**Citibike Case Analysis: Vinit Shah, Analytics Consultant¶**

\*Please look at the Python notebook for the final code. The script was my playground and contains my rough work and testing code.

**Background, Context, and Objective**

**Client:** Mayor of New York City, Bill de Blasio  
**Objective:** Help the mayor get a better understanding of citibike ridership by creating an operating report for 2017.

**Ask:**

1. Top 5 stations with the most starts (showing # of starts)

1. **Pershing Square North**
2. **E 17 St. & Broadway**
3. **Broadway & E 22 St.**
4. **W 21 St. & 6th Ave.**
5. **West St. & Chambers St.**

2. Trip duration by user type





3. Most popular trips based on start station and stop station)

1. **Central Park S & 6 Ave to Central Park S & 6 Ave**
2. **Central Park S & 6 Ave to 5 Av & E 88 St**
3. **E 7 St & Avenue A to Cooper Square & E 7 St**
4. **12 Ave & W 40 St to West St & Chambers St**
5. **Grand Army Plaza & Central Park S to Grand Army Plaza & Central Park S**

4. Rider performance by Gender and Age based on avg trip distance (station to station),

median speed (trip duration / distance traveled)





1. What is the busiest bike in NYC in 2017? How many times was it used? How many minutes was it in use?
   1. **Bike 25738 by minutes and number of starts**
   2. **Used 2,355 times for 31,340 minutes**
2. A model that can predict how long a trip will take given a starting point and destination.
   1. **CV accuracy: 0.746 +/- 0.000**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dep. Variable:** | Minutes | **R-squared:** | 0.746 |
| **Model:** | OLS | **Adj. R-squared:** | 0.746 |
| **Method:** | Least Squares | **F-statistic:** | 2.759e+06 |
| **Date:** | Mon, 30 Apr 2018 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 21:51:11 | **Log-Likelihood:** | -3.5390e+07 |
| **No. Observations:** | 12186496 | **AIC:** | 7.078e+07 |
| **Df Residuals:** | 12186482 | **BIC:** | 7.078e+07 |
| **Df Model:** | 13 |  |  |

First, let's minimize the work and load in the data set. Aggregate all the months and save the data as a csv to not require this to be done in the future.

The column names have spaces in them, would be great to remove them for working purposes. However, not necessary. If I was working on a team or a long term project, I would ocnfigure column names a little bit differently to make them easier to work with.

The dataset is massive, ~16mil rows. BigData tools would be helpful, however, most require you to pay or have an enterprise license or a limited trial. Additionally, the data is very dirty. Different files have different column names, need to account for this. Mayor de Blasio doesn't have a technical background. The graphs here are as simple yet informative as possible. I could've made more complicated plots, however, they would not be as informative for the mayor.

* Lastly, my analysis tries best to follow the crisp-dm methodology outlined below.



**Part 1: Top 5 Stations**

Let's check if there's any noise or cleanup which needs to be done before creating the chart.

1. Any missing values?
   * Mostly for Birth year and a few for User Type. We can ignore these for now and deal with them later.
2. Let's get the data in the right format
   * Trip Duration - Int
   * Start Time - DateTime
   * Stop Time - DateTime
   * Start Station ID - Categorical
   * Start Station Name - Categorical
   * User Type - Categorical
   * Birth Year - Ordinal
   * Gender - Categorical
3. Deal with trips which lasted less than 1.5 minute (90 seconds). If so, in the ideal world, we should not include this start, we may double count. If a bike is broken, a user will dock it again within a minute or two and pick-up another one.
   * Would be ideal to not include any starts where a tip lasted less than 90 seconds *and* the start station = end station.
4. Anomalies such as theft and broken docks shouldn't matter for this metric and can be dealt with later.

**Part 2: Trip Duration by User Type**

This question is a bit unclear in terms of what to do with the anomalies, so I'll be making two graphs. One with anomalies, one without.

*There are NA values in the dataset for usertype as can be seen from missing\_table. Since it's only 0.09% of the data, it's safe to remove.*

**According to Citi Bikes' website:** The first 45 minutes of each ride is included for Annual Members, and the first 30 minutes of each ride is included for Day Pass users. If you want to keep a bike out for longer, it’s only an extra $4 for each additional 15 minutes. It's safe to assume, no one (or very few people) will be willing to rent a bike for more than 2 hours, especially a clunky citibike. If they did, it would cost them an additional $20 assuming they're annual subscribers. It would be more economical for them to buy a bike if they want that workout or use one of the tour bikes in central park if they want to tour and explore the city on a bike. There may be a better way to choose an optimal cutoff, however, time is key in a client project. Or just docing and getting another bike. The real cost of a bike is accrued ~24 hours.

**Anomalies**: Any trip which lasts longer than 2 hours (7,200 seconds) probably indicates a stolen bike, an anomaly, or incorrect docking of the bike. As an avid Citibike user, I know first hand that it doesn't make any sense for one to use a bike for more than one hour! However, I've added a one hour cushion just in case. No rider would plan to go over the maximum 45 minutes allowed. However, I wplan to reduce this to one hour in the future for modelling purposes.

1. *First Half- with anomalies in dataset*
   * The graph under ax2 is a bargraph of average trip duration for each user type. It's helpful, but would be better to see a boxplot and get an idea of the distribution and see mintues instead of seconds.
   * Second graph is a basic Boxplot based with anomalies included. As we can see, there is too much noise for this to be useful. It'll be better to look at this without anomalies.
2. *Second Half - without anomalies in dataset*
   * Still not useful, let's add a column with minutes for trip Duration.
   * Boxplot with minutes is much more useful. There are still some outliers, however, it is informative.

### Part 3: Most Popular Trip

To get most popular trips, the most convenient way to do this is by using the groupby function in pandas. It's analogous to a Pivot table.

The groupby function makes it extremely easy and convenient to identify the most popular trips. Visuals and transformations can be found below.

**Part 4: Rider Performance by Gender and Age**

**Ask:** Rider performance by Gender and Age based on avg trip distance (station to station), median speed (trip duration/distance traveled)

Let's make sure the data we're working with here is clean.

1. Missing Gender and Birth Year values - Check missing\_table above
   * No for Gender. Yes for Birth Year
   * ~10% Missing Birth year. Not a big chunk of data. Can either impute missing values or drop it. Since it's less than 10% of the data, it's safe to assume the rest of the 90% is a representative sample of data and we can replace the birth year with the median, based on gender and Start Station ID. I chose this method because most people the same age live in similar neighborhoods (i.e: young people in east village, older people in Upper West Side, etc.). This will be done after anomalies are removed and speed is calculated.
2. Are there anomalies?
   * For Birth Year, there are some people born prior to 1956. I can believe some 60 year olds can ride a bike and that's a stretch, however, anyone "born" prior to that riding a citibike is an anomaly and false data. There could be a few senior citizens riding a bike, but probably not likely.
   * My approach is to identify the age 2 standard deviations lower than the mean. After calculating this number, mean-2stdev, I removed the tail end of the data, birth year prior to 1956.
3. Caulculate an Age column to make visuals easier to interpret.
4. Calculate trip distance (Miles)
   * No reliable way to calculate bike route since we can't know what route a rider took without GPS data from each bike.
   * Could use Google maps and use lat,long coordinates to find bike route distance. However, this would require more than the daily limit on API calls. Use the geopy.distance packge which uses Vincenty distance uses more accurate ellipsoidal models. This is more accurate than Haversine formula, but doesn't matter much for our purposes.
5. Caulculate Speed (min/mile) and (mile/hr)
   * (min/mile): Can be used like sprint time (how fast does this person run)
   * (mile/hr): Conventional approach. Miles/hour is an easy to understand unit of measure and one most people are used to seeing. So the visual will be created based on this understanding.
6. Dealing with "circular" trips
   * Circular trips are trips which start and end at the same station. The distance for these trips will come out to 0, however, that is not the case. These points will skew the data and visuals. Will be removing them to account for this issue.
   * For the model, this data is also irrelevant. Because if someone is going on a circular trip, the only person who knows how long the trip is going to take is therider themself, assuming they know that. So it's safe to drop this data for the model.
7. Rename Gender Values in Legend from 0,1,2 to Unknown, Male, Female, respectively.
   * The rows where Gender is unknown throws the visual off. There are a few ways to handle this:
     + Remove the missing data. This would not result in a significant loss of information since only 58073 rows have gender as unknown.
       - We can impute missing values, however given the proportion of unknowns the information gain would be negligible.
       - Based on the reasons above, I've decided to remove data with unknown gender. These rows should not have a significant imact on the predictive model later on. However, I will confirm this.
8. Determine Gender and Age performance based on Average Trip distance
   * Similar to graphs for speed. Pretty straightforward.

|  | **Trip Duration** | **Start Station ID** | **Start Station Latitude** | **Start Station Longitude** | **End Station ID** | **End Station Latitude** | **End Station Longitude** | **Bike ID** | **Birth Year** | **Minutes** | **Distance** | **Age** | **min\_mile** | **mile\_hour** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 | 15432349.00 |
| **mean** | 802.73 | 1351.79 | 40.74 | -73.98 | 1343.92 | 40.74 | -73.98 | 23770.68 | 1980.18 | 13.38 | 1.17 | 37.82 | 13.85 | 5.89 |
| **std** | 632.57 | 1345.08 | 0.07 | 0.11 | 1342.96 | 0.05 | 0.07 | 5345.42 | 10.07 | 10.54 | 9.54 | 10.07 | 664.18 | 158.59 |
| **min** | 61.00 | 72.00 | 0.00 | -74.03 | 72.00 | 0.00 | -74.07 | 14529.00 | 1956.00 | 1.02 | 0.00 | 17.00 | 0.00 | 0.00 |
| **25%** | 371.00 | 361.00 | 40.72 | -74.00 | 359.00 | 40.72 | -74.00 | 18667.00 | 1973.00 | 6.18 | 0.55 | 30.00 | 8.70 | 4.55 |
| **50%** | 615.00 | 487.00 | 40.74 | -73.99 | 486.00 | 40.74 | -73.99 | 25481.00 | 1982.00 | 10.25 | 0.90 | 36.00 | 10.47 | 5.73 |
| **75%** | 1058.00 | 3141.00 | 40.76 | -73.98 | 3140.00 | 40.76 | -73.98 | 28124.00 | 1988.00 | 17.63 | 1.48 | 45.00 | 13.17 | 6.90 |
| **max** | 7200.00 | 3654.00 | 40.81 | 0.00 | 3654.00 | 40.81 | 0.00 | 33481.00 | 2001.00 | 120.00 | 5389.14 | 62.00 | 2608057.83 | 255187.93 |

**Observations**

* We still have trips less than 90 seconds, however they seem to be legitimate trips. Checked using the code in cell above.
* We have some Start Coordinates as (0.0,0.0). These are trips which were taken away for repair or for other purposes. These should be dropped. If kept, the distance for these trips is 5,389 miles. For this reason I've dropped any points where the distance is greater than 30 miles. Additionally, we have some missing values. Since it's a tiny portion, let's replace missing values based on Gender and start location. These
* One some trips, the speed of the biker is more than 200 mph. This could be due to the formula used for distance calculation or some other error. The fastest cyclist in the world on a flat surface ever recorded biked at 82mph. It's safe to assume none of the citibike riders can approach this speed. Due to this and the fact that an average cyclist speed is 10mph, I've decided to remove all data where the speed in mph is greater than 20 mph and less than 0.1 mph. ~1.5k data points

**Part 5: Busiest Bike by Times and Minutes Used**

*Ask:*

1. What is the busiest bike in NYC in 2017?
   * Bike 25738
2. How many times was it used?
   * 2355 times
3. How many minutes was it in use?
   * 31,340 Minutes

* Busiest bike and count can be identified by a groupby function
* Function above will also identify the number of times the bike was used
* A similar groupby function which calls for the sum on minutes can identify the number of minutes the bike was used.

**Part 6.1: Predictive Model - Baseline Model**

Ask: Build a model that can predict how long a trip will take given a starting point and destination.

***Assumptions on how the Kiosk will work:*** After speaking to Daniel Yawitz (if you're looking at this, thanks for the clarification), I was told that we should assume that when a user inputs the start and end station, they swipe their key fob (if they're a subscriber) and enter their info on the kiosk (if they're a "Customer") prior to entering the start and end station. This means that we would know their gender and age. Thus these variables can be used in building the model.

Step 1.

* This dataset is massive. Almost 14 million rows. Let's work on a *random* subsample while we build and evaluate models. If I tried to build and evaluate a model on the entire dataset, each run would take me ~10+ minutes depending on the model. One good way to decide what portion of your data to work with is using a learning curve. However, my kernel keeps crashing while trying to create that learning curve. However, given the size of the data and from experience by working with senior data scientists on projects with BAML and other firms I know that I can comfortably work with a few thoushand rows of data given the fact that this is only one year of data. If we were working with data for multiple years, I'd need to reconsider this approach. However, given the reasons above, I've decided to sample 10% of the data. It's stil ~1.3 million rows and should be a representative sample since it's randomly selected. I'll be evaluating my model on df\_sample.
  + I also made the same visuals on the sample to visualize the data.

Step 2.

* Let's get a baseline. If I were to just run a simple multi-variate linear regression, what would my model look like and how accurate would it be? Need to prepare the data for a multivariate regression
  1. Drop irrelevant columns
     + Trip Duration: We have the minutes column, which is the target variable
       - Stop Time: In the real world, we won't have this information when predicting the trip duration.
       - Start Station ID: Start Station Name captures this information
       - Start Station Latitude: Start Station Name captures this information
       - Start Station Longitude: Start Station Name captures this information
       - Start Coordinates: Start Station Name captures this information
       - End Station ID: End Station Name captures this information
     + End Station Latitude: End Station Name captures this information
       - End Station Longitude: End Station Name captures this information
       - End Coordinates: End Station Name captures this information
       - Bike Id: We won't know what bike the user is going to end up using
       - Min\_Mile: Effectively the same information as end time when combined with distance. We won't have this information in the real world.
       - mile\_hour: Effectively the same information as end time when combined with distance. We won't have this information in the real world.
       - (Speed \* Distance = Trip Duration): Which is why speed is dropped
       - Birth Year: Age captures this information
       - Start Station Name and End Station Name: The distance variable captures the same information. For the model, if a user is inputting start and end station, we can build a simple function to calculate the distance which would capture the same information.
  2. Basic cleaning of data FOR NOW.This is only being done for the baseline model
     + Start Time: Requires reformatting. Will do this after baseline model
       - Dumify categorical variables
       - Scale Age
       - Don't scale distance, since it does not just represent distance, but is also indicative of the trip the rider is making (start and station)
  3. Anomalies in Trip Duration
     + I'm going to come back to an observation from earlier. Any trip which lasts longer than *45 minutes(2,700 seconds)* probably indicates a stolen bike, incorrect docking of the bike, or an anomaly. No rider would plan to go over the maximum 45 minutes allowed. Additionally, there are only ~200k rows where the trips last longer than 45 minutes. That is less than 1% of the data and can be considered anomalies. Even if we include it, the model won't be able to learn from it due to the small sample. Lastly, as an avid Citibike user, I know first hand that it doesn't make any sense for one to use a bike for more than 45 minutes! The only way a trip actually lasts more than 45 minutes is if a user couldn't find a dock at their desired station.
* Age was removed after an initial run indicated it had no effect on the model. This was also clearly indicated in some of the visuals above. Reasons for not binning age are in section 6.3.

**Part 6.2: Predictive Model - Including Date**

**Baseline Model:** Adjusted R^2 and R^2 = 56.5% (depending on random\_state)

1. The baseline model is ok, but nothing spectacular. The R\_Squared and Adjusted R\_Squared is pretty much the same, 53.6%. The F-Stat is also 0.00 which is a good sign. The gender female's p-value is high, however, we can't loose this information due to the fact that this is a categorical variable and we'd be loosing substantial information by dropping it.
2. Steps to make improvements: a) Add back time in the following format
   * Is the ride on a WEEKDAY or WEEKEND. Weekday, is rush-hour commute for the most part and probably from home to work. Weekend could be a longer, more casual ride and have higher variability.
   * Is the ride in the MORNING, AFTERNOON, EVENING, or NIGHT. The exact timing will be based on the difference in trip duration based on time of day. Will have visuals below.
   * What season is it?
     + December - Feb. = Winter
     + March - May = Spring
     + June - Aug. = Summer
     + Sept. - Nov. = Fall
3. Evaluate model and check correlation to ensure against collinearity and identify what's going on

**Part 6.3: Predictive Model - Improving Model 1**

Model 1: Negligible improvement in R^2: 67% (depending on random\_state used)

* Date seems to not have as big of an impact. May be more useful to split time and day differently, but acording to the graphs, there is no indication of a better split. Next steps will be to factor in speed and distance based on Gender and Trip. By not being able to encode start and end stations (due to the sheer number of points), we are losing crucial information on the trip itself. We need another proxy for those measures.
* Another change I could make is to bin age into buckets. However, the data indicates that age has no correlation or effect on the trip duration. This is counter-intuitive, however, I don't have a good reason to refute the data.

Next Steps:

1. Include Average Speed based on: Trip and User Type
   * Reason for Trip:
     + Some trips are up hill, others are down hill. Some routes, such as one through times square involve heavy trafic, based on intuition.
     + Reason for User Type:
       - Tourists (Customers), will usually ride more slowly with frequent stops than a Subscriber, according to the data.
2. Include average duration for each trip based on: Trip and User Type
   * Reason, for each, same as above.
     + Using seconds instead of minutes for granularity in information.
3. A bit of inexperience here:
   * The imputed columns are "dependent" variables in the since speed is derived from trip duration and distance.
     + There may be a better way to do this, however, I would need to consult someone. Would appreciate feedback on this.
     + I don't think it should be a big issue that they're "dependent" since they serve more as anchor points. But I'd appreciate feeback.

**Part 6.4: Predictive Model - Improving Model 2**

Model 2 is VERY VERY good. The model above is extremely good. But can I make it even better?

* One of the grading requirements for the project is to include an additional data source. Weather and traffic make the most sense. I could include pedestrian traffic data from citibike's website, however, I don't think it'll be too helpful due to the fact that there are now bikelanes in NYC. I could use weather data as well.
  + Concerns with using weather data:
    - Weather dictates whether or not a rider will bike, not how long they will bike. For weather to determine how long one will bike, it would have to dictate where they go. However, since grading deends on my ability to use external data, I'm going to test this hypothesis. If weather is not a strong indicatior, I will remove it in the next model.

**Part 6.5: Predictive Model - Final Model**

* As I thought, weather is not a major contributor to the model. Thus, I'll be taking out weather. Additionally, I'll be taking out date attributes as mentioned earlier.
  1. The ensemble algorithm, Random Forest takes much longer to run (even with a low n\_estimators). If the run time is 5 minutes on ~1.5 million rows, I don't want to take the risk on ~15 million rows of data.
  2. I could've used XGboost and other fancy algorithms, however, for a dataset of this size, it would take too long to run and the gains wouldn't be worth it if there are any.
  3. Final Model:
     + Predictors: Distance, Gender, Average Duration based on Trip and Gender, User Type