UberMax
Rahi Punjabi, Seung Hwan Lee
APMA 4990 Final Project
Dorian Goldman

Motivation

Driving for a for-hire vehicle service like Uber or Lyft is often a side job for people looking for multiple streams of income. To increase the revenue potential of being a driver, you want to be sure that you wait near areas with high demand throughout your shifts. Obviously, these areas of optimal revenue potential vary during the day and by day of the week and could also be affected by customer preferences of the service you're working for. We plan to develop a web app that recommends nearby locations for drivers to wait based on the time, day of the week, and current location. The app can also help a potential driver determine which service she or he would like to drive for and what days/times they will work to.

Audience

Current and potential drivers for for-hire vehicle services like Uber or Lyft

Dataset

The dataset we used catalogs all Uber rides taken in New York from April to September 2014. We found it at https://github.com/fivethirtyeight/uber-tlc-foil-response.

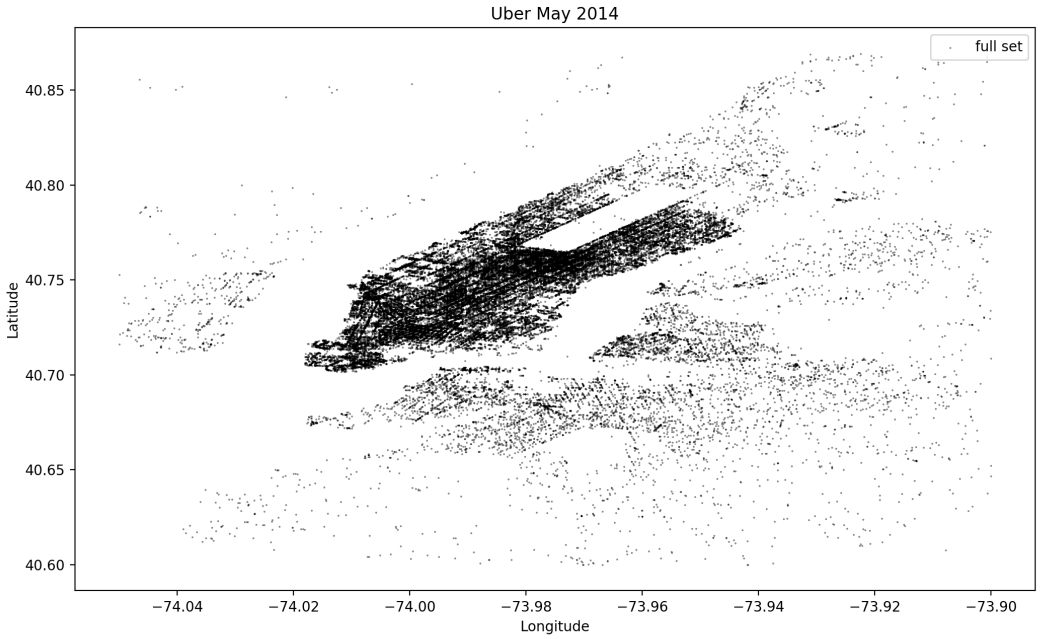

Figure 1: Geographic scatterplot visualization of all the rides taken in May 2014

Project Methodology

To create this web app, clustering and regression algorithms were performed on the dataset.

Part I: K-mean Clustering

The purpose of clustering was to divide New York into regions with high densities of Uber rides for each month. K-means clustering and density-based spatial clustering of applications with noise (DBSCAN) were tested since both clustering algorithms have previously been applied to geographic coordinate data. Since the k-means clustering algorithm generated more geographically realistic clusters, it was used to develop this web app. The k value was set to 50 by testing a range of potential k values.

Here are visualizations of the k-means clustering for the Uber ride data.

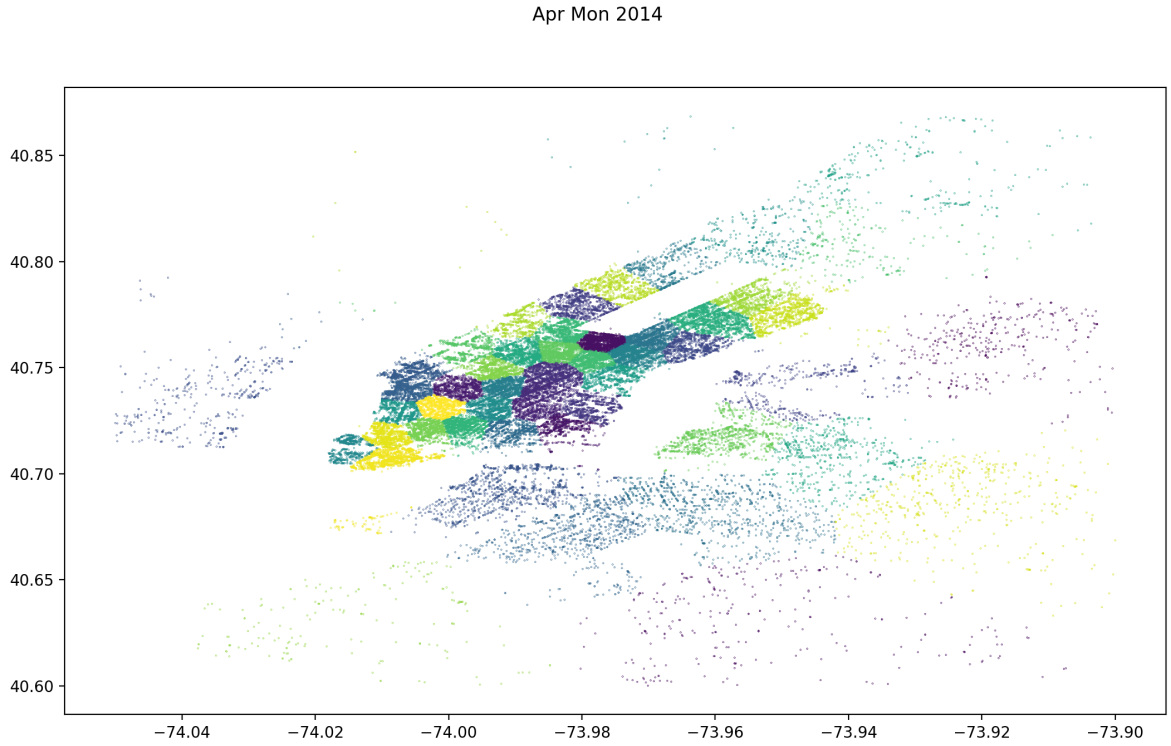

Figure 2: Clusters generated from Uber ride data for all Wednesdays in April 2014

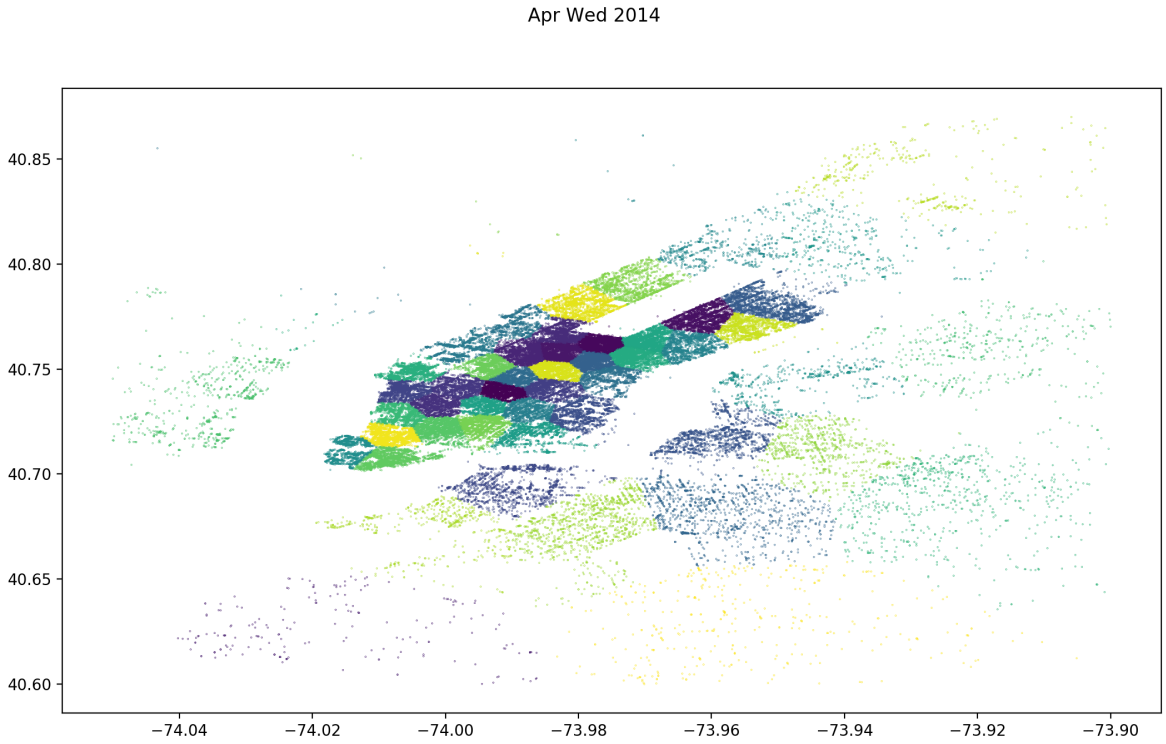

Figure 3: Clusters generated from Uber ride data for all Wednesdays in April 2014

Part II: Random Forest Regression

The purpose of regression was to determine the relationship between hour of the day and demand for Uber rides in New York for each day of the week. The hypothesis was that demand for Uber rides peak during commuting hours (6-9 am and 4-7 pm) daily. Random forest regression was applied to the dataset due to the nonlinear pattern of daily demand for Uber rides.

Initially, we ran random forest regression using the hourly Uber ride demands for each day of the week without separating by month. However, we found that the random forest regression model did not perform well (i.e. suboptimal R^2 values) due to the high variability of the demand data. This is an example of the aggregate demand data for Fridays, which exhibits high variability between hours 15 – 20.

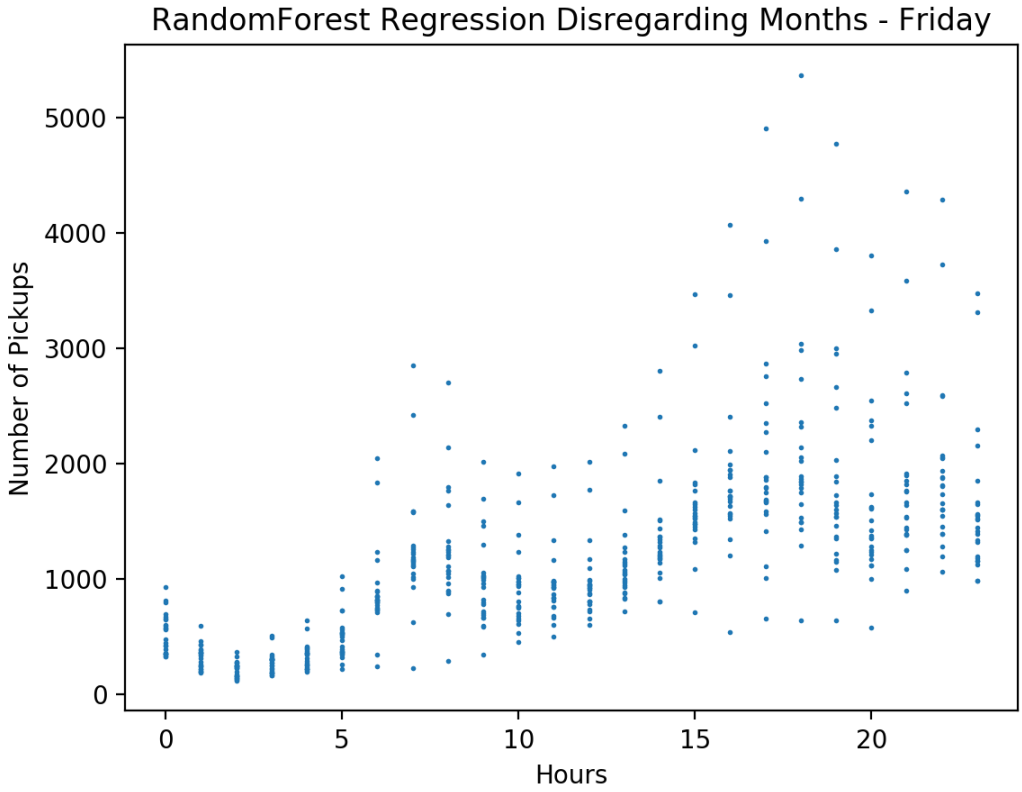

Figure 4: Random forest regression plot for Fridays across all months in dataset

This is because demand for Uber rides exhibits different monthly trends. We believe this seasonality arises from fluctuations in weather and tourism throughout the year.

Therefore, we ran random forest regression using the hourly Uber ride demands for each day-month combination (e.g. Fridays in April). Our random forest regression model performed satisfactorily on testing and training datasets across all day-month combinations. These are examples of the random forest regression plots and all the R^2 values of the training and testing datasets for each month-day combination.

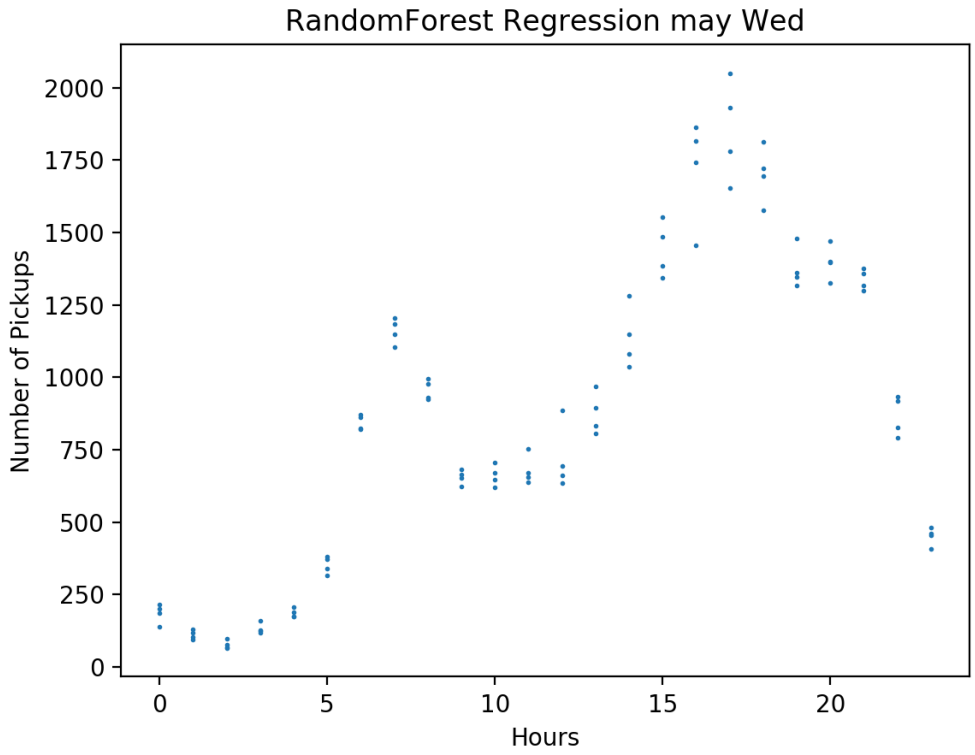

Figure 5: Random forest regression plot for Wednesdays in May

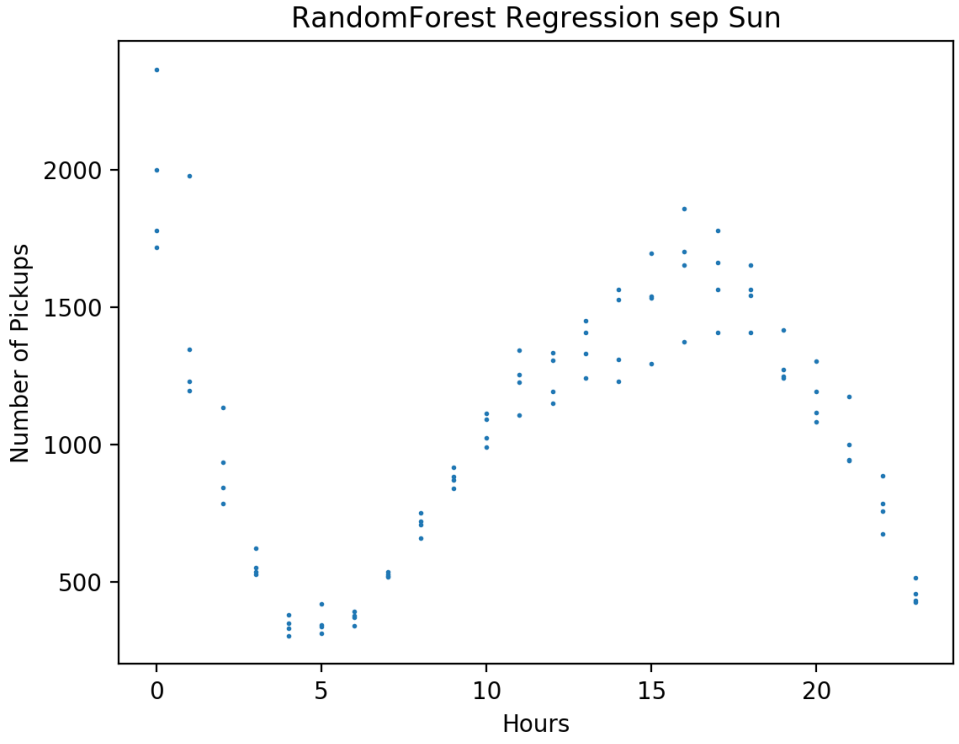

Figure 6: Random forest regression plot for Sundays in September

Month	Day	Training dataset R^2	Testing dataset R^2
April	Mon	0.742105494	0.546022757
April	Tues	0.788081962	0.798011863
April	Wed	0.616815279	0.593264185
April	Thurs	0.971777433	0.925207962
April	Fri	0.844288209	0.710653799
April	Sat	0.841281176	0.653620018
April	Sun	0.83269253	0.692915555
May	Mon	0.730797981	0.644116115
May	Tues	0.944013509	0.912172653
May	Wed	0.979916339	0.975783432
May	Thurs	0.96042089	0.912515802
May	Fri	0.857164793	0.812741967
May	Sat	0.854299452	0.795503943
May	Sun	0.841852836	0.641939716
June	Mon	0.853080895	0.776207251
June	Tues	0.929873903	0.93085543
June	Wed	0.893051215	0.927493491
June	Thurs	0.950197408	0.907794758
June	Fri	0.821231568	0.758383418
June	Sat	0.97610817	0.947649753
June	Sun	0.925869394	0.861633639
July	Mon	0.892634912	0.655666416
July	Tues	0.827292692	0.858161893
July	Wed	0.92659416	0.883121871
July	Thurs	0.833906833	0.860674268
July	Fri	0.721472203	0.357765151
July	Sat	0.565379031	0.576512936
July	Sun	0.712566772	0.276714032
August	Mon	0.971037733	0.930517164
August	Tues	0.943382557	0.801999058
August	Wed	0.973127947	0.927539611
August	Thurs	0.945427954	0.924640709
August	Fri	0.935445953	0.950025634
August	Sat	0.934724621	0.921645011
August	Sun	0.899241129	0.821894562
September	Mon	0.708619015	0.686824669
September	Tues	0.914481412	0.837526715
September	Wed	0.973045412	0.939055978
September	Thurs	0.928906469	0.862190995
September	Fri	0.987540126	0.96420491
September	Sat	0.949997287	0.906819868
September	Sun	0.929772751	0.883446694

Figure 7: R^2 values of the training and testing datasets for each month-day combination

Part III: Front-End Engineering

Our web app was deployed using Amazon Web Services. The user inputs two instances of a location (latitude, longitude coordinates), day of the week, and time of day. A SQL database is queried to extract the regional demand clusters previously generated for the particular day and month. The Euclidean distances from the user's location to the centroid of each cluster are computed and ranked. Subsequently, the ride demands for the N nearest clusters are estimated using the regression models previously developed. The centroids of the clusters with the top 5 estimated ride demands are displayed to the user.

Limitations and Future Work

A limitation of our web app is that it only works for April – September because our dataset was only complete for these six months. We tried to create a random forest regression model using the entire six months of Uber ride data we had but it did not generalize well. This is because each month appears to have a different trend due to seasonality so analyzing the data in aggregate dilutes the monthly trends. If we received Uber ride data for the months of October – March, we could apply the same algorithms and methodology and expand the functionality of this web app to all months of the year.