IE434 Deep Learning: Mathematics and Application
Predictive Analytics: NYC Citi Bike Station Deployment

Advika Pattiwar (linkedin.com/in/advika-pattiwar)
Dhruv Borda (linkedin.com/thebordadhruv)
Hrithik Rathi (linkedin.com/in/hrithik-rathi)
Suvrata Gayathri (linkedin.com/in/gayathrikappagantula)

December, 2023

Abstract

This deep dive project, guided by the expertise of Professor Richard Sowers, investigates the predictive analytics of New York City's Citi Bike station deployment. Utilizing a blend of deep learning and traditional data science methodologies, the focus is to create robust predictive models for bike rental patterns. The project primarily aims to predict the destination station of bike rides and forecast station popularity based on data from previous hours. Extensive data from various sources, including AWS S3 buckets and external URLs, are meticulously processed and organized for effective analysis. A novel approach in this project is the use of Graph Neural Networks (GNNs) to analyze daily variations in bike rentals, offering crucial insights for optimizing bike-sharing operations. The findings of this study are anticipated to significantly enhance the operational efficiency of bike-sharing systems in urban environments, contributing to sustainable and intelligent urban mobility solutions.

1 Introduction

This document presents a comprehensive study in the application of deep learning within urban bike-sharing systems, focusing on New York City's Citi Bike program. The project is motivated by several key objectives: optimizing resource utilization, identifying performance outliers, conducting root cause analysis, standardizing operations across stations, exploring growth opportunities, planning for capacity, and making informed strategic decisions. These goals stem from the imperative to transform data into actionable insights, thereby driving operational excellence and customer satisfaction in urban bike-sharing programs. The project revolves around an in-depth analysis of bike rental patterns, employing a range of data science and deep learning techniques. Two primary datasets are processed and analyzed, aiming to predict the destination station of bike rides and assess station popularity based on historical

data. Through the innovative use of Graph Neural Networks and other advanced analytical methods, this study strives to provide valuable predictions and strategies for the efficient management and expansion of bike-sharing networks.

2 Data Extraction

2.1 Data Sources

NYC Citi Bike Data: The NYC Citi Bike dataset, obtained from the official Citi Bike website, is an extensive collection of data points detailing the usage patterns of Citi Bike, New York City's bike-sharing program. This dataset encompasses various aspects of the bike-sharing system, including but not limited to start and end times of trips, station locations, trip duration, and user types (such as subscribers or casual users).

Weather Data: The weather data, sourced from the National Oceanic and Atmospheric Administration (NOAA), provides daily summary statistics for New York. This dataset includes a wide array of meteorological parameters such as temperature highs and lows, precipitation levels, humidity, wind speed, and other atmospheric conditions. The incorporation of this data is crucial for analyzing and understanding the impact of weather patterns on the usage of the NYC Citi Bike system. It allows for a comprehensive assessment of how various weather conditions influence the bike rental patterns, including seasonal variations and extreme weather events.

3 Methods for Retrieving the Data

3.1 Method 1: Downloading from External URLs

This method employs Python's 'requests' library to access and download datasets directly from specified URLs, focusing primarily on CSV files within ZIP archives. The process includes fetching the ZIP files, extracting their contents, and then reading the data into Pandas DataFrames. This approach is tailored for real-world scenarios where data is often available in compressed formats online.

3.2 Method 2: Accessing AWS S3 Bucket Data

Here, we leverage the AWS S3 service, a scalable cloud storage solution, to store and retrieve large datasets. By using the boto3 Python library, we connect to an S3 bucket, download the required data files, and subsequently load them into Pandas DataFrames. This method is particularly advantageous for handling extensive datasets and showcases an efficient way to interact with cloud-based storage solutions.

In both methods, a focus is placed on the correct parsing and formatting of datetime fields, which is vital for time-series analysis. Additionally, preprocessing steps such as cleaning, transformation, and handling of missing values are systematically performed to prepare the datasets for analysis.

3.3 Data Handling

Two distinct datasets - a debugging dataset for code testing and a larger working dataset are handled for comprehensive analysis.

3.3.1 Debugging Dataset

The debugging dataset is a smaller dataset specifically used for rapid testing and code validation. It facilitates quick iteration and efficient debugging, ensuring code robustness before application to the larger dataset. Key aspects include handling missing data during initial data extraction, applying techniques to address any missing values, and ensuring dataset completeness. The sparsity analysis includes one-hot encoding of the $'rideable\_type'$ feature, revealing a higher proportion of electric bikes over classic bikes. The $'start\_station\_name'$ is also one-hot encoded, indicating high sparsity and suggesting most stations are infrequently used as starting points, a logical outcome given the extensive number of stations.

3.3.2 Working Dataset

The main dataset of the project, it is larger and designed to closely mimic real-world scenarios. It is used for the final application of the developed models and methods. It comprises a comprehensive collection of Citi Bike rental records.

3.4 Data Pre-Processing

Comprehensive data preprocessing is a critical step in preparing the NYC Citi Bike and weather datasets for analysis. This process involves a series of steps designed to transform raw data into a format that is more suitable for exploration and modeling. The following is a summary of the key preprocessing activities undertaken:

3.4.1 Initial Data Assessment and Transformation

- *Data Type Inspection*: We began by checking the data types of all columns in the 'working_df' to understand the nature of each feature.
- *Datetime Conversion*: The 'started_at' and 'ended_at' columns were transformed into datetime objects, facilitating temporal analyses.
- *Date Hierarchy Extraction*: From the 'date' column, detailed temporal components such as year, month, week, day, weekday, and hour were derived for granular time-series exploration.

3.4.2 Temporal Feature Engineering

- *Time-Specific Features*: New columns like 'start_time', 'end_time', and 'trip_duration_in_minutes' were created to capture specific time-related insights.

3.4.3 Handling Missing Values

- *Data Cleaning*: Missing values were identified and addressed, primarily by dropping rows with NaNs to ensure the cleanliness and reliability of the dataset.

3.4.4 Data Merging and Enrichment

- *Merging Datasets*: A merge operation was executed between 'working_df' and 'DailyWeather' based on their 'date' and 'DATE' columns, respectively.
- *Join Methodology*: A left join was utilized to augment the 'working' dataset with weather-related data from 'DailyWeather'.
- *Column Streamlining*: Post-merge, redundant columns like 'DATE' and 'STATION' were eliminated from the 'working' DataFrame to streamline the data and remove duplicate information.

3.4.5 Post-Processing Inspection and Serialization

- *DataFrame Inspection*: The 'info()' method was used to review the structure, non-null counts, and data types of the merged 'working' DataFrame.
- *Data Serialization for Efficiency*: To handle the large dataset efficiently, Python's pickle module was employed. This involved creating a BytesIO object, converting the DataFrame to a pickle format, and uploading it to an AWS S3 bucket for rapid access and reuse in future analyses.

3.4.6 Categorical Data Transformation

- *Feature Encoding*: Transformation of categorical data, such as station names, was performed. Techniques like one-hot encoding were applied to convert these into formats suitable for machine learning models.

The preprocessing steps outlined above have been meticulously carried out to ensure the data is clean, structured, and ready for in-depth analysis and modeling. This groundwork is fundamental in extracting meaningful insights and patterns from the NYC Citi Bike and weather datasets.

4 Data Exploration

Comprehensive data exploration is conducted on both datasets, focusing on understanding the underlying patterns and characteristics.

Working Dataset Exploration: This phase extends the exploration to the larger dataset, emphasizing real-world applicability. It involves more extensive data visualization and statistical analysis to understand the dataset's nuances and prepare it for more complex models.

- The trip duration distribution histogram shows that trips of duration less than 20 minutes are the most frequent trips with a peak at the trip duration of approximately 4.5 minutes. This suggests that shorter-duration rides are the norm, indicating quick and possibly shorter commuting patterns.
- Based on the stratification analysis using the box plots for ride duration by rideable type, it can be concluded that the difference between ride duration for three rideable types, i.e. classic bike, electric bike and docked bike are not statistically different from each other as there is overlap between the box plots. This implies that ride duration might not vary significantly based on the type of bike used.
- Similarly, based on the stratification analysis using the box plots for ride duration distribution by user type, it can be concluded that the difference between the ride duration for the two user types, member and causal, is not statistically different from each other as there is overlap between the box plots.
- That being said, it is clear from the ride duration distribution histogram that members ride bikes significantly more frequently than casual riders. However, the frequency of rides is highest between the ride duration of 2 mins – 16 mins for both rider types.

Sparsity Analysis This was important in understanding how the data is distributed across various features, which influences the choice of machine learning algorithms and feature engineering techniques.

A thorough sparsity analysis revealed intriguing patterns:

- Rideable types displayed significant disparities in usage, with docked_bike exhibiting a high sparsity of nearly 99.76%, indicating infrequent utilization. In contrast, electric_bike emerged as the most commonly used type, followed by classic_bike.
- Start Station Name sparsity indicated that a vast majority of stations were infrequently utilized as starting points for rides, aligning with the extensive availability of stations.
- From the top and bottom 10 end stations bar chart, it can be observed that the end station with the highest number of rides has 3,700 rides coming in whereas the end station with the lowest number of rides has 1ride coming in. The considerable disparity in ride numbers between top and bottom end stations implies varying station popularity. This insight could be crucial for optimizing bike placement and resource allocation.

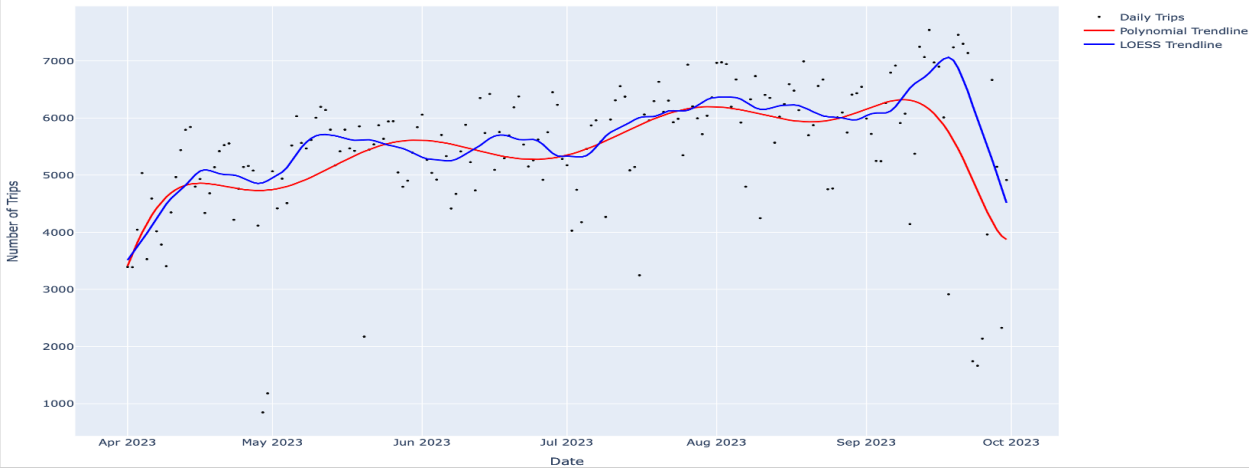

Figure 1: Citi Bike Daily Trips with Polynomial and LOESS Trendlines

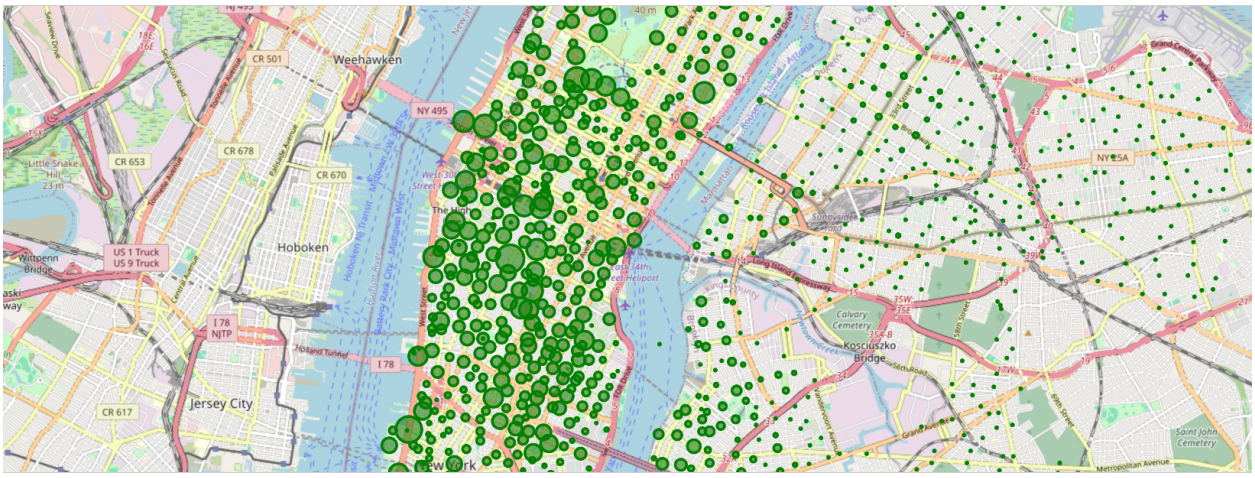

Figure 2: Start-Off Popularity

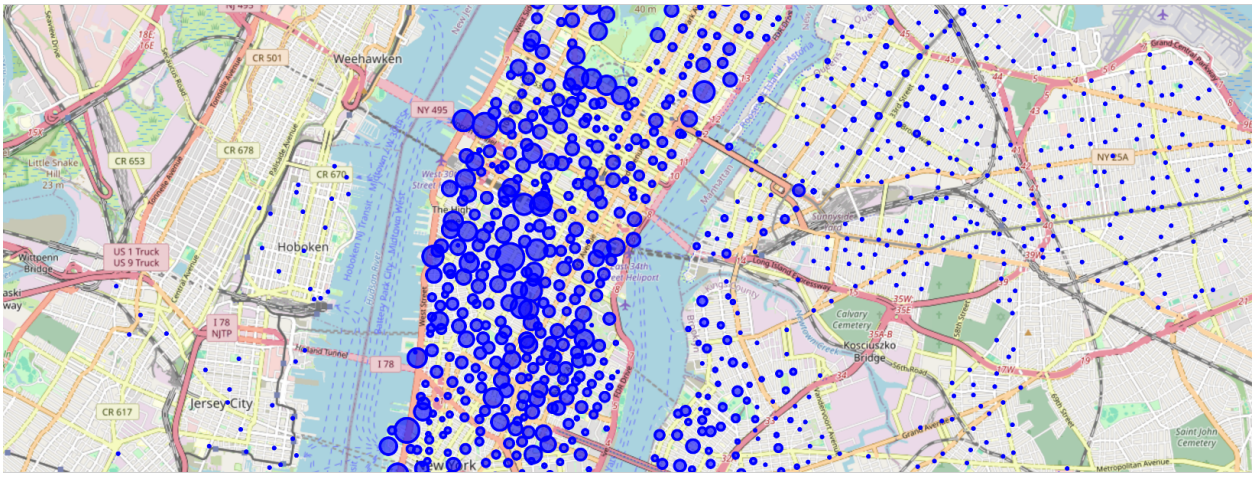

Figure 3: Drop-Off Popularity

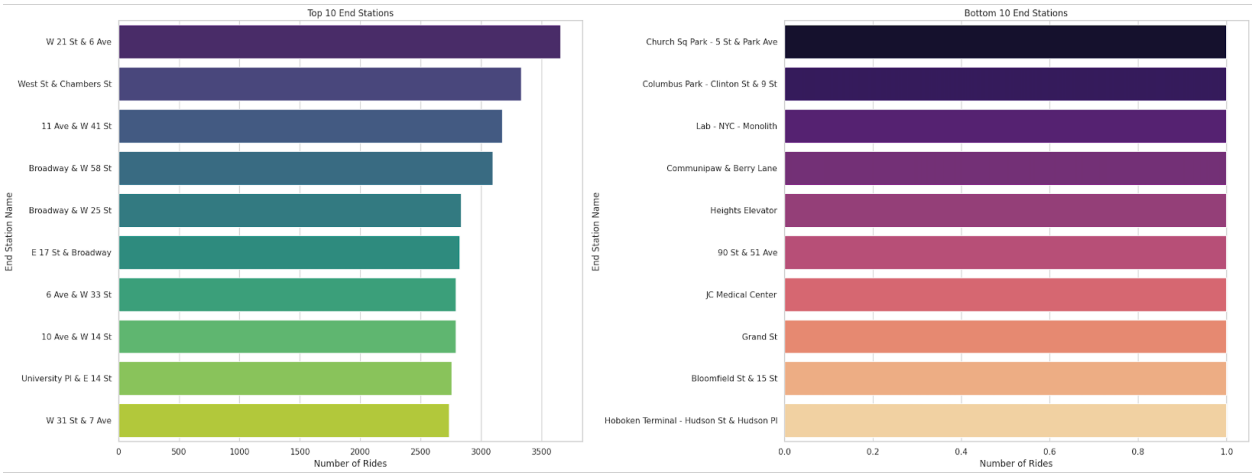

Figure 4: Top 10 Stations and Bottom 10 Stations

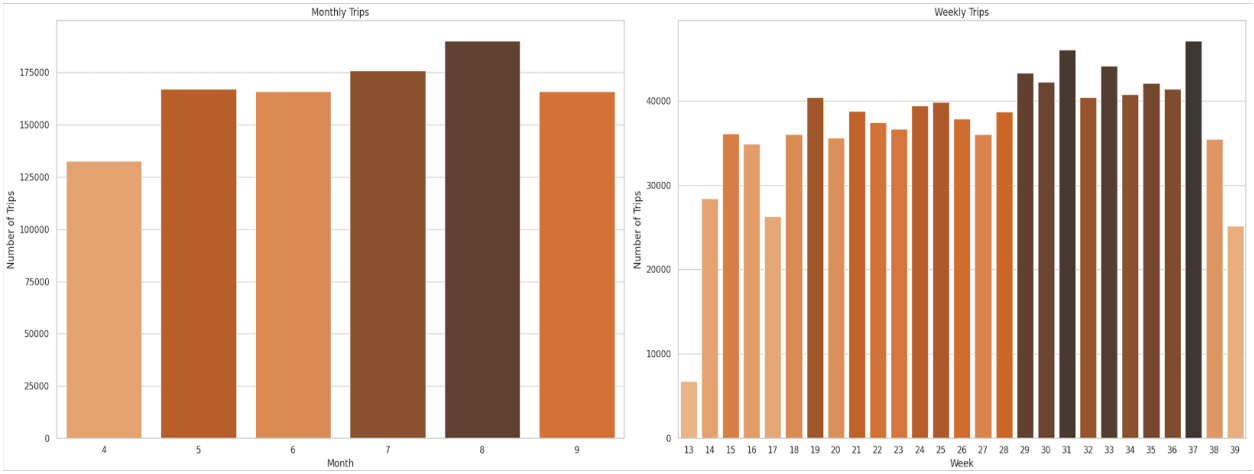

Figure 5: Monthly Summary and Weekly Summary

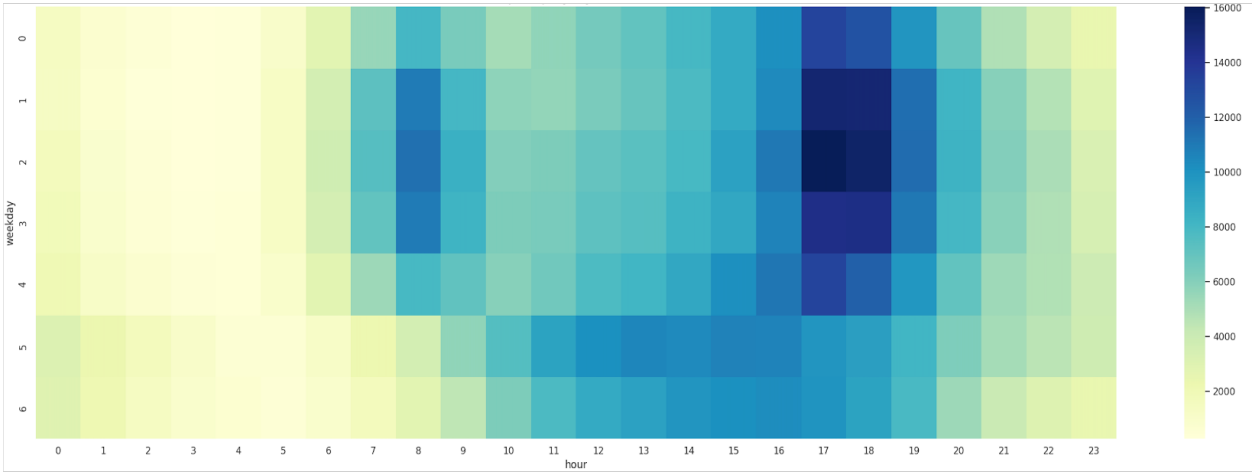

Figure 6: Heatmap of Trips by Day of Week and Hour

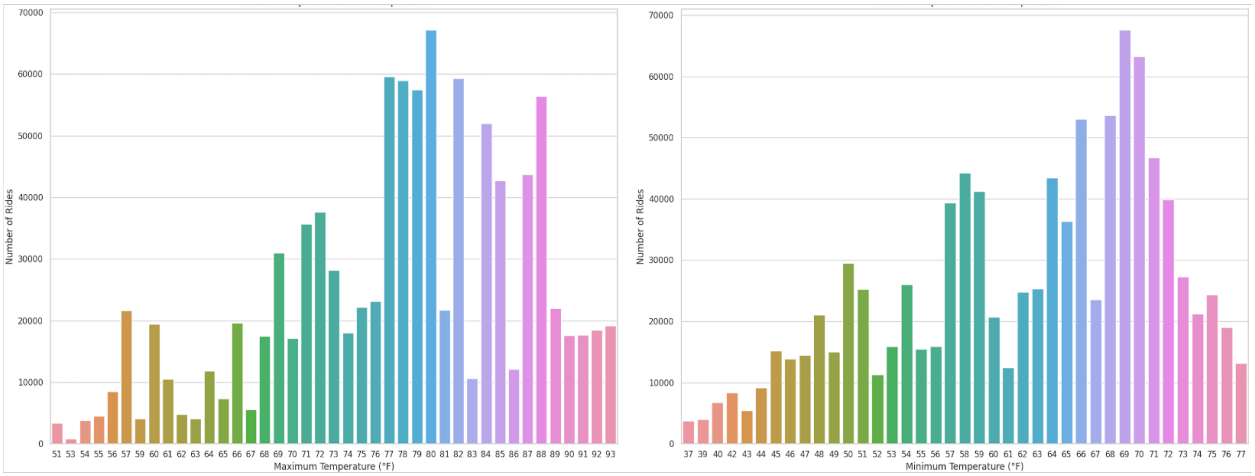

Figure 7: Number of Trips by Maximum Temperature and Minimum Temperature

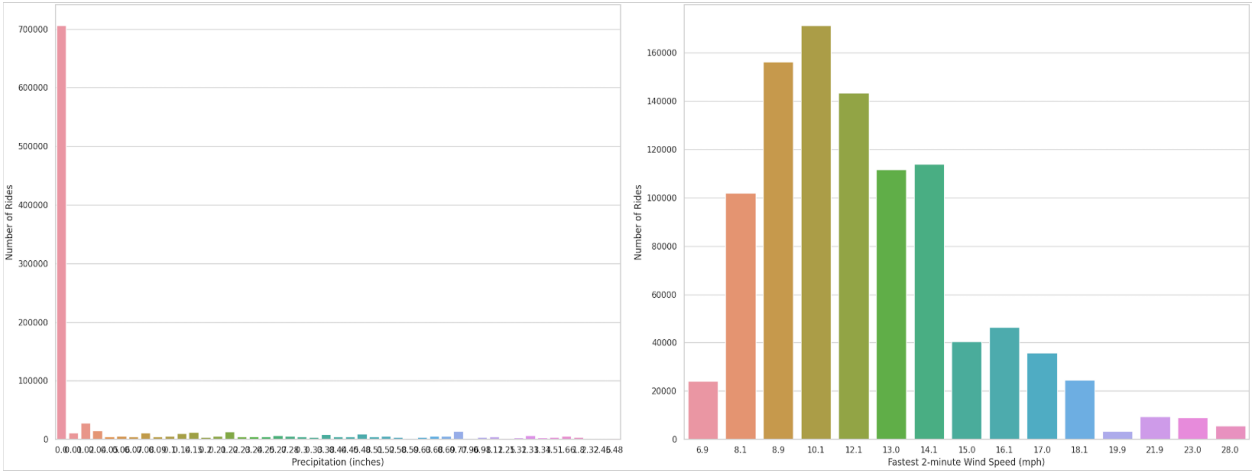

Figure 8: Number of Trips by Precipitation and Fastest 2-minute (average) Wind Speed

5 Baseline Models

Development of baseline models serves as a foundation for more advanced analyses.

1.Predicting Destination Station

This model is categorized as a classification problem since it predicts categorical outcomes (end station names). The model's objective is to predict the final destination of a bike rental, which is crucial for understanding and optimizing bike distribution across the city.

Features

- *Start Station (Frequency Encoded)*: Frequency encoding helped to represent the popularity or usage frequency of start stations, providing the model with insights into common starting points.

- *Rideable Type (One-Hot Encoded)*: Differentiated between bike types, which influence rental duration and destination preferences.
- *User Type (One-Hot Encoded)*: Distinguished between customer types (e.g., members vs. non-members), as different user groups may have different usage patterns.
- *Start Time (Cyclical Encoding for Hour and Weekday)*: Captured temporal patterns in bike usage, acknowledging the cyclical nature of time.
- *Geographical Coordinates (Normalized)*: Incorporated the spatial aspect of bike rentals, crucial for understanding how geography influences destination choices.
- *Weather Binary Indicators (WT01, WT02, WT03, WT08)*: Weather conditions can significantly impact user behavior and destination choices.
- *Weather Continuous Variables (Normalized)*: Included variables like wind speed, precipitation, and temperature, providing a comprehensive view of the weather's impact.

Target Variable

- *End Station Name (Label Encoded)*: The prediction target is appropriately transformed for use in a classification model.

Models
A ***Random Forest Classifier*** has been implemented. It is a robust choice for handling a mix of different feature types and their complex interactions.

Performance Metrics

- *Accuracy* : Ratio of correctly predicted observations to the total observations in a dataset.

2. Predicting Bike Station Popularity Based on Previous Day Data

Regression model type is used for predicting a continuous variable which is the number of bikes dropped off at a station. The goal is to predict the number of bikes that will be returned to a station on a given day, which is key for managing bike availability and distribution.

Features

- *Time-Related Features with Cyclical Encoding*: This includes the day, weekday, week, and month, all encoded to capture their cyclical nature, which is important for reflecting patterns like higher weekend usage or seasonal variations.
- *Weather Binary Indicators (WT01, WT02, WT03, WT08)*: Includes conditions like fog, thunder, etc., that can impact bike usage.
- *Weather Continuous Variables (Normalized)*: Factors like temperature, wind speed, and precipitation, which are normalized for consistency and comparability.
- *Previous Day's Bike Drop-offs (Lagged)*: This feature directly links past data to future predictions, assuming some level of continuity in rental patterns.

Target Variable

- *Number of Bikes Dropped Off (number_of_dropoffs)*: A clear and directly measurable outcome that reflects station popularity.

Models

- *Linear Regression* model has been implemented as it is simple and interpretable.
- *XGBoost*, a more advanced model capable of capturing non-linear relationships and interactions between features has also been used.

Performance Metrics

- *Mean Absolute Error (MAE)*: Provides an average error magnitude, making it easy to interpret.
- *R-squared*: Indicates how much of the variance in the target variable is predictable from the features.

Results

Linear Regression:

- Mean Squared Error (MSE): 8.24
- Mean Absolute Error (MAE): 1.97
- R-squared (R2): 0.49

XGBoost:

- Mean Squared Error (MSE): 7.84
- Mean Absolute Error (MAE): 1.97
- R-squared (R2): 0.52

Both models provided fairly similar results, with XGBoost performing slightly better than Linear Regression. However, the R-squared values indicate that there is still some variance in the data that the models cannot explain, suggesting room for further model refinement.

Overall, the baseline models provide a starting point for predicting destination stations and bike station popularity. Further improvements could be made by exploring different algorithms, feature engineering, and hyperparameter tuning. Additionally, it may be beneficial to incorporate additional data sources or explore more advanced modeling techniques to enhance predictive accuracy.

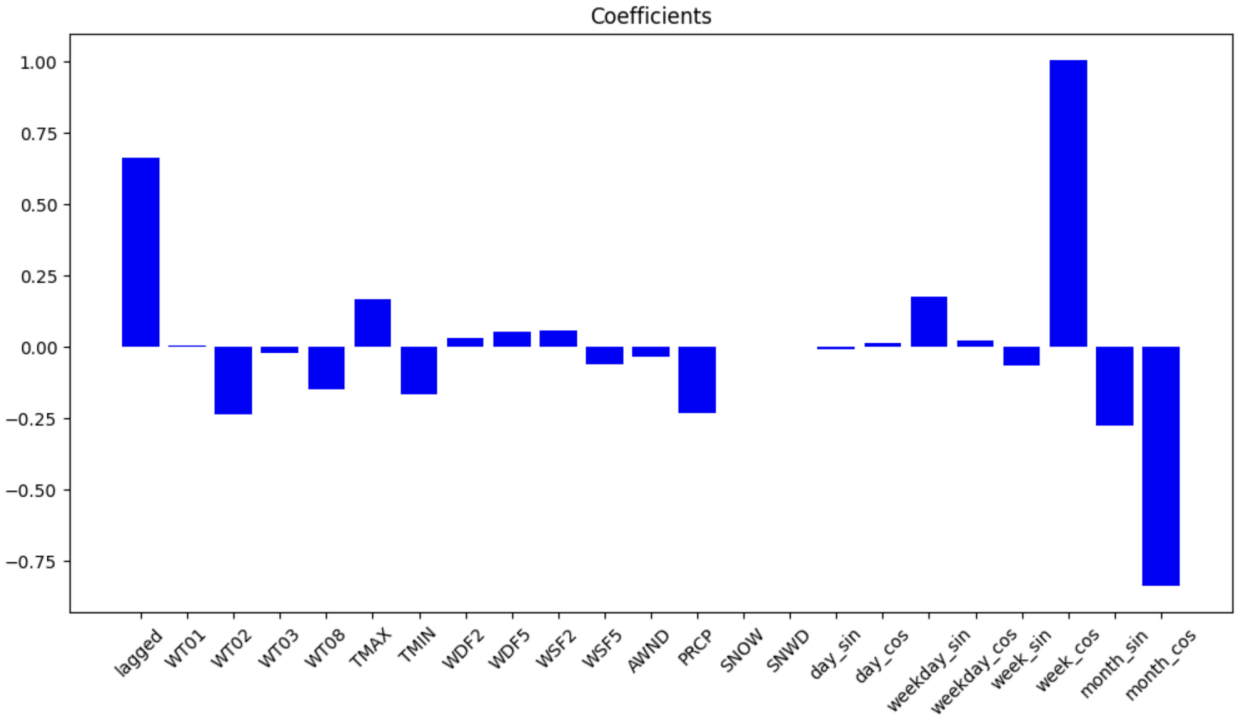

Figure 9: Linear Regressor Coefficients

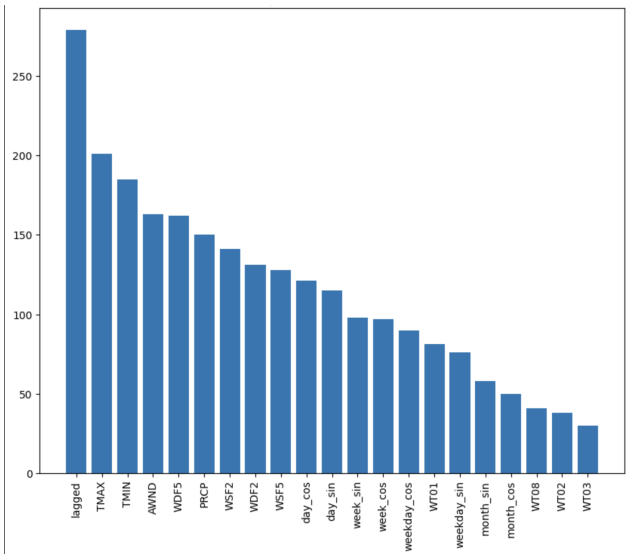

Figure 10: Feature Importance with Feature Names

6 Deep Learning Approach

6.1 Deep Learning Approach 1

Focused on predicting bike rental patterns using a neural network model. The model was trained with features such as weather, geographical information, and bike ride characteristics.

Result The model was trained and tested, demonstrating its effectiveness in predicting bike rental patterns with a certain degree of accuracy.

- During training, the model's loss steadily decreased with each epoch.
- The final test MAE was calculated to be approximately 0.766, indicating that the model's predictions were, on average, within 0.766 units of the actual values. This suggests that the GNN model was effective in predicting bike rental patterns with a reasonable degree of accuracy.

Conclusion The Deep Learning Approach 1, which utilized a Graph Neural Network (GNN) for predicting bike rental patterns, demonstrated promising results. The model effectively leveraged spatial and temporal information along with weather data to make accurate predictions.

Here are some key takeaways and conclusions:

1. *Predictive Accuracy*: The model achieved a relatively low test MAE, suggesting that it performed well in predicting the number of bike rentals. This is a crucial metric for bike-sharing platforms to optimize bike allocation and service availability.
2. *Feature Engineering*: The extensive feature engineering, including the incorporation of weather data, encoding of categorical features, and cyclical encoding of temporal features, contributed to the model's ability to capture complex patterns in the data.
3. *Graph Representatio*n: Representing the bike-sharing data as a graph allowed the model to take advantage of the spatial relationships between stations and the flow of bike rentals between them.
4. *Potential for Real-World Deployment*: The trained model could be deployed in a real-world bike-sharing system to assist in decision-making, such as station maintenance, bike redistribution, and resource allocation.
5. *Room for Improvement*: While the model demonstrated effectiveness, there may be opportunities for further improvement, such as exploring different architectures, hyperparameter tuning, and incorporating additional data sources.

In summary, the Deep Learning Approach 1 successfully applied GNNs to predict bike rental patterns, providing valuable insights for bike-sharing services. Further refinement and experimentation could lead to even more accurate and valuable predictions for optimizing bike-sharing operations.

6.2 Deep Learning Approach 2

Implemented a Graph Neural Network (GNN) to predict the number of rides between different stations. The model used daily variations in bike rentals for its predictions.

Our GNN-based bike rental prediction model consists of the following key components:

1. Graph Neural Network (GNN): We designed a two-layer Graph Convolutional Network (GCN) architecture to model relationships between bike stations. The model takes station information as input and predicts the number of rides between stations.
2. Data Preparation: We preprocessed the dataset to create node features (one-hot encoded for simplicity) and constructed graphs based on daily rental patterns. Edge features were computed during training as averages of connected node features.
3. Training and Optimization: The model was trained using Mean Absolute Error (MAE) as the loss function and the Adam optimizer. We also experimented with other optimizers such as SGD and RMSprop to assess their performance.

6.2.1 Model 1: Single Graph Model

Training Progress:

- The single graph model was trained for 10 epochs.
- The Adam optimizer achieved the best performance with an average loss of approximately 0.2128 after 10 epochs.
- The SGD optimizer exhibited more significant fluctuations in the loss but still achieved a reasonable average loss of about 0.2212.
- The RMSprop optimizer performed well, with an average loss of approximately 0.2089.

Visualization - Original vs. Predicted Rides:

- Heatmaps of adjacency matrices were generated to visualize the actual and predicted ride counts.
- The heatmap of the original rides revealed the patterns of bike rentals between stations.

Conclusion for Model 1:

- The GNN model successfully captured the relationships between bike stations.
- The Adam optimizer demonstrated the best training performance.
- The model provided valuable insights into the bike rental patterns for the entire dataset.

6.2.2 Model 2: Daily Analysis

Training Progress:

- We extended our approach to create and analyze separate graphs for each day.

- The model was trained for 10 epochs on multiple daily graphs.

Visualization - Original vs. Predicted Rides (Specific Day):

- For Model 2, we selected a specific day's graph to assess the model's performance on a daily basis.
- Heatmaps were generated to visualize the actual and predicted ride counts for the chosen day.

Conclusion for Model 2:

- The GNN model adapted well to daily variations in bike rentals.
- Training on multiple daily graphs allowed the model to capture daily patterns effectively.
- The model demonstrated the ability to provide daily-level insights into bike rental patterns.

6.2.3 Overall Evaluation

- Both Model 1 and Model 2 successfully learned and predicted bike rental patterns between stations.
- Model 2, with daily analysis, provides a more granular understanding of bike rental variations.

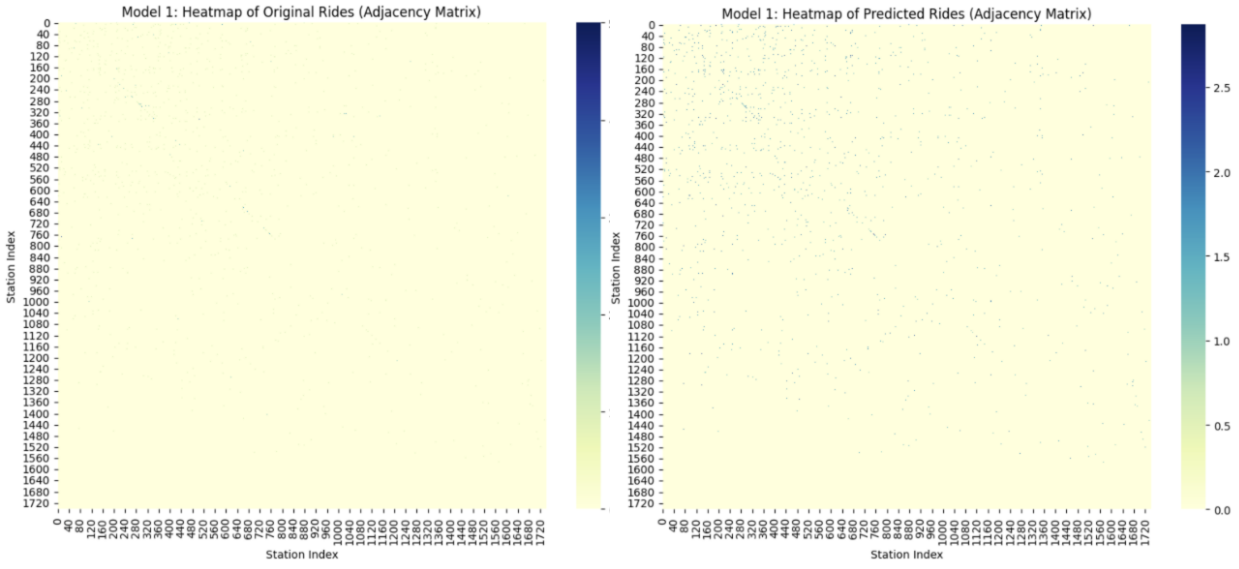

Figure 11: Model 1 Heatmaps

7 Conclusion

The project successfully applies a range of data science and machine learning techniques to provide valuable insights into bike rental patterns. It demonstrates the potential of these techniques in urban planning and transportation management.

In summary, our Graph Neural Network (GNN) approach has proven to be effective in predicting bike rental patterns between stations. By extending the model to analyze daily variations, we have gained a deeper understanding of the dynamics within the bike-sharing system. The choice of optimizer and daily-level analysis can significantly impact model performance, providing actionable insights for improving bike-sharing services. Further research and development in this area hold the potential to enhance the efficiency and sustainability of urban mobility systems.

8 Acknowledgments

We express our deepest appreciation to Professor Richard Sowers and the Department of Industrial and Systems Engineering at the University of Illinois at Urbana-Champaign for their continuous support and expert guidance throughout this project. Their profound knowledge and insight have been instrumental in navigating the complexities of this study. Special thanks are also extended to the various data providers and technical platforms, especially AWS S3 services, for their crucial role in data handling and retrieval. This project, a testament to collaborative learning and academic endeavor, would not have been possible without the invaluable contributions of all involved. We are immensely grateful for the opportunity to engage in this enriching educational experience, which has significantly contributed to our understanding and application of deep learning in real-world scenarios.

9 Contact

- Advika Pattiwar - linkedin.com/in/advika-pattiwar
- Dhruv Borda - linkedin.com/thebordadhruv
- Hrithik Rathi - linkedin.com/in/hrithik-rathi
- Suvrata Gayathri Kappagantula - linkedin.com/in/gayathrikappagantula