Optimization of Citi Bike Quantity and Placement Across New York City Areas

Ria Bendiganavale, Ali Kadiri, Veda Parulekar

GitHub Repo Link | Video Link

Project Definition

Problem Statement

We have noticed that large-scale bike systems, specifically Citi Bike, struggle with bike availability in certain locations during specific times and seasons, which is an inconvenience for users. To the company, there are concerns such as manually rebalancing bikes across stations frequently through the use of trucks, which is expensive and less efficient. Overall, this may cause a profit loss to the company as they cannot take advantage of the high demand in certain locations as well as a bad reputation to its users for the unreliable service. Our project aims to develop a system to track how many Citi Bikes are being used across all stations in New York City to optimize how many bikes are needed for efficiency and the sustainability of the business.

Strategic Aspects

Our main strategy is to make use of Citi Bike usage data to identify patterns in the demand for bikes, in both location and time, and utilize these patterns to recommend specific quantities of bikes for each dock. By using machine learning techniques, statistical modeling, and time-series analysis, we will be able to develop a data-driven recommendation system estimating the optimal number of bikes to be allocated per dock, based on where that dock is, and what time of the year it is. Analyzing the usage data provides us the opportunity to view the data in a new perspective and be able to reduce potential problems in the future. With these strategies, our goal is to not only improve city transportation but also increase user satisfaction with this service.

Novelty and Importance

Significance

Transportation continues to be an important source in urban areas. Many people use Citi bikes since they are cheap and convenient. If people cannot find bikes in their area, they can experience issues like longer commute times. Our project addresses these concerns and aims to find a solution to them. We understand the importance of transportation in New York City and want to use data management in order to combat these issues. This project serves as an opportunity for us to examine this issue in a different perspective and propose an effective plan. Through proper placement of Citi bikes in New York City throughout the year, they can continue to be a reliable source without causing problems in the long run.

Existing Issues

There are several issues when it comes to utilizing real life data. One of them includes the possibility of the bike data not updating properly. If key features such as bike availability and usage history are interrupted, it can create problems such as inadequate bikes, longer commute times, and misuse of bikes. By incorporating GPS systems, they can allow us to track bikes in real time and avoid significant discrepancies as much as possible. In addition, another issue is inaccurate or missing information from the database. If there are situations where features including ride duration, bike id, and station information are compromised, it can lead to an inaccurate representation of the Citi bike system. By monitoring and cleaning the data received, we can make sure the data stays consistent and accurate for the users.

Summary of Related Works

Our project addresses data gaps by coupling short-term, station-level ML forecasts (ARIMA/Prophet combined with tree-based models using rolling windows and real-time weather nowcasts) with a rebalancing routine, resolving the system every few hours that routes adapt dynamically to demand and maintain high service levels throughout the day. Our model is more focused on the long-term, incorporating factors such as the weather and time of day, and offers a solution to the issue of bike shortages and the misallocation of bike quantities to various docks through predicting bike usage and recommending an optimal number of bikes per docks.

Progress and Contribution

Data Utilization

Our data we used originates from the Citi Bike System Data website. Specifically, we utilized the Trip History Data obtained from the year 2023. Each month provides multiple datasets consisting of all of the bikes in New York City. Each dataset is organized into a table with the following columns: ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id, end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, and member_casual. We will use this data and store it in a centralized database where we can analyze the patterns of this data. We do this in order to make predictions based on this data for what locations are in high demand, with several dependencies such as weather and time.

Cleaning Data

Since the data we used was already quite clean, as it was downloaded directly from the Citi Bike website, the only cleaning required was removing rows with N/A's in them. We decided against removing outliers using the IQR method, as there is a heavy skew and that heavy skew would mean even by multiplying IQR with 1.5 and getting rid of those beyond it, we would be getting rid of some meaningful data, and thus would make it harder to work with comprehensive data.

```
CB2023 = pd.read_csv(r"C:\Users\Peasa\CitiBike2023_Combined_Sampled.csv", low_memory=False)
CB2023_clean = CB2023.replace(r'^\s*$', np.nan, regex=True).dropna()
print("Clean Shape:", CB2023_clean.shape)
```

Models/Techniques/Algorithms

For our project, we solely used the prophet algorithm, which is based on a procedure of forecasting time series data. This algorithm is effective for projects that may not have strong or enough data such as missing values, trend changes, and irregular events. For our project, we knew that this algorithm was the best option for us to use. Citi Bike needs to accurately predict when and where bikes should be placed, accounting for seasonal trends, location-specific demand, holidays, and even irregular or missing data. Prophet excels in these areas because it automatically detects yearly, weekly, and daily seasonality, making it well-suited to handle patterns like increased summer ridership or weekday commuting spikes. The algorithm allows for separate forecasting by location, making it scalable for systems with hundreds of stations, each with its own trends. Additionally, its interpretable output breaks forecasts down into components like trend, seasonality, and holidays, giving planners insight into why demand is expected to change. For our Prophet model, we first needed to set a cutoff point, as if there is too little data, Prophet may not work. For this, we wanted to find the bottom 5th percentile to use as the cutoff point. The following code helped us find this:

```
#First, we find a good cutoff point for the prophet model:
hourly_demand = CB2023_clean.groupby(["start_station_name", "hour"]).size().reset_index(name="rides")
station_demand_summary = hourly_demand.groupby("start_station_name")["rides"].sum().sort_values()

#Create a Histogram
station_demand_summary.plot(kind="hist", bins=50, figsize=(10,6), title="Total Hourly Rides per Station")
plt.xlabel("Total Hourly Rides")
plt.ylabel("Number of Stations")
plt.grid(True)
plt.show()

#Describe the stats
print(station_demand_summary.describe())
print("5th Percentile of Total Hourly Rides per Station:", station_demand_summary.quantile(0.05))
```

Which gave us the cutoff point, among several other valuable outputs:

```
5th Percentile of Total Hourly Rides per Station: 4.0
```

Afterwards, we decided to go forth with building the actual prediction model for every single station:

```
#Build hourly demand from  original data
hourly_demand = CB2023_clean.copy()
hourly_demand['ds'] = hourly_demand['started_at'].dt.floor('h')
hourly_demand = (
    hourly_demand
    .groupby(['start_station_name', 'ds'])
    .size()
    .reset_index(name='y')
)

#Set forecast horizon for a week = 7 days = 168 hours
HORIZON = 168

#Set cutoff point
dcutoff = 4

#Fit one Prophet model per station
forecasts = []
for station, station_data in hourly_demand.groupby('start_station_name'):
    if len(station_data) < dcutoff:
        continue
    m = Prophet()
    m.fit(station_data[['ds', 'y']])
    future = m.make_future_dataframe(periods=HORIZON, freq='h')
    fc = m.predict(future)[['ds', 'yhat']].rename(columns={'yhat': 'forecast'})
    fc['station'] = station
    forecasts.append(fc)

#Combine all forecasts
all_forecasts = pd.concat(forecasts, ignore_index=True)

#Export it into a .csv file
all_forecasts.to_csv('citibike2023_hourly_forecasts.csv', index=False)
print(all_forecasts.tail())
```

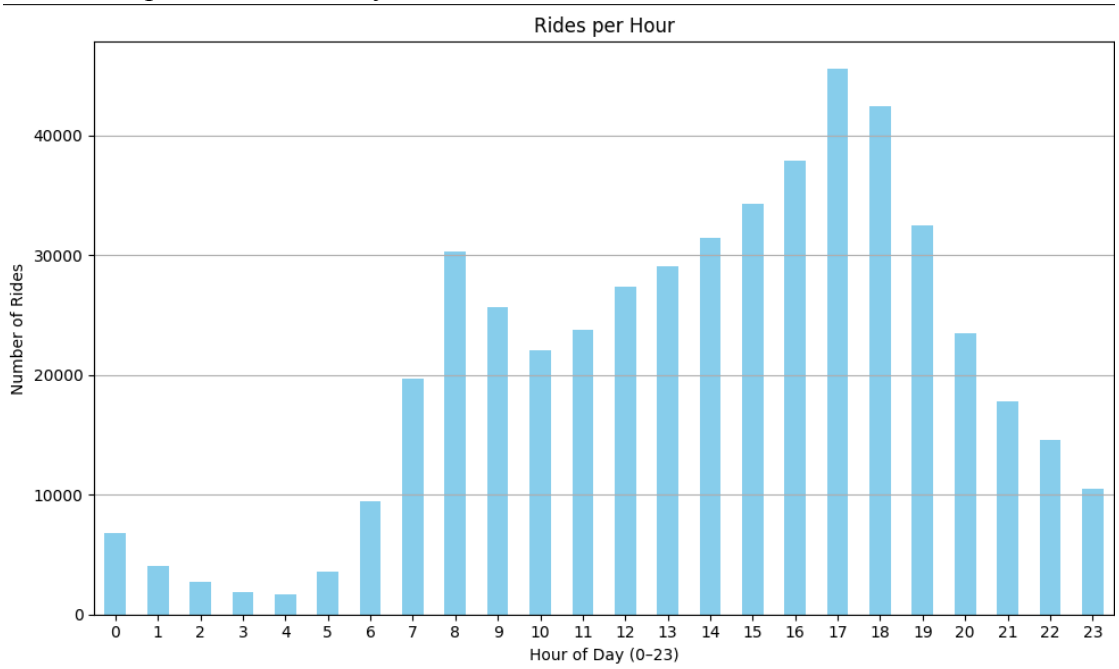
Experimental Design

We created the following graphs and plot in order to display the Citi bike dataset from 2023: Rides Per Hour, Rides Per Day of Week, Rides Per Month, Top 10 Start Stations, Top 10 End

Stations, Trip Duration, and Total Hourly Rides Per Station. In addition to these visualizations, we used Python to create codes that can execute our system in different components. We used the dataset to run our first experiment that returns the stations with the highest net inflow and outflow. The second experiment we conducted was a trip duration summary that includes the following components: count, mean, standard deviation, minimum, 25%, 50%, 75%, and maximum. The third experiment we made was taking a popular station and grouping uses by time in order to estimate their demand over time. We then made a forecasting model that can estimate the demand for all of the stations in the dataset. From here, we were able to establish the hourly demand, set the cutoff point, and applied a Prophet model for each station. Our project incorporates these experiments in order to show the forecast system we created based on the Citi bike dataset.

Key Findings & Results

After taking a representative sample size of 12,500 from every csv file due to large amounts of data that would take a large amount of time to analyze, we found a few key findings throughout this project. First off, we found out that people rely on Citi bikes to use from work rather than to work as shown in the bar graph created by Ali in the video. More people use Citi bikes throughout the day until around 5pm rather than at 8am shown through the gradual rise after the morning hours. Another finding is that people tend to use Citi bikes more during the warmer months of May to October, gradually rising in March, and tapering in November. We believe that this result was not surprising due to the warmer weather, but it is interesting to see the clear consistency during these warmer months compared to the colder ones. In addition, we found that the net outflow from starting to ending station is at Broadway and W 56st at 258. This supports that bikes should be placed at this location more than any other station. The bar graph for the number of rides per hour of the day is as seen here:



Advantages & Limitations

The approach we adopted when working on this project presented some advantages and limitations. Our approach helped us obtain a large dataset where we were able to produce graphs

and analyze the trends with Citi bikes occurring in a year. However, having a large dataset with extra attributes made it difficult to initially display our data. After organizing our data and utilizing graphical software, we overcame this hardship and produced visualizations that support our objectives with the project. Another advantage from our approach is the ability to predict the most effective placement of the bikes from user demand. Examining patterns with start and end stations allows us to create a model that shows demand in specific areas. However, a limitation that could occur is a delay in updating demand. Situations where the demand of Citi bikes in a specific area may change over time can potentially impact users' access to them. A potential improvement to our approach would be to incorporate the most recent data provided by the Citi Bike Usage Data and compare years in order to identify key differences with demand. From there, we can use our prediction model to change demand in favor of our users.

Changes After Proposal

Differences Between Proposal and Report

In our project proposal, we had said that we would formulate a simple capacitated vehicle routing problem to suggest truck routes that minimize unmet demand and travel distance. In the actual project, we did not do this. We had also stated that we would use tree-based models with rolling windows and real-time weather nowcasts, as well as that we would make predictions using weather as a dependent variable. We also did not compare models, nor did we use MAE (Mean Absolute Error) to evaluate our accuracy. Furthermore, we did not hold-out a test period,, and we did not have any back-test rebalancing decisions measuring service level improvements.

Purpose of Changes

The purpose of the majority of the changes were simply to accommodate for a lack of data, or computational power, as we were working with a very large dataset. We did not formulate a routing problem suggesting truck routes, as we did not have access to truck route data, and ultimately decided that it was beyond the scope of our project, whose main goal is to forecast bike usage per station. We also did not use tree-based models with rolling windows, nor did we use real-time weather nowcasts, because we did not have access to that data, and integrating it would have made the project much more complex, creating external API dependencies and including time-synchronized merging, which would not be feasible during our project timeline. We also did not compare models, as we believed that it was unnecessary, and that the Prophet model was adequate. We also did not use MAE, as we thought that our RMSE evaluation was sufficient.. Finally, we did not have a hold-out test period, or any back-test rebalancing decisions because our main focus was building and validating forecasting models, and simulations would have required more assumptions about operational constraints and access to data on actual rebalancing actions, which we do not have access to.

Prominent Bottlenecks

The biggest bottleneck we faced was the massive amount of data we were working with. The fact that we were working with over 35 million records of trips meant that storing it into the SQL Database took a very long time, and was very demanding on our computers. Furthermore, the large amount of data meant processing it in Visual Studio Code was incredibly time-consuming, and so we had to find creative solutions to make a representative sample of the data. Ultimately, we ended up taking the approach of taking small samples from each individual CSV file, but

finding this approach took a while and I think the challenge really taught us to think out of the box to find creative coding solutions when we hit a wall. Another big bottleneck we faced was the lack of easy access to bike inventory levels, station capacity limits, or truck route data, which made it impossible to implement the routing optimization that we had initially intended to. This led to us ultimately not doing it at all. A lack of data was a consistent struggle for us throughout this project, as it also prevented us from very easily using weather nowcasts and other real-time data, also making us have to give up on that part of our proposal. Another key bottleneck we ran into was the challenges we faced when trying to deploy our Prophet demand forecasting model as a functional application with Flask. While eventually, we were able to build it, we ran into lots of issues pickling large sets of Prophet models, solving time zone inconsistencies, and structuring user inputs to return meaningful forecasts. This took significant time, as we had to learn new skills in flask, HTML, and also use pickle for the first time

Conclusion

Summary of Contributions

Our group was successful in collaborating on a project that we were all interested in. We believed that focusing on Citi bikes was important because of how reliable they are in the transportation industry. We contributed equally to the project and completed our tasks in a timely manner. In addition, we communicated effectively and helped each other with the project in the event that we were unsure of what to do. In the proposal, we each worked on multiple sections and created a proper plan for our project. In the final report, we expanded on our proposal and created a concise report that displays our project objectives. We created the SQL database along with the Python codes for our prediction model. In addition, we made graphs highlighting patterns based on the dataset. Finally, Ali made a video explaining our project with the codes and their outputs. Overall, we had fun developing this project and are proud of what we have accomplished.

Future Directions

If we worked on this project in the future, what we would do is examine other cities and the impact Citi bikes have there. If there are some areas that could potentially rely on this form of transportation, we can utilize our software to predict which areas would be in high or low demand. It would be difficult to use our model on cities that do not have Citi bikes since we cannot use previous data. However, if those cities have similar bike patterns with New York City, we could apply them to future cities and equally distribute the bikes. Even though this plan may be long and difficult, it could help establish the future of Citi bikes in new areas.