Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп’ютерних технологій
Кафедра системного проектування

Звіт
Про виконання лабораторної роботи №2
З курсу «Системи машинного навчання»
Регресійні моделі

Виконала:
Студентка групи ФЕС-32
Філь Дарина

Перевірив:
Доцент Колич І.І.

Львів 2024

Мета: Засвоїти основи регресійного аналізу з використанням різних моделей.

Інструменти: Python, Scikit-learn, Matplotlib, Seaborn.

Теоретичні відомості

Лінійна прогресія

Лінійна регресія є основним методом машинного навчання для моделювання взаємозв'язків між змінними. Вона дозволяє передбачати значення залежної змінної на основі незалежних змінних.

Формула лінійної регресії

Формула лінійної регресії (з більш ніж однією незалежною змінною) виглядає так:

$$\boldsymbol{y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n}$$

де:

- $\boldsymbol{y}$ – залежна змінна;
- $\boldsymbol{\beta_0}$ – вільний член (intercept);
- $\boldsymbol{\beta_1}, \dots, \boldsymbol{\beta_n}$ – коефіцієнти регресії для кожної незалежної змінної;
- $\boldsymbol{x_1}, \dots, \boldsymbol{x_n}$ –незалежні змінні;

Основні поняття:

1. Залежна змінна (Target Variable): Це змінна, яку ми намагаємося передбачити або пояснити.

2. Незалежні змінні (Predictors, Features): Це змінні, які ми використовуємо для передбачення значення залежної змінної.

3. Вільний член (Intercept): Значення залежної змінної, коли всі незалежні змінні дорівнюють нулю.

4. Коефіцієнти регресії (Regression Coefficients): Значення, які визначають вплив кожної незалежної змінної на залежну змінну.

Поліноміальна регресія

Поліноміальна регресія є узагальненням лінійної регресії, яка дозволяє моделювати нелінійні взаємозв'язки між змінними.

Формула поліноміальної регресії:

$$\boldsymbol{y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_d x^d}$$

де:

- $\boldsymbol{y}$ – залежна змінна;

- $\boldsymbol{\beta_0}$ − вільний член(intercept);
- $\boldsymbol{\beta_1, \ldots, \beta_d}$ −коефіцієнти регресії для кожної незалежної змінної;
- $\boldsymbol{x_1, \ldots, x_n}$ − незалежні змінні;
- $\boldsymbol{d}$ − незалежні змінні;

Основні поняття:
1. Нелінійні взаємозв'язки: Поліноміальна регресія дозволяє моделювати залежності, які не можуть бути адекватно описані лінійною регресією.
2. Ступінь полінома (Degree of Polynomial): Визначає складність моделі. Вищі ступені дозволяють моделювати складніші взаємозв'язки, але також можуть призводити до перенавчання (overfitting).

Оцінка моделі
Після навчання моделі лінійної регресії важливо оцінити її продуктивність. Ось деякі ключові метрики для оцінки моделі:

1. Середньоквадратична помилка (Mean Squared Error, MSE):
Визначення: Середньоквадратична помилка (MSE) є середнім значенням квадратів різниць між фактичними значеннями та передбаченими значеннями. Це міра середньої величини помилки для передбачень моделі.

$$\boldsymbol{MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

де:
- $\boldsymbol{n}$ − кількість спостережнь;
- $\boldsymbol{y_i}$ − фактичне значення;
- $\boldsymbol{\hat{y}_i}$ − передбачене значення;

Інтерпретація:
- Чим менше значення MSE, тим краща модель;
- MSE враховує великі помилки більше, ніж маленькі, оскільки помилки зводяться до квадрату;

2. Середня абсолютна помилка (MAE):
Визначення: Середня абсолютна помилка (MAE) є середнім значенням абсолютних різниць між фактичними значеннями та передбаченими значеннями. Це міра середньої величини абсолютної помилки для передбачень моделі.

$$\mathbf{MAE} = \boldsymbol{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|}$$

Інтерпреція:

- Чим менше значення MAE, тим краща модель;
- MAE є більш інтерпретованою, оскільки виражено в тих же одиницях, що і залежна змінна;

3. Коефіцієнт детермінації ($\boldsymbol{R^2}$):

Визначення: Коефіцієнт детермінації R^2 показує, яка частка варіації залежної змінної пояснюється незалежними змінними моделі. Це міра того, наскільки добре модель пояснює варіацію в даних.

$$\boldsymbol{R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}i)^2}{\sum i = 1^n (y_i - \bar{y})^2}}$$

де:

- $\bar{y}$ – середнє значення залежної змінної;

Інтерпретація:

- $R^2 = 1$: Модель ідеально пояснює дисперсію залежної змінної;
- $R^2 = 0$: Модель не пояснює дисперсію залежної змінної;
- Чим ближче значення R^2 ло 1, тим краща модель пояснює дані;

Хід роботи:

Завдання

1. **Завантаження готових наборів даних з Scikit-learn:**
   - Завантажити набір sklearn.datasets.fetch_california_housing
2. **Поділ даних на тренувальну та тестову вибірки:**
   - Поділити дані на тренувальний 80% та тестовий 20% набори.
3. **Створити наступні регресійні моделі**
   - Лінійна регресія
   - Поліноміальна регресія ступенем поліному 2
   - Поліноміальна регресія ступенем поліному 3
   - …
   - Поліноміальна регресія ступенем поліному 10
4. **Навчання та оцінка моделей:**
   - Для кожної створеної моделі виконати навчання на основі набору даних з різними ступенями поліному виконати наступні операції
     a. Вивести коефіцієнти моделі з найменшим та найбільшим ступенем поліному
     b. Оцінити продуктивності моделі на тестових даних.
   - Результати оцінки похибок передбачення та коефіцієнта визначеності організувати у вигляді графіків

5. **Візуалізація:** Для регресійної моделі з найменшою похибкою передбачення побудувати графік розкиду (scatter chart), який показує залежність очікуваного та передбаченого результатів в залежності від вхідних характеристик.

   Примітка. Для кращого розділення очікуваного та передбаченого результатів рекомендується спочатку додавати очікуваний результат, а потім передбачений, а також використовувати різні кольори для обох наборів даних, наприклад холодний колір для очікуваних результатів, і теплий колір для передбачених результатів.

   Примітка. Зверніть увагу, що вектор характеристик є багатовимірним вектором, для його візуалізації доцільно вивести окремі двовимірні проекції для кожної характеристики окремо.

6. **Оформити звіт**

```python
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.pipeline import make_pipeline
import numpy as np
import matplotlib.pyplot as plt


data = fetch_california_housing()
X, y = data.data, data.target


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
Командний рядок
Microsoft Windows [Version 10.0.19045.5011]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\Дарина>
```

Рис. 1 Завантаження бібліотек та поділ набору на тренувальний та тестовий

```python
results = {}

model = LinearRegression()
model.fit(X_train, y_train)

y_pred_linear = model.predict(X_test)
mse_linear = mean_squared_error(y_test, y_pred_linear)
r2_linear = r2_score(y_test, y_pred_linear)
results["Linear"] = (mse_linear, r2_linear, model.coef_)
```

```
Командний рядок
Microsoft Windows [Version 10.0.19045.5011]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\Дарина>
```

Рис. 2 Тренування моделі лінійної регресії

```python
degrees = [2,3,4,5,6]

mse_values = []
r2_values = []

for degree in degrees:
    poly = PolynomialFeatures(degree)
    X_train_poly = poly.fit_transform(X_train)
    X_test_poly = poly.transform(X_test)

    model = LinearRegression()
    model.fit(X_train_poly, y_train)

    y_pred_poly = model.predict(X_test_poly)

    mse_poly = mean_squared_error(y_test, y_pred_poly)
    r2_poly = r2_score(y_test, y_pred_poly)

    mse_values.append(mse_poly)
    r2_values.append(r2_poly)
    results[f"Poly {degree}"] = (mse_poly, r2_poly, model.coef_)
```

```
Командний рядок
Microsoft Windows [Version 10.0.19045.5011]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\Дарина>
```

Рис. 3 Тренування моделі поліномної регресії 2-6 ступенів. Відстані ступені з 7 по 10 через те, що тренування моделі займало неймовірно довгий період часу

```python
for model_name, (mse, r2, coef) in results.items():
    print(f"{model_name}: MSE = {mse:.3f}, R2 = {r2:.3f}")
    if isinstance(coef, np.ndarray):
        print(f"Коефіцієнти: {coef[:5]} ... (показано 5 коефіцієнтів з {len(coef)})")
```

```
Linear: MSE = 0.556, R2 = 0.576
Коефіцієнти: [ 4.48674910e-01  9.72425752e-03 -1.23323343e-01  7.83144907e-01
 -2.02962058e-06] ... (показано 5 коефіцієнтів з 8)
Poly 2: MSE = 0.464, R2 = 0.646
Коефіцієнти: [ 6.02531552e-08 -1.19367676e+01 -8.42630148e-01  7.88415388e+00
 -3.83231204e+01] ... (показано 5 коефіцієнтів з 45)
Poly 3: MSE = 23.824, R2 = -17.181
Коефіцієнти: [ 1.32380362e-04 -6.84669880e+01  1.43538066e+01  7.42483563e+01
  2.25325290e+01] ... (показано 5 коефіцієнтів з 165)
Poly 4: MSE = 1646.273, R2 = -1255.304
Коефіцієнти: [ 1.01715837e-04 -2.15557876e-05 -9.65328919e-07 -1.35456189e-07
 -7.79908496e-08] ... (показано 5 коефіцієнтів з 495)
Poly 5: MSE = 47320.089, R2 = -36109.931
Коефіцієнти: [-4.84059823e-09 -1.62943074e-07  2.15299390e-10  1.08175743e-12
 -8.00906521e-11] ... (показано 5 коефіцієнтів з 1287)
Poly 6: MSE = 14882.299, R2 = -11355.988
Коефіцієнти: [-2.44036226e-14  9.21096650e-13 -2.23972676e-15  1.36592562e-15
  9.80928107e-16] ... (показано 5 коефіцієнтів з 3003)
```

```
Командний рядок
Microsoft Windows [Version 10.0.19045.5011]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\Дарина>
```

Рис. 4 Результати тренування моделей лінійної регресії та поліномної регресії різних ступенів показують, що після другого ступеня поліному коефіцієнт детермінації знижується, тоді як помилка збільшується. Це свідчить про ефект перетренування, що погіршує результати моделі. У цьому випадку, ймовірно, перетренування викликане збільшенням кількості коефіцієнтів.

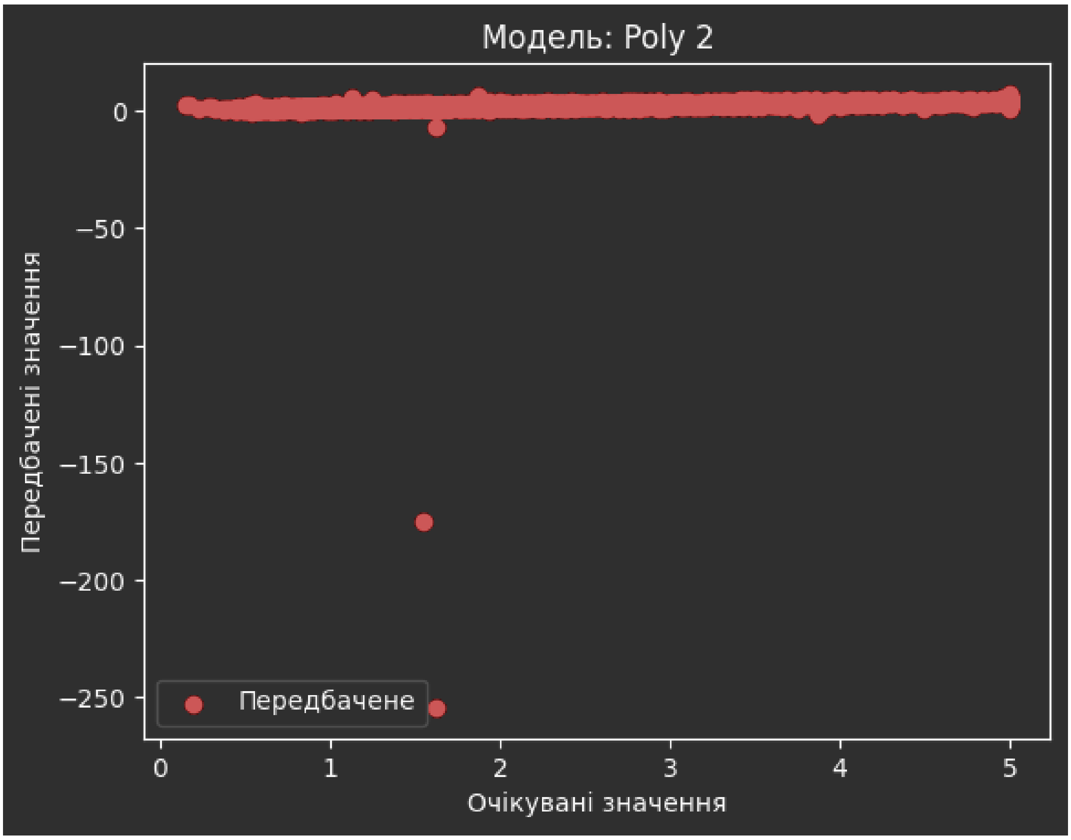
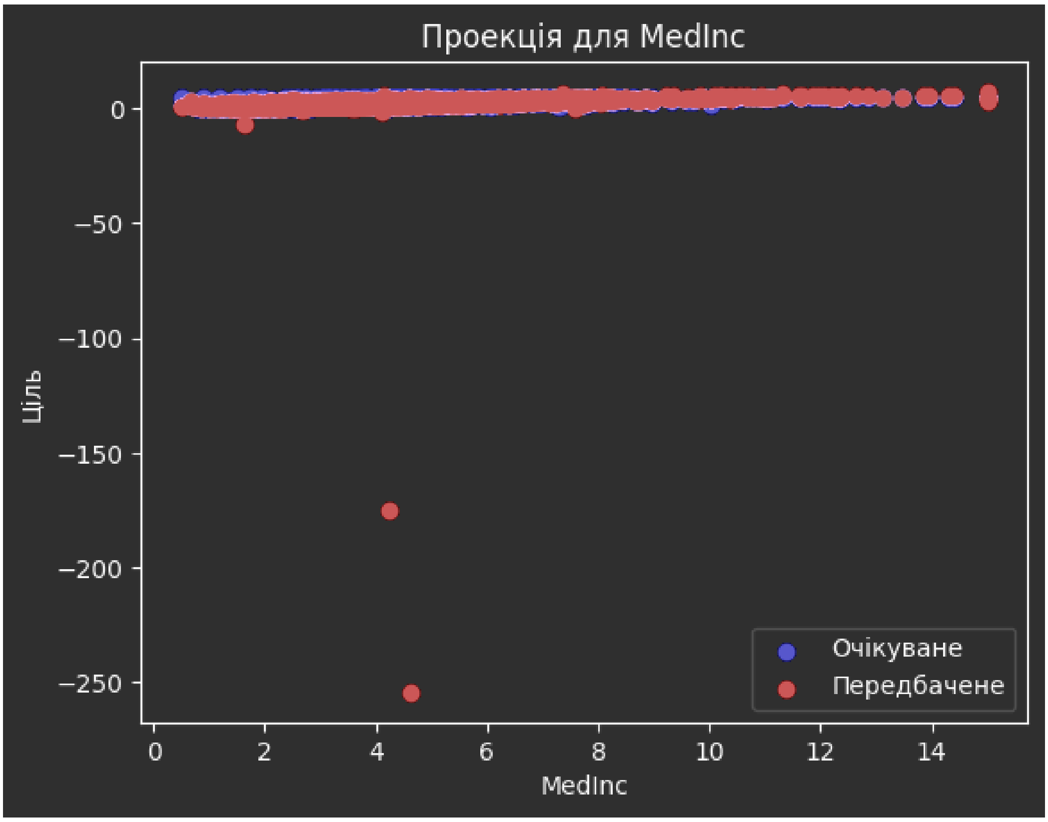
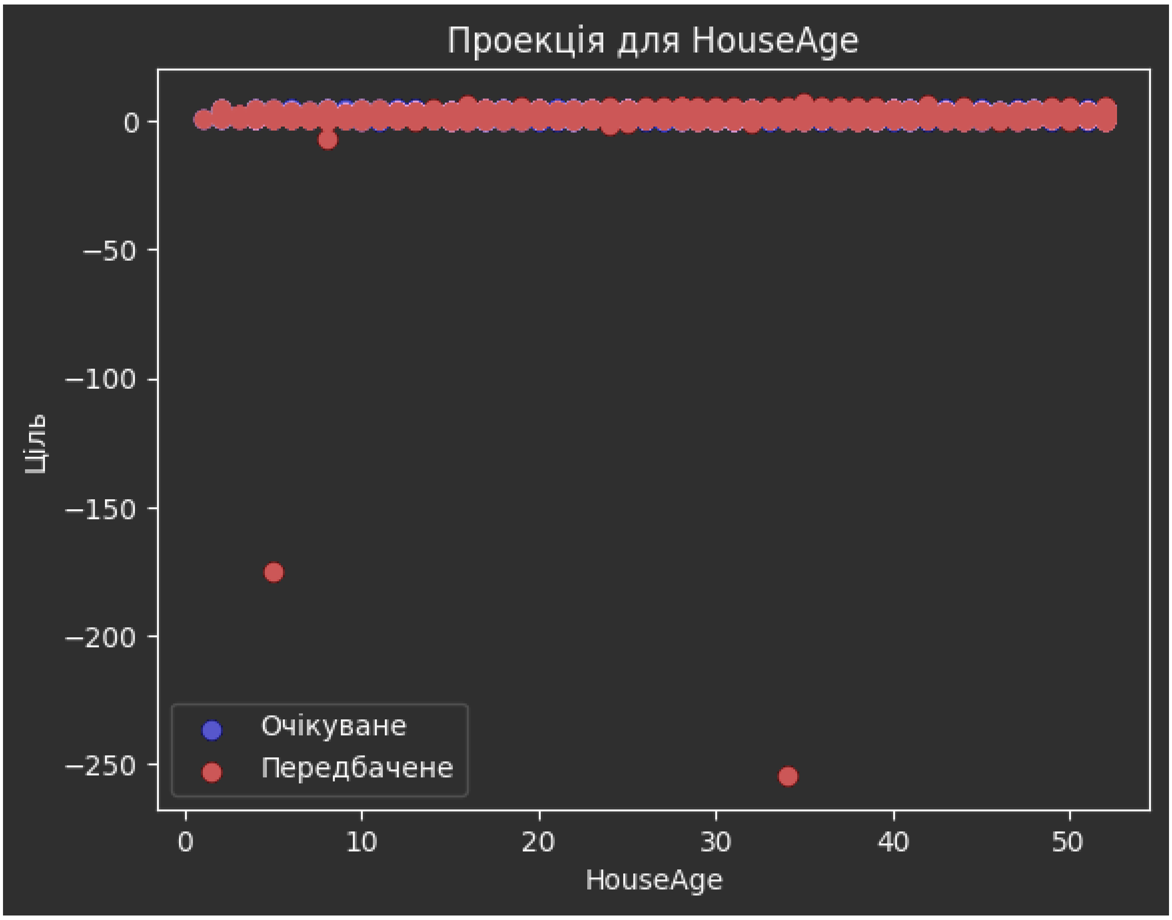
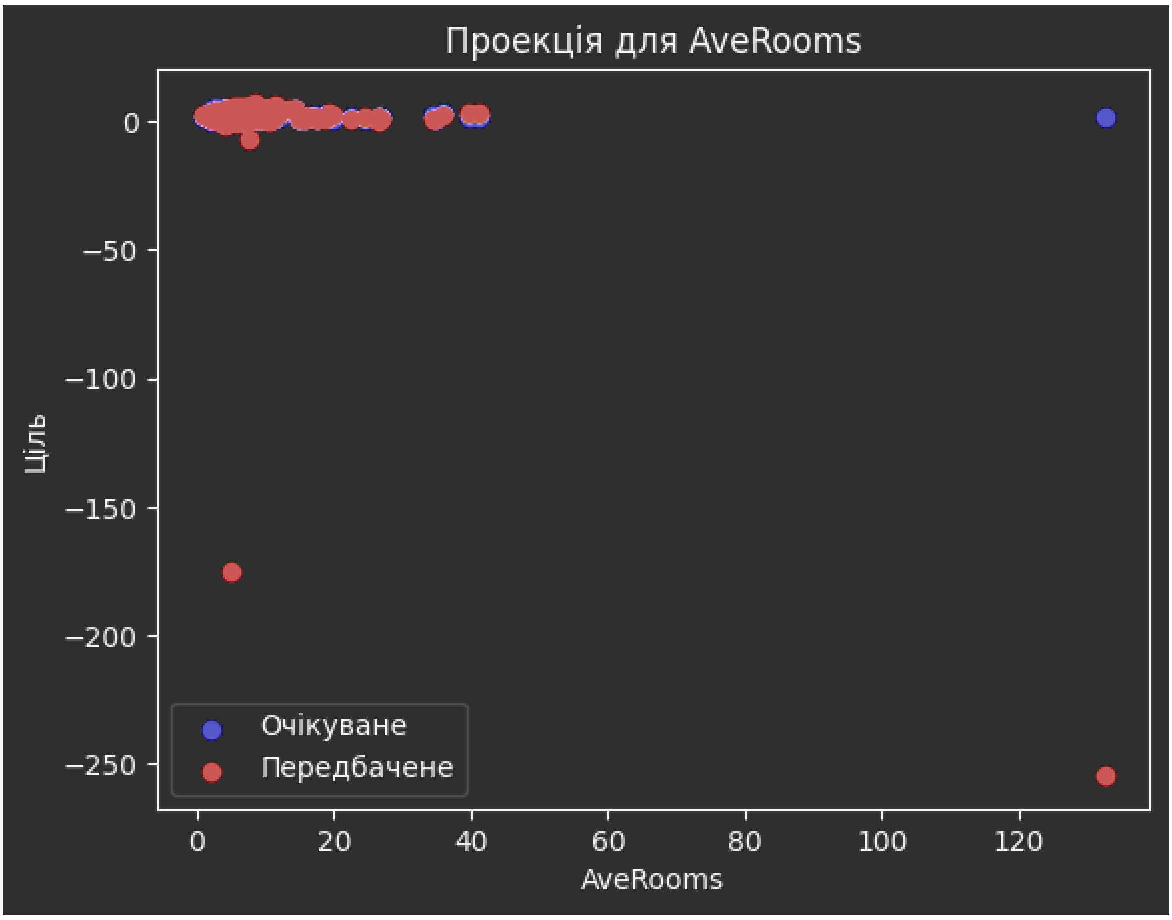


Рис. 5 Згідно з результатами тренування, наведеними на рисунку 4, модель поліномної регресії другого ступеня показала найкращі результати: середня квадратична похибка становить 0.464, а коефіцієнт детермінації – 0.646. Це свідчить про прийнятний рівень точності моделі.

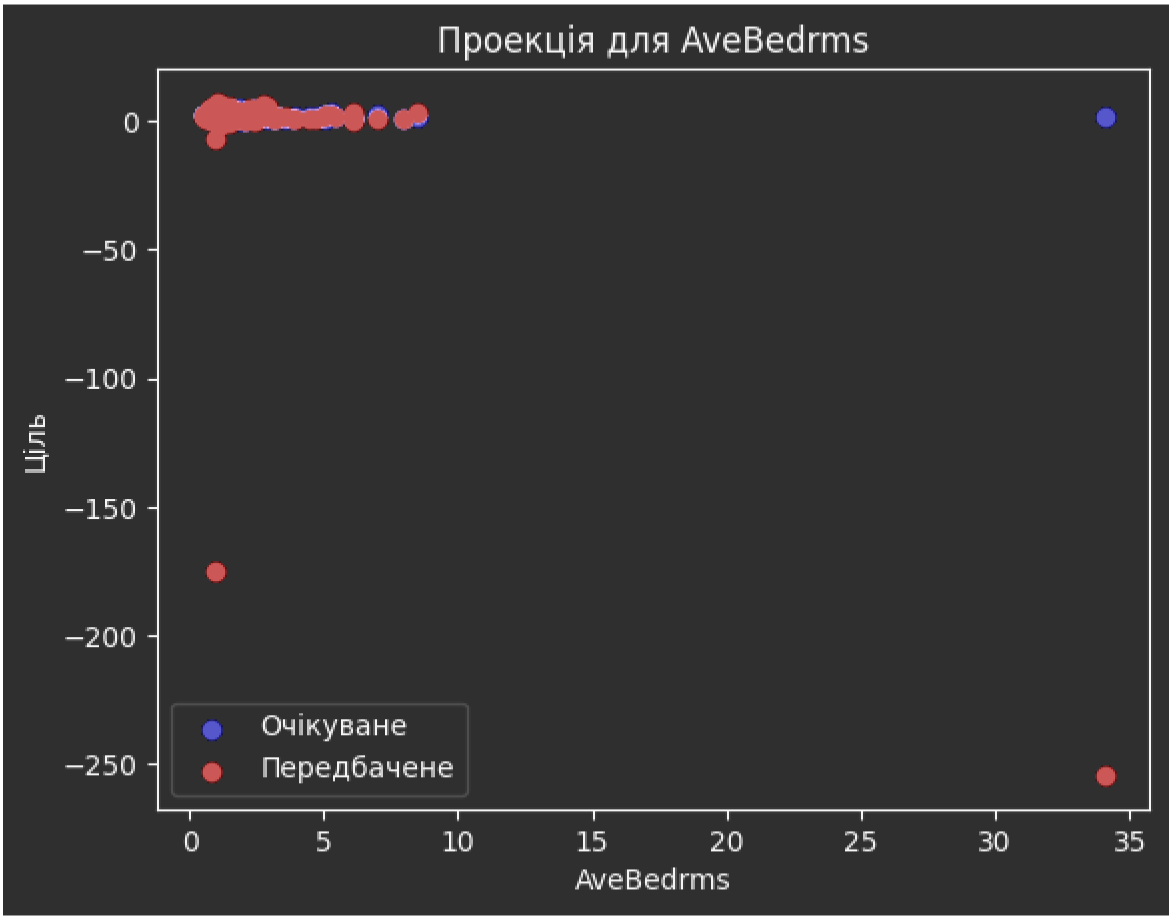


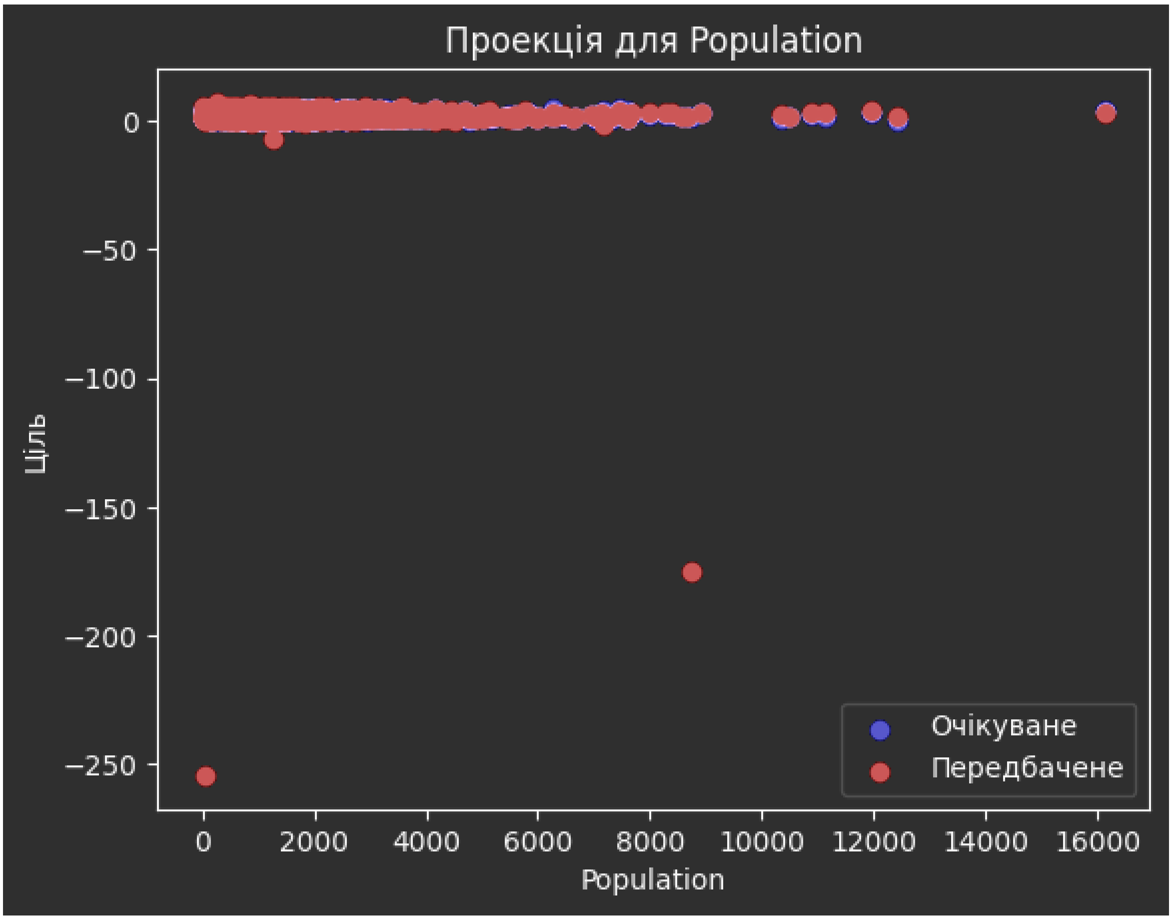


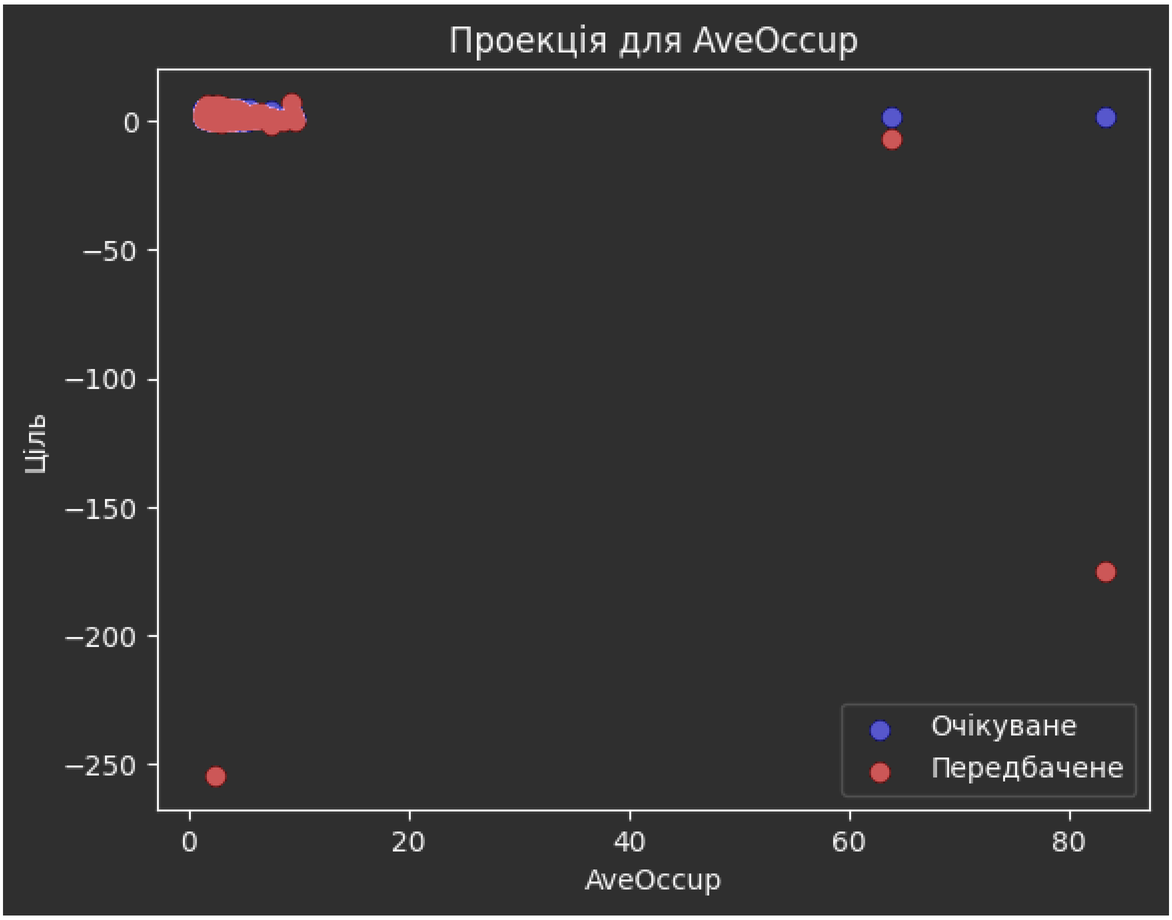
Проекція для HouseAge
0
−50
−100
−150
−200
−250
Ціль
Очікуване
Передбачене
0
10
20
30
40
50
HouseAge



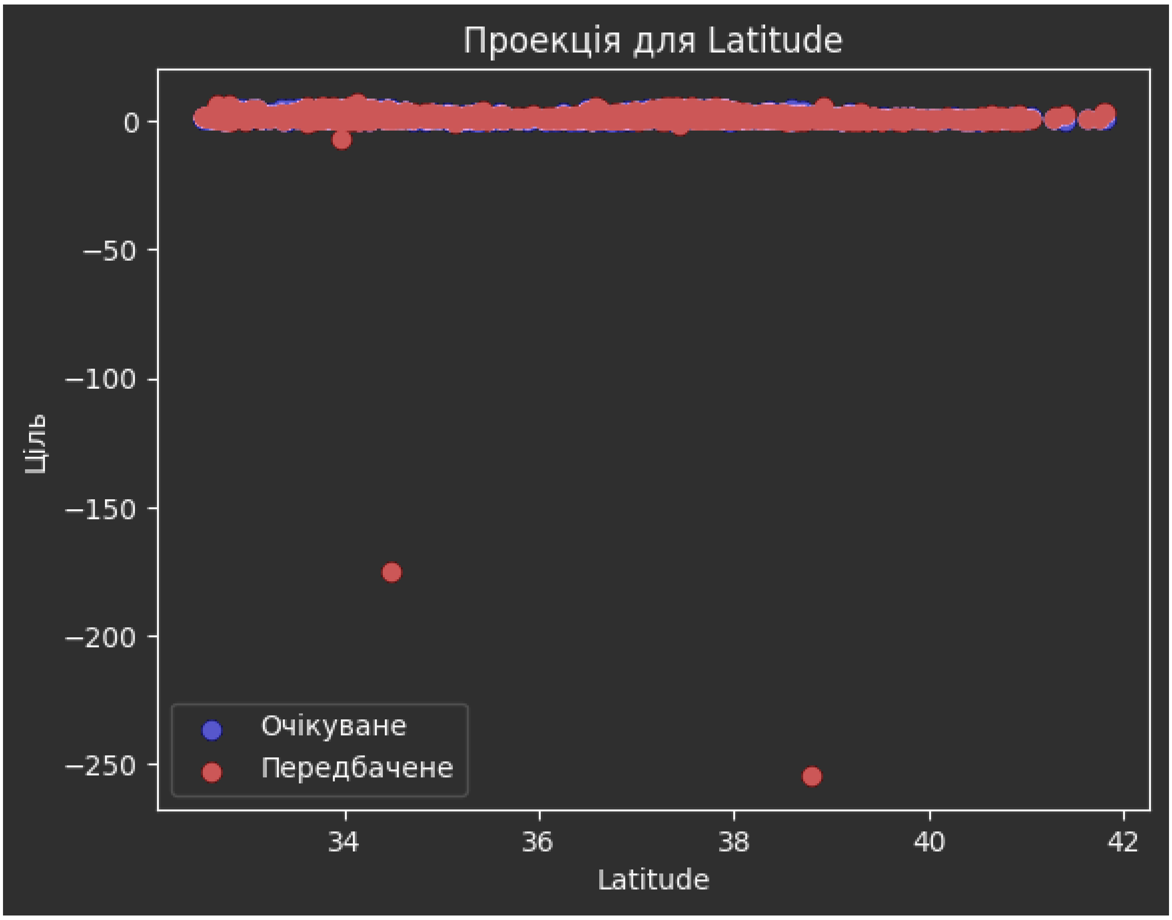
Проекція для AveRooms
0
−50
−100
−150
−200
−250
Ціль
Очікуване
Передбачене
0
20
40
60
80
100
120
AveRooms


Проекція для AveBedrms
Ціль
0
−50
−100
−150
−200
−250
0
5
10
15
20
25
30
35
AveBedrms
Очікуване
Передбачене


Проекція для Population
Ціль
0
−50
−100
−150
−200
−250
0
2000
4000
6000
8000
10000
12000
14000
16000
Population
Очікуване
Передбачене



Проекція для AveOccup
Ціль
0
−50
−100
−150
−200
−250
0
20
40
60
80
AveOccup
Очікуване
Передбачене



Проекція для Latitude
Ціль
0
−50
−100
−150
−200
−250
34
36
38
40
42
Latitude
Очікуване
Передбачене

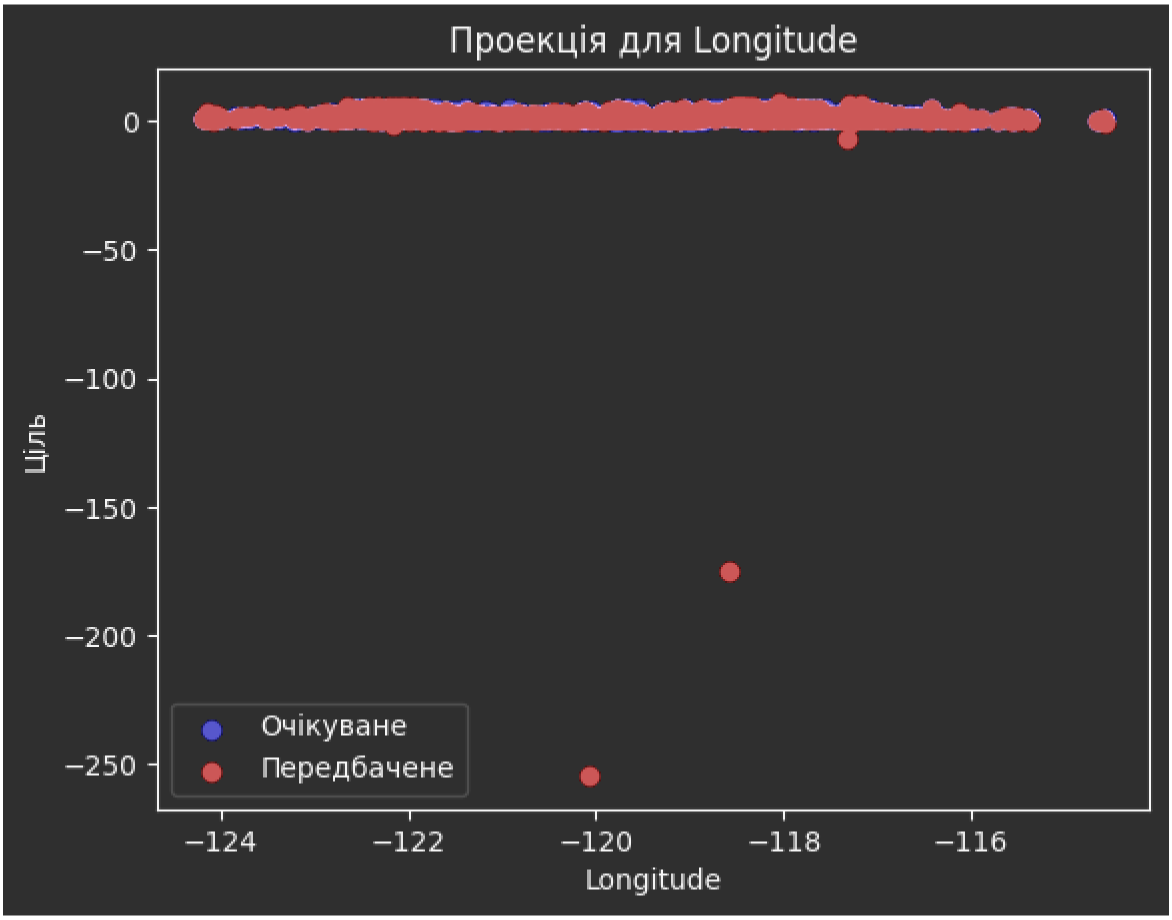


Рис. 6-13 На графіку, що показує співвідношення передбачених та очікуваних значень ознак дата-сету, видно, що більшість передбачених значень близькі до очікуваних. Однак є і похибки, де передбачені значення або не зовсім точно відповідають очікуваним, або взагалі не збігаються з ними.

Висновок: У цій лабораторній роботі я навчилась застосовувати поліномну регресію різних ступенів, ознайомилась з ефектом перетренування (перенасичення) та побачила, як цей ефект впливає на результати тренування моделі на власному прикладі.