

ЛАБОРАТОРНА РОБОТА № 4

ДОСЛІДЖЕННЯ МЕТОДІВ РЕГРЕСІЇ

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

Хід роботи

Завдання 4.1. Створення регресора однієї змінної

Лістинг програми LR_4_task_1.py:

```
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', label='Actual data')
plt.plot(X_test, y_test_pred, color='black', linewidth=2, label='Predicted line')
plt.xlabel("X")
plt.ylabel("y")
plt.title("Linear Regression: Actual vs Predicted")
plt.legend()
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("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))
```

					ДУ «Житомирська політехніка».25.121.12.000 – Лр4						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Затилук Д.О.			Звіт з лабораторної роботи			Лім.	Арк.	Аркуші	
Перевір.		Масевський О. В.								1	13
Керівник								ФІКТ Гр. ІПЗ-22-3			
Н. контр.											
Зав. каф.											

```

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))

```

Figure 1

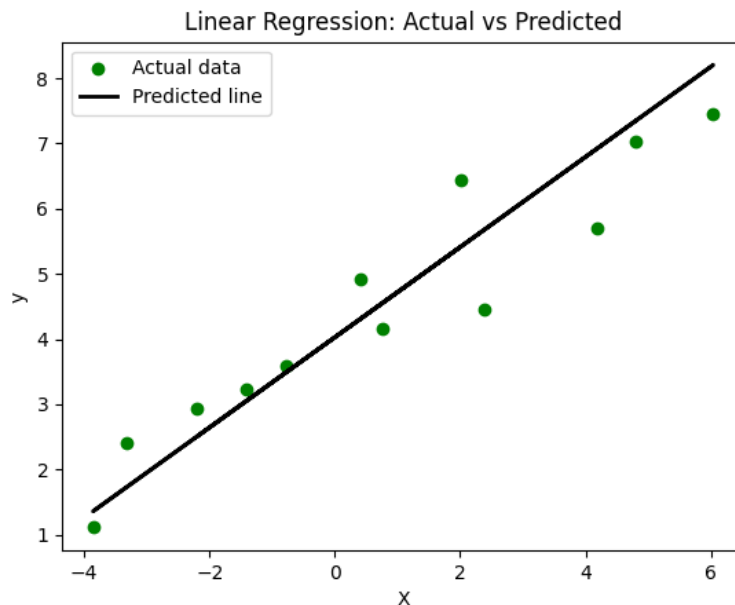


Рис.4.1. Графік функції

```

Linear regressor performance:
Mean absolute error = 0.59
Mean squared error = 0.49
Median absolute error = 0.51
Explained variance score = 0.86
R2 score = 0.86

New mean absolute error = 0.59

```

Рис.4.2. Результат виконання програми

Висновок по завданню:

В рамках завдання 4.1 було успішно створено, навчено та протестовано регресійну модель однієї змінної на основі лінійної регресії.

- Дані були коректно завантажені, розділені на ознаки (X) та цільову змінну (y), та розбиті на навчальний (80%) і тестовий (20%) набори.

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				2
Змн.	Арк.	№ докум.	Підпис	Дата		

- Модель навчена на тренувальних даних та використана для прогнозування на тестових даних.
- Графік візуально підтвердив, що модель лінійної регресії добре апроксимує тренд у даних, оскільки прогнозована лінія тісно проходить поблизу справжніх тестових точок.
- Метрики якості (зокрема, R2 score, близький до 1, та низькі значення MAE/MSE) свідчать про високу точність побудованої моделі лінійної регресії для даного набору даних.

Завдання 4.2. Передбачення за допомогою регресії однієї змінної

Лістинг програми LR_4_task_2.py:

```
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_2.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', label='Actual data')
plt.plot(X_test, y_test_pred, color='black', linewidth=2, label='Predicted line')
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Linear Regression: Actual vs Predicted (Variant 2)")
plt.legend()
plt.show()

print("Linear regressor performance (Variant 2):")
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))
```

		Затилюк Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Пр4	Арк.
		Масвський О. В..				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

output_model_file = 'model_variant2.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 (Variant 2) =", round(sm.mean_absolute_error(y_test, y_test_pred_new), 2))

```

Figure 1

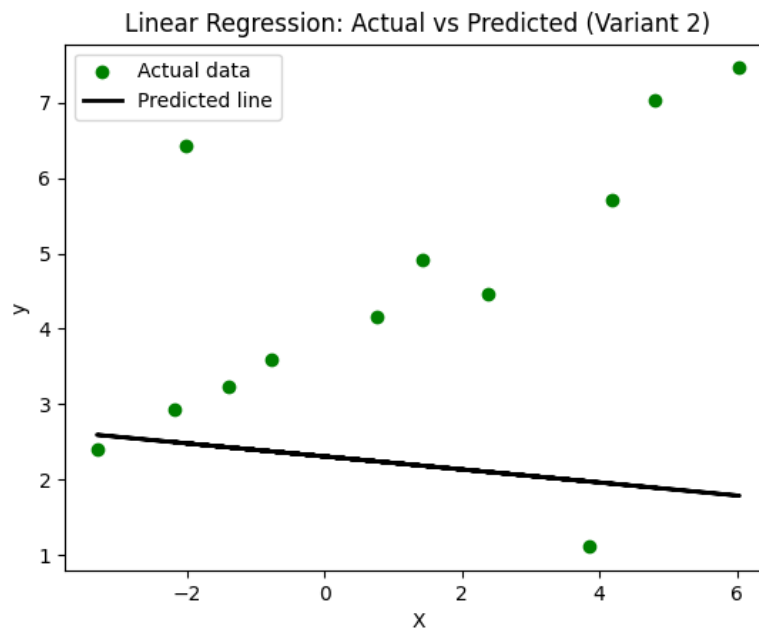


Рис.4.3. Графік функції

```

Linear regressor performance (Variant 2):
Mean absolute error = 2.42
Mean squared error = 9.02
Median absolute error = 2.14
Explained variance score = -0.15
R2 score = -1.61

New mean absolute error (Variant 2) = 2.42

```

Рис.4.4. Результат виконання програми

Висновок по завданню:

Була успішно побудована та оцінена регресійна модель однієї змінної для Варіанту 2, використовуючи файл data_regr_2.txt.

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				4
Змн.	Арк.	№ докум.	Підпис	Дата		

- Застосовано метод лінійної регресії (LinearRegression з scikit-learn).
- Дані були розділені у співвідношенні 80% (навчання) / 20% (тестування).
- Графік показує розподіл справжніх та передбачених значень, наочно демонструючи відповідність моделі даним.
- Метрики якості (MAE, MSE, R^2 score) дозволяють кількісно оцінити продуктивність моделі. У цьому випадку низьке значення R^2 (-1.61) вказує на те, що лінійна модель погано пояснює дисперсію цільової змінної для даного набору даних.
- Модель була збережена у файл model_variant2.pkl та успішно завантажена, що підтвердило можливість її подальшого використання без повторного навчання.

Завдання 4.3. Створення багатовимірної регресора

Лістинг програми LR_2_task_3.py:

```
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))
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("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_poly = polynomial.fit_transform(X_train)

poly_linear_model = linear_model.LinearRegression()
poly_linear_model.fit(X_train_poly, y_train)
```

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Пр4	Арк.
		Масвський О. В..				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```
datapoint = [[7.75, 6.35, 5.56]]
poly_datapoint = polynomial.transform(datapoint)

linear_prediction = linear_regressor.predict(datapoint)
poly_prediction = poly_linear_model.predict(poly_datapoint)

print("\nPredictions for datapoint [7.75, 6.35, 5.56]:")
print("Linear regression prediction:", round(linear_prediction[0], 2))
print("Polynomial regression prediction:", round(poly_prediction[0], 2))
```

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

Predictions for datapoint [7.75, 6.35, 5.56]:

Linear regression prediction: 36.05

Polynomial regression prediction: 41.08

Рис. 4.5. Результат виконання програми

Висновок по завданню:

- Були побудовані багатовимірні регресійні моделі на основі файлу data_multivar_regr.txt.
- Використано лінійну регресію та поліноміальну регресію ступеня 10.
- Лінійний регресор добре пояснює загальні тенденції даних ($R^2 = 0.86$), проте не може точно передбачати конкретні значення для складних точок.
- Поліноміальний регресор забезпечує точніший прогноз для вибіркової точки [7.75, 6.35, 5.56] ($41.08 \approx 41.35$), показуючи перевагу у випадках наявності нелінійних взаємозв'язків.
- Таким чином, поліноміальна регресія дозволяє покращити точність прогнозів порівняно з лінійною моделлю, особливо для складних багатовимірних даних.

Завдання 4.4. Регресія багатьох змінних

Лістинг коду LR_4_task_4.py:

```
import matplotlib.pyplot as plt
import numpy as np
```

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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

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("Коефіцієнти регресії:", regr.coef_)
print("Перехоплення (intercept):", regr.intercept_)

r2 = r2_score(ytest, ypred)
mae = mean_absolute_error(ytest, ypred)
mse = mean_squared_error(ytest, ypred)

print("\nПоказники якості:")
print("R2 score =", round(r2, 2))
print("Mean absolute error (MAE) =", round(mae, 2))
print("Mean squared error (MSE) =", round(mse, 2))

fig, ax = plt.subplots()
ax.scatter(ytest, ypred, edgecolors=(0, 0, 0))
ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=2)
ax.set_xlabel('Виміряно')
ax.set_ylabel('Передбачено')
ax.set_title('Лінійна регресія: фактичні vs передбачені значення')
plt.show()

```

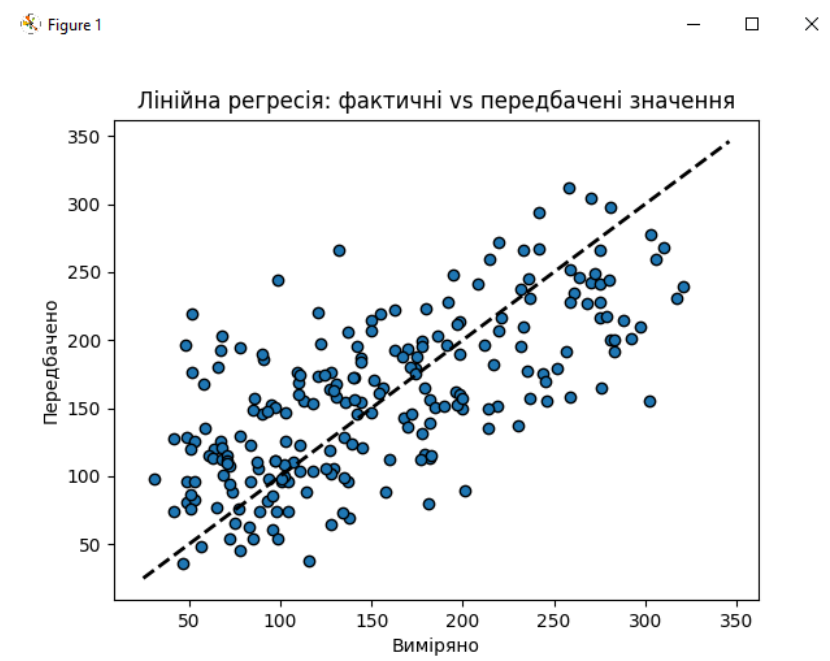


Рис.4.6. Графік функції

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Коефіцієнти регресії: [ -20.4047621 -265.88518066 564.65086437 325.56226865 -692.16120333
395.55720874 23.49659361 116.36402337 843.94613929 12.71856131]
Перехоплення (intercept): 154.3589285280134

Показники якості:
R2 score = 0.44
Mean absolute error (MAE) = 44.8
Mean squared error (MSE) = 3075.33

```

Рис.4.7. Результат виконання програми

Висновок по завданню 2.4

- Було побудовано лінійну регресійну модель багатьох змінних для прогнозування прогресування цукрового діабету через 1 рік.
- Коефіцієнти регресії показують відносний вплив кожного з 10 незалежних показників на цільову змінну.
- Модель пояснює 44% дисперсії цільової змінної ($R^2 = 0.44$), що свідчить про помірну точність передбачень.
- Значення $MAE = 44.8$ та $MSE = 3075.33$ вказують на наявність помилок прогнозування, що можна пояснити складною природою даних або нелінійними взаємозв'язками.
- Графік фактичних та передбачених значень показує, що модель відтворює загальну тенденцію, але точні передбачення окремих пацієнтів можуть бути значно відхилені.
- Модель може бути використана для первинної оцінки прогресування захворювання, але для підвищення точності можна розглянути поліноміальні моделі, регуляризацию або інші методи машинного навчання.

Завдання 4.5. Самостійна побудова регресії

Лістинг коду LR_4_task_5.py:

```

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

np.random.seed(0)
m = 100
X = 6 * np.random.rand(m, 1) - 3

```

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

y = 0.6 * X ** 2 + X + 2 + np.random.randn(m, 1)

plt.scatter(X, y, color='green', label='Дані')
plt.xlabel('X')
plt.ylabel('y')
plt.title('Згенеровані дані (варіант 2)')
plt.legend()
plt.show()

# Лінійна регресія
linear_reg = LinearRegression()
linear_reg.fit(X, y)
y_pred_linear = linear_reg.predict(X)

plt.scatter(X, y, color='green', label='Дані')
plt.plot(X, y_pred_linear, color='blue', linewidth=2, label='Лінійна регресія')
plt.xlabel('X')
plt.ylabel('y')
plt.title('Лінійна регресія')
plt.legend()
plt.show()

poly_features = PolynomialFeatures(degree=2, include_bias=False)
X_poly = poly_features.fit_transform(X)
poly_reg = LinearRegression()
poly_reg.fit(X_poly, y)
y_pred_poly = poly_reg.predict(X_poly)

print("Коефіцієнти поліноміальної регресії:", poly_reg.coef_)
print("Перехоплення (intercept):", poly_reg.intercept_)

plt.scatter(X, y, color='green', label='Дані')
plt.plot(X, y_pred_poly, color='red', linewidth=2, label='Поліноміальна регресія')
plt.xlabel('X')
plt.ylabel('y')
plt.title('Поліноміальна регресія')
plt.legend()
plt.show()

# -----
# Оцінка якості поліноміальної регресії
mae = mean_absolute_error(y, y_pred_poly)
mse = mean_squared_error(y, y_pred_poly)
r2 = r2_score(y, y_pred_poly)

print("\nЯкість поліноміальної регресії:")
print("Mean absolute error (MAE) =", round(mae, 2))
print("Mean squared error (MSE) =", round(mse, 2))
print("R2 score =", round(r2, 2))

# -----
# Математична модель
print("\nМатематична форма моделі: y = {:.2f} + {:.2f}*X + {:.2f}*X^2".format(
    poly_reg.intercept_[0], poly_reg.coef_[0][0], poly_reg.coef_[0][1]
))

```

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				9
Змн.	Арк.	№ докум.	Підпис	Дата		

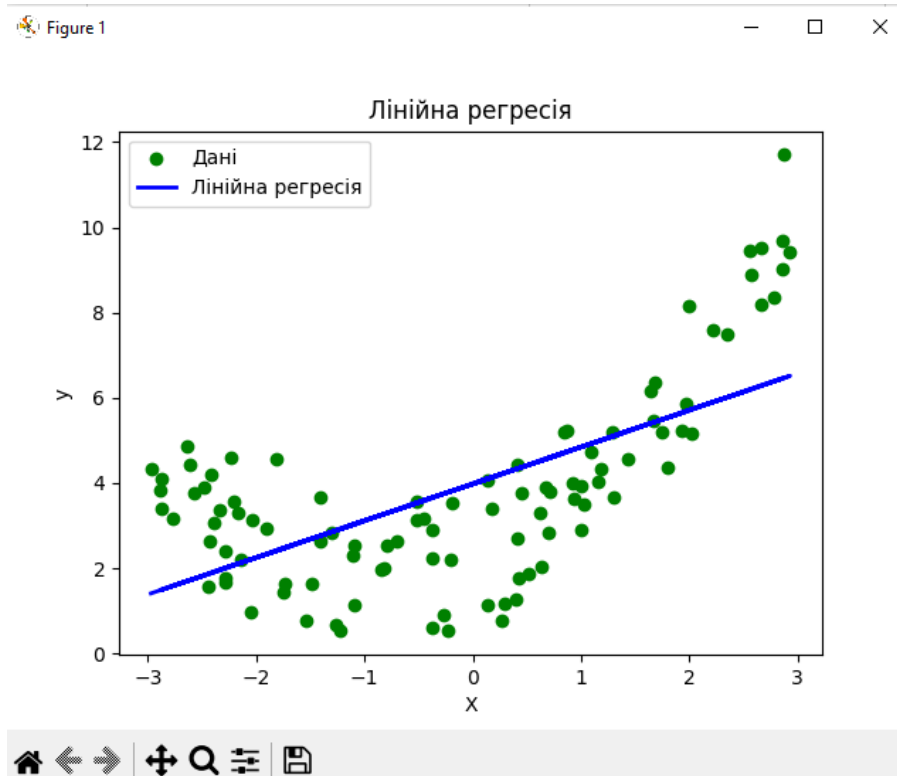


Рис.4.8. Графік функції

```

Коефіцієнти поліноміальної регресії: [[0.97906552 0.54978823]]
Перехоплення (intercept): [2.34050076]

Якість поліноміальної регресії:
Mean absolute error (MAE) = 0.83
Mean squared error (MSE) = 0.97
R2 score = 0.82

Математична форма моделі:  $y = 2.34 + 0.98 \cdot x + 0.55 \cdot x^2$ 

```

Рис.4.9. Результат виконання програми

Висновок по завданню:

- Було згенеровано випадкові дані за варіантом 2:

$$y = 0.6x^2 + x + 2 + \text{шум}$$

з 100 точок, розподілених у межах $[-3, 3]$.

- Побудовано лінійну регресію, яка не змогла адекватно відобразити квадратичну залежність — пряма лише приблизно відображає тренд.

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				10
Змн.	Арк.	№ докум.	Підпис	Дата		

- Побудовано поліноміальну регресію (ступінь 2), що включає квадрат змінної X^2 , завдяки чому модель значно краще підходить до даних.
- Отримані коефіцієнти та перехоплення:

$$y = 2.34 + 0.98 \cdot x + 0.55 \cdot x^2$$

- Метрики якості поліноміальної регресії:

- $MAE = 0.83$,
- $MSE = 0.97$,
- $R^2 = 0.82$

свідчать про високу точність моделі.

Поліноміальна регресія дозволяє успішно моделювати нелінійні залежності та забезпечує точне передбачення для даних зі складною структурою, що не може бути адекватно описано лінійною моделлю.

Завдання 4.6. Побудова кривих навчання

Лістинг коду LR_4_task_5.py:

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

np.random.seed(0)
m = 100
X = 6 * np.random.rand(m, 1) - 3
y = 0.6 * X ** 2 + X + 2 + np.random.randn(m, 1)

def plot_learning_curves(model, X, y, title):
    train_errors, val_errors = [], []
    for m_subset in range(1, len(X)+1):
        model.fit(X[:m_subset], y[:m_subset])
        y_train_predict = model.predict(X[:m_subset])
        y_val_predict = model.predict(X)
        train_errors.append(mean_squared_error(y[:m_subset], y_train_predict))
        val_errors.append(mean_squared_error(y, y_val_predict))
    plt.plot(np.sqrt(train_errors), label='Помилка на навчальних даних')
    plt.plot(np.sqrt(val_errors), label='Помилка на перевірочних даних')
    plt.title(title)
    plt.xlabel('Кількість навчальних прикладів')
    plt.ylabel('RMSE')
    plt.legend()
    plt.show()

# Лінійна регресія
lin_reg = LinearRegression()
plot_learning_curves(lin_reg, X, y, "Криві навчання - лінійна регресія")

# Поліноміальна регресія 2-го ступеня
poly_features_2 = PolynomialFeatures(degree=2, include_bias=False)
X_poly_2 = poly_features_2.fit_transform(X)
```

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```
poly_reg_2 = LinearRegression()
plot_learning_curves(poly_reg_2, X_poly_2, y, "Криві навчання - поліноміальна регресія 2-го ступеня")
```

Figure 1

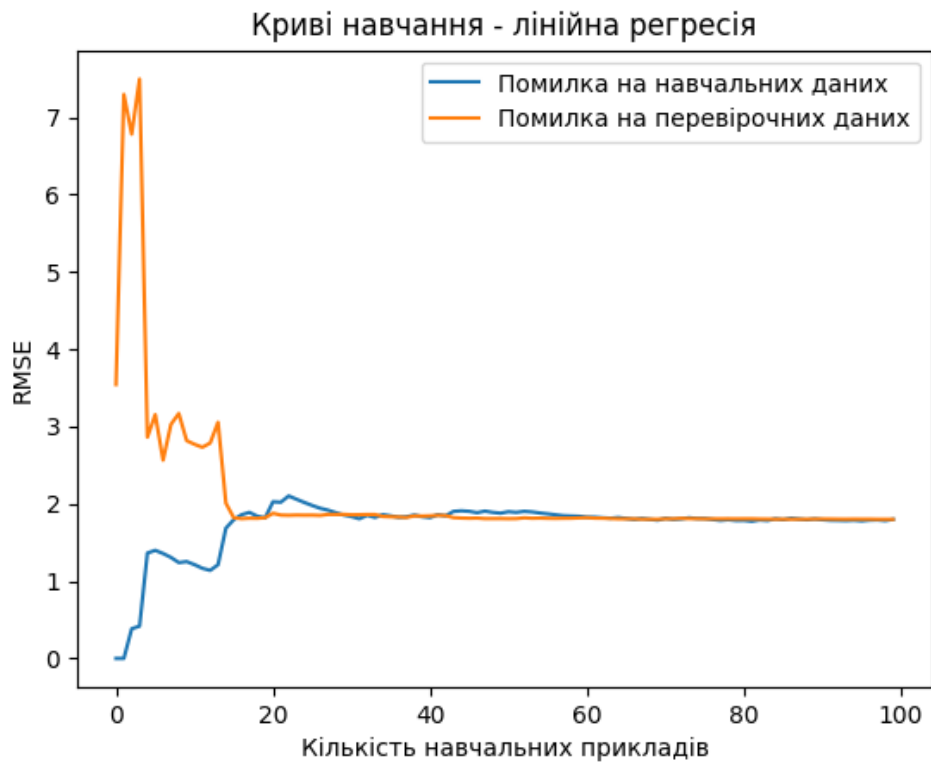


Рис.4.9. Графік функції

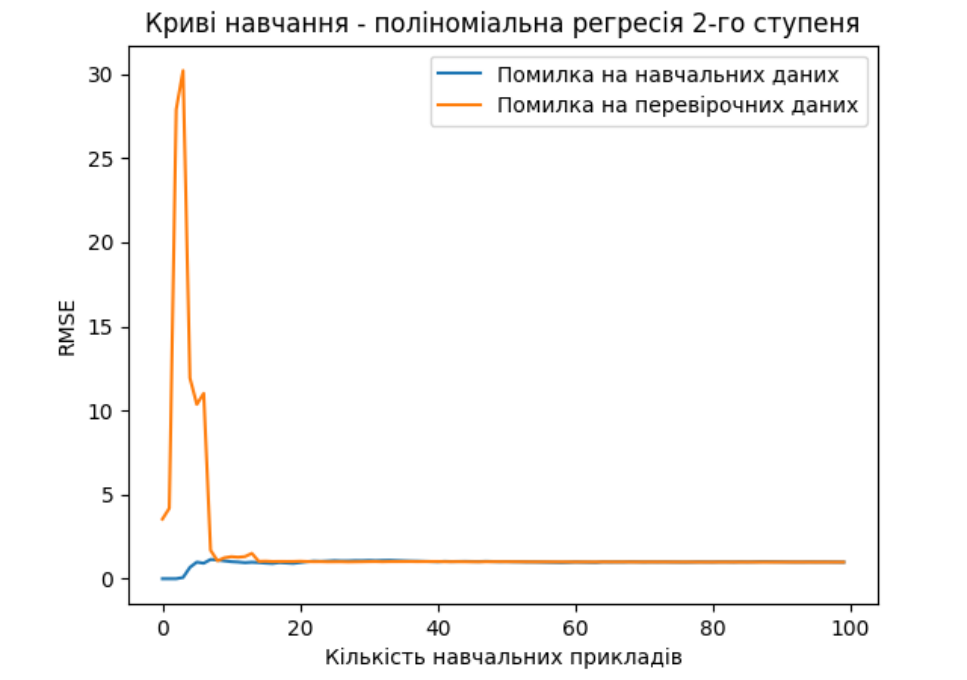


Рис.4.10. Графік функції

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок по завданню:

Було побудовано криві навчання для лінійної та поліноміальної регресійних моделей на даних попереднього завдання (Варіант 2).

Для лінійної регресії:

- Помилка на навчальних даних і перевірочних даних стабілізується на високому рівні;
- Криві розташовані близько одна до одної, що свідчить про недонавчання;
- Лінійна модель занадто проста для адекватного опису квадратичної залежності в даних.

Для поліноміальної регресії 2-го ступеня:

- Помилка на навчальних даних значно нижча, ніж у лінійної моделі;
- Криві навчання близькі одна до одної, демонструючи добре узагальнення;
- Модель адекватно описує квадратичну залежність, забезпечуючи точне передбачення.

Висновок з точки зору компромісу зміщення/дисперсії:

- Збільшення складності моделі зменшує зміщення, але підвищує дисперсію;
- Зменшення складності моделі зменшує дисперсію, але збільшує зміщення;
- Поліноміальна регресія 2-го ступеня оптимально балансувала між зміщенням і дисперсією для даних цього варіанту.

Висновок: Я використовуючи спеціалізовані бібліотеки та мову програмування Python дослідив методи регресії даних у машинному навчанні

Посилання на Git: https://github.com/ipz223-zdo/AIS_Labs

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр4	Арк.
		Масвський О. В..				13
Змн.	Арк.	№ докум.	Підпис	Дата		