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. Through the likes of Bird and Lime, this market has seen a product shift from pedal-powered bikes to e-scooters, and this change can be clearly seen throughout the streets of Austin [1]. These new era tech companies are replacing the position once held by traditional bike sharing companies, like Austin B-Cycle, and they threaten to drive traditional bike sharing companies out of business.

However, one advantage that Austin B-Cycle has over the scooter companies is its age. Scooter companies did not hit the streets until mid-2018, whereas B-Cycle landed in Austin late-2013. Through analyzing the data collected by B-Cycle’s rides, which detail start and stop locations for rides and their duration, and public data collected for daily weather, we are able to create a model that associates consumer trends for B-Cycle’s product use given the day of the year and weather. Through this analysis, we can recommend a course of action for Austin B-Cycle that will help the company combat encroaching scooter companies and set itself up for long term success in the Austin market.

Without docks, irresponsibly placed scooters can injure pedestrians and damage cars, making the rise of scooters as a public safety issue [2]. The University has taken measures to mitigate these risks by enforcing speed zones and parking zones for scooters, but a large scale city cannot manage to do so. Austin B-Cycle currently has the infrastructure set up to remain in accordance with city law and safety regulations, so with the right long term planning, B-Cycle can continue growing its share within the Austin market.

Furthermore, using the data can identify the reasons explaining why customers use B-Cycle. For instance, how long do rides last and where do customers go? Is this dependent on the day of the week or the weather? How do riders use the service in different ways? Can we identify distinct and different types of users? Creating a predictive model may help us answer these questions.

Sources:
[1] https://www.bicycling.com/news/a26623484/lime-bikes-scooters/
[2] https://www.vox.com/2018/8/27/17676670/electric-scooter-rental-bird-lime-skip-spin-cities

Solution

Data

We examined two datasets. The first, primary dataset details B-Cycle usage in Austin from 21 December 2013 to 31 October 2018 inclusive. 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 21 December 2013 to 31 July 2017 inclusive. 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 scrubbing, data engineering, and data visualization. When scrubbing data, we created unique .csv files for each question we approached. The entire group focused on feature engineering at this stage, actively discussing the pros and cons of every feature to use, and then finalized a dataset unique to each prompt. This stage includes deleting or imputing NaN values, deleting columns not relevant to the question, and creating new .csv files that would be used moving forward. After scrubbing, we worked to engineer each dataset to best fit its respective question. This includes predictive modeling, cluster analysis, XGB regressor, neural net and regression algorithms. Lastly, we focused on the visualization of our data to better analyze our results. This includes creating graphs that exhibit trends over time, scatter plots that show the correlation between predicted and real values, and cluster maps that reflect how different member types are more similar than we would have thought.

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, including (in descending order of frustration):

- 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.
- 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.
- Accuracy was too sensitive to the max training time. We found that we were guaranteed poor performance after three minutes of training.
- As far as we could tell, there were few hyperparameters other than max training time which we could tweak.
- 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.
- A lack of cross validation in Darwin.

Innovation and Impact

We innovated our data analysis by involving weather data with the B-Cycle dataset. Even though the combined datasets limited the overall chronology of the data, the combined datasets provided deeper insight when observing and modeling consumer patterns. Combining the datasets allowed us to more reliably predict when and where to expect, relocate, and otherwise handle bikes at the B-Cycle kiosks on a given day. This information can help B-Cycle manage each of its kiosk capacities throughout every day in a year, and it can help B-Cycle decide where to expand within the Austin area. Through our high level analysis of member ride patterns and overall composition over time, B-Cycle can move forward by consolidating its current Membership Types to focus on core consumer groups. B-Cycle's current dataset shows a lack of organized structure around its Membership Types, so our analysis can help them classify those core users, recreate a price model, and focus on streamlining growth in the future.

Team Engagement

Danny was mostly responsible for feature engineering and regression. Rohan and Nathan focused on Darwin models. Aparna was responsible for the report.

General Challenges

The biggest challenges we ran into all related to data analysis. The ones which were specific to Darwin were listed in the *Application of Darwin* section. On the other hand, many of our challenges related to runtime. The BCycle dataset contained over one million rows. We found that running any model on this dataset was infeasible on our computers: clustering for even 3000 rows of data took over 10 minutes to run once. As mentioned earlier, these long runtimes meant debugging took a long time, in addition to preventing data analysis on the large data set.

Lastly, and perhaps most frustratingly, we sunk a lot of time building models which either failed or had very poor predictive capacity. We spend a lot of time trying to cluster the data, but either found no meaningful clusters or could not run the code in a reasonable amount of time. 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.

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. For example, there is an "Annual", "Annual Member" and an "Annual Membership" type in this column. This led to a more intensive data cleansing process, and renaming the approximately 60 member types into 10 main types took a significant amount of time. Another issue we face was what to do with the precipitation column. Though this column gave the precipitation in inches, it sometimes had "T" for trace precipitation instead of a float. We debated what to do with this value for a long time, before ultimately deciding to substitute in the value of 0.05 inches for these rows, as this is less than the smallest nonzero amount of precipitation in that column and is lies within the range we found online for what "trace precipitation" indicates. Lastly, we also lost over a year of BCycle data, as we did not have weather data for these dates.

As for problems faced within our team, finding a time when all four of us could meet was difficult. We also face some (minimal) difficulties constructing our GitHub.

Next Steps

Following our analysis, we found two solutions: predicting the average ride length per day and predicting the total number of rides per day (accuracies discussed earlier). Based on these solutions, B-Cycle can improve their maintenance and stocking schedules to meet expected demands throughout the year. For instance, if the total number of rides on a given day is low, then B-Cycle can choose to pull bikes for maintenance on that day. Conversely, if the total number of rides is high, then 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. From a competitive perspective, B-Cycle can improve by introducing electric bikes so it can compete with various scooter services around the city. Lastly, improving the UI of the app could significantly increase user satisfaction and experience.