Predicting Bike Availability in Washington, D.C.

•••

Daniel Nason

Perspectives in Data Science/Professional Skills for Statisticians Joint Project

Agenda

Executive Summary

- *Purpose*: Capital Bikeshare tasked my team (group project) to develop a maintainable model to predict bike availability at their stations for any given time period.
- *Data*: 2019 bikeshare data was cleaned to generate a “bike availability” variable and merged with station capacity information and weather data.
- *Methods*: After exploring the data, we trained the XGBoost machine learning algorithm on the data, tuned hyperparameters and assessed its performance on unseen testing data.
- *Results*: The model accurately predicted outcomes approximately 77% of the time.
- *Maintenance*: We developed a strategy to maintain the model and proposed a user-friendly mobile application to improve operations and enhance consumer experience.
- *Discussion*: Our model has some noteworthy data-related limitations and we identify reasonable extensions that could be applied to improve model performance.

Introduction

- What was the project?
  - Capital Bikeshare tasked us with developing a model that can be utilized to predict bike availability for their stations
  - They would like to deploy this model to improve their business outcomes and generate operational efficiencies
- Key deliverables
  - Build a predictive model
  - Present findings for the model
  - Develop a strategy for model maintenance
  - Discuss the utility of the model

Background

- Bike-sharing platforms provide affordable alternatives to popular existing forms of transportation for short trips
- Our client: third-generation bike-share technology company providing scaled services in limited major metropolitan markets
- Project goal:
  - Improve the user experience by maintaining and increasing bike availability via predictive modeling
  - Lower operational costs and generate revenue sources for the client

The Raw Data

Description: Station Information

Source: Washington D.C.

Key variables:

- Station
- GIS data
- Capacity

Description: 2019 Bike Data

Source: Capital Bikeshare

Key variables:

- Station
- Bike ID
- Timestamp

Description: Weather Data

Source: NCEI (NOAA)

Key variables:

- Temperature
- Precipitation
- Timestamp

Start date	End date	Start station	End station	Bike number
1/30/2019 11:49	1/30/2019 12:00	24th & N St NW	11th & M St NW	W23345
1/30/2019 12:10	1/30/2019 12:18	11th & M St NW	7th & F St NW / National Portrait Gallery	W23345
1/31/2019 6:51	1/31/2019 6:58	New Jersey Ave & N St NW/Dunbar HS	8th & H St NW	W23345

Missing trip information

Preprocessing: Data challenges

Limitations of the data:

- Data exists, but not all in same dataset and not during same timeframe
    - Capital Bikeshare data for 2019 vs. Station information as of June 2021
- Missing data: bikes are randomly relocated between stations (‘reshuffling’)
- Response: No outcome variable for availability exists in any of the data sets
- Format: Data reporting changes from March and May 2020 (Bike ID vs. Ride ID)

Conclusions:

1) By-station availability needs to be developed from raw data
2) Account for reshuffling and missing data in the data set

Preprocessing: Bike Reshuffling

Reshuffling logic:

- Simple example for a bike:
    - For two sequential trips, if:
        - trip 1 is from station A to station B
        - trip 2 is from station C to station D
    - *Then bike has been reshuffled*
- Accounting for reshuffling with example:
    - Two new rows are created:
        - Row 1 (all other data is the same):
            - Start: B → End: Van
        - Row 2 (all other data is the same):
            - Start: Van → End: C
    - Simplifying assumption: bikes immediately transported to next station

Code for reshuffling:

```
r_data = data[~data.Bike_number.isna()].sort_values(['Bike_number', 'Start_date'])
r_data.reset_index(inplace = True, drop = True)
data_shuf_list = []
i = 0
## adding new rows: information is identical except start and end stations
while i < (len(r_data) - 1):
    if r_data.loc[i, "End_station_number"] != r_data.loc[i + 1, "Start_station_number"]:
        if r_data.loc[i, "Bike_number"] == r_data.loc[i + 1, "Bike_number"]:
            # first new row: start station = end station, end station = van
            data_shuf_list.append((r_data.loc[i, "Start_date"], r_data.loc[i, "End_date"],
                r_data.loc[i, "End_station_number"], r_data.loc[i, "End_station"], # start
                "V1", "Van", # end
                r_data.loc[i, "Bike_number"]))
            # second new row: start station = van, end station = start station
            data_shuf_list.append((r_data.loc[i, "Start_date"], r_data.loc[i, "End_date"],
                "V1", "Van", # start
                r_data.loc[i + 1, "Start_station_number"],r_data.loc[i + 1, "End_station"], # end
                r_data.loc[i, "Bike_number"]))
    i += 1 # updating loop
```

Preprocessing: Availability

Broad outline of the process (with 2019 bike data):

1. Group the arrivals and departures at a station by day and hour
2. Using prior data (December 2018), estimate the number of bikes starting at each station
3. Combine data from step 1 and step 2
4. Merge data with station information to obtain station capacity
5. Calculate ***<u>hourly bike availability</u>*** at a station as a ratio of bikes at the station relative to the capacity of the station:

$$Bike\ Availability = \frac{Prior\ hour\ bike\ count + Net\ of\ arrivals\ and\ depatures + Net\ Reshuffling}{Station\ Capacity}$$

Cleaned data and simplifying assumptions

Category	Variable	Description
Response	Availability	Ratio of bikes at station relative to station capacity.
Time	Month	Month when the bike made the trip.
	Hour	Hour when bike made the trip
	Weekday	Day of week when bike made the trip
Location	Station	Name of station where bike made trip.
	Latitude/Longitude	Geographic coordinates of station.
	Region	7 regions: Washington D.C. or the surrounding areas. VA: Alexandra, Arlington, Fairfax, Falls Church MD: Montgomery, Prince George's County
Weather	Temperature (°F)	Daily temperature information including min, max, and average.
	Precipitation	Amount of precipitation (rain and/or snow) in inches.
Other	Holiday	Whether or not it is a federal holiday.

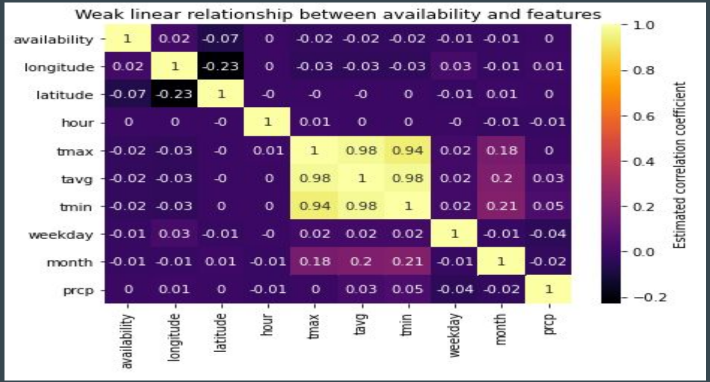
Assumptions:

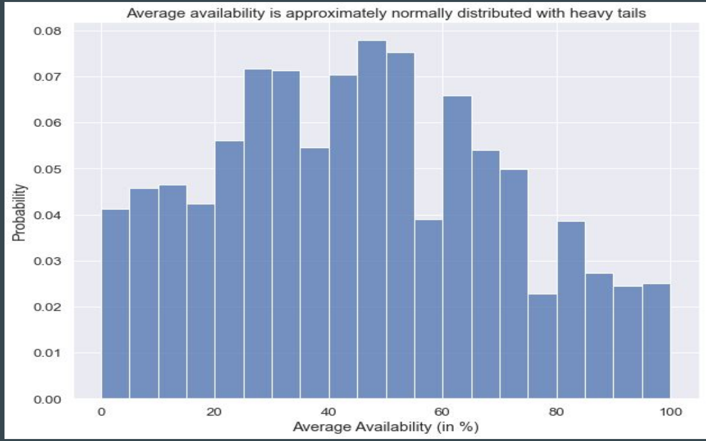
- Bikes not guaranteed to stay in same station as last trip (reshuffling)
- No change in station capacity between 2019 and 2021
- Bikes are not stolen or broken
- Ignore competition from other bikeshare platforms
- Availability is roughly between 0 and 1
    - Availability sometimes exceed capacity (2% of the time)

Data exploration: Describing the data

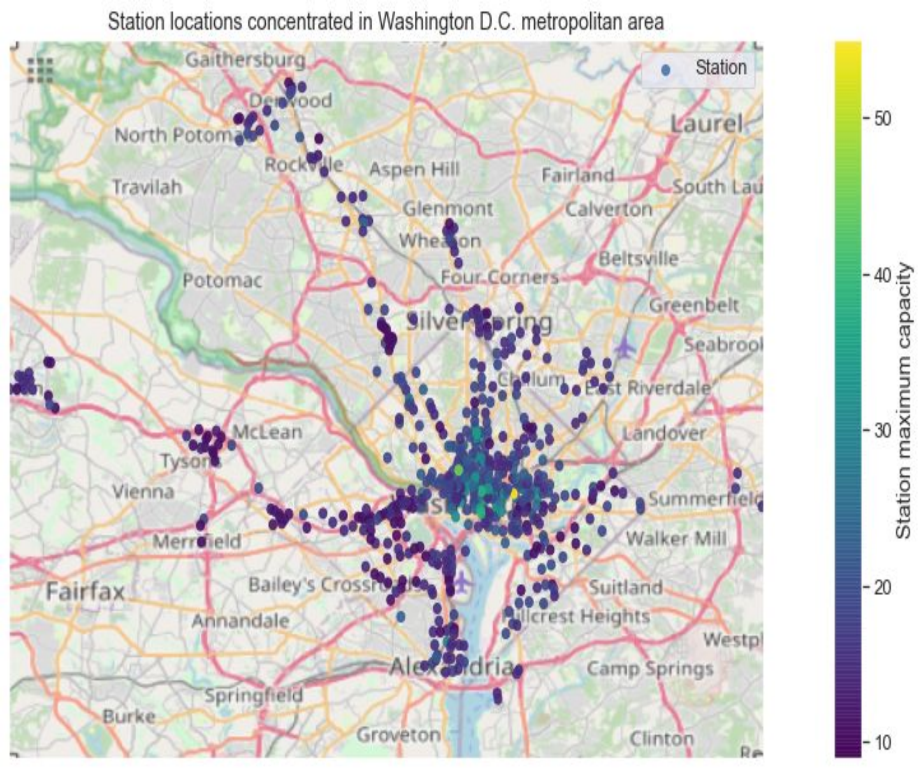
Highlights:

- More than 1.5 million observations in the dataset after cleaning and combining 12 months of data
- Response variable has bimodal distribution
  - Consequence of data cleaning
- Predictors have weak evidence of linear association with response
  - Max absolute value of a predictor's correlation with availability is less than 0.1
  - Non-linear models need to be considered



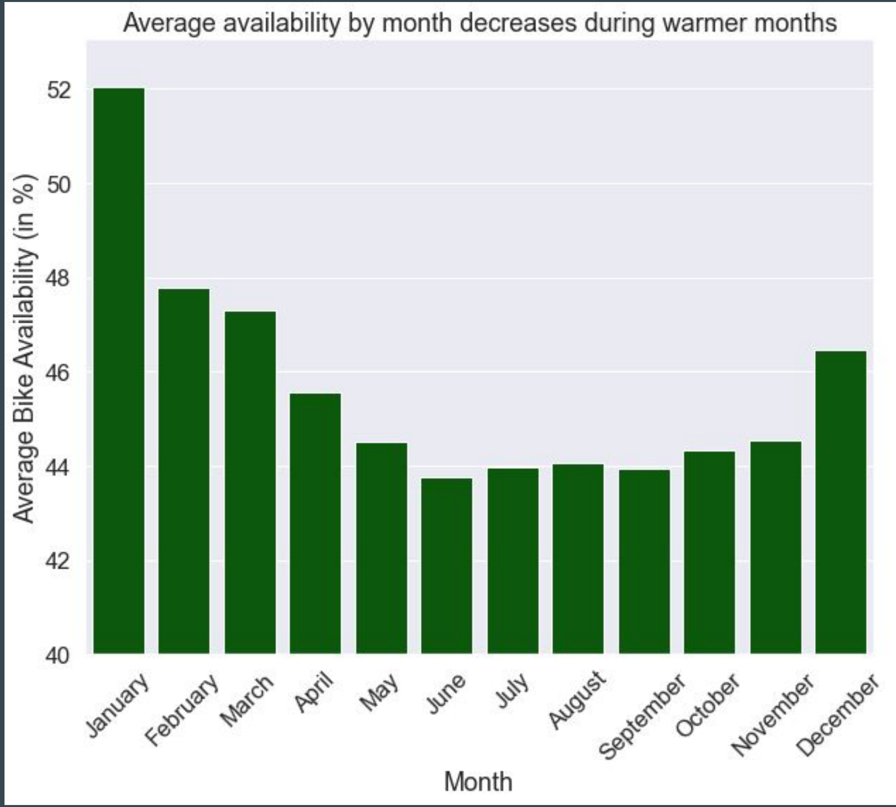


Data exploration: How are the stations distributed?



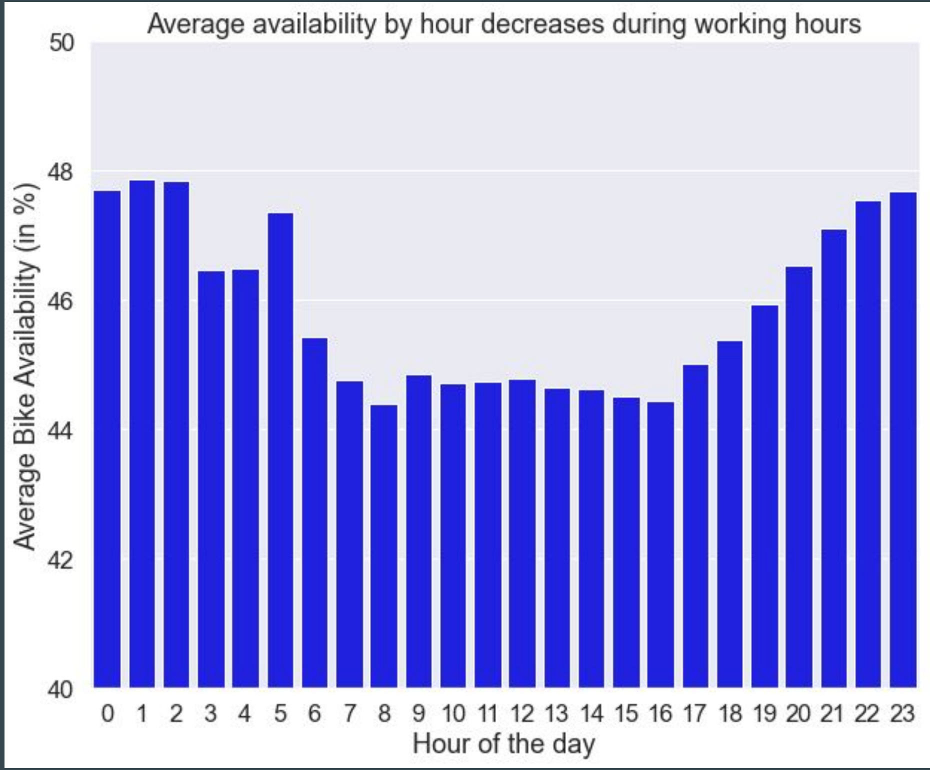
- Majority of stations located in the metropolitan area (roughly 50% of all stations)
- Stations outside the city have comparable maximum capacity
- Capacity more varied in in the city
  - In-city range: 11 to 55
  - Out-of-city range: 9 to 29
- Locations likely due to economic activity and population density

Data exploration: Availability by month



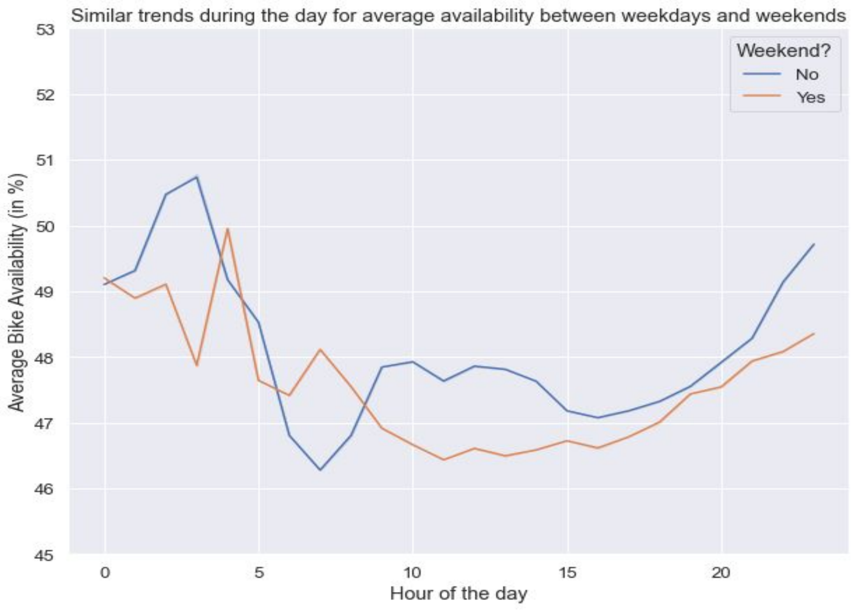
- Total average availability varies across the months
- Relatively higher average availability during cold months of the year
- Possible explanations:
  - Temperature and precipitation
  - Tourism
  - Special events in the city (sporting events, political activity, etc.)

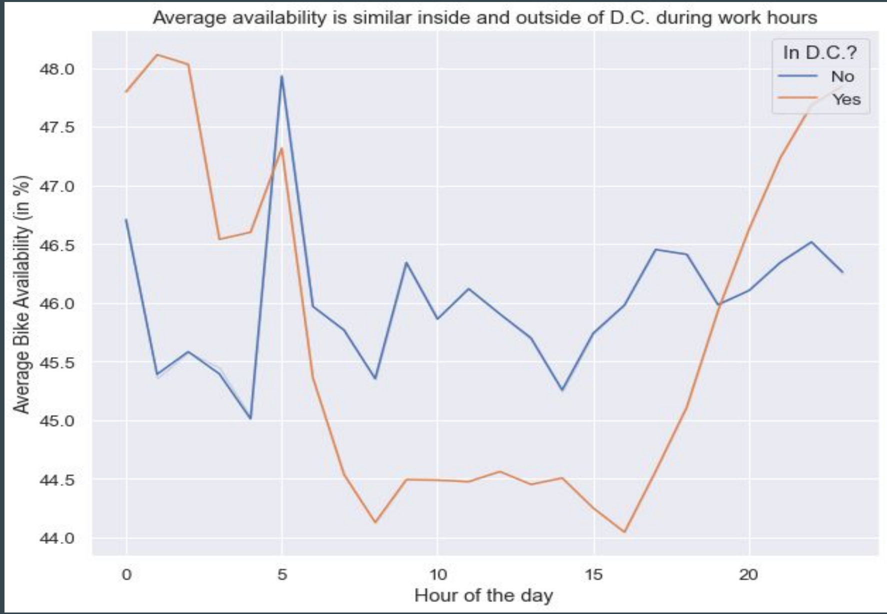
Data exploration: Availability by hour



- Total average availability also varies by hour of the day
- Average availability relatively lower between 6 AM and 10 PM
- Aligns with working hours during the day
    - Lowest availability is at the start of working hours
    - Highest usage as people commute to and from work

Data exploration: Relationship between factors and availability





- No noteworthy dependence present between average availability and factors
- Slight separation during non-work hours depending on location
    - These relationships (or lack of) will need to be considered when developing the model

Modeling approach

Considerations based on EDA and project objective:

- Model needs to be flexible enough to handle non-linear relationships
- Goal is prediction, interpretability of relationships is less important
- Data are moderately sized, so a model is needed that can handle a large number of observations quickly and effectively

Approach: Utilize the XGBoost machine learning algorithm to predict bike availability

Model development

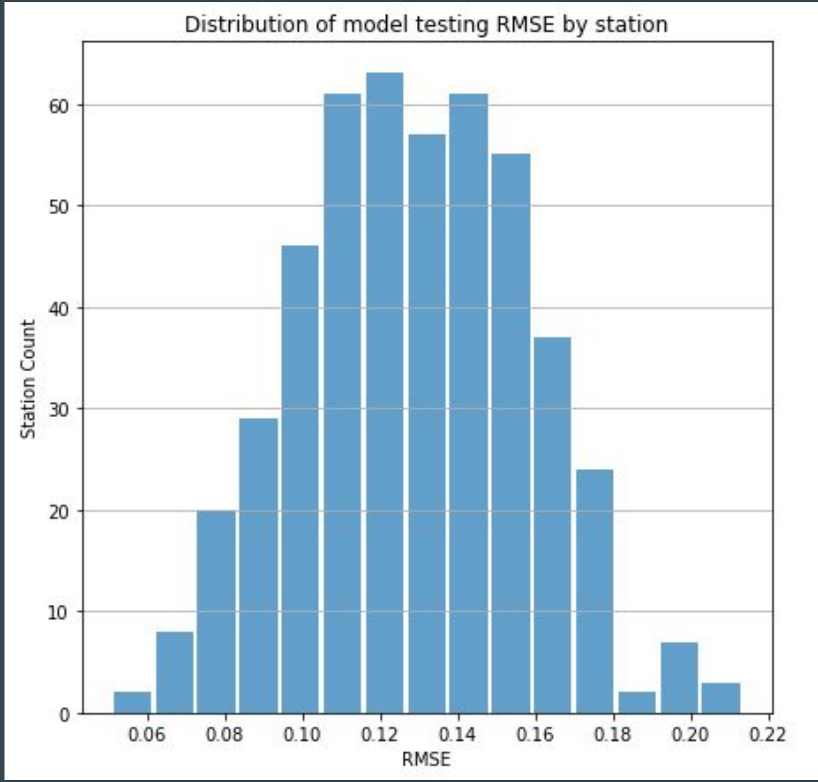
- Feature engineering: adding relevant date/time, location and weather variables
- 80%-20% train-test split applied to the data to validate model performance on unseen data
  - Hyperparameter tuning and cross-validation on training data to maximize performance
  - Fit individual models for each station due to noise in data
- Performance assessment: Accuracy (RMSE) predicting % of bikes available at a given station at a given hour and day
  - Penalizes predictions as they become worse

Assessing model performance

- Average RMSE is roughly 12.86% and is approximately normally distributed
    - Model RMSE is expected to be between 7.06% and 18.66% about 95% of the time
    - Translation: for the median capacity station (15 bikes), the model will be off on average by about 2 bikes

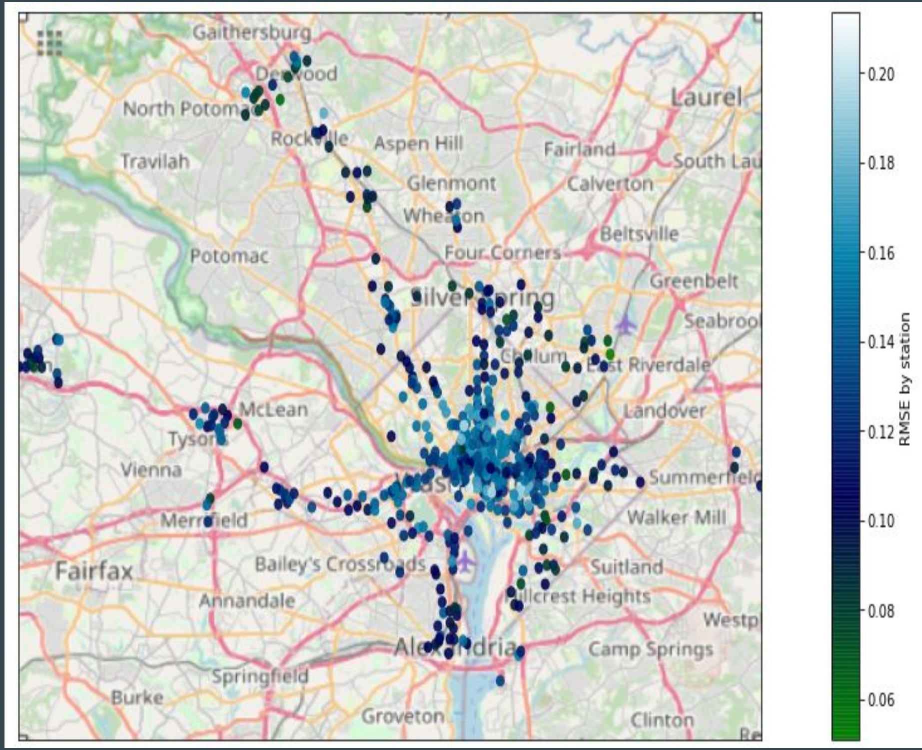
Evaluation:

- Some stations are predicted more accurately than others by the models
- More in-depth investigation needed for why the model has varied performance



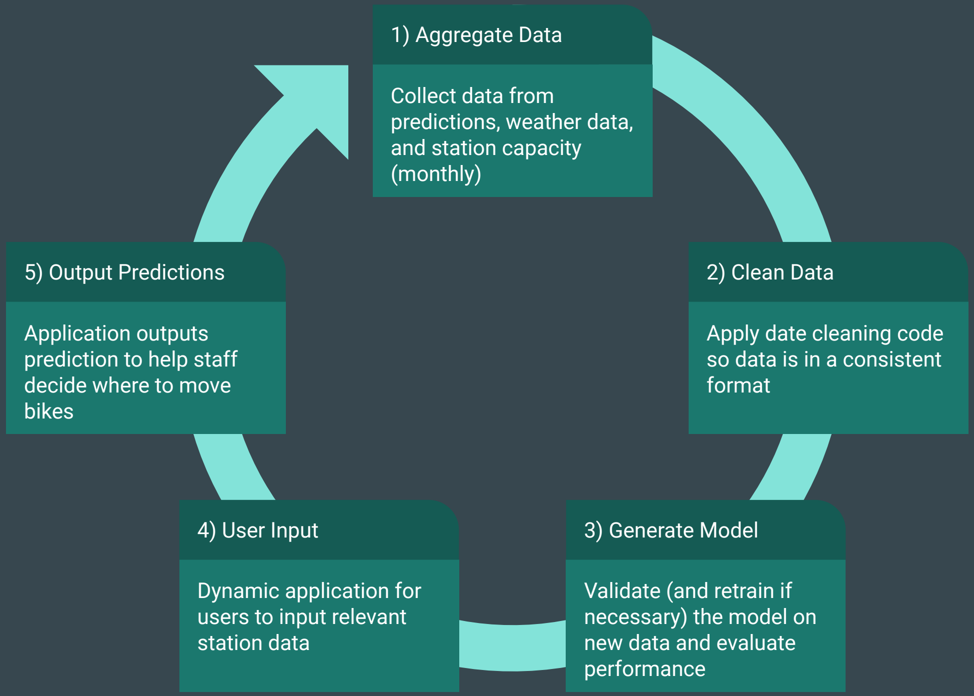
Visualizing the results

- Station location does not appear to associated with model performance
  - Therefore, the model is not biased by station location
- Smaller errors on the outskirts of the station may be due to smaller station capacity
  - Fewer bikes could imply less room for error in predictions



Model maintenance

- Predictions can help to improve business operations by approximating availability by station
- Mobile app development allows workers to make informed decisions on where to reallocate bikes
- Can extend model to consumers to estimate whether bikes will be available at a nearby location when needed
  - Improve convenience and user experience



Discussion: Limitations of the analysis

- Extensive data cleaning, assumptions needed about data validity
- Operational protocols: Capital Bikeshare sets up temporary stations for major events
  - Number of bikes at a station can exceed capacity
- Pre-COVID-19 data may no longer be valid today
  - Data limitations (Bike ID vs. Ride ID) prevent us from tracking a bike's location and therefore updating model for new data

Discussion: Extending the project

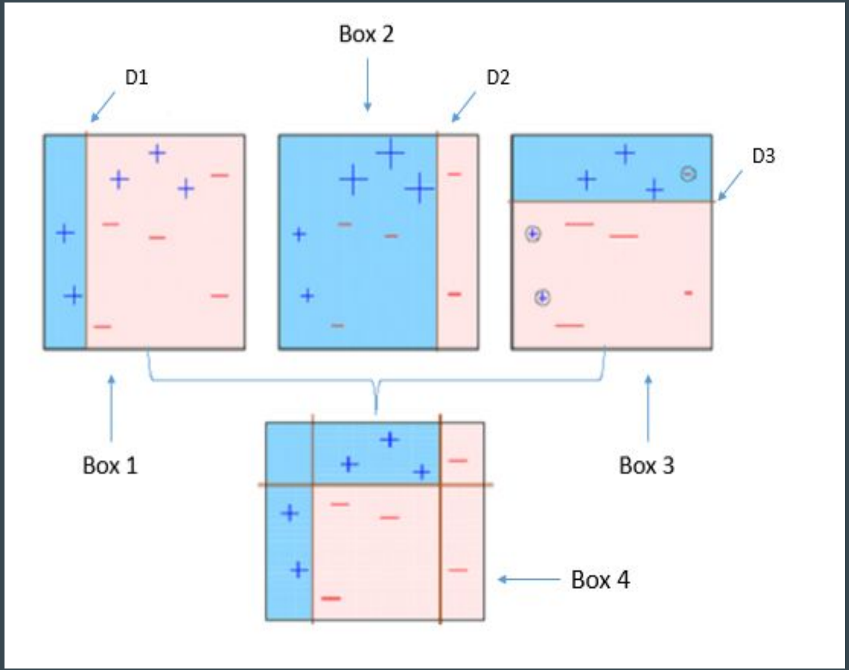
- Data improvements:
  - Reduce to scope subset of key stations
  - Add other relevant predictors such as special local events (sports, political activity, etc.) to predict potential capacity overflow
  - Metro and bus locations, college and university semester schedules
- Modeling:
  - Multi-tier classification problem
  - Regression analysis
  - Time-series modeling
- Compile functions into application to generate model predictions
- Include estimates for demand and collect customer feedback data (surveys, app tracking bike location, etc.)

Thank you! Questions?

Model overview: XGBoost (Extreme Gradient Boosting)

- Gradient boosting: ensemble learning method combines several weak learners (decision tree with max depth of 1) and sequentially corrects the predecessor's errors to form a strong learner
- Advantages:
  - Ensemble learning: avoid overfitting and minimize bias-variance trade off
  - Flexible: learns non-linear relationships and handles collinearity in predictor variables
  - Computation: algorithm optimized to rapidly compute accurate predictions
- Disadvantages:
  - Ensemble learning: loss of interpretability of the relationship between features and outcome
  - Hyperparameters: need to "tune" before applying the model, not an exact science

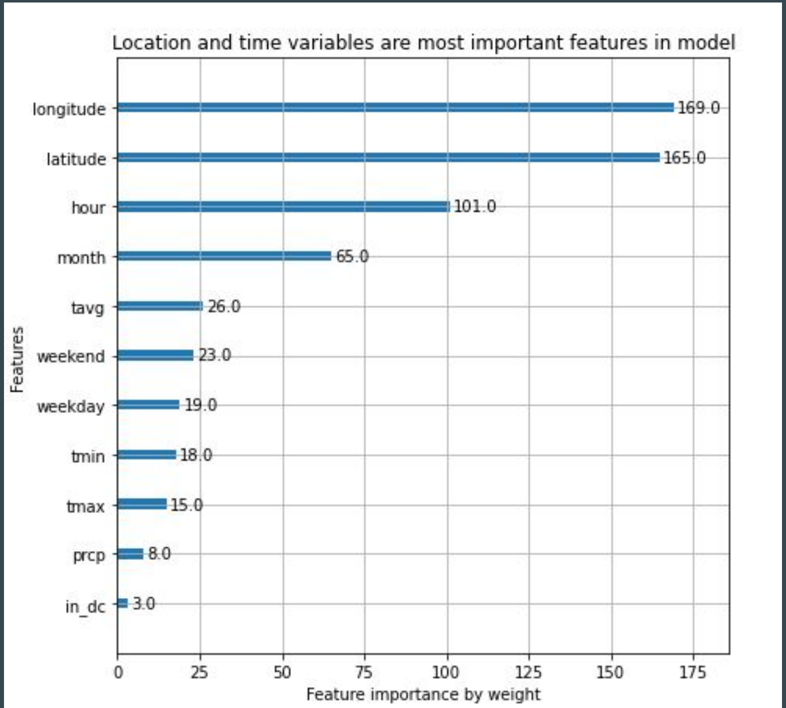
Visual Illustration of XGBoost Algorithm[1]



1 [Link to image]

Feature importances from the models

- Feature importances:
  - The four most important features are related to location or time of the day
  - Weather, time of the week, and whether or not a station is in DC are less important



Packages:

- Pandas, numpy, matplotlib, seaborn
- Scikit_learn: train_test_split, RandomSearchCV
- XGBoost: xgboost, XGBClassifier, cv, plot_importance