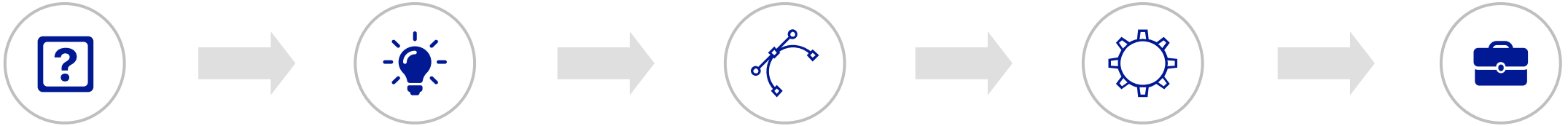
CITIBIKE STATION REBALANCING

CHRISTIAN OPPERMAN
MELANIE ZHENG
PAUL CHOI

Photo by Anthony Fomin on Unsplash

TABLE OF CONTENTS



WHY CARE? THE PROBLEM

A key issue behind CitiBike customer retention comes from poor bike and dock station availability

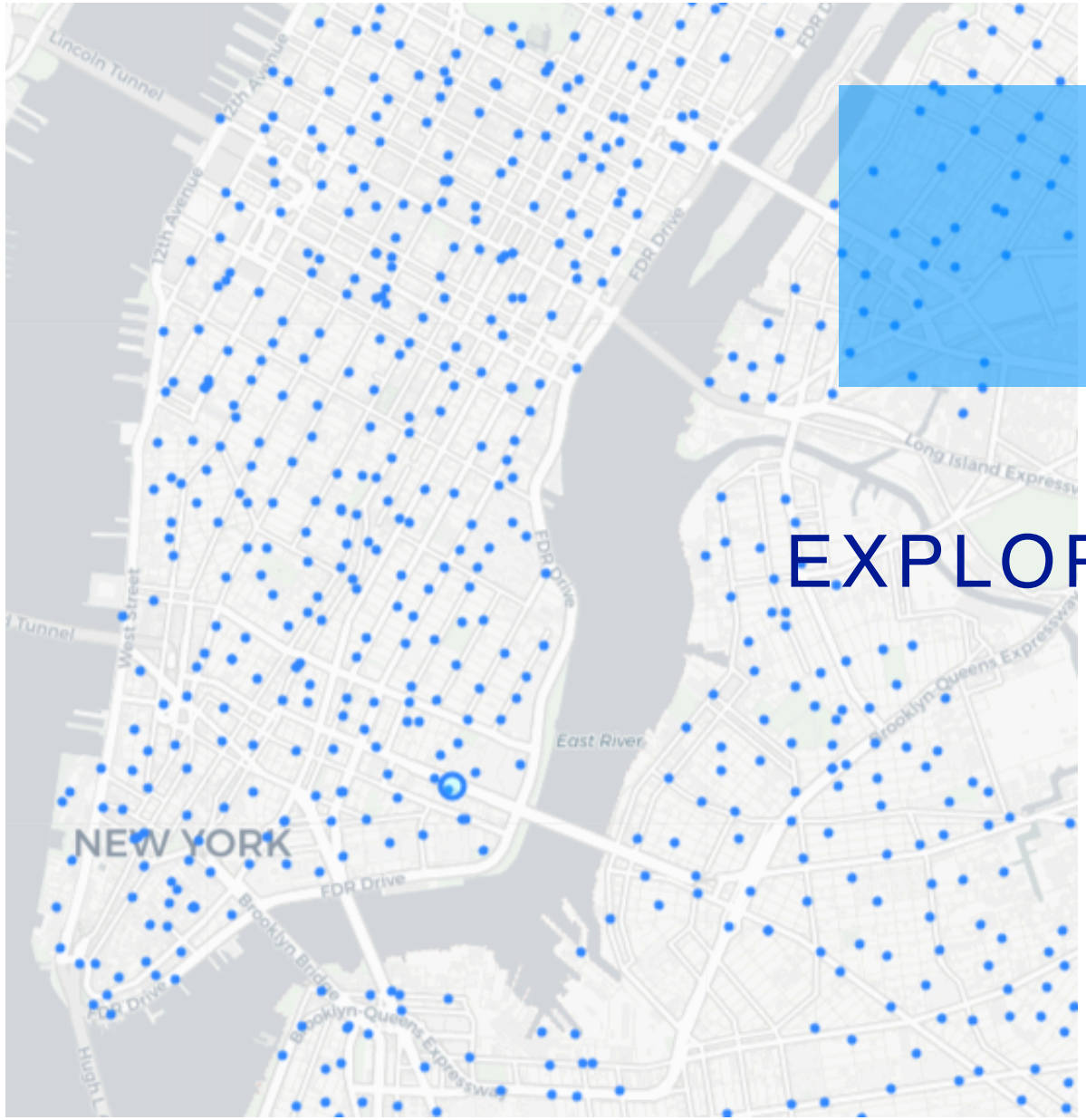
Examples of rebalancing criticism from past CitiBike users:

- "I didn't renew because, I didn't like the system. Too often I left work in Midtown and there were no bikes."
- ".. many common commutes have no bikes available at the time you need one. Midtown East is barren of bikes by late afternoon. The Lower East Side clears out by 8am. There are people who saw the promise of a bike share program, but the reality is not as useful for their situations."
- "I am an annual Citi Bike member," tweeted Samantha Chang. "This bike station at Columbia University is always poorly stocked."
- "It's inconvenient," said Denise Gruber, 43. "Sometimes they're all gone or broken. Sometimes they're all full. You have to go looking around [for a return dock]. People don't want to be overcharged because they're riding around looking for a spot."
- "I do [wish] the system had a better balance of bikes; far too often I have gone out for a bike at 10pm and not been able to find one in FiDi."

Sources: https://nyc.streetsblog.org/2018/09/26/its-not-your-imagination-something-is-seriously-wrong-with-citi-bike-right-now/
https://www.wnyc.org/story/more-half-citi-bikes-early-members-not-renew/
https://nypost.com/2015/01/12/big-drop-in-citibike-use-during-2014/

OUR SOLUTION

- Our group (a team of CitiBike data scientists) has been asked to re-evaluate CitiBike's station rebalancing strategy.
- We have developed a CitiBike station rebalancing application in Python using machine learning to create an optimal station rebalancing strategy.
- Given a date and time, we can predict **(1)** outgoing and incoming bike demand and **(2)** depletion status for each station in Manhattan.
- Our application also dynamically generates a paired list of stations to/from which to move bikes to ensure maximum rider fulfillment.



EXPLORATORY DATA ANALYSIS

UNDERSTANDING THE DATA

CitiBike anonymous trip system data (link)

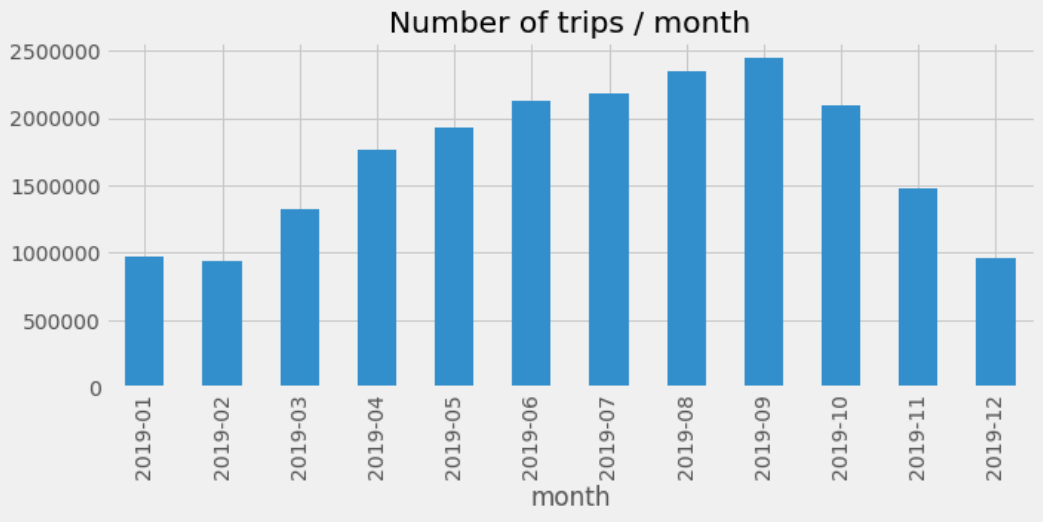
- Monthly datasets from Jun-2013 to Jan-2020
- 80+ million total observations, 15 variables
- Most useful from a ridership perspective

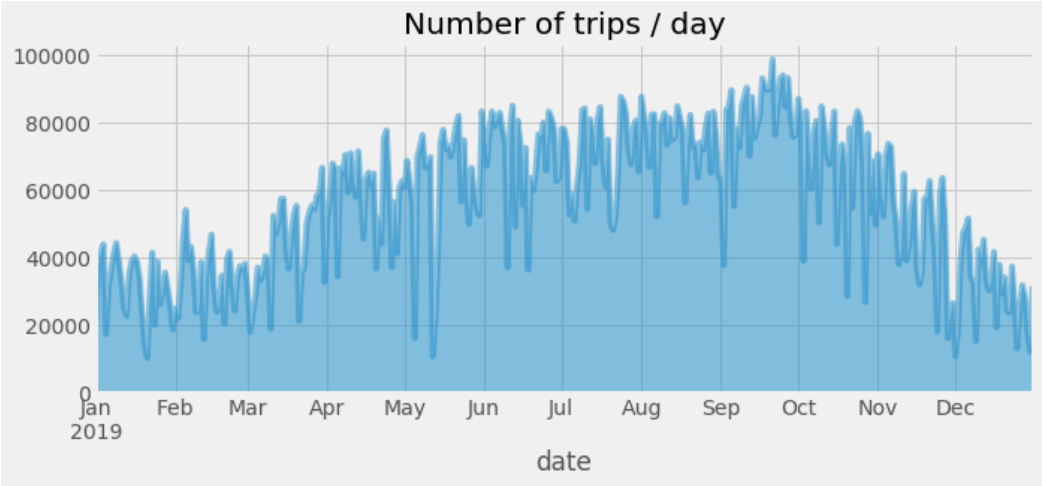
The Open Bus bike share data (link)

- Monthly datasets from Mar-2015 to Apr-2019
- 35+ million total observations, 14 variables
- Most useful from a dock stations perspective

CITIBIKE 2019 YEAR-END DATA SNAPSHOT

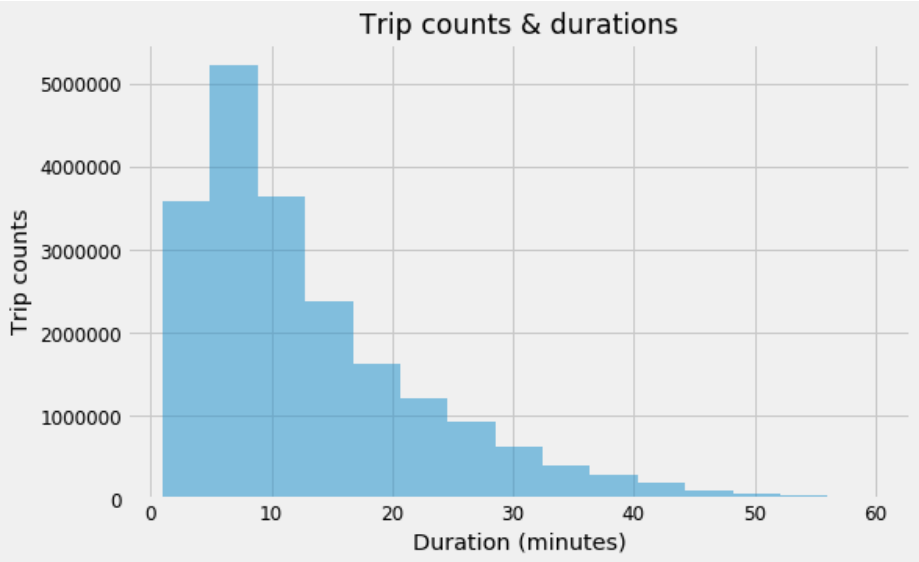
Total stations	856 active stations
Total bikes	~12,800 bikes*
Total trips taken	20.5 million
Average trips per day	56,305
Average duration per trip	16.3 minutes
% of trips taken by annual members	86%

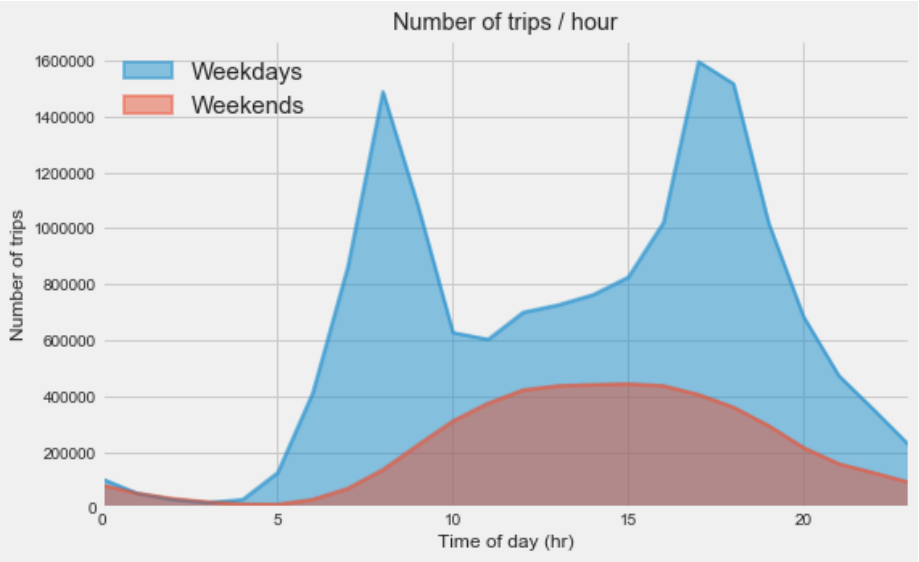


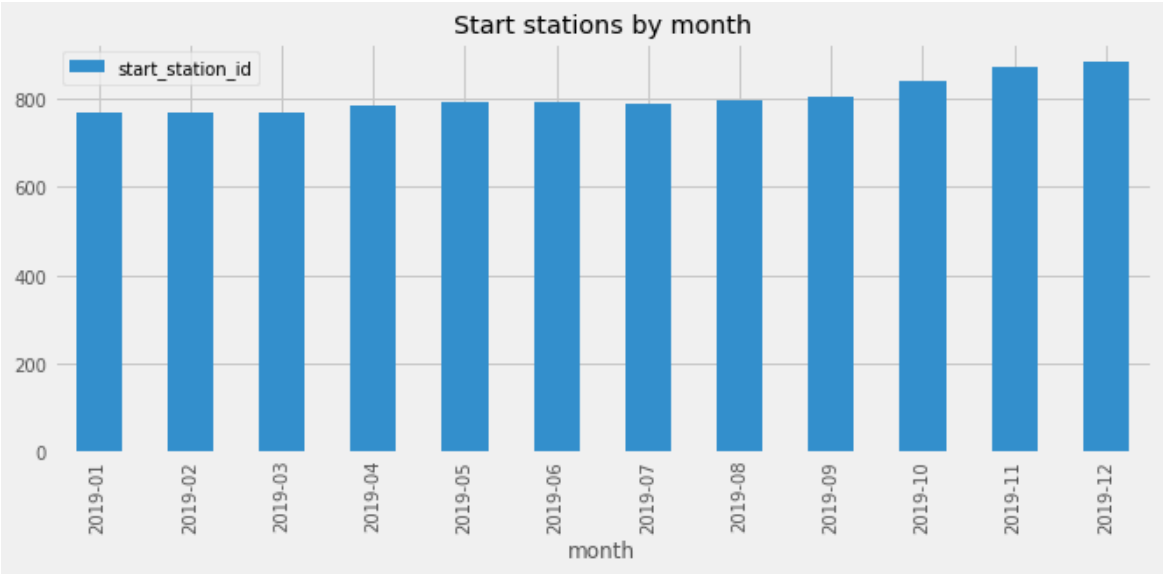


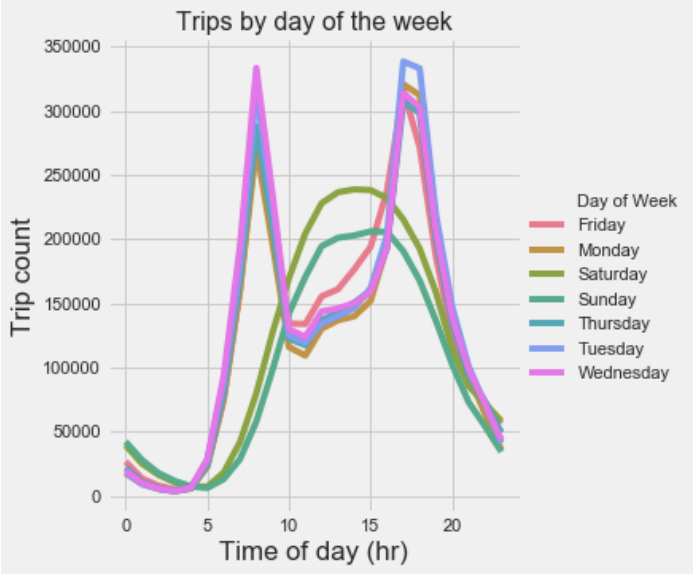
Note: *comes from CitiBike's monthly operating report for Dec-19. This number compares with our unique 'bikeid' for the month of Dec-2019 at 13,816 and 8,220 on 12-31-2019, which we attribute to definitional differences. We assume a bike could have a different 'bikeid' after being removed from the active fleet for repair & maintenance and then re-added.

CITIBIKE SYSTEM DATA INSIGHTS



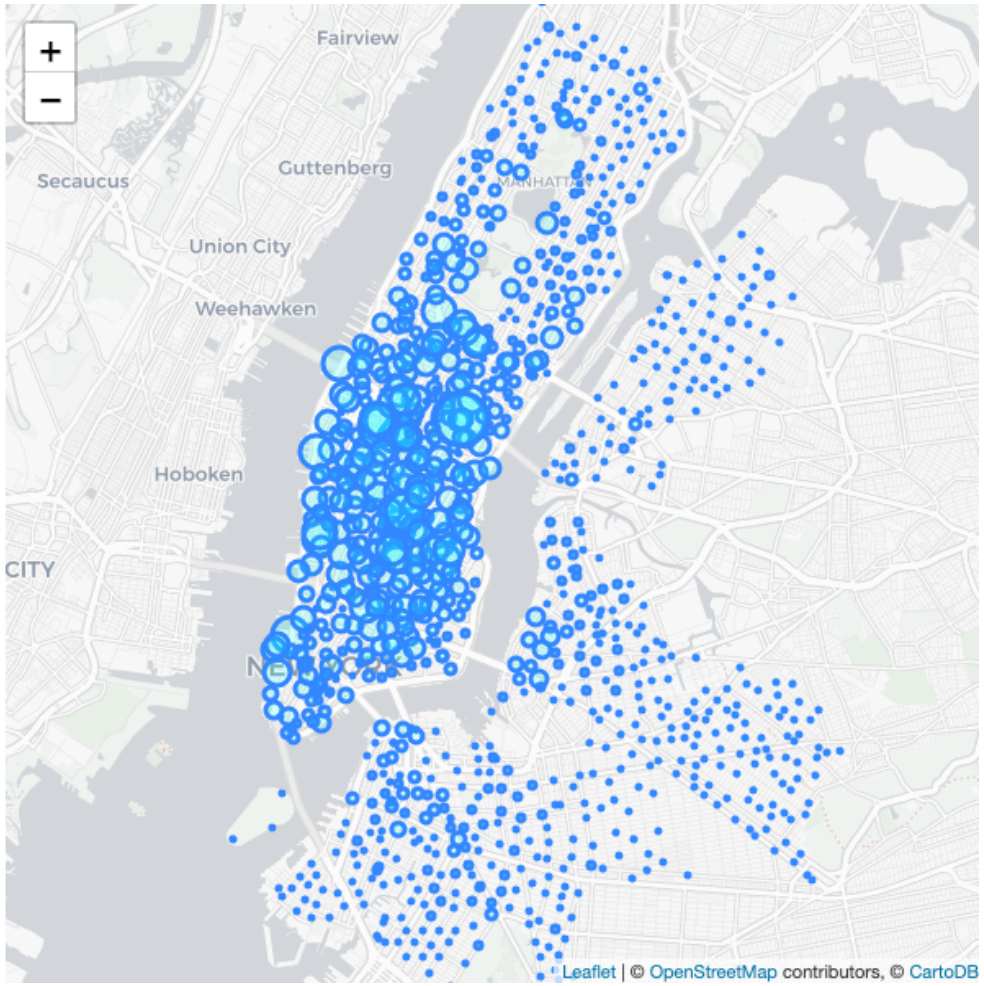






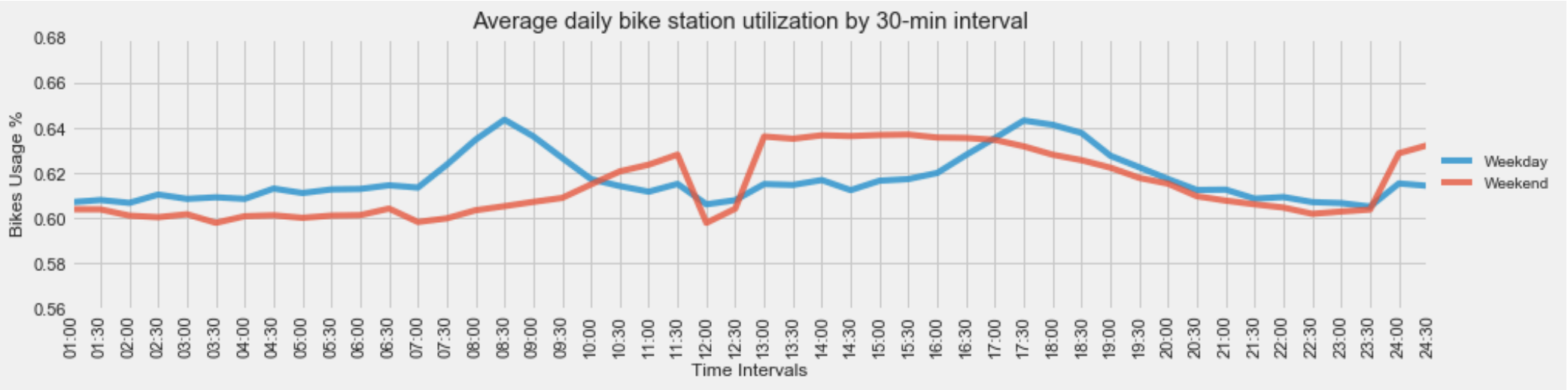
CITIBIKE SYSTEM DATA INSIGHTS (continued)

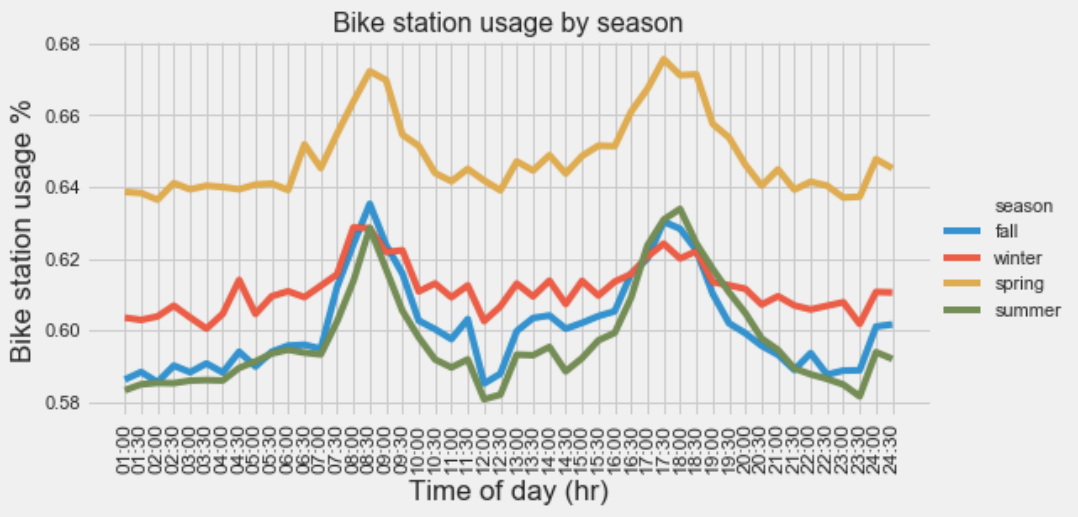
Total trip counts by start station

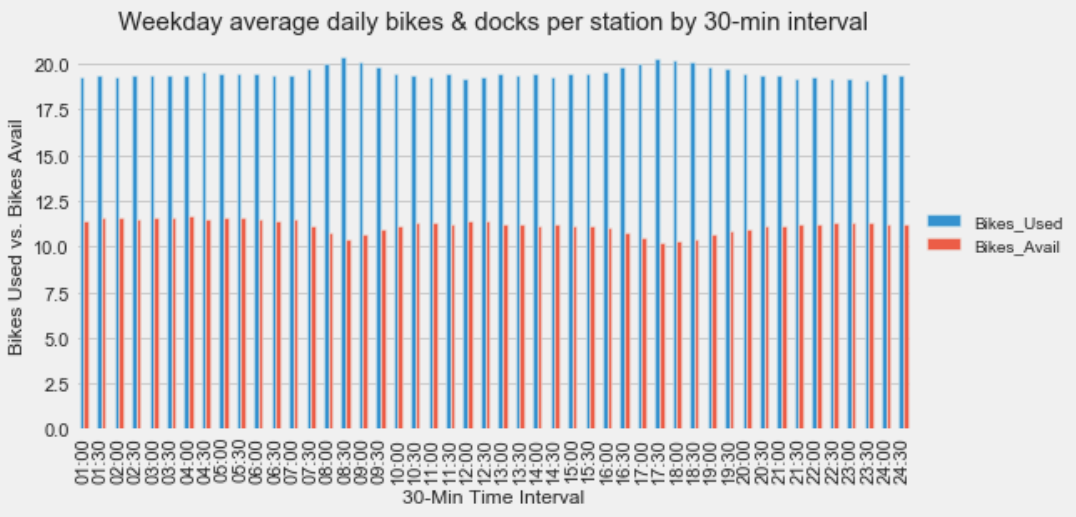


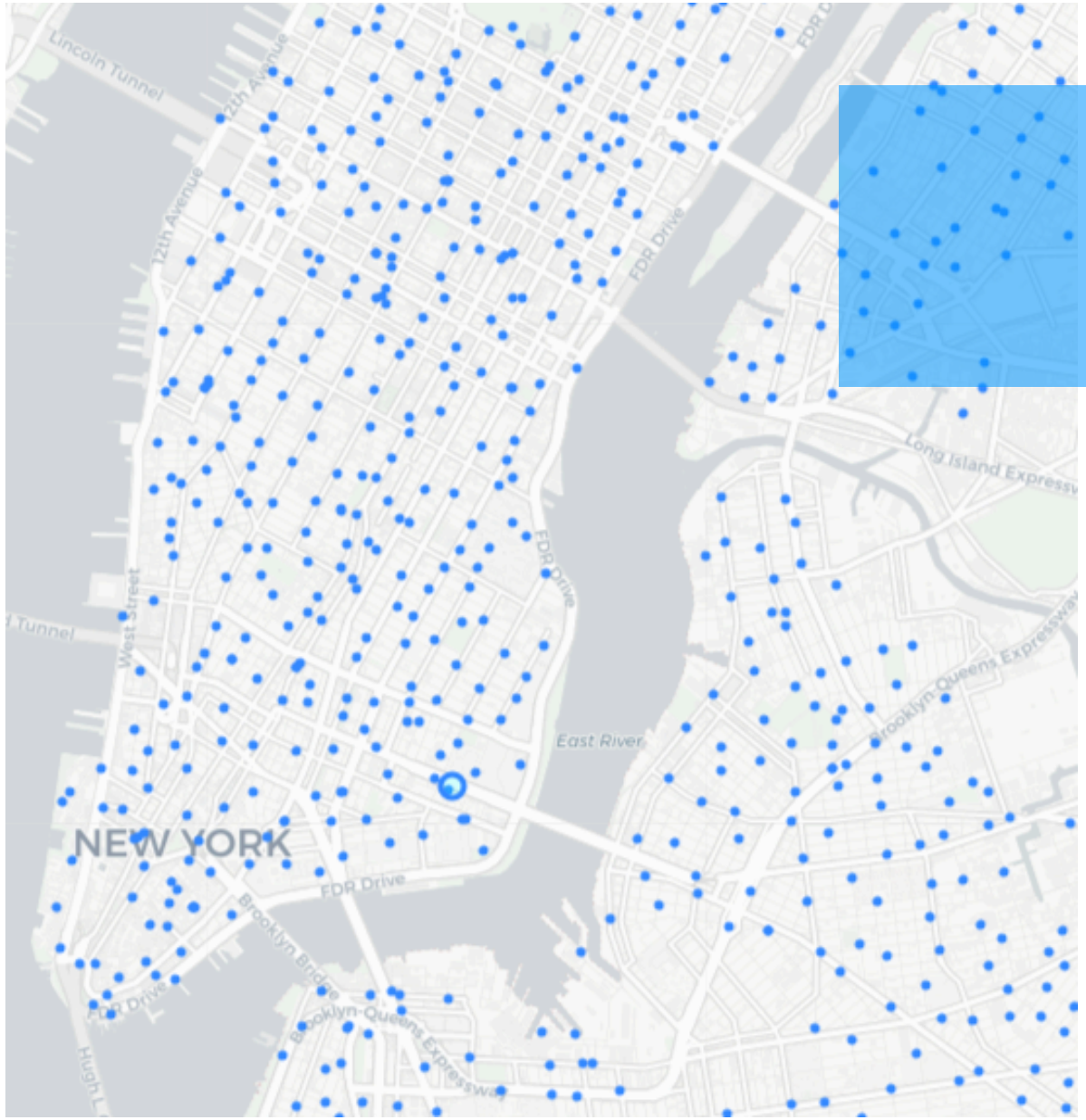
Average trip durations by start station

OPEN BUS DATA INSIGHTS









DATA PROCESSING, MODELING & RESULTS

OVERVIEW OF PIPELINE

Steps

- Data Cleaning
- Training Machine Learning Models
- Predictions
- Generate Rebalancing Strategy

DATA PROCESSING

Bike/Dock Demand

'tripduration'
'starttime'
'stoptime'
'start station id'
'start station name'
'start station latitude'
'start station longitude'
'end station id'
'end station name'
'end station latitude'
'end station longitude'
'bikeid'
'usertype'
'birth year'
'gender'

~80 million rows

Outgoing Rides: (~20million rows)

	start station id	starttime_date	season	dayofweek	starttime_interval	outgoing_bike_count
0	72	2015-01-01	winter	Thursday	11:30	1
1	72	2015-01-01	winter	Thursday	12:30	1
2	72	2015-01-01	winter	Thursday	15:00	1
3	72	2015-01-01	winter	Thursday	21:00	1
4	72	2015-01-01	winter	Thursday	7:00	1

Incoming Rides: (~20million rows)

	end station id	stoptime_date	season	dayofweek	stoptime_interval	incoming_bike_count
0	72	2015-01-01	winter	Thursday	10:30	1
1	72	2015-01-01	winter	Thursday	12:30	1
2	72	2015-01-01	winter	Thursday	17:30	1
3	72	2015-01-01	winter	Thursday	18:00	1
4	72	2015-01-01	winter	Thursday	19:00	1

DATA PROCESSING

Bike/Dock Demand (cont.)

	start station id	starttime_date	season	dayofweek	starttime_interval	outgoing_bike_count	bike_demand
0	72	2015-01-01	winter	Thursday	11:30	1	Low
1	72	2015-01-01	winter	Thursday	12:30	1	Low
2	72	2015-01-01	winter	Thursday	15:00	1	Low
3	72	2015-01-01	winter	Thursday	21:00	1	Low
4	72	2015-01-01	winter	Thursday	7:00	1	Low

	end station id	stoptime_date	season	dayofweek	stoptime_interval	incoming_bike_count	dock_demand
0	72	2015-01-01	winter	Thursday	10:30	1	Low
1	72	2015-01-01	winter	Thursday	12:30	1	Low
2	72	2015-01-01	winter	Thursday	17:30	1	Low
3	72	2015-01-01	winter	Thursday	18:00	1	Low
4	72	2015-01-01	winter	Thursday	19:00	1	Low

DATA PROCESSING

Bike/Dock Demand (cont.)

```
1  outgoingDF.outgoing_bike_count.describe()
```

```
count    1.98e+07
mean     3.19e+00
std      3.47e+00
min      1.00e+00
25%      1.00e+00
50%      2.00e+00
75%      4.00e+00
max      1.57e+02
```

Low

Medium

High

```
1  incomingDF.incoming_bike_count.describe()
```

```
count    1.99e+07
mean     3.18e+00
std      3.43e+00
min      1.00e+00
25%      1.00e+00
50%      2.00e+00
75%      4.00e+00
max      1.28e+02
```

Bike Demand

- Low : if outgoing bike count <=1
- Medium : if 1 < outgoing bike count <=4
- High: if outgoing bike count > 4

Dock Demand

- Low : if incoming bike count <=1
- Medium : if 1 < incoming bike count <=4
- High: if incoming bike count > 4

DATA PROCESSING

Station Depletion Status

‘dock_id’
‘dock_name’
‘date’
‘hour’
‘minute’
‘pm’
‘avail_bikes’
‘avail_docks’
‘tot_docks’
‘_lat’
‘_long’
‘in_service’
‘status_key’

~35 million rows

- Remove values that don’t make sense (ex. total docks > 2000).
- Clean available bike and available dock values.
- Convert ‘date’ to datetime format.
- Convert hours to 24 hour format and sort minutes into 30-minute increments.
- Create column for depletion status.
  - Based on ‘avail_bikes’/‘tot_docks’ terciles.
- Create columns for day of week and season based on datetime.

DATA PROCESSING

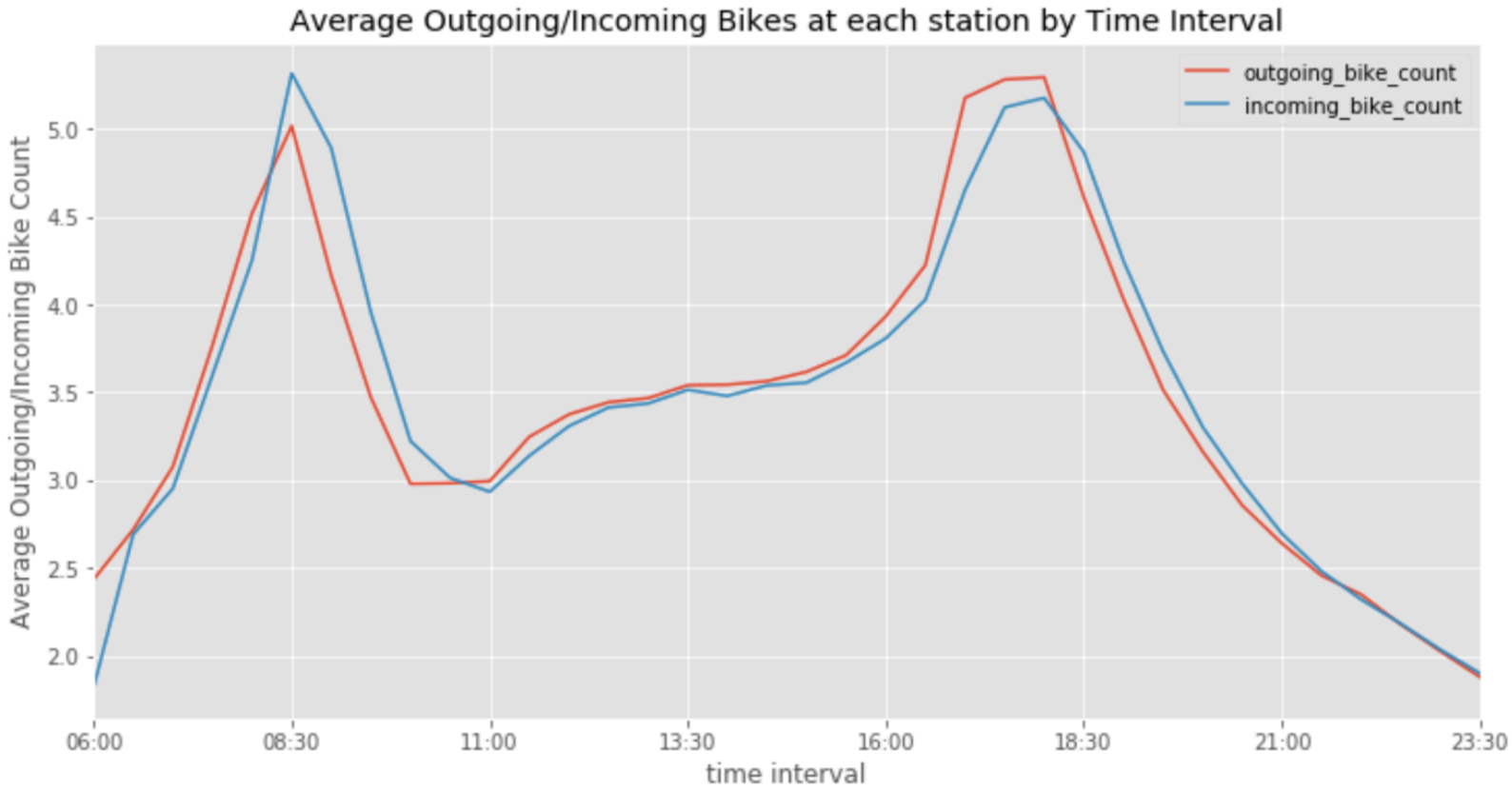
Station Depletion Status

	dock_id	dock_name	date	hour	minute	avail_bikes	avail_docks	tot_docks	_lat	_long	depletion_status	time	dayofweek	season
0	116	W 17 St & 8 Ave	2015-04-02	14	30	5	32	32	40.741776	-74.00149746	Empty Risk	14:30	3	spring
1	116	W 17 St & 8 Ave	2015-04-02	15	00	1	36	36	40.741776	-74.00149746	Empty Risk	15:00	3	spring
2	116	W 17 St & 8 Ave	2015-04-02	15	00	2	35	35	40.741776	-74.00149746	Empty Risk	15:00	3	spring
3	116	W 17 St & 8 Ave	2015-04-02	15	00	3	34	34	40.741776	-74.00149746	Empty Risk	15:00	3	spring
4	116	W 17 St & 8 Ave	2015-04-02	15	30	2	35	35	40.741776	-74.00149746	Empty Risk	15:30	3	spring

DATA EXPLORATION

Bike/Dock Demand

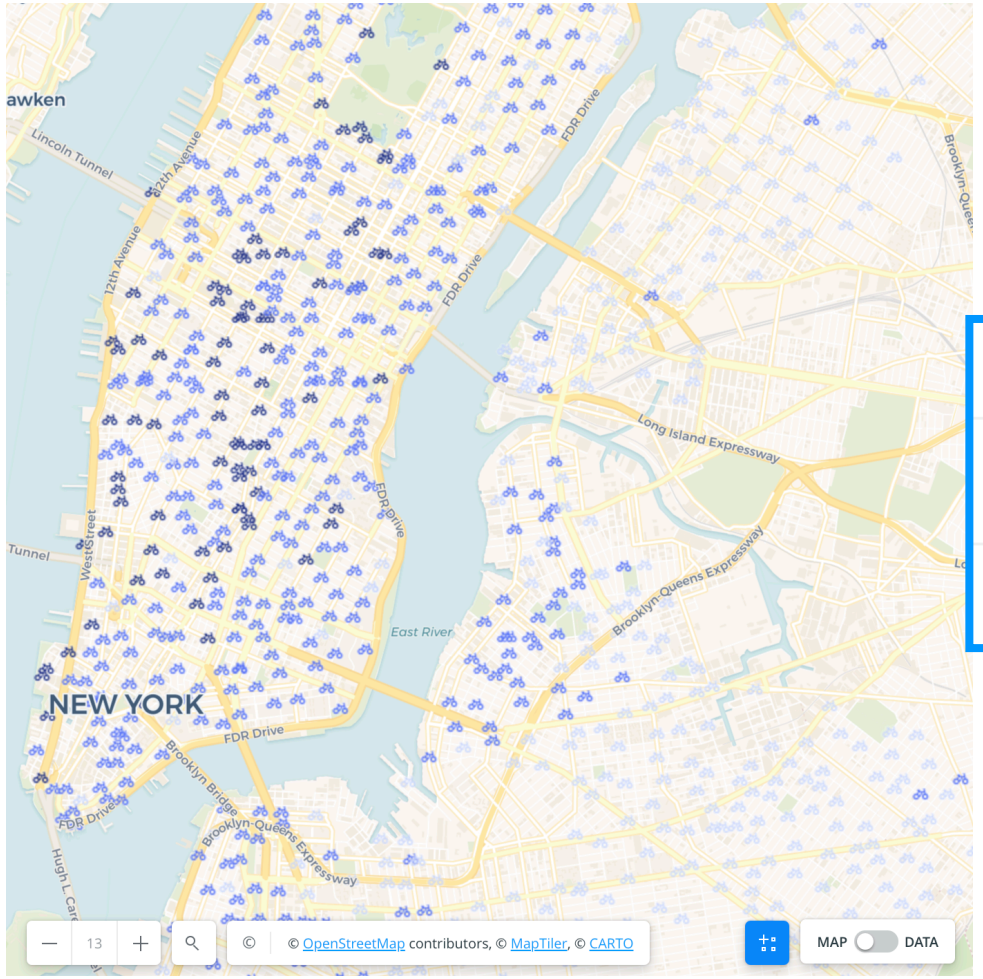
Findings on Bike Demand and Dock Demand



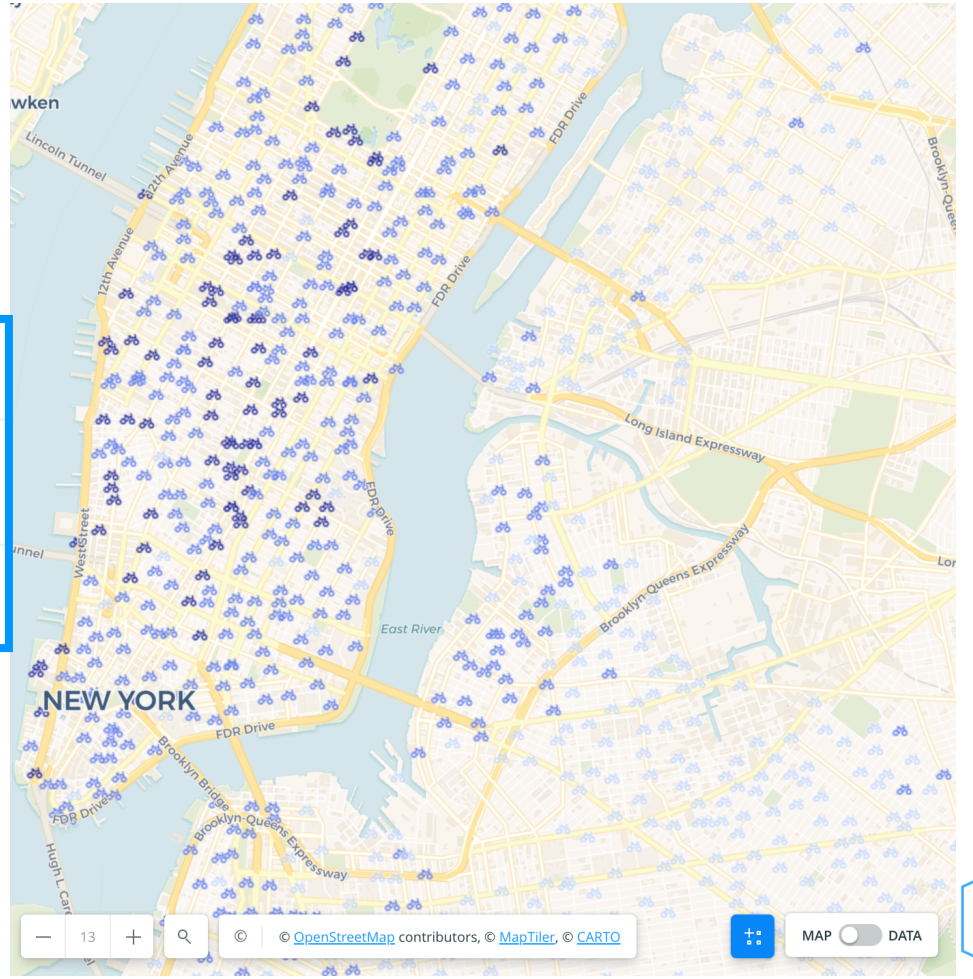
DATA EXPLORATION

Bike/Dock Demand (cont.)

Dock Demand

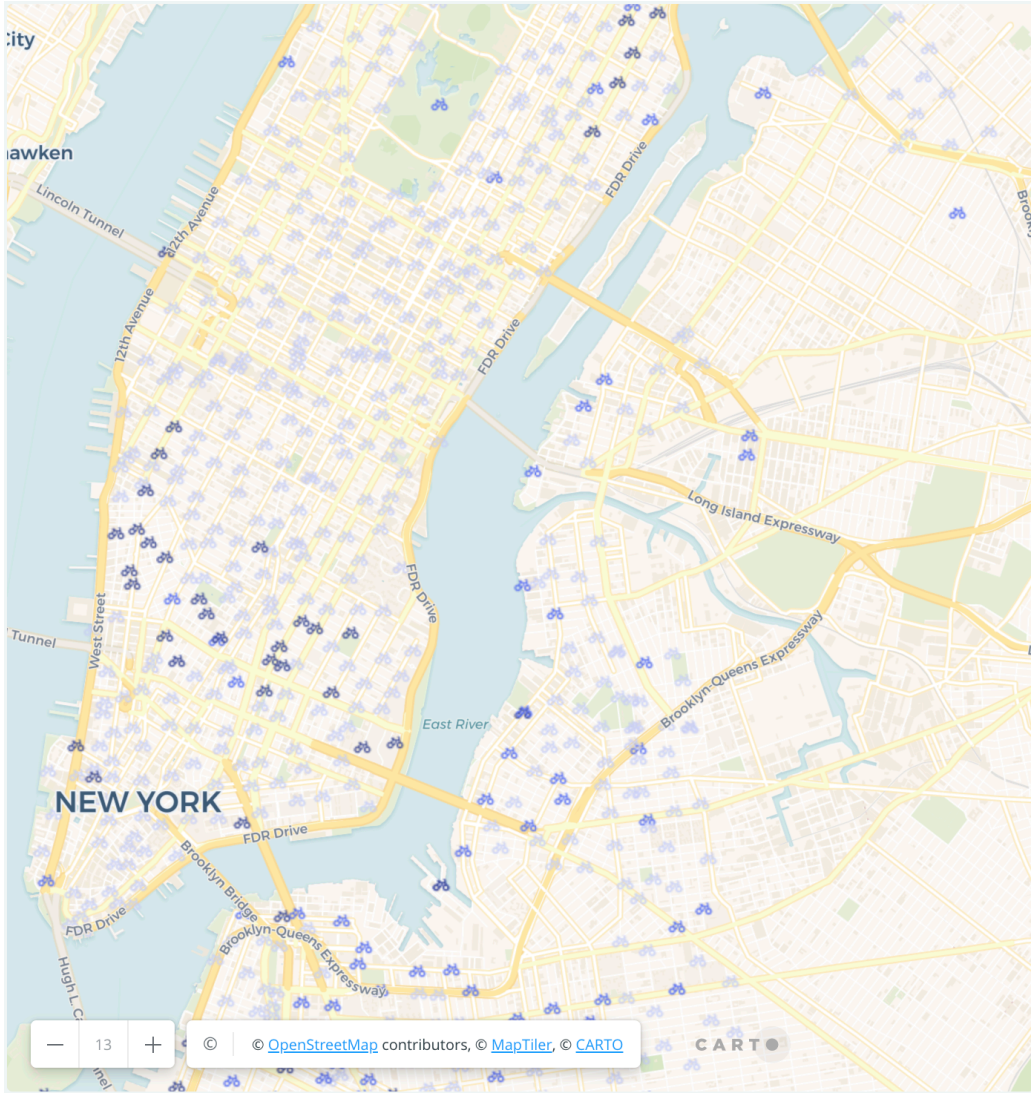


Bike Demand



DATA EXPLORATION

Station Depletion Status

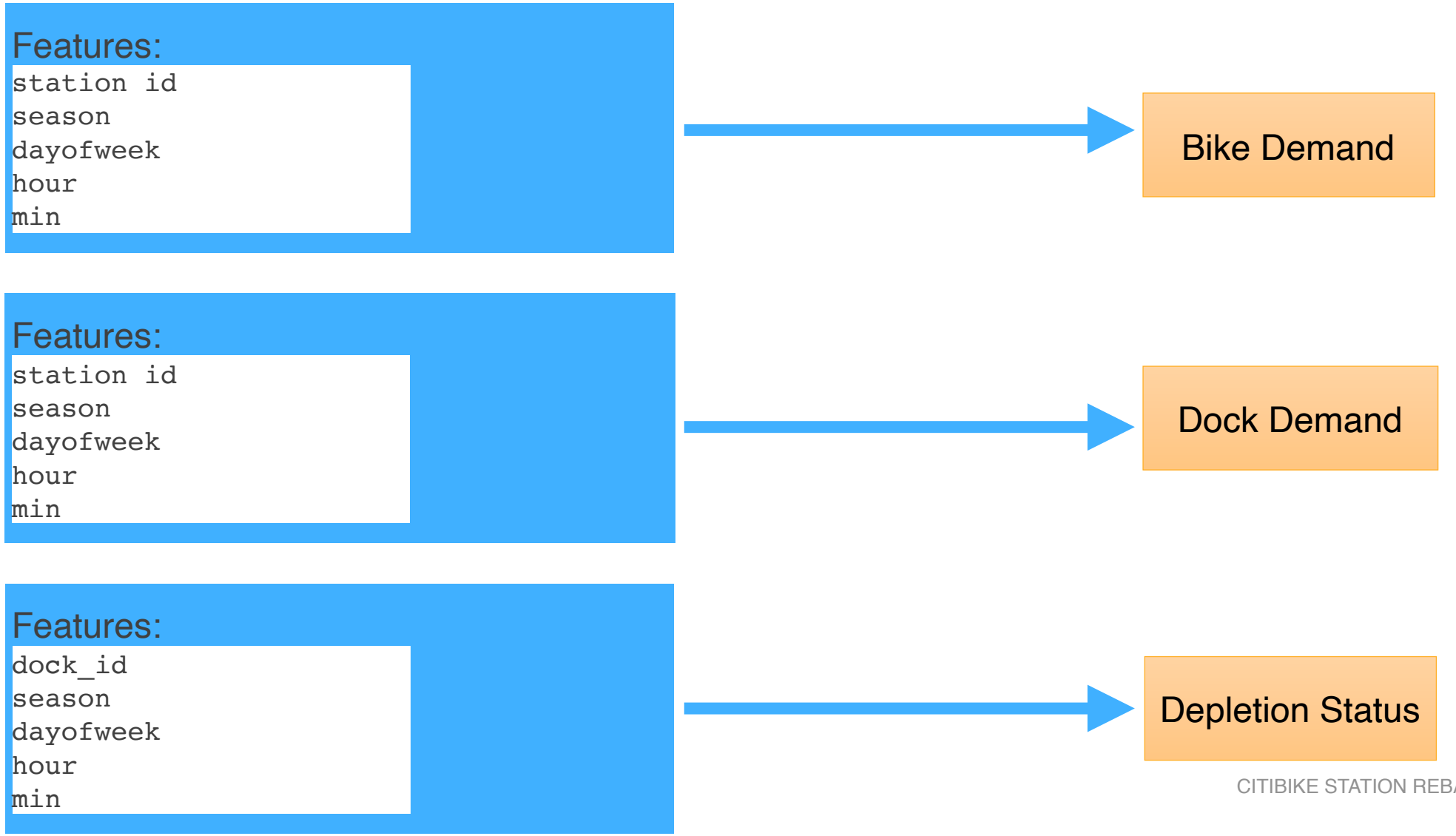


EMPTY RISK

HEALTHY

FULL RISK

MACHINE LEARNING MODELS (RANDOM FOREST)



GENERATE REBALANCING STRATEGY

General Philosophy - Stations to Rebalance Bikes From

Depletion Status	Incoming Bike Slot Demand	Outgoing Bike Demand
Full Risk	High	High
Healthy	Medium	Medium
Empty Risk	Low	Low

GENERATE REBALANCING STRATEGY

General Philosophy - Stations to Rebalance Bikes Into

Depletion Status	Incoming Bike Slot Demand	Outgoing Bike Demand
Full Risk	High	High
Healthy	Medium	Medium
Empty Risk	Low	Low

GENERATE REBALANCING STRATEGY

Steps

- Pick a date and time.
- Get predictions for outgoing bike/incoming slot demand and station depletion status for that time.
- Generate a list of stations to rebalance bikes into and a list of stations to rebalance bikes out of based on the rebalancing philosophy.

GENERATE REBALANCING STRATEGY

Steps (continued)

- For each station that needs bikes rebalanced in:
  - Find the closest geographic station (using Manhattan distance).
  - Pair those stations together.
  - Update the lists of stations in need of rebalancing.

GENERATE REBALANCING STRATEGY

Steps (continued)

- Depending on the predictions, there might still be stations that need bikes rebalanced out after all stations that needed bikes brought in have been rebalanced.
- For each such station that needs bikes rebalanced out:
  - Find the closest station that has a “Medium” demand for both incoming bike slots and outgoing bikes.
  - Rebalance bikes into the closest such station identified.
  - Update the lists of stations in need of rebalancing.

GENERATE REBALANCING STRATEGY

Visually: July 15, 2020 @ 9:00am

	station_id	bike_demand	dock_demand	depletion_status	station_lat	station_long
0	3382	2	2	Full Risk	40.680611	-73.994758
1	362	2	3	Empty Risk	40.751726	-73.987535
2	146	2	3	Empty Risk	40.716250	-74.009106
3	3834	2	1	Full Risk	40.694670	-73.906630
4	500	3	3	Full Risk	40.762288	-73.983362

GENERATE REBALANCING STRATEGY

Visually: July 15, 2020 @ 9:00am

	Rebalance Bikes To Station #	Rebalance Bikes From Station #
0	340	3786
1	3101	3899
2	3383	3539
3	312	3905
4	3452	3093
...	...	...
115	3354	3335
116	244	3303
117	3325	3908
118	3321	3382
119	3110	3373

GENERATE REBALANCING STRATEGY

Visually: September 10, 2020 @ 5:00pm

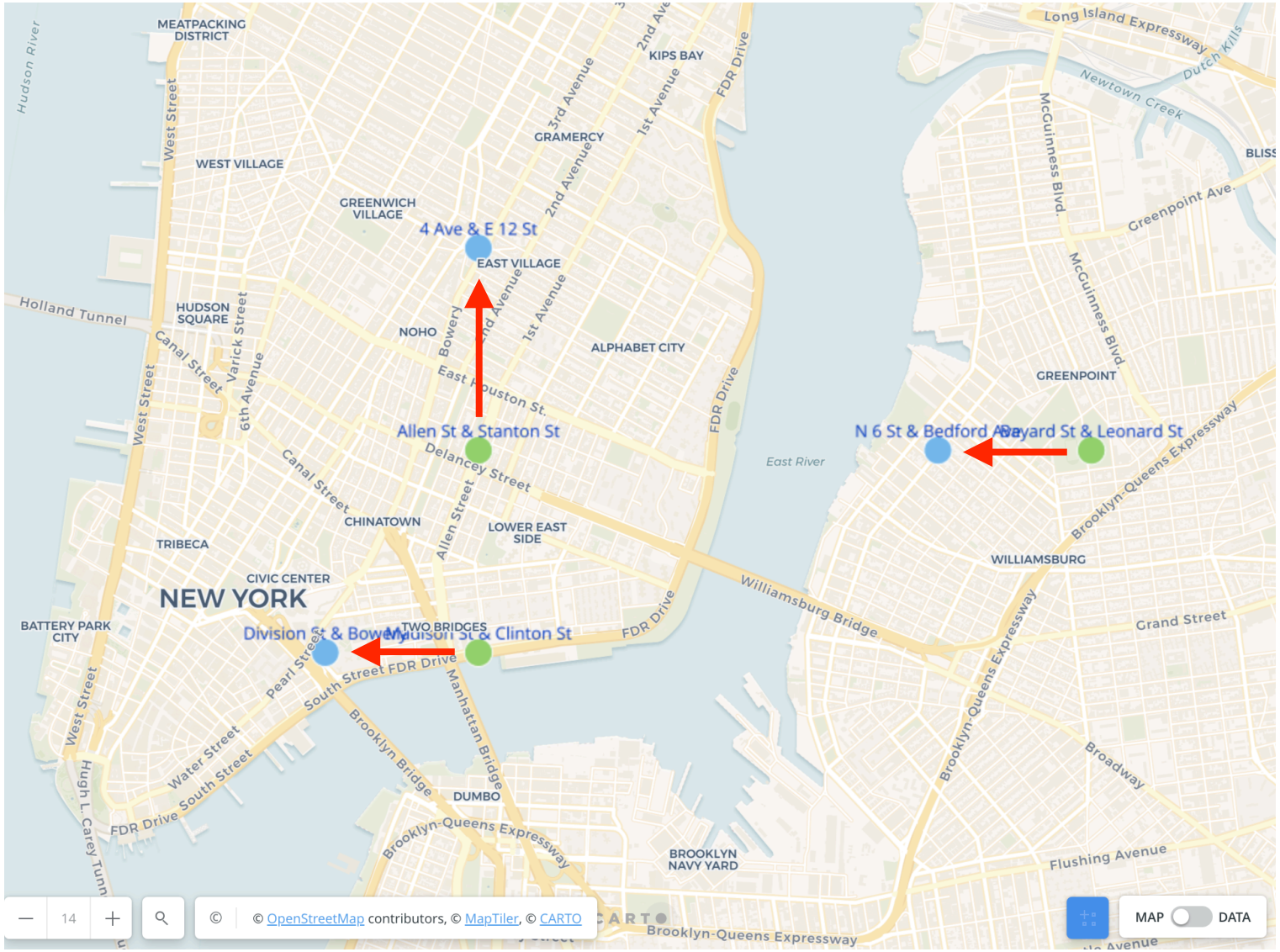
	station_id	bike_demand	dock_demand	depletion_status	station_lat	station_long
0	3382	2	2	Empty Risk	40.680611	-73.994758
1	362	3	2	Empty Risk	40.751726	-73.987535
2	146	2	2	Healthy	40.716250	-74.009106
3	3834	1	2	Healthy	40.694670	-73.906630
4	500	3	2	Empty Risk	40.762288	-73.983362
...	...	...	...	...	...	...
890	3226	2	2	Empty Risk	40.782750	-73.971370
891	3917	2	1	Healthy	40.691780	-73.978770
892	3916	2	1	Healthy	40.708485	-74.002751
893	3918	2	1	Healthy	40.723870	-73.975767
894	2010	3	2	Full Risk	40.721700	-74.002381

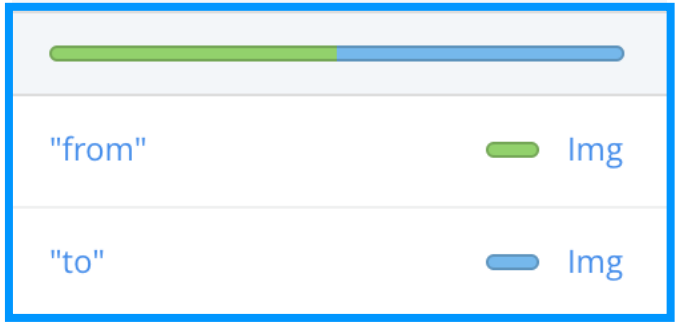
GENERATE REBALANCING STRATEGY

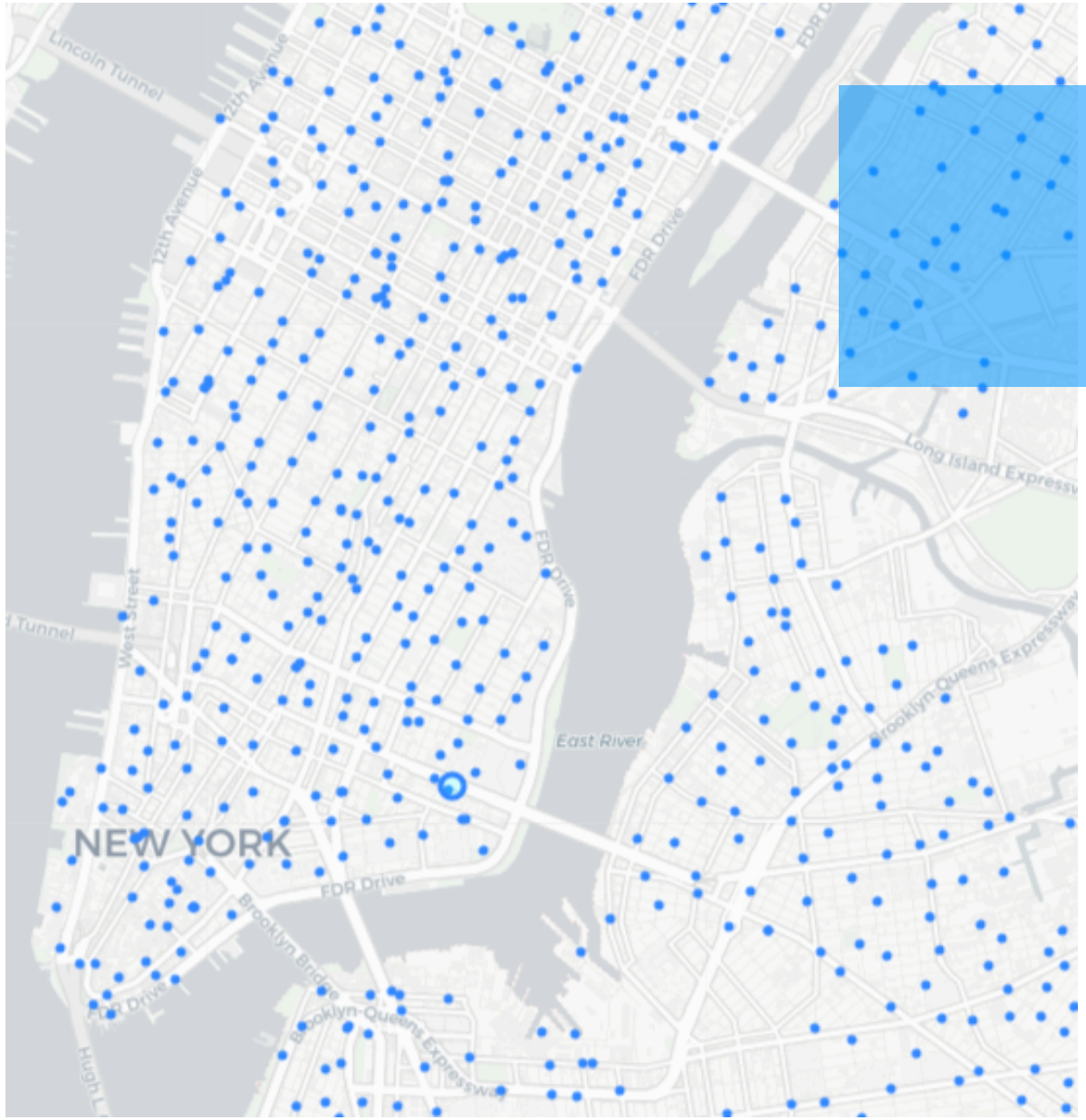
Visually: September 10, 2020 @ 5:00pm

	Rebalance Bikes To Station #	Rebalance Bikes From Station #
0	362	251
1	500	3407
2	469	3112
3	464	3016
4	3143	3092
5	311	276
6	533	248
7	520	239
8	3156	243
9	3443	324
10	3093	3409
11	3463	3335

RECOMMENDATION RESULTS: VISUALLY



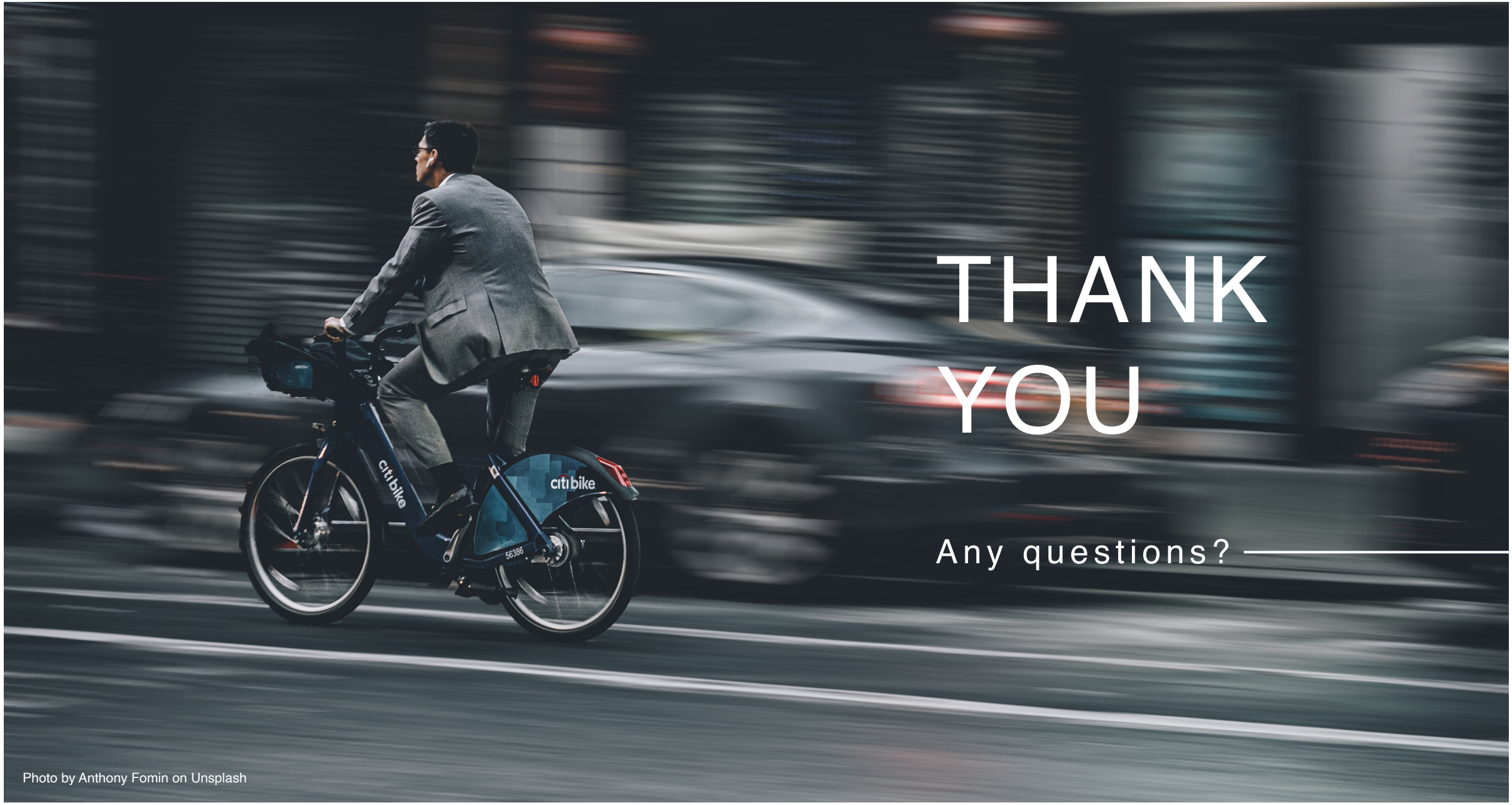




CONCLUSION & FUTURE WORK

FUTURE WORK

- Given the sparsity of the data's feature space makes it difficult to rely on machine learning, we'd like to merge our data with other datasets (particularly **weather** and **traffic** data) to increase the efficacy of our predictions.
- Because of the sheer size of the datasets used to train our machine learning models, we weren't able to test or tune the models as much as we'd like. Given more time, we'd like to:
  - Further tune the Random Forest Models.
  - Test and tune KNN, SVC, and Bayesian Classifiers.
- **Design an app** to prompt the user for a date and time and output a user-friendly visualization of the rebalancing strategy.
- **Account for one-way streets** in the rebalancing strategy (did not have access to Google's map API).
- Identify **exact number** of bikes to rebalance.



THANK YOU

Any questions?

Photo by Anthony Fomin on Unsplash