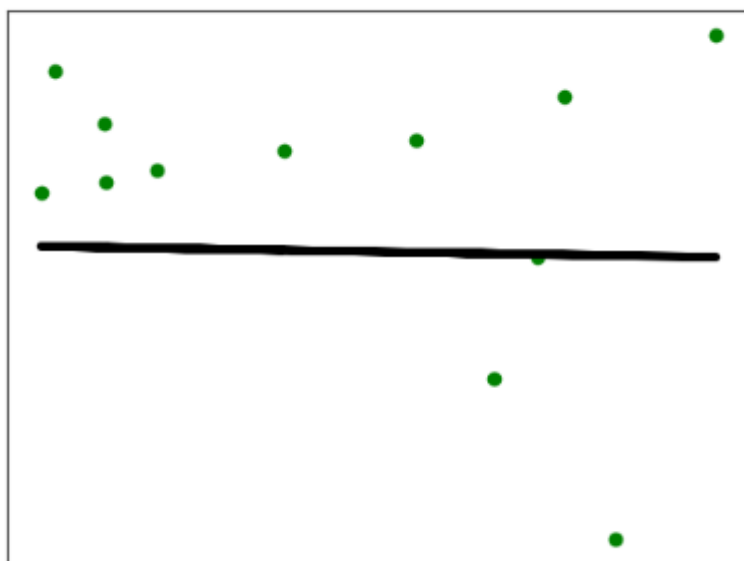
		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Результат виконання:

```
C:\Users\skoki\PycharmProjects\labs0AI\.venv\Scripts\python.exe C:\Users\skoki\PycharmProjects\labs0AI\lab3\task2.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

Process finished with exit code 0
```



Висновок: Модель лінійної регресії була піддана тренуванню на 80% даних і оцінена на залишкових 20%. Отримані метрики вказують на наступне: середня абсолютна помилка – 3.59, середня квадратична різниця – 17.39 (що свідчить про великі відхилення у прогнозах), медіана абсолютних помилок – 3.39, оцінка поясненої дисперсії – 0.02 (прогнози значно відрізняються від фактичних значень). Нова абсолютна помилка для моделі, відновленої з використанням збереженого файлу, також дорівнює 3.59, що підтверджує, що збережена і завантажена модель повертає ті ж самі результати.

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

Завдання 2.3. 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)

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:

```
C:\Users\skoki\PycharmProjects\labs0AI\venv\Scripts\python.exe C:\Users\skoki\PycharmProjects\labs0AI\lab3\task3.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:
[41.45421524]

Process finished with exit code 0
```

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

Висновок: Середня абсолютна помилка становить 3.58, середня квадратична різниця дорівнює 20.31 (значення вище ніж попереднє, що свідчить про великі відхилення у прогнозах), медіана абсолютних помилок дорівнює 2.99, оцінка поясненої дисперсії становить 0.86 (прогнози схожі до фактичних значень). Для поліноміальної регресії використовується поліном 10-го ступеня, для вхідного значення [7.75, 6.35, 5.56], поліноміальна регресійна модель прогнозує значення 41.46, в той час як лінійна регресія прогнозує значення 36.05. Обидві моделі мають схожі значення показників, що свідчить про те, що обидві моделі надають приблизно однакові результати.

Завдання 2.4. Регресія багатьох змінних

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

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)

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

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

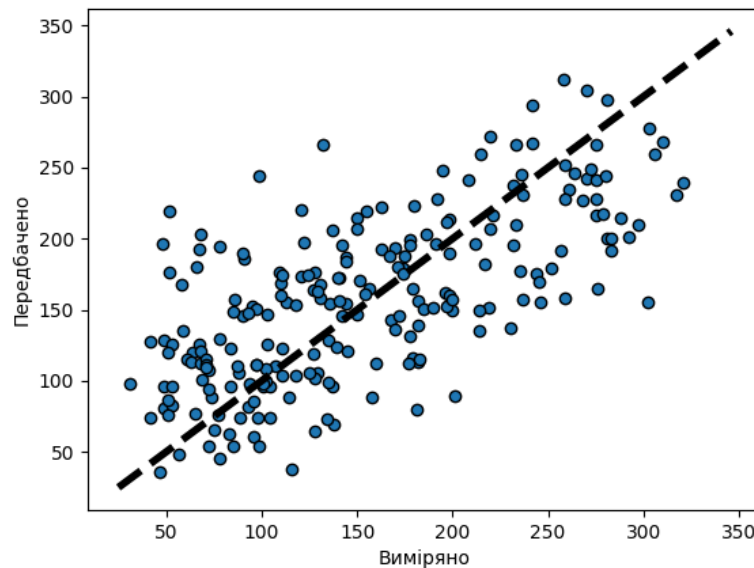
fig, ax = plt.subplots()
ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4)
ax.set_xlabel('Виміряно')
ax.set_ylabel('Передбачено')
plt.show()
```

Результат виконання:

```
C:\Users\skoki\PycharmProjects\labs0AI\.venv\Scripts\python.exe C:\Users\skoki\PycharmProjects\labs0AI\lab3\task4.py
Linear regressor performance:
regr.coef_ = [ -20.4047621 -265.88518066  564.65086437  325.56226865 -692.16120333
  395.55720874  23.49659361  116.36402337  843.94613929  12.71856131]
regr.intercept_ = 154.3589285280134
R2 score = 0.44
Mean absolute error = 44.8
Mean squared error = 3075.33

Process finished with exit code 0
```

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				7
Змн.	Арк.	№ докум.	Підпис	Дата		



Висновок: Було виконано лінійну регресію для набору даних «Diabetes» та було отримано такі результати якості: **Коефіцієнти регресії** представляються як масив чисел, вони вказують на вагу кожної ознаки $[-20.4047621, -265.88518, 564.65086437, 325.56226865, 692.16120333, 395.55720874, 23.49659361, 116.36402337, 843.94613929, 12.71856131]$, **Перетин** рівний 154.36 і представляє відсоток, на який зміщується пряма регресії, **Оцінка R2** дорівнює 0.44 (модель пояснює близько 44% варіації в цільовій змінній), **Середня Абсолютна різниця** становить 44.8, **Середня Квадратична Оцінка** дорівнює 3075.33 (показує різницю між прогнозованим і фактичним значеннями). Модель має обмежену ефективність, що підтверджує значення R2)

		Скоківський В.			ДУ «Житомирська політехніка». 22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.5. 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.

Таблиця 2.2

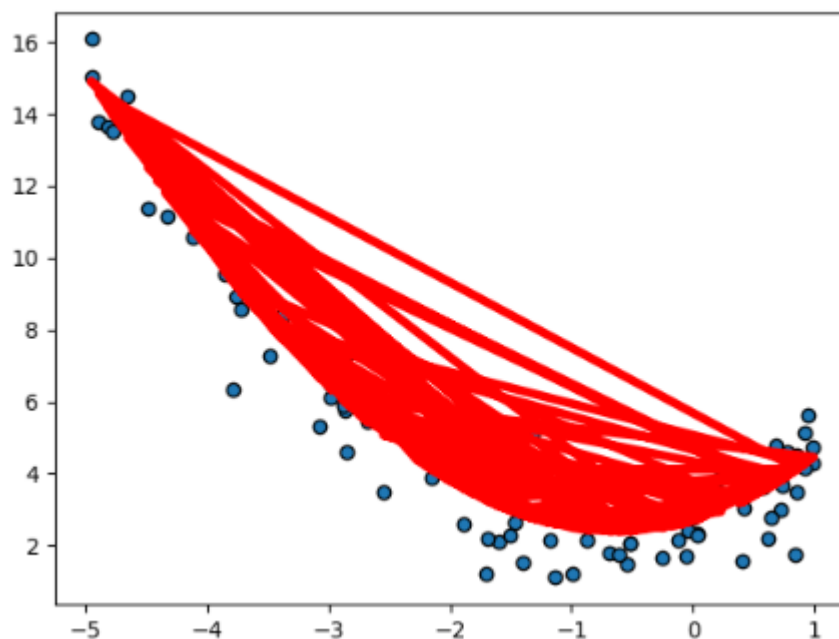
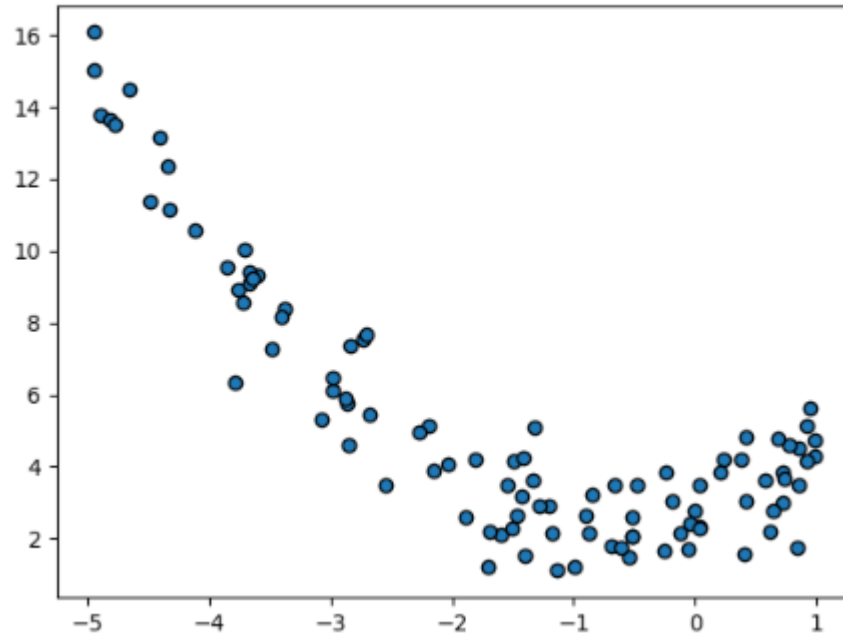
Лістинг програми:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import 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.show()
poly_features = PolynomialFeatures(degree=3, include_bias=False)
X_poly = poly_features.fit_transform(X)
print("X[0] = ",X[0])
print("X[1] = ",X[1])
print("Y[1] = ",y[1])
lin_reg = linear_model.LinearRegression()
lin_reg.fit(X_poly, y)
print("Перетин:",lin_reg.intercept_,"Коефіцієнти регресії", lin_reg.coef_)
y_pred = lin_reg.predict(X_poly)
print("X_poly = ",X_poly)
fig, ax = plt.subplots()
ax.scatter(X, y, edgecolors=(0, 0, 0))
plt.plot(X, y_pred, color='red', linewidth=4)
plt.show()
```

Результат виконання:

```
C:\Users\skoki\PycharmProjects\labs0AI\venv\Scripts\python.exe C:\Users\skoki\PycharmProjects\labs0AI\lab3\task5.py
X[0] = [-1.59591606]
X[1] = [-0.48231179]
Y[1] = [3.4939998]
Перетин: [2.68172353] Коефіцієнти регресії [[0.97095503 0.80659551 0.02275606]]
X_poly = [[-1.59591606e+00  2.54694807e+00 -4.06471534e+00]
 [-4.82311792e-01  2.32624665e-01 -1.12197619e-01]
 [-4.65148220e+00  2.16362867e+01 -1.00640802e+02]
 [-3.07470829e+00  9.45383105e+00 -2.90677727e+01]
 [ 4.17157667e-01  1.74020519e-01  7.25939938e-02]
 [-2.19266327e+00  4.80777221e+00 -1.05418255e+01]
 [-8.70738471e-01  7.58185485e-01 -6.60181270e-01]
 [-5.51450118e-01  3.04097232e-01 -1.67694455e-01]
 [-2.73378111e+00  7.47355917e+00 -2.04310749e+01]
 [-3.71640071e+00  1.38116342e+01 -5.13295672e+01]
 [-2.67814902e+00  7.17248219e+00 -1.92089762e+01]
 [-4.11526586e+00  1.69354131e+01 -6.96937276e+01]
 [-1.40612839e+00  1.97719705e+00 -2.78019290e+00]
 [-2.83956218e+00  8.06311340e+00 -2.28957119e+01]
 [-1.50006449e+00  2.25019348e+00 -3.37543533e+00]
 [-1.69885696e+00  2.88611497e+00 -4.90309650e+00]
 [ 8.36539097e-01  6.99797660e-01  5.85408102e-01]
 [ 4.05624789e-01  1.64531469e-01  6.67380426e-02]
 [-3.78228923e+00  1.43057118e+01 -5.41083396e+01]
 [-4.89140111e+00  2.39258048e+01 -1.17030708e+02]
 [ 8.57741494e-01  7.35720470e-01  6.31057975e-01]
 [-2.15751732e+00  4.65488100e+00 -1.00429864e+01]]
```

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – ЛрЗ	Арк.
		Голенко М.Ю.				9
Змн.	Арк.	№ докум.	Підпис	Дата		



Графіки моделі

Висновок: Математичне рівняння для даної моделі:

$$y = 0.7x_1^2 + 1.0x_1 + 3 + \text{гауссов шум}$$

Отримана модель регресії з передбаченими коефіцієнтами:

$$y = 3.07618256 + 1.09246979x_1 + 0.61181343x_2 - 0.02138683x_3$$

		Скоківський В.			ДУ «Житомирська політехніка». 22.121.18.000 – ЛрЗ	Арк.
		Голенко М.Ю.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Результати регресії:

$X[0] = [-1.59591606]$

$X[1] = [-0.48231179]$ – значення X для першого спостереження.

$Y[1] = [3.4939998]$ – відповідне значення Y для першого спостереження.

Перетин: $[2.68172353]$ Коефіцієнти регресії $[0.97095503, 0.80659551, 0.02275606]$

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.6. 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 linear_model
from sklearn.metrics import mean_squared_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()

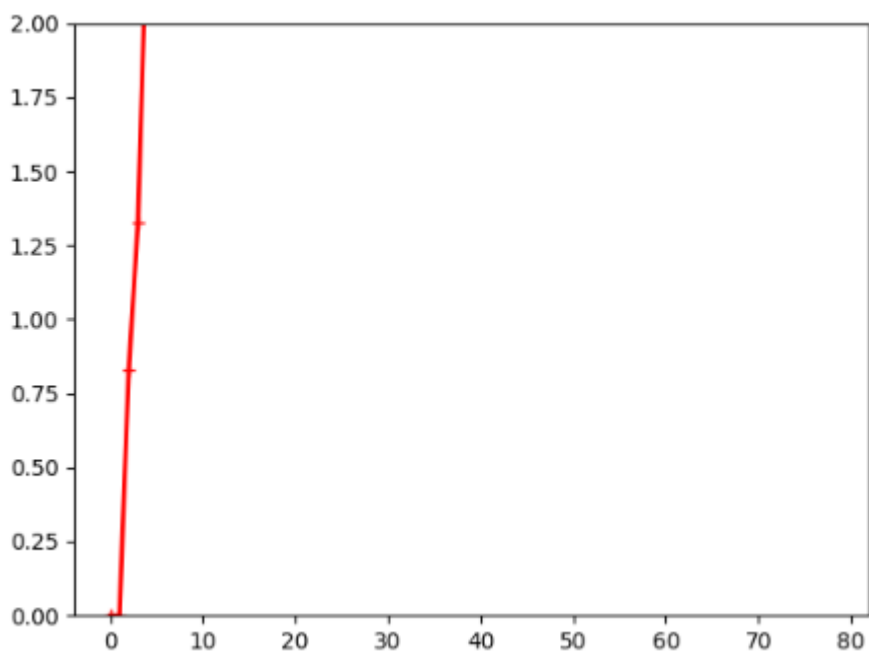
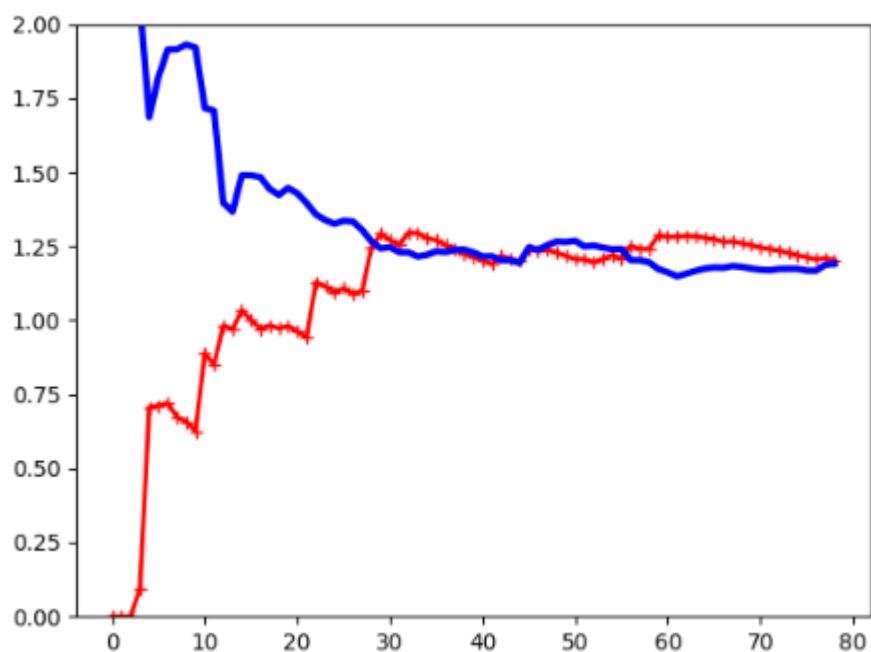
lin_reg = linear_model.LinearRegression()
plot_learning_curves(lin_reg, X, y)

polynomial_regression = Pipeline([
    ('poly_features', PolynomialFeatures(degree=2, include_bias=False)),
    ('lin_reg', linear_model.LinearRegression()),
])

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

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – ЛрЗ	Арк.
		Голенко М.Ю.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:



		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

2. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ ТА МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ЙОГО ВИКОНАННЯ

Завдання 2.7. 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
from sklearn import metrics

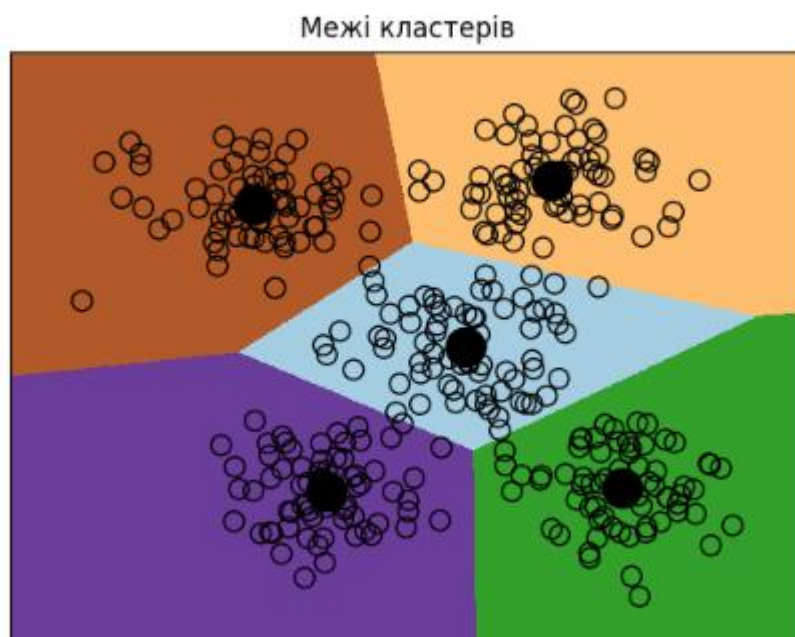
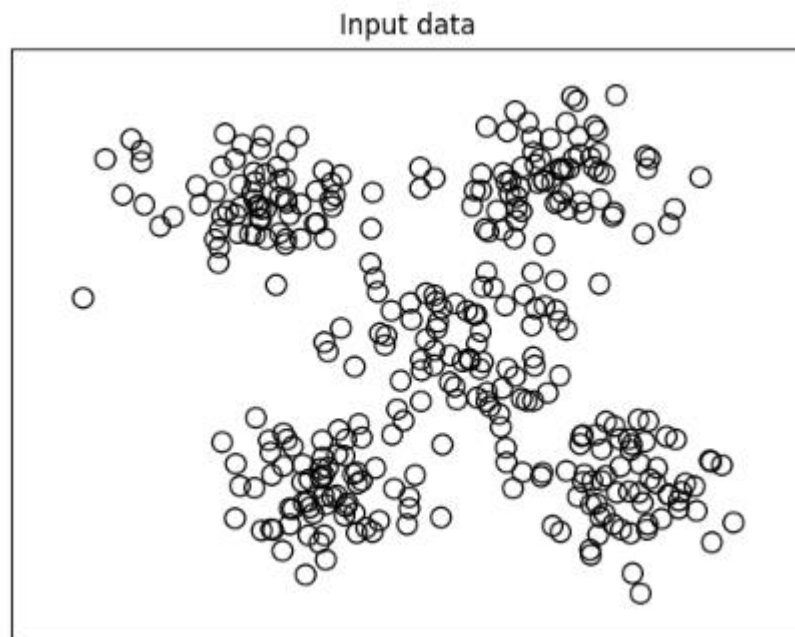
X = np.loadtxt('data_clustering.txt', delimiter=',')

num_clusters = 5

plt.figure()
plt.scatter(X[:, 0], X[:, 1], marker='o', facecolors='none', edgecolors='black', s=80)
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(())
plt.show()
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()
```

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:



Висновок: Ми використали алгоритм K-Means для кластеризації даних, було використано 5 кластерів, на графіку показано розділення простору на кластери та центри кластерів.

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.8. 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. У цьому завданні використовуйте `sklearn.cluster.KMeans` для пошуку кластерів набору даних Iris.

Лістинг програми:

```
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

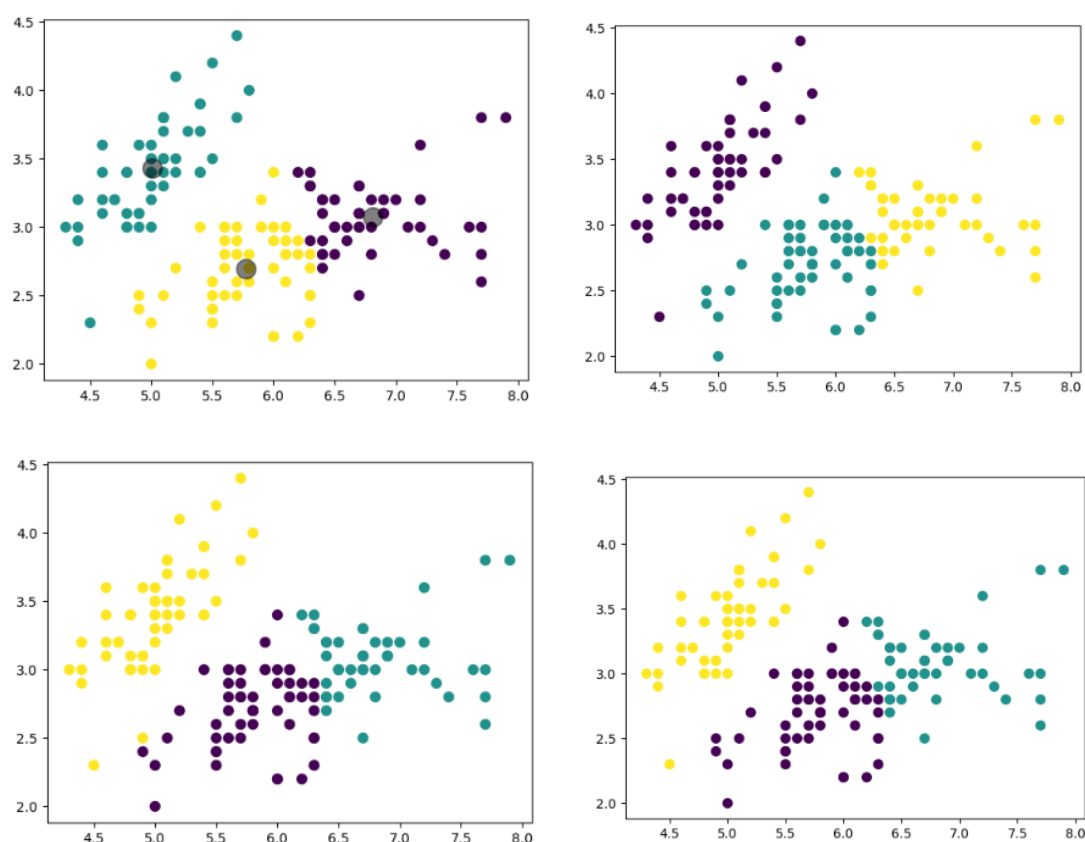
print("using find_clusters():")
centers, labels = find_clusters(X, 3)
print("n_clusters: 3, rseed: 2")
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.show()

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()
```

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:

[illegible]

Висновок: К-Means алгоритм може бути використаний для кластеризації даних, і ви можете вибирати різні параметри та початкові умови для отримання різних результатів кластеризації. Точна кількість кластерів та початкові умови можуть бути важливими для правильного розділення даних на кластери

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.9. 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 середнього. Для аналізу використовуйте дані, які містяться у файлі data_clustering.txt.

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,))

    # 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()
```

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

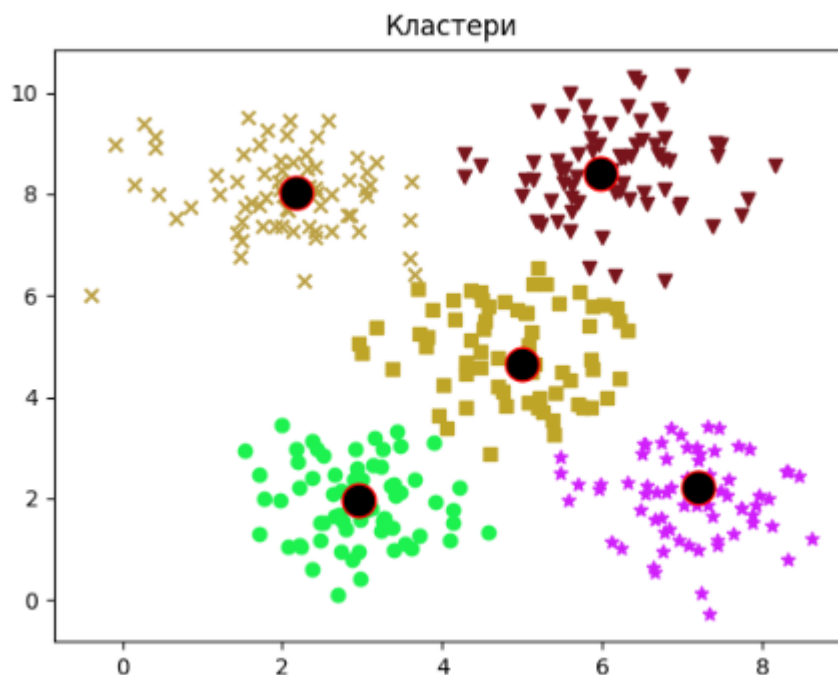
Результат виконання:

```
C:\Users\skoki\PycharmProjects\labs0AI\.venv\Scripts\python.exe C:\Users\skoki\PycharmProjects\labs0AI\lab3\task9.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
```



Висновок: Mean Shift алгоритм ефективно використовується для групування даних. В результаті його застосування було сформовано п'ять кластерів, а координати їх центрів були виведені на екран і зображені на графіку. Кожен кластер був відзначений окремим кольором для полегшення візуалізації та аналізу структури даних.

Висновок до лабораторної роботи: використовуючи спеціалізовані бібліотеки і мову програмування Python дослідив методи регресії та неконтрольованої класифікації даних у машинному навчанні.

		Скоківський В.			ДУ «Житомирська політехніка».22.121.18.000 – Лр3	Арк.
		Голенко М.Ю.				19
Змн.	Арк.	№ докум.	Підпис	Дата		