

ЛАБОРАТОРНА РОБОТА № 4

ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ

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

Завдання 2.1. Створення регресора однієї змінної

```
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()
```

Рис 2.1 – лістинг програми

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4		
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №4		
Розроб.		Прокопчук О.С					
Перевір.		Голенко М.Ю.					
Реценз.							
Н. Контр.							
Зав.каф.					ФІКТ, гр. ІПЗ-21-1(2)		
					Літ.	Арк.	Аркуші
						1	14

```

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)

# Завантаження моделі
y_test_pred_new = regressor.predict(X_test)
print("\nNew mean absolute error =", round(sm.mean_absolute_error(y_test, y_test_pred_new), 2))

```

Рис 2.2 – лістинг програми

```

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

```

Рис 2.3 – результат виконання

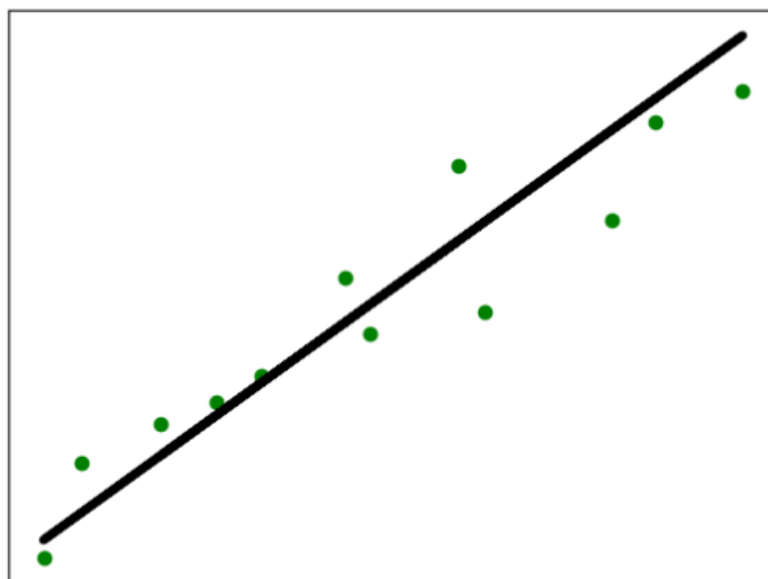


Рис 2.4 – результат виконання

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

Графік зображує незалежні змінні x та залежну змінну y . Лінійна модель регресії показує нормальні показники, однак є відхилення від норми.

Завдання 2.2. Передбачення за допомогою регресії однієї змінної

№ за списком	15
№ варіанту	5

Варіант 5 файл: data_regr_5.txt

```
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_regr_5.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()
```

Рис 2.5 – лістинг програми

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

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'

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

```

Рис 2.6 – лістинг програми

```

Linear regressor performance:
Mean absolute error = 3.31
Mean squared error = 16.98
Median absolute error = 2.66
Explain variance score = -0.14
R2 score = -0.15

New mean absolute error = 3.31

```

Рис 2.7 – результат виконання

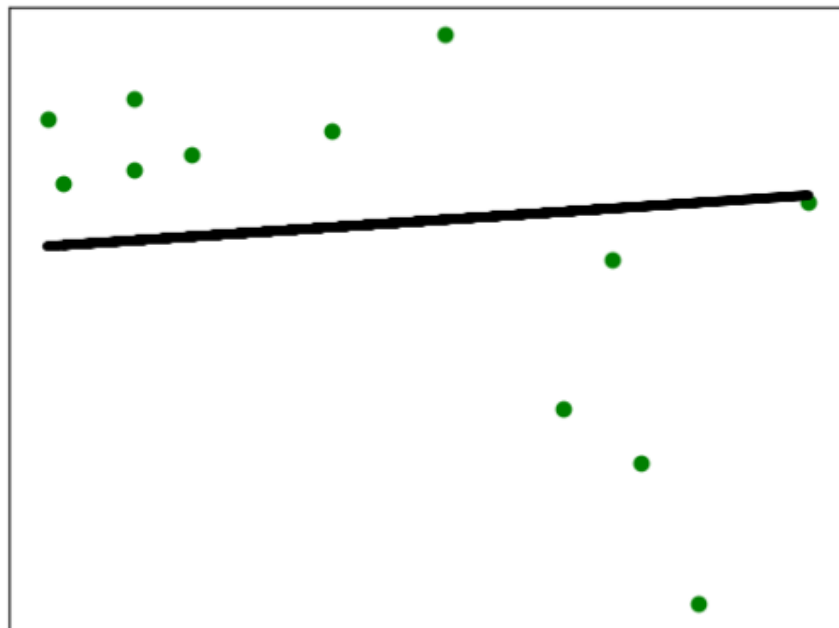


Рис 2.8 – результат виконання

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

Графік демонструє розподіл тестових даних та лінію регресії, яка добре описує залежність між незалежною змінною та результатами. Лінійна регресія в цьому випадку показує низьку точність: середнє абсолютне відхилення (3,31) і середнє квадратичне значення (16,98) свідчать про значні помилки в прогнозах.

Негативний коефіцієнт детермінації ($R^2 = -0.15$) вказує на те, що модель не здатна адекватно пояснити варіацію в даних. Це свідчить про те, що лінійна регресія не є підходящою для даного набору даних, і для кращих результатів необхідно застосувати більш складну модель, наприклад, поліноміальну регресію.

Завдання 2.3. Створення багатовимірного регресора

```
import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
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:]

# Поліноміальна регресія
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)
linear_regressor = linear_model.LinearRegression()
linear_regressor.fit(X_train, y_train)
print("\nLinear regression:\n", linear_regressor.predict(datapoint))
print("\nPolynomial regression:\n",
      poly_linear_model.predict(poly_datapoint))
```

Рис 2.9 – лістинг програми

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

```

# Створення об'єкта лінійного регресора
regressor = linear_model.LinearRegression()
regressor.fit(X_train, y_train)

# Прогнозування результату
y_test_pred = 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))

```

Рис 2.10 – лістинг програми

```

Linear regression:
[36.05286276]

Polynomial regression:
[41.46177229]

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

```

Рис 2.11 – результат виконання

У цьому прикладі поліноміальна регресія дає ближчий до реальності результат ніж лінійна регресія (41.46 - 41.35), що показує її вищу ефективність в складних залежностях.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.4. Регресія багатьох змінних

```
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()
regr.fit(Xtrain, ytrain)
ypred = regr.predict(Xtest)

print('Coefficients: \n', regr.coef_)
print('Intercept: \n', regr.intercept_)
print('Coefficient of determination: %.2f' % r2_score(ytest, ypred))
print('Mean absolute error: %.2f' % mean_absolute_error(ytest, ypred))
print('Mean squared error: %.2f' % mean_squared_error(ytest, ypred))
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('Виміряно')
ax.set_ylabel('Передбачено')
plt.show()
```

Рис 2.12 – лістинг програми

```
C:\Users\Администратор\PycharmProjects\.venv\Scripts\python.exe C:\Users\Ад
Coefficients:
[ -20.4047621  -265.88518066  564.65086437  325.56226865 -692.16120333
 395.55720874  23.49659361  116.36402337  843.94613929  12.71856131]
Intercept:
154.3589285280134
Coefficient of determination: 0.44
Mean absolute error: 44.80
Mean squared error: 3075.33
```

Рис 2.13 – результат виконання

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

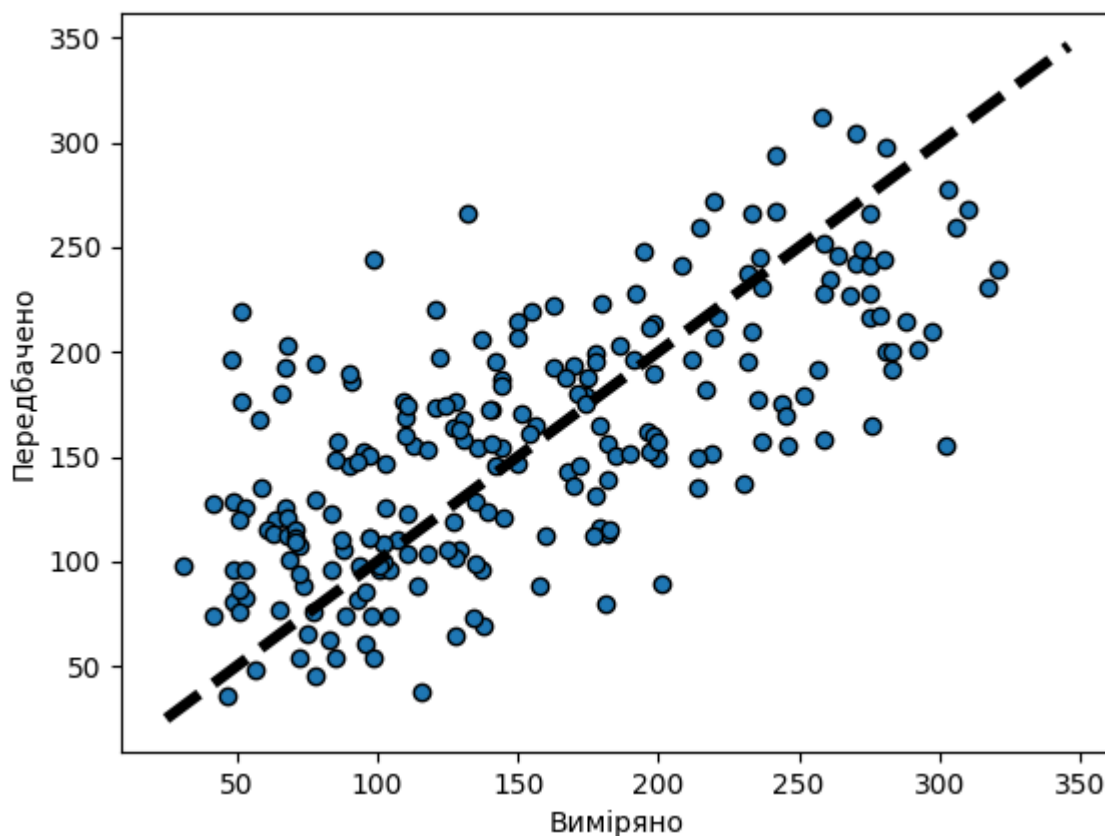


Рис 2.14 – результат виконання

З графіка видно, що з підвищенням фактичного рівня цукру в крові прогнозований рівень також збільшується, що свідчить про наявність позитивної кореляції між цими двома змінними. Лінія регресії на графіку відображає найкраще наближення до спостережуваних даних, оскільки мінімізує різницю між реальними значеннями та передбаченими моделлю. Хоча модель показує деяке співвідношення між змінними, вона здатна пояснити лише 44% дисперсії в даних, що свідчить про помірну здатність до прогнозування. Середня абсолютна похибка (MAE) моделі становить 44.8, що вказує на помітні відхилення прогнозованих значень від реальних. Крім того, середня квадратична похибка (MSE) дорівнює 3075.33, що підтверджує наявність великих помилок в прогнозах і вказує на необхідність вдосконалення моделі для досягнення більшої точності.

Завдання 2.5. Самостійна побудова регресії.

№ за списком	15
№ варіанту	5

Варіант 5

```
m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.4 * X ** 2 + X + 4 + np.random.randn(m, 1)

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

np.random.seed(0)
m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.4 * X ** 2 + X + 4 + np.random.randn(m, 1)

# Лінійна регресія
linear_regressor = linear_model.LinearRegression()
linear_regressor.fit(X, y)
y_linearpred = linear_regressor.predict(X)

# Поліноміальна регресія
polynomial = PolynomialFeatures(degree=2)
X_poly = polynomial.fit_transform(X)
poly_regressor = linear_model.LinearRegression()
poly_regressor.fit(X_poly, y)
y_polypred = poly_regressor.predict(X_poly) # Use transformed X_poly

# Візуалізація результатів
plt.scatter(X, y, color='green')
plt.plot(X, y_linearpred, color='blue', label='linear', linewidth=4)
sort_indices = np.argsort(X[:, 0])
X_sorted = X[sort_indices]
y_polypred_sorted = y_polypred[sort_indices]
plt.plot(X_sorted, y_polypred_sorted, color='red', label='polynomial', linewidth=4)
plt.legend()
plt.xlabel("X")
plt.ylabel("y")
plt.show()
```

Рис 2.15 – лістинг програми

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

```
# Оцінка продуктивності для обох моделей
print("Linear regressor performance:")
print("Mean absolute error =", round(sm.mean_absolute_error(y, y_linearpred), 2))
print("Mean squared error =", round(sm.mean_squared_error(y, y_linearpred), 2))
print("Median absolute error =", round(sm.median_absolute_error(y, y_linearpred), 2))
print("Explain variance score =", round(sm.explained_variance_score(y, y_linearpred), 2))
print("R2 score =", round(sm.r2_score(y, y_linearpred), 2))
print("\nPolynomial regressor performance:")
print("Mean absolute error =", round(sm.mean_absolute_error(y, y_polypred), 2))
print("Mean squared error =", round(sm.mean_squared_error(y, y_polypred), 2))
print("Median absolute error =", round(sm.median_absolute_error(y, y_polypred), 2))
print("Explain variance score =", round(sm.explained_variance_score(y, y_polypred), 2))
print("R2 score =", round(sm.r2_score(y, y_polypred), 2))
print('\nPolynomial model coefficients:', poly_regressor.coef_)
print('\nPolynomial model coefficients:', poly_regressor.coef_)
print('Polynomial model intercept:', poly_regressor.intercept_)
```

Рис 2.16 – лістинг програми

```
C:\Users\Администратор\PycharmProjects\.venv\Scripts\python.exe C:\Users
Linear regressor performance:
Mean absolute error = 1.11
Mean squared error = 1.89
Median absolute error = 0.9
Explain variance score = 0.57
R2 score = 0.57

Polynomial regressor performance:
Mean absolute error = 0.83
Mean squared error = 0.97
Median absolute error = 0.7
Explain variance score = 0.78
R2 score = 0.78

Polynomial model coefficients: [[0.          0.97906552 0.34978823]]
Polynomial model coefficients: [[0.          0.97906552 0.34978823]]
Polynomial model intercept: [4.34050076]
```

Рис 2.17 – результат виконання

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

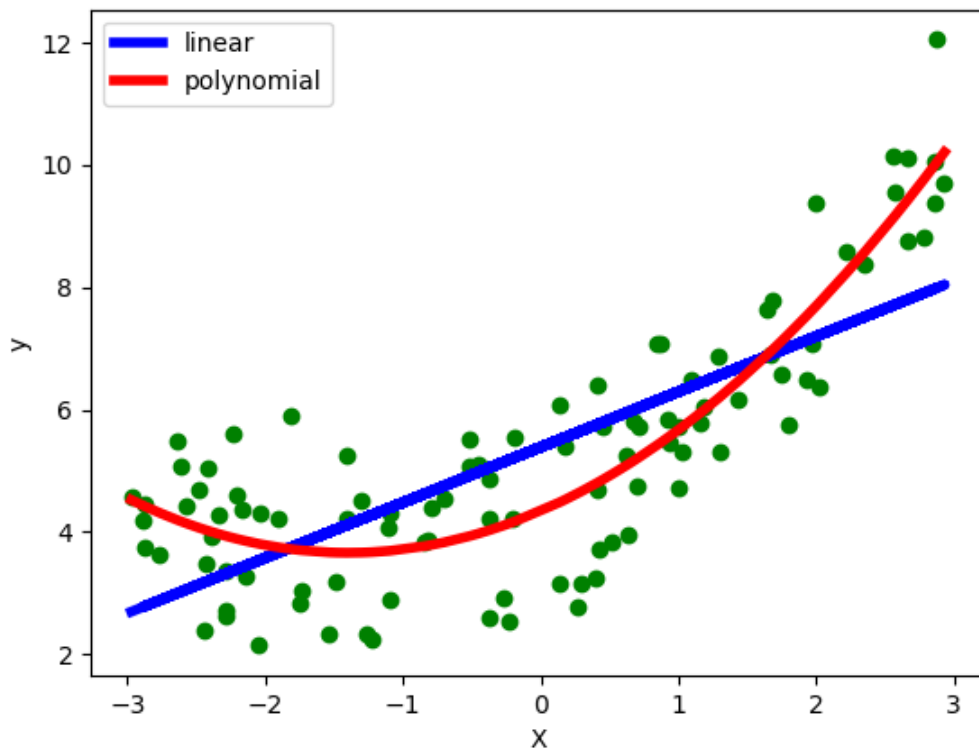


Рис 2.18 – результат виконання

$$Y = 0,4 * x^2 + x + 4$$

$$Y = 0.34 * x^2 + 0.97 * x + 4.34$$

За результатом графіка та показників ефективності видно, що поліноміальна регресія значно краще справляється з нелінійними даними, оскільки здатна моделювати криволінійні залежності, характерні для цих даних

Завдання 2.6. Побудова кривих навчання

```
from matplotlib import pyplot as plt
import numpy as np
from pyexpat import model
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures

np.random.seed(0)
m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.4 * X ** 2 + X + 4 + np.random.randn(m, 1)
```

Рис 2.19 – лістинг програми

```

def plot_learning_curves(model, X, y): 3 usages new *
    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.legend(loc = 'upper right', fontsize = 14)
    plt.xlabel('Training set size', fontsize = 14)
    plt.ylabel('RMSE', fontsize = 14)
    plt.show()

lin_reg = LinearRegression()
plot_learning_curves(lin_reg, X, y)
polynomial_regression = Pipeline([('poly_features',
PolynomialFeatures(degree = 10, include_bias = False)),('lin_reg', LinearRegression())])
plot_learning_curves(polynomial_regression, X, y)
polynomial_regression = Pipeline([('poly_features',
PolynomialFeatures(degree = 2, include_bias = False)),('lin_reg', LinearRegression())])
plot_learning_curves(polynomial_regression, X, y)

```

Рис 2.20 – лістинг програми

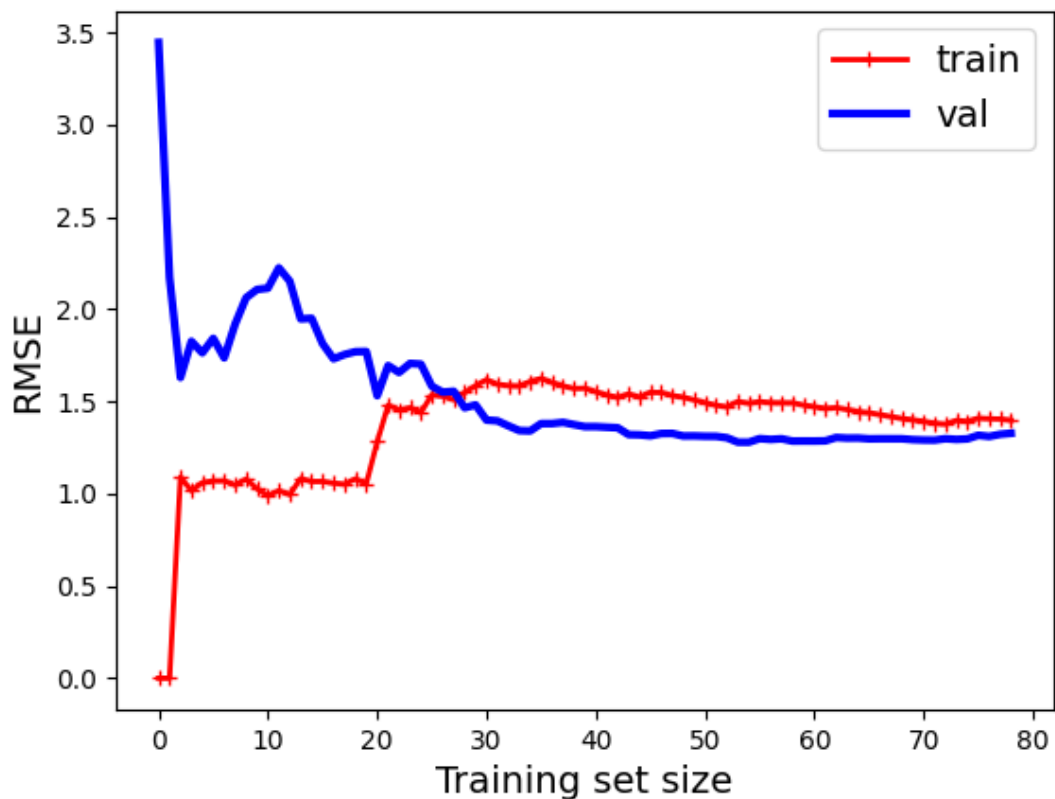


Рис 2.21 – результат виконання

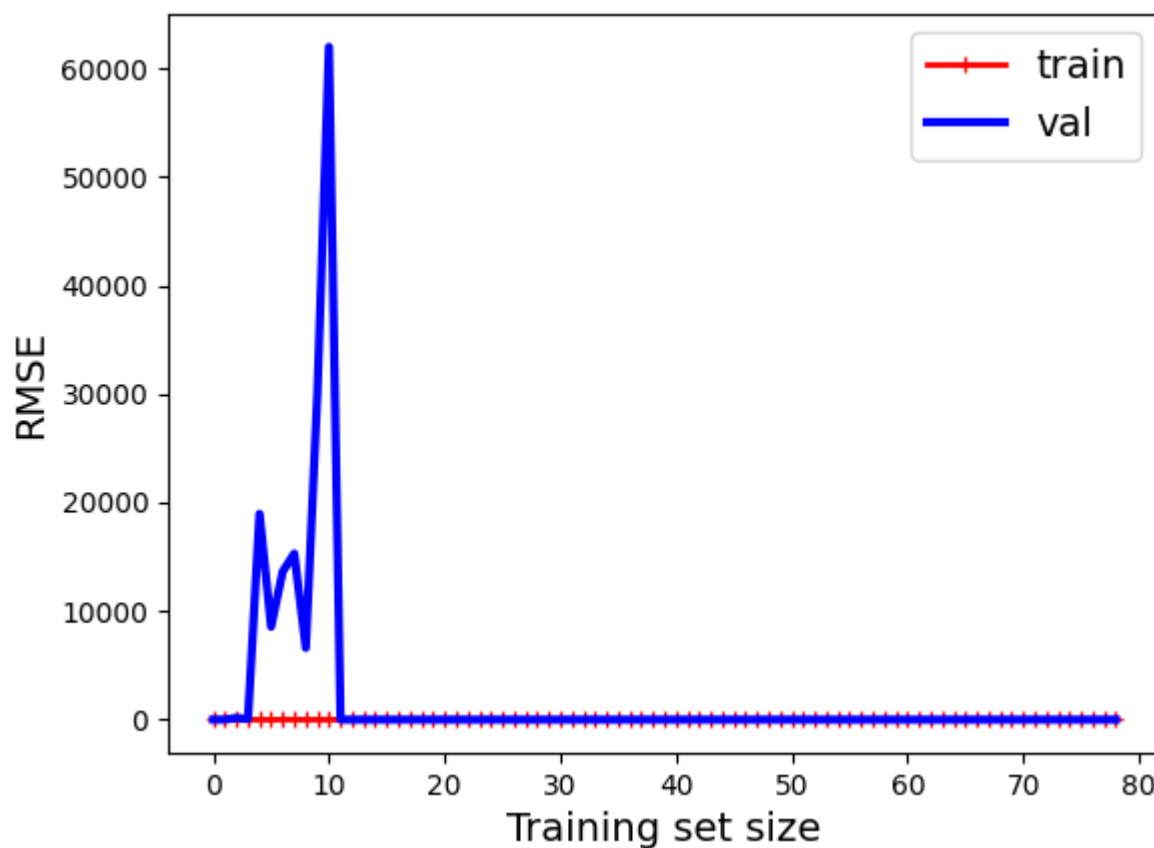


Рис 2.22 – результат виконання

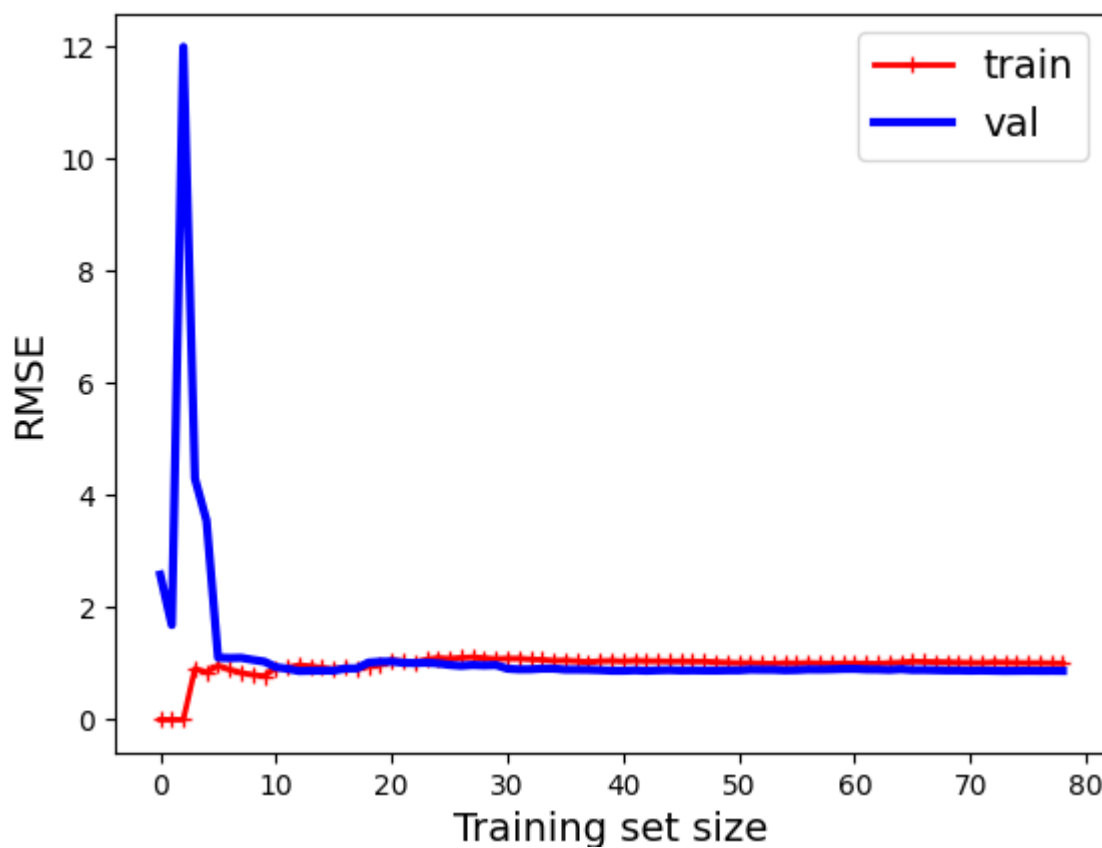


Рис 2.23 – результат виконання

Висновок: На лабораторній роботі використовуючи спеціалізовані бібліотеки та мову програмування Python дослідила методи регресії даних у машинному навчанні.

Посилання на репозиторій:

https://github.com/ipz211/shi_prokopchuk_oleksandra_ipz-21-1

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.4	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		