Training Day 18 Report

Date: 15 July 2025
Topic: Polynomial Regression

Overview

Today's session covered **Polynomial Regression**, which is used when data shows a **non-linear relationship**. Unlike Linear Regression (straight line), Polynomial Regression fits a **curved line** to better capture patterns.

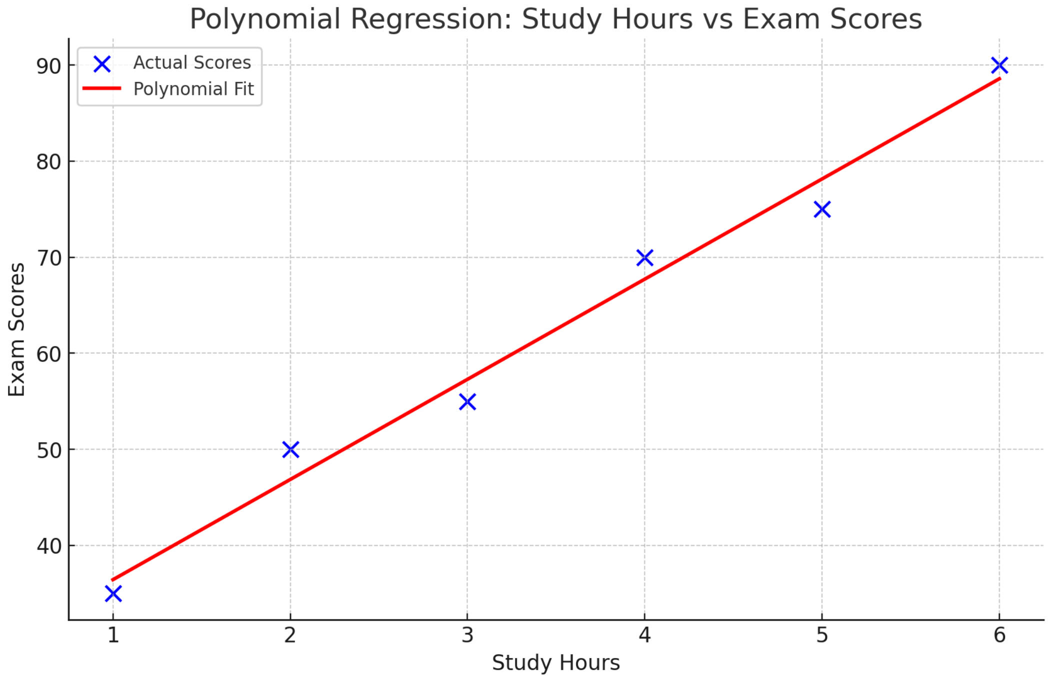
Key Concepts

- **Linear Regression** → straight line fit.
- **Polynomial Regression** → curved line fit.
- Equation:
  y=b0+b1x+b2x2+⋯+bnxny = b_0 + b_1x + b_2x^2 + \dots + b_nx^n

Code Example

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

# Data: Study Hours vs Scores
X = np.array([1, 2, 3, 4, 5, 6]).reshape(-1, 1)
y = np.array([35, 50, 55, 70, 75, 90])

# Polynomial Transformation
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

# Train Model
model = LinearRegression()
model.fit(X_poly, y)

# Predict
y_pred = model.predict(X_poly)

# Plot
```

```
plt.scatter(X, y, color="blue", label="Actual")
plt.plot(X, y_pred, color="red", label="Polynomial Fit")
plt.xlabel("Study Hours")
plt.ylabel("Exam Scores")
plt.title("Polynomial Regression")
plt.legend()
plt.show()
```



Learning Outcome

- Learned the difference between **Linear** and **Polynomial Regression**.
- Understood how to fit a **curve** to data.
- Visualized that Polynomial Regression captures non-linear patterns better than a straight line.