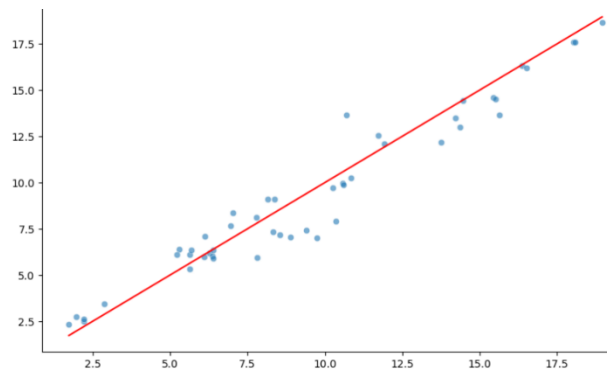


Лабораторна робота № 5. Дослідження методів регресії: Linear Regression, Polynomial Regression, Ridge Regression та Lasso Regression

Мета: сформулювати практичні навички в застосуванні різних типів регресійних моделей для аналізу та передбачення даних. Здобувачі навчатимуться підбирати оптимальний тип регресійної моделі в залежності від специфіки даних та завдання, оцінювати ефективність моделей за допомогою стандартних метрик, а також розуміти та застосовувати методи регуляризації задля покращення загальної якості моделі.

Теоретичні відомості

Лінійна регресія є методом навчання з учителем, який використовується, коли цільова або залежна змінна є **неперервним дійсним числом**. Вона встановлює зв'язок між залежною змінною y та однією або декількома незалежними змінними x за допомогою



«лінії найкращої відповідності». Цей метод працює на принципі методу найменших квадратів (Ordinary Least Squares, OLS) або середньоквадратичної помилки (Mean Square Error, MSE). У статистиці OLS є методом оцінки невідомих параметрів лінійної регресійної функції, його метою є мінімізація суми квадратів різниць між спостережуваними залежними змінними у наданому наборі даних та тими, що передбачені лінійною регресійною функцією.

Основною метою регресійного аналізу є визначення залежності цільової (залежної) змінної від однієї чи декількох предикторних (незалежних) змінних та прогнозування цільової змінної на основі відомих значень незалежних змінних.

Linear Regression (Лінійна регресія) виражає залежність через лінійну функцію: $y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n + \epsilon$, де y – цільова змінна, $x_1, \dots, x_n$ – предиктори, $\beta_0, \dots, \beta_n$ – коефіцієнти моделі, а ϵ – випадкова похибка.

Polynomial Regression (Поліноміальна регресія) розширює лінійну модель, дозволяючи предикторам впливати на цільову змінну нелінійно: $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n + \epsilon$. Це дозволяє моделювати більш складні залежності.

Ridge та **Lasso Regression** (Регресія Ріджа та Лассо) використовуються для регуляризації моделі, що допомагає запобігати перенавчанню, шляхом додавання штрафу до коефіцієнтів моделі.

Ridge Regression мінімізує суму квадратів коефіцієнтів (L2-регуляризація) $L_{Ridge}(\theta) = \sum_{i=1}^n (y_i - \sum_{j=1}^m \theta_j x_{ij})^2 + \alpha \sum_{j=1}^m \theta_j^2$, де $L_{Ridge}(\theta)$ – функція втрат для Ridge Regression, y_i – фактичне значення цільової змінної для i -го спостереження, x_{ij} значення j -ї ознаки для i -го спостереження, θ_j – коефіцієнт моделі для j -ознак, n – кількість спостережень, m – кількість ознак, α – параметр регуляризації.

Lasso Regression сприяє створенню розріджених моделей, додаванням штрафу у вигляді абсолютних значень коефіцієнтів (L1-регуляризація) $L_{Lasso}(\theta) = \sum_{i=1}^n (y_i - \sum_{j=1}^m \theta_j x_{ij})^2 + \alpha \sum_{j=1}^m |\theta_j|$, де $L_{Lasso}(\theta)$ – функція втрат для Lasso Regression. Lasso Regression може призводити до того, що деякі коефіцієнти стануть рівними нулю.

Регуляризація допомагає уникнути перенавчання моделі, шляхом запровадження штрафу за великі значення коефіцієнтів. **Ridge Regression** зменшує розмір коефіцієнтів, але зазвичай не зводить їх до нуля. Натомість, **Lasso Regression** може звести деякі коефіцієнти до нуля, що дозволяє виконувати відбір ознак в ході тренування моделі.

Приклад виконання роботи

Завдання

1. Оберіть та завантажте датасет відповідно до свого варіанту. Завантажте ваш набір даних як тимчасовий файл у середовище Google Colab та ініціалізуйте його.
2. Проаналізуйте датасет та визначте цільову змінну та ознаки, які будуть використовуватися для регресії.
3. Виконайте очищення даних: видаліть або заповніть пропущені значення, опрацюйте викиди якщо це потребується. Створіть графіки для розуміння розподілу ознак та їх взаємозв'язку з цільовою змінною.
4. Виконайте тренування моделей: 1) лінійної регресії; 2) поліноміальної регресії; 3) Ridge регресії; 4) Lasso регресії. Виконайте оцінку та візуалізуйте результати за наступними метриками: MSE, R^2 для кожної моделі. Побудуйте графіки фактичних та передбачених значень для кожної моделі.
5. Виконайте прогнозування за допомогою різних моделей для довільних вхідних даних відповідно до вашої цільової змінної.
6. Проаналізуйте отримані результати, обґрунтуйте вибір найкращої моделі для даного набору даних.
7. За бажанням проєкспериментуйте з гіперпараметрами Degree та Alpha для моделей задля покращення результатів.

Контрольне запитання №1



Вкажіть результуюче значення MSE для Polynomial Regression.

Відповідь: _____

Контрольне запитання №2



Вкажіть результуюче значення R^2 для Lasso Regression.

Відповідь: _____

Контрольне запитання №3

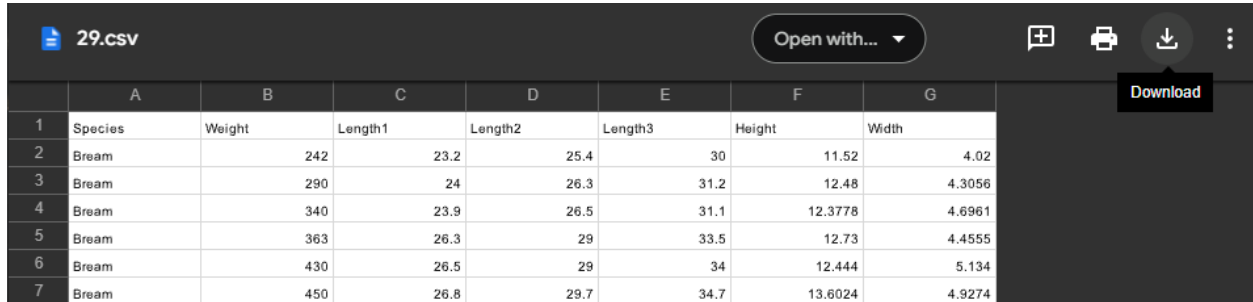


Вкажіть найкраще значення показника R^2 серед усіх моделей.

Відповідь: _____

Хід роботи

1. Збережемо та імпортуємо набір даних за варіантом.



	A	B	C	D	E	F	G
1	Species	Weight	Length1	Length2	Length3	Height	Width
2	Bream	242	23.2	25.4	30	11.52	4.02
3	Bream	290	24	26.3	31.2	12.48	4.3056
4	Bream	340	23.9	26.5	31.1	12.3778	4.6961
5	Bream	363	26.3	29	33.5	12.73	4.4555
6	Bream	430	26.5	29	34	12.444	5.134
7	Bream	450	26.8	29.7	34.7	13.6024	4.9274

Рисунок 4.1. Датасет на Google Drive

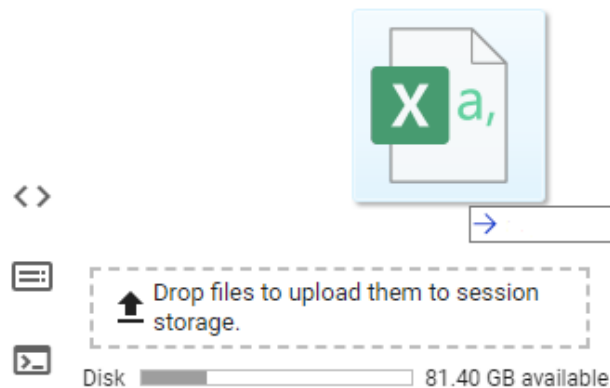


Рисунок 4.2. Завантаження датасету до Google Colab

2. Встановлення необхідних бібліотек та модулів

```
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.pipeline import make_pipeline
```

3. Ініціалізація набору даних в Google Colab.

```
# Завдання №1. # Завантаження датасету
data = pd.read_csv("/content/29.csv")
data.head()
```

	Species	Weight	Length1	Length2	Length3	Height	Width
0	Bream	242.0	23.2	25.4	30.0	11.5200	4.0200
1	Bream	290.0	24.0	26.3	31.2	12.4800	4.3056
2	Bream	340.0	23.9	26.5	31.1	12.3778	4.6961
3	Bream	363.0	26.3	29.0	33.5	12.7300	4.4555
4	Bream	430.0	26.5	29.0	34.0	12.4440	5.1340

Рисунок 4.3. Перші 5 рядків набору даних

4. Визначимо цільову змінну, як Height, ознаками будуть: Weight, Length1, Length2, Length3, Width.

5. Виконаємо очищення даних

```
# Перевірка на відсутні значення
print(data.isnull().sum())
```

```
Species      0
Weight       0
Length1      0
Length2      0
Length3      0
Height       0
Width        0
dtype: int64
```

Рисунок 4.4. Пошук нульових значень у датасеті

Порожні значення відсутні, тож видалення або заповнення виконувати не треба.

```
# Видалення або заповнення відсутніх значень
# data.dropna(inplace=True) # Видалення

# data.fillna(data.mean(), inplace=True) # Заповнення середніми значеннями
```

Виконаємо візуалізацію однієї з ознак набору даних, а саме «Length2»

```
# Візуалізація викидів
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['Length2'])
plt.title('Boxplot before removing outliers')
plt.show()
```

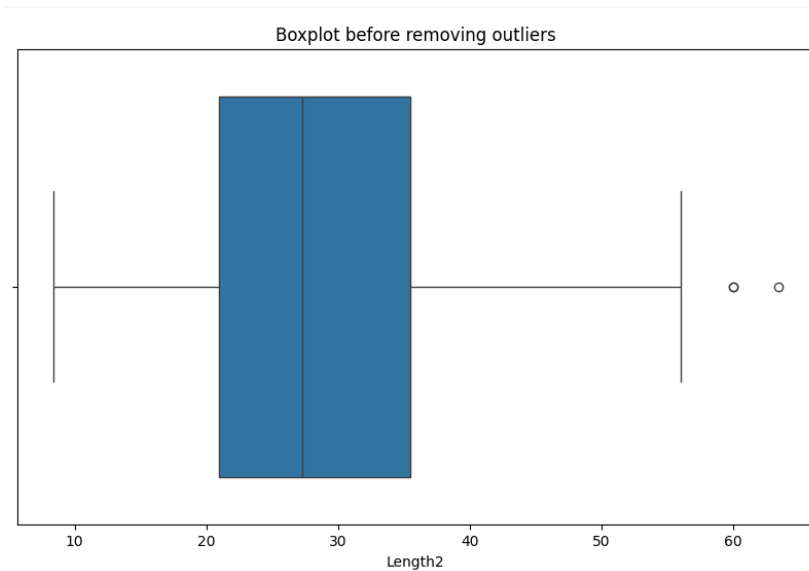


Рисунок 4.5. Діаграма викидів для Length2 перед очищенням

Виконаємо очищення викидів для «Length2».

```
# Видалення викидів
feature_processing = 'Length2'
q_low = data[feature_processing].quantile(0.01)
q_hi = data[feature_processing].quantile(0.99)
data = data[(data[feature_processing] < q_hi) & (data[feature_processing]
> q_low)]
```

```
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['Length2'])
plt.title('Boxplot after removing outliers')
plt.show()
```

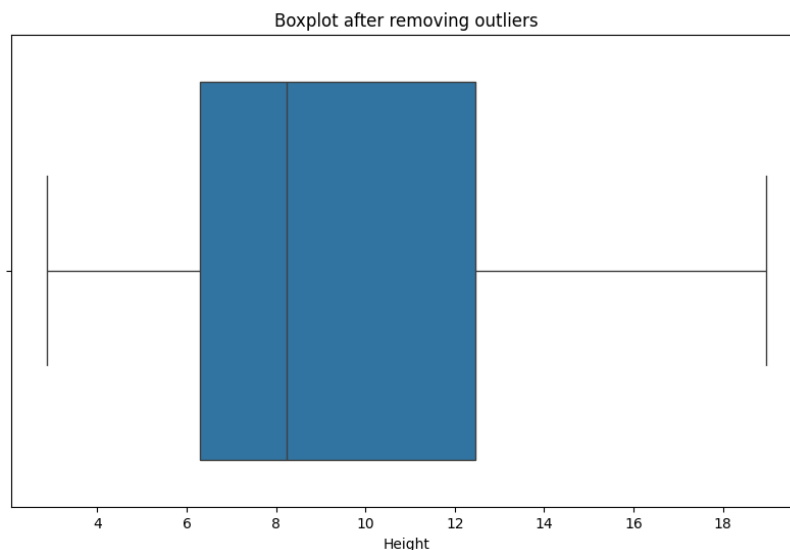


Рисунок 4.6. Діаграма викидів для Length2 після очищення

Подібні операції виконаємо для інших ознак.

6. Розподілимо дані на тренувальні та тестові набори з урахуванням ознак та цільової змінної

```
# Завдання №2. Застосуємо функції активації SiLU, Soft, Plus, PReLU
# Розділення даних на тренувальні та тестові набори
features = ['Weight', 'Length1', 'Length2', 'Length3', 'Width'] # ознаки
X = data[features]
y = data['Height'] # цільові змінні

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

7. Візуалізуємо взаємозв'язки між ознаками та цільовою змінною

```
# Візуалізація взаємозв'язку між ознаками та цільовою змінною
sns.pairplot(data, x_vars=features, y_vars='Height', kind='reg')
plt.show()
```

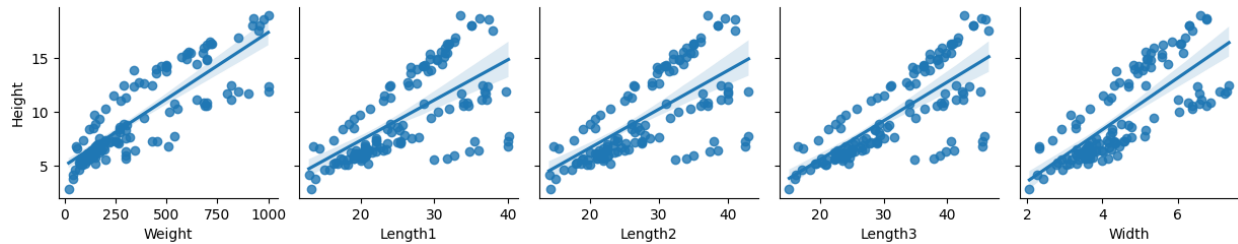


Рисунок 4.7. Візуалізація взаємозв'язку між ознаками та цільовою змінною

```
# Візуалізація кореляції між ознаками
sns.heatmap(data[features + ['Height']].corr(), annot=True, fmt=".2f")
plt.show()
```

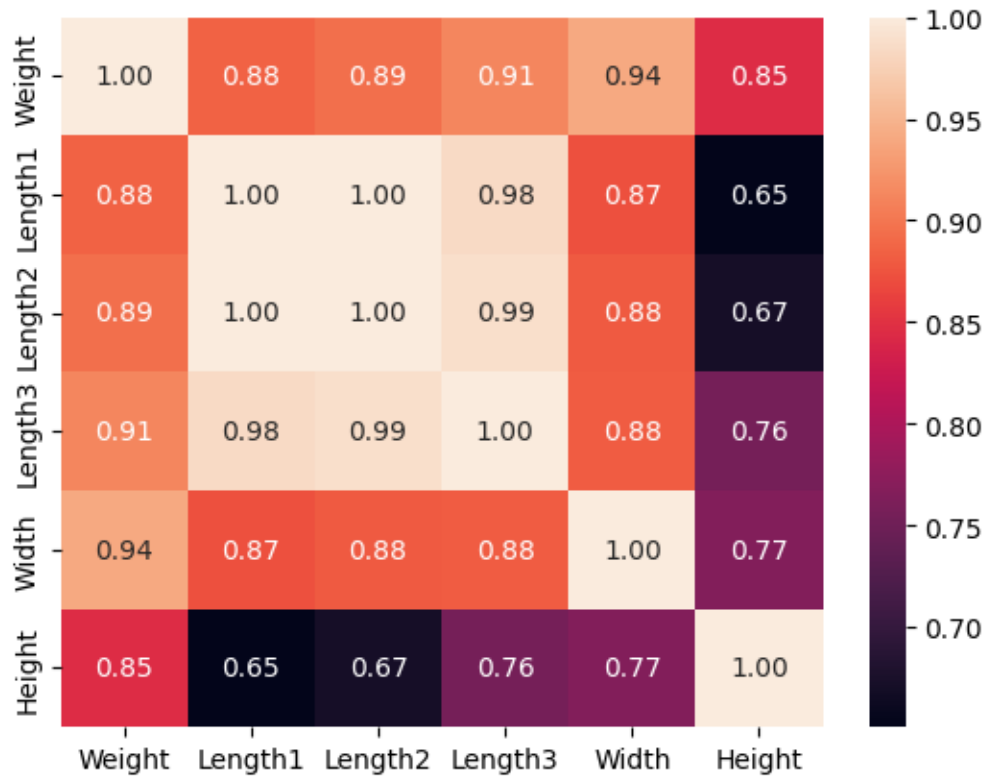


Рисунок 4.8. Візуалізація кореляції між ознаками

8. Виконаємо тренування моделей Linear Regression, Polynomial Regression, Ridge Regression, Lasso Regression та їх оцінку за показниками MSE та R^2 .

```
# Linear Regression
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)

# Polynomial Regression
polynomial_model = make_pipeline(PolynomialFeatures(degree=2),
LinearRegression())
polynomial_model.fit(X_train, y_train)

# Ridge Regression
ridge_model = Ridge(alpha=1.0)
ridge_model.fit(X_train, y_train)

# Lasso Regression
lasso_model = Lasso(alpha=0.1)
lasso_model.fit(X_train, y_train)
```

9. Виконаємо оцінювання моделей за показниками MSE та R^2

```
# Оцінювання моделей
models = [linear_model, polynomial_model, ridge_model, lasso_model]
model_names = ['Linear Regression', 'Polynomial Regression', 'Ridge
Regression', 'Lasso Regression']

for model, name in zip(models, model_names):
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    print(f"{name}: MSE = {mse:.2f}, R^2 = {r2:.2f}")
```

```
Linear Regression: MSE = 1.15, R^2 = 0.93
Polynomial Regression: MSE = 0.92, R^2 = 0.94
Ridge Regression: MSE = 1.11, R^2 = 0.93
Lasso Regression: MSE = 1.19, R^2 = 0.92
```

Рисунок 4.9. Оцінка моделей регресії за показниками MSE та R^2

10. Виконаємо візуалізацію графіків моделей регресії.

```
import ipywidgets as widgets
from IPython.display import display

def plot_model_results(model_name):
    if model_name == 'Linear Regression':
        model = linear_model
    elif model_name == 'Polynomial Regression':
        model = polynomial_model
    elif model_name == 'Ridge Regression':
        model = ridge_model
    elif model_name == 'Lasso Regression':
        model = lasso_model
    else:
        print("Модель не вибрана")
        return

    y_pred = model.predict(X_test)

    plt.figure(figsize=(10, 6))
    sns.scatterplot(x=y_test, y=y_pred, alpha=0.6)
    sns.lineplot(x=y_test, y=y_test, color='red')
    plt.xlabel('Actual Heights')
    plt.ylabel('Predicted Heights')
    plt.title(f'Actual vs Predicted Heights: {model_name}')
    plt.show()
```

```
# Створення випадкового списку для вибору моделі
model_dropdown = widgets Dropdown(
    options=['Linear Regression', 'Polynomial Regression', 'Ridge
Regression', 'Lasso Regression'],
    value='Linear Regression',
    description='Model:',
    disabled=False,
)

# Створення віджету для інтерактивного відображення графіків
interactive_plot = widgets.interactive(plot_model_results,
model_name=model_dropdown)
display(interactive_plot)
```

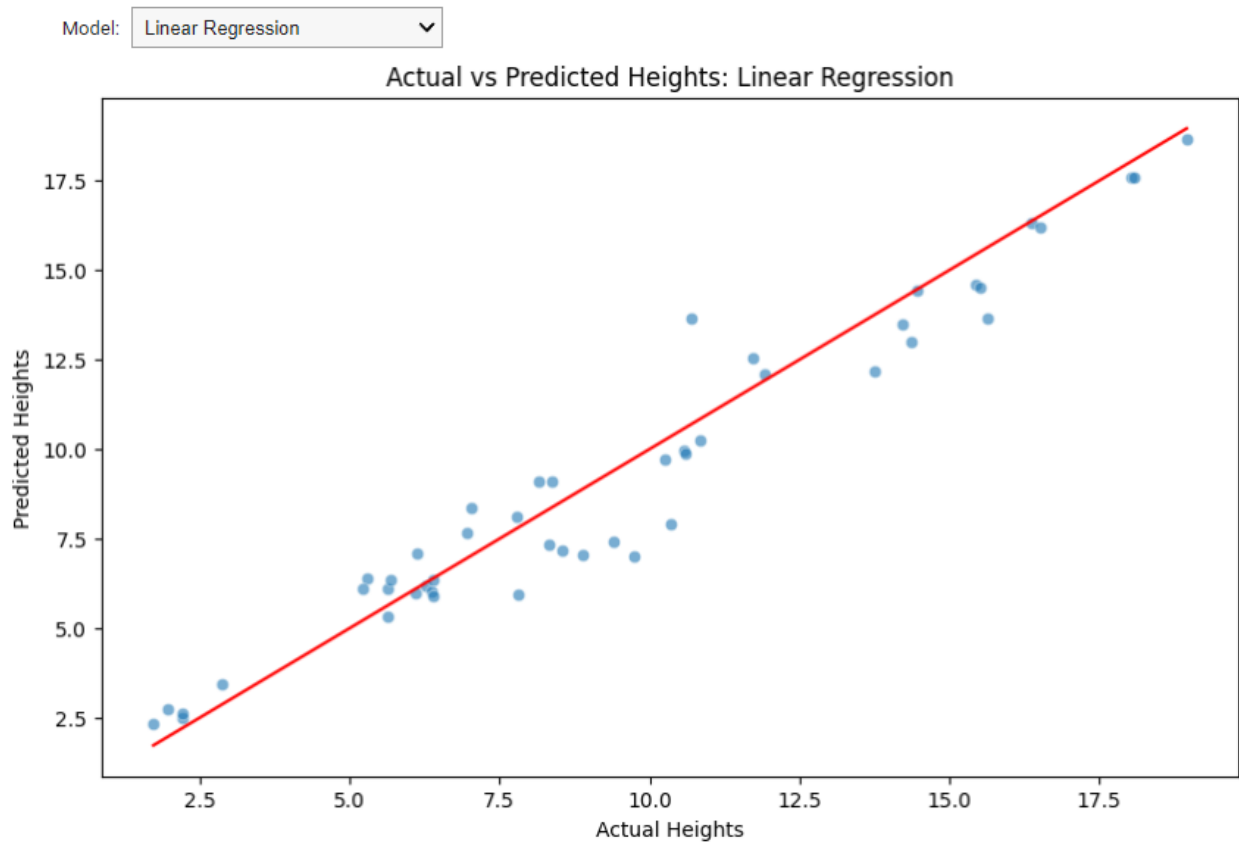


Рисунок 4.10. Графік реальних та прогнозованих значень Height за допомогою моделі Linear Regression

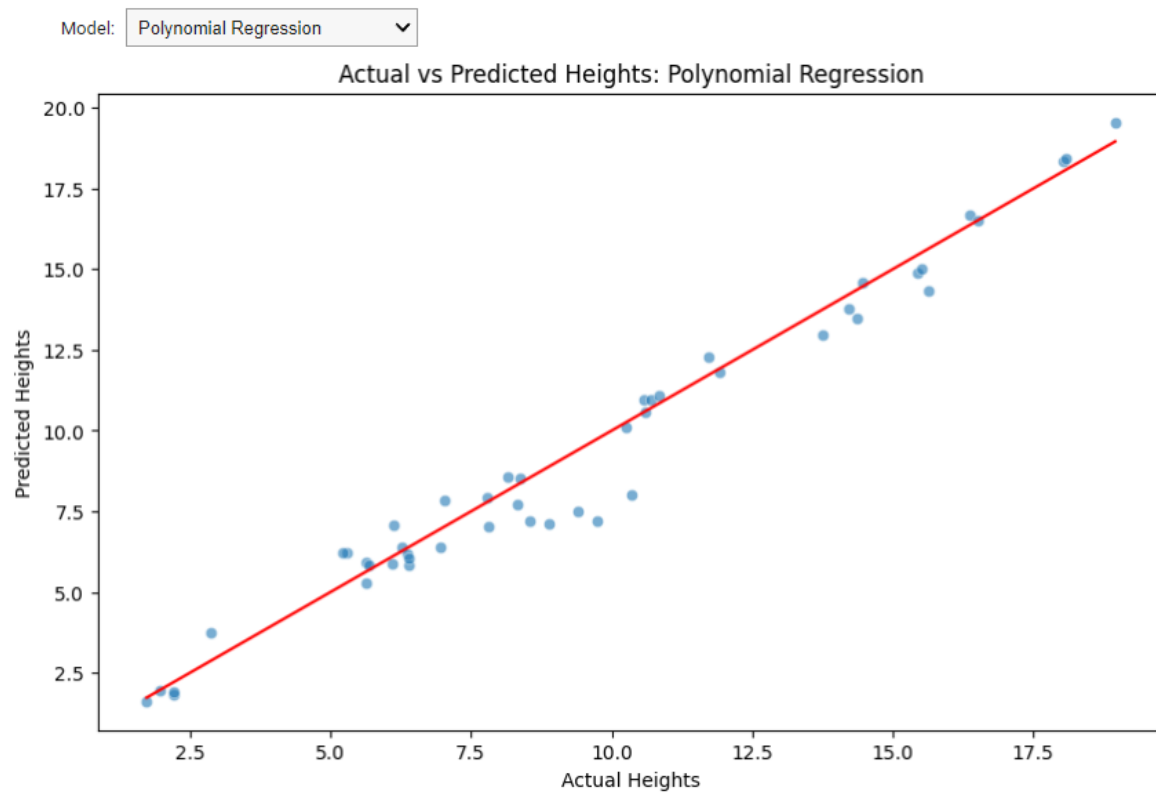


Рисунок 4.11. Графік реальних та прогнозованих значень Height за допомогою моделі Polynomial Regression

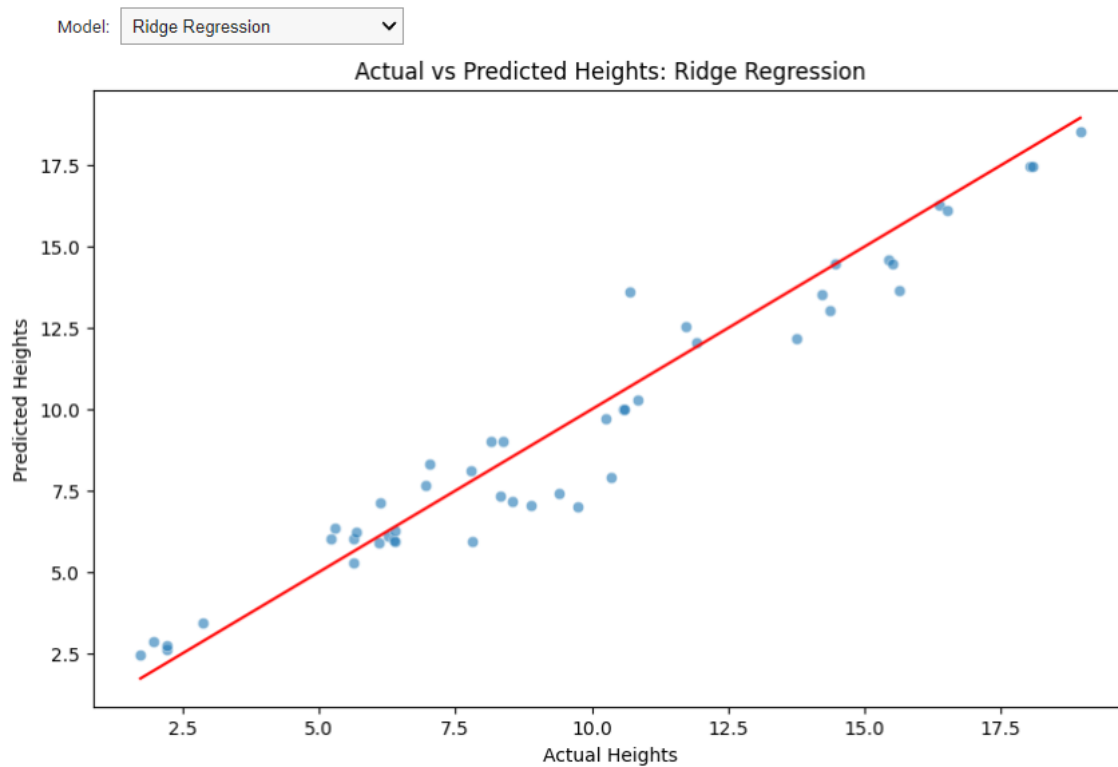


Рисунок 4.12. Графік реальних та прогнозованих значень Height за допомогою моделі Ridge Regression

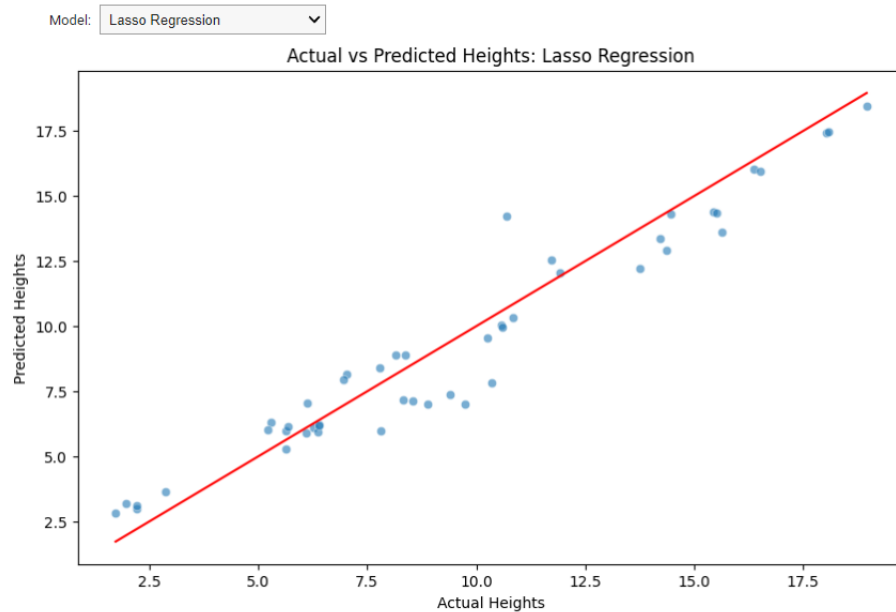


Рисунок 4.13. Графік реальних та прогнозованих значень Height за допомогою моделі Lasso Regression

11. Виконаємо прогнозування на довільних даних

```
new_X = [70, 16, 17, 18, 3] # Приклад нового значення ...

predictions = {title: model.predict([new_X]) for model, title in
zip(models, model_names)}

for title, pred in predictions.items():
    print(f"{title} predicted heights: {pred[0]}")
```

```
Linear Regression predicted heights: 4.2153492851751775
Polynomial Regression predicted heights: 3.8886067868371175
Ridge Regression predicted heights: 4.332190347354458
Lasso Regression predicted heights: 4.5056590783654045
```

Рисунок 4.14. Результати прогнозування різних моделей регресії

12. Виконаємо тестування гіперпараметрів Degree та Alpha

```
# Додавання віджетів для налаштування гіперпараметрів
degree_selector = widgets.IntSlider(min=1, max=5, value=2,
description='Degree:')
alpha_selector = widgets.FloatSlider(min=0.01, max=1, value=0.1,
step=0.01, description='Alpha:')
model_selector = widgets.Dropdown(options=model_names,
description='Model:')
```

```

# Функція для тренування моделей з різними налаштуваннями
def train_with_parameters(degree, alpha, model_name):
    if model_name == 'Polynomial Regression':
        model = make_pipeline(PolynomialFeatures(degree=degree),
LinearRegression())
    elif model_name == 'Ridge Regression':
        model = Ridge(alpha=alpha)
    elif model_name == 'Lasso Regression':
        model = Lasso(alpha=alpha)
    else: # Linear Regression
        model = LinearRegression()

    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    print(f"{model_name}: MSE = {mse:.2f}, R^2 = {r2:.2f}")

# Візуалізація результатів
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred, alpha=0.6)
sns.lineplot(x=y_test, y=y_test, color='red')
plt.xlabel('Actual Heights')
plt.ylabel('Predicted Heights')
plt.title(f'{model_name} Results')
plt.show()

widgets.interactive(train_with_parameters, degree=degree_selector,
alpha=alpha_selector, model_name=model_selector)

```

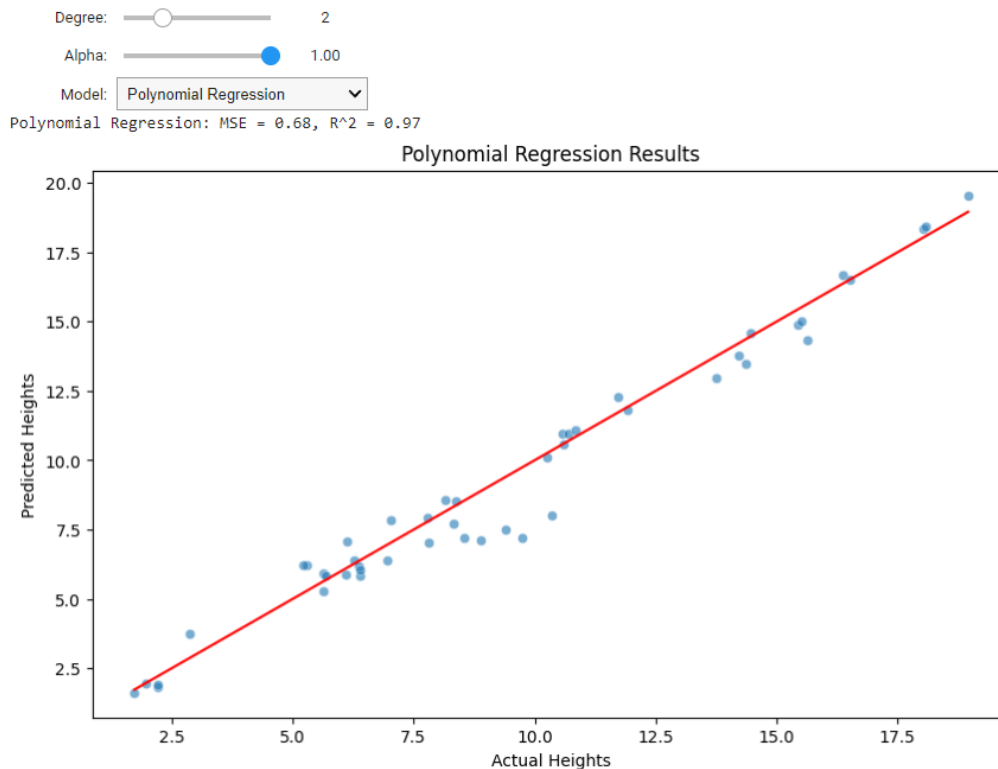


Рисунок 4.15. Можливості для кастомізації гіперпараметрів

Таким чином, аналізуючи отримані дані, можна зазначити, що використання моделі Polynomial Regression продемонструвало найкращі результати MSE (0.92) та R² (0.94). Моделі Linear Regression, Ridge Regression та Lasso Regression надали близькі один до одного результати за показниками MSE та R², що також є високими та дозволяють виконувати задачу регресії для даного набору даних.

Налаштування гіперпараметрів Degree та Alpha дозволило досягти результатів MSE (0.68), R² (0.97) для моделі Polynomial Regression, тобто результати регресії були покращені.

Контрольне запитання №1



Вкажіть результуюче значення MSE для Polynomial Regression.

Відповідь: **0.92**

Контрольне запитання №2



Вкажіть результуюче значення R² для Lasso Regression.

Відповідь: **0.92**



Контрольне запитання №3

Вкажіть найкраще значення показника R^2 серед усіх моделей.

Відповідь: **0.94**

Висновки: ...

Завдання для виконання лабораторної роботи

1. Оберіть та завантажте датасет відповідно до свого варіанту. Завантажте ваш набір даних як тимчасовий файл у середовище Google Colab та ініціалізуйте його.

Посилання на датасети					
1	6	11	16	21	26
2	7	12	17	22	27
3	8	13	18	23	28
4	9	14	19	24	29
5	10	15	20	25	30

2. Проаналізуйте датасет та визначте цільову змінну та ознаки, які будуть використовуватися для регресії.
3. Виконайте очищення даних: видаліть або заповніть пропущені значення, опрацюйте викиди якщо це потребується. Створіть графіки для розуміння розподілу ознак та їх взаємозв'язку з цільовою змінною.
4. Виконайте тренування моделей: 1) лінійної регресії; 2) поліноміальної регресії; 3) Ridge регресії; 4) Lasso регресії. Виконайте оцінку та візуалізуйте результати за наступними метриками: MSE, R^2 для кожної моделі. Побудуйте графіки фактичних та передбачених значень для кожної моделі.
5. Виконайте прогнозування за допомогою різних моделей для довільних вхідних даних відповідно до вашої цільової змінної.
6. Проаналізуйте отримані результати, обґрунтуйте вибір найкращої моделі для даного набору даних.
7. За бажанням проєкспериментуйте з гіперпараметрами Degree та Alpha для моделей задля покращення результатів.

Контрольне запитання №1



Вкажіть результуюче значення MSE для Polynomial Regression.

Відповідь: _____

Контрольне запитання №2



Вкажіть результуюче значення R^2 для Lasso Regression.

Відповідь: _____

Контрольне запитання №3



Вкажіть найкраще значення показника R^2 серед усіх моделей.

Відповідь: _____



Welcome to Google Colab

[https://colab.research.google.com/drive/1w2OST7DJyDYcgeiFeT4ZIPmdr9zS2Wh-
?usp=sharing](https://colab.research.google.com/drive/1w2OST7DJyDYcgeiFeT4ZIPmdr9zS2Wh-?usp=sharing)