

Bikenomic - NYC Citibike Optimization

Charan Kandasamy Raja
ck3740@nyu.edu

Aravindsrinivas Krishnamoorthy
ak11115@nyu.edu

Abstract—This project investigates the use of live station status and historical data from Citibike, a key commuting resource in New York City and parts of New Jersey, to enhance ridership and reduce bike idle time. The study focuses on identifying high utilization Citibike stations based on patterns of bike pickups and drop-offs across various times and days. By analyzing historical trends, we aim to devise strategies that encourage riders to utilize underused stations, thereby balancing demand across the network. Proposed interventions include incentivizing users to walk further to less busy stations, allowing those at congested stations to access bikes more readily. This approach seeks to optimize the distribution of bikes and enhance the efficiency of the Citibike system.

I. INTRODUCTION

Citibikes are an integral part of New York City and they help people commute throughout the boroughs of the city and parts of New Jersey. The project aims to use the Citibike's live station status and historical data to identify a model which can help the company increase ridership and drop idle time on these bikes.

As a part of this project, we are attempting to use historical data to identify which Citibike stations have traditionally been high utilization stations either because more people pick up their bikes from those stations or because the station sees an increased drop off rate with more people ending their journey in these stations through different days of the week and different times of the day

Using this data we aim to identify a method to incentivize users to take a slightly longer walk to a farther station which may not see as high ridership that time of the day allowing users who are willing to pay more to take bikes from high priority stations.

II. OBJECTIVES

We have the following objectives that we aim to achieve from this project ¹¹

- Identify trends and patterns we have been seeing in Citibike Ridership data over the years.
- Building a Model which when provided a station id, time of day and day of the week can classify the station as a high frequency station, normal frequency station and low frequency station
- When a user requests a bike from a location, we aim to provide the user with multiple options - High Frequency

Stations with surge pricing, Normal Frequency Stations, Low Frequency Stations with Discounts

III. DATA COLLECTION

For our data, we made use of the following datasets:

- Citibike Historical Trip Data - We took data from the year 2023 to understand patterns and build our model. Dataset: <https://s3.amazonaws.com/tripdata/index.html>
- Citibike Station information Data - This data is static data which contains the information of all 2200 Citibike stations which are currently active in NYC and surrounding areas. Dataset: https://gbfs.citibikenyc.com/gbfs/en/station_information.json

```
{
  "capacity": 61,
  "lon": -74.0131813287735,
  "region_id": "71",
  "name": "Whitehall St & Bridge St",
  "lat": 40.7036617052413,
  "external_id": "c1a4d909-0a00-475a-8e82-18ed13a4eb01",
  "rental_uris": {
    "android": "https://bkn.lft.to/lastmile_qr_scan",
    "ios": "https://bkn.lft.to/lastmile_qr_scan"
  },
  "eightd_has_key_dispenser": false,
  "eightd_station_services": [],
  "short_name": "4962.02",
  "has_kiosk": true,
  "rental_methods": [
    "KEY",
    "CREDITCARD"
  ],
  "electric_bike_surcharge_waiver": false,
  "station_type": "classic",
  "station_id": "c1a4d909-0a00-475a-8e82-18ed13a4eb01"
},
```

Data Sample for Station Information.

- Citibike Station Status Data - This data contains the live status of any citibike station which we update every time a new request comes (for the sake of this MVP), but would ideally be updated every 5-10 seconds in order to account for any new rentals that have been made. Dataset: https://gbfs.citibikenyc.com/gbfs/en/station_status.json

```
{
  "is_renting": 1,
  "is_returning": 1,
  "legacy_id": "1843486667668726558",
  "num_docks_available": 1,
  "eightd_has_available_keys": false,
  "num_ebikes_available": 11,
  "num_bikes_available": 23,
  "station_id": "1843486667668726558",
  "num_scooters_unavailable": 0,
  "num_docks_disabled": 0,
  "num_scooters_available": 0,
  "is_installed": 1,
  "num_bikes_disabled": 1,
  "last_reported": 1715013706
},
```

Data Sample for Station Status.

IV. WHY IS THIS A BIG DATA PROBLEM?

The dataset from the year 2023 is notably large, exceeding 4 GB when fully extracted. Such a size can be challenging for many conventional computers to process efficiently, if at all. This suggests that leveraging a horizontally scaled machine learning model would be more appropriate for handling this volume of data. Spark ML, renowned for its ability to distribute data processing across multiple nodes, stands out as a particularly suitable tool for this task.

Moreover, the CitiBike system itself is a significant source of data, capturing detailed records of each bike trip. These records include various attributes such as the start and end times, the starting and ending points, the duration of each trip, and the user type, categorized as either a subscriber or a casual customer. Given the extensive network of thousands of bikes and stations spread across New York City, the volume of data generated is immense and accumulates swiftly, epitomizing the challenges associated with big data.

This data is not static but generated dynamically in real-time as each bike trip occurs. Such a continuous flow of data necessitates the use of advanced technologies capable of rapid data processing and analysis. Technologies that can handle these demands are crucial for deriving timely insights, such as understanding current usage patterns or assessing station crowding. These insights are vital for making informed operational decisions, such as the strategic redistribution of bikes to meet varying demand levels across the city. This scenario underscores the importance of using robust, scalable technologies like Spark ML to manage and exploit the full potential of large-scale, real-time data sets effectively.

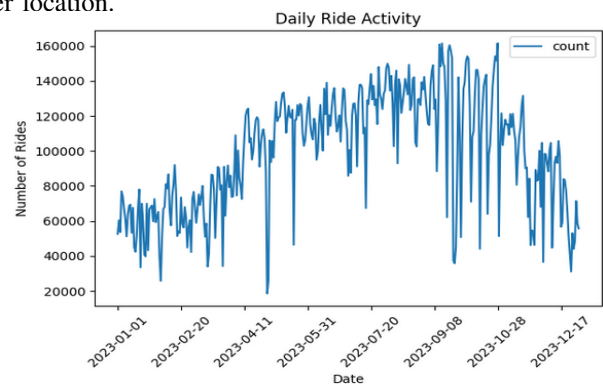
V. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is a fundamental step in the data analysis process as it allows us to make sense of the data we have collected, highlighting patterns, anomalies, key variables, and the relationships between them. When applied to Citibike data, EDA not only aids in understanding the dynamics of bike usage but also supports better decision-making for service improvements and strategic planning.



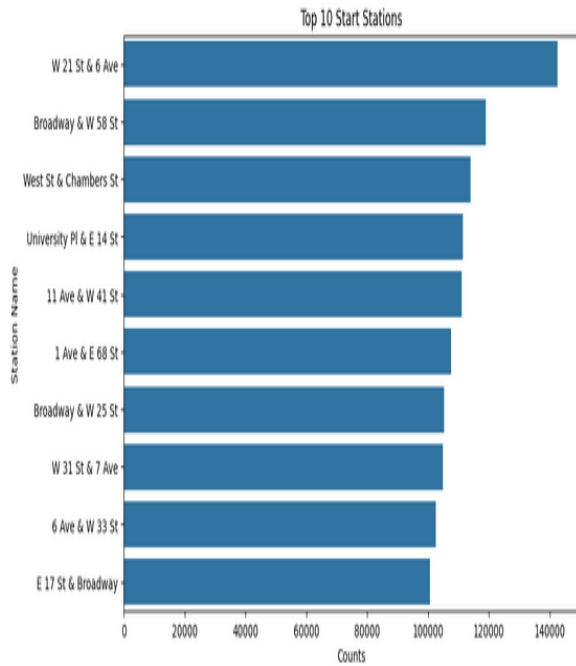
Data Sample for Station Status.

We can see that stations are spread out almost uniformly covering the entire New York City. This would give us plenty of options to make sensible alternate station locations for any user location.



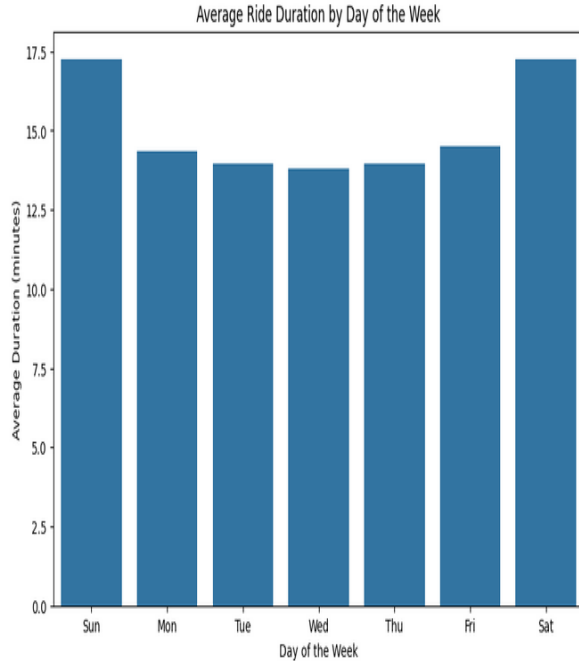
Data Sample for Station Status.

We can observe that the ridership data has a clear seasonal trend to it. More rides are taken in the warmer months of May, June, July, August, and September. The ridership count also varies a lot in the transition seasons Fall and Spring when cold and warm days occur equally.



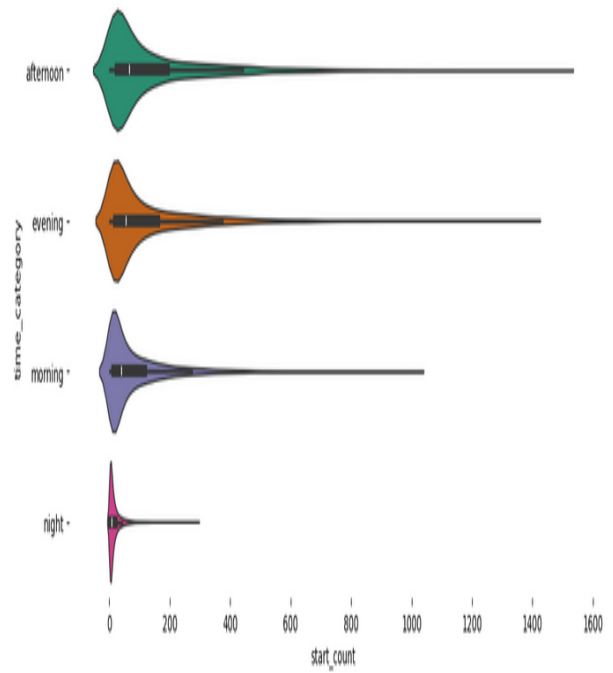
Data Sample for Station Status.

We can see that the most popular stations are in Midtown Manhattan and around Broadway. The station at "W 21 St & 6 Ave" has the highest usage, significantly more than the others, indicating it is a major hub for Citibike users.



Data Sample for Station Status.

All year round the average duration of rides increases significantly around the weekend when people presumably have more leisure time or are not in a rush.



Data Sample for Station Status.

We can observe that a lot more rides are started in the afternoon and evening times vs night and early morning. This can help us decide when our service will see the most traffic and help us manage computation models efficiently.

VI. MODEL

We evaluate the best station for starting and ending the ride based on a metric we created called 'bike_availability_diff'. It is the estimated difference between the number of bikes starting and ending at that station within a specific timeframe. This metric directly reflects the net change in bike availability at a station. A positive value indicates more bikes left than arrived, suggesting an outflow surplus; conversely, a negative value points to an inflow surplus, indicating more bikes arrived than departed. Understanding these trends helps identify stations that consistently face shortages or surpluses, which can prioritize redistributive interventions.

We built a model on Pyspark ML using Random forest regression to predict the value of 'bike_availability_diff' for any given station based on the day of the week it is and the time of request. This model takes in 36 million data points and the 'bike_availability_diff' output for each station is computed and stored in a CSV file to be accessed by the frontend.

VII. APPLICATION

In the development of our application[1], we employed several technologies tailored to meet the specific needs of our project. Flask, a lightweight and versatile Python web framework, was chosen to serve as the backbone of our webserver. This framework is particularly well-suited for small to medium web applications, providing the necessary flexibility and ease of use for rapid development.

For the user interface, we integrated HTML and JavaScript to construct the front end. HTML was used to structure the


web pages, while JavaScript enhanced interactivity, allowing users to interact seamlessly with the web elements. This combination ensured a responsive and user-friendly interface, essential for engaging users effectively.

The backend database functionality was handled by MongoDB, a NoSQL database known for its high performance and flexibility. Each station's status and detailed information were stored as separate documents within a MongoDB collection. This document-oriented database was selected because it natively supports JSON-like data structures, making it highly efficient for storing and querying the structured yet flexible data retrieved from our source JSON files.

Additionally, we utilized the geopy library's Nominatim service to convert textual location data into precise geographical coordinates. This conversion was crucial for functionalities such as identifying the nearest stations to a user's location. By translating addresses or location names into coordinates, we could pinpoint the exact locations required for further processing.

Furthermore, to visually represent the data on maps, we incorporated OpenMaps, an open-source mapping service. OpenMaps was instrumental in displaying the coordinates of the nearest start and stop stations, providing users with intuitive, graphical representations of their biking options. This visual component not only enhanced the user experience but also aided in the practical planning of routes and understanding station layouts within the city.

Together, these technologies formed a cohesive and robust framework, supporting the complex functionalities required to manage and present large-scale data in an accessible and efficient manner.



Bikenomic - Citibike Location Finder

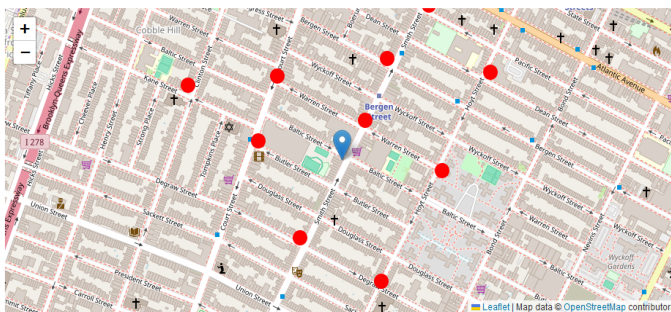
Start Location:

End Location:

[Convert](#)

Data Sample for Station Status.

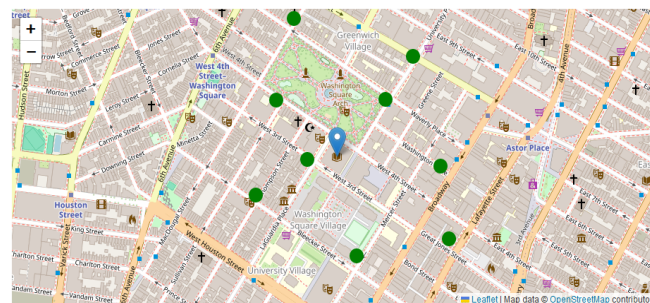
Starting Location



Kane St & Clinton St: 3 bikes available
Degraw St & Smith St: 13 bikes available
Butler St & Court St: 22 bikes available
Pacific St & Smith St: 46 bikes available
Degraw St & Hoyt St: 24 bikes available
Bergen St & Smith St: 47 bikes available
Hoyt St & Warren St: 35 bikes available
Warren St & Court St: 27 bikes available
Dean St & Hoyt St: 65 bikes available
Warren St & Smith St: 33 bikes available

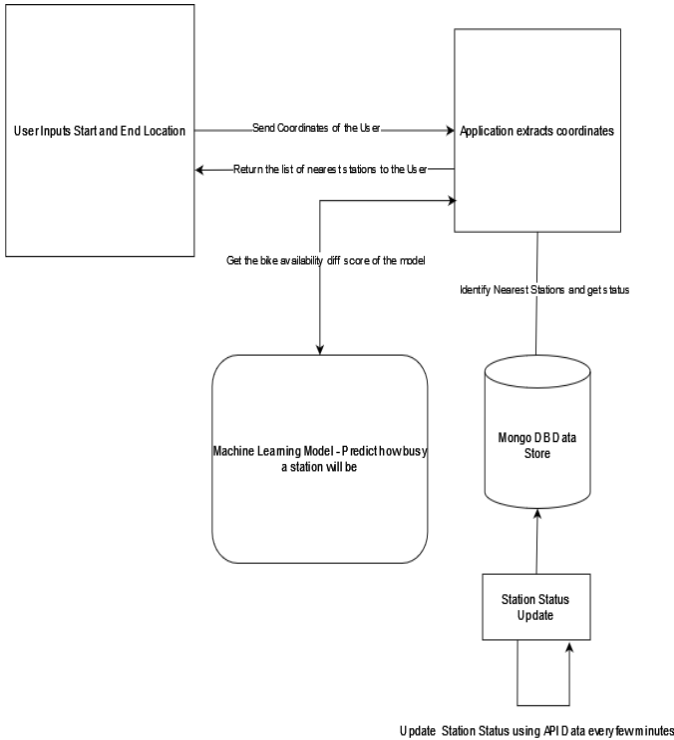
Data Sample for Station Status.

End Location



Washington Pl & Broadway: 6 docks available
Mercer St & Bleecker St: 5 docks available
LaGuardia Pl & W 3 St: 11 docks available
Great Jones St: 6 docks available
Sullivan St & Washington Sq: 14 docks available
University Pl & E 8 St: 37 docks available
MacDougal St & Washington Sq: 4 docks available
Thompson St & Bleecker St: 12 docks available
E 9 St & 5 Ave: 12 docks available

Data Sample for Station Status.



Data Sample for Station Status.

VIII. CONCLUSION

The "Bikenomic" project has significantly contributed to our understanding of urban bike-sharing systems, specifically targeting improvements in Citibike's operational efficiency and customer satisfaction in New York City. By leveraging historical and live station data, our project identified high-utilization stations and proposed targeted interventions to balance bike distribution across the network. The implementation of a machine learning model, developed using Pyspark ML, allowed for the prediction of station usage patterns, thereby facilitating smarter, data-driven decisions in bike deployment.

Our findings demonstrate that strategic redirection of users to less congested stations, coupled with dynamic pricing models, can effectively reduce idle bike times and enhance overall system responsiveness. This approach not only optimizes the use of existing resources but also improves user experience by reducing wait times and increasing the availability of bikes when and where they are needed most.

Furthermore, the technology stack comprising Flask, MongoDB, and OpenMaps, supported by robust backend and frontend frameworks, ensured a seamless integration of complex functionalities into a user-friendly application. This integration highlighted the practical applicability of our project, offering a blueprint for similar urban mobility solutions.

IX. FUTURE WORK

To enhance the capabilities and reach of our current project, several advancements can be proposed. Firstly, expanding the diversity and sophistication of the machine learning models tested can significantly improve the ranking function used in

our application. By experimenting with a broader array of algorithms, such as deep learning models or advanced ensemble techniques, we could potentially increase the accuracy and efficiency of the system. This would enable a more refined analysis of station utilization patterns, thus optimizing bike availability and user satisfaction.

Secondly, integrating live trip data represents a substantial opportunity to elevate the project's real-time analytics capabilities. Currently, our access to real-time data is limited; however, implementing a Kafka Pipeline could bridge this gap. Kafka, a robust streaming platform, would allow for the continuous ingestion of live data streams, facilitating real-time processing and immediate insights into user behaviors and station dynamics. This addition would enable the application to adjust more dynamically to fluctuating demands and operational challenges, enhancing overall system responsiveness.

Lastly, scaling the project to include bike-sharing platforms in other cities could dramatically increase its impact and applicability. By adapting the developed models and systems for use in different urban settings, we can explore diverse data ecosystems and user interactions. This expansion would not only validate the model's adaptability and scalability but also provide insights into various urban mobility patterns, potentially influencing broader transportation policies and strategies.

These enhancements would collectively elevate the project's analytical depth, operational efficiency, and geographical relevance, making it a more powerful tool in the urban planning and transportation sectors.

REFERENCES

- Citibike Historical Trip Data. (2023) [Trip Data](#)
- Citibike Station Information Data. (2023) [Station Data](#)
- Citibike Station Status Data. (2023) [Live Status Data](#)
- Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., and Xin, D. (2016). MLlib: Machine Learning in Apache Spark. *Journal of Machine Learning Research*, 17(34), 1-7.
- Flask (2023). Flask Documentation. <https://flask.palletsprojects.com/en/2.0.x/>.
- OpenStreetMap Contributors. (2023). OpenMaps. Retrieved from <https://www.openstreetmap.org>.
- Nominatim. (2023). Nominatim Geocoding API Documentation. Retrieved from <https://nominatim.org/release-docs/develop/api/Overview/>.