

Лабораторна Робота №3  
Тема: Дослідження методів регресії

Посилання на гітхаб: <https://github.com/ViMIMercurysMight/python-ai.git>

Завдання 2.1. Створення регресора однієї змінної

*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]

# 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()
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()
print("Продуктивність лінійної регресії:")
print("Середня абсолютна похибка =",
      round(sm.mean_absolute_error(y_test, y_test_pred), 2))
print("Середня квадратична помилка =",
      round(sm.mean_squared_error(y_test, y_test_pred), 2))
print("Середня абсолютна помилка =",
      round(sm.median_absolute_error(y_test, y_test_pred), 2))
print("Пояснена оцінка дисперсії =",
      round(sm.explained_variance_score(y_test, y_test_pred), 2))
print("R2 оцінка =", round(sm.r2_score(y_test, y_test_pred), 2))
```

					<i>Житомирська політехніка</i> 22.121.06.000 – Лр		
Змн.	Арк.	№ докум.	Підпис	Дата	<div>Звіт з лабораторної роботи</div> <div>Літ.    Арк.    Аркушів</div> <div><div></div><div>1</div></div> <div>ФІКТ Гр. ПІ-61</div>		
Розроб.	Медведєв В.В..						
Перевір.	Філіпов В.О						
Керівник							
Н. контр.							
Зав. каф.							

```

output_model_file = 'model.pkl'

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

# Завантаження моделі
y_test_pred_new = regressor.predict(X_test)
print("\nНова середня абсолютна помилка =",
      round(sm.mean_absolute_error(y_test, y_test_pred_new), 2))

```

```

Продуктивність лінійної регресії:
Середня абсолютна похибка = 0.59
Середня квадратична помилка = 0.49
Середня абсолютна помилка = 0.51
Пояснена оцінка дисперсії = 0.86
R2 оцінка = 0.86

Нова середня абсолютна помилка = 0.59

Process finished with exit code 0

```



Рисунок 1. Результат виконання

Ми можемо використовувати цей спосіб для статистичного аналізу, що показує зв'язок між двома змінними. Лінійна регресія може створити модель прогнозування за «випадковими» даними, показуючи тенденцію в них. Наприклад для цін, тривалості життя тощо.

Завдання 2.2. Передбачення за допомогою регресії однієї змінної  
Варіант 1 файл: data\_regr\_1.txt

*Лістинг програми*

```

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

```

		Медведєв В.В			Житомирська політехніка 22.121.06.000 – Лр3	Арк.
		Філіпов В.О				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

print("TEST")
# Вхідний файл, який містить дані
input_file = 'data_regr_1.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()
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()

print("Продуктивність лінійної регресії:")
print("Середня абсолютна похибка =",
      round(sm.mean_absolute_error(y_test, y_test_pred), 2))
print("Середня квадратична помилка =",
      round(sm.mean_squared_error(y_test, y_test_pred), 2))
print("Середня абсолютна помилка =",
      round(sm.median_absolute_error(y_test, y_test_pred), 2))
print("Пояснена оцінка дисперсії =",
      round(sm.explained_variance_score(y_test, y_test_pred), 2))
print("R2 оцінка =", round(sm.r2_score(y_test, y_test_pred), 3))

output_model_file = 'model.pkl'

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

# Завантаження моделі
y_test_pred_new = regressor.predict(X_test)
print("\nНова середня абсолютна помилка =",
      round(sm.mean absolute error(y_test, y_test_pred_new), 2))

```

		Медведєв В.В			Житомирська політехніка 22.121.06.000 – Лр3	Арк.
		. Філіпов В.О				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

TEST
Продуктивність лінійної регресії:
Середня абсолютна похибка = 0.59
Середня квадратична помилка = 0.49
Середня абсолютна помилка = 0.51
Пояснена оцінка дисперсії = 0.86
R2 оцінка = 0.859

Нова середня абсолютна помилка = 0.59

```

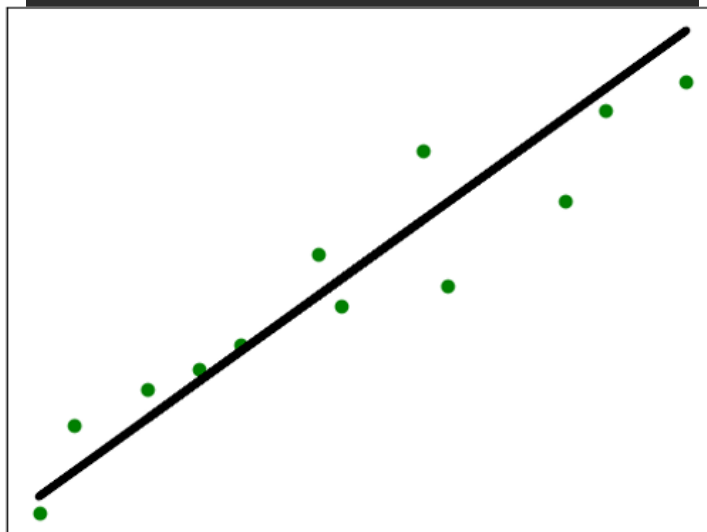


Рисунок 2. Результат виконання

Графік вийшов ідентичний через повну подібність вмісту файлу data\_singlevar\_regr.txt та файлу data\_regr\_1.txt. З графіка видно, що залишки розподілені більш-менш рівномірно відносно осі. Виходячи з  $R^2$  оцінки можна зробити висновок, що продуктивність цієї моделі машинного навчання на основі регресії є середньою

data_regr_1.txt - Notepad	data_singlevar_regr.txt - Notepad
-0.86,4.38	-0.86,4.38
2.58,6.97	2.58,6.97
4.17,7.01	4.17,7.01
2.6,5.44	2.6,5.44
5.13,6.45	5.13,6.45
3.23,5.49	3.23,5.49
-0.26,4.25	-0.26,4.25
2.76,5.94	2.76,5.94
0.47,4.8	0.47,4.8
-3.9,2.7	-3.9,2.7
0.27,3.26	0.27,3.26
2.88,6.48	2.88,6.48
-0.54,4.08	-0.54,4.08
-4.39,0.09	-4.39,0.09
-1.12,2.74	-1.12,2.74
2.09,5.8	2.09,5.8
-5.78,0.16	-5.78,0.16
1.77,4.97	1.77,4.97
-7.91,-2.26	-7.91,-2.26
4.86,5.75	4.86,5.75
-2.17,3.33	-2.17,3.33
1.38,5.26	1.38,5.26
0.54,4.43	0.54,4.43
3.12,6.6	3.12,6.6
-2.19,3.77	-2.19,3.77
-0.33,2.4	-0.33,2.4
-1.21,2.98	-1.21,2.98
-4.52,0.29	-4.52,0.29
-0.46,2.47	-0.46,2.47
-1.13,4.08	-1.13,4.08
4.61,8.97	4.61,8.97
0.31,3.94	0.31,3.94
0.25,3.46	0.25,3.46
-2.67,2.46	-2.67,2.46
-4.66,1.14	-4.66,1.14
-0.2,4.31	-0.2,4.31
-0.52,1.97	-0.52,1.97
1.24,4.83	1.24,4.83
-2.53,3.12	-2.53,3.12
-0.34,4.97	-0.34,4.97
5.74,8.65	5.74,8.65
-0.34,3.59	-0.34,3.59
0.99,3.66	0.99,3.66
5.01,7.54	5.01,7.54
-2.38,1.52	-2.38,1.52

Рисунок 3. Демонстрація подібності файлів

		Медведєв В.В			Житомирська політехніка	Арк.
		Філіпов В.О			22.121.06.000 – Лр3	
Змн.	Арк.	№ докум.	Підпис	Дата		4

### Завдання 3. Створення багатовимірного регресора

Лістинг програми

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

# 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
regressor = linear_model.LinearRegression()
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()

print("Продуктивність лінійної регресії:")
print("Середня абсолютна похибка =",
      round(sm.mean_absolute_error(y_test, y_test_pred), 2))
print("Середня квадратична помилка =",
      round(sm.mean_squared_error(y_test, y_test_pred), 2))
print("Середня абсолютна помилка =",
      round(sm.median_absolute_error(y_test, y_test_pred), 2))
print("Пояснена оцінка дисперсії =",
      round(sm.explained_variance_score(y_test, y_test_pred), 2))
print("R2 оцінка =", round(sm.r2_score(y_test, y_test_pred), 3))

output_model_file = 'model.pkl'

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

# Завантаження моделі
y_test_pred_new = regressor.predict(X_test)
print("\nНова середня абсолютна помилка =",
      round(sm.mean_absolute_error(y_test, y_test_pred_new), 2))

# Поліноміальна регресія
```

		Медведєв В.В			Житомирська політехніка 22.121.06.000 – Лр3	Арк.
		Філіпов В.О				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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

```

```

C:\Users\ViMi\PycharmProjects\lab_3\venv\Scripts\python.exe C:/Users/ViMi
Продуктивність лінійної регресії:
Середня абсолютна похибка = 3.58
Середня квадратична помилка = 20.31
Середня абсолютна помилка = 2.99
Пояснена оцінка дисперсії = 0.86
R2 оцінка = 0.865

Нова середня абсолютна помилка = 3.58

Linear regression:
[36.05286276]

Polynomial regression:
[41.46197042]

Process finished with exit code 0

```

Рисунок 5. Результат виконання

#### Завдання 4 Регресія багатьох змінних

Лістинг додатку

```

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

# Завантажимо набір даних
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)

# Використаємо лише одну "розмірність"
diabetes_X = diabetes_X[:, np.newaxis, 2]

# Розіб'ємо дані на тестові та тренувальні
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]

diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]

# Створимо модель лінійної регресії

```

		Медведєв В.В			Житомирська політехніка 22.121.06.000 – Лр3	Арк.
		Філіпов В.О				
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

regr = linear_model.LinearRegression()

# Навчимо
regr.fit(diabetes_X_train, diabetes_y_train)

# Зробимо передбачення
diabetes_y_pred = regr.predict(diabetes_X_test)

# Коефіцієнти
print("Regression coef: \n", regr.coef_)
print("Regression intercept: \n", regr.intercept_)
# Середня абсолютна похибка
print("Mean absolute error :",
      round(mean_absolute_error(diabetes_y_test, diabetes_y_pred), 2))
# Помилка середньої похибки
print("Mean squared error: %.2f" % mean_squared_error(diabetes_y_test, diabetes_y_pred))
print("R2 score: %.2f" % r2_score(diabetes_y_test, diabetes_y_pred))

fig, ax = plt.subplots()
ax.scatter(diabetes_y_test, diabetes_y_pred, edgecolors=(0, 0, 0))
ax.plot([diabetes_y.min(), diabetes_y.max()], [diabetes_y.min(), diabetes_y.max()], 'k--', lw=4)
ax.set_xlabel('Виміряно')
ax.set_ylabel('Передбачено')
plt.show()

```

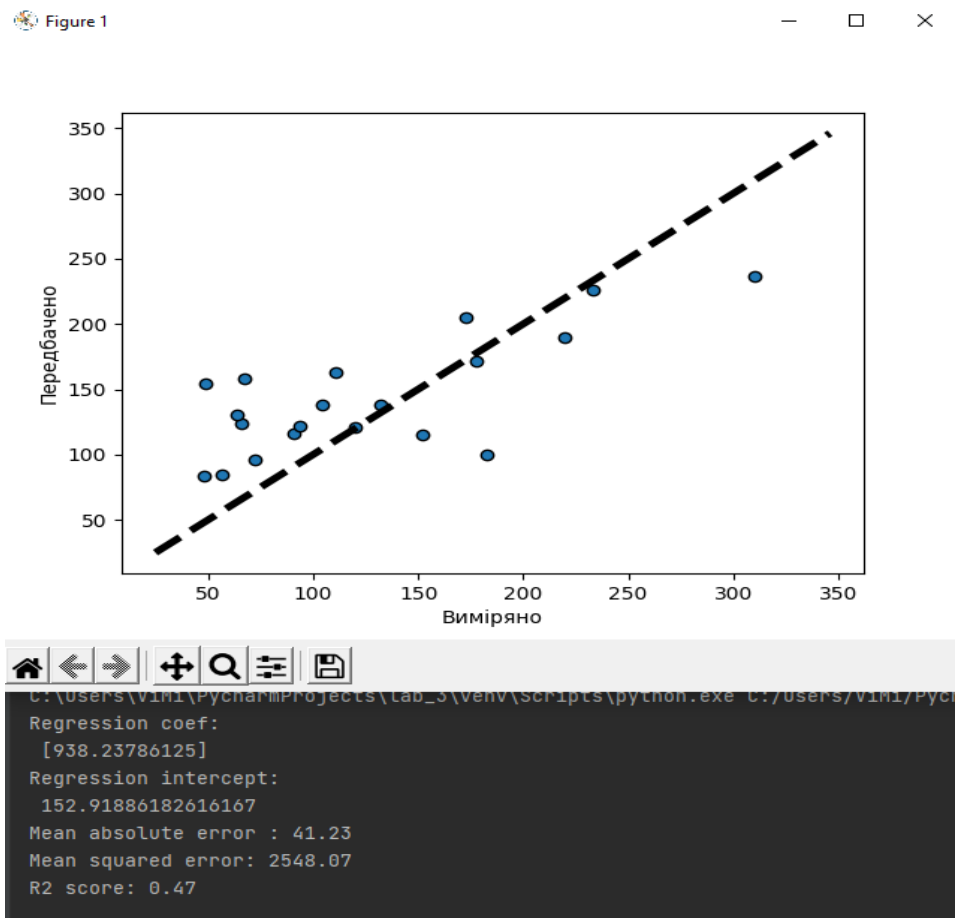


Рисунок 6. Результат виконання

		Медведєв В.В			Житомирська політехніка 22.121.06.000 – Лр3	Арк.
		Філіпов В.О				7
Змн.	Арк.	№ докум.	Підпис	Дата		

## Завдання 5. Самостійна побудова регресії

### Варіант 1

```
m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.5 * X ** 2 + X + 2 + np.random.randn(m, 1)
```

*Лістинг програми*

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

m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.5 * X ** 2 + X + 2 + np.random.randn(m, 1)

X = X.reshape(-1, 1)
Y = y.reshape(-1, 1)

lin = LinearRegression()
lin.fit(X, y)

poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)

Y_NEW = lin2.predict(X_poly)
r2 = r2_score(Y, Y_NEW)

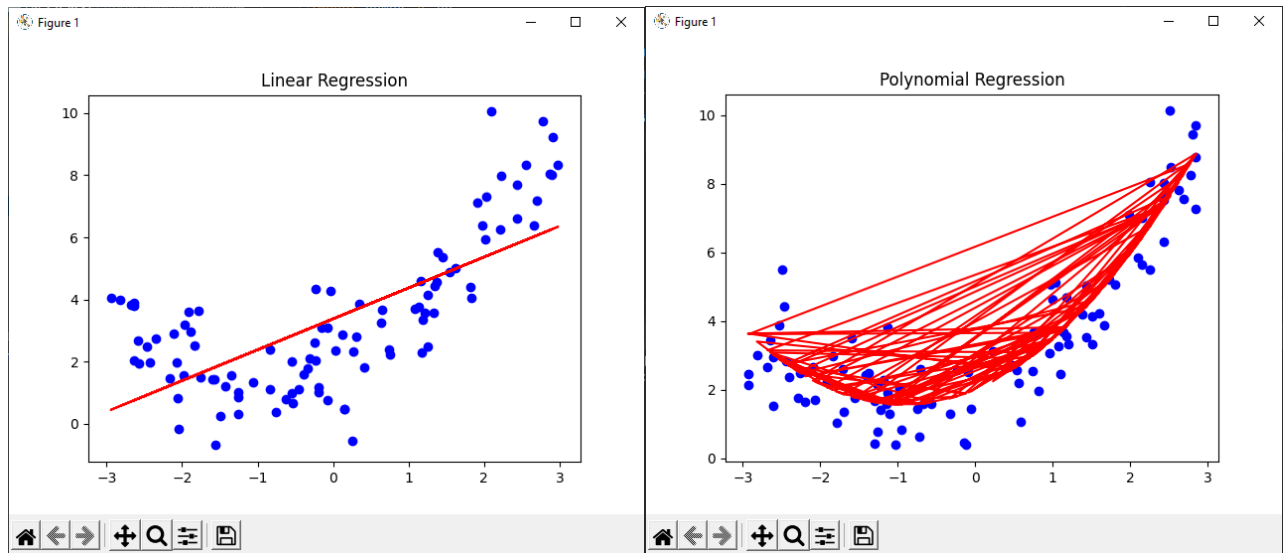
print('R2: ', r2)

# Visualising the Linear Regression results
plt.scatter(X, y, color='blue')
plt.plot(X, lin.predict(X), color='red')
plt.title('Linear Regression')
plt.show()

# Visualising the Polynomial Regression results
plt.scatter(X, y, color='blue')
plt.plot(X, lin2.predict(poly.fit_transform(X)), color='red')
plt.title('Polynomial Regression')
plt.show()
```

		Медведєв В.В			Житомирська політехніка 22.121.06.000 – Лр3	Арк.
		. Філіпов В.О				8
Змн.	Арк.	№ докум.	Підпис	Дата		





C:\Users\ViMi\PycharmPro  
R2: 0.8700557921966817

Рисунок 7. Графіки Лінійної та Полімінальної регресії та  $R^2$  оцінка

Побудована модель є середньою через значне відхилення від осей.

## Завдання 2.6. Побудова кривих навчання

*Лістинг програми:*

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
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='train')
    plt.plot(np.sqrt(val_errors), "b-", linewidth=3, label='val')
    plt.show()

m = 100
X = np.linspace(-3, 3, m)
y = 3 + np.sin(X) + np.random.uniform(-0.5, 0.5, m)

X = X.reshape(-1, 1)
Y = y.reshape(-1, 1)

lin = LinearRegression()
lin.fit(X, y)
```

		Медведєв В.В			Житомирська політехніка 22.121.06.000 – Лр3	Арк.
		Філіпов В.О				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

poly = PolynomialFeatures(degree=2, include_bias=False)
X_poly = poly.fit_transform(X)

poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)

Y_NEW = lin2.predict(X_poly)
r2 = r2_score(Y, Y_NEW)

polynomial_regg = Pipeline([
    # ("poly_features", PolynomialFeatures(degree=2, include_bias=False)),
    ("poly_features", PolynomialFeatures(degree=10, include_bias=False)),
    ("lin_reg", LinearRegression()),
])

plot_learning_curves(polynomial_regg, X, y)

```

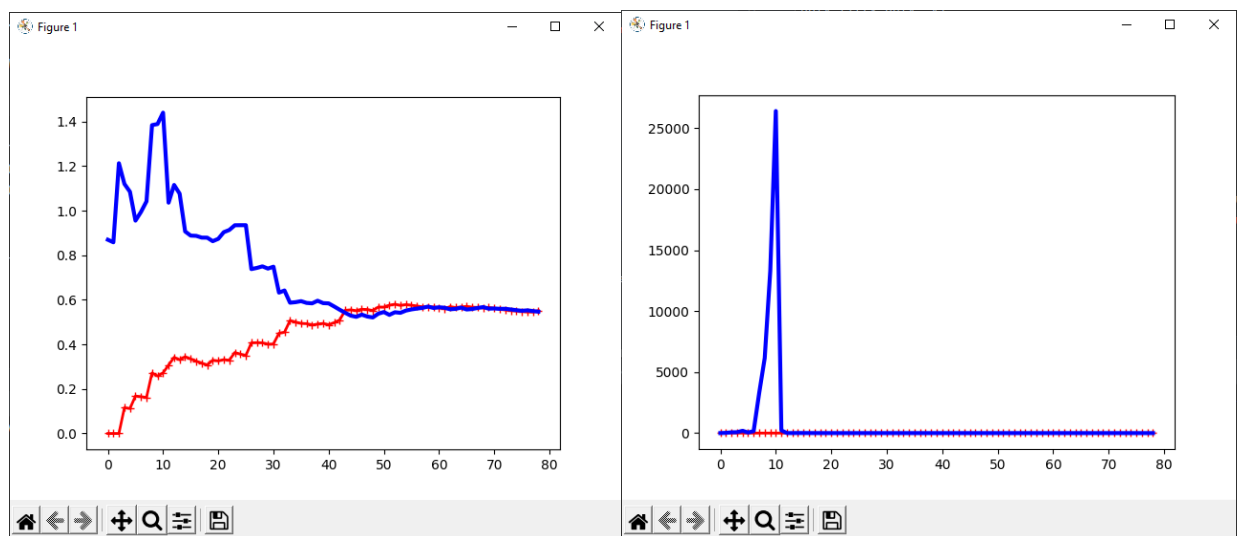


Рисунок 8. Крива навчання для моделі за ступенем 2(зліва) та 10(справа)

**Висновок:** Розглянули особливості методів регресії та побудови графіків на основі отриманих даних.

		Медведєв В.В			Житомирська політехніка 22.121.06.000 – Лр3	Арк.
		Філіпов В.О				10
Змн.	Арк.	№ докум.	Підпис	Дата		