Citi Bike Investigation

Michael Harder
Brown University - Data Science Initiative
December 4, 2019
https://github.com/Michael-Harder/NYC_Citi_Bike_-Analysis

Agenda

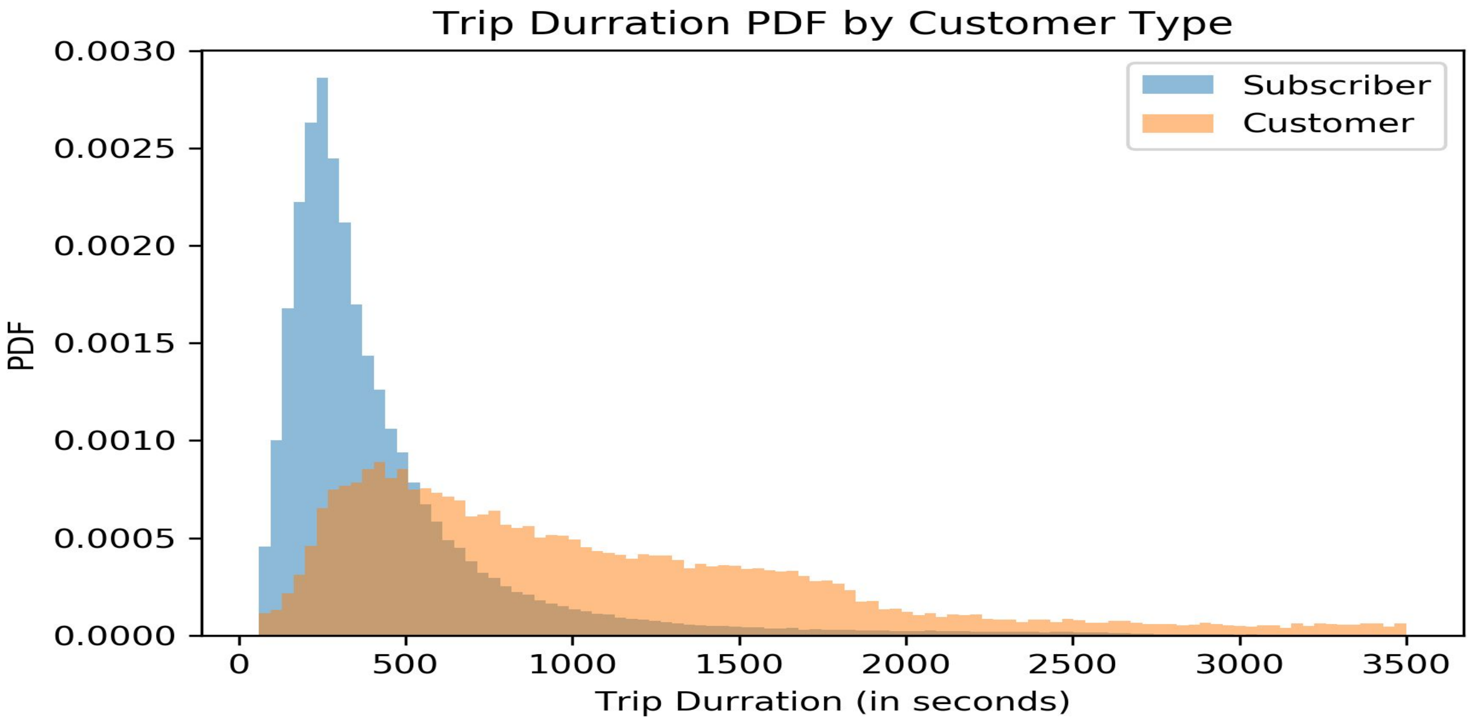
- Introduction
- Cross Validation
- Results
- Outlook
- Questions
- Appendix

Introduction

Citi Bike	• **New York City public bike share program** launched in an effort to not only reduce traffic, carbon emissions, and roadwear, but also improve public health • Operational since 2013 • Via the **NYC Open Data** initiative the city has publicly published various data sets including **Citi Bike trips data from 2013 to present** ○ Data for this project was collected via www.citibikenyc.com/system-data
Problem	• Like any customer based business model, Citi Bike can benefit from understanding more about their **customers' behavior** • Citi Bike trips data can illuminate how **annual subscription riders differ from 24-hour or 3-day pass riders** • Machine learning allows us to **classify trips** - allowing us to predict if a **trip was conducted by a subscription rider or an everyday customer** - providing an interesting lense into how their behaviors differ

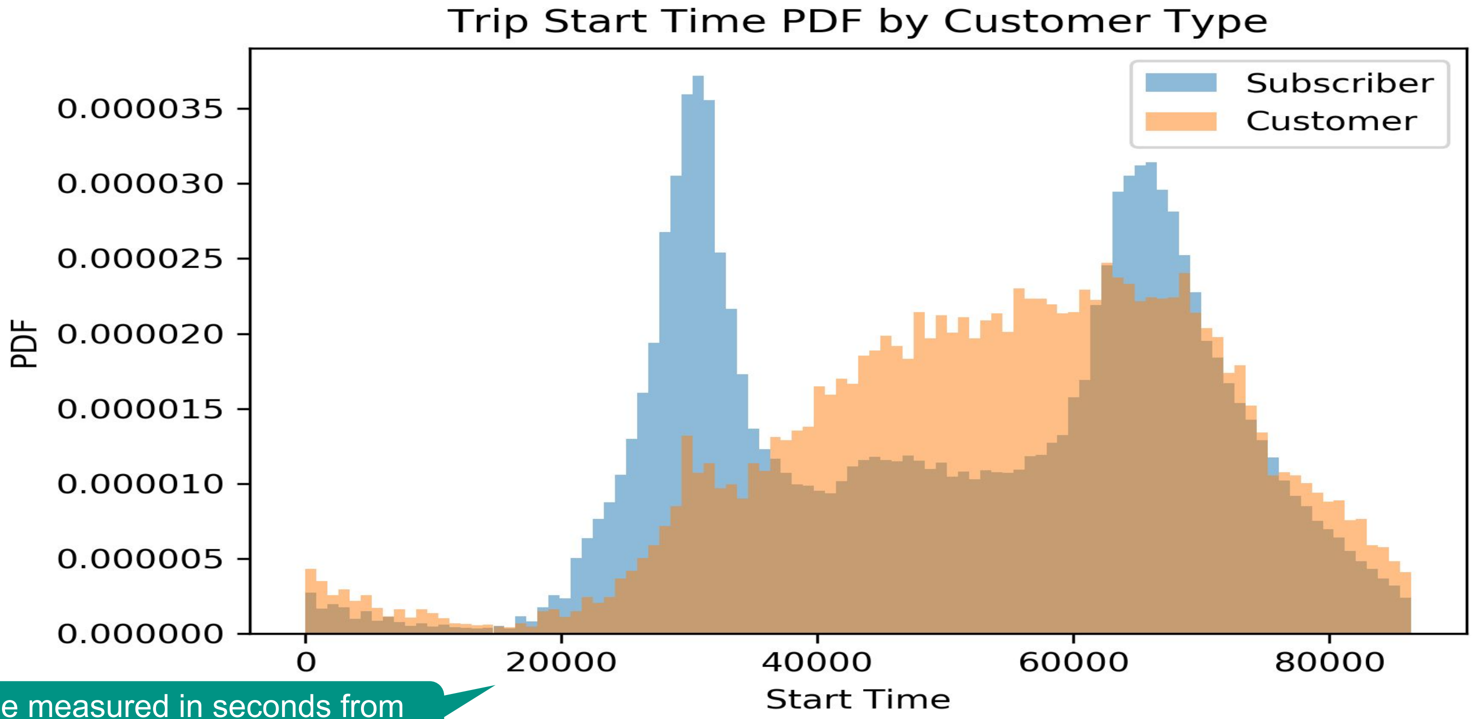
EDA

Subscribers tend to take shorter trips



EDA

Subscribers' start times have two peaks over the course of the day - this could represent commuter behavior



Start time measured in seconds from the start of each day (midnight)

Agenda

- Introduction
- Cross Validation
- Results
- Outlook
- Questions
- Appendix

Cross Validation

Data Splitting	● 80/20 train test split ● **Stratified K-fold** ○ Used because of unbalanced data set (92% subscribers)
CV Pipeline	● The pipeline applied across ML classification methods tested: ○ Split data ○ Preprocess: standard scalar and one-hot encode ○ Apply the appropriate ML algorithm ○ Set up our parameters ○ Prepare a gridsearch ○ Apply k-fold cross validation

Cross Validation

Models Tested	Parameters Tuned
Logistic Regression	• C = [0.1, 1.0, 10, 100] • Lasso regularization*
SVC	• C = [1.e-03 - 1.e+04] • Gamma = [1.e-03 - 1.e+04] • RBF Kernel*
Random Forest	• Min Splits = range(2, 25, 5) • Max Depth = range(1, 30, 5)

** Other regulators/kernels were not tested*

Agenda

- Introduction
- Cross Validation
- Results
- Outlook
- Questions
- Appendix

Results

All three models performed similarly

Random Forest

- Original Data Frame: test accuracy = **0.9515** +/- .0031 -(base = 92%)
- Balanced Data Frame: test accuracy = **0.8262** +/- 0.0131 -(base = 50%)

Logistic Regression

- Original Data Frame: test accuracy = **0.9438** +/- 0.0025 -(base = 92%)
- Balanced Data Frame: test accuracy = **0.8240** +/- 0.0181 -(base = 50%)

SVC

- Original Data Frame: test accuracy = **0.9412** +/- 0.0025 -(base = 92%)
- Balanced Data Frame: test accuracy = **0.83** +/- 0.02 -(base = 50%)

Results
First cut - SVC

Random Forest

- Original Data Frame: test accuracy = **0.9515** +/- .0031
- Balanced Data Frame: test accuracy = **0.8262** +/- 0.0131

Logistic Regression

- Original Data Frame: test accuracy = **0.9438** +/- 0.0025
- Balanced Data Frame: test accuracy = **0.8240** +/- 0.0181

SVC

- Original Data Frame: test accuracy = 0.9412 +/- 0.0025
- Balanced Data Frame: test accuracy = 0.83 +/- 0.02

- *SVC was not decerinably better from the other models*
- *Computing power limitations; limited to 5 random seeds and reduced data frame to a 1% random sample without replacement from the original data frame for 22,794 observations*

Results

Final choice - random forest

Random Forest

- Original Data Frame: test accuracy = 0.9515 +/- .0031
- Balanced Data Frame: test accuracy = 0.8262 +/- 0.0131

- Original RF_score = 8.8834
- Balanced RF_score = 24.8261

Logistic Regression

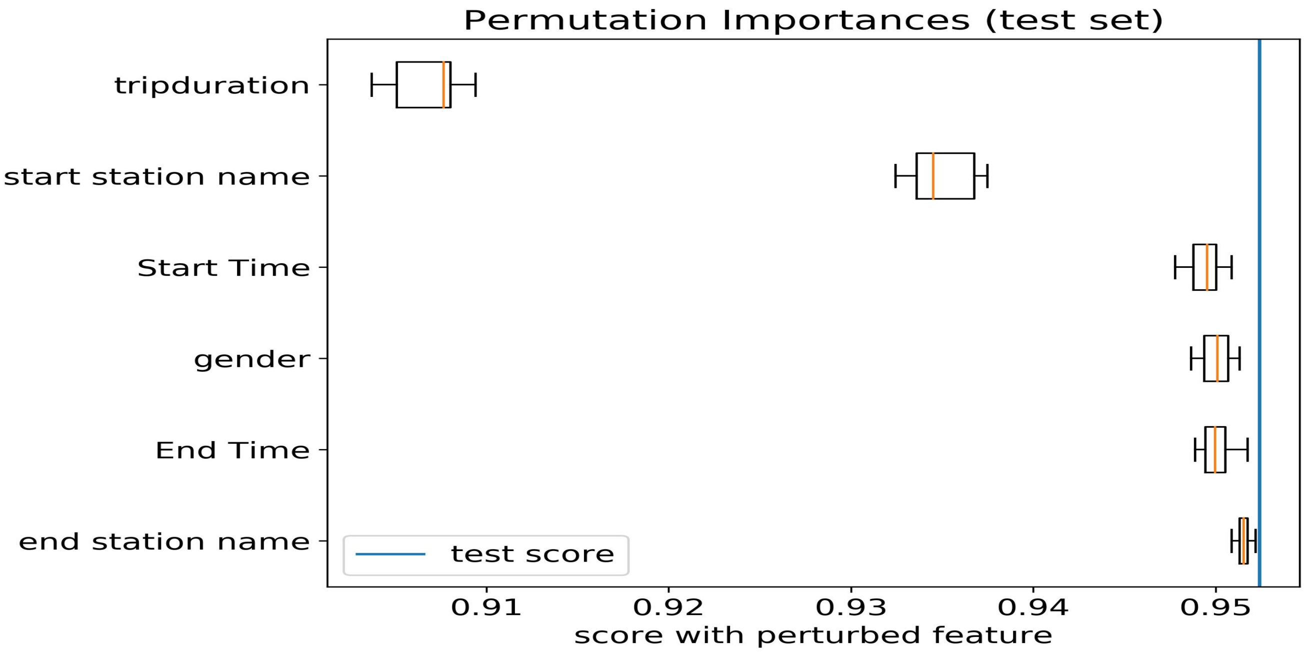
- Original Data Frame: test accuracy = 0.9438 +/- 0.0025
- Balanced Data Frame: test accuracy = 0.8240 +/- 0.0181

- Original LR_score = 8.2084
- Balanced LR_score = 17.9212

- *Scores calculated by: (average model test accuracy - base accuracy) / standard error*
  - *Larger scores are preferible*
- ***Random Forest out performed logistic regression** on both the original data frame and the balanced data frame*

Results

Permutation Feature Importance - trip duration and start station



- *Trip duration: the two user types use Citi Bike differently. Subscribers appear to use the bikes more out of utility*
- *Start Station: subscribers start their trips at certain stations. Could help identify growth strategy.*

Agenda

- Introduction
- Cross Validation
- Results
- Outlook
- Questions
- Appendix

Outlook
Improvements to be made

Missing Values:

- Could not properly apply the **MCAR test** for my missing data
- There could be a more rigorous way to handle the feature with missing values (birth year)
- Tried to leave the missing values in and treat them as **another category in the one-hot-encoder**; however, I could not get it to run with the missing values
- Apply **XGBoost**

Computing Power:

- **SVC** could have been **tested and properly compared** to the other models
- allowed for a **larger random sample** to be used with replacement from the original dataset
- The data frame tested for random forest and logistic regression contained 22,794 observations (just 3% of my data set). The balanced data frame only contained 1,746 observations.

Parameter Tuning:

- Try **different Kernels** for SVC
- Try **different normalizers** for Logistic regression (like ridge)

Agenda

- Introduction
- Cross Validation
- Results
- Outlook
- Questions
- Appendix

Agenda

- Introduction
- Cross Validation
- Results
- Outlook
- Questions
- Appendix

Preprocessing – initial investigating

- Dataset:
  - Limited data to trips from August 2017 to August 2019
  - 759,807 rows of trips data by 12 columns
    - Feature columns included - start time, end time, trip duration, start station name, end station name, start station longitude, start station latitude, end station longitude, end station latitude, user type, birth year, gender

- Initial Cleaning:

  - Dropped the following columns
    - Start Station ID - data set includes start station name. ID used for internal purposes
    - End Station ID - data set includes end station name. ID used for internal purposes
    - Bike ID - ID number used for internal purposes

  - Start time and end time:
    - Provided as strings in format “yyyy-mm-dd hh:mm:ss.ssss”
    - Trimmed this string to get the time and converted it to seconds from start of day so it could be preprocessed with standard scaler as a float64

Preprocessing – encoding

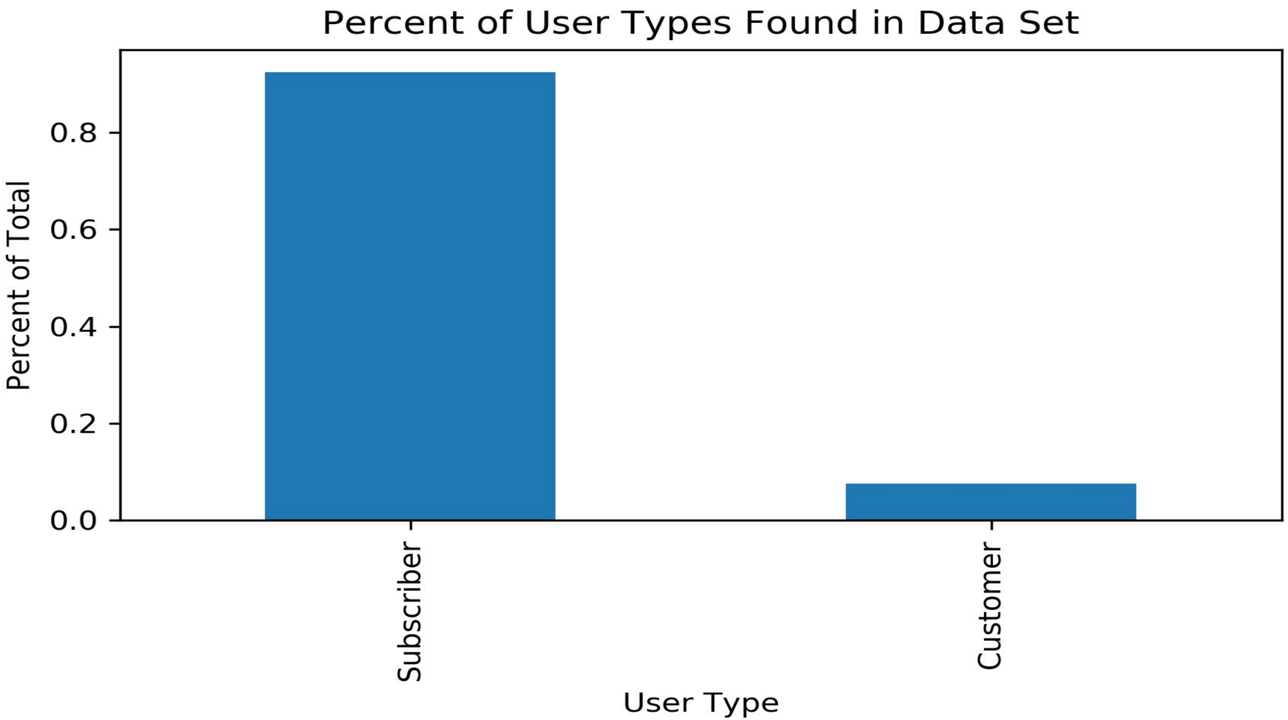
One-Hot Encode	• Applied to **categorical** variables: ○ Start station name ○ End station name ○ Gender
Standard Scaler	• Applied to **continuous** variables: ○ Trip duration ○ Start station longitude ○ End station longitude ○ Start station latitude ○ End station latitude ○ Start time ○ End time ○ Birth year
Label Encoder	• Applied to the **categorical target variable:** ○ User type

Preprocessing – missing values

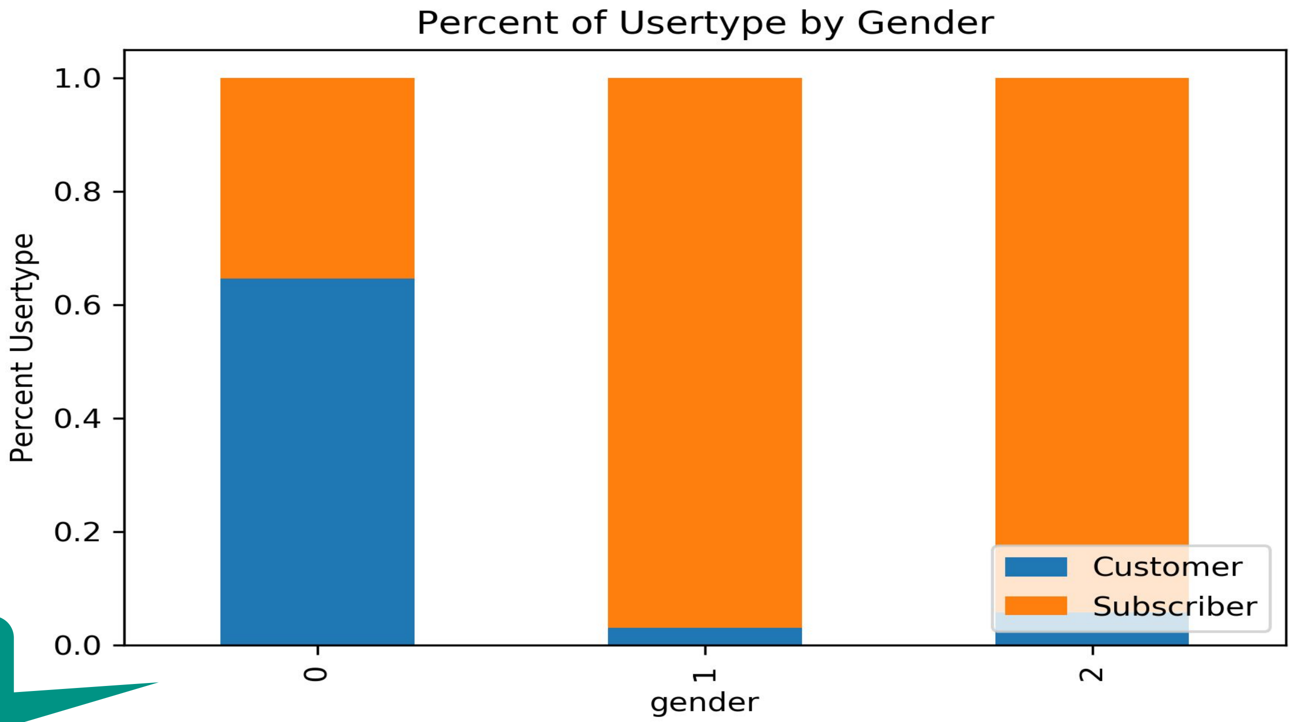
- Before standard scalar was applied there were missing values to consider:
  - 1.12% of rows contained missing data
  - The only feature containing missing data was Birth Year
- MCAR test was applied to investigate the MCAR p value
  - Received error Andras “has never seen before”
- Considering this small percentage of points with NaNs, the small fraction of NaNs in each feature, and the difficulties with the MCAR test I dropped the rows with missing values
  - Note: this was sanctioned per Andras

EDA

Dataset seems to be unbalanced



EDA
Less information is known for everyday customers



0 = unknown
1 = male
2 = female

EDA

Start station may be correlated to trip duration