IE434- Deep Learning

Project Report

NYC Citi Bike Rentals

Deep Dive – 9

Selvamani Subramaniam
Deepanshu Malhothra
Prannoy Kathiresan
Vishal Dayalan

Executive Summary

The project aims to predict the most popular Lyft Bike destination stations in New York City based on various features related to bike rentals. The objective is to optimize bike placement and distribution, enhancing user convenience and potentially increasing the efficiency of the bike-sharing system.

The project spans various milestones, including data extraction, exploration, baseline learning, deep learning, and feature importance analysis. The various learnings from the course have been incorporated into executing each part of the project to achieve the milestones and successfully complete it.

Problem Statement

The key challenge addressed by this project is predicting the demand at specific Lyft Bike destination stations in Jersey City. By leveraging historical trip details, the goal is to forecast the demand for a given station at a particular day and time. This prediction is crucial for planning and optimizing bike placement, ensuring that bikes are available where and when they are needed the most.

The Project aims to predict popular bike destination stations aids in optimizing bike placement and distribution, enhance the convenience of users by ensuring bikes are available at high-demand locations, increase the operational efficiency of the bike sharing system, potentially reducing downtime and improving service reliability.

Data Description

The Dataset employed has the following features:

- `rideable_type`: Type of bike used (classic_bike, electric_bike).
- `start_station_name`: Name of the starting station.
- `end_station_name`: Name of the destination station.
- `start_lat`: Latitude of the starting station.
- `start_lng`: Longitude of the starting station.
- `end_lat`: Latitude of the destination station.
- `end_lng`: Longitude of the destination station.
- `member_casual`: Type of user (Customer, Member).
- `start_month`: Month of the start date.
- `stop_month`: Month of the stop date.
- `start_day_of_week`: Day of the week for the start date.
- `stop_day_of_week`: Day of the week for the stop date.
- `start_hour`: Hour of the day for the start time.
- `end_hour`: Hour of the day for the end time.
- `tripduration_minute`: Duration of the trip in minutes.

These features capture various aspects of bike rides, including the type of bike, start and end locations, user type, time-related information, and trip duration.

The sample data set is displayed below.

	rideable_type	start_station_name	end_station_name	start_lat	start_lng	end_lat	end_lng	member_casual	start_month	stop_month	start_day_of_week	stop_day_of_week	start_hour	end_hour	tripduration_minute	demand
0	classic_bike	Mama Johnson Field - 4 St & Jackson St	South Waterfront Walkway - Sinatra Dr & 1 St	0.755607	0.747216	0.400846	0.304860	Customer	3	3	4	4	15	15	-0.033168	537
1	electric_bike	Baldwin at Montgomery	Grove St PATH	0.353913	0.378314	0.323884	0.228441	member	3	3	4	4	16	16	-0.037164	631
2	electric_bike	Baldwin at Montgomery	Grove St PATH	0.353913	0.378314	0.323884	0.228441	member	3	3	6	6	17	17	-0.021181	565
3	classic_bike	Baldwin at Montgomery	Grove St PATH	0.353913	0.378314	0.323884	0.228441	member	3	3	6	6	15	15	-0.029172	605
4	classic_bike	Baldwin at Montgomery	Grove St PATH	0.353913	0.378314	0.323884	0.228441	member	3	3	4	4	12	12	-0.013190	414
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1521595	electric_bike	Madison St & 1 St	Columbus Dr at Exchange Pl	0.665912	0.758533	0.311868	0.279800	Customer	9	9	4	4	21	22	-0.013190	147
1521596	classic_bike	Monmouth and 6th	Bergen Ave & Stegman St	0.394022	0.612992	0.266324	0.011277	member	9	9	6	6	17	18	-0.009194	24
1521597	electric_bike	4 St & Grand St	Madison St & 10 St	0.737421	0.822512	0.458187	0.264579	member	9	9	2	2	16	16	-0.041159	147
1521598	classic_bike	4 St & Grand St	Madison St & 10 St	0.737421	0.822512	0.458187	0.264579	member	9	9	1	1	11	11	-0.045155	82
1521599	classic_bike	4 St & Grand St	Madison St & 10 St	0.737421	0.822512	0.458187	0.264579	Customer	9	9	4	4	20	21	-0.037164	115

1521600 rows × 16 columns

Data Preprocessing and Extraction

Data preprocessing and extraction play pivotal roles in the success of any data-driven project. These crucial steps ensure that raw, often heterogeneous data is transformed into a clean, standardized format that is conducive to analysis and modeling. Data preprocessing involves cleaning, formatting, and organizing the data, addressing issues such as missing values and outliers. This enhances data quality, making it more reliable for subsequent analyses.

Furthermore, extraction involves obtaining data from various sources, often in diverse formats, and consolidating it into a unified dataset. This consolidation simplifies the handling of data, promotes ease of access, and enhances overall efficiency. The significance of these processes lies in their ability to set the stage for meaningful insights, accurate modeling, and informed decision-making. A well-preprocessed and extracted dataset not only facilitates exploratory data analysis and feature engineering but also ensures that predictive models are built on a solid foundation, ultimately leading to more robust and reliable results.

The following steps were involved in the data preprocessing and extraction of our data set.

Step 1: Downloading and Extracting the Dataset

The process began with downloading a compressed zip file containing 24 months of historical data for JC 2021 and JC 2022. Once downloaded, the file was extracted to access the raw data.

Step 2: Data Conversion and Merging

Converted to CSVs: The extracted data, which was in a raw format, was converted into CSV files for easier handling and manipulation.
Merged to a Single CSV: To facilitate comprehensive analysis, the individual CSV files were merged into a single dataset. This consolidation simplified the subsequent preprocessing steps.

Step 3: Format Conversion

Converted to Pickle File: The unified dataset was then converted to a pickle file. Pickle files are binary files that efficiently store Python objects, offering faster loading times compared to other formats like CSV.

Step 4: Train-Test Split

Split into Train and Test Sets: To evaluate the model's performance effectively, the dataset was split into training and testing sets. The training set was used to train the predictive model, while the testing set served to assess the model's accuracy on new, unseen data.

Step 5: Post-COVID Focus

Focused on Post-COVID Impact: The dataset was analyzed with a specific focus on the post-COVID period. This involved exploring and understanding the variations and trends in bike rentals after the COVID-19 pandemic, considering potential changes in commuting patterns and user behavior.

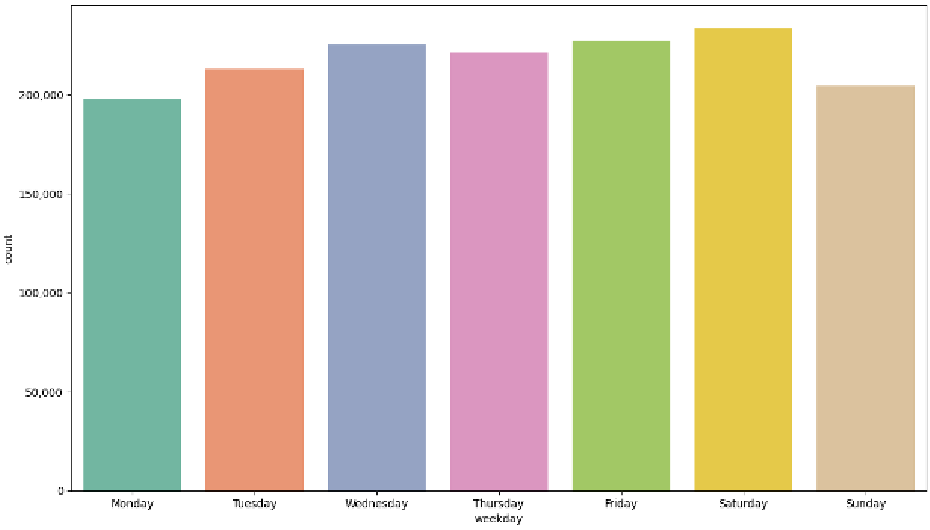
The data extraction and preprocessing and steps ensured that the dataset was in a suitable format for further analysis and model development. The conversion to CSVs and pickle files, along with the train-test split, set the stage for uncovering insights into the impact of public commute in the post-COVID period.This well-prepared dataset could then be used for exploratory data analysis, feature engineering, and the development of predictive models aimed at understanding and forecasting demand at Lyft Bike destination stations.

Data Exploration & Visualization

Data visualization and exploration are integral components of the data analysis process, offering valuable insights into patterns, trends, and relationships within the dataset. In our project focused on predicting demand at Lyft Bike destination stations in Jersey City, the exploration phase was crucial for understanding the dynamics of bike rentals.

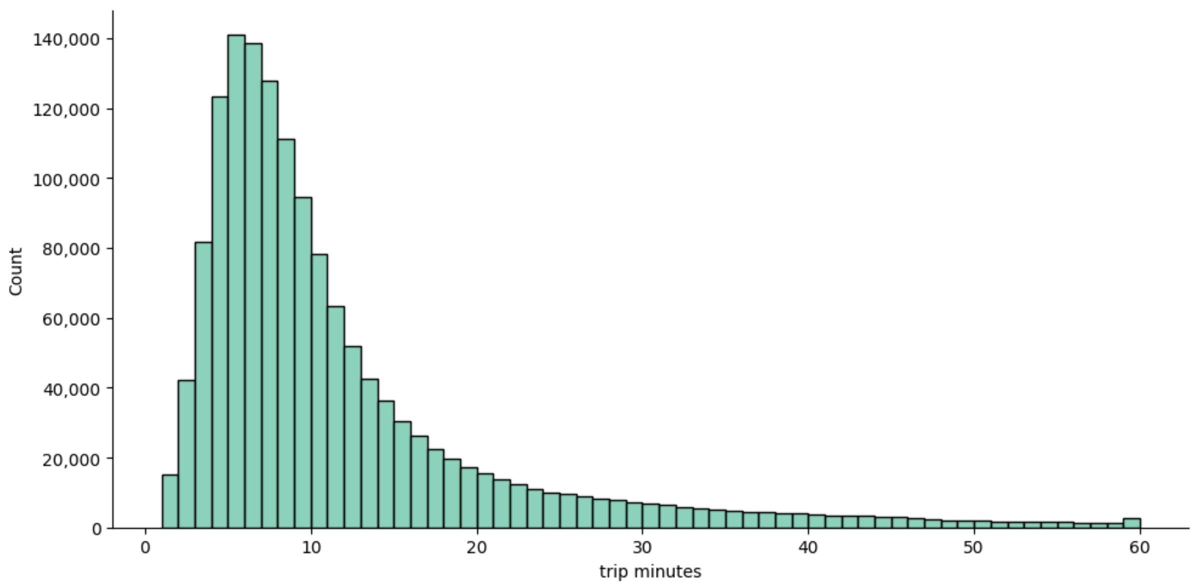
Day-Wise Frequency

A bar chart was employed to visualize the frequency of trips for each day. Notably, the 15th of each month stood out with over 40,000 trips, indicating a recurring peak. Additionally, weekdays exhibited higher usage compared to weekends, emphasizing the importance of considering the day of the week in demand forecasting and resource allocation.



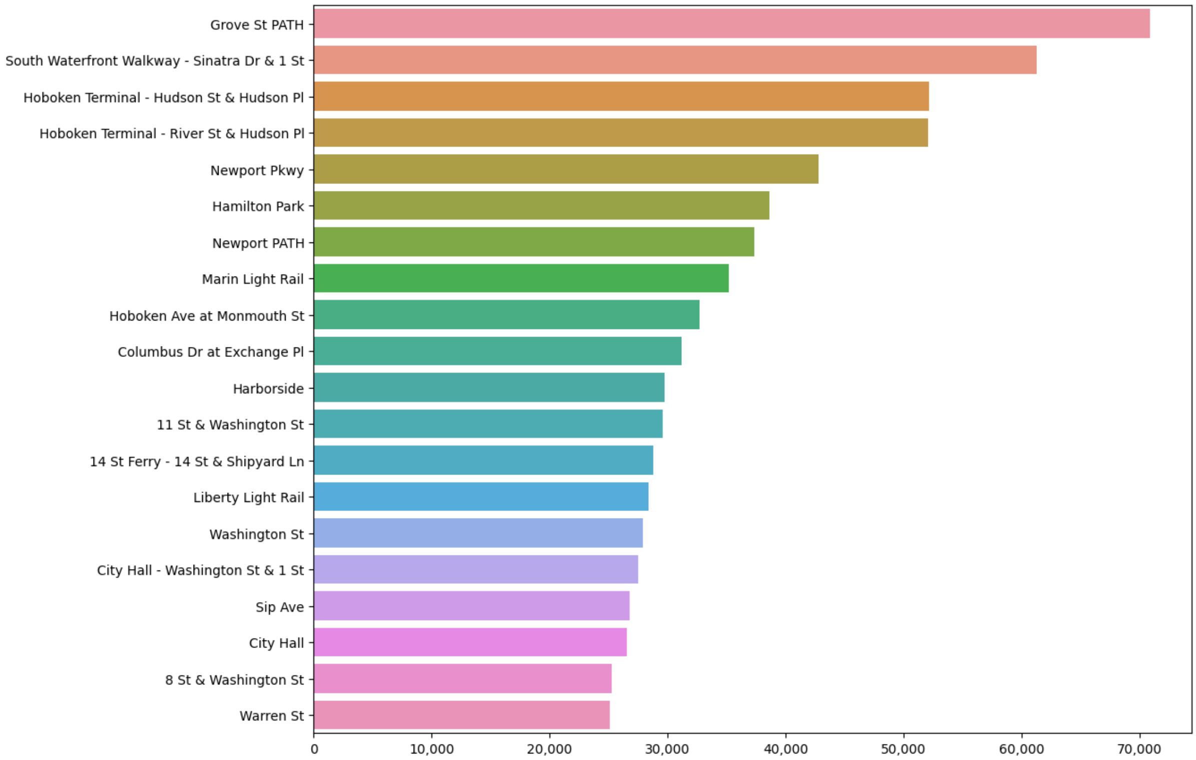
Trip Duration Frequency

To grasp the overall picture, we visualized the frequency of trip durations, plotting trip duration on the x-axis and its corresponding count on the y-axis. The graph revealed that trips lasting less than 10 minutes accounted for the majority of rides. Specifically, 6-minute trips were the most frequent (~140,000), followed by 7-minute (~139,900) and 8-minute (~127,000) trips. This observation suggests a preference for short trips, indicating that start and end locations are typically close.



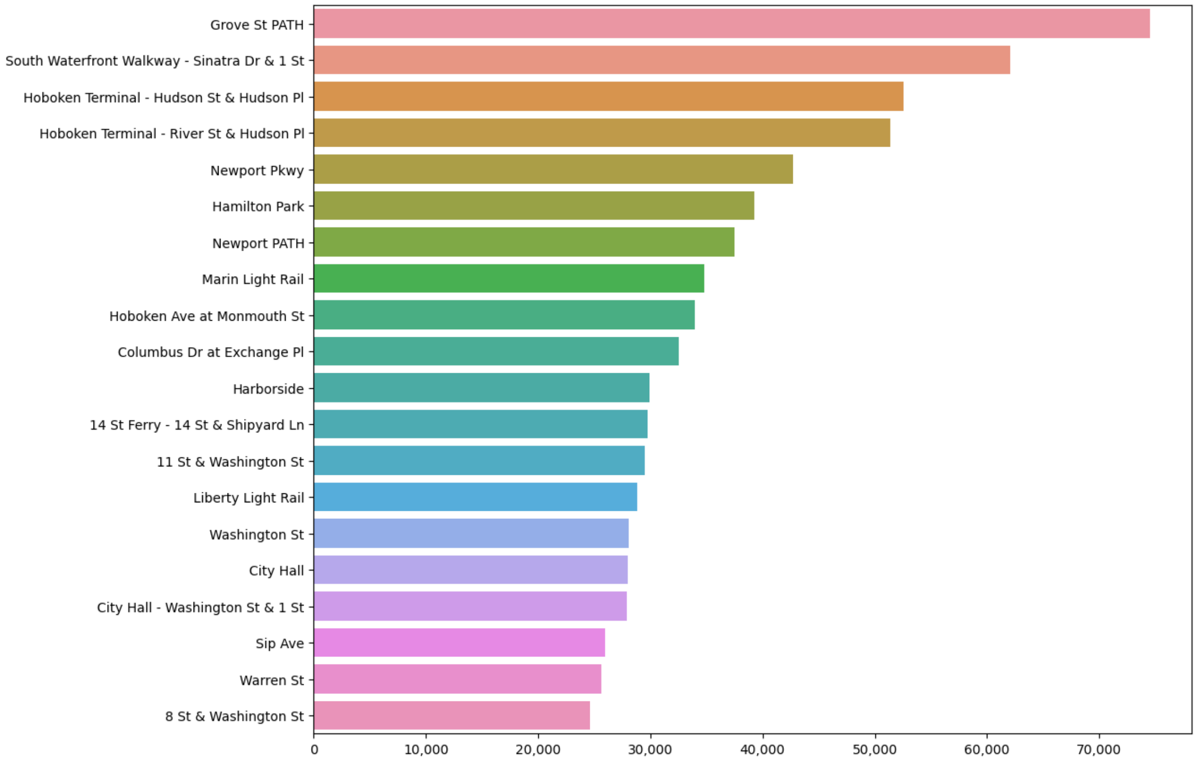
Start-Station Frequency (Top 20)

Exploring the start-station frequency for the top 20 stations using a horizontal bar chart highlighted "Grove St PATH" as the most frequently accessed start station. Focusing on these top stations is crucial for strategically placing and distributing Citi-bikes, optimizing accessibility and user convenience.



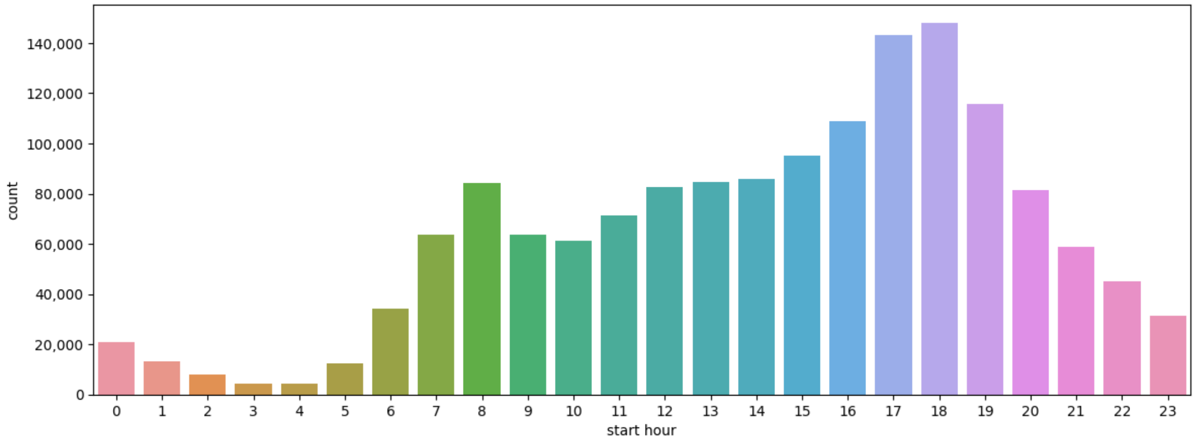
End-Station Frequency (Top 20)

A similar approach was taken for end-station frequency, revealing "Grove St PATH" as the top dropoff location. Concentrating on the top 20 end stations aids in planning the effective placement and distribution of bikes, aligning with user demand patterns.



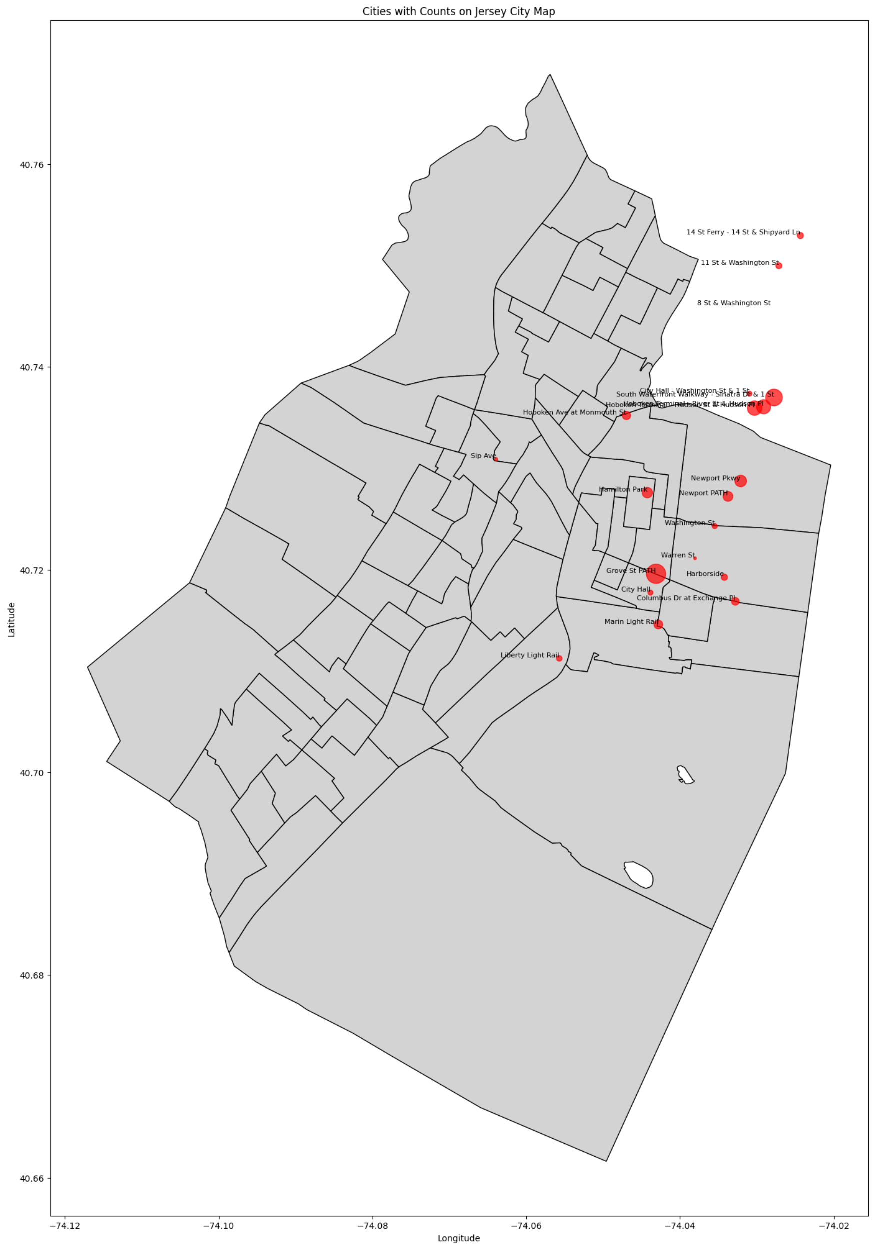
Start-Time Frequency

We further visualized the frequency of trip start times using a bar graph. The data showed that over 140,000 trips commenced between 1800 - 1900 hrs, with another substantial peak (~90,000 trips) occurring around 8 am. This information is crucial for optimizing bike placement during peak hours, ensuring efficient distribution based on demand patterns.



Jersey City Map Inference

The map of Jersey City displayed end locations of rides, with red circles representing the relative count of observations. The top 20 sites where rides concluded highlighted "Grove St PATH" as the most popular dropoff location. This spatial insight is invaluable for strategically locating bike stations to meet user demand effectively.



In summary, data visualization and exploration provide key insights into trip patterns, temporal trends, and popular locations. These findings are instrumental in shaping strategies for bike placement, resource optimization, and enhancing the overall efficiency of the Lyft Bike rental system in Jersey City.

Baseline Learning

We developed a baseline model using a Simple Neural Network to predict the demand at Lyft Bike destination stations in Jersey City, NY. This initial model aimed to lay the groundwork for more advanced predictive models by employing a Simple Neural Network. The key features considered for prediction included rideable type, geographical coordinates, member type, and various temporal aspects. The following paragraphs provide a detailed overview of the baseline learning process and the model's architecture.

Data Preprocessing

The data preprocessing steps were crucial in preparing the dataset for model training. Categorical features, specifically 'rideable_type' and 'member_casual,' underwent one-hot encoding to transform them into a format suitable for machine learning. Numerical features were standardized using Standard Scaler to ensure that no single feature dominated the model training. Additionally, the data was grouped based on 'start_day_of_the_week,' 'start_hour,' and 'end_station_name.' This grouping facilitated the creation of a calculated field called 'demand,' essential for the subsequent supervised learning task.

Model Architecture

The selected features for the model encompassed a range of parameters, including rideable type, spatial coordinates, member type, and temporal information.

The selected features for the model included 'rideable_type,' 'start_lat,' 'start_long,' 'end_lat,' 'end_long,' 'member_casual,' 'start_month,' 'stop_month,' 'start_day_of_the_week,' 'stop_day_of_the_week,' 'start_hour,' 'stop_hour,' and 'trip_duration_minute.'

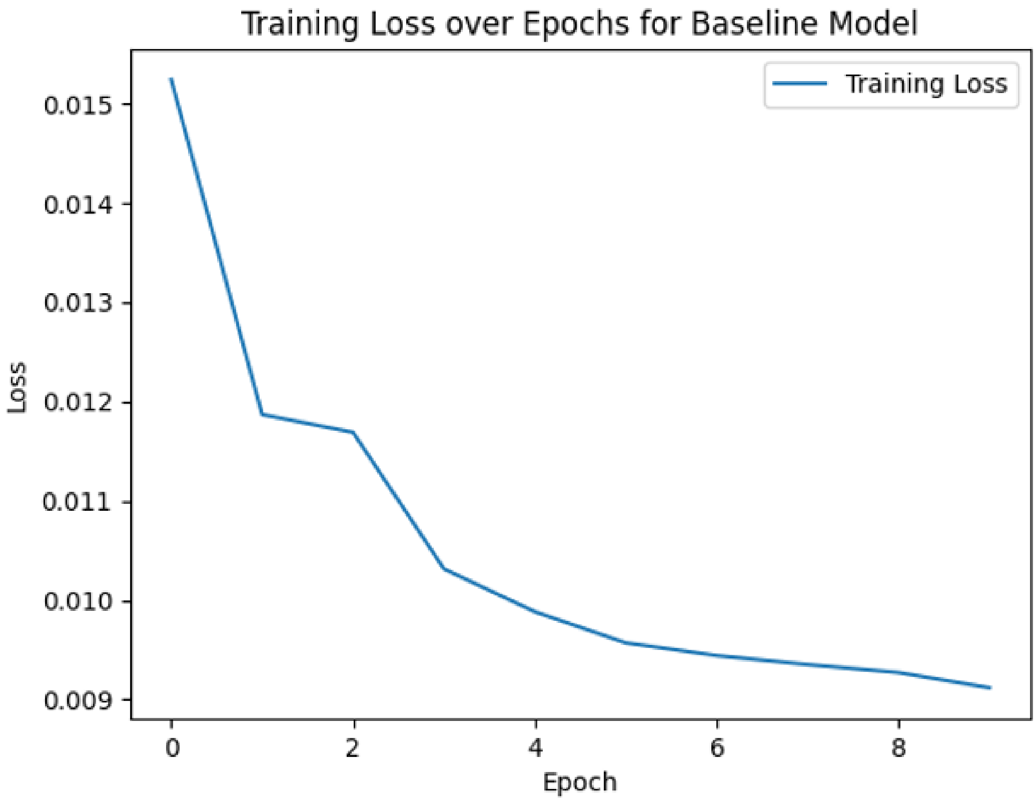
The neural network architecture employed for the baseline model consisted of two layers. It incorporated a fully connected linear layer, followed by a Rectified Linear Unit (ReLU) activation function, and another fully connected linear layer. These architectural choices aimed to capture complex relationships within the data and make accurate predictions regarding the demand for bike stations.

Hyperparameters and Loss Function

Critical hyperparameters, including learning rate (lr), epochs, batch size, and optimizer, were set to {0.001, 10, 32, ADAM}. These values were determined through iterative experimentation to achieve optimal model performance. The loss function chosen for the baseline model was Mean Squared Error (MSE), measuring the squared residual between predicted and actual demand values. This loss function provided a quantitative assessment of the model's accuracy in predicting the demand for Lyft Bike destination stations.

Baseline Learning Performance

The baseline model demonstrated promising performance metrics. The Mean Squared Error (MSE) on the test set was calculated as 0.0015, indicating a relatively small squared residual between the actual and predicted demand values. Moreover, the loss value on the training set exhibited a reduction from 0.0152 to 0.0091 over 10 epochs. This reduction in loss values highlighted the model's capacity to improve its predictions with continued training.



In conclusion, Milestone-2 successfully established a solid baseline model for predicting bike demand, providing a foundation for subsequent advancements. The implementation of this model sets the stage for exploring more sophisticated techniques in Milestone-3, where we will delve into the application of a Gated Recurrent Unit (GRU) in our predictive modeling journey.

Deep Learning

For the Deep Learning milestone, we implemented a Gated Recurrent Unit (GRU) for predicting the demand for Lyft Bike end stations based on 16 selected features. As a variant of recurrent neural networks (RNNs), the choice of the GRU model was informed by its suitability for handling sequential data, effective capture of temporal dependencies, and learning patterns from historical sequences. Models with distinct hyperparameters were trained and evaluated.

GRU Model Implementation

The Gated Recurrent Unit (GRU) was selected as the model of choice due to its capabilities in handling sequential data and capturing temporal dependencies effectively. Being a variant of recurrent neural networks (RNNs), the GRU is designed to address the challenge of learning long-range dependencies within sequential data. The model was applied to predict the demand for end stations in the bike-sharing dataset, leveraging features such as rideable type, geographical coordinates, member type, and temporal aspects.

Training Process

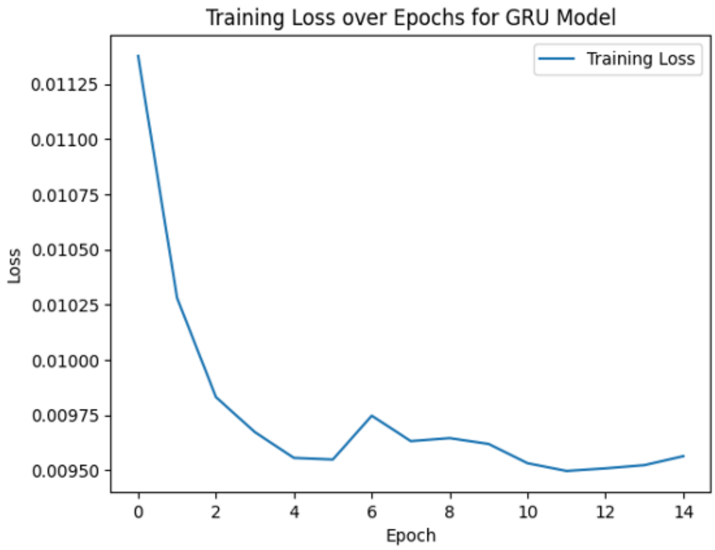
The training process involved multiple epochs, with each epoch updating the model's parameters to enhance its predictive capabilities. The displayed loss values for each epoch provide insights into the model's convergence and performance. The Mean Squared Error (MSE) on the test set was used as a metric to evaluate the model's accuracy in predicting the demand for end stations.

Hyperparameter Tuning

Several experiments were conducted to fine-tune the hyperparameters of the GRU model, including the hidden size, learning rate, batch size, and the number of epochs. These hyperparameters play a crucial role in determining the model's performance and convergence. Through iterative experimentation, optimal settings were identified to achieve the desired level of accuracy within a reasonable training duration.

Hyper-parameters	{0.001, 10, 32, 50}

Hyper-parameters	{0.1, 15, 64, 100}



Optimizer Exploration

The impact of different optimizers on the model's performance was explored, with a specific focus on comparing the AdamW and Adam optimizers. AdamW, chosen for its application of weight decay to update steps, was initially employed for hyperparameter tuning. Subsequently, the Adam optimizer was utilized with identical parameters to investigate its performance in comparison to AdamW.

Hyper-parameters	{0.001, 10, 128, 100}

Conclusion

After extensive experimentation and hyperparameter tuning, the GRU model demonstrated optimal performance with specific settings: a hidden size of 100, a learning rate of 0.001, a batch size of 128, and training conducted over 10 epochs. These configurations were chosen considering the project's time constraints, ensuring that the model achieves accurate predictions within a reasonable time frame. The exploration of different optimizers, specifically AdamW and Adam, provided insights into their impact on the model's convergence and performance, contributing to the overall refinement of the predictive model.

Feature Importance

The nature of sequential data processing in Gated Recurrent Unit (GRU) models posed challenges for traditional feature importance techniques. The spread of input features across multiple time steps adds complexity, making it more challenging to straightforwardly apply methods such as gradient-based techniques or permutation importance to GRUs. Unlike feedforward models, where feature importance can be inferred directly from gradients or permutation experiments, GRUs require alternative approaches due to their sequential and dynamic nature.

Linear Regression as an Alternative

Given the challenges of directly applying traditional feature importance methods, an alternative approach involves using linear regression. In this method, coefficients of a linear regression model are utilized to calculate weights for each feature. The rationale behind this approach lies in identifying the correlation between features and the target variable. The higher the weight (magnitude) assigned to a feature, the more significant its correlation, indicating its importance in predicting the target.

Weight Interpretation

It's crucial to note that the weights derived from linear regression are sign-invariant, meaning that their direction (positive or negative) doesn't inherently indicate the nature of correlation. Instead, the magnitude of the weight is the key factor. Higher magnitudes signify stronger correlations, implying more substantial importance for the corresponding features.

Feature Importance Ranking

```
Rank 1: end_lng : 0.8471573956039891
Rank 2: start_lat : 0.027009764661260177
Rank 3: member_casual_member : 0.0045341548531796574
Rank 4: end_hour : 0.002035528124128154
Rank 5: rideable_type_classic_bike : 0.0018088129201581354
Rank 6: stop_day_of_week : 0.0011883416346306647
Rank 7: rideable_type_electric_bike : 0.0008314827805634868
Rank 8: stop_month : 0.0005206808500088228
Rank 9: start_month : -0.0007151188407709952
Rank 10: tripduration_minute : -0.0010775361407049779
Rank 11: start_hour : -0.0022935633074943652
Rank 12: rideable_type_docked_bike : -0.002640295700721454
Rank 13: start_day_of_week : -0.0031733774370082827
Rank 14: member_casual_Customer : -0.004534154853177073
Rank 15: start_lng : -0.06880580901285406
Rank 16: end_lat : -0.6807360669038712
```

The output of the linear regression analysis provides a feature importance ranking. In the presented example, the top-ranked features based on their weights are listed. For instance, 'end_lng' holds the highest weight, signifying its substantial correlation and importance in predicting the target. On the other hand, features with negative weights, such as 'start_hour' and 'start_day_of_week,' suggest a negative correlation, indicating their potential role in predicting lower demand.

RESULTS AND OBSERVATIONS

Baseline Model with Simple Neural Network

The baseline model aimed to predict the demand for Lyft Bike end stations using a simple neural network with two fully connected layers and ReLU activation. The features considered included rideable type, geographical coordinates, member type, months, days of the week, and trip duration. The mean squared error (MSE) on the test set was observed to be 0.0015. This baseline served as a reference point for evaluating the performance of more complex models.

Deep Learning Notebook with GRU

The implementation of a Gated Recurrent Unit (GRU) for demand prediction introduced a model capable of handling sequential data and capturing temporal dependencies. Hyperparameter tuning was conducted to optimize the model, resulting in an MSE of 0.0090 on the test set. Further experiments explored the impact of different optimizers, with AdamW showing faster convergence.

Feature Importance Analysis

Feature importance analysis provided insights into the factors contributing to the model's predictions. The GRU model identified "end_lng" as the most crucial feature, followed by "start_lat" and "member_casual_member." Interestingly, geographical coordinates, member type, and temporal features played significant roles, highlighting their influence on predicting demand for Lyft Bike end stations.

Observations

Sequential data processing, inherent in GRU models, presented challenges for traditional feature importance techniques.
Linear regression was employed as an alternative for feature importance, revealing the weighted influence of each feature.
"end_lng," representing the end longitude, emerged as the most critical feature for demand prediction, indicating the geographical significance of end station locations.
The baseline model provided a reference point, showcasing the improvement achieved with the more complex GRU model.
The choice of optimizer, such as AdamW, influenced the training speed and convergence, showcasing the importance of optimizer selection in model performance.

Challenges

Sequential Data Processing with GRU

Implementing a Gated Recurrent Unit (GRU) for predicting demand introduces the challenge of effectively processing sequential data. The spread of input features across multiple time steps complicates the application of traditional feature importance techniques, making it necessary to explore alternative methods to interpret and understand the model's decisions.

Feature Importance in GRU:

GRU models, being inherently designed for sequential data, pose challenges for conventional feature importance analysis techniques. The nature of GRUs involves capturing dependencies across multiple time steps, making it non-trivial to directly apply techniques like gradient-based methods or permutation importance. This challenge necessitates creative solutions to extract meaningful insights into feature importance.

Significance of Geographical Coordinates:

The feature importance analysis revealed the crucial role of geographical coordinates, specifically "end_lng," in predicting demand for Lyft Bike end stations. While this information is valuable, it presents challenges in interpretation, as the significance of longitude values may not be immediately intuitive. Further exploration and visualization may be required to extract actionable insights from these geographical features.

Model Complexity and Interpretability:

The adoption of more complex models, such as the GRU, enhances predictive capabilities but may compromise interpretability. Balancing model complexity with the need for transparent and interpretable results poses a common challenge. Striking the right balance becomes crucial, especially in scenarios where clear and explainable insights are essential for decision-making.

Hyperparameter Tuning and Time Constraints:

Experimenting with hyperparameters to optimize the model's performance is a critical step. However, this process must be conducted within practical time constraints. The challenge lies in identifying the most effective configurations for hyperparameters that ensure model accuracy while adhering to project timelines.

Impact of Optimizer Choice:

The choice of optimizer, as demonstrated in the exploration of AdamW and Adam optimizers, presents challenges in terms of understanding their impact on training speed, convergence, and overall model performance. Identifying the most suitable optimizer involves iterative experimentation and careful consideration of trade-offs.

Limited Feature Interpretability in GRU:

The inherently complex architecture of GRU models, designed for sequential data, limits the interpretability of individual features. Understanding the contribution of specific features to model predictions becomes more intricate, requiring innovative approaches to extract meaningful insights.

Scaling to Large Datasets:

As the dataset grows in size, scaling the GRU model to handle large volumes of sequential data introduces challenges related to computational resources, memory requirements, and training times. Optimizing the model's efficiency while maintaining predictive accuracy becomes a critical consideration.

Conclusion

In the journey of exploring and predicting demand for Lyft Bike end stations, we embarked on a comprehensive analysis that involved data preprocessing, baseline modeling, deep learning with GRU, hyperparameter tuning, and feature importance analysis. The insights gained and challenges encountered shed light on the complexities of working with sequential data and advanced deep learning models in the context of a bike-sharing dataset.

Model Performance:

The implementation of a Gated Recurrent Unit (GRU) demonstrated its effectiveness in capturing temporal dependencies within sequential data. The model exhibited optimal performance with carefully tuned hyperparameters, achieving desirable accuracy and convergence within a reasonable number of training epochs. The iterative exploration of optimizers provided valuable insights into their impact on model training.

Hyperparameter Tuning:

Fine-tuning hyperparameters revealed the critical role of settings such as hidden size, learning rate, batch size, and epochs in achieving optimal model performance. Balancing the trade-offs between model complexity and computational efficiency is essential, especially when operating within time constraints.

Feature Importance:

The feature importance analysis, while challenging with GRU models, highlighted the significance of geographical coordinates, specifically "end_lng," in predicting demand. However, the inherent complexity of the model makes interpreting feature contributions a nuanced task, requiring a careful balance between model accuracy and interpretability.

Challenges Encountered:

Challenges, including the sequential nature of GRUs, limited feature interpretability, and the impact of geographical coordinates, underscored the intricacies of working with advanced deep learning architectures. Addressing these challenges necessitates a combination of domain expertise, innovative methodologies, and a nuanced understanding of the dataset.

Next Steps:

Moving forward, further exploration could involve refining feature engineering strategies, considering additional external factors, and enhancing interpretability through visualization techniques. Additionally, the deployment of the trained model for real-time predictions and continuous monitoring would contribute to the practical application of the insights gained.

The journey through data preprocessing, baseline modeling, deep learning, and feature analysis provided a holistic view of predicting demand in a post-COVID era for Lyft Bike end stations. The insights gained contribute not only to the understanding of bike-sharing dynamics but also to the broader application of advanced modeling techniques in the realm of transportation and demand prediction.