Question 1: Regression Analysis

Bike Sharing Demand Prediction

Project Report

Submitted by:

Aryan Malik (BT2024006)
Naman Jindal (BT2024203)
Abhyudaya Singh (BT2024180)

Department:

Computer Science and Engineering
IIITB

December 7, 2025

Contents

Abstract

This report details the implementation of regression models to predict bike-sharing demand. We implemented Linear Regression, Polynomial Regression (degrees 2, 3, 4), and a Quadratic Model with interaction terms. Models were evaluated on a strictly held-out test set. The **Quadratic Model with Interactions** achieved the lowest MSE, demonstrating that demand is best explained by the complex interplay between features rather than individual factors alone.

1 Problem Statement and Data

1.1 Objective

The goal is to predict the hourly bike rental count ('count') based on environmental settings (temperature, humidity, windspeed) and seasonal factors.

1.2 Dataset & Preprocessing

The dataset contains timestamped records. We performed the following preprocessing steps (see Appendix A.1 for code):

- **Feature Engineering:** Extracted 'hour', 'dayofweek', 'month', and 'year'.
- **Encoding:** Categorical variables were One-Hot Encoded. Numeric variables were Standard Scaled.
- **Splitting:** A strict 80-20 Train-Test split was used to prevent data leakage.

2 Methodology: Model Selection

We compared three types of models to analyze the bias-variance trade-off (see Appendix A.2 for definitions):

2.1 1. Linear Regression (Baseline)

Assumes a direct linear relationship ($y = w^T x + b$). This model serves as a baseline to detect underfitting.

2.2 2. Polynomial Regression (Degrees 2, 3, 4)

Introduces non-linearity ($x^2, x^3 \dots$) but explicitly excludes interaction terms. This tests if individual feature curvature (e.g., non-linear temperature effects) sufficiently explains the variance.

2.3 3. Quadratic Model with Interactions

A degree-2 polynomial that includes interaction terms ($x_i \cdot x_j$). We hypothesize that features are interdependent (e.g., the effect of 'humidity' depends on 'temperature').

3 Results and Analysis

3.1 Performance Evaluation

The models were evaluated strictly on the test set. The full results are provided in Table 1 (Appendix B).

- **Linear Regression:** MSE ≈ 10027, $R^2 \approx 0.696$
- **Polynomial (Deg 4):** MSE ≈ 9806, $R^2 \approx 0.703$
- **Quadratic Interaction:** MSE $\approx$ **9755**, $R^2 \approx$ **0.704**

3.2 Discussion: Which model is better and why?

Based on the experimental results, the **Quadratic Model with Interactions** is the superior model.

3.2.1 1. The Role of Interactions

The interaction model achieved the lowest MSE (9755.92). Even though the Degree 4 Polynomial is mathematically more complex in terms of curvature, it failed to beat the Degree 2 Interaction model.

- **Reasoning:** A pure polynomial model assumes features act independently. However, in reality, a rainy day at 5 PM (rush hour) causes a much larger drop in bikes than a rainy day at 3 AM. This conditional relationship is captured by the interaction term $x_{hour} \cdot x_{weather}$, which is absent in pure polynomial models.

3.2.2 2. Bias-Variance Trade-off

The Linear model performed worst, proving the data is non-linear (High Bias). While increasing the polynomial degree ($2 \rightarrow 4$) improved accuracy, the gains diminished. The Interaction model utilized "smarter" complexity (cross-features) rather than just "higher" complexity (powers), allowing it to generalize better without the overfitting risks associated with high-degree polynomials.

4 Conclusion

The analysis confirms that bike-sharing demand is driven by complex, interdependent factors. The Quadratic Interaction model strikes the optimal balance, leveraging cross-feature dependencies to provide the most accurate predictions.

A Appendix: Implementation Code

A.1 Preprocessing Code

```
    df["datetime"] = pd.to_datetime(df["datetime"])
    df["hour"] = df["datetime"].dt.hour
    df["dayofweek"] = df["datetime"].dt.dayofweek
    df["month"] = df["datetime"].dt.month
    df["year"] = df["datetime"].dt.year.map({2011: 0, 2012: 1})

    y = df["count"]

    numeric_features = ["temp", "atemp", "humidity", "windspeed"]
    categorical_features = [
        "season", "holiday", "workingday", "weather",
        "hour", "dayofweek", "month", "year"
    ]

    X = df[numeric_features + categorical_features]

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42
    )
```

A.2 Model Definition Code

```
    # Linear Model
    models["linear"] = Pipeline([
        ("preprocess", preprocess_linear),
        ("regressor", LinearRegression())
    ])

    # Polynomial Models without interaction terms
    for d in [2, 3, 4]:
        num_poly = Pipeline([
            ("scaler", StandardScaler()),
            ("powers", PowerFeatures(degree=d))
        ])

        preprocess_poly = ColumnTransformer([
            ("num", num_poly, numeric_features),
            ("cat", cat, categorical_features)
        ])

        models[f"poly_deg_{d}_no_inter"] = Pipeline([
            ("preprocess", preprocess_poly),
            ("regressor", LinearRegression())
        ])

    # Quadratic Model with interaction terms
    num_quad = Pipeline([
        ("scaler", StandardScaler()),
        ("poly", PolynomialFeatures(degree=2, include_bias=False))
    ])

    preprocess_quad = ColumnTransformer([
        ("num", num_quad, numeric_features),
        ("cat", cat, categorical_features)
    ])

    models["quadratic_with_interactions"] = Pipeline([
        ("preprocess", preprocess_quad),
        ("regressor", LinearRegression())
    ])
```

A.3 Training and Evaluation Code

```
    for name, model in models.items():
        print(f"Training {name}...")
        model.fit(X_train, y_train)

        preds = model.predict(X_test)
        mse = mean_squared_error(y_test, preds)
        r2 = r2_score(y_test, preds)

        results.append({
            "model": name,
            "test_MSE": mse,
            "test_R2": r2
        })
```

B Appendix: Results Table

Model Configuration	Test MSE	Test R^2 Score
Linear Regression	10027.06	0.6962
Polynomial Degree 2 (No Interactions)	10004.12	0.6969
Polynomial Degree 3 (No Interactions)	9816.74	0.7025
Polynomial Degree 4 (No Interactions)	9806.02	0.7029
Quadratic with Interactions	**9755.92**	**0.7044**

Table 1: Comparison of Model Performance on Test Set

C Plots: Actual vs Predicted