COMP 551: Linear Regression Project Report

Aymane Radouane (261231812) & Nickolai Tsoukanov (261161485)
& Williams Lendjoungou (261167713) & Team Members

February 28, 2026

Abstract

The dataset used in this project contains two years of bike sharing data. This report addresses the problem of predicting bike sharing demand using linear regression and evaluates performance using Mean Squared Error (MSE). A baseline linear regression model achieved a training MSE of approximately 5.85×10 and a test MSE of 5.35×10. Introducing polynomial and interaction features across all variables substantially reduced training error (MSE 8.68×10) but resulted in severe overfitting, with test MSE increasing to approximately 2.29×10. Restricting nonlinear feature engineering to continuous variables improved generalization, yielding a training MSE of approximately 4.83×10 and a test MSE of 4.53×10, which represents the best overall performance.

1 Introduction

In this project, we model a bike sharing dataset, specifically *"day.csv"*. This dataset consists of 731 rows (1 for every day) and has several attributes associated with each day, such as date, weather conditions, and count of bikes rented (refer to Appendix **??** for a complete list of attributes). Our aim is to perform linear regression on a cleaned version of said dataset that accurately predicts the daily *cnt* (count) attribute. Furthermore, we decided it beneficial to perform an additional linear regression on the ratio of registered bikes versus casual bikes rented. These are key parameters of the dataset. Shedding light not only on how many bikes are being rented, but also who is renting them.

2 Data Preprocessing and Exploration

2.1 Clean up

- **Validating Types** Some features in the dataset were not fit for linear regression. For example, one column in the *day.csv* file is the *instant* attribute. It trivially has no correlation to the count parameter. Two other features: *casual and registered*, were also removed, since their sum returns the value for *count*. No null values were found anywhere in the table.

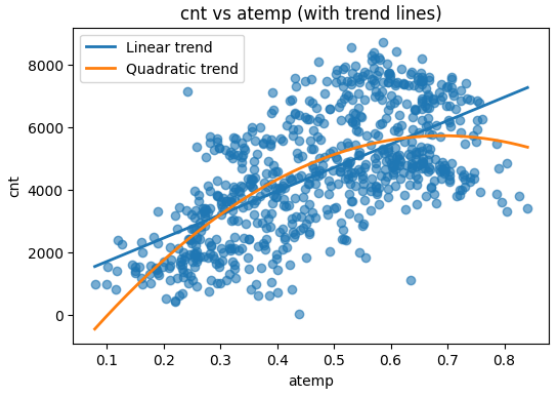
- **IQR Threshold** Other features such as *temp, atemp, hum* & *windspeed* had a high spread of values and contained outliers (extreme values). These extreme values heavily influence the weights assigned during regression from the true value. Outliers were found by performing the IQR test on all possible features with a 1.5 multiplier threshold. For the continuous features mentioned, a Numpy clipping function was used to bring them to threshold values.

- **Standardization (Z-Normalization)** It was decided that standardization, also known as z-normalization, should be applied to all features because larger scale features have a tendency to dominate those with small numbers. This normalization also in turn improved the stability of weights and efficiency of calculations. Specifically, this was done on transformed engineered attributes.
- **One Hot Encoding** One hot encoding was added to discrete features to have them locked in a range of 0 and 1.

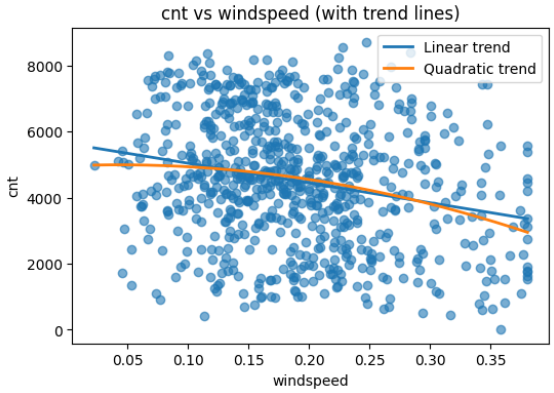
2.2 Data Exploration

To help determine which features to perform linear regression on, all remaining cleaned-up features were plotted against the *count* parameter. Allowing further elimination of redundant features, and determining which features should be used in non-linear transformations.

Scatter plots with linear and quadratic fits were made for continuous features, where *temp, atemp, hum* can be seen to roughly have a quadratic relationship to the *count* feature in graph **??**. This subtle key invites us to focus non linear feature engineering on continuous variables. Note that windspeed has a linear correlation.



(a) Example of a quadratic correlation to the target attribute.



(b) Example of a linear correlation to the target attribute.

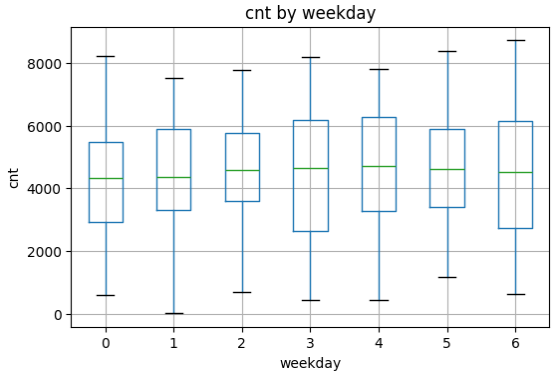
Figure 1: Comparing Continous Features to the Target Variable.

For discrete features, box and whisker plots were plotted, as seen in figure **??**. The *working day* and the *weekday* features showed little variance in *count* results, potentially being redundant for use in linear regression. The rest of the features, *yr, season, mnth, holiday, weathersit* showed the opposite, and influence of the *count* variable. After data exploration, it was decided that linear regression would use the following attributes: *temp, atemp, hum, windspeed, yr, season, mnth, holiday, weathersit.*

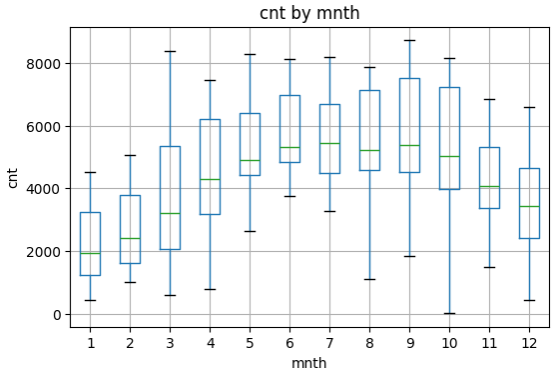
3 Methods

3.1 Testing set

The dataset was split into a testing and training set randomly. Such an approach was chosen over a naive split because it allows the training set to have data points over the entire period of 2 years, rather than a specific truncated portion of time.



(a) Example of a discrete feature showing little variance in *count* results



(b) Example of a discrete feature having large variance in *count* results

Figure 2: Box and Whiskers Plot to Deduce Influence of Feature on *Count* Target Variable.

3.2 Linear Regression

For linear regression, all chosen and polished features were converted into a design numpy matrix and a result matrix. The *lstsq* function was used from the numpy.linalg library, see listing 1. This returns a Beta matrix containing all weights desired for linear regression.

Feature importance was approximated by the absolute value of each coefficient, since larger magnitudes indicate stronger conditional effects on the predicted rental count. For categorical variables, each coefficient measures the difference between that category and the reference category (the dropped level).

```
Beta = np.linalg.lstsq(X_train, y_train, rcond=None)[0]
```

Listing 1: Linear Regression Function

3.3 Feature Engineering

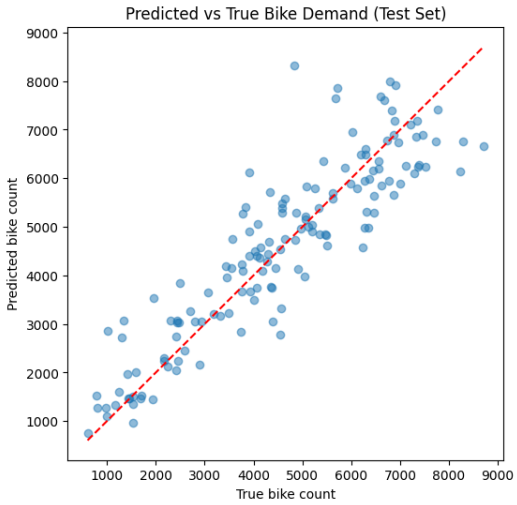
For non-linear transformations, we added second-degree polynomial features and pairwise interaction terms to sets of normalized input variables.

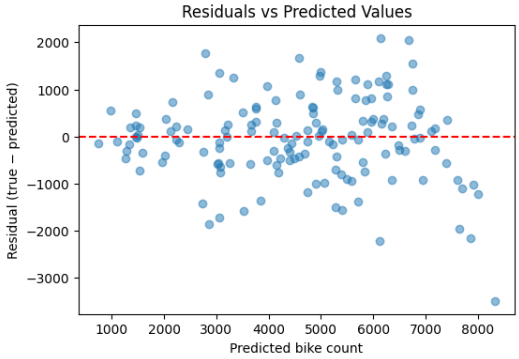
4 Results

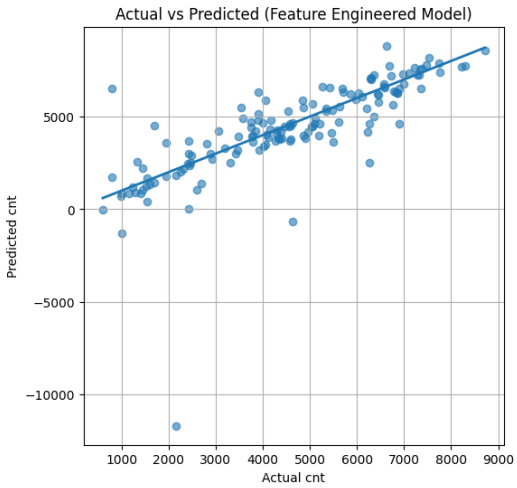
Linear regression was performed on 10 features (split into 32 due to one hot encoding), as mentioned in the data clean up section above. The results table below shows that an MSE of $5.35 \cdot 10^5$ was achieved (against the test set). The relationship of $count_{pred}$ to $count_{true}$ is displayed in the figure below, along with its residuals. Following this result, the weights of all parameters were ranked in decreasing order to further eliminate unnecessary features, such as *month*, as seen in figure 4. Selected MSE's of regressions trails can be found in table 1, where lower ranking attributes were removed. In the same vein, the lowest MSE found for the ratio of casual verse registered bikes was $1.60 \cdot 10^{-2}$ (against test set), with a rank of 7. For the feature engineering model, the lowest MSE was found by including all 10 features as well, coming in at $4.53 \cdot 10^5$. The relationship of $count_{pred}$ to $count_{true}$ is displayed in figure 3.

Model Variation	Rank	MSE (train)	MSE (test)
Baseline LR (count)	32(full)	$5.84 \cdot 10^5$	$5.35 \cdot 10^5$
Baseline Linear Regression (count)	20	$6.25 \cdot 10^5$	$5.60 \cdot 10^5$
Baseline Linear Regression (count)	10	$7.03 \cdot 10^5$	$6.49 \cdot 10^5$
Baseline Linear Regression (ratio)	10	$1.15 \cdot 10^{-2}$	$1.60 \cdot 10^{-2}$
Feature Engineered Model (count)	32(full)	$8.7 \cdot 10^4$	$2.29 \cdot 10^6$
Feature Engineered Model (count)	4(continuous)	$4.83 \cdot 10^5$	$4.53 \cdot 10^5$

Table 1: Model Performance on the Test Dataset.







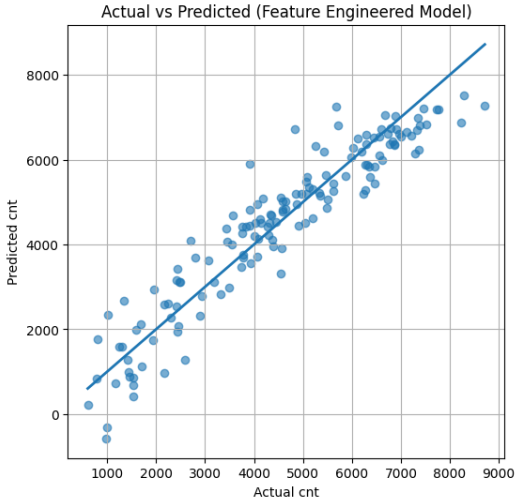
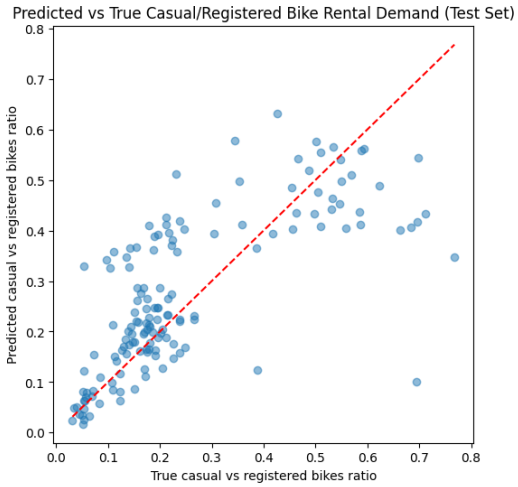
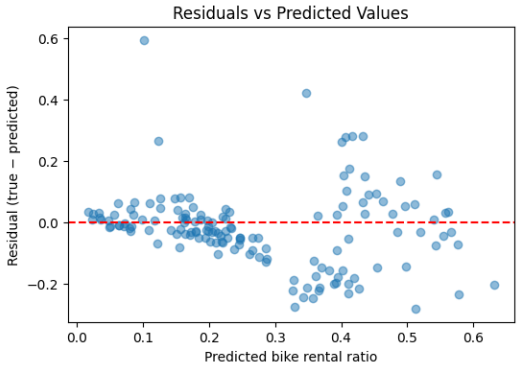


Figure 3: Ranking the Weights for Linear Regression to Predict Count Variable.

	feature	beta	abs_beta
1	yr	1976.248224	1976.248224
11	weathersit_1	1277.059155	1277.059155
10	season_4	1073.556026	1073.556026
12	weathersit_2	863.330866	863.330866
29	mnth_9	802.519942	802.519942
20	weekday_6	758.828487	758.828487
13	weathersit_3	-750.965664	750.965664
3	temp	649.783873	649.783873
2	workingday	560.728533	560.728533
8	season_2	433.249644	433.249644
25	mnth_5	399.125268	399.125268
7	season_1	-380.581952	380.581952
26	mnth_6	309.652740	309.652740





5 Discussion and Conclusion

Feature importance analysis indicates that temporal and categorical variables are the primary drivers of bike rental demand. In particular, the year indicator has the largest conditional effect, a strong overall growth in usage between the two years. Weather and seasonal indicators also rank highly, which is normal for biking. In contrast, continuous weather variables such as temperature, humidity, display wind speed have smaller marginal effects once seasonal and categorical factors are accounted for, suggesting that much of their influence is mediated through broader temporal patterns.

A key limitation is that coefficient-based importance depends on predictor encoding and scale. One-hot encoded variables can inflate apparent importance relative to continuous features, especially with many categories. Furthermore, strongly correlated temporal variables can redistribute explanatory power, complicating causal interpretation. Future work could utilize model-agnostic measures like permutation importance or nonlinear models to better capture weather and seasonal interactions.

To investigate whether nonlinear transformations could improve linear regression performance, we initially added second-degree polynomial features and all pairwise interaction terms to the full

set of normalized input variables. Although this approach significantly reduced training error, it led to strong overfitting, as reflected by a large increase in test MSE. This behavior is attributable to the application of nonlinear transformations to one-hot encoded categorical variables, which substantially increased model dimensionality and introduced many sparse and redundant features.

To mitigate this issue, we proposed a targeted feature-engineering strategy by applying nonlinear transformations only to continuous variables. Guided by exploratory analysis, we selected temperature, apparent temperature, humidity, and wind speed and added a limited set of second-order polynomial and interaction terms for these features, while leaving categorical variables unchanged. This approach improved generalization performance and reduced overfitting, demonstrating the importance of selective, domain-informed feature engineering.

Statement of Contributions

Aymane worked on data clean up, standardization, analysis and baseline linear regression code, Williams developed data exploration strategies, data parsing, normalization and non-linear feature engineering, Nickolai aided with data organization and parsing, handled linear regression against the ratio of casual versus registered users, and focused on the report.

All students worked collaboratively on both the code and report, and reviewed all pieces.

Statement on the use of LLMs

LLM use was very limited during this project.