Forecasting Station Demand of NYC Citi Bike Data

Dylan Aubrey, Harsh Govindji, Kevin Baer, Joshua Rusit

Introduction:

New York City is one of the largest and most populated metropolitan areas in the world. Millions of residents and tourists rely heavily on public transportation to navigate the city every day. Part of the public transportation ecosystem is a shared rental bike service called the Citi Bike Program. This is a station-based service that allows users to check out a bike at a docking station and return it to another one near their destination. This system provides an eco-friendly way for people to commute short distances without having to rely on cars or crowded public transit.

Understanding and forecasting the demands of the station is critical to maintaining operational efficiency across the Citi Bike network. Accurate predictions ensure there are enough bikes at each location to match citizen needs while preventing overcrowding or shortages. Several factors, like commuting patterns, weather conditions, and tourism, significantly influence demand at individual stations. By leveraging neural networks alongside traditional machine learning methods, we aim to forecast station demand based on the time of the day by analyzing historical trip patterns and integrating external weather data.

Data Description:

We began with the original Citi Bike trip dataset for initial modeling of the problem. We then integrated external weather data to add contextual information and improve the accuracy of our predictions. Below is a description of both datasets used in the project:

Citi Bike Dataset

This dataset contains 5,287,447 entries from September 2025. Each row represents a single trip taken by a user. The dataset includes temporal, geographic, and user-related information that allows us to analyze usage patterns across different stations and times. Below is the information we have:

Column Name	Description
ride_id	Unique identifier for rides
rideable_type	Identifier for bike type: Electric or Classic
started_at	Timestamp for start of trip
ended_at	Timestamp for end of trip
start_station_name	Name of starting station
start_station_id	Unique identifier for station
end_station_name	Name of ending station
end_station_id	Unique identifier for station
start_lat	Starting latitude coordinate of trip
start_lng	Starting longitude coordinate of trip
end_lat	Ending latitude coordinate of trip

end_long	Ending longitude coordinate of trip
member_casual	Type of member: Member or Casual

Weather Dataset

This dataset contains hourly weather observations for New York City during the same time period. Weather conditions are known to significantly impact bike-sharing demand, with factors like temperature, precipitation, and wind affecting user decisions. Below is a description of the columns:

Column Name	Description
date	Timestamp of weather
temperature_2m	Temperature at given time
precipitation	Amount of precipitation in meters
snowfall	Amount of snowfall (cm)
snowdepth	Depth of snow on the ground (cm)
wind_speed_10m	Speed in mph

To create our final modeling dataset, we aggregated trip data at the station and hour level to calculate demand, defined as the number of trips starting from each station per hour. We converted all timestamps to Eastern Time (America/New_York) to ensure proper alignment between the Citi Bike trips and weather observations. The weather data was then merged with the aggregated demand data by matching hourly timestamps.

We engineered several categories of features for our model, including temporal features (hour of day, day of week, month, and a binary weekend indicator) to capture cyclic patterns in bike usage. To incorporate historical demand patterns, we created lag features at 1 hour (recent demand), 24 hours (same time yesterday), and 168 hours (same time last week), along with a 24-hour rolling average of demand to capture trends. These lag features required us to sort the data by station and time, and we removed rows where lag values were unavailable at the start of each station's time series. Beyond the raw weather variables, we created categorical features, including rain intensity (no rain, light rain, heavy rain) and temperature categories (cold, cool, warm, hot) to capture non-linear weather effects on demand, and a binary flag for rainy conditions (precipitation > 0.1mm) was also included. Station features included the mean latitude and longitude coordinates for each station, along with learned station embeddings in the neural network to capture station-specific usage patterns.

After feature engineering and removing rows with missing lag values, our final dataset contained 551,073 hourly observations across 1,925 unique stations. We split the data chronologically using 70% for training, 15% for validation, and 15% for testing to ensure the model could generalize to future time periods.

Key Methodology:

Before discussing the key methodology used in this project, it is important to address some of the models that didn't work well to motivate our final approach. To begin, we fit a linear model. We found that only a small proportion of variance was captured by this model, given that a lot of our features were not linear in nature. For example, if the temperature was too hot or too cold, we would presume an adverse effect on bike demand, but this non-linear relationship can't be captured by a simple additive coefficient. Additionally, we explored logistic regression for trip classification and KNN for bike type prediction, which provided valuable insights into usage patterns but were not suitable for the forecasting task.

Given these limitations, we focused our efforts on neural networks capable of capturing non-linear patterns and complex interactions between features. Here we present our weather-integrated neural network approach for predicting hourly station demand. We found in our regression analysis that the data suffered from heteroskedasticity so we transformed the predictor in order to stabilize the variance. The predictions were then converted back to normal units for interpretability about demand. Our neural network architecture consisted of 3 hidden layers all with ReLu activation functions. The first layer had 128 layers, the second layer had 64 layers, and the last layer had 32 layers. Each layer was fully connected. In order to train we used 30 epochs minimizing our loss function MSE. One key realization in our neural network creation was that we were working with time series data. In order to avoid leakage during training we opted to not use a random split, but split on time. Below are the results from our testing.

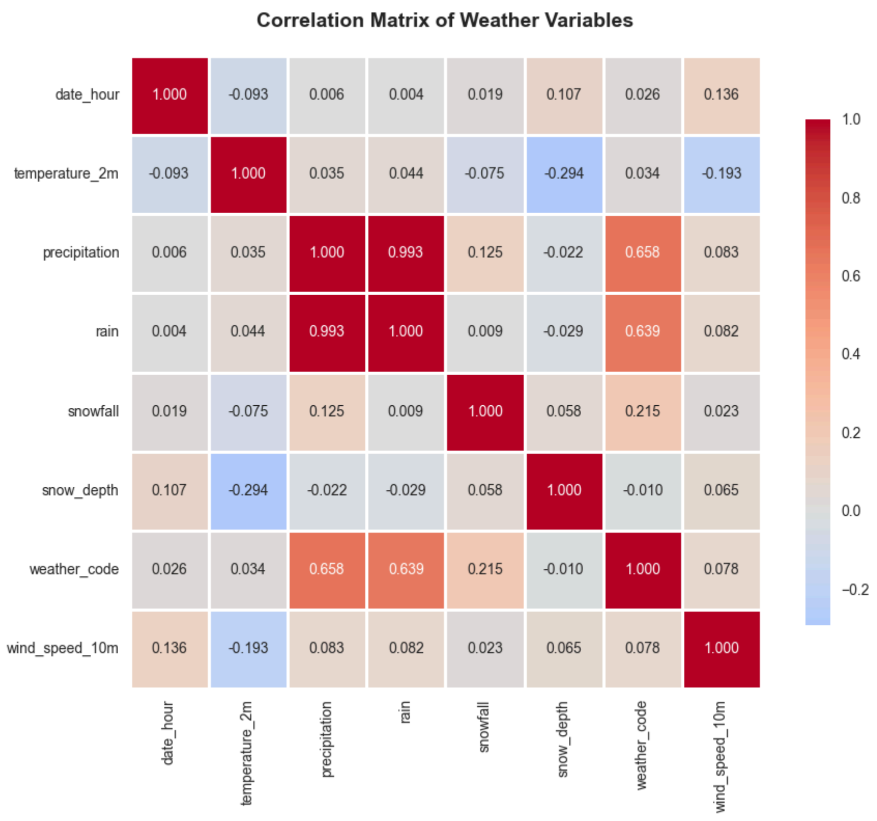
Results:

The results from our neural network showed the best performance of all of the models. Since we transformed the predictor we had to convert our units back into bikes. The log transformed dataset reached a validation MSE of 0.1839 after the last epoch. When we convert our units back in terms of bikes so we can interpret and compare with other algorithms. In order to get a more accurate accuracy measure we used rolling cross validation to ensure that we never test predictions on past data. We received a mean absolute error of 1.91. This means that our average prediction accuracy for predicting demand of NYC bike rental stations at a given hour is roughly 1.91 bikes. In addition to this we got a RMSE of 3.25. There are a few limitations to our model. To begin, the lag features create a dependence on recent demand patterns rather than weather. Next, for lower volume stations the data is more sparse and noisy, and thus predictions have lower accuracy. Finally, we discuss how to use the final code. The final file we used for the neural network is called final_nn.ipynb and can be run by running the cells in order. It gives the model creation, training, and evaluation with comments for each step.

Appendix

i.

We approached this problem from the perspective of understanding more about how our weather variables impact the ride demands. Our exploratory data analysis (EDA) process was designed to visualize and understand the weather factors that correlated with demand changes. The first thing that we looked at was a correlation heatmap.



	date_hour	temperature_2m	precipitation	rain	snowfall	snow_depth	weather_code	wind_speed_10m
date_hour	1.000	-0.093	0.006	0.004	0.019	0.107	0.026	0.136
temperature_2m	-0.093	1.000	0.035	0.044	-0.075	-0.294	0.034	-0.193
precipitation	0.006	0.035	1.000	0.993	0.125	-0.022	0.658	0.083
rain	0.004	0.044	0.993	1.000	0.009	-0.029	0.639	0.082
snowfall	0.019	-0.075	0.125	0.009	1.000	0.058	0.215	0.023
snow_depth	0.107	-0.294	-0.022	-0.029	0.058	1.000	-0.010	0.065
weather_code	0.026	0.034	0.658	0.639	0.215	-0.010	1.000	0.078
wind_speed_10m	0.136	-0.193	0.083	0.082	0.023	0.065	0.078	1.000

Figure 1: Correlation Heatmap

This heatmap helped us identify features in our weather data that were highly correlated, potentially causing issues in the future with interpreting model outputs. The main issues were that the rain column overlaps greatly with precipitation (remember this is New York, not Vermont or something!) Also, our weather code variable has high correlation with precipitation and isn't adding any sort of interpretative understanding. Therefore, we get rid of these two variables in our future work.

Next, in Figure 2, we look at the pickups in New York City by location, faceting based on whether it was in the morning, afternoon, or evening.

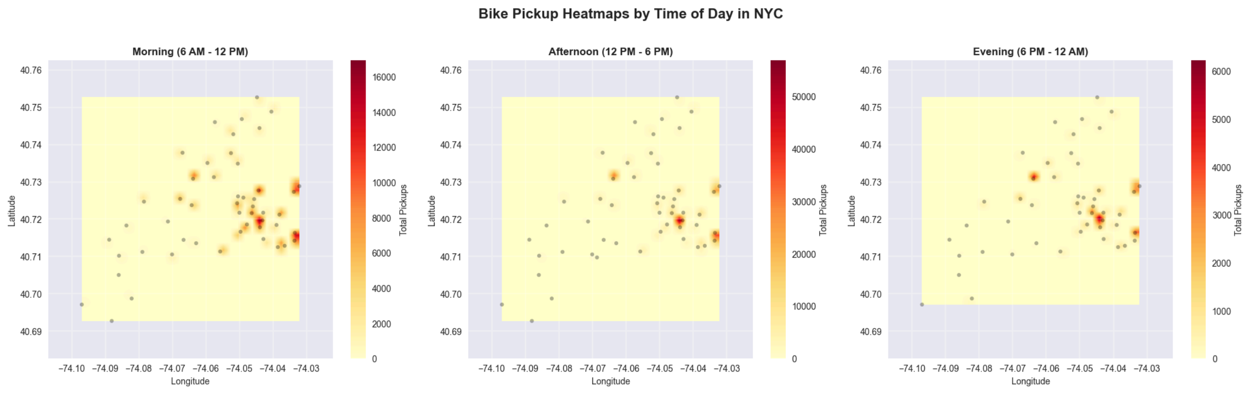


Figure 2: Pickups by Time of Day

Based on this figure, we can see that certain areas are hotspots, but they are mainly hotspots throughout the entire time period. Morning traffic seems a little heavier than afternoon and evening traffic.

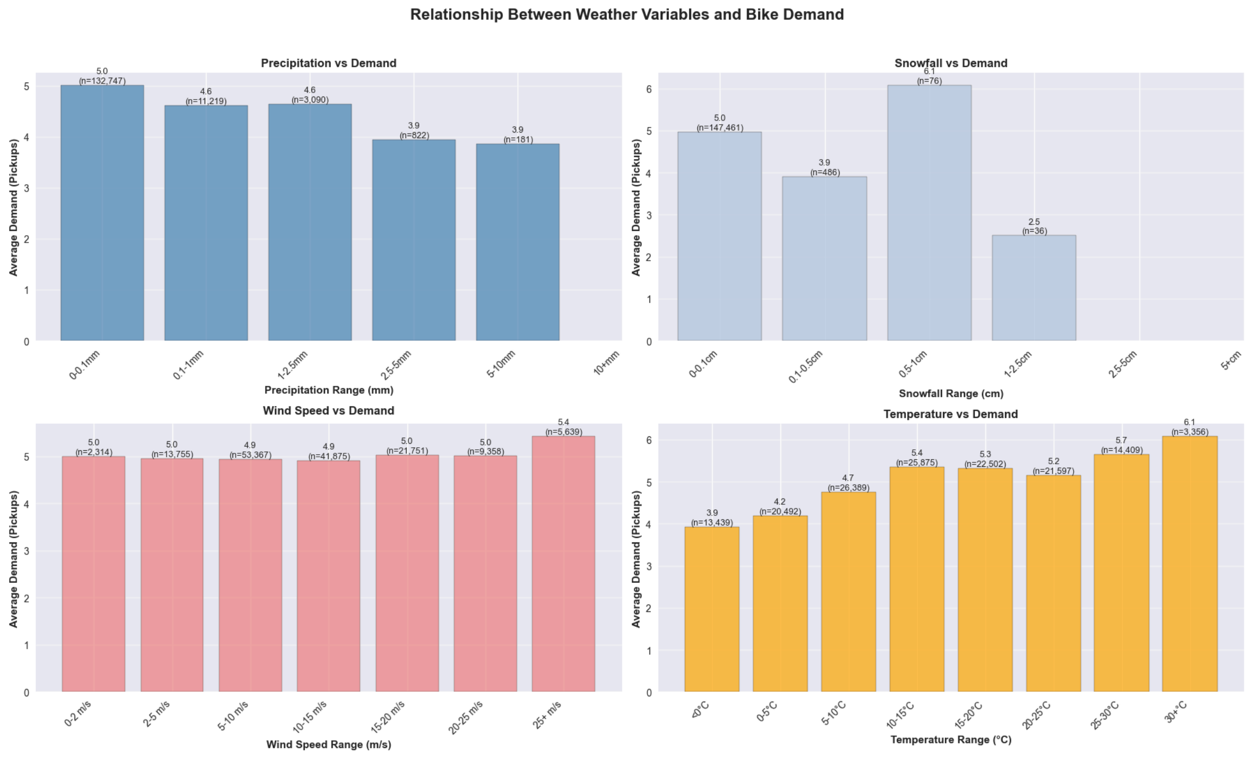


Figure 3: Weather vs. Bike Demand

In Figure 3, we explore how precipitation, snowfall, wind speed, and temperature impacts bike demand through binning. We can see that as precipitation increases, bike demand decreases. This follows from our intuition – people will be less interested in riding bikes if it's pouring rain out, opting for safer and drier methods of transportation. We see a similar impact with snowfall, Although there is a weird spike (albeit with a small sample size) in the 0.5-1cm bin. Further inspection would be required, but personally I'm suspicious it's just a small sample size issue (like potentially this snow level primarily occurred during rush hour, for example). We see that wind speed had no clear impact on bike demand. We also see that as the temperature got warmer, the demand got higher. This matches our intuition because people are more comfortable biking when the weather is nice out, they can wear short sleeves, and enjoy the sun.

Note that we cover the splitting of our data in the main part of the report.

ii.

Our data pre-processing was mainly working to get useful features from our date-time columns and ensure our data is in rows where each row is a specific ride containing weather data. Adding the external weather data really helped us find our project goals and perspectives. We did some aggregation and filling forward of some missing weather data to ensure that our data was as accurate as possible. We added some lagged variables, as well as some rolling means. We also standardized our numeric features. After completing all of this pre-processing, our data could easily be pushed into all of our various supervised learning techniques to see how best to predict demand.

iii.

To do some linear regression, we had to one-hot encode all our variables, and fit the model. To start, we did a train/val/test split, where we train on the train set, apply to the validation set, pick the model that does the best on the validation set, and then measure final accuracy on the test set. This allows us to directly compare our Linear Regression with Ridge Regression with Lasso Regression. From this analysis, the Lasso Regression had the highest (best) r-squared value, although it was extremely close. Therefore, we continued with the Lasso and did some diagnostic methods. From there, we could see quite clearly that a linear pattern was not a good fit for this data, judging by the increasing residuals as our fitted values got higher, and weird behavior in our QQ plot and Scale-Location Plot (Figure 4). Because of this, we stuck with more complex non-linear models in the future that assume less about the patterns in the data. However, we did do one Variable Importance Plot (Figure 5), which indicated that our lagged value and rolling average were by far the most impactful variables. This makes sense, because they take into account direct past behavior (which is frequently the best predictor of future behavior). We also see that the hour category is quite important to understanding the raw demand numbers. So these are features to watch out for as we continue with our other methods.

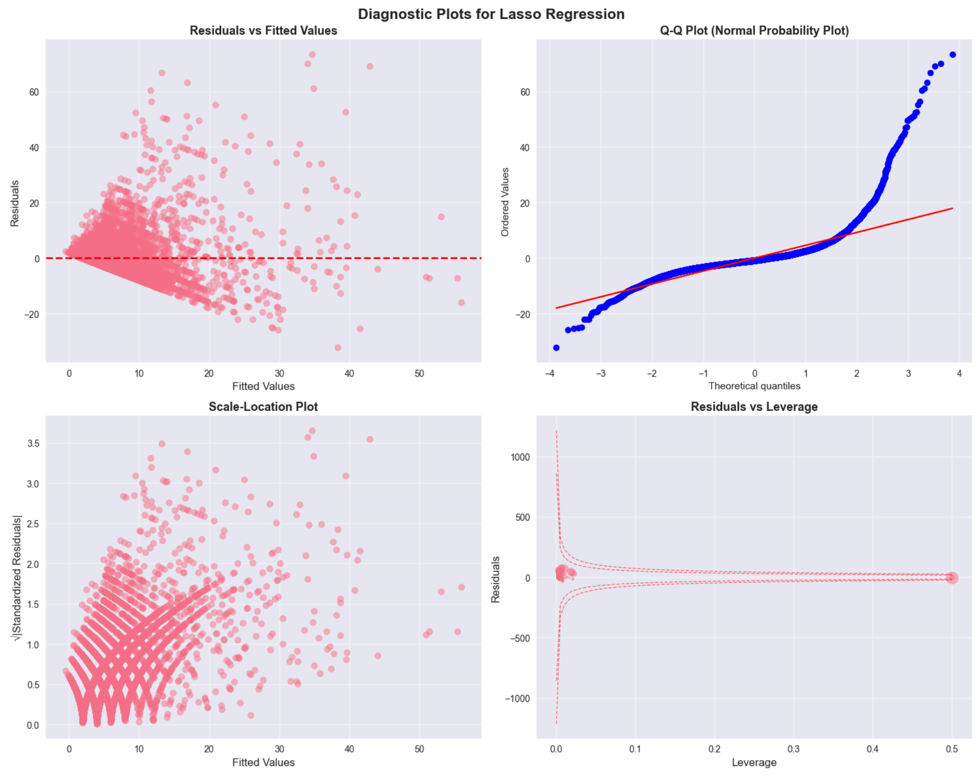


Figure 4: Diagnostic Linear Plot

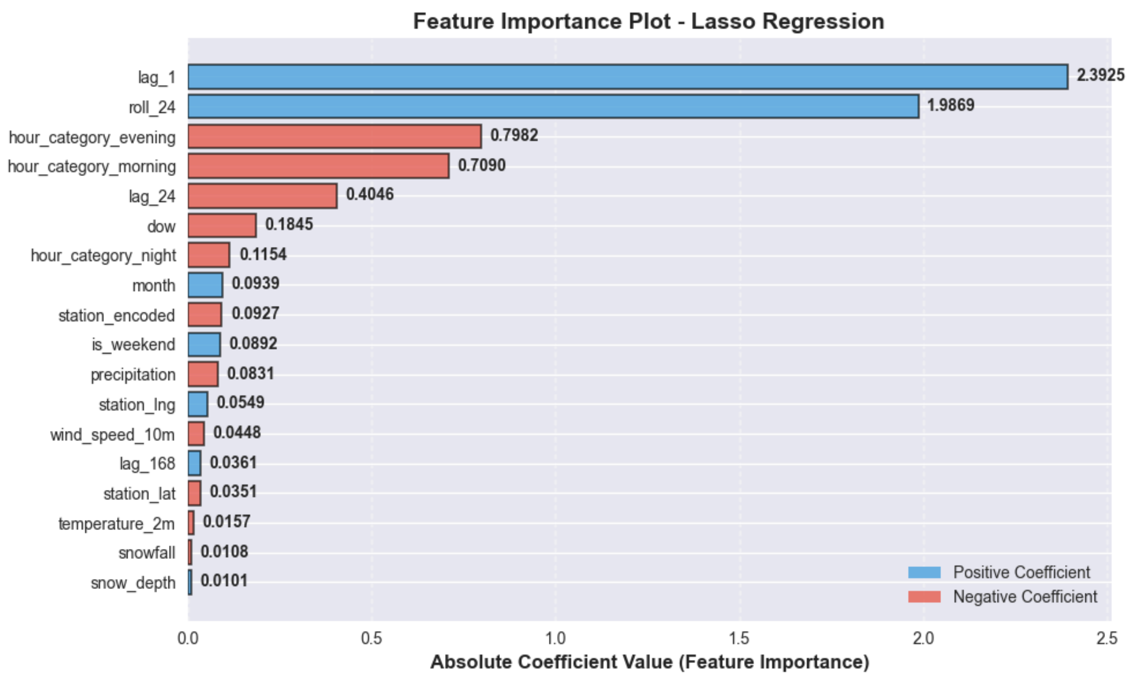


Figure 5: Variable Importance Plot

iv.

We applied logistic regression as an exploratory step to better understand trip patterns in the NYC Citi Bike data before building our station demand forecasting model. Specifically, we classified trips as either short or long based on trip duration to identify factors that influence riding behavior. The binary response variable "long_trip" was created by labeling trips with durations of 15 minutes or more as long trips and shorter ones as short trips. The features used for prediction included bike type, member type, and geographic coordinates, which were preprocessed using standardization for numeric features and one-hot encoding for categorical variables. We used a 70-30 train-validation split with stratification and trained the model using the LBFGS solver with balanced class weights.

The model achieved a training accuracy of 71.46% but had a near-zero true positive rate, struggling to identify long trips. The validation set ROC curve yielded an AUC of 0.5661, and we selected an optimal threshold of 0.2618 using Youden's J statistic, improving validation accuracy to 70.45%. Five-fold cross-validation produced a mean AUC of 0.556 +/- 0.0012 and a mean accuracy of 71.46%. The low AUC indicated that geographic coordinates and bike type had limited predictive power for trip duration. This insight motivated us to focus on temporal features and external factors like weather data when building our demand forecasting model. Regularization was not needed as the model didn't show severe overfitting.

v.

We used K-Nearest Neighbors (KNN) to explore usage patterns by classifying bike type, either classic or electric, based on trip characteristics, including duration, latitude distance, longitude distance, and total distance. This analysis helped us understand user behavior patterns before integrating weather data into our demand forecasting model. Features were standardized using StandardScaler before applying KNN.

We tested k values of 1, 10, 15, 25, 50, and 100. At k = 1, the model overfitted with 99.97% training accuracy but only 68.18% validation accuracy. Performance stabilized at k = 25, achieving 74.68% training accuracy and 71.99% validation accuracy, making it our optimal choice. The confusion matrix showed a high true positive rate of 89.08% for electric bikes but only a 41.14% true negative rate for classic bikes, with an F1 score of 0.8312. The ROC curve yielded an AUC of 0.7512 on the validation set. Five-fold cross-validation confirmed consistent performance with a mean AUC of 0.7406 +/- 0.0010 and a mean accuracy of 71.49% +/- 0.0006.

This analysis revealed that electric and classic bikes have distinct usage patterns in terms of distance and duration. These insights informed our feature engineering decisions when integrating weather data, suggesting that temporal aggregations of trip characteristics at the station level could be valuable predictors of demand. We did not implement decision trees or random forests for this classification task.

vi.

We applied clustering techniques to explore behavioral differences across Citi Bike start stations as part of our exploratory analysis. Trip data were aggregated at the station level, and features were constructed to capture overall usage intensity and temporal riding patterns, including total trip counts, average trip duration, and the proportion of rides occurring on weekends, during morning and evening commute hours, and late-night periods. All features were standardized prior to clustering to ensure comparability across dimensions.

We used K-means clustering, selecting the number of clusters via the elbow method based on within-cluster sum of squares (WCSS). Inspection of the elbow plot indicated that marginal reductions in WCSS diminished beyond six clusters, leading us to choose k = 6 as a reasonable balance between model complexity and interpretability.

Our final clustering had an inertia of about 6420 and a silhouette score of 0.17. While the silhouette score is relatively low, this is expected given the scale and heterogeneity of the data, as well as substantial overlap in station usage patterns across time and rider types. Cluster assignments were nevertheless highly consistent across random initializations, with a Rand index of 0.999 when comparing the model using different random states, indicating strong stability in the resulting station groupings.

Principal component analysis (PCA) was applied for visualization purposes, projecting the standardized feature space into two dimensions to facilitate interpretation of the clustering results. We then created a PCA scatter plot and cluster-level radar plots to reveal broad behavioral distinctions among stations, including commuter-focused, high-volume, and late-night–oriented usage profiles. Although PCA and clustering provided useful exploratory insights, they were not incorporated into the final forecasting model, since we were focusing on supervised prediction of station demand.

vii.

We discuss the use of Neural Networks in our main document.

viii.

We used multiple different hyperparameter tuning methods throughout our project. In our logistic regression analysis, we used cross-validation to evaluate model performance at different probability thresholds, ultimately selecting the primary threshold of 0.2619 using Youden's J statistic to maximize the balance between true positive and true negative rates. When using KNN, we compared multiple models with different k values to see which model produced the smallest error. We tested k values of 1, 10, 15, 25, 50, and 100. We found that the accuracy began to stabilize around k = 25, which was a good balance between underfitting and overfitting. For tuning parameters in clustering, we used a scree plot to find the optimal number of clusters for k-means clustering. The optimal number ended up being 6, since the decrease in WCSS began to level off after that many clusters. Overall, throughout the project, we relied on validation set performance and cross-validation metrics to guide our hyperparameter selection decisions, ensuring our models generalized well to unseen data.