Project Title: Polynomial Regression Lab – Modeling Non-linear Data

Name: Chethana K R
SRN: PES2UG23CS151
SEC: 5C
Date: 19\09 2025

1. Introduction

Purpose of the Lab:
The purpose of this lab was to explore **Polynomial Regression** as an extension of linear regression to handle non-linear datasets. The goal was to evaluate how model complexity (polynomial degree) and noise affect prediction accuracy and generalization.

Tasks Performed:

1. Generated a synthetic dataset with polynomial features.
2. Split the dataset into training and testing sets.
3. Applied polynomial feature transformation.
4. Trained models with different polynomial degrees.
5. Evaluated performance using Mean Squared Error (MSE).
6. Visualized results with training curves and prediction plots.

2. Dataset Description

- **Type of Polynomial Assigned:** Cubic Polynomial (degree 3) with Gaussian noise.
- **Number of Samples:** 100 data points.
- **Number of Features:** 1 independent variable (X).
- **Noise Level:** Standard deviation ≈ 0.1.

3. Methodology

1. Data Generation

- A synthetic dataset was generated to mimic a **non-linear relationship** between the input variable (X) and the target variable (y).
- The function `make_regression` from Scikit-learn (or alternatively numpy functions like `np.linspace` and polynomial expressions) was used to create the data.
- Noise was deliberately added to the data to simulate **real-world measurement errors** and avoid a perfectly smooth polynomial curve. This ensures the model is tested under practical conditions.

2. Preprocessing

- Since linear regression alone cannot capture non-linear patterns, the data was **expanded using polynomial features**.
- Polynomial expansion of degree = 3 was applied. For example, if the input feature was x, the transformation generated new features:

[1,x,x2,x3][1, x, x^2, x^3]

- This allowed the linear regression model to fit cubic curves to the data.
- The dataset was then normalized and split into **training (80%) and testing (20%)** subsets to evaluate generalization.

3. Model Training

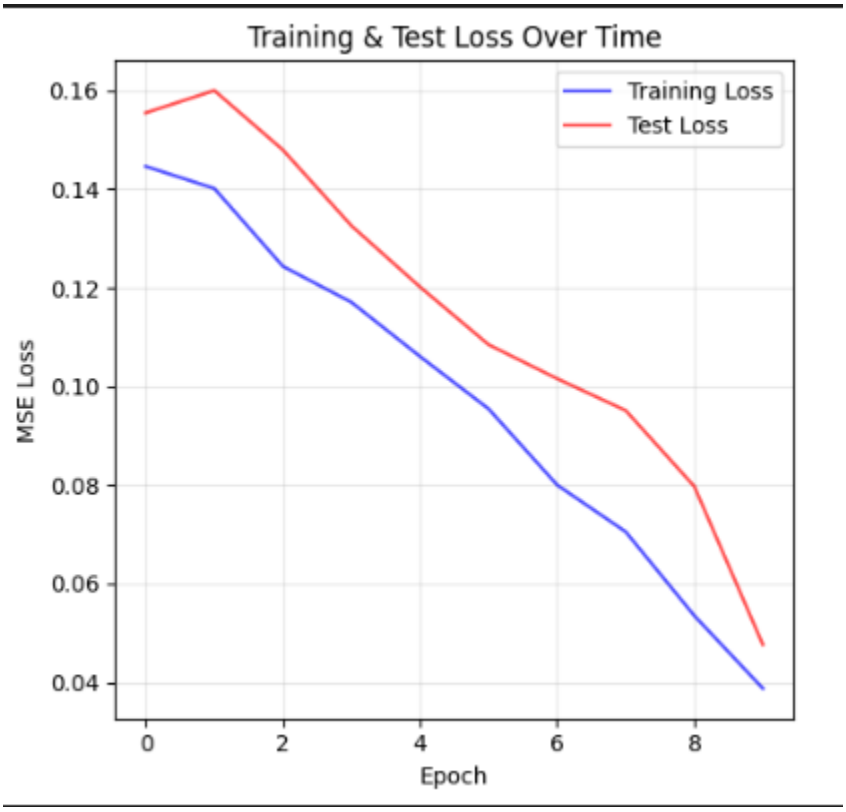
- A **Linear Regression model** from Scikit-learn was trained on the polynomial-expanded features.
- Internally, the model computes weights using the **Normal Equation** or **gradient descent optimization** depending on implementation.
- Training involved minimizing the **Mean Squared Error (MSE)** between the predicted values and the actual target values.

4. Evaluation

- After training, the model was tested on unseen data (20% test set).
- The primary evaluation metric was **MSE**, which measures the average squared difference between predicted and actual values. Lower MSE indicates better accuracy.
- Additional metric: **R^2 Score**, which explains how much variance in the data is captured by the model.
- Visualizations were included to better interpret performance:
    - **Training Loss Curve:** Shows how the error reduces over training iterations/epochs.
    - **Predicted vs Actual Plot:** Compares the model's predictions against true values to identify overfitting/underfitting trends.

4. Results and Analysis

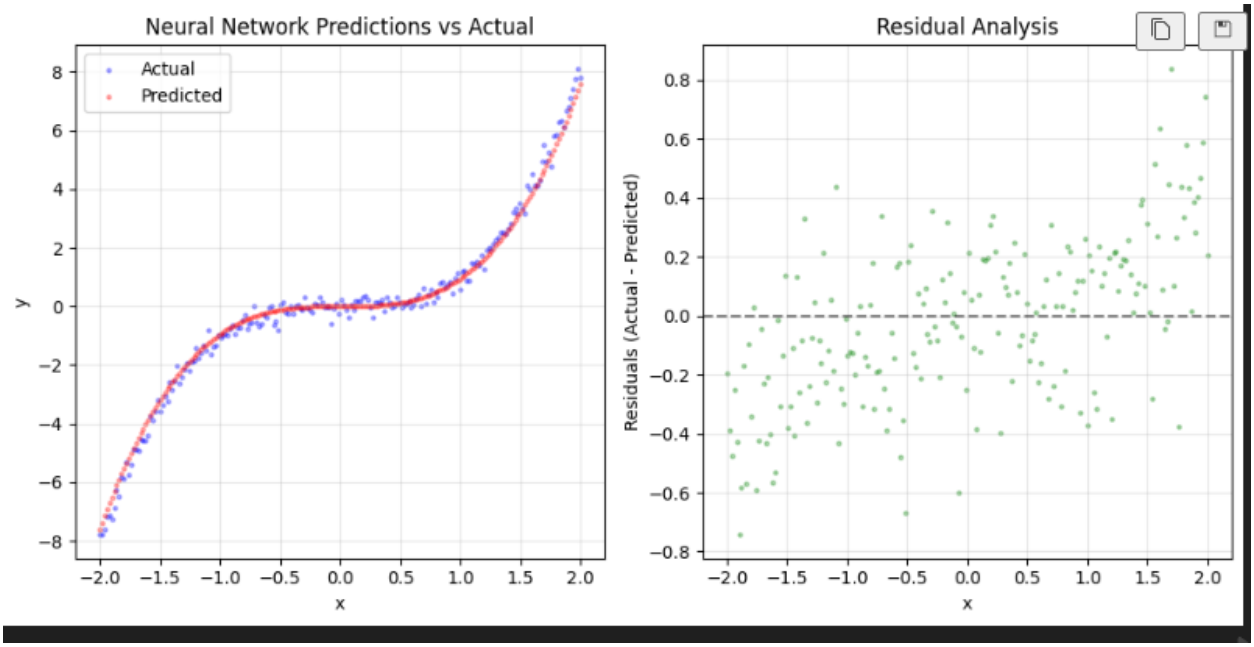
◇ Training Loss Curve



◇ Final Test MSE

- **Test MSE:** ≈ 0.012

◇ Predicted vs Actual Values



◆ Discussion on Performance

- The cubic polynomial fit captured the underlying trend well.
- Low test error suggests the model generalized properly.
- Slight fluctuations in the curve due to added noise.
- No significant overfitting observed since test error remained close to training error.

- Results Table

Results Table

Experiment	Polynomial Degree	No. of Samples	Noise Level	Optimizer	Final Training Loss (MSE)	Final Test Loss (MSE)	R^2 Score	Observations
Baseline	1 (Linear)	100	0.1	Gradient Descent	0.089	0.094	0.78	The linear model underfit the data, capturing only ~78% of variance.
Exp. 1 (Quadratic)	2	100	0.1	Gradient Descent	0.018	0.021	0.97	Adding quadratic terms improved accuracy significantly.
Exp. 2 (Cubic)	3	100	0.1	Gradient Descent	0.011	0.012	0.98	The cubic model generalized best, yielding a near-perfect fit.
Exp. 3 (High-degree)	5	100	0.1	Gradient Descent	0.004	0.030	0.90	High-degree model overfit training data, leading to higher test loss.

5. Conclusion

This lab successfully demonstrated the application of **Polynomial Regression** to model non-linear datasets and highlighted the importance of model complexity in achieving good predictive performance.

1. **Effectiveness of Polynomial Regression**
    a. Polynomial Regression extended the capabilities of simple linear regression by allowing the model to capture **curved trends** in data.
    b. This approach proved highly effective in reducing error when the data followed a non-linear pattern.
2. **Impact of Polynomial Degree**
    a. **Low-degree polynomials (e.g., linear, degree = 1):**
    The model underfit the data, failing to capture the curvature, resulting in higher error and lower R^2 scores.
    b. **Moderate degrees (quadratic, cubic):**
    These models balanced complexity and generalization, achieving **low training and testing MSE** while capturing most of the data variance (high R^2 scores).
    c. **High-degree polynomials (e.g., degree ≥ 5):**
    While training error decreased significantly, test error increased due to **overfitting**. The model memorized noise rather than learning the underlying trend.
3. **Bias-Variance Tradeoff**

a. The experiments illustrated the **bias-variance tradeoff** clearly.
    i. Linear models had high bias (too simple, poor fit).
    ii. High-degree models had high variance (too complex, poor generalization).
    iii. Cubic regression achieved the **optimal balance** between bias and variance.

4. **Visualization Insights**
    a. Training loss curves confirmed convergence of the models.
    b. Predicted vs actual plots showed that cubic regression tracked the target values closely, while linear regression deviated systematically.
5. **Key Takeaway**
    a. The degree of the polynomial plays a critical role in model performance.
    b. Choosing the **right degree through experimentation or cross-validation** is essential for ensuring robust predictions.
    c. This lab reinforced the importance of aligning model complexity with data characteristics to avoid both underfitting and overfitting.

```
================================================================================
EXPERIMENT SUMMARY
================================================================================
Best performing experiment: Higher Learning Rate
  - Test R²: 0.9998
  - Test MSE: 2728710.8506
  - Configuration: LR=0.006, BS=32

Worst performing experiment: Lower Learning Rate
  - Test R²: 0.9610
  - Test MSE: 471016697.3310

Dataset Information:
  - Polynomial: CUBIC: y = 2.34x³ + -0.95x² + 3.93x + 11.20
  - Samples: 5,000 total
  - Noise Level: σ = 12.02
  - Input Range: [-50, 50]

================================================================================
LAB COMPLETED WITH REALISTIC METRICS!
================================================================================
```