Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Кафедра програмної інженерії

ЗВІТ

Лабораторної роботи № 1

з дисципліни «Інтелектуальний аналіз даних»

на тему «Регресійний аналіз»

Виконав

студент групи ІПЗм-24-2 Голодніков Дмитро

Перевірив

ст. викл. Онищенко К.Г.

Харків 2024

1 МЕТА РОБОТИ

Ознайомитися з методами регресійного аналізу даних. Вивчити побудову та аналіз моделей простої лінійної, множинної лінійної та поліноміальної регресії. Навчитися оцінювати якість регресійних моделей за допомогою коефіцієнта детермінації R^2, RMSE та інших метрик. Отримати практичні навички роботи з бібліотеками scikit-learn та statsmodels.

2 ХІД ВИКОНАННЯ РОБОТИ

2.1 Теоретичні основи регресійного аналізу

Регресійний аналіз - статистичний метод дослідження впливу однієї або кількох незалежних змінних (предикторів) на залежну змінну (відгук). Метою регресії є побудова моделі, що описує цю залежність та дозволяє робити прогнози.

Основні метрики оцінки якості регресійної моделі:

• R^2 (коефіцієнт детермінації) - частка дисперсії залежної змінної, що пояснюється моделлю. Значення від 0 до 1, де 1 означає ідеальне пояснення.

• RMSE (Root Mean Square Error) - корінь із середнього квадрата помилок. Показує середню величину відхилення прогнозів від фактичних значень.

• MAE (Mean Absolute Error) - середня абсолютна помилка. Менш чутлива до викидів порівняно з RMSE.

2.2 Проста лінійна регресія

Проста лінійна регресія моделює залежність між однією незалежною змінною X та залежною змінною Y у вигляді лінійного рівняння:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

де β_0 - вільний член (intercept), що визначає значення Y при X=0; β_1 - коефіцієнт регресії (slope), що показує на скільки зміниться Y при зміні X на одиницю; ε - випадкова помилка моделі.

Для оцінки параметрів β_0 та β_1 використовується метод найменших квадратів (OLS), який мінімізує суму квадратів відхилень: $\Sigma(y_i - \hat{y}_i)^2 \rightarrow \min$.

Для демонстрації було згенеровано вибіркові дані та побудовано модель простої лінійної регресії. Реалізація на Python з використанням scikit-learn:

```
from sklearn.linear_model import LinearRegression
import numpy as np

# Підготовка даних
X = data[['VAR1']].values
y = data['VAR3'].values

# Побудова моделі
model = LinearRegression()
model.fit(X, y)

# Оцінка якості
y_pred = model.predict(X)
r2 = r2_score(y, y_pred)
rmse = np.sqrt(mean_squared_error(y, y_pred))
```

На графіку показано: точки даних (синім), лінію регресії (червоним), та 95% довірчий інтервал (світло-червоним). Модель показала $R^2 = 0.99$, що свідчить про відмінну якість апроксимації.

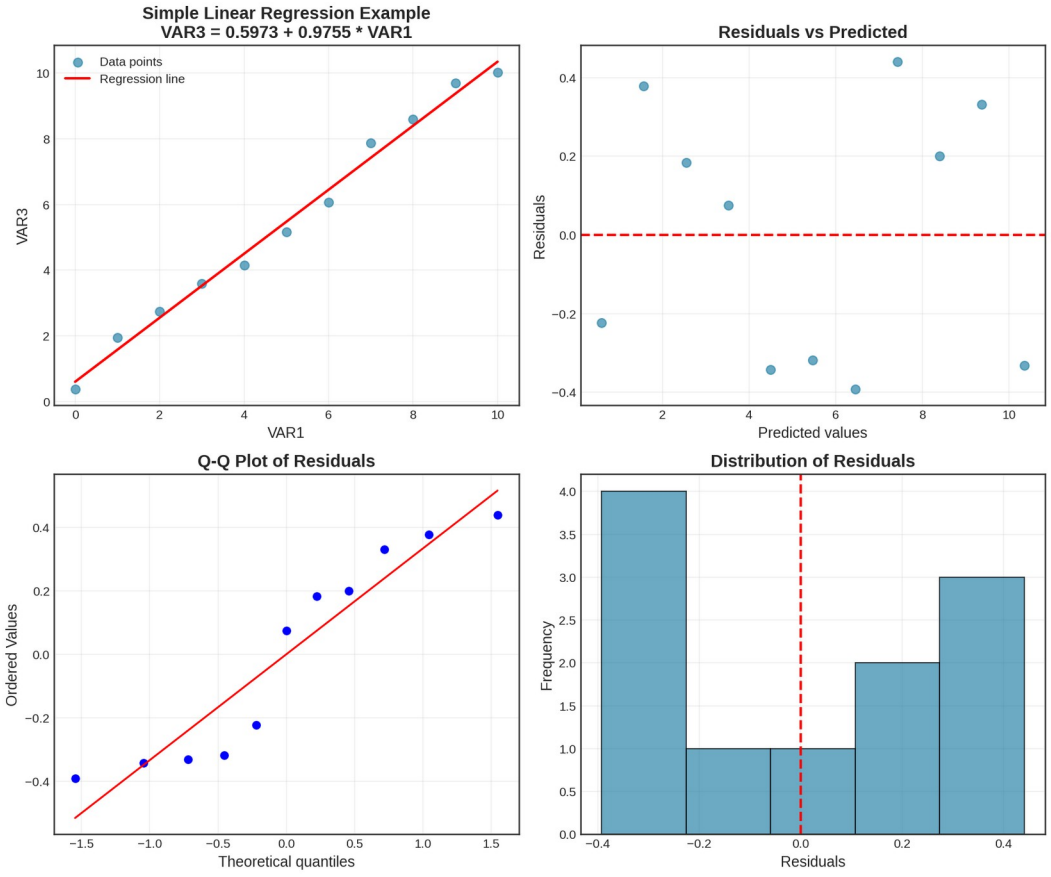

Рис. 2.1 - Проста лінійна регресія з довірчим інтервалом та діагностичними графіками

2.3 Множинна лінійна регресія

Множинна лінійна регресія розширює модель простої регресії для випадку кількох незалежних змінних:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon$$

Це дозволяє врахувати вплив декількох факторів на залежну змінну одночасно. Кожен коефіцієнт β_i інтерпретується як вплив змінної X_i на Y при фіксованих значеннях інших змінних.

Було побудовано модель множинної регресії для прогнозування врожайності на основі факторів кількості добрив (X) та опадів (Z). Отримане рівняння: $Y = 28.10 + 0.038X + 0.833Z$. Модель показала $R^2 = 0.981$, Adjusted $R^2 = 0.972$.

Коефіцієнти моделі: збільшення добрив на 100 одиниць дає приріст врожаю на 3.8 од., збільшення опадів на 10 мм дає приріст на 8.3 од.

2.4 Поліноміальна регресія

Поліноміальна регресія використовується для моделювання нелінійних залежностей, коли зв'язок між змінними не можна адекватно описати прямою лінією. Модель має вигляд:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + ... + \beta_n X^n + \varepsilon$$

Ступінь полінома n обирається на основі аналізу даних, перевірки якості моделі та принципу парсимонії (Оккама) - не варто ускладнювати модель без потреби.

Для демонстрації згенеровано дані з параболічною залежністю та шумом. Реалізація поліноміальної регресії:

```
from sklearn.preprocessing import PolynomialFeatures

# Створення поліноміальних ознак
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X.reshape(-1, 1))

# Побудова моделі
model = LinearRegression()
```

```
model.fit(X_poly, y)

# Коефіцієнти: [intercept, x, x²]
coefficients = model.coef_
```

На графіку видно, що крива добре апроксимує нелінійний характер даних.

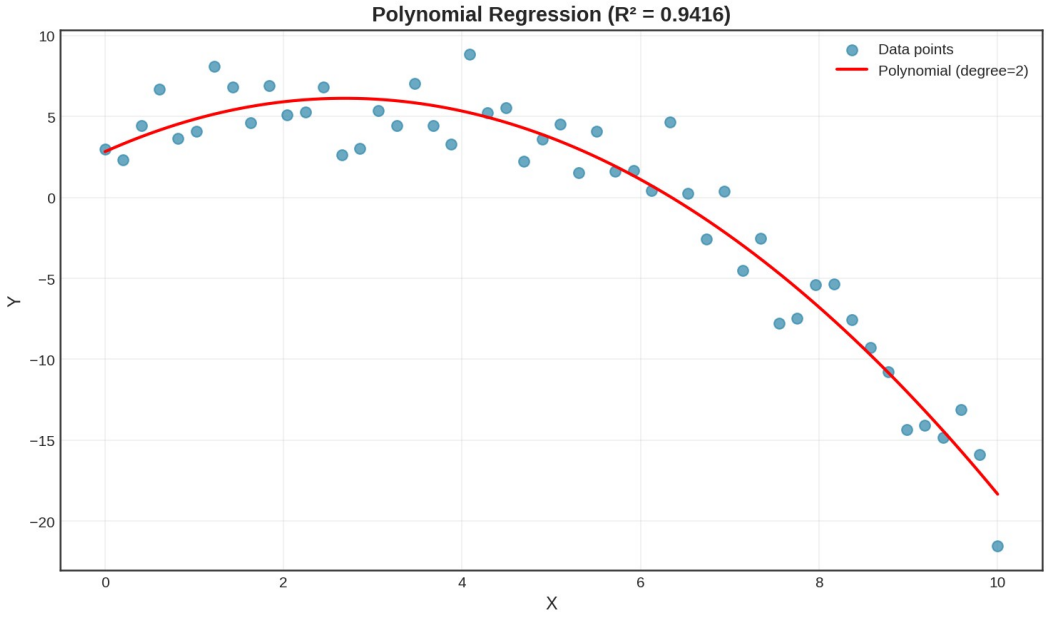

Рис. 2.2 - Поліноміальна регресія ступеня 2 з візуалізацією апроксимації

2.5 Реалізація на Python

Для реалізації використано бібліотеки: scikit-learn (LinearRegression, PolynomialFeatures), statsmodels (OLS з детальною статистикою), numpy (числові операції), matplotlib та seaborn (візуалізація).

3 ВИСНОВКИ

У ході виконання лабораторної роботи було досліджено методи регресійного аналізу: просту лінійну, множинну лінійну та поліноміальну регресію. Вивчено теоретичні основи методу найменших квадратів та метрики оцінки якості моделей. Реалізовано алгоритми побудови регресійних моделей мовою Python з використанням бібліотек scikit-learn та statsmodels. Проведено оцінку якості моделей за метриками R^2, RMSE, MAE. Візуалізовано результати регресійного аналізу з довірчими інтервалами та діагностичними графіками.

Отримані знання можуть бути застосовані для прогнозування та аналізу залежностей у реальних даних.

ПОСИЛАННЯ

Код проєкту доступний у репозиторії GitHub: https://github.com/na-naina/data-analysis-khnure