On Your Bike - building a classifier for Capital Bikeshare data in Washington DC 2017

Tomack Gilmore
March 20, 2018

This report is an analysis of customer data from the Capital Bikeshare scheme in Washington DC during the whole of 2017. The data is available from the company's website:

`https://www.capitalbikeshare.com/system-data`

The files we will consider in particular are the following:

1. 2017-q1_trip_history_data.csv
2. 2017-q2_trip_history_data.csv
3. 2017-q3_trip_history_data.csv
4. 2017-q4_trip_history_data.csv

which contain information for each quarter of 2017.

The goal of this report is to identify whether there is a difference between those users who are registered users of the bikeshare system and those who are casual, non-registered users. After analysing the data we construct a classifier using machine learning techniques that will identify whether a ride was made by a registered or casual user.

1 The Data

Capital Bikeshare is a bike sharing scheme that serves the following areas in the US: Washington, D.C.; Arlington County, Virginia; the city of Alexandria, Virginia; Montgomery County, Virginia; and Maryland and Fairfax County, Virginia. The system consists of more than 440 stations where users can start or finish their ride, with over 3700 bicycles distributed across the network.

Users can be registered members or casual users, and each bike can be used for a time limit of 30 minutes, with a grace period of 15 minutes if a user tries to return a bike to a station that is full.

The dataset consists of records of every single journey made during the whole of 2017. There were a total of 3,758,207 journeys made, and for every trip we have the following information:[1]

- The duration of the ride in miliseconds.
- The date and time the ride started.
- The date and time the ride ended.
- The number of the start station.
- The address of the start station.
- The number of the end station.
- The address of the end station.
- The bike number.
- The member type of the ride – whether it was made by a casual or registered user.

Before beginning the analysis proper the dataset needs to be cleaned and tidied – erroneous entries must be removed, as do journeys that begin or end outside the Washington DC area. Also we need to ensure the data types in each column are uniform and consistent. Entries in the 'member type' column are replaced with a 0 if they are casual members, 1 if registered. The 'station name' addresses are also replaced with full addresses that are recognised by the Google directions API (this will come in handy later on).

Once the data has gone through this pre-processing phase[2] we end up with a dataframe (like a spreadsheet) containing a total of 3,206,106 records of journeys, where each record has an entry in each of the following columns:

- Duration (ms) – int64
- Start date – datetime64[ns]
- End date – datetime64[ns]
- Start station number – float64
- Start station – string
- End station number – float64

[1] To load the data call the function 'load_data()' from the module 'bike_functions.py'.

[2] Achieved by calling the function 'clean_and_tidy()' from the module 'bike_functions.py'.

- End station – string
- Member type – int64 (Boolean)
- Duration (mins) – float64

Having cleaned our dataset we can begin a proper analysis of the data.

2 The Analysis

We will first look at some exploratory statistics, just to get a rough picture of what's going on inside our dataset. The average journey time was just over 19 minutes, however the median journey time was much lower (just under 12 minutes) and 75% of journeys took just over 20 minutes. This indicates that the mean is being distorted by a few journeys that last for a long time – indeed, the longest journey taken lasted 1439 minutes (nearly 24 hours). This data is displayed in the box and whisker plot below, where outliers have been excluded (but not forgotten).

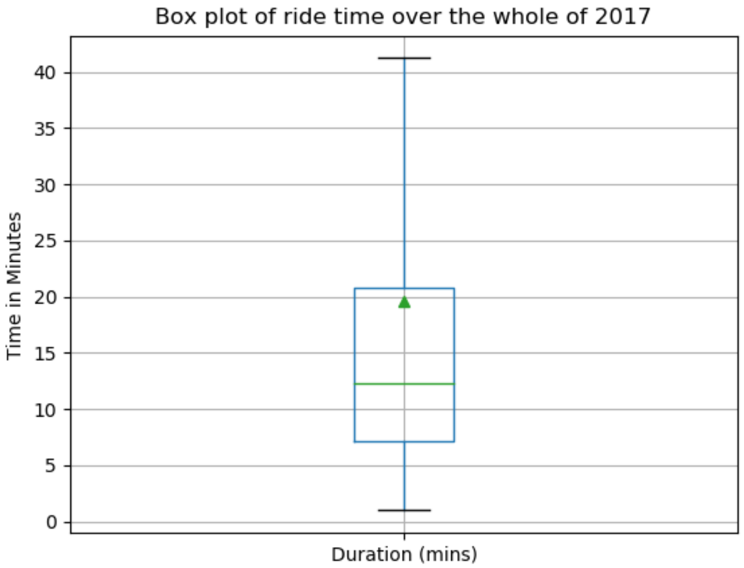


Figure 2.1: Box plot of ride times, where the green triangle indicates the mean.

The types of journeys that users made were, on the whole, trips from A to B, however 3.8% of bikes were taken from and then returned to the same station. We can conclude from these initial observations that the majority of rides lasted less than the 30 + 15 minute limit and the vast majority of rides were between distinct start and end stations.

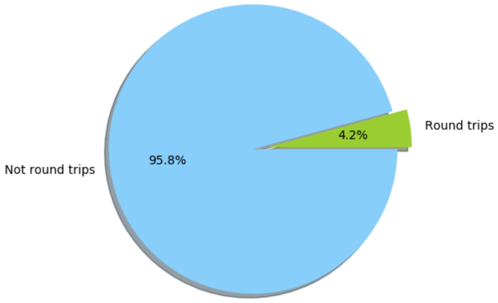
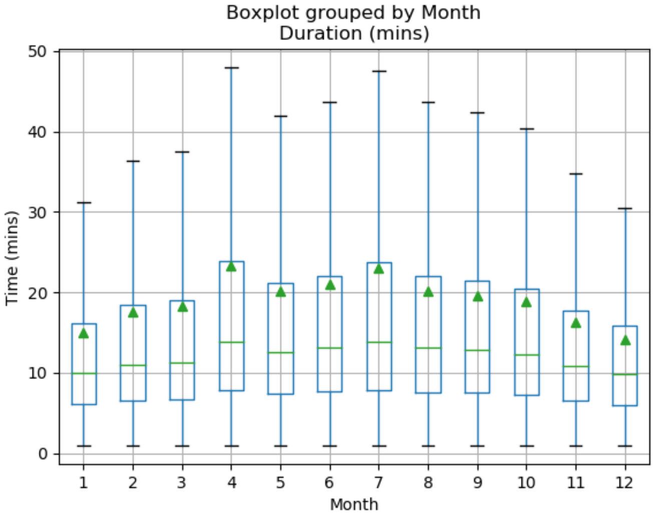


Figure 2.2: Proportion of total journeys that were round trips.

We can take a brief look at the average amount of time spent on journeys on a monthly basis across the year below. The time spent on bikes increases during the summer months, as we may expect.



Let's now dig a little deeper into the behaviour of casual and registered users. The box plot (Figure 2.3) shows the ride times for casual and registered users. It is quite clear from this plot that registered users spend far less time on the bikes, and the amount of time spent per journey also varies a lot less than for casual users. The mean journey time is also much closer to the median for registered users, implying that the outliers – those journeys that were exceptionally long – are perhaps more often due to casual users than registered ones.

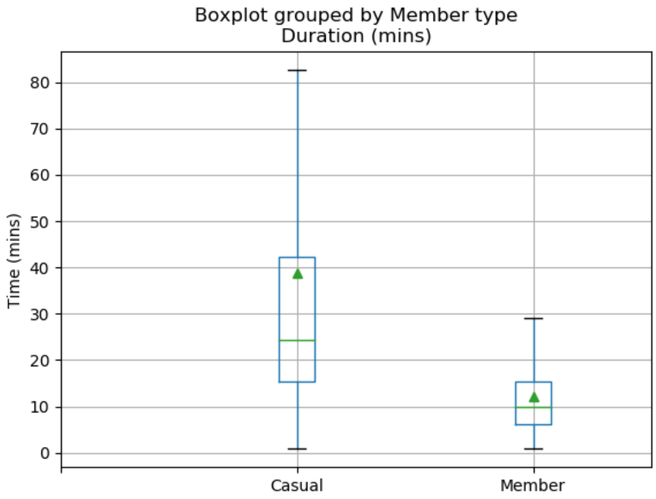


Figure 2.3: Box plot of ride times of registered/casual users.

We can also look at the usage rates on a monthly basis for casual/registered users. Registered users use the bikes far more consistently across the year.

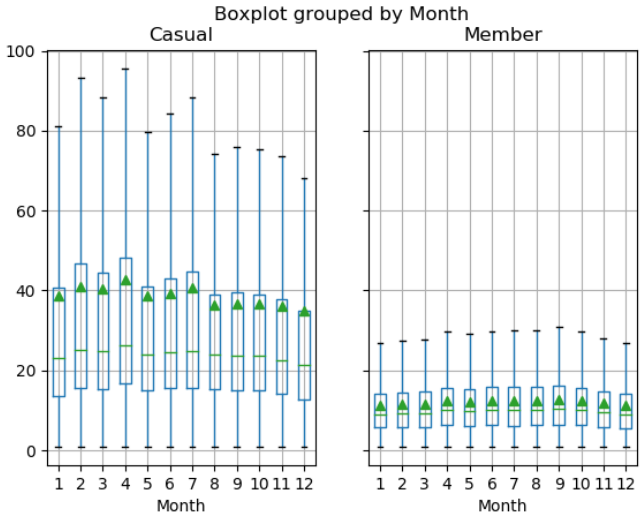
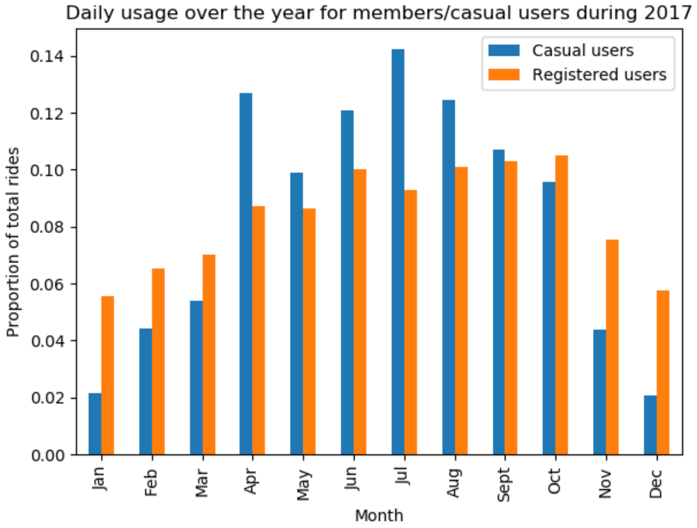


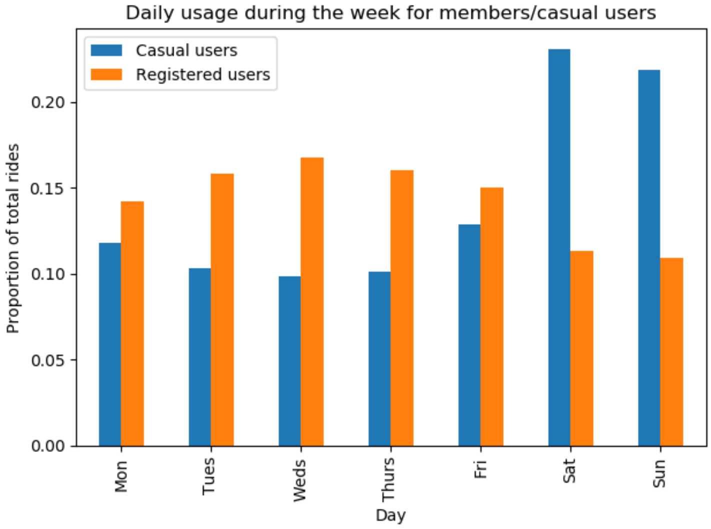
Figure 2.4: Box plot of ride times of registered/casual users.

Let’s now look at the proportion of rides that took place each day for the whole of 2017 for the two groups.

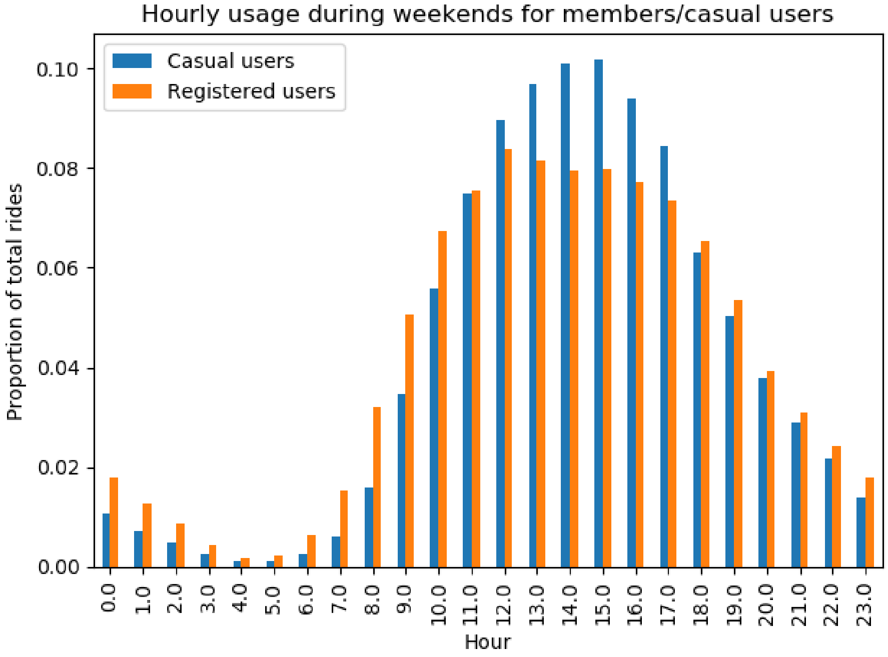


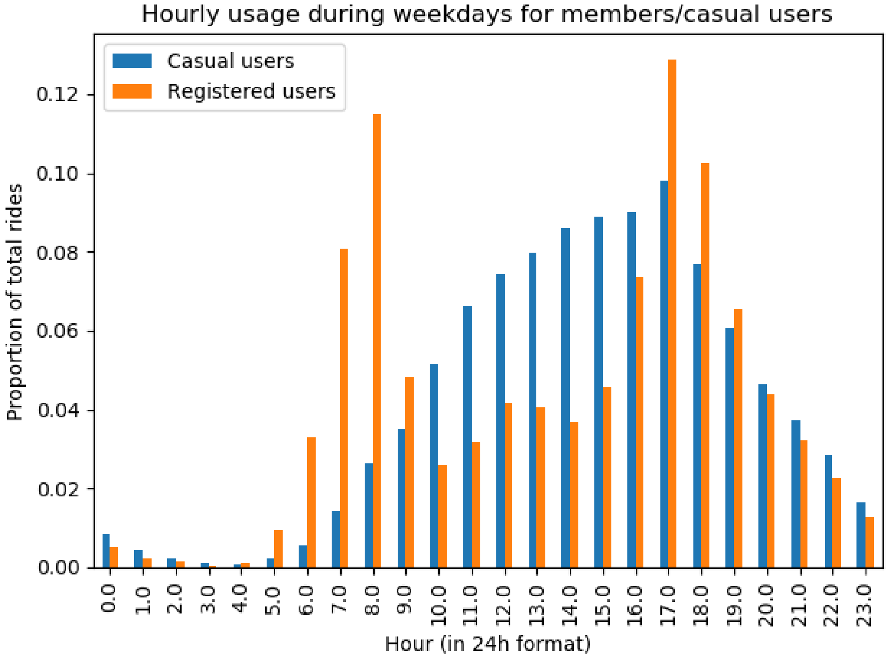
It is clear from this graph that overall the scheme is used more in the summer, however casual users ride more than registered ones during spring and summer, and use the scheme a lot less than registered users during autumn and winter.

What about during the week? Is there a clear difference in usage of the scheme on a weekly basis? Consider the following graph.



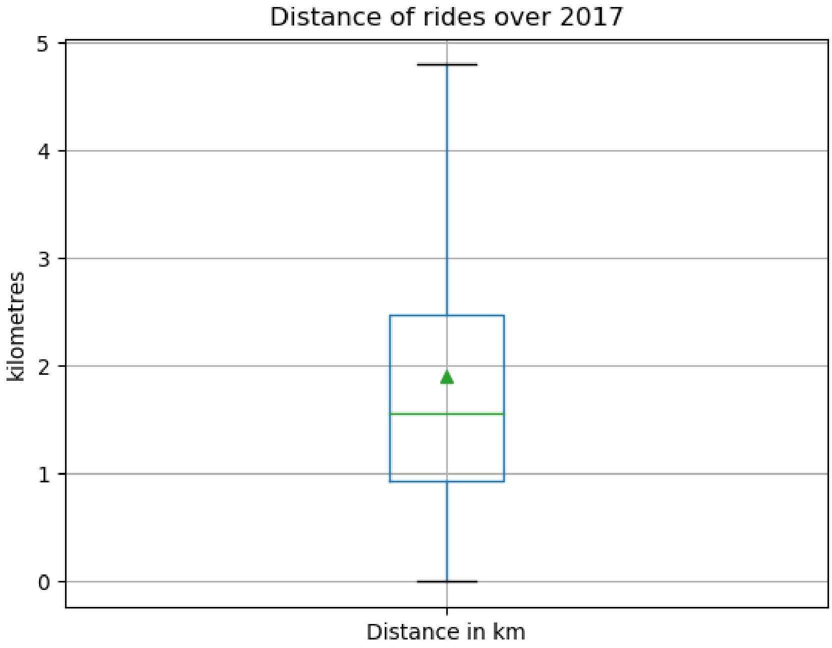
Registered users favour the scheme during the week, while casual users make use of it far more on the weekend. With that in mind we can take a look at the hourly usage rate on a weekday or weekend below.





It looks as though registered users use the bikes for commuting during the week, since we can see peaks between 7-9 am and 4-7 pm. Casual users, on the other hand, might be more likely to use a bike in the evening to get home form work, but certainly not so much in the morning. We can conclude that the month, the day of the week, and the time of day all seem to be useful indicators for whether a user is registered or not.

What about the distance that users travel on the bikes? It's possible that this might also be a useful indicator of whether a ride was made by a registered or casual user. In order to find a crude estimate of the distance travelled during each journey we use the Google geocoding API to find the latitude and longitude of each bike station, and then apply the Haversine formula[3] to uncover the distance as the crow flies between stations.[4] The figure below shows the distance travelled per ride during the whole of 2017.

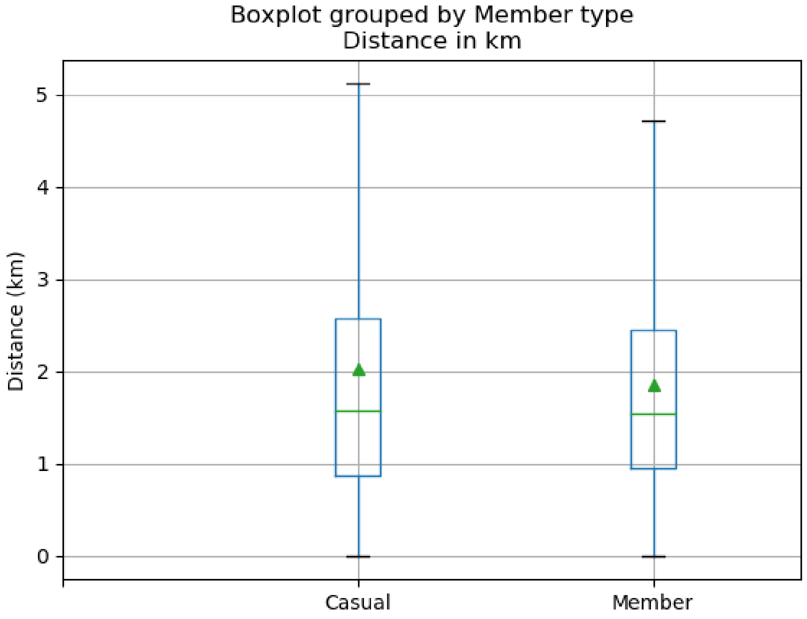


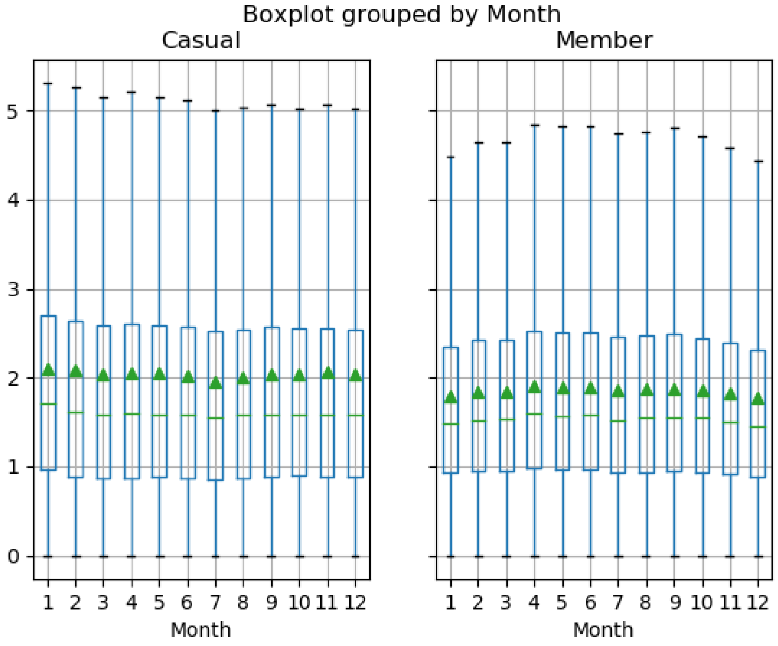
The figure at the top of the following page shows the distances travelled by registered and casual users. It seems there is little difference in the ride distances travelled by users – the average distance travelled is sightly lower for registered users, as is the upper quartile and the upper whisker, but not by much. Maybe there is a more pronounced difference on a monthly basis? The figure at the bottom of the following page shows the ride distances per month across the year for casual and registered users. It looks as though across the board the distance travelled is slightly lower for registered users, and it can be shown that there is a significant statistical difference between the two user types.[5]

[3] https://en.wikipedia.org/wiki/Haversine_formula

[4] This is achieved by calling the function 'get_lat_long()' from 'bike_functions.py'.

[5] This can be seen by calling the function 'ttesting()' from 'bike_functions.py'.





Further information we can extract from the data are the locations that registered and casual users travel to on their bikes. By digging into the data we see that the favoured start and end stations differ between casual and registered users. The top five journeys made by casual users

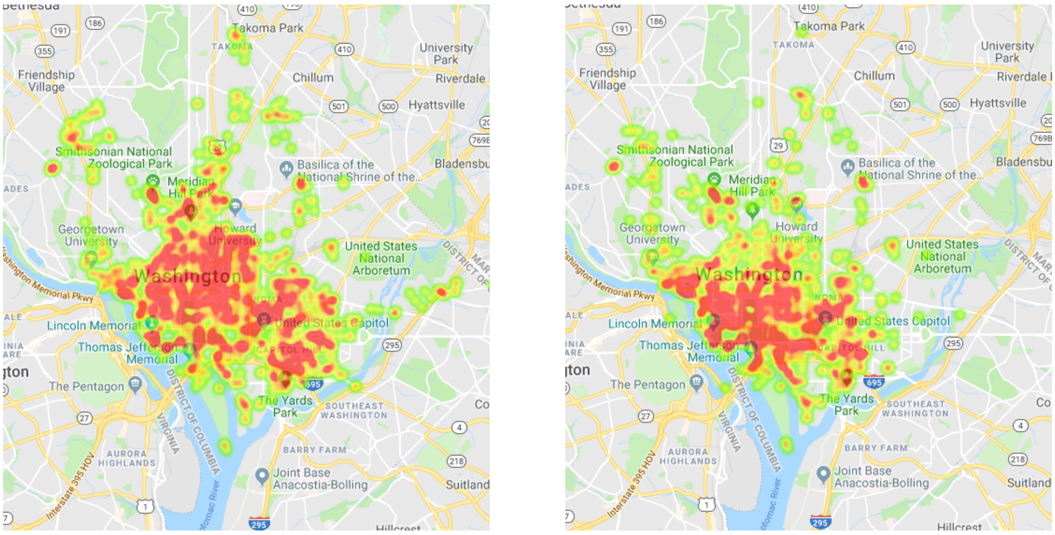


Figure 2.5: Popular areas with registered users (left) and casual ones (right).

are in the following table:

Start station		End station
Jefferson Dr SW & 14th St SW	→	Jefferson Dr SW & 14th St SW
1000 Jefferson Dr SW	→	1000 Jefferson Dr SW
Lincoln Memorial	→	701 E Basin Dr SW
Jefferson Dr SW & 14th St SW	→	Lincoln Memorial
Lincoln Memorial	→	Lincoln Memorial

Casual users appear to favour round trips, with three of the top five rides starting and ending at the same bike station. The top five journeys made by registered users, however, were between different stations:

Start station		End station
701 Pennsylvania Ave SE	→	East Capitol St NE & 13th St NE
Adams Mill Rd NW & Columbia Rd NW	→	Woodley Pl NW & Calvert St NW
East Capitol St NE & 13th St NE	→	701 Pennsylvania Ave SE
Woodley Pl NW & Calvert St NW	→	Adams Mill Rd NW & Columbia Rd NW
4th St SW & M St SW	→	E St SW & 4th St SW

By using the Google Directions API it is possible to find the top 2000 routes taken by both casual and registered users. The most popular streets frequented by each group can be found in the figure at the top of this page[6] – clearly there is a distinction between the journeys made by each

[6] These are in fact interactive html files, open the files 'MostPopularStreetsHeatMap-0.html' and 'MostPopularStreetsHeatMap-1.html' in a browser to zoom in and out and look around.

type of user.

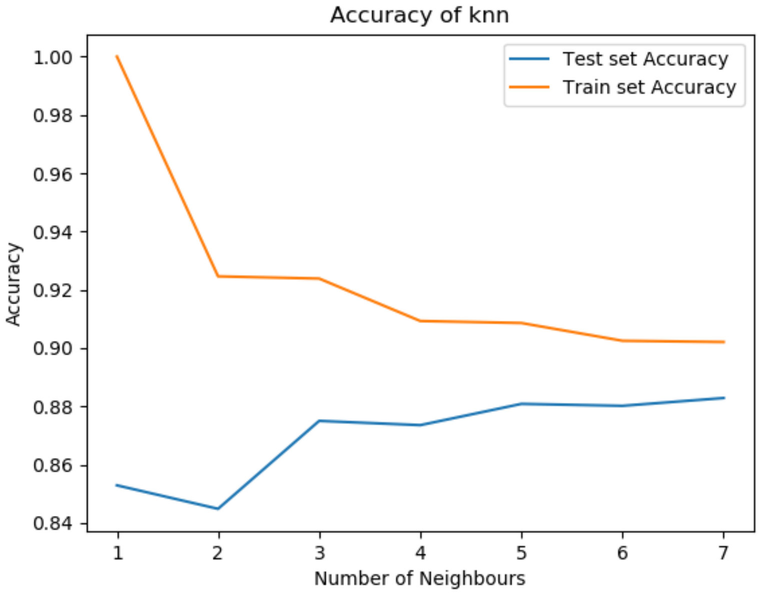
One further benefit of using the Google Directions API is that we can also extract the distance of each journey according to google, which is far more accurate than the distance calculated before. We can also extract the predicted time that the journey will take according to Google, allowing us to benchmark the time each journey took and calculate an estimate of the average speed of each journey. Note the caveat – we calculate these values based on the assumption that each user will take the shortest path according to Google maps.

3 Building a classifier

We now construct a *k-nearest neighbours* classifier that will learn to classify whether a ride was made by a registered or casual user. For each ride made between stations that feature in the top 200 journeys for either casual or registered users we feed the following information into our algorithm:

Start station number; Average speed m/s; Benchmarked time (seconds); GoogleDistance;
Member type; End station number; Month; Day; Hour.

Calling the function 'knn_test(krange)' from 'bike_functions.py' with $krange = 8$ yields the following plot showing how accurate a k-nearest neighbours algorithm is at classifying rides as being made by casual or registered users, where k runs from 1 to 7. Clearly with 5 nearest neighbours such a classifier is correct just over 88% of the time.


Accuracy of knn
Test set Accuracy
Train set Accuracy
Accuracy
Number of Neighbours