Zachary Billey
May 29, 2018

Capstone project

Problem definition

I conducted this analysis from the standpoint of a cab driver trying to increase profit. The central question is then: where does a cabdriver want to be to get the highest paying rides? Can this dataset be used to find "good" places for a cabdriver to get fares?

Description of data

I used the 1% sample of the New York Taxicab data from 2013 from the course website:
`https://canvas.uw.edu/courses/1188735/files/47572017?module_item_id=8236949`

It is a subset of the full dataset for 2013. Presuming it is from the data gathered by the NYC Taxi and Limousine Commission. The full data can be accessed here:
`http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml`

The data set contains two .csv files:
The `NYC_Taxi_2013_One_Percent_Trip.csv` file contain the columns

- medallion
- hack_license
- vendor_id
- rate_code
- store_and_fwd_flag
- pickup_datetime
- dropoff_datetime
- passenger_count
- trip_time_in_secs
- trip_distance
- pickup_longitude
- pickup_latitude
- dropoff_longitude
- dropoff_latitude

The `NYC_Taxi_2013_One_Percent_Fare.csv` file contain the columns:

- medallion
- hack_license
- vendor_id
- pickup_datetime
- payment_type
- fare_amount
- surcharge
- mta_tax
- tolls_amount
- total_amount
- tip_amount

Feature Engineering

The first question: how valuable is a given taxi ride? A long ride that only pays only twice as much as three much shorter rides that could be accomplished in the same time span is going to be worse. So I want a feature that captures the rate of income. A simple measure of money and time is easy to calculate.

$$\frac{\text{Fare} + \text{Tip}}{\text{Trip Time}}$$

However, this ignores a few things. First off, gas and vehicle maintenance costs money. This is often done as a cost per mile. So one adjustment is to subtract mileage times a per mile cost from the money made on the trip. The second major thing is accounting for the time waiting for the next fare or driving to pick it up. This can be roughly accounted for by adding some additional chunk of time to each trip to represent that. This penalizes short enough trips where people getting in/out of the taxi and waiting for the next fare represent a significant portion of the driver's time.

$$\frac{\text{Fare} + \text{Tip} - \text{Miles} \times \text{Cost Per Mile}}{\text{Trip Time} + \text{Added Time}}$$

The cost per mile can be calculated a number of ways. I used an estimate from AAA for a mid-sized sedan at 52.9 cents/mile.[1] Another option is to use the IRS's business mileage deduction rate which was 56.5 cents/mile in 2013.[2] However, the two rates are fairly similar, so it shouldn't make a huge difference.

The additional time is harder to find information for. I picked 5 minutes (300 seconds) arbitrarily. This is a parameter that could be varied to see how much it affects the results.

Also, since I want to find when and where a taxi cab driver wants to be, I binned the pickup times to hours of the day (0-23).

Ultimately the goal of this analysis will be to find places/times where you can do better than city average on this adjusted income feature.

[1] `https://newsroom.aaa.com/tag/driving-cost-per-mile/`

[2] `https://www.irs.gov/tax-professionals/standard-mileage-rates`

Data exploration and cleanup

Figure 1 shows what the adjusted hourly income of the city looks like by pickup coordinate. A few things stand out. There are some very high hourly income rides in New Jersey but they seem sparse. LaGuardia airport is pretty obvious as an island of higher fares in the Northeast, and there appears to be a few high value pickups along the Long Island expressway. Initially, there appears to be a lot of really local variation in values. However, that may just indicate bad data wildly shifting those small points.

So I cleaned up the data. First, there were several points that appeared to have latitude/longitude flipped, so I took any locations that indicated pickup South of the equator and flipped their pickup latitude and longitude coordinates. I then windowed the trip time and the trip distance values to exclude unreasonable points. I chose to include trips of lengths between 0.01 and 200 miles and durations of 1 minute to 10 hours. I also only looked at data points between 40.65N and 40.84N and 74.05W and 73.83 W. This excludes places that are part of New York City, especially Staten Island, but the data set seemed rather sparse outside of this region.

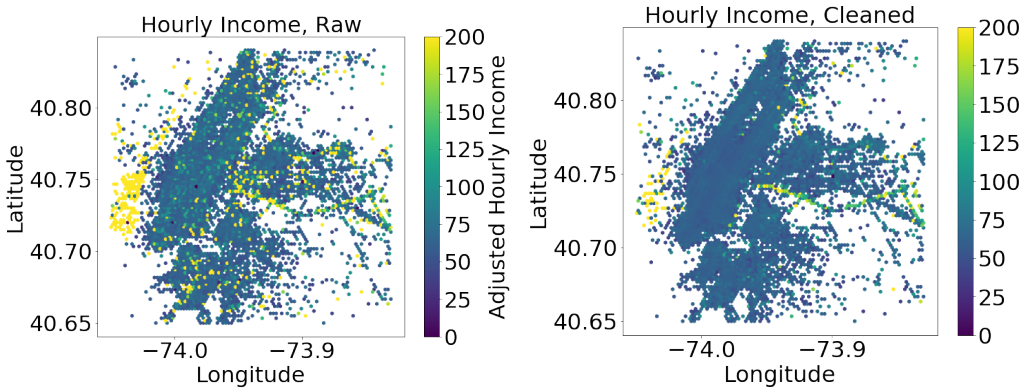

Figure 1: Initial hexplots based on the adjusted hourly income feature, looking at the mean adjusted hourly income in each cell. Coordinates are by pickup location. The right plot is after cleaning. Much of the spikiness was removed by windowing the trip time and trip mileage values

Model Selection

Throw a Gradient Boosted Classifier at it

While my original intent was to cluster the city into zones, and then look at them that way, I felt that a good first approach was to put the pickup coordinate data into a gradient boosted classifier, trying to predict "high hourly income" and "low hourly income" regions. I tried a range of thresholds for the cutoff between "high" and "low." Using a value of 60 seemed to produce a plot that was reasonably balanced between "high value" regions and "low value" regions.

However, Figure 2 reveals that most of the regions counted as "high value" have low density. Nor do the predictions seem to really map well at all to the map of high mean hourly income regions. Given the later results, I expect part of this is that using pickup locations only is a poor predictor of whether a ride will be high profit or not. With this type of classifier, it seemed like

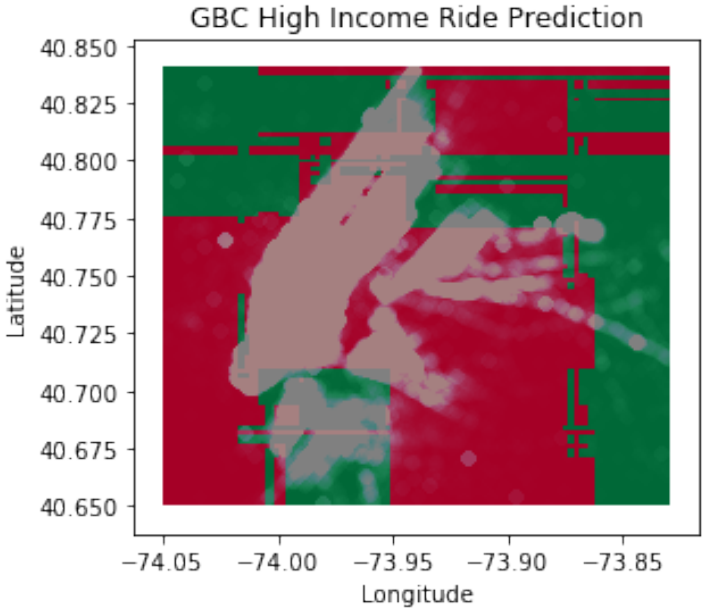

Figure 2: A plot of the city with pickup locations marked in gray and the GBC predictions marked green for high value and red for low value. It does not paint a very coherent picture.

dealing with low density regions would still be a problem, even with a better fit, so I decided not to continue investigating this model further.

Kernel Density Estimation

I want a map that identifies areas with rides with high hourly income, but also takes density into account so I am not suggesting a region that has one ride in the data set that just happened to be very well paid. It would be very handy to rate a model continuously by pickup coordinate. While it is not an exact analogy to this situation, kernel density estimators are used to turn a discrete sampling of data into a continuous probability distribution. In the same way, I wish to turn a discrete distribution of taxicab rides into a continuous score of the profitability of the city.

One of the obstacles here is that I want to be able to weight the points by their adjusted hourly income, but the sklearn and SciPy KDE functions do not support weighting. Fortunately, a search of stack overflow provided a modified version of the SciPy code by Till Hoffman[3] that could use weighting for the input points.

I used a Gaussian kernel with a bandwidth of 0.08 degrees on the latitude and longitude pickup data, weighting by adjusted hourly income. Then I identified the highest density regions by setting a cutoff threshold for the kernel density function, 0.5 of the maximum value, and counting those areas as "high value" areas. I then compared the mean of the hourly adjusted income for rides in a test set with the mean of rides in the entire city.

[3] https://stackoverflow.com/questions/27623919/weighted-gaussian-kernel-density-estimation-in-python#27623920

Unfortunately, the city-wide mean was $62/hr as compared with $58/hr for my high value regions! Now, this is very likely because the highest hourly income rides are from very low ride density areas. For example, Jersey City and the expressway both showed high hourly income in the exploration set but don't appear to have a very high ride density. This is maybe understandable. But I want to do better.

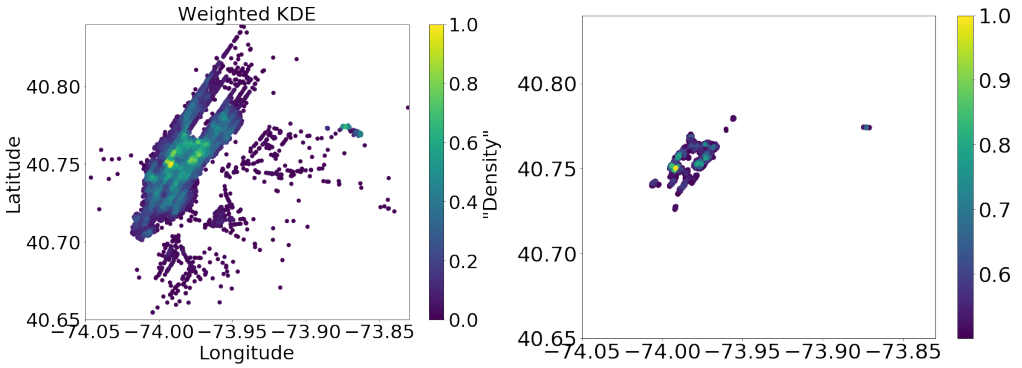

Figure 3

I then tried to make the adjusted hourly income more important. I tried using a weight that was a higher power of adjusted hourly income. Using the square of the adjusted hourly income did shift the regions identified. However the mean hourly income of the regions was $60, no improvement over the straight weighting model. Higher powers showed even worse predictive performance.

This again may be because using coordinates alone is insufficient to do a very good job of predicting high value rides. Time turns out to be necessary.

Clustering and Ranking

The model I finally settled on was to first use a clustering algorithm to separate the city into zones. I then calculated the mean hourly income with respect to the zone and the hour of the day. I used that to predict the best location for getting high hourly income fares by ranking the zones by their mean adjusted hourly income.

For the clustering algorithm, I used the sklearn implementation of the mean shift algorithm since it fulfilled the criteria of dividing even fairly evenly distributed data up into clusters, like the taxi rides in Manhattan, and also had a built in function to predict the cluster labelf of new points. This is important for testing the model.

To try and make the zones map to more high-income areas, I trained the clustering algorithm only on rides with an adjusted hourly income higher than $90 /hr. I used the same bandwidth that produced good results in the KDE algorithm: 0.08 degrees.

After clustering the points into zones, I broke the data set up by hour of the day (0-23) and then calculated the mean adjusted hourly income for each zone. Then I discarded zones that had too few rides, or too low a ride "density" with the zone area calculated by square of the average distance from the cluster center. Then finally I ranked the zones by mean adjusted hourly income. At each hour, the zone with the highest mean hourly income was predicted as the best zone

Model Results

The model was tested with K-fold cross validation with a K of 8. The mean performance of the top zone by hour was compared with the mean adjusted hourly income of the city by hour, and the city throughout the day.

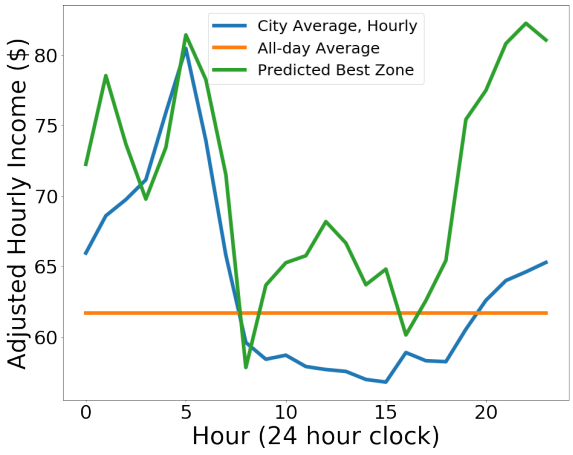

Figure 4

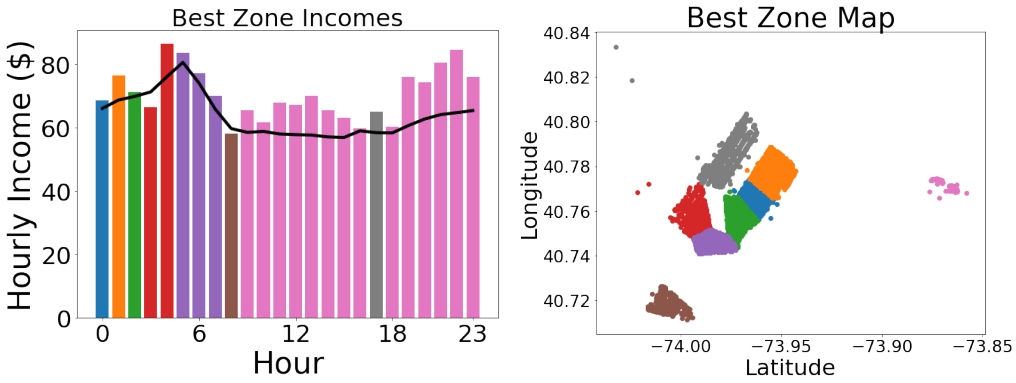

Figure 5

For the first part of the day, midnight to 8:00am, the best zone model actually doesn't do better than the city average. The advantage appears to come from knowing the right time to pickup fares from LaGuardia airport, centering around peak high adjusted income times: noon-3pm and 7pm to midnight.