MTA HACKATHON

TEAM 02 & 09

1. DATA

a. Count the number of people who will use the train at a day level using the number of sales made each day. We can argue that the number of sales made on a specific day will, on average, be the number of people traveling on the train on that day. We have data for 1 year which we can use. However, we would need to train the model with current data as the usage in COVID-19 situations will be different.

b. Count the number of people in a specific train at a particular station using the number of ticket scans, the train's intended destination, information regarding the route of the train (inbound/outbound), and the train's status (early/on-time/late) to map the train's approximate location.

c. Get all the details for each station such as the accessibility (number of working elevators and escalators), and the number of people in the station. The number of people in a specific station can be approximated by counting the number of scanned tickets with that given station as either the source or the destination. (Very likely that the actual number will be higher than the calculated number, which will increase the level of safety for riders).

d. The occupancy of each car in a train would be a helpful parameter in the model.

e. Typical busyness of a station would be another useful parameter. (For example, Grand Central Terminal is significantly more busy than Tuxedo). This information can be evaluated using source and destination information from tickets.

2. FORECASTING

a. Apply time series models to predict the number of passengers on the daily level for the next week (In the future, the number of predicted days can be modified). This model needs to run every day in order to obtain an updated forecast.

b. Similarly, apply time series models to predict the number of passengers on the train for the next 8 hours of the same day.

3. APP INPUTS

a. The user will have two options. Either they can view how busy the trains and stations are for the entire next week, or they can view the current status of the busyness of the trains.

b. If the user chooses to view the current status of the trains, they can specify the station, route, and time and obtain a view of the timeframe with colors (red for busy, orange for moderately busy, and green for low usage).

c. If the user chooses to view the next week, a similar view can be seen on a day by day basis for the entire course of the week. This will be based on the forecasts made.

4. LOGIC FOR THE COLORS:

a. We would like to apply a machine learning model, but since we do not have the required annotated data, we can initially use a rule-based model. We will use the number of scanned tickets for the station in the route and the number of users who have searched for the station and route in the app. For example, if the users chose a day by day view :-

- **i.** If Number of scans > safety threshold then red
- **ii.** If Number of scans + num of searches > safety threshold then red
- **iii.** Occupancy of all cars red then red
- **iv.** Number of scans + Number of search > 0.8* safety threshold then orange
- **v.** If %elevators working < 60% then orange
- **vi.** Else green

b. Case when a weekly view is chosen

- **i.** If Number of searches > safety threshold Then red
- **ii.** If forecast > safety threshold then red
- **iii.** Forecast < safety threshold and number of search > safety threshold then orange
- **iv.** If %elevators working < 60% then orange
- **v.** Else green

We can use a decision tree with the above parameters using labels assigned as soon as we receive access to sufficient data to replace this rule-based model.

5. MODEL API ENDPOINT

http://dcba10cc-10a6-46df-a2ae-5285d2d8b8bc.eastus2.azurecontainer.io/score