Bikelytics: Spatio-Temporal Analysis of Bike Trip Data

Rishikesh Gawade
Rutgers University
Piscataway, NJ, USA
rkg63@scarletmail.rutgers.edu

Laxman Srikanth
Rutgers University
Piscataway, NJ, USA
ls1274@scarletmail.rutgers.edu

Dheeraj Goli
Rutgers University
Piscataway, NJ, USA
dheeraj.goli@rutgers.edu

***Abstract*— Bikelytics is an interactive dashboard that visualizes the trips data recorded by the bike-sharing service to help discover patterns in the ridership trends and even identify potential areas of improvement by analyzing how the flow of riders varies over time between any stations or regions of interest. By considering factors such as the seasonality, weather, the rider behavior at any station, demographics and the overall connectivity of any region, we aim to provide actionable insights to the service operators to take measures in scaling and optimizing the service and addressing any bottlenecks that impact the flow of riders. For this project, we will be using the trip history dataset obtained from Chicago's Divvy Bike website [1], and the hourly weather trend [2] obtained from the city administration's official government website. The system uncovers valuable information from the data using sophisticated processing techniques for data visualization.**

***Keywords*— Bike-share system, bike riding, ridership, rider, cities, Chicago, seasonality, weather, temperature, humidity, wind, peak hours, stations, flow clustering, scatter map, trend analysis, Apache Spark, Flask, Pandas, ETL.**

I. Project Description

Nowadays, there is a growing need for sustainable public modes of transport, especially in the urban areas where the population is increasing and so are the expenses on private modes of transport. As a measure, several companies, in coordination with the city administrations, are setting up bike sharing services in these cities and have been continually striving to optimize and expand their services to boost the intra-connectivity of the city using bikes. A significant challenge they face is to maintain the implemented services, as the volatile ridership trends tend to affect the efficient functioning of these services.

We are aiming to help these operators by using interactive visualization techniques to deep dive into the trip data they continually collect. Using **spatio-temporal visualization** [3] techniques like flow map and trend charts, the operators will be able to assess the overall ridership flow and its behavior. Additionally, by analyzing the rider behavior with respect to any station, the operator can assess the **commuting pattern** to and from the station to identify and **address any potential bottlenecks** at the station due to the varying patterns, thereby, allowing them to make its operation more efficient.

Moreover, as any rider is naturally concerned about the atmospheric conditions while deciding to ride a bike, we have identified **weather** as a significant factor, along with the day and time, to have an influence over the ridership trends. Extreme weather conditions directly dictate the ridership trend, and it obviously leads to a significant drop in the usage of the service. Thus, in an attempt to assess the impact of weather on the ridership, we aim to allow visual comparison of trends [4] of various factors of weather and demographics with that of the overall ridership.

A. Stage 1 - Requirements and Data Gathering Stage

Our **primary dataset** is based on the **monthly ridership data** published by some of the most popular bike sharing systems in the US. For now, we are considering only the **Chicago's Divvy Bike dataset** [1] published as **.CSVs** on a monthly basis by the operator. For the scope of this project, we will be considering the data for the entirety of 2021 only, as every month an average of 200K+ trips are recorded. Our data has nearly **3.5 million trips** recorded, collectively having a size of **1.5 GB** approximately. Each record is of size **250 bytes**. Our **secondary dataset** is the **Chicago's weather dataset** available as **.CSV** file for the entire year [2]. It is of size **8 MB**, having nearly **9000 records**, each of size **70 bytes**. We plan to correlate this data with the daily trips data for 2021. Both these datasets are **at rest**, and **spatio-temporal in nature**. The **interesting features** of both these datasets are as follows:

1. Trip Dataset:

- Source Station (ID, name, capacity, latitude and longitude).
- Destination Station (ID, name, capacity, latitude and longitude).
- Ride ID.
- Start Time
- End Time
- User Type
- Birth Year
- Gender

2. Weather Dataset:

- Date.
- Time.
- Temperature.
- Humidity
- Wind Speed
- Precipitation

This application is chiefly designed for:

- **Bike Share Operator:** The users on the operator side would typically be some **business analysts** with no technical background. Through the proposed views in the **Dashboard**, the operator can assess the overall performance of the service by seeing the trends in ridership, number of customers and subscribers, and revenue. They can view the flow of riders between regions and stations using the **highly interactive flow map**[5], and see the least common routes and stations. Through the **Station View**, the operator can view hourly station-wise trends with respect to peak and non-peak hours and type of the day. Finally, through the **User view**, the operator can view how the demographics vary over time and its distribution of the user type.
- **City Transport Council Advisor:** Since these bike-sharing services work in coordination with the **transport department of the city's administration (CDOT for the city of Chicago)**, the advisor in the city's administration can analyze the bike-sharing system's performance and decide whether to provide

any subsidies to the operator and accordingly optimize other transport options like buses, trains, etc. based on the popularity of the bike-sharing system.

- Project Timeline and Division of Labor.

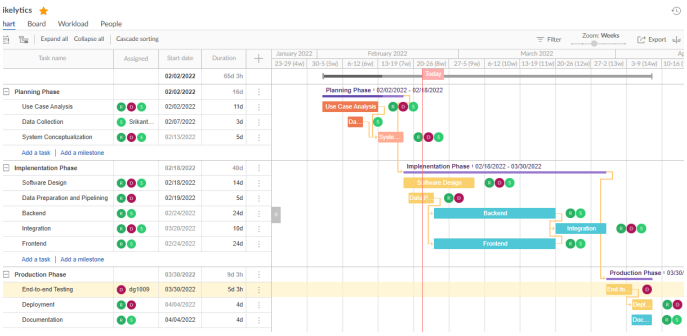

Fig. 1. 8 weeks Gantt chart to be followed to complete the project

- Rishikesh Gawade: Backend, EDA and Interaction Planning and
- Srikanth Sista: UI and Visualization
- Dheeraj Goli: Data Pre-processing and Integration

B. Stage 2 - The Design Stage

- **Data Flow Diagram:** The figure below describes the overall end-to-end system, right from raw, unprocessed data collected from source to providing actionable insights on UI in highly perceivable, visual manner. The flow chiefly consists of 3 key stages - Data Processing, Backend Querying Service and the UI layer.

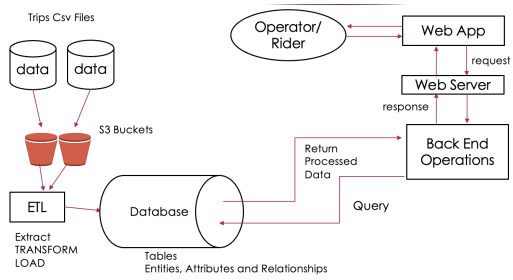

Fig. 2. Data flow chart

- **Data Processing:** We have identified the key features of the datasets and we aim to load them into the database using the ER diagram given below. As the beginning of the pipeline, all the **.CSVs** will be loaded in an **S3 bucket** which will continuously be monitored by some Spark jobs. These jobs will read the new .CSVs and load the data in **normalized form** in the **database** (Amazon RedShift). As the data is quite extensive, it would be time consuming to read the entire data every time for analytical purposes. Hence, we propose to create some **Spark jobs** for pre-processing the most frequently used data for faster querying. In a nutshell, we aim to create hot and cold storages of data.
- **Backend Service:** Moving ahead, the hot and cold stores will queried by our backend service written in **Flask** for catering to the data interaction needs of the user as delegated by the UI layer. This service will have predefined queries for accessing the data with respect to every kind of proposed graphs and charts and the data will be forwarded to the UI in JSON format.

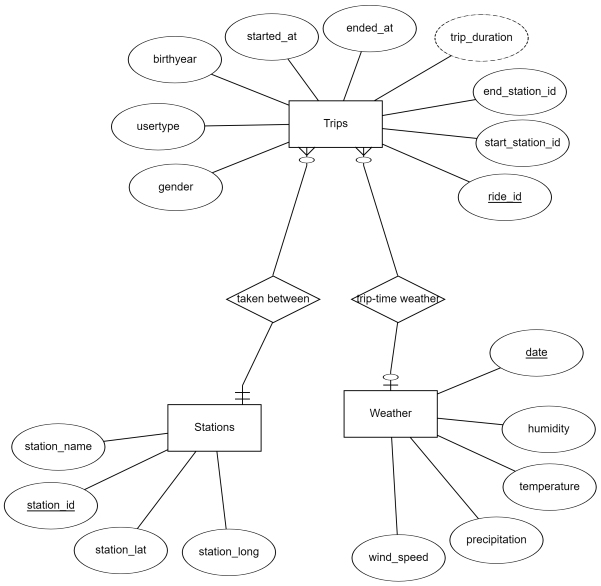

Fig. 3. The ER Diagram.

- **UI Layer:** Based on the users' interactions, the UI layer will submit requests for data relevant to any particular chart to the backend service and upon receiving, it will render the data using the proposed interactive charts implemented using **AMCharts**, **Plotly.js** libraries and **flow map** [5] for spatio-temporal visualization. User can choose to drill down or up into the charts and accordingly more requests can be sent to the backend for fetching relevant data.

C. Stage 3 - Implementation

- Bikelytics, as a web application, employs a client-server architecture with **MVC (Model, View, Controller)** design pattern, wherein the user interacts with the views to control what data the model will return to create a new/updated view.
- At the backend, we have a Python script running that checks for new trips' data files in a designated folder, and processes any new CSV files that are dumped into it, to prepare data for all the views as follows:
  - For the main **Dashboard**, the script processes and counts the daily number of rides for the trend analysis, the **performance metrics** like revenue, ridership change, and subscriber and customer variation. It also calculates **route-wise** ride count for **flow anaylsis**.
  - For the **Station's page**, it creates a station-wise summary of daily trends, usage as per day type and time, and hourly incoming and outgoing behavior.
  - For the **User's page**, it computes the the trends and the statistics of the **demographics** of the riders.
- The UI has a Bootstrap enabled theme by **AdminKit**[6] for overall layout. It allows arranging all the content in cards. It makes the overall UI modular and allows easy shifting of views.
- The charts have been created using charts and figures from AMCharts[7], Plotly.js[8] and Flowmap.Blue[5]. All these charts and figures are already interactive, and allow adding more interactivity as they readily handle different events based on how user interacts with them. Using simple logic in **Javascript**, these charts have been linked with each other.

D. Stage 4 - Visual Representation

- In Bikelytics, there are three main interactive pages that the user or admin can play around with. They are dashboard page, stations page and user stats.
  - The main **Dashboard** contains two line graphs comparing the ridership data with the weather trends.
  - The **Station** page contains the map with bike stations marks. A user can select any station and all the corresponding charts are synchronised according the selected station. It contains two bar graphs with hourly outgoing and incoming stats, and a heat map and all the charts are dynamic which change as per the selected user station.
  - The **User's** page contains dynamic charts that can be customised by the user which show User Distribution Based on Gender and User Type, and Usertype-wise Age-group Distribution.

E. Stage 5 - Interactions

We have strived to make the user experience as smooth as possible by trying to exploit all possible interactions that are readily available in the charts we have implemented. Additionally, as these charts and figures also allow for handling of events triggered upon **user interaction**, we have used JavaScript to link these charts and figures together. As a result, when a user interacts with one chart, the other linked charts and plots get updated as per the selected data.

As an instance, the **scatter map** implemented on the Station's page allows the user to select any station of interest. This interaction is called as **vertex selection** [9]. Upon selecting, both **barcharts, line chart and the heatmap** below get **updated** to display the **information relevant to the selected station**. In case the user doesn't know where a particular station of interest is on the map, we have also provided dropdown to allow the user to select the station from there. Upon selecting, the map gets updated to show the selected station. This **linking** between the map and the dropdown is **bidirectional**.

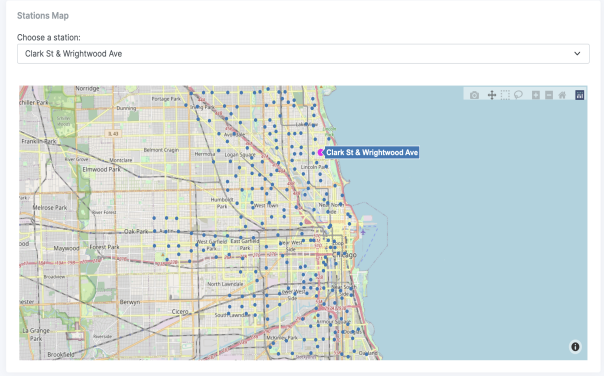

Fig. 4. Scatter Map used to display all the stations in the city. User can click on any point, or select the station from the dropdown and view the statistics and trends given below.

Another complex instance is present in the the **flow map** [5] implemented on the Dashboard page. Here, along with selecting a station, the user can also select the route connecting 2 stations, allowing him to assess the strength of the connectivity between the 2 stations. This interaction is called as **edge selection**.

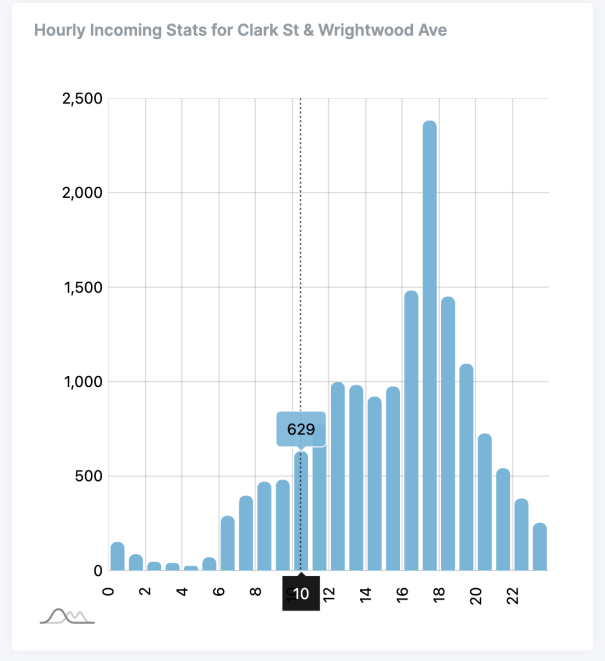

Fig. 5. Hour-wise number of rides coming in at the selected station.

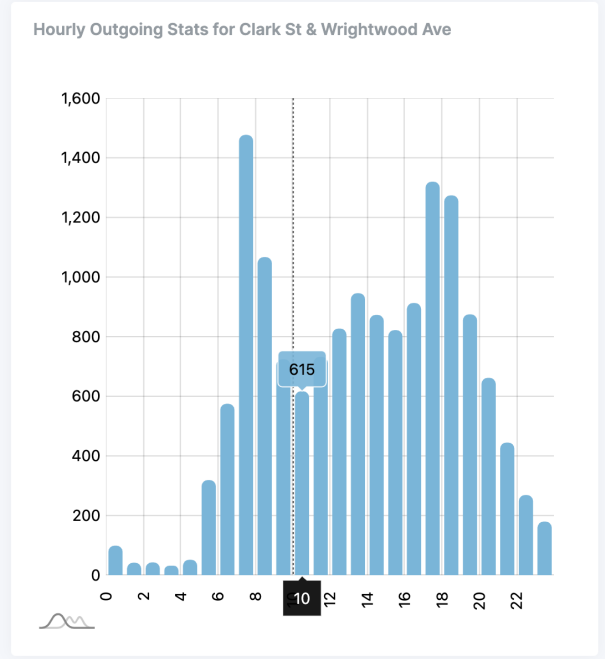

Fig. 6. Hour-wise number of rides going out from the selected station.

These interactions, altogether, help complete the MVC, as it allows the interacts with the view and it helps him control the model to determine which data needs to be displayed. Overall, the following **interaction mechanisms** have been incorporated throughout the application:

- Dragging
- Hovering
- Zooming

- Vertex Selection
- Edge Selection
- Linking of different views.
- Brushing
- Sliders
- Labeling

II. Insights

A. Questions Posed and Addressed

Using the rides data, the operator expects to have basic questions answered, so that he is able to go deeper into the visualized data to extract actionable insights and thereby make decisions to improve the service.

- Firstly, an operator would be interested in **determining how the bike-sharing system is performing**. He would be interested in getting some **important performance metrics** on the daily basis, and the **overall trend** of its **usage**.
- Secondly, as the overall **goal** of the system is to **boost connectivity** between all feasible parts of the city, the operator would be interested in seeing the **strongly and weekly connected regions** in the city, and the **overall flow** between the regions.
- Consequently, while trying to answer the previous questions, he would be inherently curious to know the **least popular routes** and **stations** used by riders, so that he can think of taking measures to boost their popularity.
- To dive deeper into **route-wise analysis**, he would be willing to see how the **stations** on any route of interest are **performing**, which is why he would want to see its usage trend on daily basis, as well as with respect to **hour of the day and type of the day**.
- Often, if at all any **bottleneck** occurs in the system, it happens at a set of stations. Thus, the operator would be willing to know where and at what time there is a **chance for a bottleneck** to occur at any given station.
- Every time someone takes a ride, the system records **basic rider information** like gender, age, and whether he is a customer or a subscriber. Thus, an operator would be curious to know **how these demographics play a role in the overall ridership trends**. For example, what percent of riders in the previous month were customers, and what was the most common age group among those, etc.
- As a user, whenever someone thinks of riding a bike, he/she naturally thinks of the **ongoing weather situation** and based on that he/she decides to ride or not. Thus, an operator would be willing to know **how much of an impact the weather has caused on the ridership trend**.

B. Analysis

1. Dashboard Page:

Overall, each page consisting of different views tells a short story in itself, which when put together, tells the big picture. The **Dashboard** tells the overall performance of the system over time, depicts the **flow** and **popularity of the routes** and the stations using the rides recorded so far.

At the top, the cards (Fig. 7) display the main **performance metrics** of interest, namely - **ridership change**, **number of subscriber to customer change**, and **revenue growth or decline** - all in comparison with the **previous month**. These high level metrics are important as they form the most critically describe the performance of the system.

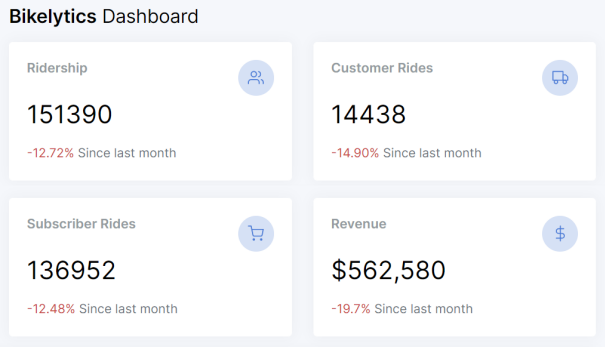

Fig. 7. Performance of the service in comparison with the previous month.

These metrics can be coupled with the **trend chart** shown below (Fig. 8) to determine the behavior of the system. For example, as an operator, anyone willing to see that the **customers are turning into subscribers**, as the subscriptions keep the riders tied to the ride-sharing system for longer time. Hence, in case the user notices a decline in the number of customers, but a proportional increase in the number of subscribers and revenue, while the ridership trend remains constant at least, it indicates that possibly a considerable population has bought subscriptions for riding.

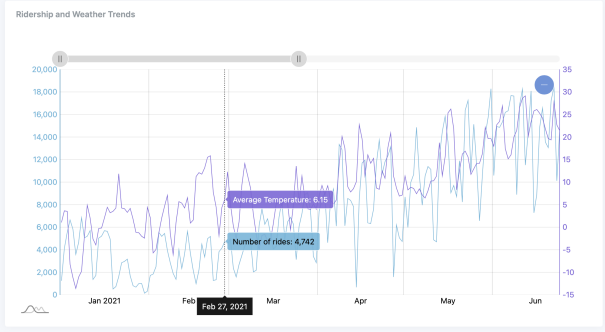

Fig. 8. Ridership Trend for the year 2021 along with the changes in Weather.

Looking at the **ridership trend** chart only, which also has the weather trend incorporated into it, the operator can study the trend and would be able to assert that whether **fall in the ridership** occurred due to the **change in weather or seasonality**. For example, a sudden drop in temperature coupled with a sharp increase in wind speed for the day would make the weather less favorable for the riders to opt for a bike ride.

However, this case will not be always true. It can happen that the ridership fell due to some **bottlenecks** in the system, especially at some of the most used stations that dictate the overall flow of bikes between the stations or urban regions. For example, for some huge festival-like event taking place in the downtown of Chicago can mostly like cause the stations near the venue to be flooded with bikes, while also leaving some stations from where the riders traveled to the venue devoid of bikes.

Thus, due to this overflow of riders into a region, the operator would be interested in **evaluating the flow of riders** to the origin stations, to compare with the outflow. To help with this, the **flow map** implemented (Fig. 9) will allow the operator to **study the flow patterns** and **variations** in them **over time**. The flow map allows **spatio-temporal analysis** of the ridership trends, as it depicts regions of strong **connectivity** using **directed edges** of varying thickness between stations and regions. Thus, if anyone **selects a pair of station**

and study the directed edges between them coupled with the **overall inflow and outflow** from both stations, the operator will be able to determine if any of the station needs addressing in the form of any action so as to avoid any **potential bottlenecks**.

Fig. 9. Flow Map depicting the flow of riders between regions and stations, with the edges whose thickness encode the strength of connectivity between the stations.

Additionally, as the operator continually strives to **improve the connectivity** and **boost ridership**, he can use the same flow map to analyze whether his actions have led to a stronger connectivity and/or a flow between regions. In general, he can study how the flow patterns in the city changes over time, and make decisions on which regions he can target to boost the ridership on priority. For eg. the Fig. 9 shows the flow of riders for the month of February of 2021 (selected in the left window).

Fig. 10. Station selection highlights only the paths coming in and going out of the selected station/region. Hovering over the directed edge shows the number of trips taken between the 2 stations connected by the edge.

In a bid to boost the ridership in any targeted region, the operator will be interested in knowing the **least popular routes** (Fig. 11) and **stations** (Fig. 12), and assess the factors causing these routes and stations to be underutilized. It could be that the listed route is an outlier, i.e. some rider went wild and took a trip from the northernmost station to the southernmost one. In other cases, the operator would be interested inspecting the stations on this particular route individually. For this, the operator should switch to the **Station's page**.

Least Popular Routes

Start Station Name	End Station Name	Number of Trips	Average Trip Duration (Seconds)
Western Blvd & 48th Pl	Millard Ave & 26th St	13	2102
Lincoln Ave & Diversey Pkwy	Sawyer Ave & Irving Park Rd	32	1878
Lincoln Ave & Diversey Pkwy	Oakley Ave & Irving Park Rd	51	1385
California Ave & Lake St	Halsted St & Roscoe St	58	983
California Ave & Lake St	Broadway & Waveland Ave	66	1619
California Ave & Lake St	Southport Ave & Clybourn Ave	71	2011
Western Ave & Monroe St	Southport Ave & Clybourn Ave	72	1471
Lincoln Ave & Diversey Pkwy	Troy St & Elston Ave	89	682
Kedzie Ave & Lake St	Broadway & Sheridan Rd	103	827
Halsted St & Maxwell St	Smith Park	119	1102

Fig. 11. Least popular routes throughout the year.

Least Popular Stations

Station Name	Number of Rides
Western Blvd & 48th Pl	317
Sawyer Ave & Irving Park Rd	514
Millard Ave & 26th St	683
Oakley Ave & Irving Park Rd	851
Broadway & Waveland Ave	1170
Troy St & Elston Ave	1367
Broadway & Waveland Ave	1431
Kedzie Ave & Lake St	1732
Southport Ave & Clybourn Ave	2135
California Ave & Lake St	2473

Fig. 12. Least popular station throughout the year.

2. Station's Page:

Upon knowing the routes of particular interest that the operator would want to study, he/she strives to dig deeper into the route by inspecting the stations that define the route. On the Station's page, the operator can select any station of interest either through the scatter map, or the drop-down list, in case he/she doesn't know the exact location of the station on the map.

Upon **selecting a station of interest** (Fig. 4), the operator can get insights through the daily trend and hour-wise average usage of the charts. By comparing the **hour-wise incoming trips** (Fig. 5) and **outgoing trips** (Fig. 6), the operator can look out for a potential bottleneck causing at that station. For example, for 5 PM, the average number of incoming rides is way more than the outgoing numbers. This causes a bottleneck because the incoming riders can potentially not find a spot to dock their bikes. This works as an actionable insight to the operator, as he then takes a call to reshuffle the bikes at the station to some other station.

On the flip side, if the number of number of outgoing trips at 5PM is way more than the number of incoming trips, then the station is very likely to run out of bikes. This, again, works as actionable insight to the operator, as he can call to supplying the station with some bikes to ensure smooth flow of riders originating to and from the station.

Coupling these bar charts with the **trend chart** (Fig. 13), the operator can ensure that if the **trend of ridership** to and from the station is steady, then his actionable insights have brought a balance to the station.

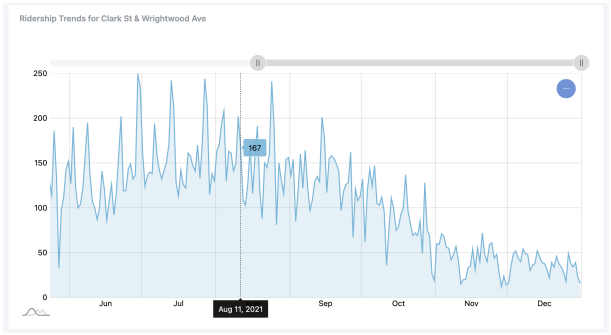

Fig. 13. Ridership Trend at the selected station.

The **heatmap** (Fig. 14) captures the **general usage** of the station with respect to **hour of the day** and **day of the week**. This gives a detailed idea of when the station is mostly used by riders. For example, if the operator sees dark blocks from 8 AM to 18 PM on all the weekdays, then it is most likely used by people who commute to work. This implies that the riders are more likely to be subscribers, and belong to the working age-group of the population. Thus, such demographics can potentially be inferred from the heatmap.

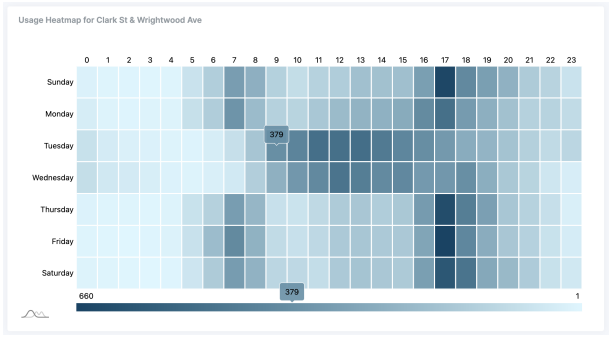

Fig. 14. Heatmap for the selected station.

3. Users' Page:

The Users' page visualizes the **basic demographics data** obtained from the riders. It depicts how the **distribution of riders** with respect to their **type**, **gender** and **age-group** changes over **time**. In the line-bar chart (Fig. 15), the operator can assess the gender-wise variation in the riders over time. It also helps him assess the variation in riders with respect to their type (customer or subscriber). The operator can, thereby, think of incentivizing the targeted users based on their type, gender or age-group, or even any combination of these three to encourage them to use the bike-sharing system more.

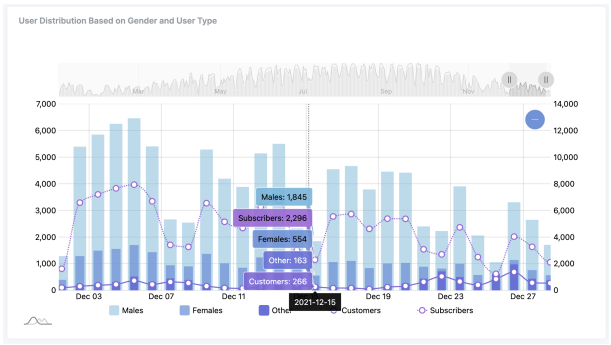

Fig. 15. Demographics Trend with respect to type and gender of the users.

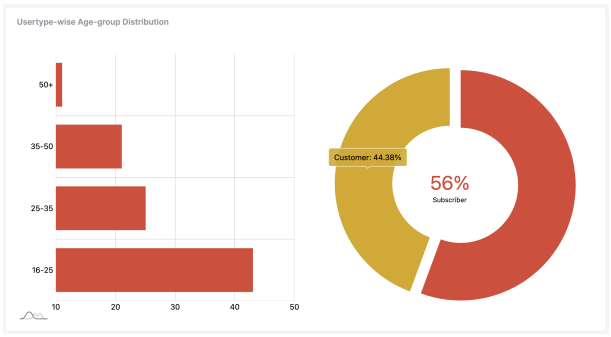

Fig. 16. Summarizing the overall usage with respect to the User type and drilling down to see age-group distribution.

III. Evaluation

Overall, we have striven to realize as much potential out of the data as possible without using any algorithms. As the data is transactional in nature, has timestamps and location information, as well as some categorical information, we are able to exploit a huge amount of information by trying to visualize it using different views and even linking the views to create a coherent flow in the information being deduced from the visuals.

The interactions listed above enable us to **seamlessly update views** based on the actions performed by the user, allowing him to perform pinpointed analysis. For example, selecting a station on the Station's page allows him to extract necessary information regarding the selected station. Thus, overall, using the views created in the three pages, the operator can summarize the high-level picture of the service, that includes **its overall performance and trends, describe the flow between regions and stations, and inspect any station** to determine whether it meets the expected performance or if there are any potential bottlenecks.

The generic nature of the data and all the charts and maps allow this system to be **reused for any bike-sharing systems** that have been setup in other popular cities. For example, the NYC Citi Bike, Metro Bike Share in Los Angeles can adopt this entire dashboard, and by simply plugging the data, they can derive the same information that we have striven to do in this project.

However, there are **certain limitations** to the type of data being used in this system. This system allows route-wise and station-wise analysis because the data is based on predefined stations that have been installed throughout the city. In case anyone wants to use, say **the Taxi rides data with this system, as the pickup and drop locations are entirely random,** there can be infinite possible routes present in the city. Hence, in such cases, route-wise and station-wise

analysis is impossible as there are no predefined pickup and drop locations.

As an effective way of evaluation, a contest helps in gathering different perspectives on what information is being visualized. Throughout the class, by the method of peer evaluation, we received some valuable suggestions. One key suggestion was to try finding a way to visualize the flow of riders between station. For this, after doing a lot of exploration, we were able to fixate on using the **flowmap [5]**. Another valuable suggestion was to allow selecting a station using **a dropdown list**, as a user would generally find it challenging to locate a particular station on the map.

IV. Future work

- We wish to **extend our analysis of our project to all major metropolitan cities**. Once we have data for different cities and states we would be able to improve our analysis further and find patterns. This indeed would help the bike renting companies to show targeted which would improve their revenue even further
- **Build a mobile version** of the application so that people can get insights on the go.
- **Compare the usage of different modes of transport** in the city. This would give us insights into different locations where bike ridership is less and we can take measure accordingly to improve it.

References

[1] (2021) Chicago divvy bike data. [Online]. Available: https://ride.divvybikes.com/system-data

[2] (2021) Chicago weather data. [Online]. Available: https://data.cityofchicago.org/Parks-Recreation/Beach-Weather-Stations-Automated-Sensors/k7hf-8y75/data

[3] (2015) Flow clustering analysis of mobility. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0137922

[4] (2019) Effects of weather on bike mobility. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0966692318307282

[5] I. Boyandin. (2019) Geographic flow maps for visualizing movements between locations. [Online]. Available: https://flowmap.blue/

[6] [Online]. Available: https://adminkit.io/

[7] [Online]. Available: https://www.amcharts.com/

[8] [Online]. Available: https://plotly.com/javascript/

[9] K. J. E. G. Keim, D. and F. Mansmann, "Solving problems with visual analytics," 2010.