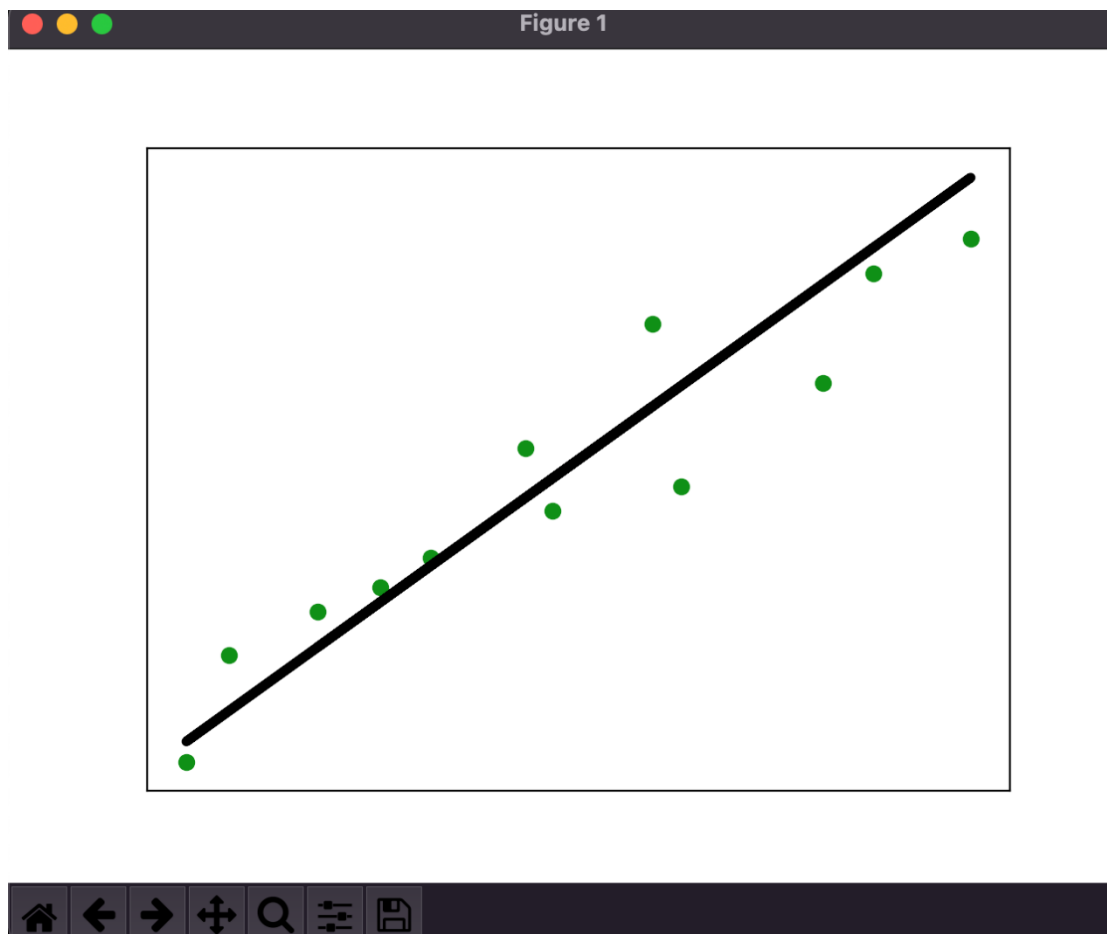


ЛАБОРАТОРНА РОБОТА № 3

ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ

Завдання 2.1. Створення регресора однієї змінної



					ДУ «Житомирська політехніка».20.121.08.806 – ІПЗк		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Колесник А. С.			Звіт з лабораторної роботи	Літ.	Арк.
Перевір.							1
Реценз.						ФІКТ Гр. ІПЗк-19-1	
Н. Контр.							
Затверд.							

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

```
import pickle
```

```
import numpy as np
```

```
from sklearn import linear_model
```

```
import sklearn.metrics as sm
```

```
import matplotlib.pyplot as plt
```

```
# Input file containing data
```

```
input_file = 'data_singlevar_regr.txt'
```

```
# Read data
```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

data = np.loadtxt(input_file, delimiter=',')

X, y = data[:, :-1], data[:, -1]


# Train and test split

num_training = int(0.8 * len(X))

num_test = len(X) - num_training


# Training data

X_train, y_train = X[:num_training], y[:num_training]


# Test data

X_test, y_test = X[num_training:], y[num_training:]


# Create linear regressor object

regressor = linear_model.LinearRegression()


# Train the model using the training sets

regressor.fit(X_train, y_train)

```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

# Predict the output

y_test_pred = regressor.predict(X_test)


# Plot outputs

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()


# Compute performance metrics

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))

```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

# Model persistence

output_model_file = 'model.pkl'


# Save the model

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

    pickle.dump(regressor, f)


# Load the model

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

    regressor_model = pickle.load(f)


# Perform prediction on test data

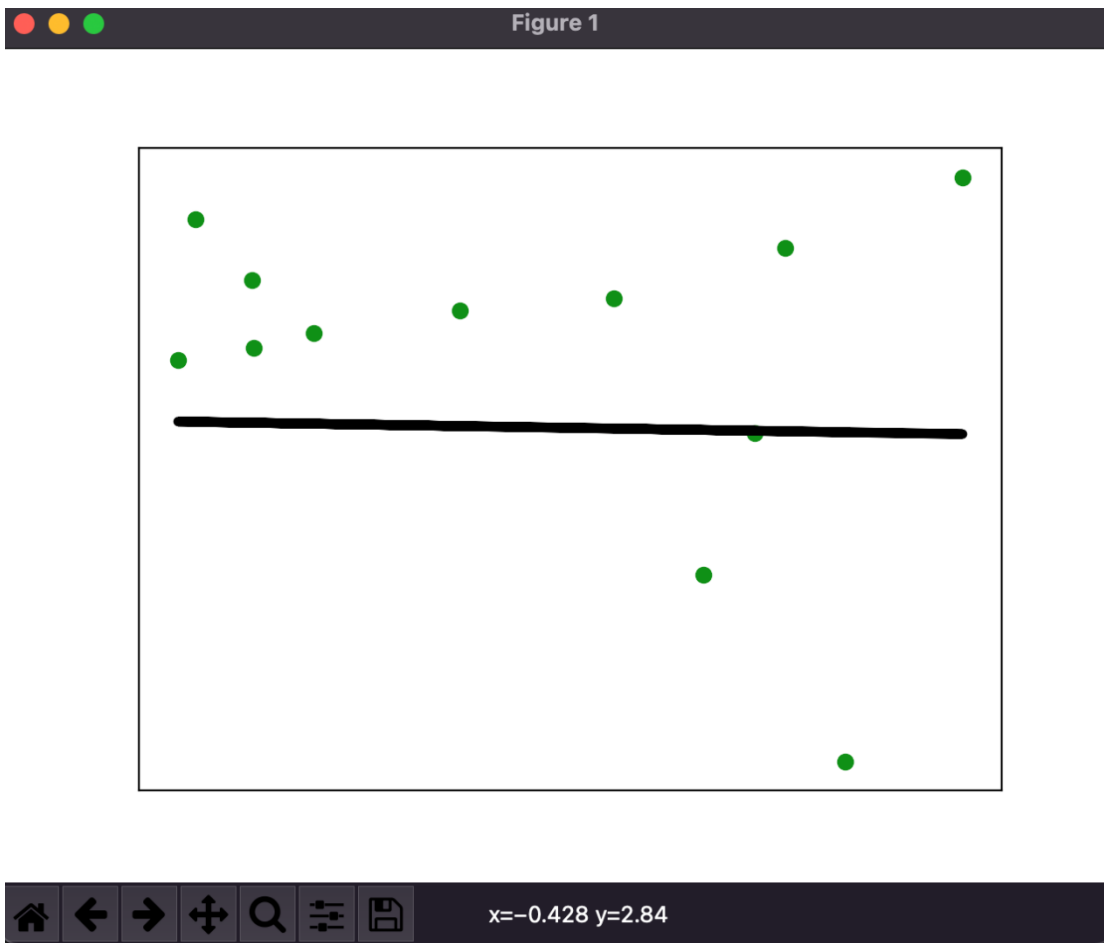
y_test_pred_new = regressor_model.predict(X_test)

print("\nNew mean absolute error =", round(sm.mean_absolute_error(y_test,
y_test_pred_new), 2))

```

Завдання 2.2. Передбачення за допомогою регресії однієї змінної

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				5
Змн.	Арк.	№ докум.	Підпис	Дата		



```
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
```

```
import pickle

import numpy as np

from sklearn import linear_model
```

```

import sklearn.metrics as sm

import matplotlib.pyplot as plt


# Input file containing data

input_file = 'data_regr_3.txt'


# Read data

data = np.loadtxt(input_file, delimiter=',')

X, y = data[:, :-1], data[:, -1]


# Train and test split

num_training = int(0.8 * len(X))

num_test = len(X) - num_training


# Training data

X_train, y_train = X[:num_training], y[:num_training]


# Test data

```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

X_test, y_test = X[num_training:], y[num_training:]

# Create linear regressor object

regressor = linear_model.LinearRegression()

# Train the model using the training sets

regressor.fit(X_train, y_train)

# Predict the output

y_test_pred = regressor.predict(X_test)

# Plot outputs

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()

```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

# Compute performance metrics

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))


# Model persistence

output_model_file = 'model.pkl'


# Save the model

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

    pickle.dump(regressor, f)


# Load the model

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

    regressor_model = pickle.load(f)

```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```
# Perform prediction on test data
```

```
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))
```

Завдання 2.3. Створення багатовимірного регресора

```
Linear Regressor performance:  
Mean absolute error = 3.58  
Mean squared error = 20.31  
Median absolute error = 2.99  
Explained variance score = 0.86  
R2 score = 0.86
```

```
Linear regression:  
[36.05286276]
```

```
Polynomial regression:  
[41.46072631]
```

```
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'
```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```
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()
```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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("Explained 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]]

```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

poly_datapoint = polynomial.fit_transform(datapoint)

poly_linear_model = linear_model.LinearRegression()

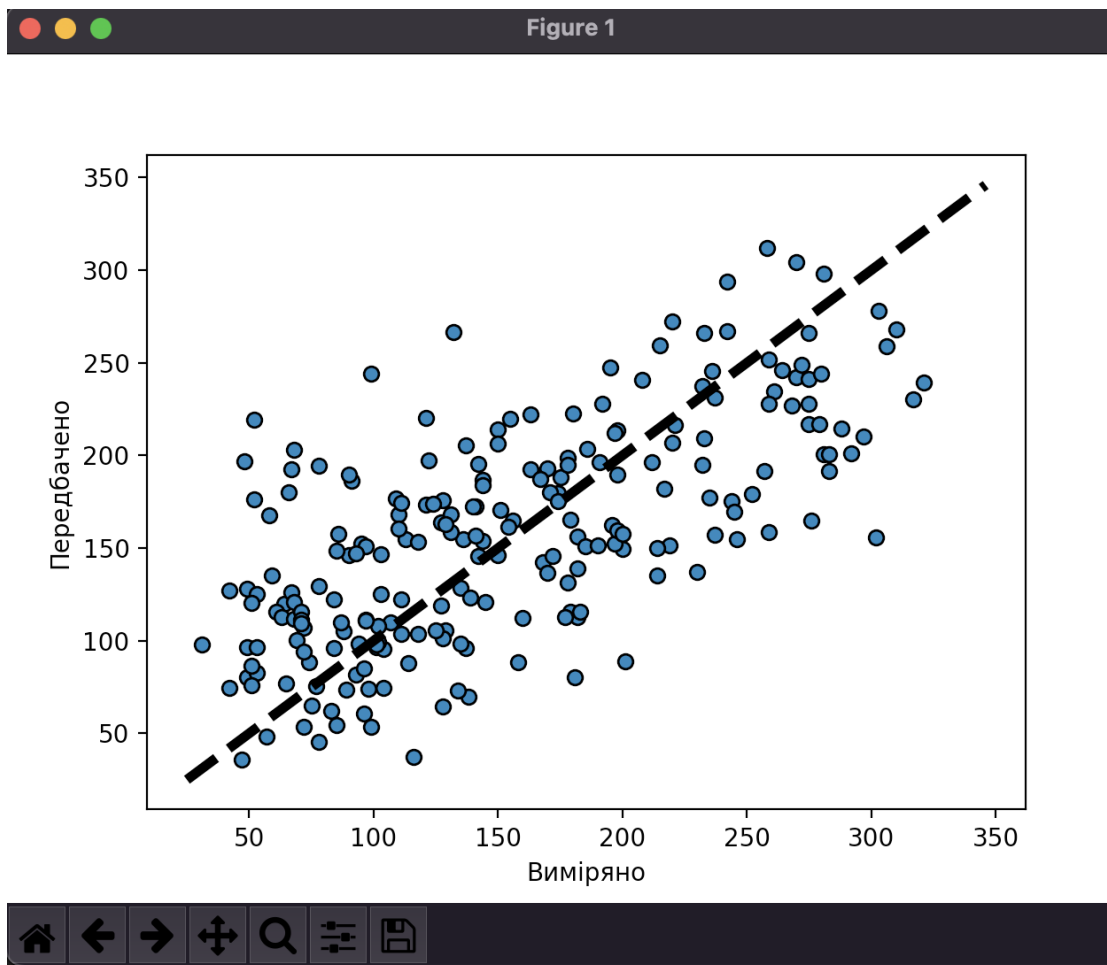
poly_linear_model.fit(X_train_transformed, y_train)

print("\nLinear regression:\n", linear_regressor.predict(datapoint))

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

```

Завдання 2.4. Регресія багатьох змінних



[-20.41129305 -265.88594023 564.64844662 325.55650029 -692.23796104

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

395.62249978 23.52910434 116.37102129 843.98257585 12.71981044]
154.3589882135515 <function r2_score at 0x12726b310> <function
mean_absolute_error at 0x1272605e0> <function mean_squared_error at
0x127260f70>

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets, linear_model

from sklearn.metrics import mean_squared_error, r2_score

from sklearn.metrics import mean_absolute_error

from sklearn.model_selection import train_test_split

diabetes = datasets.load_diabetes()

X = diabetes.data

y = diabetes.target

Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size

= 0.5, random_state = 0)

regr = linear_model.LinearRegression()
```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

regr.fit(Xtrain, ytrain)

ypred = regr.predict(Xtest)

print(f'{regr.coef_} {regr.intercept_} {r2_score} {mean_absolute_error} {mean_squared_error}')

fig, ax = plt.subplots()

ax.scatter(ytest, ypred, edgecolors = (0, 0, 0))

ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw = 4)

ax.set_xlabel('В и м і р я н о')

ax.set_ylabel('П е р е д б а ч е н о')

plt.show()

```

Завдання 2.5. Самостійна побудова регресії

```

import numpy as np

from sklearn.linear_model import LinearRegression

m = 100

X = np.linspace(-3, 3, m)

y = 2 * np.sin(X) + np.random.uniform(-0.5, 0.5, m)

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

```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

```
print(reg.score(X, y))
```

```
print(reg.coef_)
```

```
print(reg.intercept_)
```

```
print(reg.predict)
```

```
0.005510732299240484
```

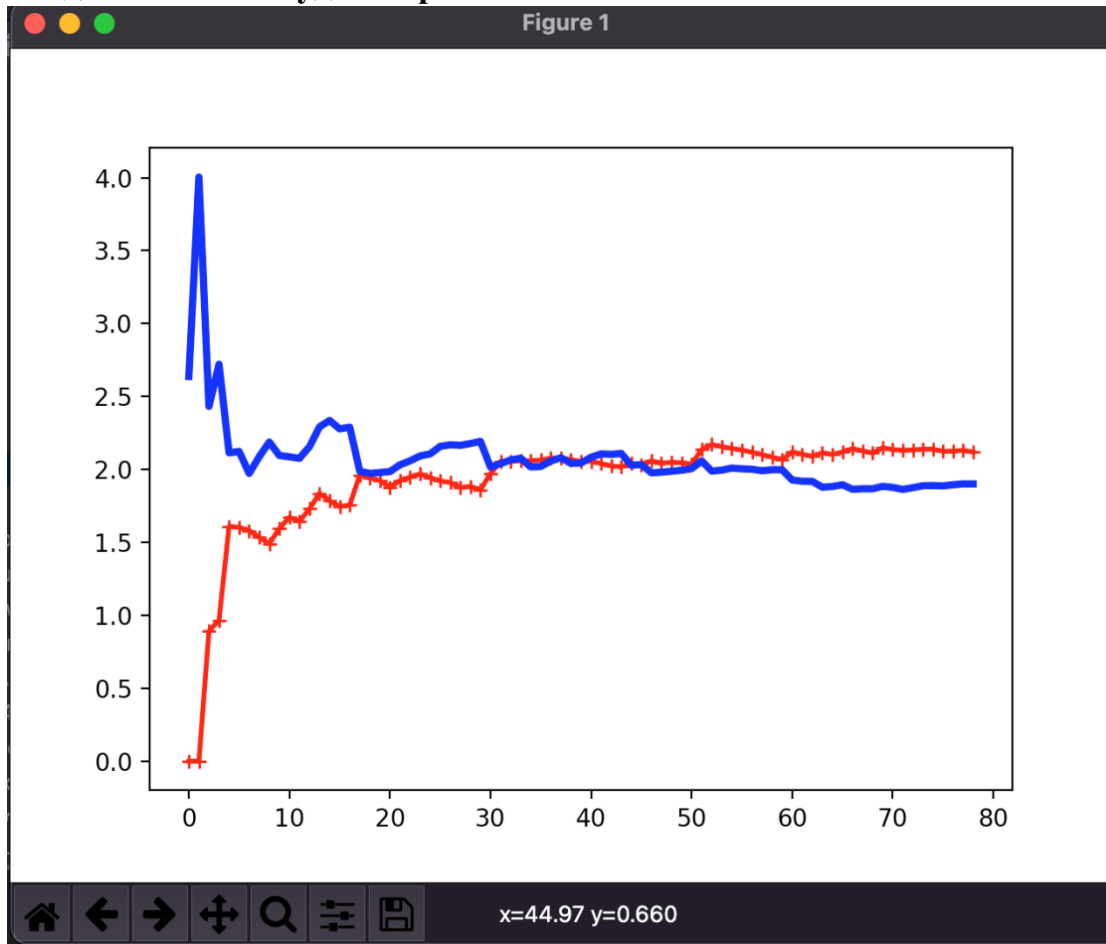
```
[[0.07205751]]
```

```
[3.34274893]
```

```
<bound method LinearModel.predict of LinearRegression(>
```

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
		.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.6. Побудова кривих навчання



```
from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error

from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt

import numpy as np

from sklearn.preprocessing import PolynomialFeatures
```

```

m = 100

X = np.linspace(-3, 3, m)

y = 2 * np.sin(X) + np.random.uniform(-0.5, 0.5, m)

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()

lin_reg=LinearRegression()

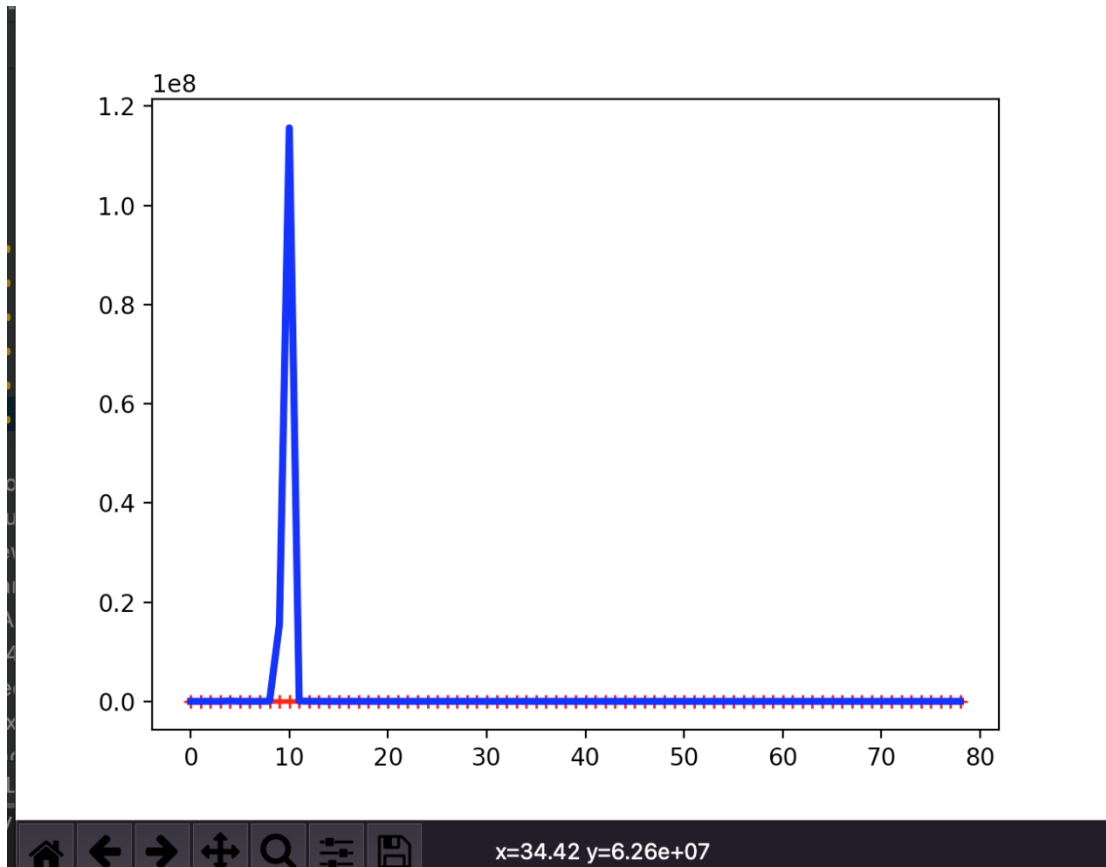
plot_learning_curves(lin_reg,X,y)

```

```
from sklearn.pipeline import Pipeline
```

```
polynomial_regression = Pipeline([("poly features", PolynomialFeatures(degree=10,  
include_bias=False)),("lin_reg", LinearRegression()),])
```

```
plot_learning_curves(polynomial_regression,X,y)
```



Висновок: Отже я, використовуючи спеціалізовані бібліотеки та мову програмування Python, дослідив методи регресії даних у машинному навчанні.

		Колесник А. С.			ДУ «Житомирська політехніка».20.121.00.000 – ІПЗк	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		