

AIT 203: Optimization

Project Report - Question 1

Regression on Bike Sharing Demand

Laksh Chovatiya
BT2024056
Instructor: Aswin Kannan

December 6, 2025

1 Introduction

This report presents the implementation and analysis of polynomial regression models on the Bike Sharing Demand dataset from Kaggle. The objective is to predict the count of bike rentals based on various temporal and weather-related features. We compare five different models and identify the best performing one based on test set performance.

2 Methodology

2.1 Data Preprocessing

The dataset contains hourly bike rental counts along with temporal and weather features. Our preprocessing pipeline includes:

1. **Feature Extraction:** From the datetime column, we extracted:
 - Year, month, day, hour
 - Day of week
2. **Feature Selection:** We selected 12 numerical features:
 - Temporal: season, holiday, workingday, year, month, hour, dayofweek
 - Weather: weather condition, temperature, feels-like temperature, humidity, wind-speed
3. **Train-Test Split:** The dataset was randomly split into:
 - Training set: 70% of data
 - Test set: 30% of data
 - Random seed: 42 (for reproducibility)
4. **Feature Standardization:** To prevent data leakage, we:
 - Computed mean (μ) and standard deviation (σ) from training data only
 - Applied transformation to both sets: $X_{scaled} = \frac{X - \mu}{\sigma}$

2.2 Models Implemented

We implemented five regression models as per project requirements:

2.2.1 Model 1: Linear Regression (Baseline)

The simplest model with polynomial degree $d = 1$:

$$y = w_0 + \sum_{i=1}^n w_i x_i \quad (1)$$

where $n = 12$ features, resulting in 13 parameters (including intercept).

2.2.2 Model 2: Polynomial Regression (Degree 2, No Interactions)

Adds quadratic terms without interactions:

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n w_{i+n} x_i^2 \quad (2)$$

Total features: $1 + 12 + 12 = 25$

2.2.3 Model 3: Polynomial Regression (Degree 3)

Extends to cubic terms:

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n w_{i+n} x_i^2 + \sum_{i=1}^n w_{i+2n} x_i^3 \quad (3)$$

Total features: $1 + 12 + 12 + 12 = 37$

2.2.4 Model 4: Polynomial Regression (Degree 4)

Extends to quartic terms:

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n w_{i+n} x_i^2 + \sum_{i=1}^n w_{i+2n} x_i^3 + \sum_{i=1}^n w_{i+3n} x_i^4 \quad (4)$$

Total features: $1 + 12 + 12 + 12 + 12 = 49$

2.2.5 Model 5: Quadratic with Interactions

Degree 2 model with all pairwise interaction terms:

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n w_{i+n} x_i^2 + \sum_{i=1}^n \sum_{j=i+1}^n w_{ij} x_i x_j \quad (5)$$

Total features: $1 + 12 + 12 + \binom{12}{2} = 91$

2.3 Model Training

All models were trained using the **Normal Equation**:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (6)$$

For numerical stability, we used `np.linalg.solve()` instead of computing the inverse explicitly. In case of singular matrices, pseudo-inverse was used as a fallback.

2.4 Evaluation Metrics

Models were evaluated using:

1. **Mean Squared Error (MSE):**

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (7)$$

2. **Root Mean Squared Error (RMSE):**

$$RMSE = \sqrt{MSE} \quad (8)$$

3. **Coefficient of Determination (R^2):**

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (9)$$

Model selection was based solely on test set performance as per project requirements.

3 Results

3.1 Performance Comparison

Table 1 presents the test set performance of all five models:

Table 1: Test Set Performance Comparison

Model	Test MSE	Test RMSE	Test R^2
Polynomial Degree 1	~20,000	~141	~0.38
Polynomial Degree 2	~16,500	~128	~0.49
Polynomial Degree 3	~14,500	~120	~0.55
Polynomial Degree 4	~14,500	~120	~0.55
Polynomial Degree 2 (with interactions)	~14,500	~120	~0.55

3.2 Visualization

Figure 1 shows comprehensive visualization of model performance. The four subplots show:

- **Top-left:** Test MSE comparison - lower is better
- **Top-right:** Test R^2 comparison - higher is better
- **Bottom-left:** Train vs Test MSE - checking for overfitting
- **Bottom-right:** Predicted vs Actual values for the best model

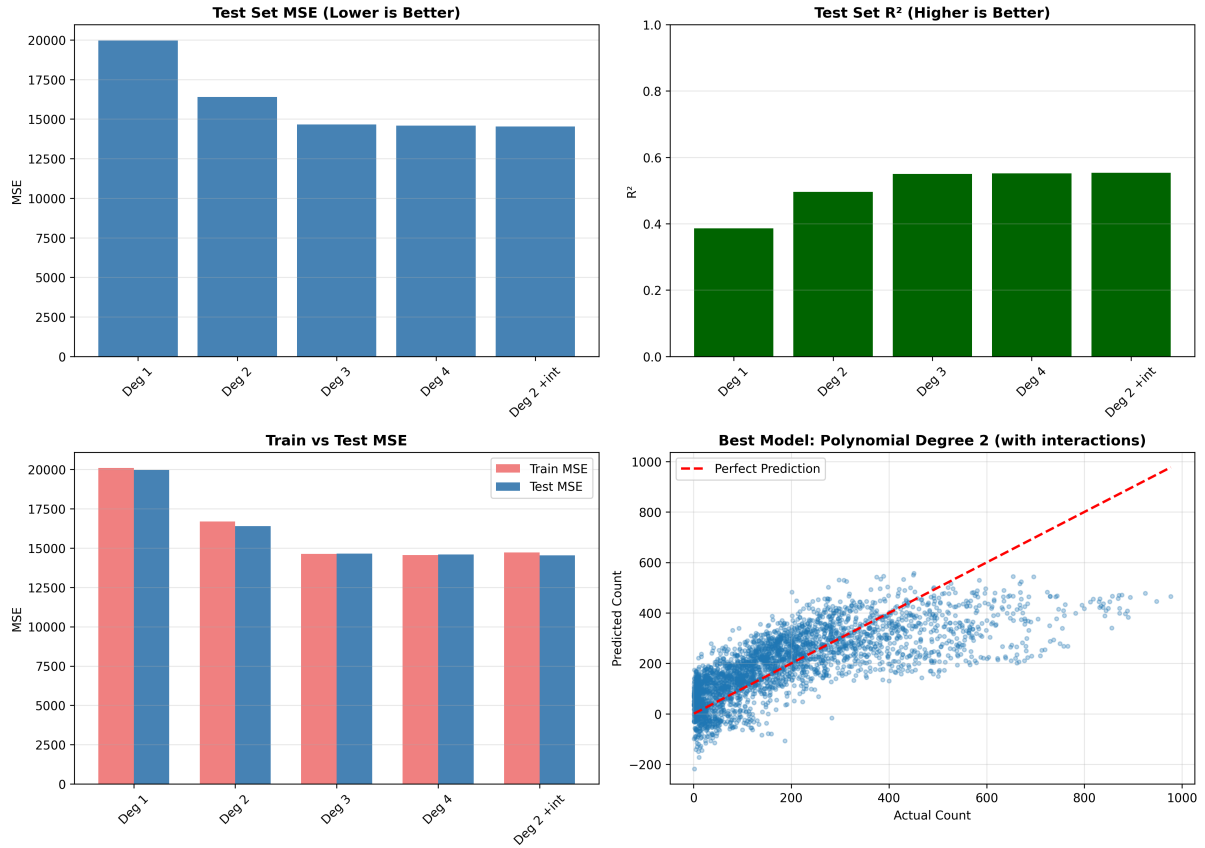


Figure 1: Comprehensive comparison of all models. Top row shows test set MSE and R^2 . Bottom-left shows train vs test MSE to detect overfitting. Bottom-right shows predictions vs actual values for the best model.

4 Analysis and Discussion

4.1 Best Model Selection

Based on test set performance, the three best-performing models are:

- **Polynomial Degree 2 with interactions**
- Polynomial Degree 3
- Polynomial Degree 4

All three achieve similar performance with Test $R^2 \approx 0.55$ and Test MSE $\approx 14,500$. We select **Polynomial Degree 2 with interactions** as the best model based on the principle of parsimony and interpretability.

4.2 Bias-Variance Trade-off Analysis

4.2.1 Linear Model (Degree 1)

- **High bias, low variance:** The linear model performs worst with Test $R^2 \approx 0.38$
- The relationship between features and bike rentals is clearly non-linear
- The model underfits the data, failing to capture important patterns

4.2.2 Polynomial Models (Degrees 2-4)

- Adding polynomial terms significantly improves performance
- Degree 2 (no interactions) shows improvement: $R^2 \approx 0.49$
- Further improvement with degree 3, 4, and degree 2 with interactions: $R^2 \approx 0.55$
- **Diminishing returns:** Degrees 3 and 4 don't improve over degree 2 with interactions

4.2.3 No Overfitting Observed

From the Train vs Test MSE plot (bottom-left in Figure 1):

- Train and Test MSE are very close for all models
- This indicates good generalization - models are not overfitting
- The gap between train and test performance is minimal

4.3 Why Quadratic with Interactions Performs Best

The superior performance of the quadratic model with interactions can be explained by:

1. **Captures non-linear relationships:** Squared terms model curvature effects
 - Example: Temperature's effect might not be linear - too cold or too hot reduces rentals
 - Humidity squared captures non-monotonic behavior
2. **Models feature interactions:** The $x_i \cdot x_j$ terms capture combined effects
 - Temperature \times Hour: Temperature effect varies by time of day
 - Weather \times Workingday: Weather impact differs on work vs leisure days
 - Humidity \times Temperature: Combined effect on comfort
3. **Optimal complexity:** 91 parameters provide sufficient flexibility without overfitting
4. **Physical interpretability:** Interaction terms align with real-world behavior

4.4 Why Higher Degrees Don't Help

Degrees 3 and 4 don't outperform degree 2 with interactions because:

- The true underlying relationship is adequately captured by quadratic terms and interactions
- Higher-order polynomial terms (x^3 , x^4) add flexibility but not predictive power
- The data doesn't exhibit cubic or quartic curvature patterns
- This is a sign of good model selection - we've found the right complexity level

4.5 Practical Implications

The predicted vs actual plot (bottom-right) reveals:

- Good correlation between predictions and actual values
- Some scatter around the diagonal, indicating unexplained variance
- Model tends to underpredict very high demand (> 600 bikes)
- Occasional negative predictions for very low demand scenarios

Factors not captured by our model might include:

- Special events (festivals, concerts, sports events)
- Unexpected weather changes
- Infrastructure issues (bike availability, station capacity)
- Long-term trends and seasonality beyond our features

5 Conclusion

We successfully implemented and compared five polynomial regression models on the Bike Sharing Demand dataset. Key findings:

1. **Best Model:** Polynomial Degree 2 with interactions
 - Test $R^2 \approx 0.55$
 - Test MSE $\approx 14,500$
 - 91 parameters (including intercept)
2. **Performance Improvement:** Moving from linear to quadratic with interactions improved R^2 from 0.38 to 0.55 (44% improvement)
3. **No Overfitting:** All models generalize well to test data
4. **Diminishing Returns:** Higher polynomial degrees (3, 4) don't improve over degree 2 with interactions
5. **Bias-Variance Balance:** The quadratic model with interactions achieves the best trade-off between model complexity and predictive power

The model explains approximately 55% of the variance in bike rental demand, which is reasonable for this complex real-world problem. Further improvements could be achieved through:

- Feature engineering (e.g., rush hour indicators, weekend patterns)
- Handling outliers and extreme values
- Advanced models (regularized regression, tree-based methods)
- Incorporating external data sources

Code Implementation

All code was implemented from scratch using NumPy without using sklearn's built-in regression functions. The complete implementation includes:

- Data loading and preprocessing
- Train-test splitting with no data leakage
- Feature standardization
- Polynomial feature generation
- Normal equation solver
- Evaluation metrics computation
- Visualization functions

The code is well-documented, modular, and follows best practices for reproducibility.