Predict Divvy Bike Checkouts Hourly

Kathleen Melonashi Neelu Nerella

1 Introduction

Many divvy bike users often encounter problems with attempting to locate available Divvy parking spots or even finding a divvy bike at the nearest station to them. If users are unable to find an empty parking spot at their initial station, then they'll have to travel more and spend more money as they search for the next nearest station. However, this can result in increased cost, wasted time and the inconvenience of having to walk to or from their intended destination to access a Divvy station. To address this issue of Divvy bike users, we developed a predictive model. This model aims to predict the number of bikes checked out at any given Divvy bike station at different times of the day. By doing so, we can make users' trips more efficient. This project involves the analysis of Divvy bike-sharing data from October 2023 in Chicago.

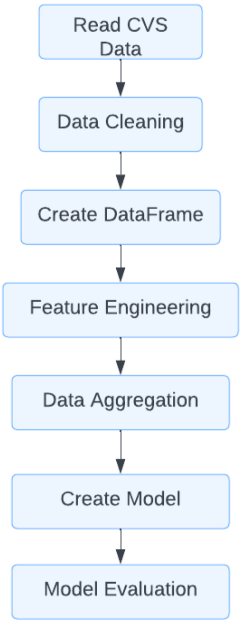


Figure 1: Flowchart of our project.

2 Data

In this statistical visualization, the count of rides is represented given the hour of the day. The scatter plot, with each point representing a distinct hour and its corresponding ride count, provides a comprehensive overview of the temporal distribution of bike rides. Peaks and troughs in the plot unveil patterns in user activity, highlighting peak hours of bike usage and potential lulls in demand.

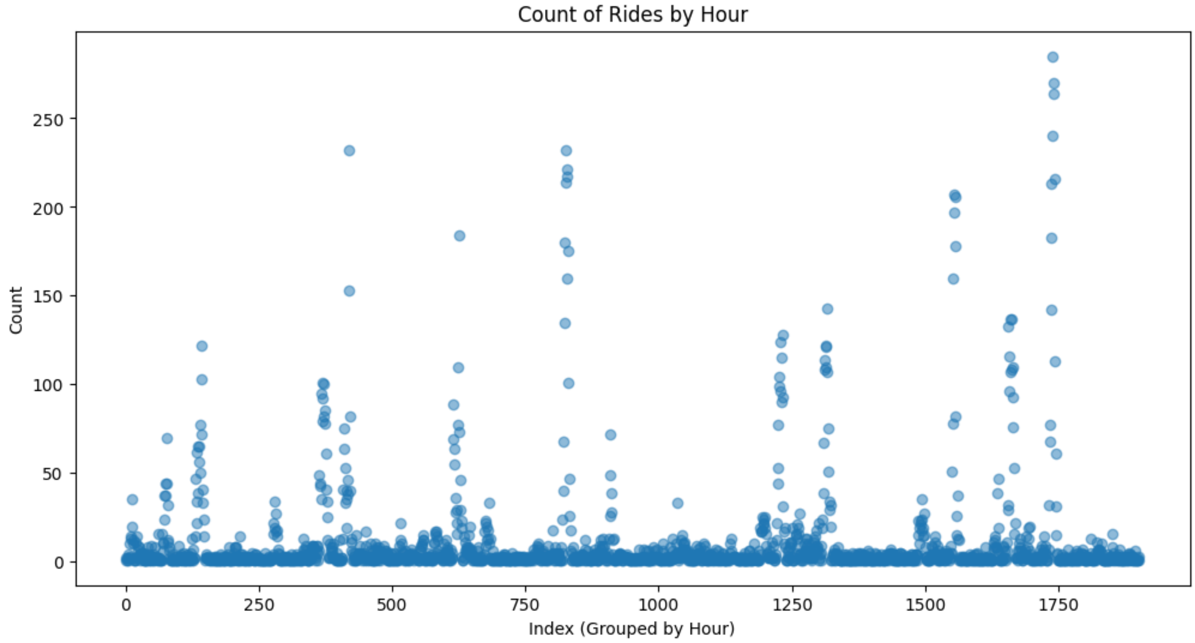
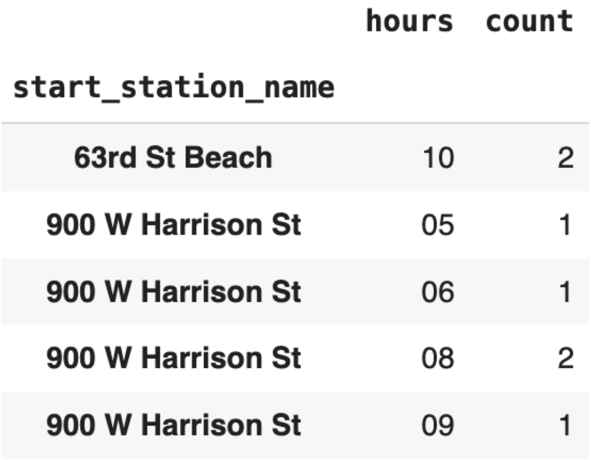


Figure 2: Count of rides grouped by hour (October 2023)

3 Methods

In this analysis, the goal was to predict the number of bikes/scooters getting checked out of a station every hour using four regression models: Decision Tree Regressor, Linear Regression, SVM Regression, and K-Nearest Regression. The dataset underwent pre-processing steps, including data cleaning to handle missing and non-numeric values, and identification of the top 10 stations based on ride counts. Feature engineering was then applied to extract temporal information such as day, month, and hour from the 'started at' timestamp, providing additional dimensions for modeling. After all of the data cleaning and feature engineering, we ended up with a data frame with the starting station name, the hour, and the count as features. Shown below are the first 5 rows in the final dataset.

Before working on the models, the dataset was first split into training and testing sets with the start station name and the hours as the base features and the count of bikes being checked



	hours	count
start_station_name		
63rd St Beach	10	2
900 W Harrison St	05	1
900 W Harrison St	06	1
900 W Harrison St	08	2
900 W Harrison St	09	1

Figure 3: Final Data Frame

out as the target label. All the models were then trained using the training data and then evaluated with Mean Squared Error (MSE) and R-Squared metrics.

3.1 Decision Tree Regression

After running the model and running the predicted values, the Decision Tree Regression model demonstrated reasonable performance, achieving an MSE of 80.86 and R-Sqaured value of 0.80. Cross-validation was also applied to assess the model's generalizability, revealing relatively consistent but sub optimal performance across folds.

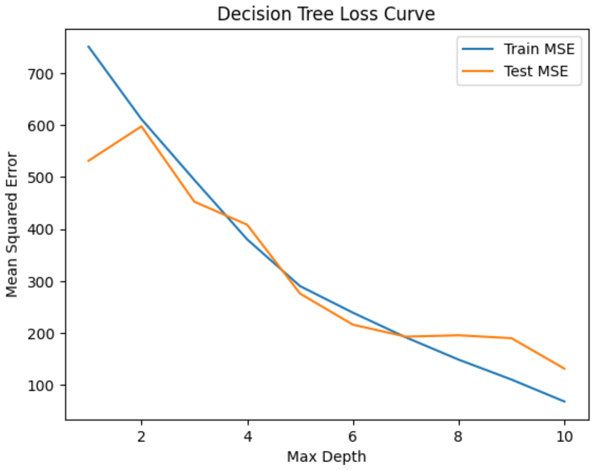


Figure 4: Decision Tree Loss Curve

3.2 Linear Regression

Conversely, the Linear Regression model exhibited unexpected and impractical results, with an exceedingly high MSE and a negative R-squared. This raised concerns about the model's fit and the quality of the data. Despite the linear nature of the problem, the Linear Regression model seemed to struggle with the given features. Further exploration and debugging of the Linear Regression implementation, as well as a closer examination of the dataset, are necessary to identify and address the issues leading to the model's poor performance.

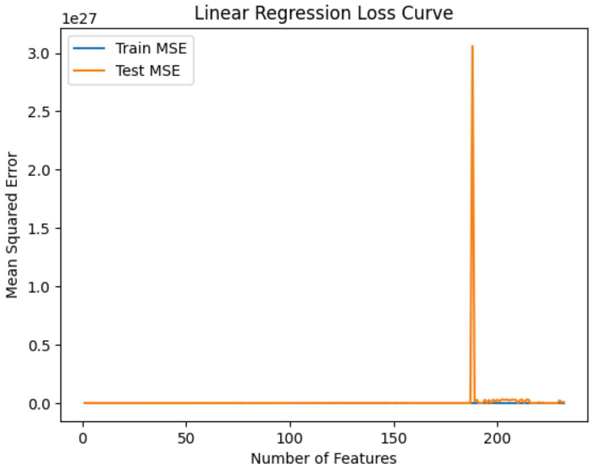


Figure 5: Linear Regression Loss Curve

3.3 SVM Regression

Likewise, the SVM model's performance was less ideal across different kernel choices, meaning it may not be well-suited for the dataset or requires further tuning of hyperparameters. Similarly to Linear Regression, the SVM model also returned an extremely high MSE and a negative R-Squared. The SVM Regression also seemed to struggle with the given features.

3.4 KNN Regression

The KNN model showed promise, with a Mean Squared Error (MSE) of 220.13 and an R-squared of 0.46. The loss curves demonstrated a decreasing training error with stable testing error, indicating a moderate level of model complexity. When calculating the accuracy of the predicted ytest values and the actual ytest values, the model outputted an accuracy of 77.69%.

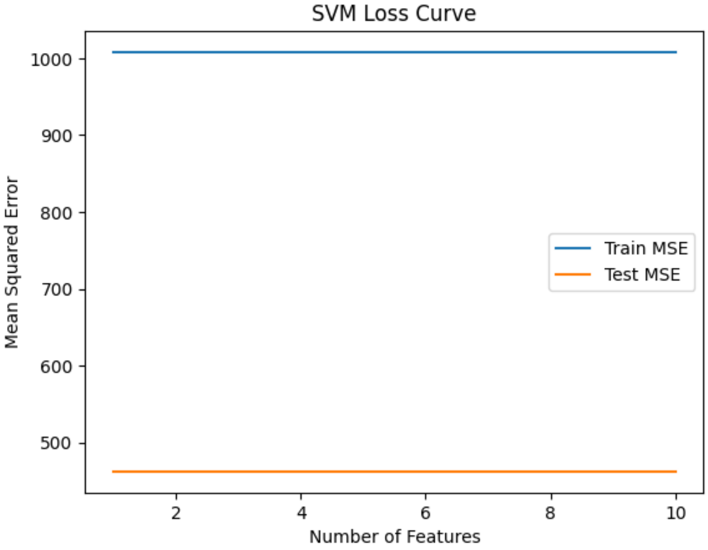


Figure 6: SVM Loss Curve

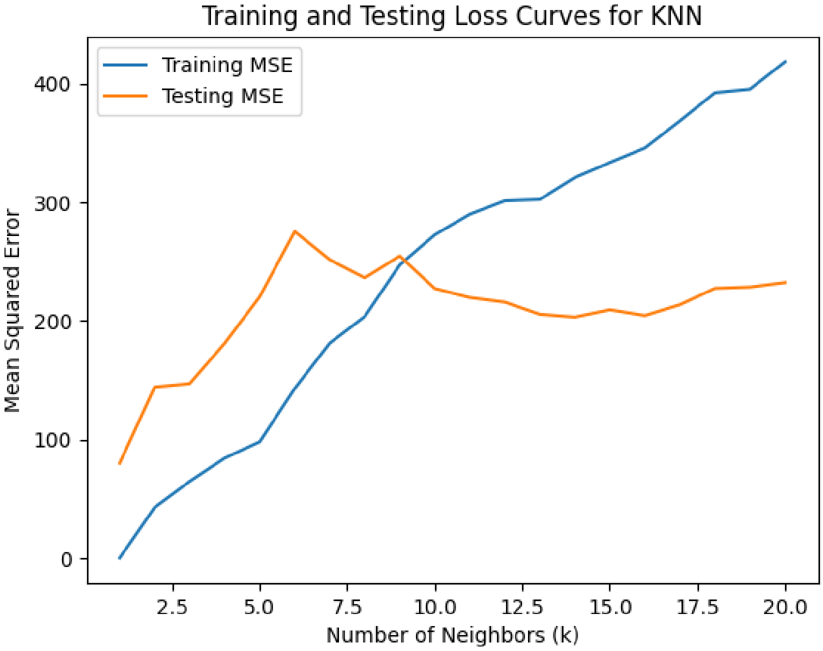


Figure 7: KNN Loss Curve

4 Results and Discussion

Out of all four models Decision Tree was the most successful one. The observed outcomes, particularly the extraordinarily high Mean Squared Error (MSE) and the negative R-squared value, suggest significant issues with the linear regression model in this context. After further exploration and debugging of both the Linear Regression implementation as well as the SVM Regression, and a closer examination of the dataset, it was discovered that these two models don't work that well with categorical features in the xtrain and xtest variables, thus leading to both the models' poor performance.

Outliers or Extreme Values: One common reason for an excessively high MSE is the presence of outliers or extreme values in the dataset. Outliers can disproportionately influence the model's coefficients, leading to inaccurate predictions. After examining and clearing outliers in the dataset we still got a high MSE.

Overfitting: The model was overfitting the training data, capturing noise rather than underlying patterns. Overfitting occurs when a model is too complex relative to the amount of training data available.

Data Scaling: Linear regression models are sensitive to the scale of input features. Inappropriate scaling can affect optimization algorithm convergence and coefficient interpretation.

The performance of SVM was inadequate for a variety of kernel selections, suggesting possible discrepancies between the properties of the dataset and the model. It might be required to adjust hyperparameters or investigate other models in order to improve its prediction abilities.

KNN Regression exhibited promise, reporting an MSE of 220.13 and an R-squared of 0.46. The loss curves revealed a decreasing training error with stable testing error, indicative of moderate model complexity. The accuracy of 77.69% suggests reasonable predictive capability.

5 Conclusion

To sum up, our attempt to forecast the number of hourly Divvy bike checkouts resulted in insightful information about the potential and difficulties of using machine learning to optimize urban mobility. The Decision Tree Regressor's predictive powers showed respectable results, providing a viable path to improve Divvy bike operations' efficiency. However, Linear Regression and SVM Regression encountered significant challenges, likely related to issues with categorical features, outliers, overfitting, and data scaling. On the other hand the KNN model's ability to capture patterns in the dataset and its accuracy of 77.69% indicate its potential for predicting the number of bikes checked out at Divvy stations. Moving forward, resolving these issues will be essential to realizing the full potential of our predictive model and, as a result, enhancing the usability and accessibility of Divvy bike services. To further continue predictive accuracy, it will be essential to carry on improving the models in the future, looking into new features, and taking into account different algorithms. Furthermore, dealing with problems like overfitting and outliers will help produce predictions that are more trustworthy and strong. Ultimately, by making it easier for users of Divvy bikes to find available bikes and parking spaces, the creation of precise predictive models holds the promise of greatly enhancing user experience in Chicago.

Link of our code on GitHub: https://github.com/neelunerella/DivvyBikePredictions

Reference

1. Divvy Bikes. (n.d.). System Data. Retrieved from `https://divvybikes.com/system-data`
2. Python Soldiers. (2022, Mar 1). Title of the video. [Video]. YouTube. `https://www.youtube.com/watch?v=qLDreXmJAD4&ab_channel=PythonSoldiers`
3. DataPixe. (2022, May 7). Title of the video. [Video]. YouTube. `https://www.youtube.com/watch?v=d-S42Urmns8&ab_channel=DataPixe`
4. Zakary Krumlinde. (2022 July 19) Using ML to Predict Hourly bike sharing checkouts