Analyzing Austin B-Cycle Product Consumption Trends

Nathan Alvarez Olson, Rohan Chaudhry, Danny Diaz, Aparna Kakarlapudi

Problem Statement

Currently, there has been a nationwide boom in the "micro-mobility" market, which includes car sharing and electric scooters. One example of this is B-Cycle, a public bicycle sharing company owned and operated by the city of Austin since 2013. However, it has seen a sharp decrease in rides in the past year, particularly with UT students and in the downtown area. This is largely the result of the introduction of dockless scooters, operated by companies such as Lime and Bird, to Austin. This is of particular concern to Austin, as the city partnered with a non-profit to provide the B-Cycle service, and in fact used federal grants to purchase the roughly $2.5 million in equipment that makes up the B-Cycle system. As such, declining usage of B-Cycle means that Austin's large investment is going to waste.

Through analyzing the data collected by B-Cycle ride and daily weather data, we hope to create a model that predicts consumer trends for customers' use of B-Cycle. We hope to answer questions such as "How long do rides last and where do customers go?", "Is this dependent on the day of the week or the weather?", "Can we predict the number of rides in a day?", and "Can we identify distinct and different types of users?". Ultimately, we will have mixed success in answering these questions. Through this analysis, we hope to recommend a course of action for Austin B-Cycle that will help Austin preserve the value of its investment, combat encroaching scooter companies, and set itself up for long term success.

Solution

Data used: We examined two datasets. The first, primary dataset details B-Cycle usage in Austin from December 2013 to July 2018. It contains 1.08 million rows— with each row detailing a single B-Cycle ride—and twelve features, including date, check-out time, trip duration, check-out and check-in location, and membership type of the user. The second, supplementary dataset details weather in Austin from December 2013 to July 2017. It has 1319 rows— one for each day—and 21 features, including date; high, low, and average temperature, humidity, wind speed, visibility and dew point; precipitation in inches; and any weather events that occurred.

Approach and assumptions: Our project was divided into three main phases: data exploration, data engineering, and model building. The first stage involved creating graphs that highlighted trends over time, reflected how member type composition changed over time, and showed the distribution of trip length. The second stage included deleting or imputing NaN values, deleting columns not relevant to the question, and creating new .csv files that would be used moving forward. We then joined the weather and B-Cycle datasets on date, allowing us to use weather data in our predictive models. Much work was then put into feature engineering, such as grouping the rides by day and finding the average trip length per day. Lastly, we used Darwin to create predictive models. These three stages are expounded upon in the notebooks.

Application of Darwin: Darwin was used to build two models with predictive capacity, which we will detail shortly. We also built other models, but they either did not work or had very weak predictive power. These failed models are detailed in the *General Challenges* section.

The first model we built, which had an R^2 score of roughly 0.72 to 0.80 depending on the training / test set split, aimed to predict the total number of rides in a day. The features the model had at its disposal included temperature data, the weekday, the month, and the average ride duration during that day. The second model we built, which had an R^2 score approximately equal to 0.65, had the same features available, but instead aimed to predict the average ride duration during a day. When this model was run initially, the R^2 score was nearly half of the final result, due to the existence of some days with an unusually large average ride duration. When we graphed the predicted trip duration versus the actual trip duration for this first-run model, we saw these outliers. The fact that Darwin showed us the very points which were wrecking the accuracy of our model was very useful. After removing the days which had an average trip duration longer than one hour, we saw our R^2 score double.

These two models constitute what we would consider to be our immaculate successes with Darwin. We loved the results Darwin gave us, and we especially appreciated the way in which it automatically chose both the best model and the best parameters. Specifically, we would not have thought to use an XGBRegressor model and a temporal convolutional neural net, but Darwin chose these models for us.

However, there were many aspects of Darwin which we found to be frustrating and non-intuitive. Firstly, creating and training the model was a little bit of a black box. When it worked well, this was not a problem, but when we could not get the model to work, we had few ideas on how to fix it. Secondly, code which had previously worked would often fail to run multiple times in a row. Since running the model often took around fifteen minutes, this resulted in a lot of wasted time. Furthermore, accuracy was too sensitive to the max training time, as we found that training for over three minutes guaranteed poor performance. As far as we could tell, there were few hyperparameters other than max training time which we could tweak. Also, downloading predictions took surprisingly long, and would sometimes randomly fail. If one re-ran the code because it failed, we found that we needed to delete the previous model that Darwin created, otherwise the R^2 score would decrease with each re-run of the code. Nathan encountered a strange error while attempting to predict the return kiosk. This error was reported to Darwin. A satisfactory solution was never found, and the initial error was unclear in the first place. Lastly, we did not like the lack of cross validation in Darwin.

Team Engagement

The roles were not clearly defined. However, Danny focused on feature engineering and regression, Rohan and Nathan focused on Darwin models, and Aparna focused on the report.

General Challenges

The biggest challenges we ran into related to model building. Those specific to Darwin were listed in the *Application of Darwin* section. Aside from Darwin, many of our challenges related to runtime. The B-Cycle dataset contained over one million rows and we found that running any model on this dataset was infeasible on our computers. Lastly, and perhaps most frustratingly, we sunk a lot of time building models and clustering data which either failed or had very poor predictive capacity. We tried using both Darwin and a decision tree regressor in scikit-learn to predict the trip time for each individual ride, but had horrible accuracy. In the case of Darwin, this gave an R^2 close to zero. We tried predicting return kiosk for each ride using Darwin, but ran into an error when downloading the predictions which we could not solve. We tried predicting member type, but achieved no meaningful results. Some of these failures are included in the notebook.

On the other hand, though feature engineering was perhaps the biggest time sink in our project, it was much more successful than data analysis. However, we still faced some small issues. From B-Cycle's raw dataset, almost 60 member types were listed in its "Membership Type" column, but the majority of that 60 comprised of similar but differently named members. We ended up consolidating them into 10 types. Additionally, the precipitation column gave the precipitation in inches, it sometimes had "T" for trace precipitation instead of a float. We decided to substitute in the value of 0.005 inches for these rows, as this is less than the smallest nonzero amount of precipitation in that column and aligns with the definition of trace precipitation we found. Lastly, we also lost over a year of B-Cycle data, as we did not have weather data for these dates.

Next Steps

We created two models that predict the average ride length per day and the total number of rides per day. Using these models, B-Cycle could improve their maintenance and stocking schedules to meet expected demands throughout the year. On days with a low number of expected rides, B-Cycle can pull bikes for maintenance. On days with a high number, B-Cycle can stock more bikes in their stations. In the future, if B-Cycle can find a way to predict the number of checkouts at particular stations, this solution would improve their maintenance schedules even further. Additionally, looking into where bikes are going after being checked out at a particular station may help B-Cycle determine where they can add additional stations. B-Cycle currently has at least 7 Member Types, and our analysis suggests that B-Cycle consolidate its Member Types based off the five clusters shown in the *Member Ride Duration with Weather* notebook.