This report was written as part of the requirements for SENG474 / CSC578D: Data Mining, offered at the University of Victoria, taught by Dr. Alona Fyshe

Bay Area Bike Share

Predicting Station Status For Load Balancing

James Woo
V00816927
jameswoo@uvic.ca

Ramunas Wierzbicki
V00806259
mwierz@uvic.ca

Jakob Roberts
V00484900
jakob.m.roberts@gmail.com

Cole McGinn
V00780893
cmcginn@uvic.ca

Jordan Vlieg
V00779758
jvlieg@uvic.ca

Abstract—**This project focuses on predicting the station status given the time of day and weather. The team used open data provided by Bay Area Bike Share and completed data analysis which determined that the project was feasible. Furthermore, an in depth analysis and description of approach for selecting a data mining algorithm was done. The results have shown that decision trees have predicted station status with better than random accuracy and precision, however, recall was poor, with the algorithm performing worse than a random classifier.**

I. Problem Description & Problem Overview

Biking has become a popular choice for individuals to commute to work, tour around a city, or simply get from point A to B. This gives them the exercise they desire and peace of mind in terms of their contribution to cleaner alternatives. Some commuters may choose to purchase a bike that fits their needs. However, they need to maintain the bike and ensure that their bike is parked in a safe location. Additionally, tourists generally do not want to purchase a bike for their trip and instead rely on expensive companies to supply bikes for them to enjoy city amenities. To solve this problem, companies have started Bike Sharing Systems. Commuters no longer have to worry about maintenance and theft, while tourists, through a fee smaller than most pure rentals, can also enjoy the benefits to tour around. Bike sharing systems have become an international business and can be found in cities across Europe, Asia, and America.

Bay Area Bike Share is a public bike share program where riders can rent from 700 available bikes in 70 locations. The Bay Area Bike Share system currently faces a challenge: determining the amount of bikes needed in each station in order to service as many riders as possible. On any given day, the rideability of the day can change. Rideability is defined in terms of time of day and weather forecast. The day of the week and hour of the day greatly impacts the rideability; riders will typically rent bikes during the weekday during peak commute time hours, rather than late at night. Poor weather would indicate lower rideability, whereas good, clear weather would indicate higher rideability. Using the rideability attribute, this project focuses on predicting days in advance how many bikes or empty docks a station will have. First, a prediction can be made to determine how many bikes are required at each location for each hour of the day and Bay Area Bike Share can perform the load balancing before bikes are rented out. Additionally, further predictions can be made during peak rental hours and Bay Area Bike Share can perform further load balancing. This will ensure that the available 700 bikes are distributed appropriately based on usage statistics and expected availability. In turn, Bay Area Bike Share profits from the optimization, where a high percentage of the bikes are rented out, and riders are more likely to rent bikes due to increased availability.

II. Related Work

Similar work has been done in the bike share data analysis area. One particular dataset of interest is the Capital Bikeshare dataset. A Kaggle challenge was completed in May 2015 [2]. Several teams competed, and the team reviewed some work that was completed. For example, a project was completed to predict the demand for bikes in Washington D.C. [3]. The project focused on data visualizations on the count of bike rentals by time. Different patterns were also observed depending on the day of the week, where more rentals were made during the weekdays between 6-9AM and 4-8PM compared to the weekends. Additional analysis was completed to compare the count of bike rentals based on season (spring, summer, fall, winter), and account type (registered vs. non-registered). The team has planned on incorporating some of the ideas of this project. For example, in addition to the number of bike rentals made based on time, the team will incorporate weather data as well. Together, the two will be the base of the rideability attribute. Further related work on Bay Area Bike Share was found through Kaggle [4]. Several data analysis was completed by several groups on the open Bay Area Bike Share data set. The team found the analysis on "Time-Based Exploration of Bicycle Trip Data" the most useful to retrieve ideas. The purpose of the analysis was to determine how frequency of use varies over time. This was useful for the team because it provided a validation for the idea that rideability is affected by time of day and season.

Currently, Bay Area Bike Share has plans to expand the system. However, existing customers are unhappy with current service. Searching on the Bay Area Bike Share Facebook page, an existing customer has complained, seen in Figure 1 below.

Figure 1: Bay Area Bike Share Complaint [5]

Additionally, the team went through Yelp reviews to determine how many customers are experiencing similar problems. After reading through several reviews, the team discovered that roughly 10% of the 133 reviews contained complaints regarding the issue of availability. An example of the review is shown in Figure 2.

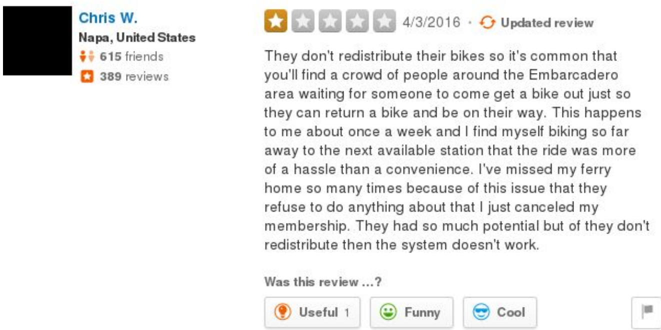

Figure 2: Bay Area Bike Share Yelp Review [6]

Based on real customer feedback, it was determined that the problem defined in this report exists. Data analysis must be done to determine when, where, and why the problem occurs.

Firstly, the team performed an initial analysis from the dataset for the Bay Area Bike Share, which is available online [1]. The team found that the bikes were frequently distributed in a way that customers could neither dock nor rent a bike at particular locations and times. The initial analysis allowed the team to narrow down what problems are a priority to be solved. One such issue is determining how many bikes a particular station will need before day start and if there is additional need for further load balancing during the day.

Secondly, the team wants to investigate and determine whether a bike station will be full or empty at a specific hour given the rideability at that hour. The goal is to prevent unavailability cases as they prevent customers from docking or renting bikes.

Thirdly, the team will investigate the most effective method of automating the process using several data mining algorithms, and pick the algorithm that provides the highest accuracy, precision, and recall.

Lastly, the team will provide an in depth analysis of the method chosen, the results, and conclusions.

III. Data Source and Data Content

During the initial data analysis for 2016 (year 3 data, 35,517,186 unique entries), it was found that, on average, any station is empty 0.9% (12.96 minutes) of the time and full 0.4% (5.76 minutes) of the time in a day. This information indicated that the program, when viewed as an average, has good service when it comes to usage and ability to dock bikes, however later analysis showed that issues were clustered around problem stations, and some stations had very high unavailability.

The year 3 status dataset contains the following information:

- Station ID
- Bikes Available
- Docks Available
- Time

The team discovered that this dataset is inconsistent with Bay Area Bike Share station information. In the dataset, the station IDs sometimes do not align with the station information. Additionally, there are some stations that do not exist in the station information data, but appear as a station in the dataset. To alleviate this issue, data with an invalid station ID is ignored.

The year 3 station dataset contains the following information: station ID, name, latitude and longitude, dock count, landmark, and installation. The team investigated the number of docks for each station. The minimum number of docks found was 11 and the maximum number of docs found was 35.

The year 3 trip dataset contains many columns which provide the date, trip duration, starting and ending terminal, and additional information that was not used. The year 3 weather dataset contains many columns which provide the date, mean temperature, weather forecast, and additional information that was not used. Together, the team investigated the trip frequency based on the date and weather information.

The features chosen for the training and testing model is as follows: *month*, *day of week*, *hour*, *mean temperature*, and *forecast rain*. Where each feature was assumed to be equally weighted. As mentioned in the previous sections, these attributes define the rideability attribute. The month, day of week, and hour gives the time aspect of rideability. The mean temperature and forecast rain provide the weather aspect of rideability.

The team hopes to be able to predict the station status days in advance by providing the features and station number as an input and expect an output classification for the given station at a given time. This information can then be used by Bay Area Bike Share in order to determine the number of staff required in order to perform load balancing.

First, the team was able to generate an infographic of Bay Area Bike Share year 3 status dataset. Provided in Appendix A, Figure A1, this infographic describes how many minutes at the given hour any station was empty or full over the whole year. From the graph, it is clear that there are empty and full bike stations at all times of the day. This indicates that prediction work may be necessary all the time. Furthermore, it should be noted that stations are empty more often than they are full, which shows that more customers are in transit than bikes are sitting in docks. Stations which are empty very often, but are not empty for very long are of special interest. The team believes that these stations are so in demand that it is beneficial for customers to form makeshift queues to access a bike, since customers are continually returning bikes to the station. To increase customer satisfaction and decrease waiting time, Bay Area Bike Share should monitor these stations, and transport additional bikes to these locations.

Next, another infographic was generated. Found in Appendix A, Figure A2 shows the number of occurrences a station had either an empty or full station for more than thirty minutes over the year. From the graph, it is clear that there are problem stations which are could greatly benefit from load balancing. To supplement Figure A2, a third infographic was generated, which can be found in Appendix A, Figure A3. The infographic shows the amount of time a station was either an empty or full. According to the two Figures, A2 and A3, there are several stations that are empty much more often than they are full. This indicates that these stations are likely used by commuters, either going or coming back from work, and seldom returned to that station by other customers. Therefore, customers would highly benefit from reallocation of bikes before they need them. Ensuring that stations have sufficient bikes or docks will decrease customer complaints, and increase satisfaction. The team also determined the problem stations. The problem stations are defined as any station that exceed 40 instances per year of either empty or full events. The list of problem stations for Year 3 are given in Table 1 below.

ID	Name	Docks	ID	Name	Docks
39	Powell Street BART	19	62	2nd at Folsom	19
41	Clay at Battery	15	63	Howard at 2nd	19
45	Commercial at Montgomery	15	64	2nd at South Park	15
47	Post at Kearney	19	65	Townsend at 7th	15
48	Embarcadero at Vallejo	15	66	South Van Ness at Market	19
49	Spear at Folsom	19	67	Market at 10th	27
50	Harry Bridges Plaza	23	69	San Francisco Caltrain 2	23
51	Embarcadero at Folsom	19	70	San Francisco Caltrain	19
54	Embarcadero at Bryant	15	71	Powell at Post	19
56	Beale at Market	19	72	Civic Center BART	23
57	5th at Howard	15	73	Grant Avenue at Columbus Avenue	19
60	Embarcadero at Sansome	15	76	Market at 4th	19
61	2nd at Townsend	27	82	Broadway St at Battery St	15

Table 1: Problem Stations

The team believes that stations that exceed 40 instances per year is a good estimate that they are problematic. It should be noted that the problem stations found are between the minimum and maximum dock count, 11 and 35 respectively. These stations are also found at popular destinations for commuters and tourists alike.

Additionally, to verify that weekdays are more popular in terms of rental rate, another infographic was generated. This was also generated to justify that the day of the week affects rideability. Found in Appendix A, Figure A4, it shows the number of trips made by day. The blue dots represent the number of trips made on a weekday, whereas the red dots represent the number of trips made of a weekend. It is clear from the infographic that more people are renting bikes during the weekday rather than the weekend. This is most likely due to daily commuters, rather than casual riders. Additionally, the infographic shows that the number of trips made are correlated to average temperature. The colder the average temperature, the lower the ridership. Conversely, the warmer the average temperature, the higher the ridership. This also justifies that weather influences rideability.

Finally, the team verified the popular commute times. In Appendix A, Figure A5 shows the frequency of trips made based on time of day. From the figure, it can be inferred that peak commute times are around 9AM and 5PM. This also justifies that time of day influences rideability.

IV. METHOD AND APPROACH

In order to determine the station status given rideability, several classification algorithms can be used, these include, but are not limited to Gaussian Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, Decision Trees, Random Forests, or Perceptron. Classifying the station status will determine whether a given station requires load balancing. The team proposes the following classes: *Empty*, *Full*, and *None*.

The classes indicate the station status, where Empty means that the station is nearly empty, Full means the station is nearly full, and None means the station is neither Full nor Empty, given the rideability. To determine if a station is Empty, Full, or None, the team used a threshold T in order to classify the station given the rideability. If the number of bikes is less than T, then the station is considered Empty. If the number of available docks is less than T, the station is considered Full. If neither of these conditions are met, the station is considered None. This approach made classification simple. Another approach was to predict the number of bikes available given rideability. However, this approach was dismissed because the team believed that this approach would introduce more difficulty and lower accuracy in the results. The team chose a threshold of T=4 because, in a practical sense, we believe each dock should have at least 3 bikes or docks available at all times. From the previous data analysis, the team found that the minimum number of docks was 11. Setting the threshold higher than 4 would produce negative results for stations with low number of docks. In this example, any station with 11 docks would most likely report either empty or full, when in reality, they are neither.

To determine the optimal algorithm for this dataset, the team ran every algorithm described. The results were then analyzed and compared against a random classifier, and the algorithm which provided the highest accuracy, precision, and recall was chosen. The random classifier simply chose a classification, in our case, either 0, 1, or 2. This results in an accuracy of 33.3% since it has a one third chance of guessing correctly,. The recall of the random classifier is also 33.3% since it will correctly predict ⅓ of all instances. The precision was calculated using the standard formula and it was found that the precision would be 0.149 for an empty prediction, and 0.088 for a full prediction, both of which are very low in comparison to decision tree or random forest.

V. RESULTS

Based on the approach described in the previous section, the team ran multiple predictions for several stations with the different algorithms described in the approach. Table 2 below summarizes the results found for three different algorithms, their respective F1 scores for Empty and Full status predictions, and accuracy for the given threshold. These predictions were made using the assumption station ID's should be part of the prediction.

	Station Status (T = 4)		
Algorithm	F1 Empty	F1 Full	Accuracy (%)
Gaussian Naive Bayes	0.023	0.008	79.8
Decision Tree	0.157	0.212	75.3
Random Forest	0.126	0.115	78.1
Random Classifier	0.175	0.096	33.3

Table 2: F1 Scores and Accuracy for Status Prediction

The team evaluated several algorithms described previously, however, the team found that Bernoulli Naive Bayes, Multinomial Naive Bayes, and Perceptron all performed poorly, predicting no empty or full instances. For that reason, the team removed these algorithms from the analysis. Furthermore, the table shows that Gaussian Naive Bayes did make predictions, however, the precision and recall was too low, much lower than a random classifier. Again, for this reason, the team has removed Gaussian Naive Bayes from further analysis. Decision Tree and Random Forest were two algorithms that performed relatively better than a random classifier. The results reveal something interesting about the predictions. The accuracies are relatively high, but the F1 harmonic means scores were low. This is because the dataset contains mostly None classifications, which artificially inflate the accuracies.

In comparison, Table 3 shows the results of two predictions, each prediction on a single station. The team has chosen stations 45 and 11 for the purpose of demonstration since station 45 is a more active station and station 11 is not.

P - Precision, R - Recall, E - Empty, F - Full, A - Accuracy

	Station 45 (T=4)				
Algorithm	PE	RE	PF	RF	A (%)
Decision Tree	0.457	0.338	0.203	0.156	49.2
Random Forest	0.506	0.326	0.201	0.081	52.7
Random Classifier	0.119	0.33	0.056	0.33	33.3
	Station 11 (T=4)				
Algorithm	PE	RE	PF	RF	A (%)
Decision Tree	0.003	0.014	0.003	0.007	82.9
Random Forest	0.006	0.009	0.14	0.007	91.2
Random Classifier	0.119	0.33	0.056	0.33	33.3

Table 3: Precision, Recall, Accuracy of Two Stations

This analysis revealed that problematic stations will produce better predictions, whereas non-problematic stations produce poor predictions. See Appendix B for the full results for problematic stations. This revelation allowed the team to focus on station by station predictions, rather than using station ID as an attribute for predictions. For problematic stations, the predictions have higher precision, recall, but suffer in accuracy. This is a tradeoff that the team considered. Some interesting results of running predictions on all problematic stations were found. No station was able beat a random classifier in all categories (precision, recall, and accuracy). The stations typically performed better on predicting empty state rather than full state. Additionally, the results showed that random forest provided higher accuracies compared to decision trees. However, most of the time, decision trees offered higher precision or recall. The team believes that precision and recall is more important compared to accuracy for this application. Therefore, the best algorithm was determined to be decision tree.

VI. Discussion

The team believes that choosing the right features heavily impacted the results of the predictions. The current predictions are using five manually picked features. The manually picked features did have justification, however, the team did not methodologically analyze the chosen features or compared them to additional sets of features.

A possible reason as to why the bayesian methods produced no results may be due to the data. The training set was very large, heavily biased towards the ‘none’ classification, and the features were assumed to be independent. However, time and weather are relatively dependent (rising temperature throughout the day, dipping temperature throughout the night) and directly influence the rideability. The team also believes that a factor in the low recall ability of the classifier is due to the long term prediction nature of the classifier. The classifier has no access to information such as the current number of available bikes in a station when making a prediction for a few days in the future.

Given more time, the team would have explored further into feature selection, which requires a more in depth knowledge of the bike sharing domain. The goal of feature selection in this project's case is not to remove or add new feature but to help the team determine features which will increase the accuracy, precision, and recall of the predictions. The team could use feature selection methods such as the filter method, wrapper method, or embedded methods to do so. Scikit learn also provides recursive feature elimination, which the team could have explored.

The predictions made in this project can help Bay Area Bike Share with load balancing. Bay Area Bike Share can allocate proper resources days in advance to improve customer satisfaction through load balancing.

VII. Conclusion and Future Work

Bay Area Bike Share is a great program for commuters and tourists alike. However, existing customers are unsatisfied with the current program. There are several complaints that customers are unable to dock or rent bikes, which inevitably reduce customer satisfaction and ultimately the company's profits. In order to combat this problem, Bay Area Bike Share should invest in predictive load balancing. With the use of the projects prediction model, Bay Area Bike Share could plan days in advance to ensure that sufficient staff is available in order to carry out load balancing.
The results of the project showed that the prediction system is able to perform better than a random classifier in terms of precision, and accuracy. However, the current system cannot beat a random classifier in all categories, namely recall. The best algorithm determined was decision tree, which produced the best precision and recall values compared to other classifiers like random forest. Further work to select features using formal methods will most likely increase the accuracy, precision, and recall of the predictions. If implemented, the team would encourage use of current data along with our training to boost the accuracy of the predictions. Although the predictions are planned to be used days in advance, predicting the day of would allow for incredible accuracy and really prove the use of our work.

Future improvement to our work is to take into account more features like current data, large events, and possible other features that are determined to have a larger impact on predictions. In order to make our prediction algorithm a true success in reality, we need to ensure the features mentioned above are taken into account and refined properly.

VIII. Distribution of Tasks Among Members

Democratic Project Manager - Ramunas Wierzbicki
Executive Data Acquisition - Jordan Vlieg
Executive Code Review - James Woo
Executive Algorithm Design - Cole McGinn
Executive Domain Expert - Jakob Roberts

In the final project, team members were each assigned a main role to this ensure that each task is completed properly. The team as a whole worked together to look at the dataset and determine what can be analyzed. Additionally, the team fully contributed to writing the report and the final presentation. Specifically, Cole designed a Python script to predict station status and worked on the report. Ramunas worked on creating python scripts to generate additional visualizations and organized and formatted the final presentation. Jakob, Jordan and James worked on verifying the results of the predictions made, determined the precision, recall, and accuracies of the results, and worked on determining the problem stations along with running the predictions for all of them. Overall, the team distributed tasks equitably and had no issues working together.

References

[1] Bay Area Bike Share, "Introducing bay area bike share, your new regional transit system," in Bay Area Bike Share, 2013. [Online]. Available: http://www.bayareabikeshare.com/open-data. Accessed: Oct. 2, 2016.

[2] Kaggle, “Bike Sharing Demand”. [Online]. Avilable: https://www.kaggle.com/c/bike-sharing-demand Accessed: Nov 21, 2016

[3] bikesharekaggle, “The BikeShare Kaggle Challenge”. [Online]. Available: https://sites.google.com/site/bikesharekaggle/ Accessed: Nov 21, 2016

[4] Kaggle, “SF Bay Area Bike Share”. [Online]. Available: https://www.kaggle.com/benhamner/sf-bay-area-bike-share Accessed: Nov 21, 2016

[5] Facebook, [Online]. Available: https://www.facebook.com/BayAreaBikeShare/posts/695791523918193 Accessed: Nov 1, 2016

[6] Yelp, “Recommended Reviews for Bike Area Bike Share”. [Online]. Available: https://www.yelp.com/biz/bay-area-bikeshare-san-francisco?hrid=-3RMMdnC1aVPX3s9-1RqHQ&utm_campaign=www_review_share_popup&utm_medium=copy_link&utm_source=(direct) Accessed: Nov 1, 2016

Appendix A: Figures

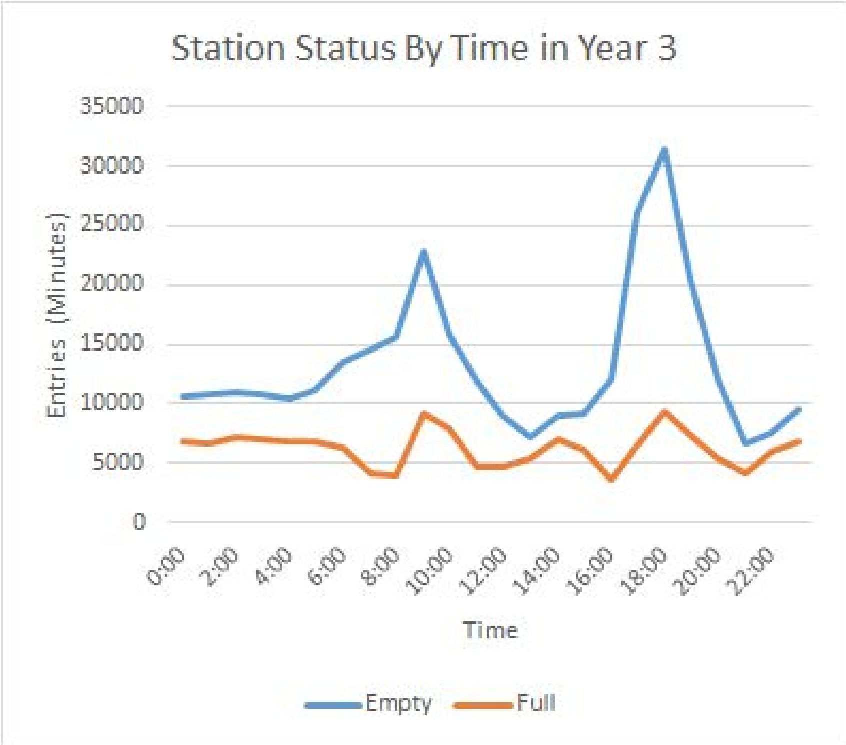

Figure A1: Station Status By Time in Year 3

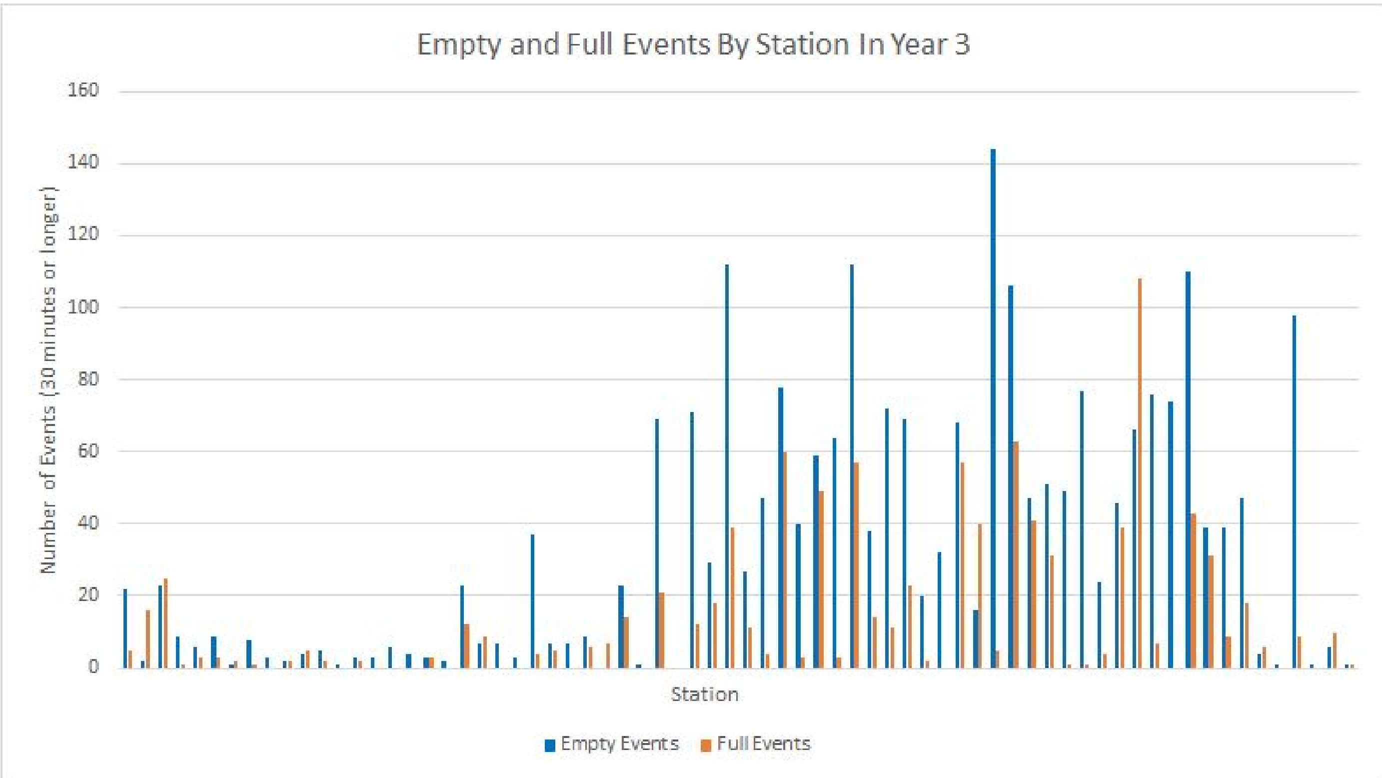

Figure A2: Empty and Full Events By Station, for Year 3

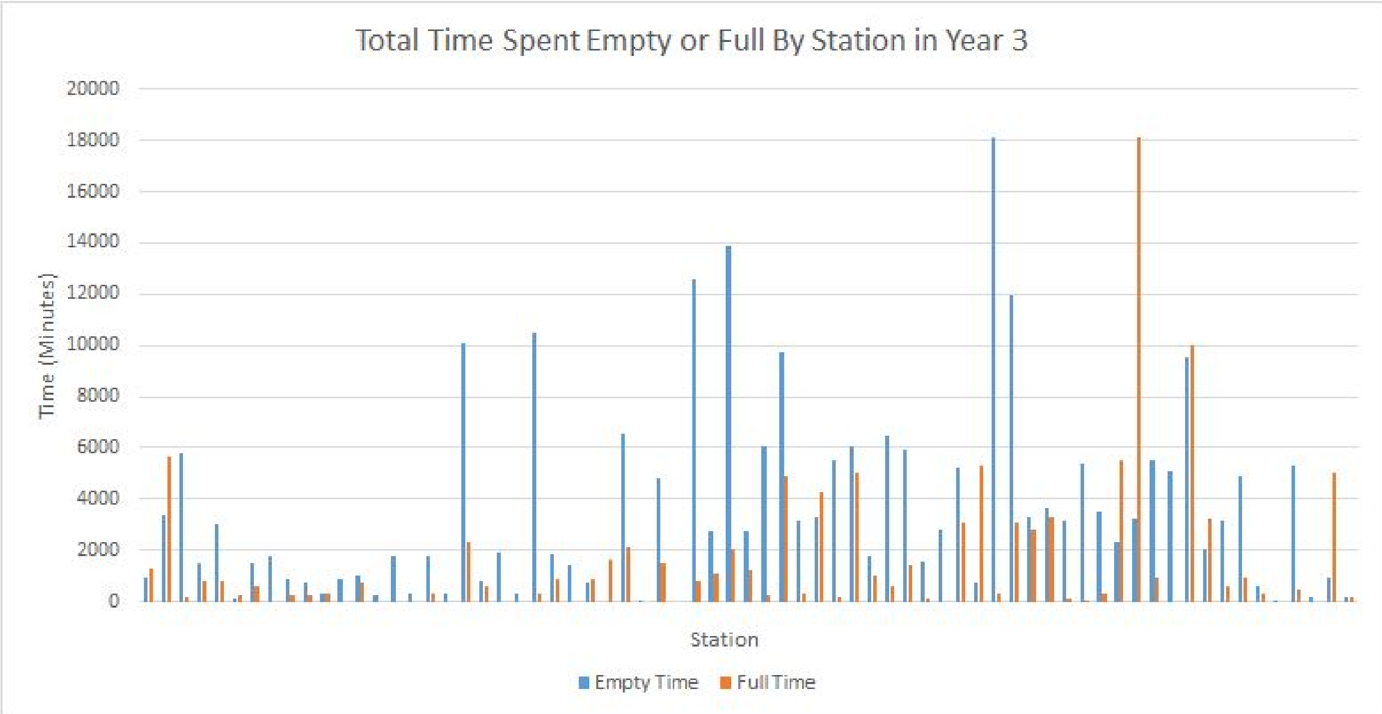

Figure A3: Total Time Spent Empty or Full By Station, for Year 3

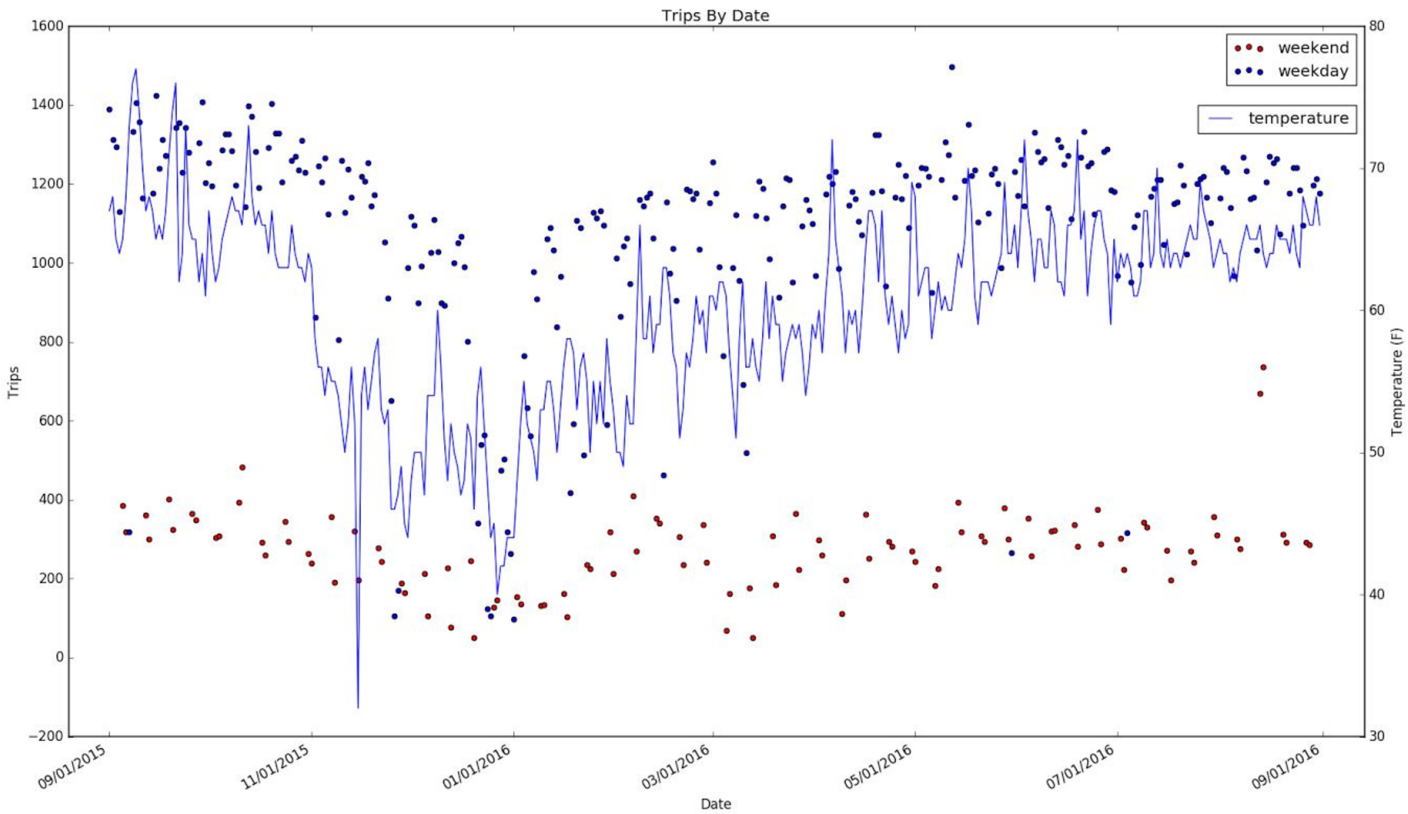

Figure A4: Trips By Date, for Year 3

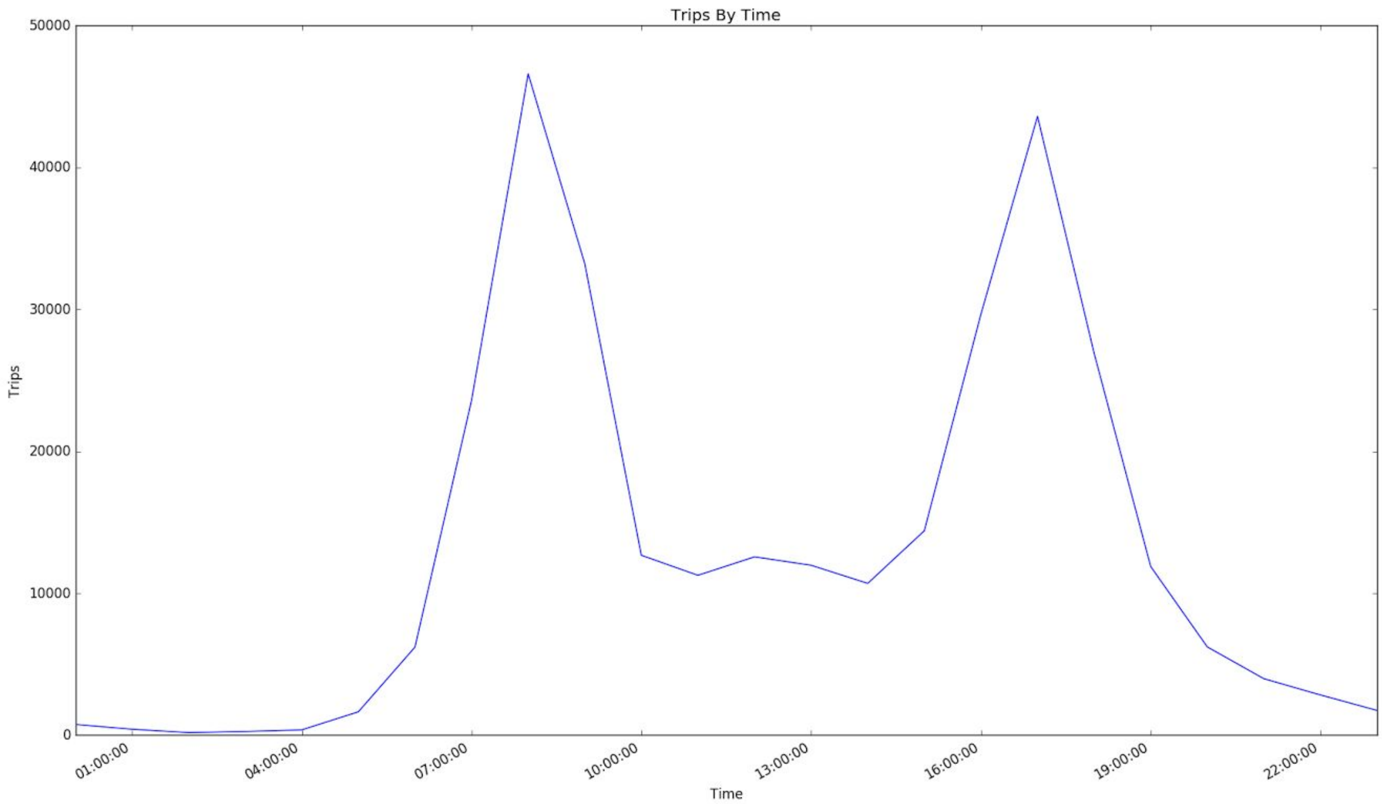

Figure A5: Trips By Time, for Year 3

Appendix B: Problematic Station Predictions

DT - Decision Tree, RF - Random Forest, RC - Random Classifier, P - Precision, R - Recall, E - Empty, F - Full, A - Accuracy

	Station 39 (T=4)						**Station 41 (T=4)**						**Station 45 (T=4)**				
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)
DT	0.192	0.100	0.056	0.213	63.5	DT	0.369	0.359	0.172	0.120	56.6	DT	0.457	0.338	0.203	0.156	49.2
RF	0.142	0.032	0.065	0.162	70.1	RF	0.389	0.309	0.164	0.057	60.0	RF	0.506	0.326	0.201	0.081	52.7
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3
	Station 47 (T=4)						**Station 48 (T=4)**						**Station 49 (T=4)**				
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)
DT	0.292	0.311	0.406	0.063	67.7	DT	0.439	0.441	0.109	0.115	56.0	DT	0.072	0.092	0.005	0.008	75.3
RF	0.337	0.244	0.055	0.040	72.6	RF	0.441	0.366	0.099	0.059	59.7	RF	0.069	0.042	0.031	0.093	81.8
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3
	Station 50 (T=4)						**Station 51 (T=4)**						**Station 54 (T=4)**				
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)
DT	0.241	0.159	0.259	0.233	64.3	DT	0.285	0.190	0.065	0.184	66.9	DT	0.290	0.241	0.372	0.528	54.9
RF	0.221	0.091	0.296	0.183	68.9	RF	0.300	0.110	0.082	0.101	71.8	RF	0.312	0.178	0.384	0.530	57.1
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3
	Station 56 (T=4)						**Station 57 (T=4)**						**Station 60 (T=4)**				
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)
DT	0.238	0.239	0.057	0.072	70.1	DT	0.225	0.178	0.086	0.143	59.4	DT	0.265	0.280	0.213	0.236	53.3
RF	0.230	0.135	0.088	0.051	75.3	RF	0.190	0.086	0.070	0.067	65.8	RF	0.277	0.219	0.228	0.183	57.6
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3
	Station 61 (T=4)						**Station 62 (T=4)**						**Station 63 (T=4)**				
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)
DT	0.118	0.110	0.203	0.156	79.2	DT	0.348	0.291	0.050	0.012	57.1	DT	0.374	0.317	0.216	0.231	65.1
RF	0.152	0.055	0.117	0.017	82.5	RF	0.405	0.251	0.042	0.004	61.1	RF	0.415	0.270	0.266	0.157	68.1
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3
	Station 64 (T=4)						**Station 65 (T=4)**						**Station 66 (T=4)**				
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)
DT	0.147	0.232	0.192	0.168	64.9	DT	0.192	0.200	0.098	0.125	60.5	DT	0.169	0.199	0.012	0.050	73.9
RF	0.152	0.150	0.187	0.110	70.9	RF	0.190	0.130	0.106	0.082	66.0	RF	0.188	0.127	0.009	0.025	80.2
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3
	Station 67 (T=4)						**Station 69 (T=4)**						**Station 70 (T=4)**				
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)
DT	0.192	0.164	0.008	0.029	77.6	DT	0.308	0.250	0.260	0.222	62.8	DT	0.501	0.374	0.439	0.246	55.6
RF	0.207	0.106	0.017	0.029	80.9	RF	0.302	0.172	0.279	0.150	65.8	RF	0.538	0.369	0.502	0.194	56.9
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3
	Station 71 (T=4)						**Station 72 (T=4)**						**Station 73 (T=4)**				
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)
DT	0.149	0.153	0.034	0.071	67.4	DT	0.267	0.274	0.023	0.395	68.1	DT	0.240	0.175	0.251	0.339	50.6
RF	0.165	0.102	0.041	0.036	75.5	RF	0.311	0.197	0.016	0.198	74.8	RF	0.229	0.110	0.249	0.230	56.0
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3
	Station 76 (T=4)						**Station 82 (T=4)**										
	PE	RE	PF	RF	A (%)		PE	RE	PF	RF	A (%)						
DT	0.192	0.349	0.067	0.085	65.3	DT	0.256	0.208	0.154	0.227	60.0						
RF	0.230	0.306	0.069	0.026	72.9	RF	0.279	0.174	0.133	0.115	65.7						
RC	0.119	0.33	0.056	0.33	33.3	RC	0.119	0.33	0.056	0.33	33.3						

Table B1: Problematic Station Prediction Results