

Machine Learning: Regression

PRESENTED BY PRINCE SINGH
23-12-2025



Abstract

This project explores the fundamental principles of Supervised Machine Learning, specifically focusing on regression analysis. By implementing Linear and Polynomial (Quadratic and Cubic) regression models using Python, this study evaluates the effectiveness of different loss functions—Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Overview

- > Introduction
 - Theoretical Framework
 - Methodology
 - Implementation
 - Results
 - Conclusion
 - Thank You
-

Introduction

4

- **Objective:** Transitioning from Linear Models ($y = mx + c$) to Polynomial Curves to capture real-world data curvature.
- **Model Flexibility:** Implementing Quadratic and Cubic functions to eliminate "underfitting" and allow the model to "hug" the data.
- **Optimization Feedback:** Utilizing Loss Functions to guide the model's accuracy:
 - MSE (Mean Squared Error):** Mathematically precise; equalizes average distances but reacts heavily to outliers.
 - MAE (Mean Absolute Error):** Robust and stable; balances the distribution of points to ignore extreme noise.

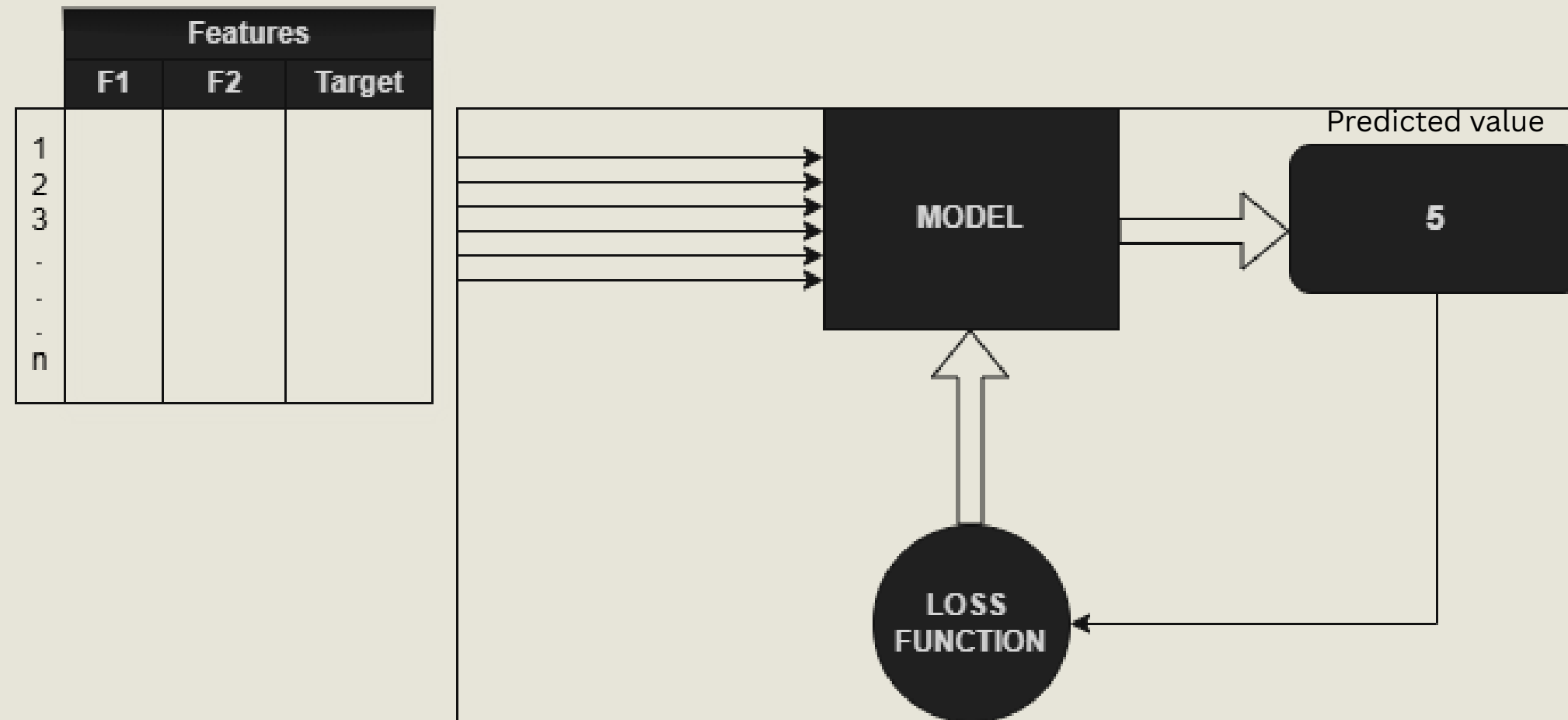
Theoretical Framework

5

Mathematical Metrics and Loss Functions

- Loss Function: A method to quantify how far the predicted value (y_{pred}) is from the actual value (y_{actual})
- MSE (Mean Squared Error): Squares the residuals: $MSE = \frac{1}{n} \sum (y - y_{pred})^2$
- MAE (Mean Absolute Error): Takes the absolute difference: $MAE = \frac{1}{n} \sum (|y - y_{pred}|)$

FUNCTIONING OF MODEL



Data Usage for different model training phases

80% Training

10% Validation

10% Testing

1. Every entry is sent to the model
2. The model predicts the value
3. The feedback (Loss Function) calculates the deviation in the predicted value.
4. The model learns and try to reduce error.

Methodology

```
# -----  
# 1. DEFINE ERROR METRICS (The "Feedback"  
Rules)  
# -----  
  
FUNCTION Compute_MSE(m, c):  
    PREDICTION = (m * x) + c  
    ERROR      = MEAN( (Actual_Y -  
PREDICTION)2 )  <-- Penalizes Outliers  
    RETURN ERROR  
  
FUNCTION Compute_MAE(m, c):  
    PREDICTION = (m * x) + c  
    ERROR      = MEAN( |Actual_Y -  
PREDICTION| )  <-- Robust to Outliers  
    RETURN ERROR
```

```
# -----  
# 2.THE OPTIMIZATION LOOP(Finding the "Best Fit")  
# -----  
  
INITIALIZE [m, c] at [0, 0]  
  
# "Goal: Adjust m and c until Error is at its  
lowest possible point"  
BEST_MSE_PARAMS = MINIMIZE( Compute_MSE )  
BEST_MAE_PARAMS = MINIMIZE( Compute_MAE )  
  
# -----  
-----  
# 3. OUTPUT FINAL MODELS  
# -----  
  
PRINT "MSE Model: y = {m_mse}x + {c_mse}"  
PRINT "MAE Model: y = {m_mae}x + {c_mae}"
```

Implementation

> ASSIGNMENT 1

Linear regression on linear data with noise and outliers.

Understanding MSE sensitivity to outliers.

> ASSIGNMENT 2

Failure of linear regression on non-linear data resulting high residuals.

Need a more nuanced method to capture data.

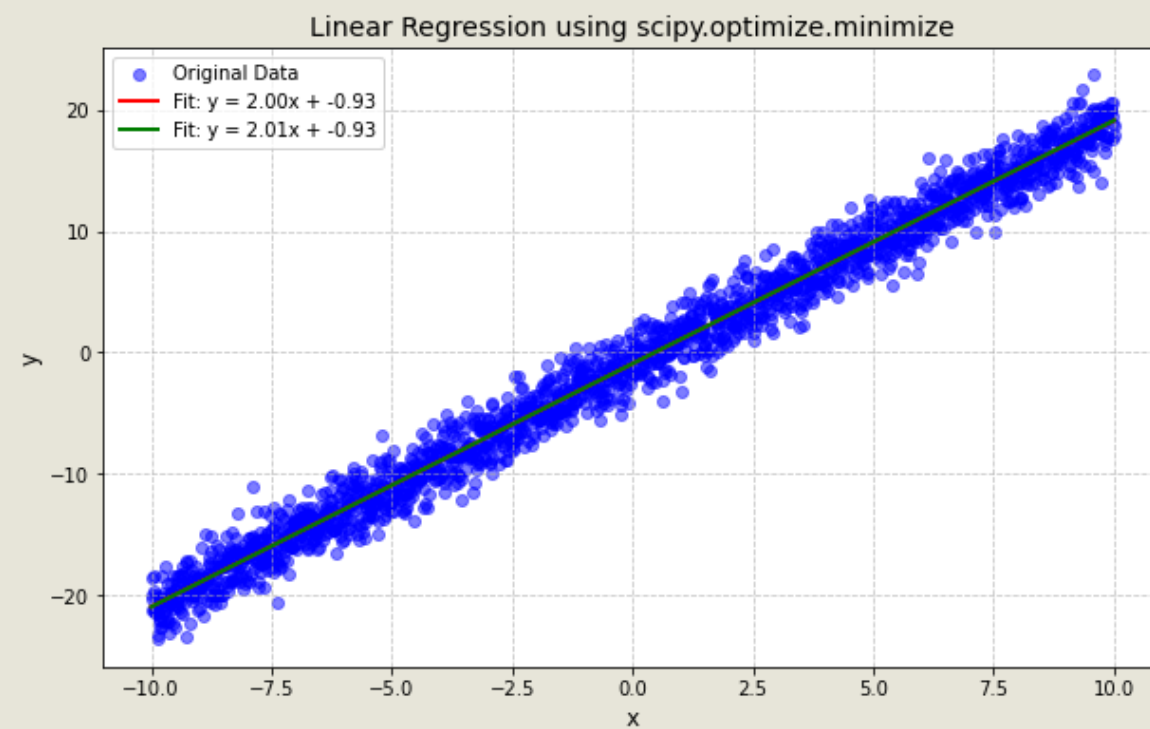
> ASSIGNMENT 3

Polynomial regression on non-linear data gave more accurate parameters.

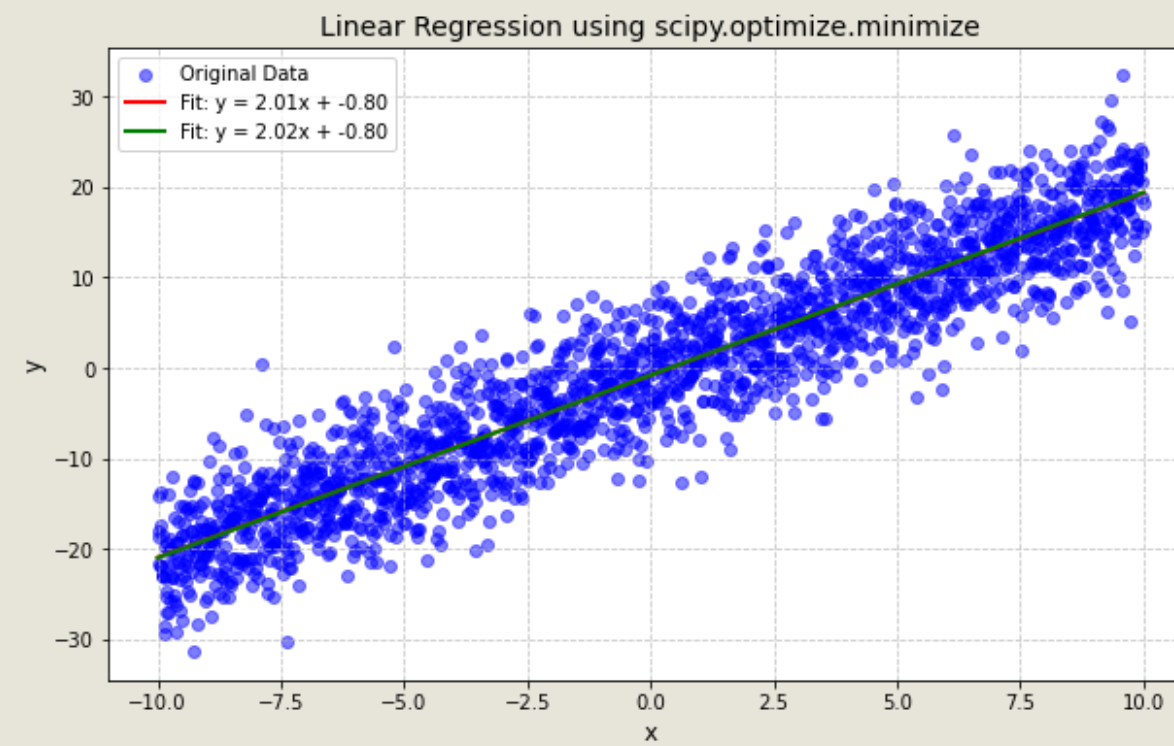
Even polynomial regression failed for circular data.

Result-1

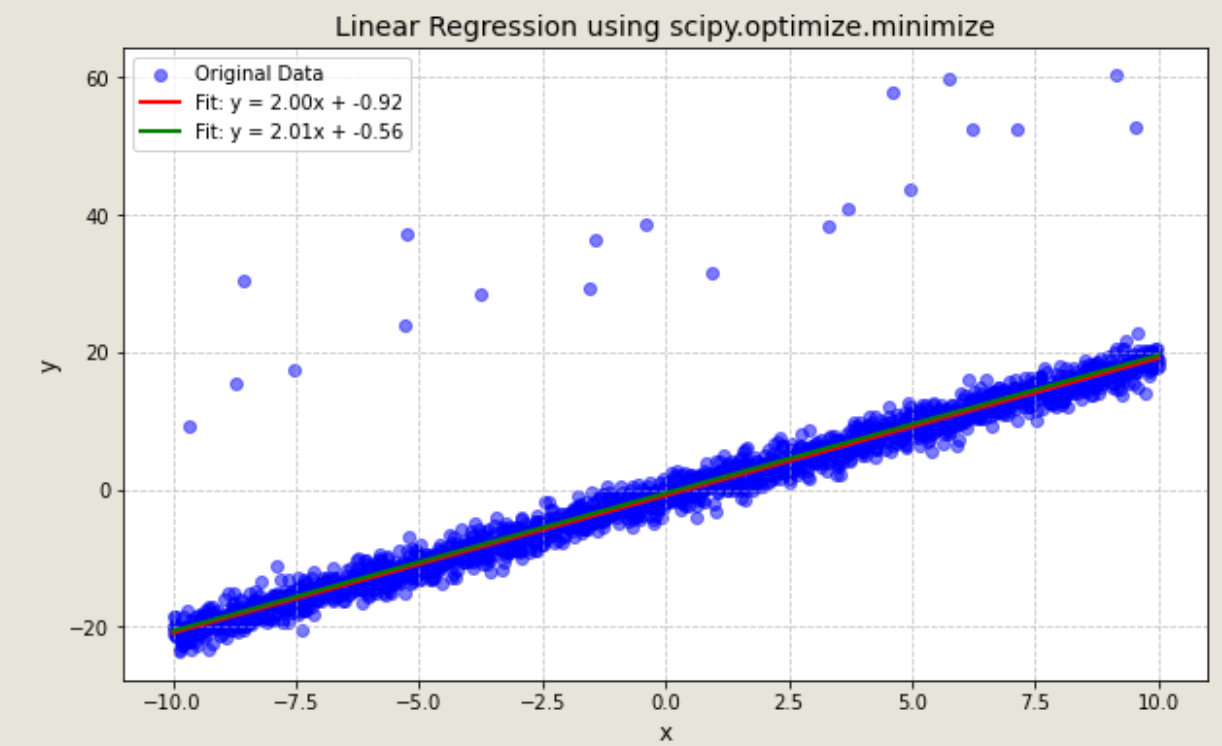
12



Less Noise
Similar MSE and MAE results



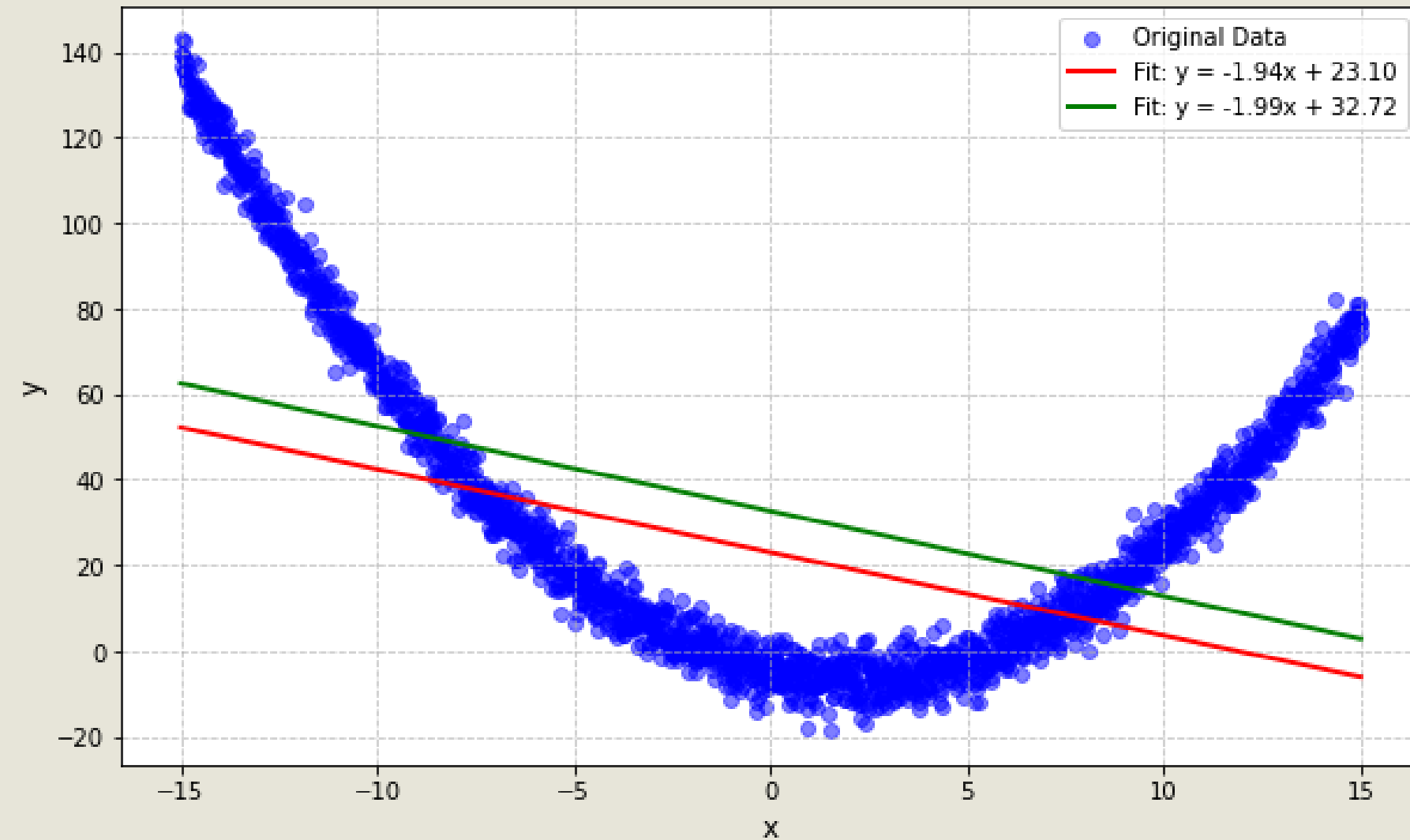
More Noise
MSE is more accurate than MAE results



Less Noise but significant outliers
MAE results more accurate than MSE

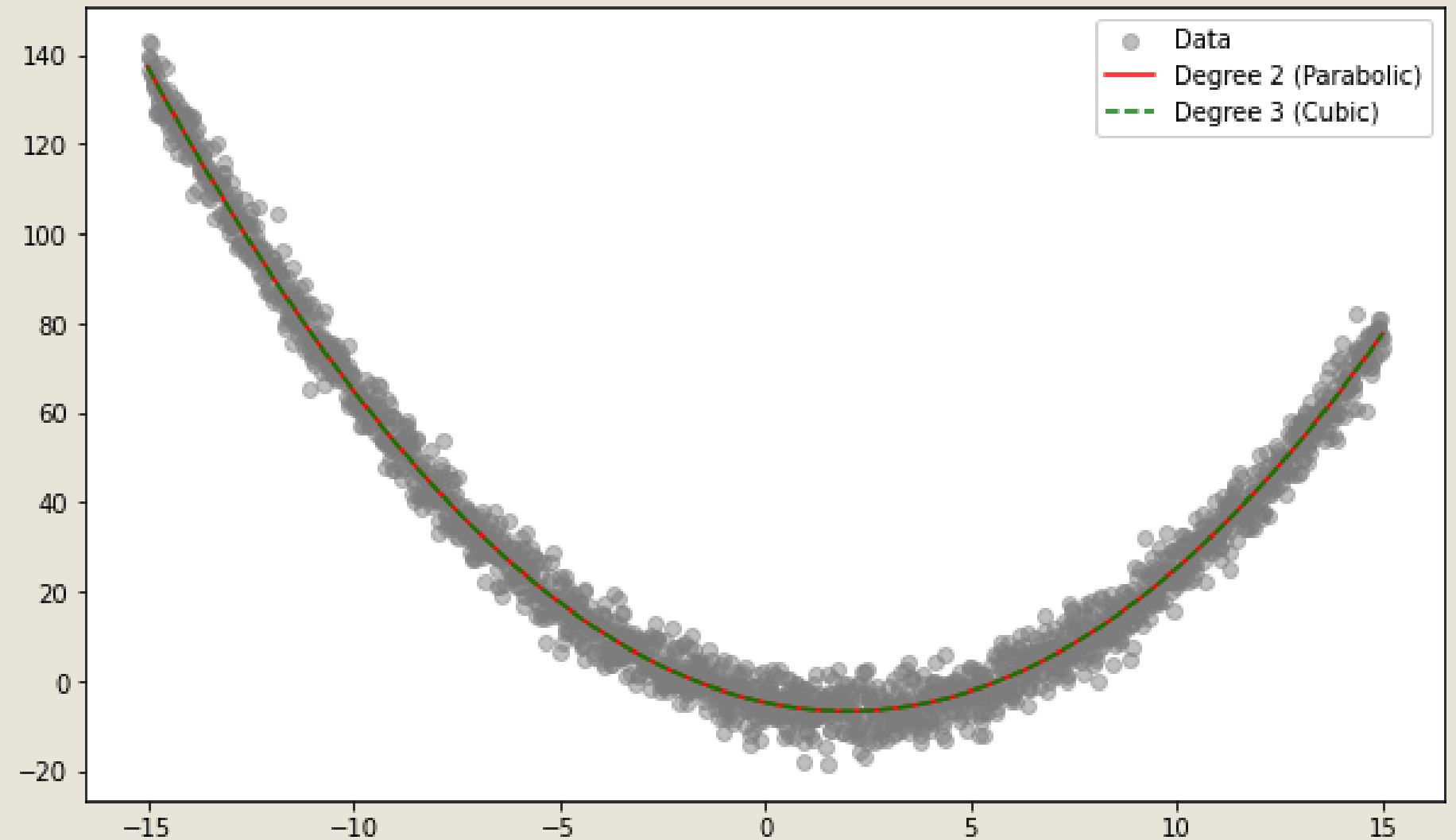
Result-2

Linear Regression using scipy.optimize.minimize



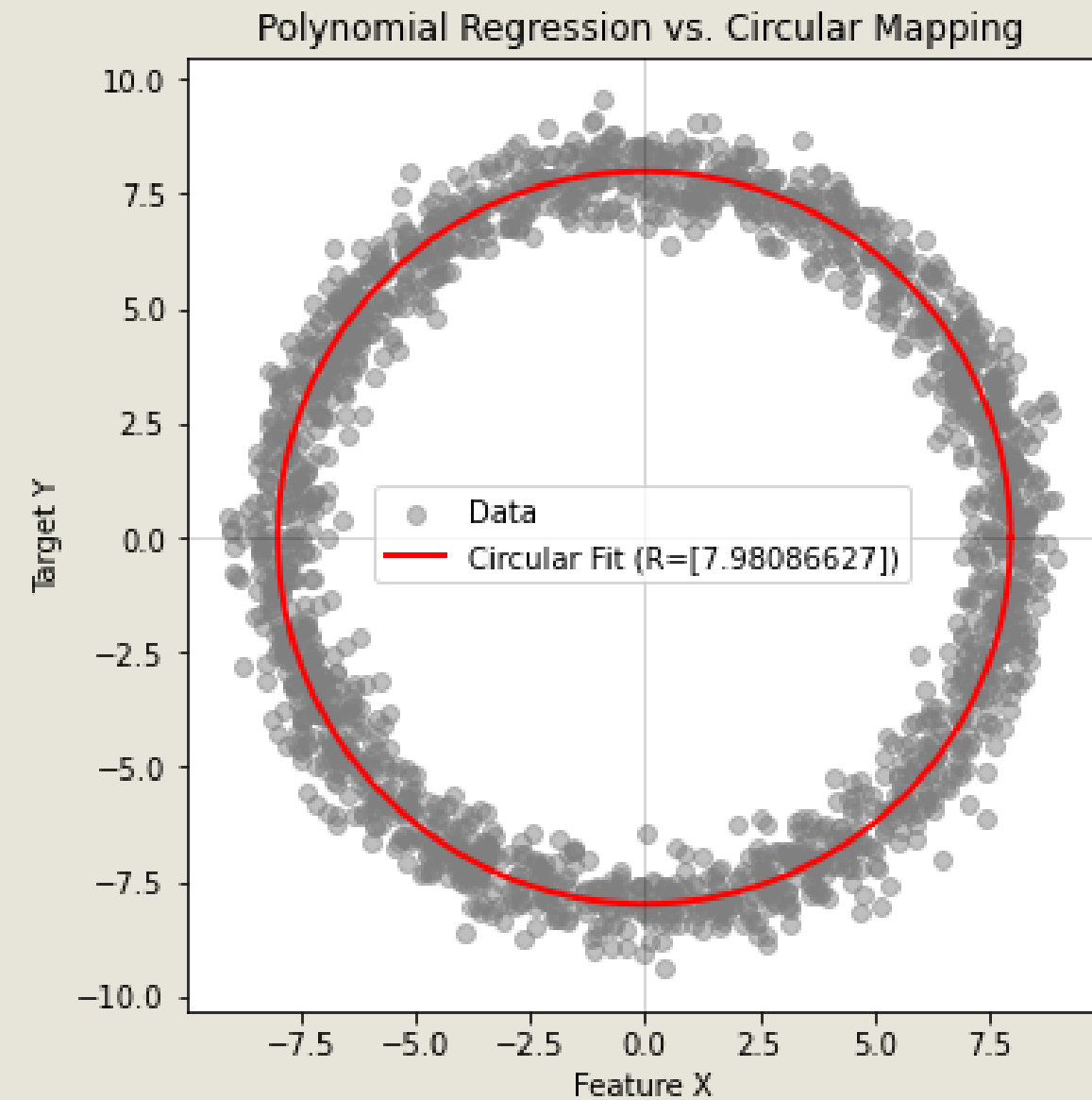
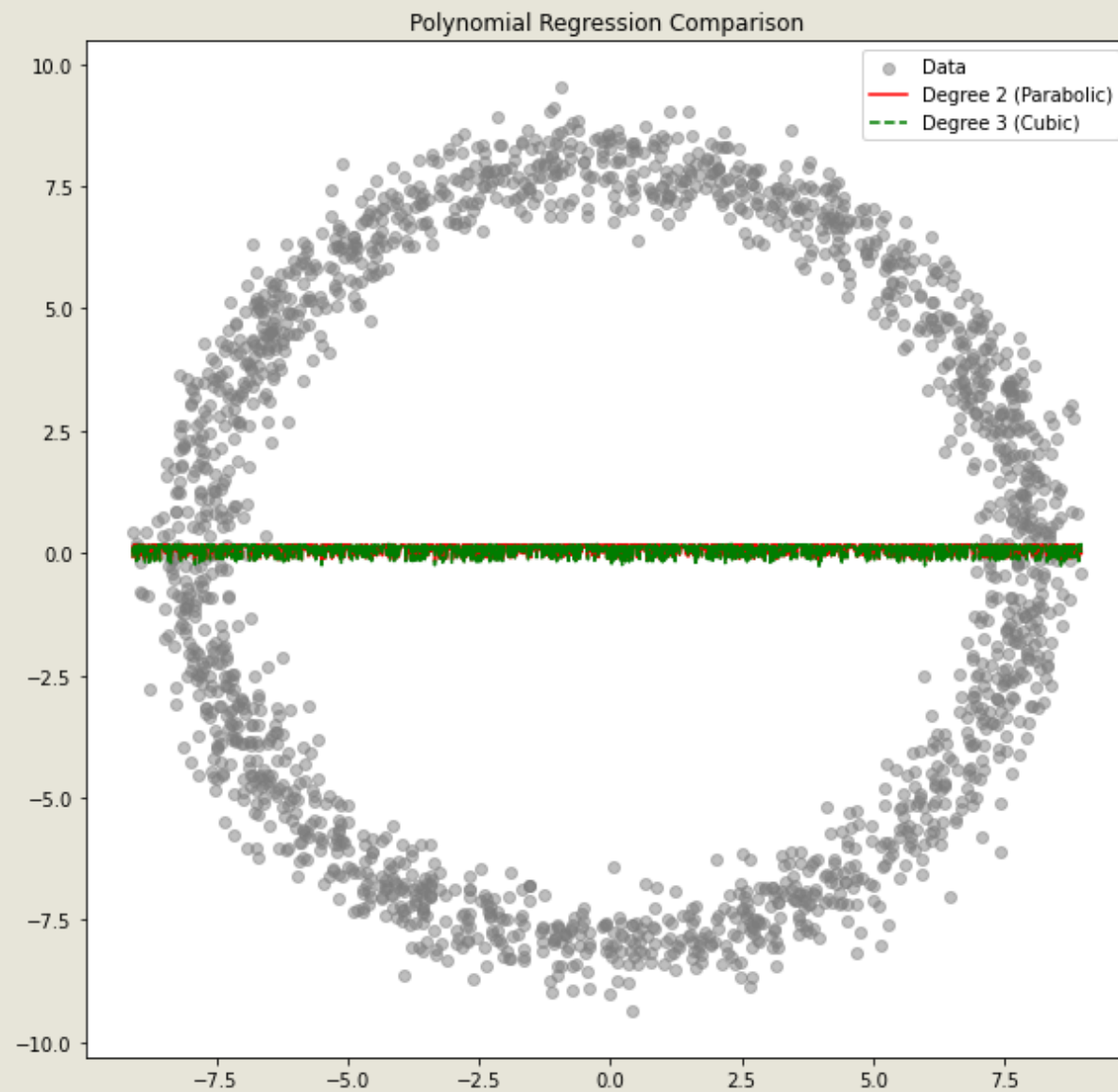
Linear regression on non-linear data resulted in high residuals and information loss

Polynomial Regression Comparison



Polynomial regression hugged the data more accurately with minimum possible data loss.

Result-3



Regression methodology fails in case of a circular data as single x corresponds to two different y(s) upside and down.

For such data we transition from coordinate plan to polar coordinates.

Conclusion

Data Scenario	Recommended Model	Recommended Metric	Reason
Clean, Linear	Linear Regression	MSE	Precise and mathematically stable.
High Outliers	Linear Regression	MAE	Prevents the line from "tilting" toward noise.
Curved Trends	Polynomial (Cubic)	MSE/MAE	Reduces high mean distance of linear fits.
Circular Data	None (Non-Functional)	N/A	Fails as x maps to multiple y values.





Thank
You!