Classifying Differences in Behavior of 1-Time Users and Subscribers in San Francisco's Bike Share program

Alexander Schuler
Lehigh University
ajs520@lehigh.edu

ABSTRACT
This paper applies decision tree classification models to trip data on rental bike share trips taken by subscribers and one-time customers in San Francisco. It then gives possible interpretations of findings and performs exploratory analysis based on questions raised by features of the resulting classification models.

CCS Concepts
• **Machine Learning -> Classification Models**

Keywords
Customer: An individual who paid for a single ride;

Subscriber: An individual who pays a flat fee for unlimited rides;

1. INTRODUCTION
When working with data related to customer behavior, it's often beneficial to be able to place customers into distinguished behavior groups. In this study, we look at ride data with customers already separated into 2 groups, those who subscribe to the service and those who paid per-ride for a trip. Our hope is to be able to look at trends in each group to be able to perform more targeted marketing, as opposed to employing a more general strategy that would attempt to encompass and attract the entire group of potential riders.

2. DATA OVERVIEW
The data used for this analysis comes from the San Francisco bay area bike share program run by Ford. The data ranges from 2013 through 2015, and is split into separate CSV files about each of the drop-off / pickup stations, individual trips, and weather data.

Of this set, we examine the trip, weather, and station tables. The trip file contains information on the start and end times of a trip, duration, date, starting point, end point, and whether the person taking the trip was a subscriber or customer. The weather table included mean, high, and low temperature, visibility, humidity, wind, and rainfall information. The station table contained a mapping from station id's to their names, as well as the latitude and longitude of each station.

Columns used:

trip	duration, start_station, end_station, start_date, end_date, subscription_type
weather	date, mean_temperature, precipitation_inches
station	id, name, lat, long

3. EXPLORATORY ANALYSIS (1)
Before using a classification algorithm to pick out differences in the behavior of customers and subscribers, it could prove useful to take a general survey of the data. A simple look at the most popular trips for customers and subscribers immediately highlights a difference in preferred trips. The two preferred trips for subscribers are between San Francisco Caltrain and Townsend and 7th. This is a relatively short (0.5 miles) ride in the Mission Bay area. This is a predominantly up-and-coming residential area filled with university buildings and luxury apartments. For customers, the most common rides centered around the Embarcadero. This area contains multiple parks, the San Francisco Museum of Modern Art, and upscale shops in the Union Square area.

The Bike stations are scattered mostly throughout the northeastern part of San Francisco, ranging from Telegraph Hill in the North down to the Design District in the south.

Plot of Bike Stations

4. DATA CLEANING AND FEATURE EXTRACTION
With this particular data set, we were lucky to have gotten complete, labeled data and no erroneous-looking entries (such as trips starting before they ended).

Verification of Trip Dates

```
In [27]: trip_parsed.where(trip_parsed.start_date > trip_parsed.end_date).count()
Out[27]: 0
```

However, in order to get the most out of our decision tree modeling, I first had to remove redundant (such as start_station_name and start_station_id) and useless fields (such as trip id and bike id). In addition, I had to separate out the date

fields out into a usable representation holding the hour of day, day of week, and the month. I then discarded the year and day of month attributes as people's behavior is determined more by what day of the week it is, what time it is, or which season it is, and not at all by what year or day of the month it is. These individual pieces of the data are useable by the classifier, unlike a 64-bit Unix timestamp, which a decision tree is completely unable to correctly interpret.

After converting dates, I joined in weather data for each trip in hopes that the classifier might find some variance in the responses of customers and subscribers to rain and/or hot and cold temperatures. Since spark's decision trees only work with numerical data (categorical is fine, but it won't do strings), I also had to encode the subscriber/ customer column into a binary 0 or 1 format.

5. DECISION TREE CLASSIFICATION

5.1 Determining a Baseline Measure of Success

Before beginning to work with a decision tree and evaluating the success of our classifier, I needed to determine what kind of accuracy scores would imply that our model was picking up on differences in the behavior of the two groups and not just randomly guessing at each data point. To establish a mark to beat, I computed the probable accuracy of a random classifier. This very naïve random classification model simply takes the percentage of the time that a label occurs in the test set, and guesses that value that percentage of the time.

In our case, the data is split with 85% of rides being subscribers and 15% being customers, so it would randomly guess "subscriber" 85% of the time, and "customer" the remaining 15% of the time. We can compute the expected accuracy of this type of classifier with simple statistics, saving us the need to build and run such a machine.

G_i: The case where the random classifier guesses i

$$P(Correct) = P\left((G_i \cap i) \cup \left(G_j \cap j\right) \cup \dots\right)$$

$$P(Correct) = P(G_{customer}) * P(customer) + P(G_{subscriber}) * P(subscriber)$$

$$P(correct) = .15 * .15 + .85 * .85$$

$$P(correct) \cong 0.75$$

From this analysis, we see that we should expect our decision tree to perform with at least 75% accuracy, and that any value close to 75% will indicate that the classifier is unable to determine any difference between our labeled categories, and is instead almost randomly guessing at which group the current vector belongs to.

5.2 Data preparation and assembly for use with Spark

In order for spark's machine learning libraries to be able to work with any data, they require it to be in a vector format. Spark provides an assembler to convert multiple columns into a compact format as either a DenseVector (suitable for data with many non-zero values) or SparseVector (suitable for data with mostly 0 / null fields). Spark is also capable of sorting out and transforming categorical data with its VectorIndexer class, which transforms columns with few distinct values into the sequence of natural numbers starting at 0 to prevent the tree from mistaking them for numerical features. This tends to boost accuracy slightly. Spark will also take a grid of testing parameters on which it tests the cross products to find hyperparameters that best fit the data. For example, running a pipeline with grid:

Param	Values
maxDepth	[5,20]
maxBins	[100,300]

will build and compare 4 trees, 1 with depth 5 and 100 bins, another with depth 5 and 300 bins etc.

5.3 Initial Findings

Best Hyperparameters for Tree 1

param	value
cacheNodeIds	False
checkpointInterval	10
featuresCol	features
impurity	gini
labelCol	subscription_type
maxBins	300
maxDepth	5
maxMemoryInMB	256
minInfoGain	0.0
minInstancesPerNode	1
predictionCol	prediction
probabilityCol	probability
rawPredictionCol	rawPrediction
seed	956191873026065186

Feature Importances for Tree 1

feature	importance
duration	0.834542485493
dow	0.117450985255
hod	0.0249854897286
end_station_id	0.0121902410478
start_station_id	0.0100493202024
precipitation_inches	0.000781478273574
mean_humidity	0.0
mean_temperature_f	0.0
month	0.0

Accuracy on Testing Data vs Training Data

```
test accuracy:  0.9030566511294371
train accuracy: 0.9012710953900065
```

This first run gave us a tree capable of determining whether a ride was taken by a customer or a subscriber with 90% accuracy. This was far above that of a random guesser, indicating that there is in fact significant difference between the behavior of one-time customers and that of subscribers. This 15 percentage point increase in accuracy equates to a 20% increase in the number of accurate classifications and a 60% decrease in the number of wrong classifications over randomly guessing.

However, in order to be sure that this tree was not overfitted to our particular dataset and might be applicable outside of it, I withheld 10% of the data as part of a test set. The model performed with similar accuracy on both the training and withheld testing set, indicating that it is not overfitted to our dataset and will be useful with new data points.

Insight into the behavior of the riders and how the tree classifies them is found in the feature importances. From looking at them, we see that the vast majority of the decision on this classification comes from the duration of a trip. From a quick glance at the average ride times for customers vs subscribers, we find that customers take an average of almost 6 times as long as subscribers, indicating that they are likely riding for pleasure, while subscribers are using the bicycles to get from point to point. A quick look at the most popular trips for customers confirms this, with 3 of the 6 most popular trips starting and ending at the same station, while none of the 20 most popular trips for subscribers start and end at the same station.

Most Popular Trips and Durations for Customers

avg(duration)	start_station_name	end_station_name
2277.2373569582874	Harry Bridges Plaza (Ferry Building)	Embarcadero at Sansome
4830.986984815619	Embarcadero at Sansome	Embarcadero at Sansome
7012.690811535882	Harry Bridges Plaza (Ferry Building)	Harry Bridges Plaza (Ferry Building)
1591.8852061438965	Embarcadero at Sansome	Harry Bridges Plaza (Ferry Building)
2617.575546719682	Embarcadero at Vallejo	Embarcadero at Sansome
9761.536992840096	University and Emerson	University and Emerson
3261.5261538461536	Harry Bridges Plaza (Ferry Building)	Embarcadero at Vallejo
2577.3153724247227	Steuart at Market	Embarcadero at Sansome
9894.771381578947	Market at 4th	Market at 4th
11246.759098786828	Powell at Post (Union Square)	Powell at Post (Union Square)
2225.5421052631577	Embarcadero at Sansome	Market at 4th
3476.451499118166	Market at 4th	Embarcadero at Sansome
1169.0115384615385	2nd at Townsend	Harry Bridges Plaza (Ferry Building)
5465.2368932038835	Embarcadero at Vallejo	Embarcadero at Vallejo
3484.506	Embarcadero at Bryant	Embarcadero at Sansome
11574.085889570551	Powell Street BART	Powell Street BART
3365.7418032786886	Embarcadero at Sansome	Grant Avenue at Columbus Avenue
3011.952577319588	Market at Sansome	Embarcadero at Sansome
2274.7	Market at 4th	Harry Bridges Plaza (Ferry Building)
1644.66452991453	Embarcadero at Sansome	Steuart at Market

Most Popular Trips and Durations for Subscribers

avg(duration)	start_station_name	end_station_name
304.033553719008247	San Francisco Caltrain 2 (330 Townsend)	Townsend at 7th
261.9496299342105	Townsend at 7th	San Francisco Caltrain (Townsend at 4th)
512.5369298448715	2nd at Townsend	Harry Bridges Plaza (Ferry Building)
601.6602139582272	Harry Bridges Plaza (Ferry Building)	2nd at Townsend
641.7166448230669	Embarcadero at Folsom	San Francisco Caltrain (Townsend at 4th)
406.83478032096815	Embarcadero at Sansome	Steuart at Market
541.4378698224853	Steuart at Market	2nd at Townsend
513.4485210466439	2nd at South Park	Market at Sansome
470.2312590448625	Harry Bridges Plaza (Ferry Building)	Embarcadero at Sansome
649.9483309143686	Temporary Transbay Terminal (Howard at Beale)	San Francisco Caltrain (Townsend at 4th)
699.2162083089526	San Francisco Caltrain (Townsend at 4th)	Temporary Transbay Terminal (Howard at Beale)
272.1467101363367	Townsend at 7th	San Francisco Caltrain 2 (330 Townsend)
549.0951666166317	San Francisco Caltrain 2 (330 Townsend)	Powell Street BART
413.7670488852447	Market at Sansome	2nd at South Park
715.708918753818	Steuart at Market	San Francisco Caltrain (Townsend at 4th)
703.198574969021	San Francisco Caltrain (Townsend at 4th)	Harry Bridges Plaza (Ferry Building)
656.9465623981753	Market at 10th	San Francisco Caltrain (Townsend at 4th)
629.1047331319235	San Francisco Caltrain (Townsend at 4th)	Embarcadero at Folsom
388.9381692574023	San Francisco Caltrain 2 (330 Townsend)	5th at Howard
454.1010137581463	Powell Street BART	San Francisco Caltrain 2 (330 Townsend)

Interestingly, despite the fact that only one of the most popular 10 trips (Harry Bridges Plaza to Embarcadero at Sansome) is shared between customers and subscribers, starting and ending station were each given less than 2% of the weighting in deciding if a trip was taken by a subscriber or by a customer.

5.4 Feature Combination

I was surprised by how little starting and ending locations played into the tree's decision, and wanted a better way to encode the idea of a trip into a data point useable by the classification algorithm. The logic behind this is that since certain routes might be popular for tourists and others might be busy commuter paths, the idea of a route should be included in the decision. To establish the idea of a route as a journey between 2 endpoints, I encoded the start and end of a trip as a single feature to force the algorithm to look at start and end as paired data, instead of as separate variables. Since the start and end were identified with integers less than 100, a simple map over the two columns into an addition and multiplication of the values ensured a new unique identifier for every trip.

The results from this tree were, surprisingly, not any better or worse than the first model with similar accuracy. The ending station still ended up above the start location in feature importance, and the "trip" column encompassing both ended up between the two. This indicates that the decision tree did not have any need for the new field, and that that it had already picked up on any correlation that existed between the starting and ending position. This makes sense considering the way the algorithm works. As it partitions data based on a classifier, it treats each sub-tree as another set to classify, thus allowing it to pick up on inter-dependence of variables such as start and end location.

Feature Importances for Tree With Combined Start and End Points (zipped_travel)

feature	importance
duration	0.843158237583
dow	0.118376259974
hod	0.0327543137693
end_station_id	0.00266962317413
zipped_travel	0.00146838903985
precipitation_inches	0.00118460289957
start_station_id	0.00038857356059
mean_humidity	0.0
mean_temperature_f	0.0
month	0.0

Test vs Train Accuracy With Combined Fields

```
test accuracy:  0.9006183282679323
train accuracy: 0.900114278143063
```

5.5 Random Decision Forest

In a final attempt to increase the accuracy of the model, I ran the data through a decision tree forest instead of a single tree. This method creates multiple different trees and relies on the "wisdom of the crowd" approach: with multiple trees attempting to classify the data, the times a single tree might be wrong about a data point, we hope that its classification of a point ends up in the minority and that other, correct, trees drown out its decision. Using a random forest with Spark was as easy as swapping out the decision tree classifier for a random forest classifier in the pipeline.

Ultimately, the random decision forest was unable to produce a significantly higher accuracy than the previous models, hitting an accuracy of 90.2%, as compared to the 90.0% reached by the other two models. This is only a 2% decrease in error (.2 percentage points), and is not worth it for the added expense of having to train and run multiple trees instead of a single tree. Also notable is that no feature had an importance of 0 in this model, but many were close to 0. This happens when only a few trees in the ensemble assign a feature any importance.

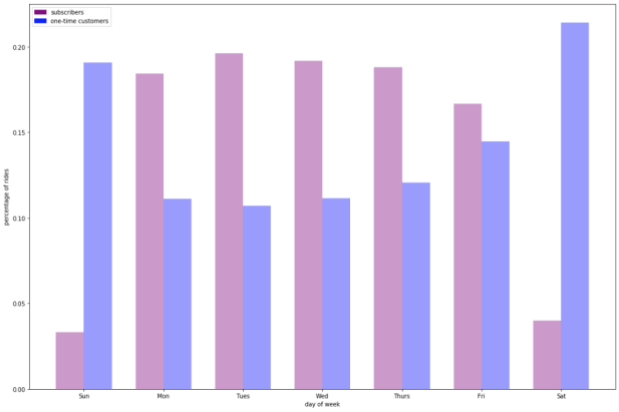
Feature Importances for Random Forest

feature	importance
duration	0.704155565276
dow	0.221362403209
hod	0.0358751265074
end_station_id	0.0175903231818
zipped_travel	0.00687027182159
start_station_id	0.00641836693469
mean_temperature_f	0.00260448948399
mean_humidity	0.00174652723307
month	0.00171099220739
precipitation_inches	0.00166593414512

6. Interpretation and Exploration of Results

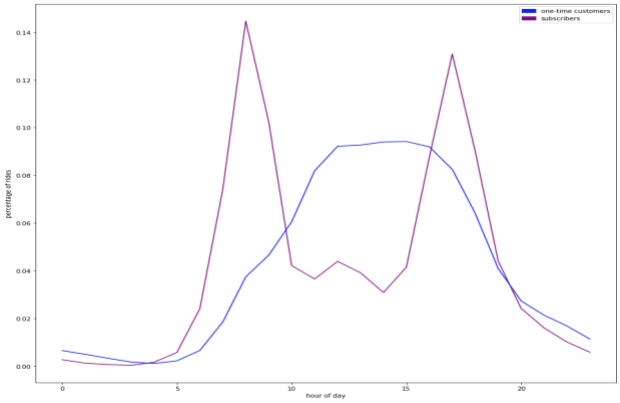
With the knowledge of which features end up differentiating rides and which ones have little to no impact, it makes sense to attempt to make sense of the differences and look for insight into the data in the most important features. Having already examined possible reasons for the large gap in duration between customer and subscriber, this section will focus on the impact that the day of the week, hour of day, and starting and ending locations of trips.

Percentage of Rides on Each Day of the Week by Customer Type



Looking at preferences for which day of the week people like to rent bikes on, we find a marked preference for weekend rides from customers, and a preference for weekdays for subscribers. This finding reinforces the theory that subscribers tend to use the bike sharing program as a way to commute to and from their work, while customers tend to use the bikes for leisure. This would also explain the decline in subscriber use and increase in customer use on Fridays.

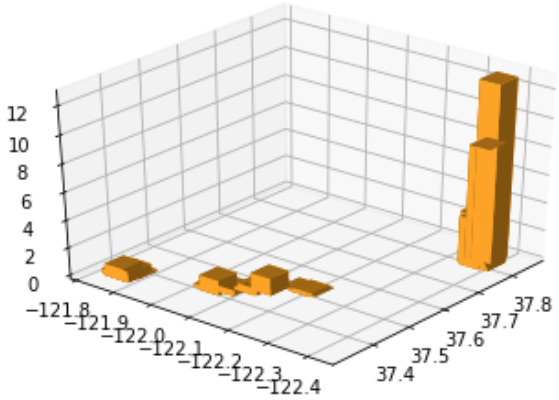
Percentage of Rides by Hour of Day by Customer Type



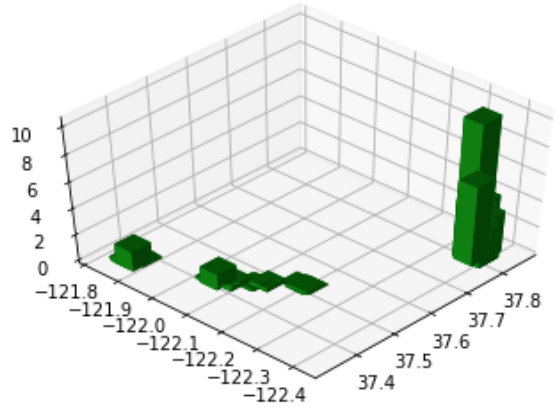
The hourly breakdown of use provides further light on the differences between the ways that customers and subscribers use the service. We see very low usage from both groups in the early morning and late at night, but two distinct peaks around 8AM and 5PM from subscribers. This coincides with typical office commutes and further reinforces the theory that those who subscribe to the service are predominantly using the bicycles to go to and from work. We also find that customer usage hits a plateau between noon and 4PM. Knowing from previous analysis that customers tend to use bicycles on the weekends, we can theorize that this group uses them for leisure on weekend afternoons, and likely just after lunch time.

Ending locations, which factored in to the tree's decisions very little, understandable looked quite similar between customers and subscribers. The plots below show the percentage of rides ending at each station. The latitude and longitude of the station are represented on the bottom axes, and the up-down axis is the percentage of rides ending at that particular station. From a quick, qualitative glance, we can see that there is little difference between the two, with the most popular ending stations for both customers and subscribers being in northeastern part of the city.

Customer Popularity of End Stations



Subscriber Popularity of End Stations



Ultimately, from data extracted about these trips, we can conclude that subscribers tend to use the bike share to get to and from work, while customers use the bikes for leisure. Therefore, the company running the bike share could employ separate marketing techniques to capture those looking to use a bike for fun and those looking to use them to make a commute.