EECS 4415 Project Report

Taxi vs. Uber in NYC

Mahmoud Alsaeed
York University

Ashkan Moatamed
York University

Koko Nanah Ji
York University

Dong Hoon Lee
York University

1 Abstract

In this project, we aimed to compare New York City transportation methods involving an automobile (i.e., Taxi and Uber). At every stage of the project, we attempted to devise solutions such that they would be easily extendable to data of similar domain from other cities and even countries. The data can be used to answer a variety of useful questions which are not limited to the following:

- Economical Questions: How do the businesses affect each other? Are some becoming obsolete and being pushed out of the industry?
- Hotspots Across Time and Space: Which areas have a higher demand than others and how are they affected by the time of the day?

We used the state-of-the-art big-data technologies such as Apache Spark and HDFS to be able to answer all the above questions and more in an efficient, distributed, and scalable fashion. Furthermore, Spark's Machine Learning libraries were instrumental in performing parts of the analytics and can serve as a basis for future prediction and modeling algorithms.

2 Introduction and Motivation

The project is based on analyzing transportation methods available in every city in the modern world. The focus of the project is on automobile methods not including buses, trains, and subway. Due to New York City's importance, we chose to build our application only to analyze NYC data while designing it in a way that would be extendable to practically any other city in the world. We were able to gather Taxi and Uber data from NYC, and so we started by performing individual analyses on each dataset to extract vital information while also completing comparative studies to contrast the two services. Movement through the city we live in is a big part of everyone's life and so by being able to extract characteristics of this phenomenon while also being able to predict it, would be crucial to improving it. Due to the large populations in today's metropolitans, the transportation data is extremely dense thus eliminating any traditional methods of analyzing it efficiently. Some useful questions that can be asked are as follows:

- Traffic Questions: Can traffic-jams be reduced or shortened by changing the paths taken by automobiles?
- Residential Questions: How does the type of the residential area (i.e., rural vs. urban) affect the usage of a service?
- Income Questions: How does the income of a household correlate to the service usage or lack thereof?
- Weather Questions: How does the weather affect service usage? Does it increase, decrease or not significantly alter service usage?
- Business Questions: How can businesses improve their services based on customer needs and habits?

Using all of these analyses and more, businesses can take a step toward becoming more data-driven and as a result, better understand their customers and even increase their revenue. While customers, on the other hand, can take advantage of real-time analytics provided to them through a website or an app so that they can make better choices based on actual facts instead of intuition and experience alone.

3 Data and Processing Dimensions

Our data is structured and of medium size (order of million entries approximately 20GB). Due to the restrictions imposed by the service providers, the data comes in monthly if not yearly. However, in the case of Uber, it is possible to retrieve the data for each day on the following day. Therefore, the overall sink rate is still low. To maintain scalability, we perform all data cleaning through Spark by using Map/Reduce jobs and so it is highly automatable so long as the schema does not change which unfortunately it did for parts of our data thus creating extra complications. The current system requires complete data, but over time it can be extended to handle semi-complete and potentially incomplete data through Machine Learning algorithms. During the cleaning stage, unnecessary parts of the raw data are discarded while the rest is wholly used for all further analytics. After the extensive data cleaning process, the size of the data is decreased from 20GB to 1.2GB. Therefore, the selectivity of a query does not affect the system since we only provide summaries of the analyzed data in text and graph form. Due to Spark's design, most jobs will be quite fast unless in cases where large batches of data have to be analyzed all at once when for example the data for an entire year becomes available at once instead of incrementally. Due to using all required data, the output is exact, and so precision is maximized, but this can be changed later on by incorporating sampling algorithms that can be completed much faster but provide approximate answers. We mainly use basic statistical methods such as average, median, and variance but it can be extended to use prediction algorithms.

Additionally, we chose to only analyze the data from April-June for 2014 and April-June 2015, the main reason behind this decision was the lack of data in the Uber datasets. The Uber datasets only provided the pickup date, pickup time and borough information from the above time periods. On the other hand, the Taxi dataset was quite extensive; they provided us with the number of passengers, pick up and drop off date, time, location, and much more detailed information. Unfortunately, we had to discard all these data to create a unified schema because we were not able to use them since Uber did not provide these extensive details. Since the Uber datasets were hindering our analysis, we decided to compensate for that by examining the data across multiple dimensions. We investigated how the number of Uber and Taxi rides and their averages change per time, space, and weather. To add the weather dimension, we had to use another dataset which provided us with details about the weather in NYC for every day since the 19th century. We had to clean the data and remove anything that we deemed unnecessary such as data about sea level pressure and humidity and then join the cleansed dataset with the Uber and the Taxi datasets.

4 Solution Architecture

For the current solution, we downloaded all of the needed data from Kaggle and then started to batch process it. The processing first begins by cleaning the data by removing unnecessary fields while also creating a unified schema to be used for both datasets. Unfortunately, the schemata of the two datasets are very different in that they contain different attributes but also represent the same data differently. Specifically, addresses are represented by coordinates in the Taxi dataset but as borough identification numbers in the Uber dataset which is where the unification need arises from. We initially found some modules that claimed to be able to unify the location representation, but after much testing, we realized that they were inadequate since they were either producing highly unreliable data or due to not being parallelizable. The current solution utilizes the reverse_geocode module which uses the nearest-neighbor algorithm to extract NYC boroughs from latitudes and longitudes. We found an online service that could do the mapping with very high accuracy, but it limited the number of requests to it per day. The offline library, however, has a file with location and their latitudes and longitudes and then it uses that and the nearest neighbor algorithm to map a given coordinate to an area. As a further improvement, we can try to supply our file with more location, coordinate pairs for more accurate results.

In addition to the representation issues, we also experienced many difficulties with the Uber data due to lacking much of the information that we needed to be able to compare it to the Taxi data. As a later extension to the project, we could collect the Taxi data monthly through the NYC government official website. However, Uber provides the data for each day on the following day and so to perform useful analysis, we shall save the data for each day from Uber and then a run single batch job at the end of each month. The batch jobs will remain practically unchanged compared to the current solution since the data collection does not affect the processing.

For data serving, we opted to store the data in files instead of using a serving layer such as Apache HBase mainly due to lack of resources such as time. However, the extension that would use HBase is not very difficult to achieve, and it would further improve queries through the use of Cubes and other big-data technologies. Through HBase, we could save much of the needed analytics results instead of recalculating them across different scripts. Therefore, the serving layer would not just be useful for clients but also useful for the system itself. On the visualization end, due to the highly visual nature of the data, we use actual maps of NYC while also using line graphs and bar charts.

To sum it all up, data is ingested from CSV files, stored and processed through Apache Spark using Map/Reduce, MLlib, and just standard RDD operations. Finally, the resulting analytics information is formatted and displayed using folium and matplotlib for the visualizations.

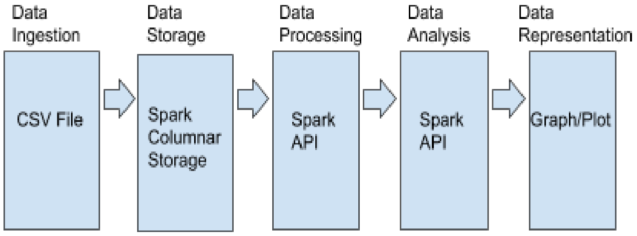

Figure 1: Data Flow in the System

A significant difficulty with the solution was configuring and using the reverse_geocode module which uses the nearest neighbor algorithm to convert latitudes and longitudes to NYC boroughs. Specifically, we experienced difficulties with setting up its scheduling and threading schemes to optimally execute on our datasets. The issue was that the available module documentation was lacking in its parallelization settings and suggested that the default mode is the best one. We later realized that the default mode is only useful for situations when the reverse geocoding is being performed on large arrays of latitudes and longitudes since it would partition the collection and create separate threads to handle each partition. However, we were parsing the data one tuple at a time, and so we were passing arrays of size 1 to the module. In the default mode, it was extremely slow to the point that if we had not figured out how to change it, it would have taken about two weeks just to preprocess the data. However, after configuring the module, we were able to finish the preprocessing of approximately ten million tuples (about 20GB) in only one hour.

Another issue that we faced was the visualization of the data which included a lot of formatting through the pandas module and the graphing through matplotlib and folium modules. Again, part of the difficulty was related to incomplete documentation. Specifically, manipulating Spark data frames into a structure accepted by folium was challenging since it required a lot of trial and error to be able to get to a working system.

Furthermore, we learned the hard way how to optimize Spark scripts to run faster and more efficiently. In the earlier version of the scripts, we were loading the entire Taxi and Uber datasets to memory and then processing it. After trying to run the previous version of the code, we found that it took too long to compute the result, and so we tried to find the reason behind this bottleneck. We ended up discovering that, it is better to analyze the data in different sections. Therefore, we start by thoroughly examining the Uber dataset and then moving on to the Taxi dataset. This resulted in a significant decrease in the total runtime of the scripts.

5 Evaluation of Results

Evaluation of the Economical Questions: These questions mainly consider the space dimension and the number of rides. We knew beforehand that Uber is dominating transportation market, this fact is known all over the world, but we were more interested in seeing, whether the sudden increase in Uber usage significantly decreases Taxi usage in NYC?

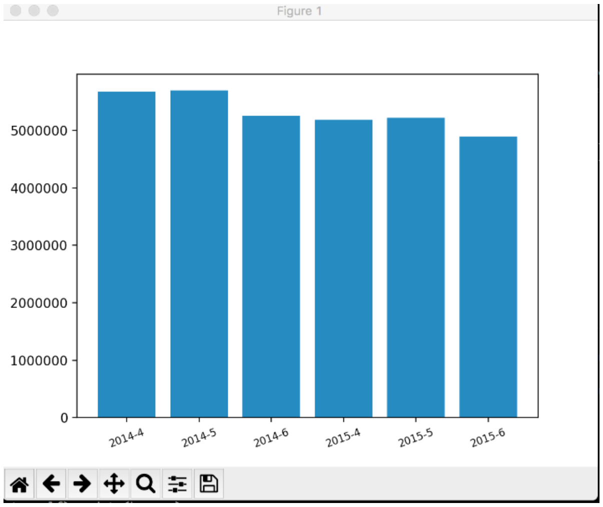

Figure 2: The number of Taxi rides in 2014 and 2015 June-April, generated by using the data from economics.py

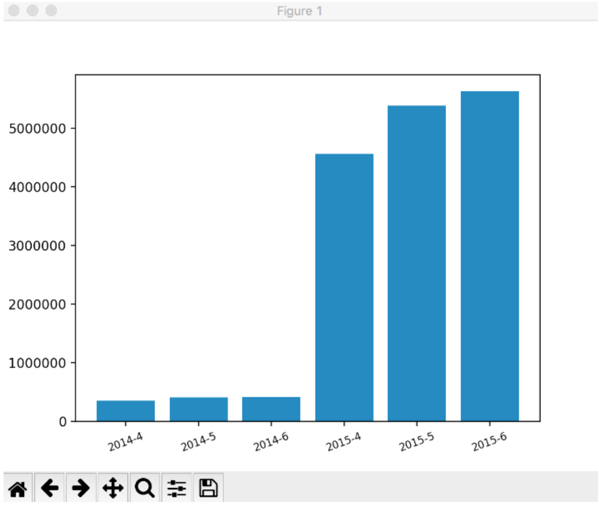

Figure 3: The number of Taxi rides in 2014 and 2015 June-April, produced by using the data from economics.py

It can easily be observed, that Uber usage increased by almost 400% between 2014 and 2015. However, the interesting result is that Taxi usage only decreased by less than 10%. We can conclude that there was a market between Taxi and personal cars that Uber utilized and that the existence of Uber has negatively impacted the Taxi business. However, we can say that there is a market for both due to the minimal effect on each other.

Evaluation of the Weather, Weekday, and Business Questions:

This question looks at how the average usage of Uber and Taxi changes in different weather conditions.

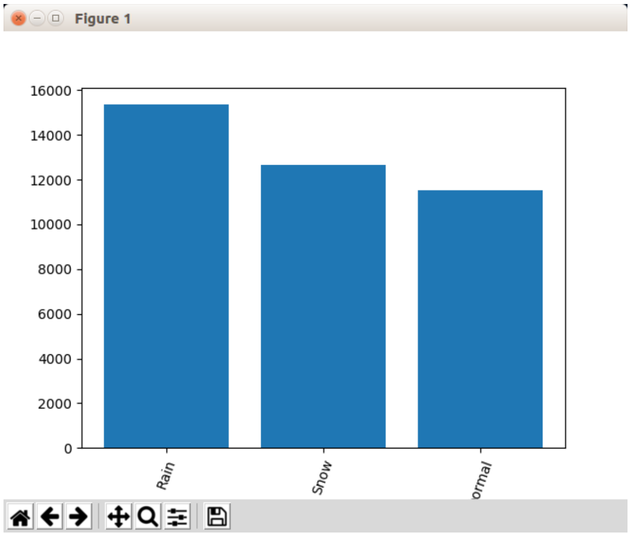

Figure 4: The average number of Uber rides across various weather conditions in 2014.

We first analyzed the Uber weather data and found that the number of Uber rides in Rain and Snow is higher than the number of rides in Normal weather conditions in 2014. To interpret this result, we did some quick research in human psychology. We discovered that most people consider Uber to be much more straightforward than a Taxi due to the difference between using an app versus talking to the operator.,

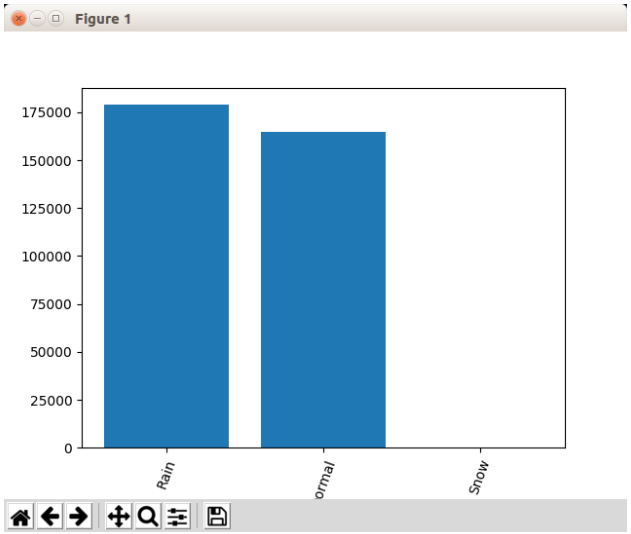

Figure 5: Uber usage in different weather conditions in 2015.

We can see that in 2015 the Uber usage in Rain is still higher than Normal weather conditions. We a concluded that most people would prefer to save money if the weather permits by walking to their destination or taking public transportation. However, in abnormal weather conditions, taking public transportation is not convenient, believe me, I have tried it.

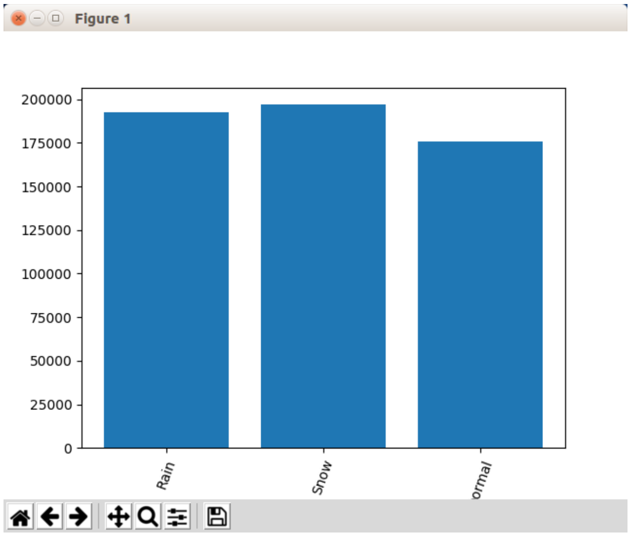

Figure 6: The average number of taxi rides in different weather conditions 2014

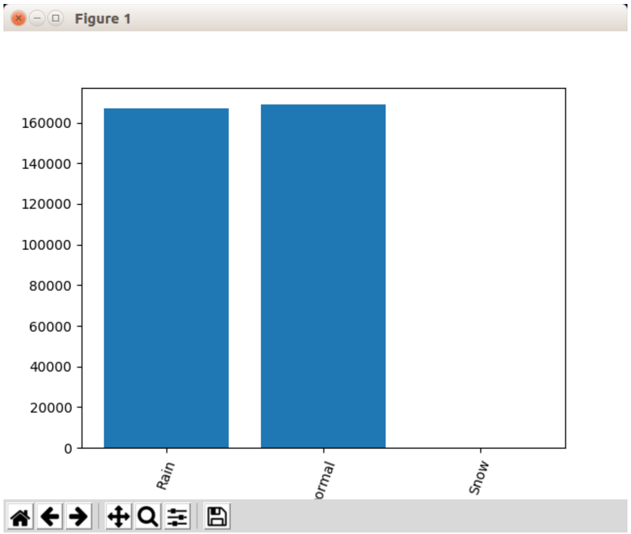

Figure 7: The average number of trips in different weather conditions in 2015.

It can vividly be observed that Taxi usage in Normal weather conditions is higher than the Taxi usage in abnormal weather conditions, and the same reasoning used for the Uber data can be applied here.

Evaluation of the Bussines Improvement Questions:

The following consists of some conclusions that can be drawn from the data to improve the services:

- Uber could utilize and promote itself as a convenient way of transportation in abnormal weather conditions. Furthermore, Uber could charge more in unusual weather conditions, as it's harder to drive in Snow and Rain.
- Taxi services should provide customers with a way of ordering them without the need for human interactions (only few Taxi companies do so). This, in turn, would increase their usage in abnormal weather conditions.
- Uber usage in Normal weather conditions is lower than abnormal weather conditions. Therefore, there is some improvement that can be made in that area.

Please note that the Snow data for 2015 Uber and Taxi usages is empty, only because in that period, there was no snow.

Evaluation of Hotspots and Residential Questions:

Unlike the previous questions, we were only able to draw a very basic conclusion for these questions. We found that both Taxi and Uber usage increase during rush hours and that Manhattan has the highest usage of Uber and Taxi. This is due mainly to the fact that Manhattan is densely populated. Even though the results of this questions were trivial, we decided to take one step further and provide the user with an interactive map that can be used to see how the Uber and Taxi usage changes per day. Creating the interactive map was one of the hardest tasks that we performed.

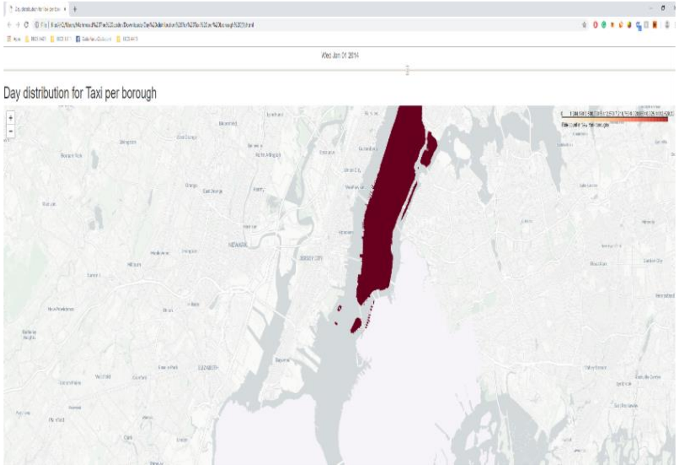

Figure 8: NYC borough transportation heat map.

As mentioned before, one of the most significant limitations in the project was the lack of information in the Uber data. If the Uber data provided similar information as the Taxi data, we could have done some if not all of the following:

- Create an application that recommends the Uber driver what the best spot and time to pick up the customer is to maximize the profit. This application would be updated on the go by analyzing streaming data and reporting the results through the app.
- Create an application that recommends the cheapest area to find a Taxi or an Uber to customers. Typically, the more demand for Uber, the higher the cost. This is known as a surge in the field of Economics.

6 Summary statistics

Taxi usage summary is as follows:

```
('Monday', 4297514.0)

('Tuesday', 4734085.0)

('Wednesday', 4869398.0)

('Thursday', 4920833.0)

('Friday', 4831733.0)

('Saturday', 4397988.0)

('Sunday', 3858086.0)
```

Uber usage summary is as follows:

```
('Monday', 77225.76923076923)

('Tuesday', 85644.61538461539)

('Wednesday', 90051.84615384616)

('Thursday', 98632.53846153847)

('Friday', 101079.69230769231)

('Saturday', 105486.30769230769)

('Sunday', 86541.61538461539)
```

7 References

- “Apache Spark Tutorial.” www.tutorialspoint.com, Tutorials Point, www.tutorialspoint.com/apache_spark/
- “Spark.apache.org. (n.d.).” Quick Start - Spark 2.4.0 Documentation. [online] Available at: https://spark.apache.org/docs/latest/quick-start.html.
- “Folium.” Folium - Folium 0.5.0 Documentation, python-visualization.github.io/folium/docs-v0.5.0/modules.html.
- “Pandas.DataFrame. (n.d.).” Retrieved from https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.DataFrame.html
- Thampi, Ajay. "Thampiman/reverse-geocoder." GitHub, 15 Sept. 2016, github.com/thampiman/reverse-geocoder.
- Zonination. “Zonination/Weather-Us.” GitHub, 23 June 2016, github.com/zonination/weather-us.