

ЛАБОРАТОРНА РОБОТА № 3

ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ

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

Хід роботи:

Завдання 1: Створення регресора однієї змінної

Результат:

Код:

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

input_file = 'data_singlevar_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:]

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))
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Фещенко Д.М.			Звіт з лабораторної роботи 3				Літ.	Арк.	Аркушів	
Перевір.		Пулюко І.В.									1	
Керівник									ФІКТ Гр. ПІ-59(2)			
Н. контр.												
Затверд.												

```
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))
```

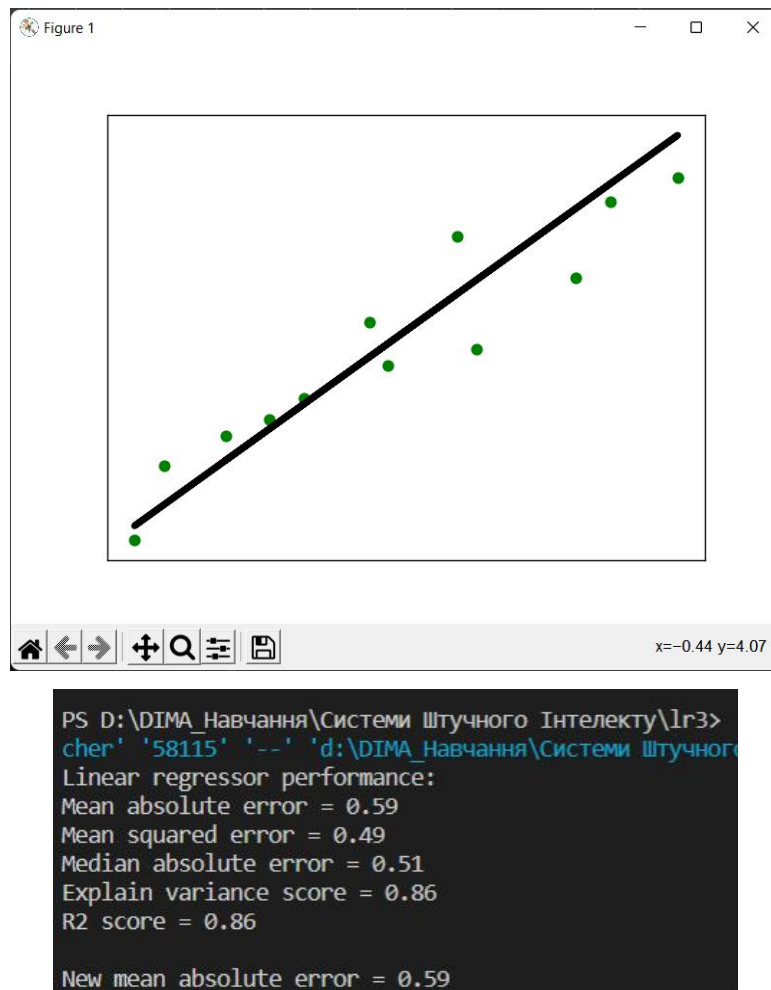


Рис. 1.1. Результат виконання

Модель добре справляється з поставленим завданням.

Хід роботи:

Завдання 2: Передбачення за допомогою регресії однієї змінної

23
3

Варіант 3 файл: data_regr_3.txt

Результат:

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

Код:

```
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 = '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))
```

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
		.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

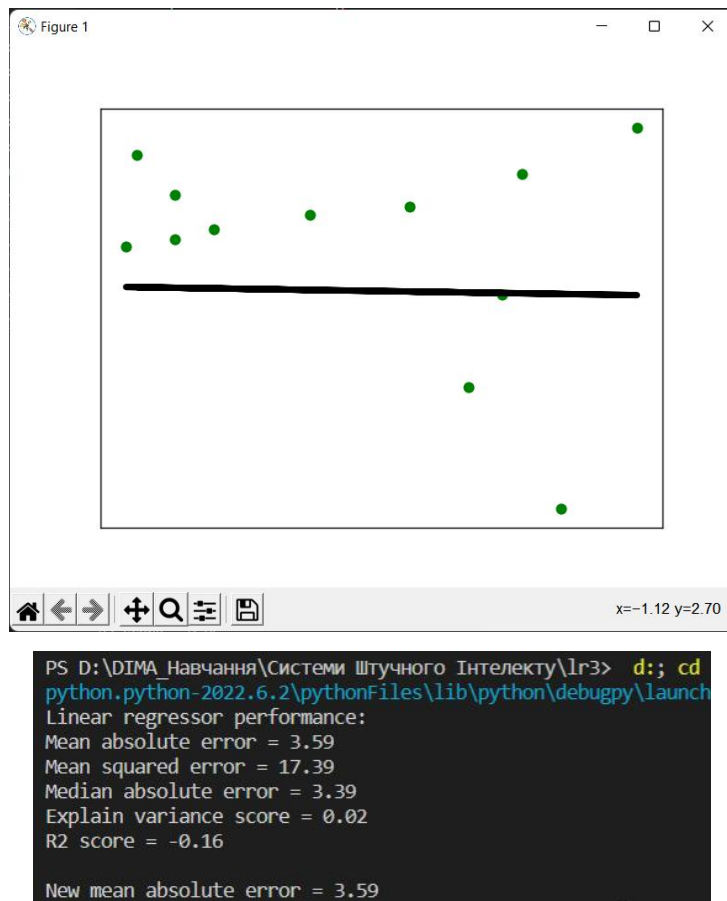


Рис. 2.1. Результат виконання

Цього разу в датасеті є аномальні дані MSE vs MAE, до яких модель не пристосована.

Хід роботи:

Завдання 3: Створення багатовимірної регресора

Результат:

Код:

```
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)
```

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
		.				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))

```

```

PS D:\DIMA_Навчання\Системи Штучного Інтелекту\lr3> d:; cd
python.python-2022.6.2\pythonFiles\lib\python\debugpy\launch
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:
[41.46504705]

```

Рис. 3.1. Виведення характеристик та порівняння моделей

Поліноміальний регресор справляється краще за лінійний при регресії з декількома характеристиками.

Хід роботи:

Завдання 4: Регресія багатьох змінних

Результат:

Код:

```

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

```

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
		.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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()

```

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
		.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

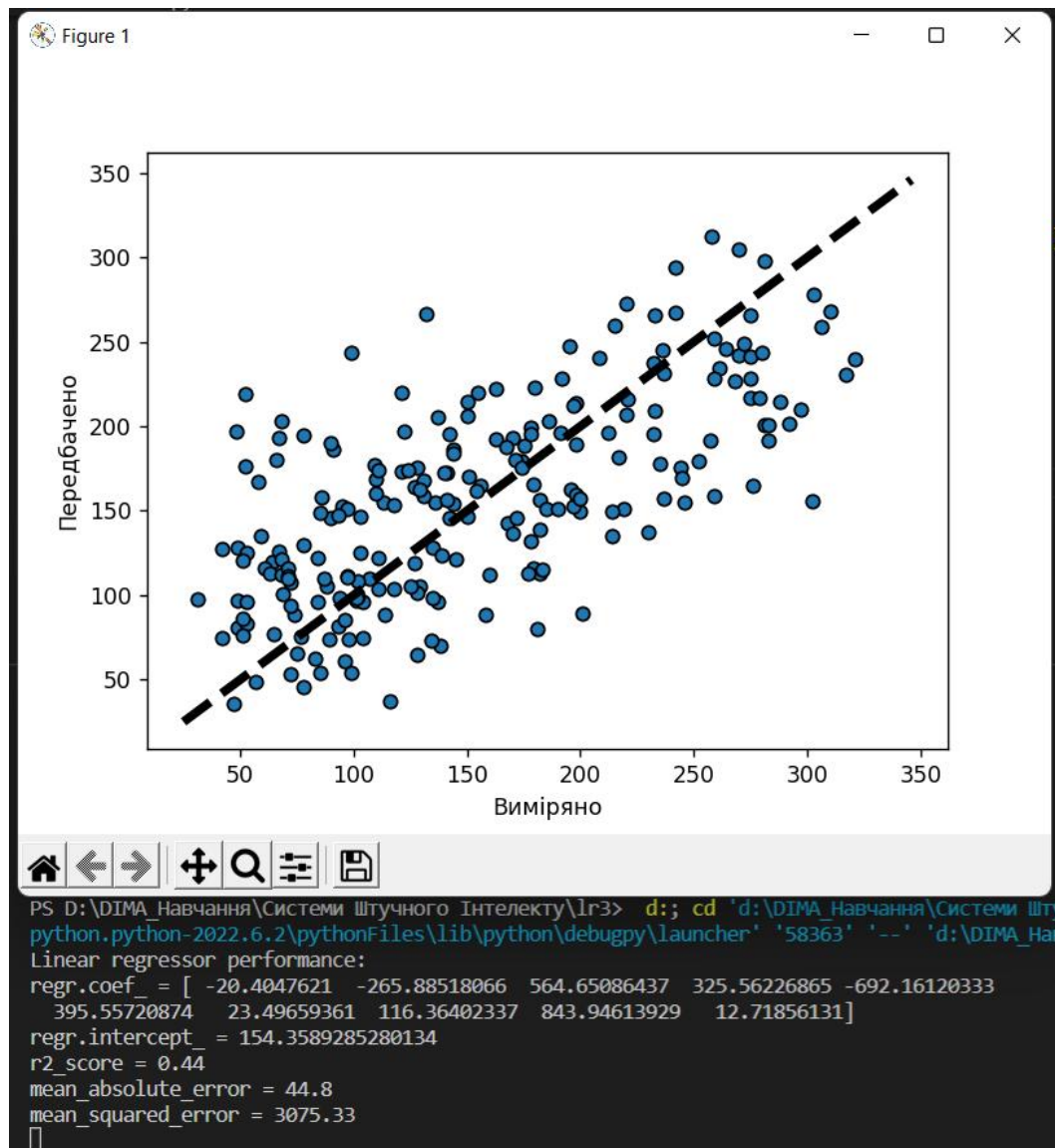


Рис. 4.1. Результат

Похибка є великою, але обрана регресія працює краще ніж звичайна регресія з використанням середніх значень.

Хід роботи:

Завдання 5: Самостійна побудова регресії

№ за списком	21	22	23	24	25	26	27	28	29	30
№ варіанту	1	2	3	4	5	6	7	8	9	10

Варіант 3

```

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

```

Результат:

Код:

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

```
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
```

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

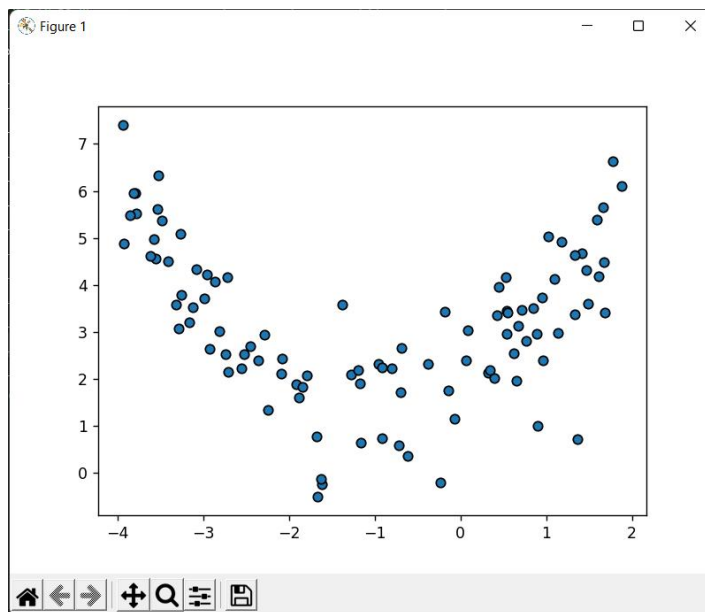
```
fig, ax = plt.subplots()
ax.scatter(X, y, edgecolors = (0, 0, 0))
plt.show()
```

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

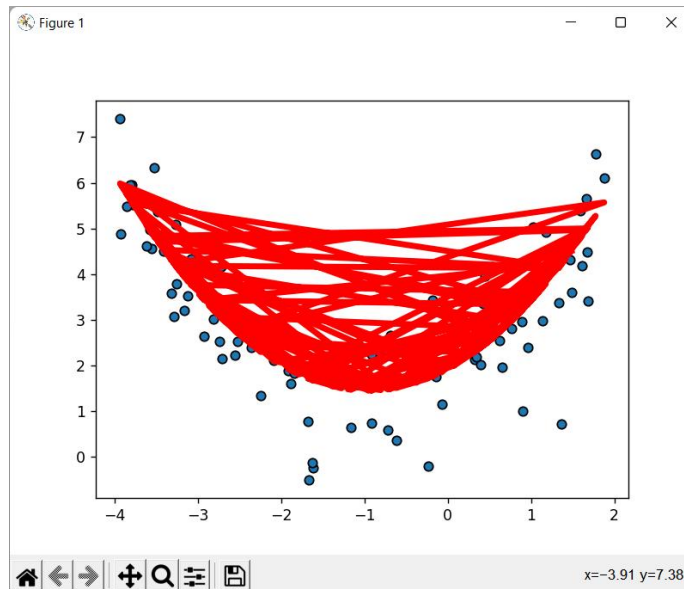
```
poly_features = PolynomialFeatures(degree=2, include_bias=False)
X_poly = poly_features.fit_transform(np.array(X).reshape(-1, 1))
```

```
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.show()
```



		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
		.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```
> d:; cd 'd:\DIMA_Навчан
python.python-2022.6.2\pythonFiles\lib\python\debugpy\launcher' '58475' '--
[1.33322787] [3.3671564]
[1.92216926] [[0.98179797 0.5118774 ]]
PS D:\DIMA_Навчання\Системи Штучного Інтелекту\lr3> []
```

Рис. 5.1. Результат

Модель рівняння:

$$y = 0.5x^2 + 1x + 2 + \text{гаусовий шум}$$

Отримана модель регресії з передбаченими коефіцієнтами:

$$y = 0.53x^2 + 1.1x + 2$$

Хід роботи:

Завдання 6: Побудова кривих навчання

Результат:

Код:

```
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) - 4
y = 0.5 * X ** 2 + X + 2 + 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)):
```

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
		.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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=2, include_bias=False)),
    ('lin_reg', linear_model.LinearRegression()),
])

```

```

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

```

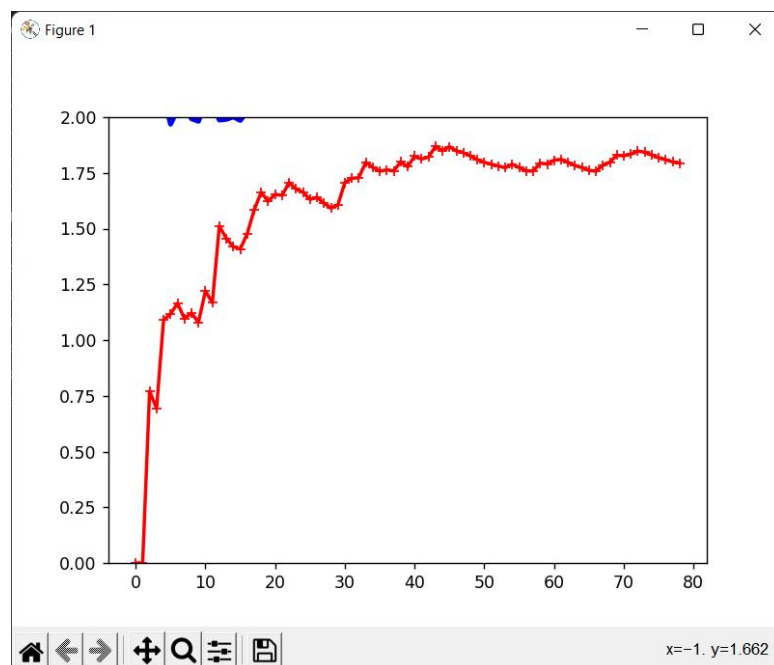


Рис. 6.1. Криві навчання для лінійної моделі

		Фещенко Д.М.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА 22.121.23.000-ЛР03	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

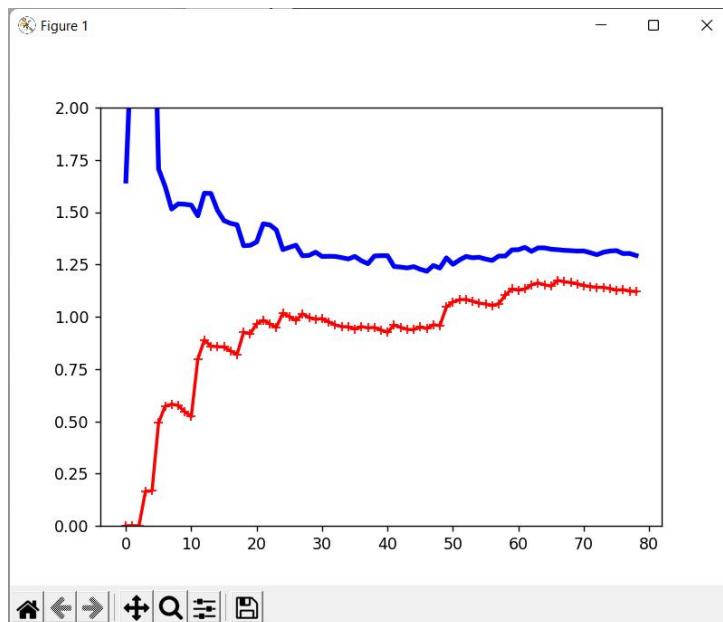


Рис. 6.2. Криві навчання для поліноміальної моделі

Висновок: на цій лабораторній роботі ми використовували спеціалізовані бібліотеки та мову програмування Python дослідили методи регресії даних у машинному навчанні.