Understanding Linear Regression

By Anam Riaz

Introduction to Linear Regression

- • Predicts a dependent variable (Y) using an independent variable (X).
- • Assumes a linear relationship between X and Y.
- • Equation: $Y = mX + c$

- Key Components:
- • m: Slope (rate of change)
- • c: Intercept (value of Y when $X = 0$)

Real-Life Applications

- • Predicting sales based on advertising spend.
- • Estimating house prices using square footage.
- • Analyzing the relationship between study hours and exam scores.

Example 1: Sales vs. Advertising Spend

- Advertising Spend (X) | Sales (Y) |

Advertising Spend (X)	Sales (Y)
1	2
2	4
3	5
4	4.5
5	6

- Objective: Find the equation $Y = mX + c$ and predict sales for $X = 6$.

Solution for Example 1 (Part 1)

- Formulas:
- $m = [n\sum(XY) - \sum(X)\sum(Y)] / [n\sum(X\text{^}2) - (\sum(X))\text{^}2]$
- $c = [\sum(Y) - m\sum(X)] / n$

- Values:
- $\sum(X) = 15$, $\sum(Y) = 21.5$, $\sum(XY) = 73$, $\sum(X\text{^}2) = 55$

$$m = \frac{n\sum(XY) - \sum(X)\sum(Y)}{n\sum(X^2) - (\sum(X))^2}$$

$$c = \frac{\sum(Y) - m\sum(X)}{n}$$

Solution for Example 1 (Part 2)

- Slope (m):
- m = [5(73) - (15)(21.5)] / [5(55) - (15)^2] = 0.85

- Intercept (c):
- c = [21.5 - (0.5)(15)] / 5 = 1.75

- Equation:
- Y = 0.85X + 1.75

- Prediction for X = 6:
- Y = 0.85(6) + 175 = 6.885

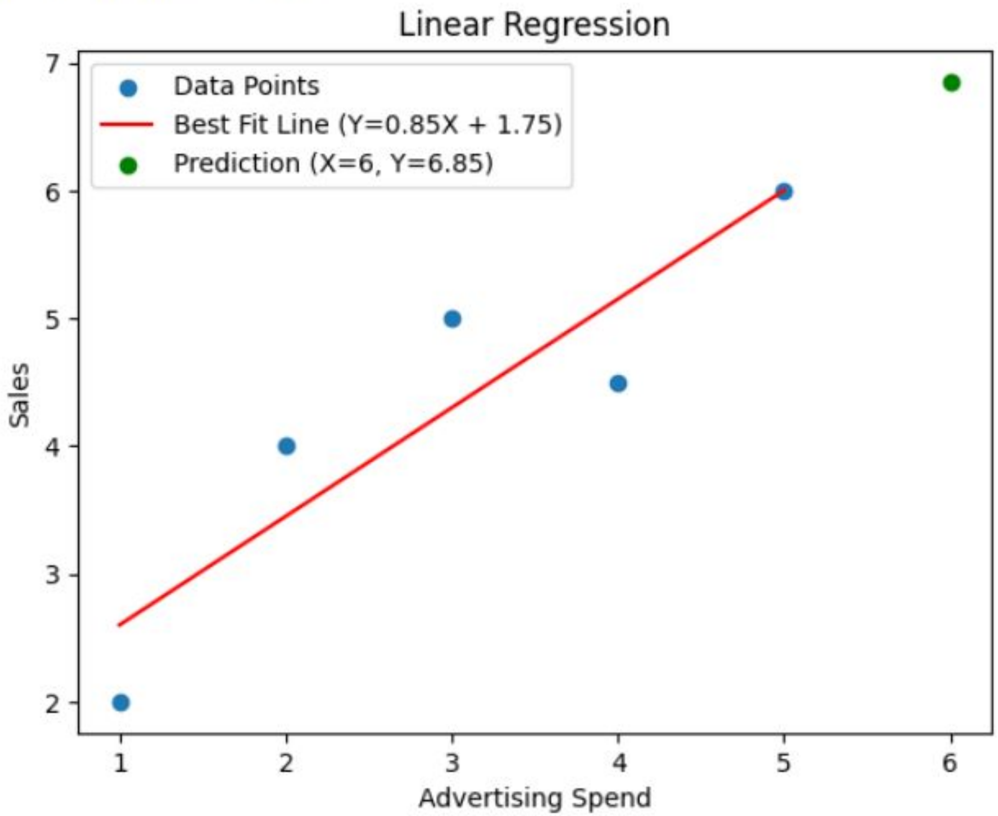
Python Implementation (Example 1)

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

# Data
X = np.array([1, 2, 3, 4, 5])
Y = np.array([2, 4, 5, 4.5, 6])

# Calculate slope (m) and intercept (c)
n = len(X)
m = (n * np.sum(X * Y) - np.sum(X) * np.sum(Y)) / (n * np.sum(X**2) - (np.sum(X))**2)
c = (np.sum(Y) - m * np.sum(X)) / n

# Prediction for X = 6
X_new = 6
Y_new = m * X_new + c

print(f"Equation: Y = {m:.2f}X + {c:.2f}")
print(f"Prediction for X=6: {Y_new:.2f}")

# Plot
plt.scatter(X, Y, label='Data Points')
plt.plot(X, m * X + c, color='red', label=f'Best Fit Line (Y={m:.2f}X + {c:.2f})')
plt.scatter(X_new, Y_new, color='green', label=f'Prediction (X=6, Y={Y_new:.2f})')
plt.legend()
plt.xlabel('Advertising Spend')
plt.ylabel('Sales')
plt.title('Linear Regression')
plt.show()
```

```
Equation: Y = 0.85X + 1.75
Prediction for X=6: 6.85
```

Python Implementation (Example 1)

```
Equation: Y = 0.85X + 1.75
Prediction for X=6: 6.85
```



Example 2: House Prices

- | Square Footage (X) | Price ($1000) (Y) |

Square Footage (X)	Price ($1000)(Y)
800	200
1000	250
1200	300
1500	350
1800	400

- Objective: Predict the price of a house with X = 1600.

Solution Example 2

Slope (m):

$$m = \frac{5(1,200,000) - (6,300)(1,500)}{5(8,300,000) - (6,300)^2}$$

Intercept (c):

$$c = \frac{\sum(Y) - m\sum(X)}{n}$$

Equation:

$$Y = 0.25X + 0.5$$

Python Implementation (Example 2)

```python
# Data
X = np.array([800, 1000, 1200, 1500, 1800])
Y = np.array([200, 250, 300, 350, 400])

# Calculate slope and intercept
n = len(X)
m = (n * np.sum(X * Y) - np.sum(X) * np.sum(Y)) / (n * np.sum(X**2) - (np.sum(X))**2)
c = (np.sum(Y) - m * np.sum(X)) / n

# Prediction for X = 1600
X_new = 1600
Y_new = m * X_new + c

print(f"Equation: Y = {m:.2f}X + {c:.2f}")
print(f"Prediction for X=1600: {Y_new:.2f}")

# Plot
plt.scatter(X, Y, label='Data Points')
plt.plot(X, m * X + c, color='red', label=f'Best Fit Line (Y={m:.2f}X + {c:.2f})')
plt.scatter(X_new, Y_new, color='green', label=f'Prediction (X=1600, Y={Y_new:.2f})')
plt.legend()
plt.xlabel('Square Footage')
plt.ylabel('Price ($1000)')
plt.title('Linear Regression')
plt.show()
```

Python Implementation (Example 2)

```
Equation: Y = 0.20X + 50.79
Prediction for X=1600: 367.25
```