Taxi Business: NYC Taxi Lucrative Locations and Drivers

Futong Han
Vrije Universiteit Amsterdam
hanfutong0804@gmail.com

Kai Zhang
Vrije Universiteit Amsterdam
main@kai.sh

Kalle Janssen
Vrije Universiteit Amsterdam
k6.janssen@student.vu.nl

1. INTRODUCTION

The yellow cabs in New York City (NYC) are some of the most iconic taxi cabs in the world and a popular site for tourists. But these cabs are not driving around for tourists to look at, they need to make money. Taxi cabs need to transport a large number of passengers in order to stay profitable or increase their prices if they are unable to achieve a high quantity of rides. Those of us who have been on a taxi ride in Amsterdam and NYC know that the rates in Amsterdam are quite expensive compared to their counterparts in NYC [15]. This means that a yellow cab needs to transport more passengers per hour or for a longer amount of time than the taxis in Amsterdam to stay profitable. This project will not focus on the economics of taxis but was heavily inspired by it.

To achieve this higher amount of passengers per hour it can be useful for a taxi driver to get information for where to head to next. In practice this does not happen, instead each taxi driver has to decide where to go next based on their own experience [16] [10]. They can either drive towards a "hot-spot" that typically has a high probability for new customers or they can drive around hoping they will get a new passenger.

According to salary.com [6], there is quite a large gap between low and high earning taxi drivers where the bottom 10% earn nearly $30,000 and the top 10% more than double. This large gap is likely due to an increase in total working hours but can also be influenced by the efficiency of experienced taxi drivers when looking for new passengers. One research argues that the current taxi fleet could be reduced to 72% of the current size while still transporting the same amount of passengers [19]. This shows the large inefficiencies in the current taxi system which could be improved by reducing idle time which in turn would increase profits. This can be done by learning what the high earning drivers do differently and let other drivers benefit from it.

In short the goal of this research is to: (1) Find lucrative pickup places for taxi drivers and (2) Find commercially successful drivers, who earn much more than others per hour driven (corrected for time-of-day). [2] These results will then be visualized appropriately. Then, a data-driven algorithm will be made which can help drivers find the closest most profitable location. This location will not just be based on the average most profitable location but will also be based on day, time of day and weather. This will help inexperienced drivers increase their revenue. This will all be done in an intuitive and understandable visualization in order to let readers get the most important information in a short amount of time.

To achieve this target, this project will be based on the New York Official Taxi Dataset from TSL[4]. This data set includes information gathered from taxis between 2009 and 2020. This information includes GPS information, tips, income per trip and more. However, the data set provided by TSL does not include the individual taxi drivers. For the individual drivers an extra data set is used provided by Illinois Databank, New York City Taxi Trip Data (2010-2013) [8]. This data set includes the hack license numbers which can be used to track individual drivers which is essential for calculating high earning taxi drivers.

In this report, related works that will be used to achieve the final goal of this project will be described in section 2. Section 2 will also explain how these researches were conducted and how this project will further build on these related works. In section 3, the research questions, their corresponding hypotheses, which software/hardware limitations need to be conquered and what trade-offs need to be made to achieve the final target of this project will be discussed. IN Section 4 the setup of this project will be discussed and how the projects will be implemented It presents the input data and the output data product plus the pipeline that was created to achieve the goal. In the conclusion, section 5, the research questions will be answered and discussed using the hypotheses as described in section 3. In this section other interesting findings, profitable areas and commercial successful driver characteristics will also be shared. Finally, the future work that this project was not able to achieve will be described.

2. RELATED WORK

2.1 Lucrative pickup places

Many efforts have been made to find busy pickup locations and increasing the income of taxi drivers. Each and every paper uses the same formulas and pipeline to generate profitable paths, routes and points for taxi drivers [17][12][10][18][16][9]. All these researches calculate all nearby profitable paths, routes or points using the formulas shown below and in Figure 1.

- P_{id} = The a path or a point with id
- Pr_{id} = The Pid's probability of pick-up a passenger
- A_{id} = The Pid's all trips average income
- E_{id} = The Pid's estimated income

$$E_{id} = Pr_{id} * A_{id} \tag{1}$$

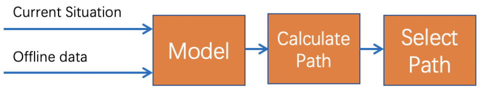


Figure 1: Driver Profitable Model Core

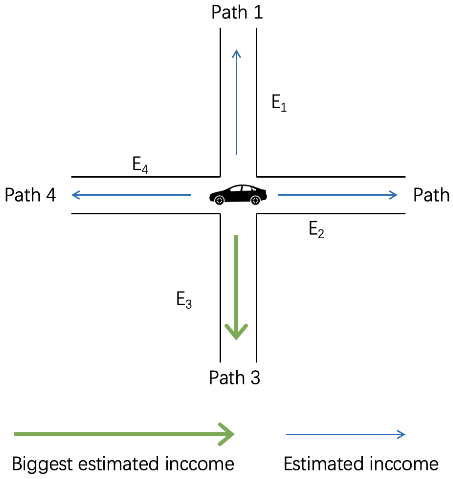


Figure 2: Core Driver Profitable Path/Point Selecting Model

There are countless number of points in the provided data set. Selecting which paths/point to go and how to calculate the possibility of pick-up possibility in the next unit time is a difficult thing to do and differs from paper to paper.

In [17] a density cluster algorithm was used and introduced stopping time as a parameter to find all good stop points. [9] also used density algorithm to get the hot-spots. In [12], the roads were transformed to polygons with fixed width and mapped all the pickup points to corresponding routes. This research had almost the same probability core formula, they used taxi pick-up trip number divided by the sum of taxi vacancy cruise trip numbers and taxi pick-up trip number as the picking up probability. [16] used a grid based solution. It transferred the map into lots of same small size square regions, and clustered the trips to their belonging regions. For the probability part it used ARIMA (Auto regressive integrated moving average) model, it used past 168 hours data to predict next hour's taxi demand for each region. Next profit locations for taxi drivers were suggested by constructing a spatio-temporal profitability map. [10] used Monte Carlo Tree Search, which also implemented on AlphaGo, to calculate the best routes and probability for drivers. [18] used K-means clustering to cluster roads segments and used OPTICS density cluster methods to discover what essentially the same parking places were and then used Poisson distribution to get the probability of getting a passenger in next unit time.

After reading the above mentioned paper, we decided to use a density cluster algorithm DBSCAN to analyze where the hot-spots are and then analyze the related data to see how and what can influence these hot-spots. After a week of developing this algorithm we came to the conclusion that this would not yield the expected results, this will be further discussed in section 4. The new plan was to use to a method similar to the one used in this paper [12], where the pickup points are grouped to the nearest edge. The next step was to find a way to calculate the pickup probability. A method used by three of the above mentioned papers calculated the pickup percentage had a high precision but it needed to calculate all the empty cruise routes which takes a lot of time. Since this project only had four years of data that included the unique drivers [8], the probability data might have been already out of date. The ARIMA is not suitable for multiple years of data but only for the recent few days. Finally a Poisson distribution as the probability function in the final solution was used. Using the Poisson distribution it not only easy to do calculations on such a large data set but it can also use recent years of trip data. The Poisson distribution and pick-up calculation formula is shown as followed:

$$Pr(N(t) = n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \tag{2}$$

Pr is the probability, N denotes a function relation, t represents time, n represents pick-up passenger number and λ represents the average pick up times on the point/edge in the unit time. So, the probability that the driver won't pick a passenger on the edge in coming unit time is $Pr((1) = 0)$, and in the coming unit time that a taxi driver can pick a pick up a customer at specific point/edge's probability is:

$$\begin{aligned} Pr_{pick} &= 1 - Pr_{not\_pick} \\ &= 1 - Pr((1) = 0) \\ &= 1 - \frac{(\lambda * 1)^0 e^{-\lambda * 1}}{0!} \\ &= 1 - e^{-\lambda} \end{aligned} \tag{3}$$

Using the methods for calculating hot-points/edges and the passenger probability calculating formula the core driver profits model can now be implemented. This model can point out profitable points and edges based on historic data. The next step is to return personalized profitable points based on the current location of the taxi driver and let them traverse through different profitable points. For example, a path near the driver can have a very high estimated income, but its next path's estimated income can does not have have a high estimated income. This path and profitable points should then not be recommended to the nearby drivers. Or in another circumstance, there are high profitable points on the map but these points are far from the taxi driver's current location. Letting the taxi driver drive to those profitable points to just get one customer is likely not an economic profitable strategy. For this final implementation the brute force algorithm from [12] is used. It uses breadth-first search algorithm to find all potential routes for the current driver based on location, then uses the estimation income formula to calculate the estimated income for all routes. The routes estimate income is the sum of all its edges' estimated income. The edge estimated income follows the formula (1) shown as before. And the customer pick-up probability calculation is not only based on Poisson distribution but also based on the previous edge in the route. Finally, the top-k income routes for the taxi driver are selected. The routes estimated income formula and edge pick-up probability illustration graph is as follows:

- $E_{rid,ith}$ = The #id route ith edge's estimated income
- E_{rid} = The #id route's estimated income
- $Pr_{rid,ith}$ = The #id route ith edge's probability of pick-up a passenger
- Pr_{rid} = The #id route's probability of pick-up a passenger
- $\lambda_{rid,ith}$ = The #id route ith edge's average pick-up trip numbers
- $A_{rid,ith}$ = The #id route ith edge's all trips average income
- N_{rid} = The #id route's sub-edges number

$$
\begin{aligned}
&Pr_{rid,i} = \begin{cases} 1 - e^{-\lambda_{rid,1}} & (i = 1) \\ (1 - \sum_{n=1}^{i-1} Pr_{rid,n}) * (1 - e^{-\lambda_{rid,i}}) & (1 < i \leq N_{rid}) \end{cases} \\
&Pr_{rid} = \sum_{n=1}^{N_{rid}} Pr_{rid,n} \\
&E_{rid,ith} = Pr_{rid,ith} * A_{rid,ith}(1 \leq i \leq N_{rid}) \\
&E_{rid} = \sum_{n=1}^{N_{rid}} E_{rid,n}
\end{aligned} \tag{4}
$$

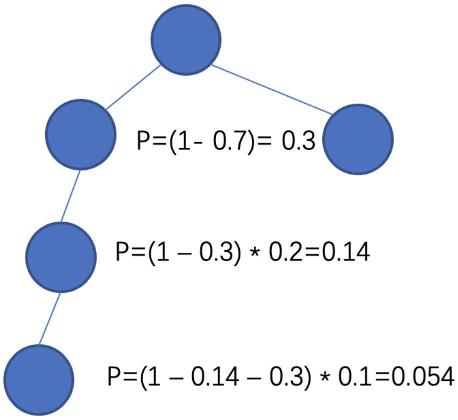


Figure 3: Pick-up Probability Calculation Example

The lucrative pickup places model structure overview is shown as figure 4:

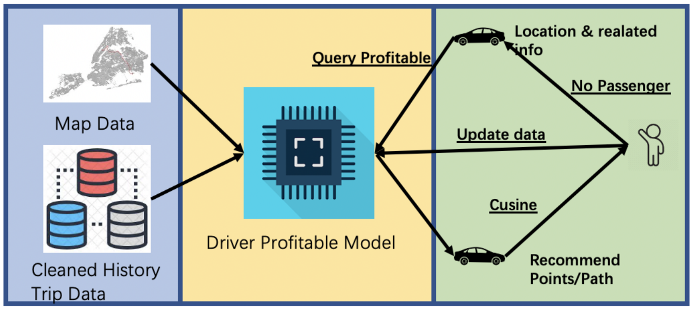


Figure 4: lucrative pickup places activity flow

2.2 Find commercial success driver pattern

There are already many excellent researches for predicting taxi drivers' fare. Inspired by solutions in the Kaggle project "New York City Taxi Fare Prediction"[5] and the report [11] a decision tree was used to find the most important features[3] that contributed to the driver's final income. After that, SQL queries were used to find out profitable drivers' common patterns, so that other drivers can learn from the patterns and behave like them.

3. RESEARCH QUESTIONS

3.1 Project Research Questions

The research methodologies used for this project have been described in the Related Work section, the next step is to formulate research questions and corresponding hypotheses. The following research questions are based on the goals of this project which is finding lucrative pickup places for taxi drivers and finding successful drivers driving patterns. To achieve these goals 4 research questions and their hypotheses will be discussed in the following paragraphs.

1. Where are common lucrative pickup locations and what are their attributes? From the early findings and the projects on Kaggle[5] it became apparent that Midtown Manhattan in NYC always requires the most taxis and likely is a highly profitable area. If there are other locations that are also always busy and profitable it would be interesting to see if those locations are near popular locations like shopping centers, train stations, JFK airport, etc.

2. What are important features that can influence taxi drivers' income? And how does the average income change based on these features? There are many different attributes logged in the NYC taxi data set and more will be added after cleaning the data. Some examples of these features could be weather, time or day. We expect the most influential feature to be the time of day, it is expected that during the start and end of the workday the business heavy locations thrive but this might change during the weekend. These features can then be used as input in the model for predicting profitable locations near the current location of the taxi driver.

3. What are the most profitable locations based on the current situation (location, time, etc)? As mentioned in the previous research question different features can influence the next most profitable location. This question will be in the form of a visualization that shows taxi drivers the most likely next lucrative pickup location based on the current situation.

4. What are the driving patterns of successful and unsuccessful drivers? Based on the question 2's important features, what are the successful drivers preferred working pattern and in which conditions do they prefer not to work? This will be answered using a visualization which will present successful drivers and their patterns. Other taxi drivers will be able to benefit from this visualization by learning what their more successful colleagues do.

3.2 Technical Research Question

Besides these four question, there are also has some technical questions and challenges in this project:

1. Can pandas interact with pyspark? Is it efficient to complete the goal of this project? The technical stack of this project is based on python since lots of spatial python libraries are based on pandas and geopandas. These two libraries are essential for data engineering and feature engineering. However, these two libraries are not designed for big data since they only use one core during calculations. So the main challenge is if these libraries are able to finish the tasks in order to finalize final target. Or would it be better to write parallel functions or even implement other methods? Moreover, are these libraries able to cooperate with pyspark to do big data calculations? How efficient is this? These questions need to be researched during the data visualization.

2. How to store the data and load the visualization efficiently on a disk in order to response to users' requirements? The NYC taxi data set is a big data set and there could be lots of important features that can influence high income pick up points. These features can have multiple values, this results in a large amount of different combinations which increases the size of the visualization data. This has to be stored efficiently so that the system can recommend profitable points quickly and ensure that it can be scaled accordingly when the data changes.

3. What problems should be kept in mind during the process of data storage and use? A main concern is the speed at which data can be read. The data storage should be done in an efficient matter, one method to increase the speed is partitioning. Partitioning data can increase the response time which in turn can increase the speed of the final visualization.

4. PROJECT SETUP

4.1 Input data

1. NYC Taxi data from TLC
The original data set was provided by the City of New York [4]. It contains information gathered from different types of taxis in New York between 2009 and 2020 where each month has a separate CSV file. This data set consists of four smaller data sets: green cabs, yellow taxis, FHV trips and high volume FHV trips. The green cabs can only operate in specific districts in NYC so this was not so useful for this project. Furthermore, the yellow taxi trip data after 2016, the FHV trip data and the High Volume FHV trip data do not have GPS information which is essential in this project so these data sets were also ignored. The main data set that can be used is the yellow taxi data from year 2009 to year 2016. This data set has a number of different features, the features that will be used in this project can be seen in table 1 with their corresponding descriptions. The data in this data set will be the main focus for finding the most profitable areas in the years 2009 to 2016.

Data	**Description**
pickup_datetime	The date and time when the meter was engaged.
dropoff_datetime	The date and time when the meter was disengaged.
pickup_latitude	The latitude when the meter was engaged.
pickup_longitude	The longitude when the meter was engaged.
dropoff_latitude	The latitude when the meter was disengaged.
dropoff_longitude	The longitude when the meter was disengaged.
trip_distance	The elapsed trip distance in miles reported by the taximeter.
tip_amount	This field is automatically populated for credit card tips. Cash tips are not included.
total_amount	The total amount charged to passengers. Does not include cash tips.

Table 1: TLC Yellow Taxi Trip Data Selected Header

Data	**Description**
medallion	a permit to operate a yellow taxi cab in New York City, it is effectively a (randomly assigned) car ID.
hack license	a license to drive the vehicle, it is effectively a (randomly assigned) driver ID.

Table 2: Illinois Dataset Extra Header Description

2. Illinois Databank NYC Taxi Trips Data
To find commercial successful taxi drivers, the individual taxi drivers need to be differentiated from each other. But as can be seen in Table 1 there is currently no way to track an individual taxi driver. To combat this problem an extra data set provided by Illinois NYC Taxi Trips Data [8] is included in this project. This data set essentially has the exact same data as the TLC data set but the *medallion* and *hack license* numbers are also included. The descriptions of these two can be seen in Table 2. This data can be used to track taxi drivers to find successful, average and low-income drivers. The available years in this data set are 2010 to 2013 which is slightly less than the TLC data set.

3. New York Weather
Weather can influence people's daily routine and also their taxi usage patterns. To research this, *beautifulsoup* [13] was used to extract weather information from *timeanddate.com* month by month. All the data is saved in csv format like a table and the original .html page is saved as backup and partitioned by year and month. The following columns are included in this weather data set: hour, day, month, year, high temperature, low temperature, wind Speed, humidity, and weather description. The only problem of this webcrawl is that it is not available for a free user to fetch data more than 11 years ago. So we only have New York Weather

Data after Sept. 2009.

4. New York Map Data
This project demands a high level of spatial calculation, like finding nearest edge (road) from a point, finding hot-spot coordinate location information for the visualization and so on. As explained, requesting spatial data is essential, however, requesting data from online maps services is too expensive and has a too high latency for the big data calculations. To conquer this problem, the New York Map Data from OpenStreetMap was fetched. This map is an open sourced project by osmnx library [7] which fetches map data from a remote server and and stores it local disk with .mph format for future use.

5. New York Zone Shape File
For the visualization NYC will be divided into Taxi Zone, the New York Taxi Zones from New York TLC Official Dataset[4] will be used and saved as *.shp* format locally. The taxi zones were created by the government scientifically to view taxi traffic situation, dispatch taxis to suitable area and research. This file will be used to clean the coordinates on the map, finding profitable areas and searching for high-income district attributes. An extra benefit of using this data set is that if this project in the future needs to be combined with a data sets that only uses zone-id's it can be easily implemented which extends the scalability of this project.

4.1.1 All Initial Data Size

The original taxi data set from TLC is saved in monthly .csv format files, each file is around 2.3GB. This project will use 90 months of data between 2009 and 2016 for a total size of around 207GB.

The extra trip data set with hack license numbers from the Illinois Databank [8] contains data collected between 2010 and 2013. Like the previous data set this data set is also saved in separate monthly .csv files for a total of 48 months. The size of this data set is 37.7GB compressed and 142.4GB decompressed. Due to the IO latency data decompressing took 5 hours to complete on databrick's platform. After unzipping we found the trip data of each month was logged in two files, one is data_trip file and the other file is data_fare file. Splitting the data in two is not convenient, therefore we need to combine them in a later stage and save the combined data with the same directory structure as the original TLC data set. Using the same structure for all files means that the same data clean program and data feature program can be used on them. Using the same structure also contributes to time saving at the algorithm deployment stage.

The New York related data includes: map file, zone file and weather file which in total size is 75.1 MB.

Thus, after data preparation stage the input raw file total size is around 349.5 GB.

4.2 Data Process Pipeline

This section will present our data process stages in detail, the first step, Gather Data Stage, has already been discussed in the previous section. The data processing stage overview can be seen in Figure 5.

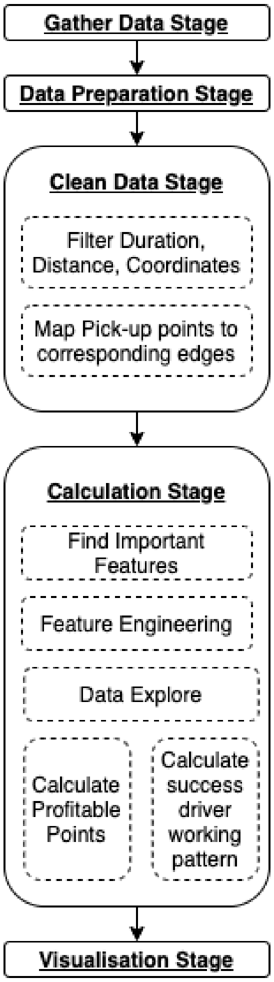

Gather Data Stage
Data Preparation Stage
Clean Data Stage
Filter Duration, Distance, Coordinates
Map Pick-up points to corresponding edges
Calculation Stage
Find Important Features
Feature Engineering
Data Explore
Calculate Profitable Points
Calculate success driver working pattern
Visualisation Stage

Figure 5: lucrative pickup places activity flow

4.2.1 Data Preparation Stage

The data preparation stage aims at forming scattered data into the correct directory structure, extracting useful data and dropping useless data in the data set to prepare for later use. Besides, updating input data set's schema with the right data types is also an important task.

To process the TLC NYC Taxi data set (TLC data set), *pyspark* was used to read in the data month by month and followed by the data in Table 1 to rename and select the needed columns. Finally, the processed data was saved in *.parquet* format with *gzip* compression. Each months data's processing time takes around 2.5 minutes and the file size for each month is now around 400 MB on disk compared to the original 2.3 GB.

Regarding the Illinois Databank NYC Taxi Trips Data(Illinois dataset) raw data, the first thing we did was combining *trip_data.csv* and *trip_fare.csv* of each month with *pyspark*'s *join()* method by *"medallion"*, *"hack_license"*, and *"pickup_datetime"* columns. After that, column renaming and stated columns selecting (displayed in Table 1 and Table 2) procedures are included. In the last step the data was saved in *.parquet* format as well as with *gzip* compression. The whole time to prepare this data set took 30 minutes on databricks, and the total size of this data set drops from 142.4 GB to 39.6 GB.

The TLC data set's data saved in the "yellow" folder, and Illinois data set was saved in the "foil" folder. Both the TLC data set and Illinois data set were saved in the directory structure:
"DATA_DIR/TRIP_TYPE/YEAR/MONTH.gz.parquet". For example TLC data set year 2010 January's data of our group01 saved at "/mnt/group01/yellow/2010/1.gz.parquet" and Illinois data set year 2010 January's data of our group01 saved at "/mnt/group01/foil/2010/1.gz.parquet"

Extracting all roads and intersections from the New York Map is the last step in the Data Preparation Stage. For the intersections data, x (Longitudes) and y (Latitudes) are important to locate where the intersection are on the map where *osmid* denotes the unique id to search it in the data set. For the roads data, u (the road start intersection osmid) and v (the road end intersection osmid) pair is not only important to locate where the road is on the map but also is the unique id to search it in the dataset, *length* and *oneway* are important features to instruct drivers in the future.

4.2.2 Clean Data Stage

In this stage the invalid trips in the TCL trip data set and Illinois data set will be removed. There are three main filters, one semi-filter and getting the nearest edges for pick-up points in this stage. The three main filters are based on the coordinates, trip distance and trip duration. Sometimes the GPS in the taxi can not report the exact coordinate and will instead return 0, less than -90 or more than 90 degrees. These trips and trips outside of NYC using New York bound from [1] were removed from the data set.

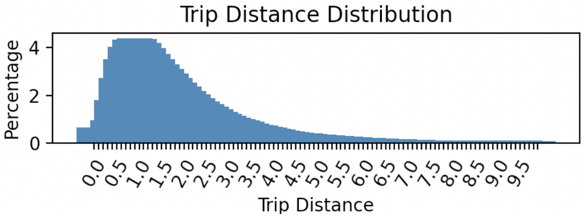


Figure 6: Trip Distance Distribution

The second filter removed the trips whose trip distances were too short. Trips shorter than 200 meters were removed. 200 meters only takes a person less than 3 minutes to walk (5 km/hr), it seems unlikely that someone would pay for a taxi when walking is likely faster. Figure 6 shows that these trips account for less than 2% of the data set which even if some of these trips are real trips will not influence the data set in a large way.

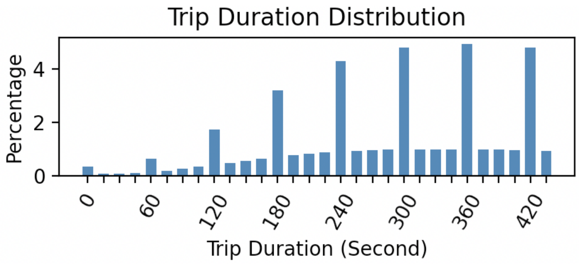


Figure 7: Trip Duration Distribution

The last main filter removed trips that were either too short or too long. Trips shorter than 45 seconds or longer than 4 hours were removed, trips shorter than 45 seconds only account for no more than 1% of the data set (Figure 7).

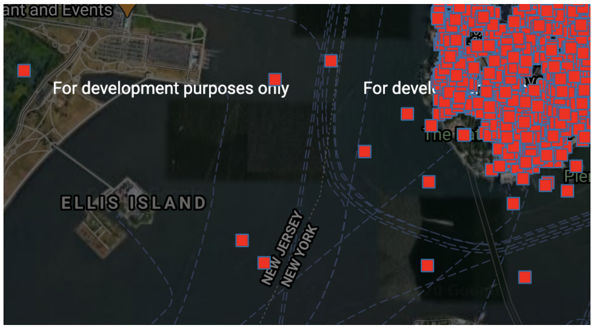


Figure 8: Trip Duration Distribution

The semi-filter was used to filter out the inaccurate drifted pick-up coordinates, sign pick-up points related taxi zone id and find the pick-up points' nearest road on the map. There are wrong points in the data set, for example as the figure 8 shows the pick-up coordinates in data set can be on the river or on the sea, where the taxi or passenger should not locate at.For calculating the hot-spots these points are not valid points, but for the successful driver characteristics or trip information, these coordinate data should not be filtered out. So that is why this filter is a semi-filter, we tagged the points really in New York city with taxi zone, otherwise we leave it as null. This procedure needed to use *geopandas*, finding the nearest edges also needed *geopandas*. *Geopandas* dataframe vs *pyspark* dataframe takes a long time so we did find related zone location operation and nearest edge operation at the same time in this section.

All in all, in this stage 46 hours was spent to clean the TLC and Illinois data on the databricks platform and each month's data contains around 15,000,000 records. It took such a long time that 5 notebooks ran in parallel to clean the data. Most time was spent on *pandas* module parquet reading time, *geopandas* module pick-up coordinates nearest edge finding time and *geopandas* results finding time. The pyspark *.toPandas()* methods took too long so we saved the results from *geopandas* then read the file by *pyspark*, this will be discussed later in the conclusion.

4.2.3 Calculation Stage

In this stage the feature engineering was done first and then the forest tree important feature module from *sklearn* was used to find the core features that can influence taxi driver's income. After the important features have been found in the previous step, we found profitable nodes and high income taxi driver exploration in parallel.

In the feature engineering stage we added "weather" related features and "if work at weekend" feature for both the TLC dataset and Illinois dataset. In order to find the most successful drivers, we added a few more features to the Illinois dataset. We added "cruise duration" feature to describe cruise time between when a taxi driver dropped of their previous customer and got their next customer, "sleep in day" feature to describe if a driver works more during the day or midnight, ”work time” feature to show a driver’s day work length in seconds and ”trip count” feature to show a driver’s day trip count.

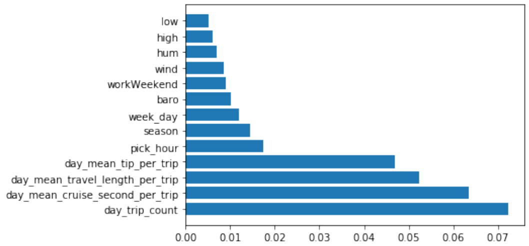


Figure 9: Decision Tree Feature Importance

After this stage, *sklearn* was used to find important features that can influence drivers’ income the most. These results can be seen in Figure 9. As can be seen in this figure the most important features are ”trip numbers per day”, ”average cruising time”, ”average trip distance”, ”pick-up hour”, ”season”, ”day in week”, ”weather” and ”if the taxi driver is work at weekend”.

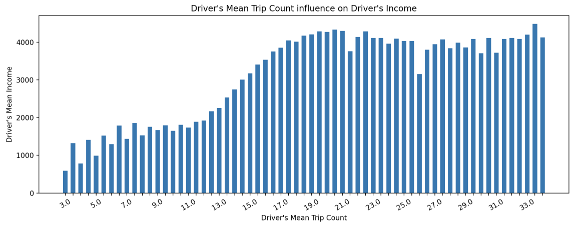


Figure 10: Driver’s Average Month Income VS Driver’s Average Pick-up Trips Counts

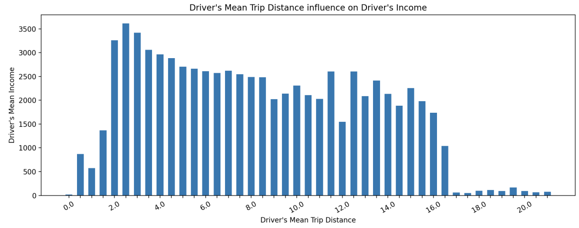


Figure 11: Driver’s Average Month Income VS Driver’s Average Trip Distance

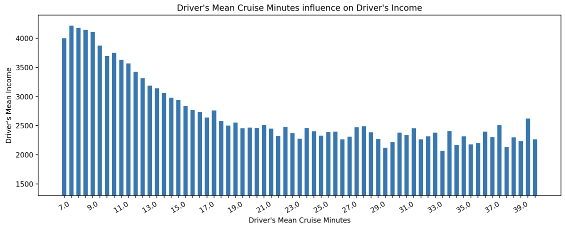


Figure 12: Driver’s Average Month Income VS Drivers Average Cruise Time

Each driver has their own ”average trip numbers” such as ”average trip distance” and ”average cruising time”. The figures 10, 11, and 12 show that the more pickup trips per day the more a driver can earn. Long distance trips do not improve the earnings of drivers and will perform worse than more short distance trips.

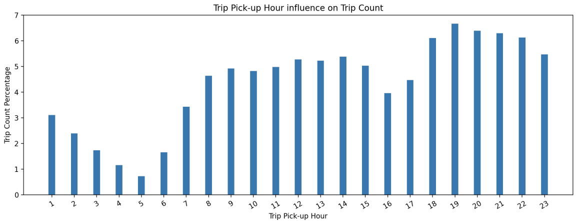


Figure 13: Trip Percentage VS Trip Hour

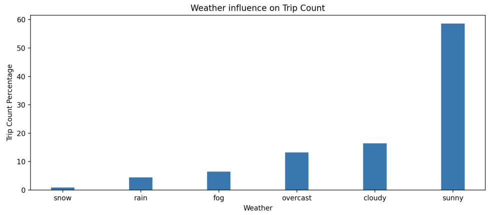


Figure 14: Trip Percentage VS Trip Weather

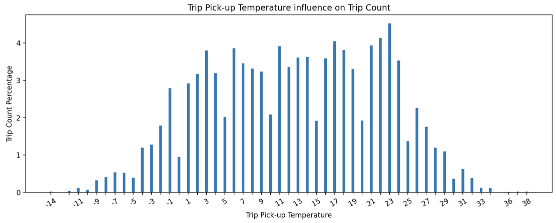


Figure 15: Trip Percentage VS Trip Temperature

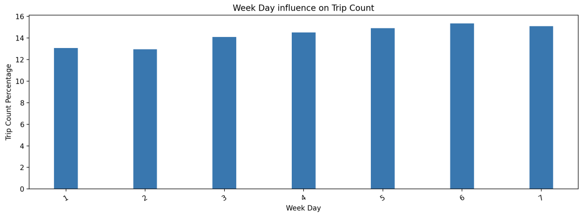


Figure 16: Trip Percentage VS Trip Week Day

Based on the findings from driver’s related graphs, we explored what can influence taxi trip demanding of the days. Each trip has its own ”pick up hour”, own ”pick up weather description”, own ”pick up temperature” and own ”pick up week day”. In figures 13, 14, 15 and 16 can be seen that each feature can influence the demand of trips. However, trip week day’s influence on trip demanding is not obvious as the others.

Based on the previous results we decided to use the features ("minute", "hour", "season", "weather", and "temperature") as those features can influence trip demanding greatly as driver input and based on the methodology introduced in "Related Work" section to give them profitable paths/points recommendation. And use related features to show successful driver characteristics.

In the last step of this part we used the TLC data set processed by *PySpark* SQL to generate final recommended profitable routes/points data. And use the Illinois data set to generate successful driver characteristics visualization.

4.3 Density Cluster Method (Abandoned)

The "find nearest edge method" was not the first choice for this project, we also tried the DBSCAN method. We spent 1.5 hours to cluster all pick-up points in the Illinois data set, and only got 450 clustered points. It showed only a few points on Manhattan island while Manhattan should be one of the most busiest locations and displayed other points in rural areas where it should not be so busy. This is an over-clustered result for all points. Then we tried to cluster each month's data first then re-cluster the previous clustered result.

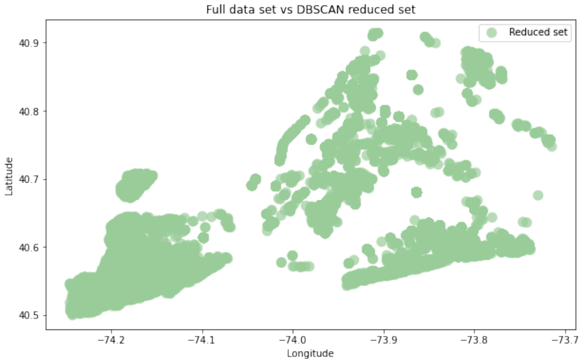


Figure 17: Clustered points

The merge-clusters result can be seen in Figure 17. The results seem better, however, it still shows lots of scattered points in the outskirts of NYC and less points in city center. We can not get the correct high-income points in the outskirts and the points in city center are still over clustered. It is difficult to overcome in such a short time, so we abandoned this method. But later in the review stage, we found OPTICS to search for real hotspots could solve this problem.

5. CONCLUSIONS

5.1 Profitable/Hot spots

This section will answer the research questions proposed in Section 3. **1. Where are common lucrative pickup locations and what are their attributes?**

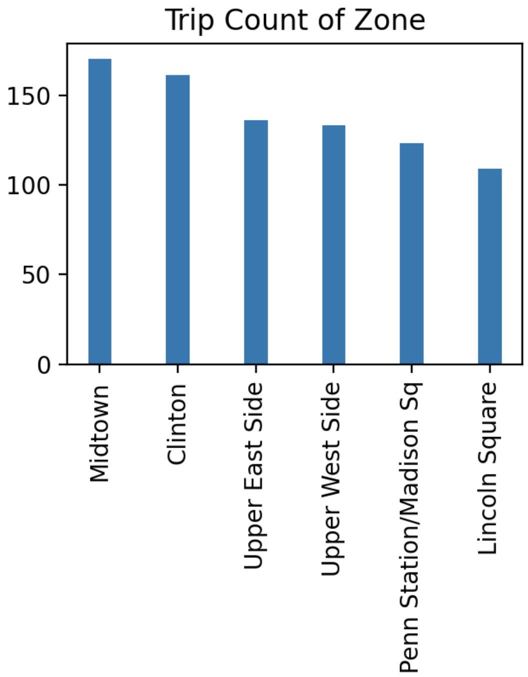


Figure 18: Common Profitable Zones

The previous research findings as described in Section 2 were used to make a strict filter. This filter was used to search all the trips at 5 am, snowy weather and with extreme temperatures. The results of this search can be seen in Figure 18, this figure illustrates the amount of demand each zone had with the above mentioned requirements. Manhattan Midtown, Brooklyn Clinton, Manhattan Upper East Side, Manhattan Upper West Side, Manhattan Penn Station and Manhattan Lincoln Square always have very high taxi demand and are the most profitable areas for drivers. These areas are all very popular entertainment (art exhibition/bars/shopping) locations or are part of the financial center of NYC. An interesting observation is that JFK Airport is not as busy as one might think. Airports are usually great locations for taxi drivers but in this case Manhattan is much more popular so it would be better to go to Manhattan. Another surprise is the popularity of Brooklyn Clinton, Clinton is an area south of the busy Manhattan but has a trip count comparable to Manhattan.

2. What are important features that can influence taxi drivers' income? And how does the average income change based on these features?

This question has already been discussed in Section 4.2.3: Calculation Stage. The most influential features for the income of taxi drivers are lower cruise times (taxi is unoccupied) and lower average trip distances. Taxi rates are calculated in a way that decreases the costs per minute as the trip duration increases, this means that a minute in a short taxi ride is more expensive than a minute in a longer taxi ride. Lower trip distances can ensure that taxi drivers can transport more different passengers for shorter amount of time which in turn increases the earnings per hour. Other features that can influence taxi drivers' income are pick-up hour, weather and if it is a weekday.

3. What are the most profitable locations based on the current situation (location, time, etc)?
Based on the path estimation income formula described in the Related Work Section and calculation stage's result data. We made a web page with server to find the most profitable paths. Instead of just finding the most profitable points, a driver can enter their current location, time, season, weather and temperature. **Caution:** the path to the next income points does not have to be the profitable cruise path.

This method to find the nearest road intersection of input coordinate is server based. An example of this web page is shown below: (select input, get new zone status (figure 19) , right click to get most profitable route figure (figure 20), hover and get route info (figure 21 and figure 22)):

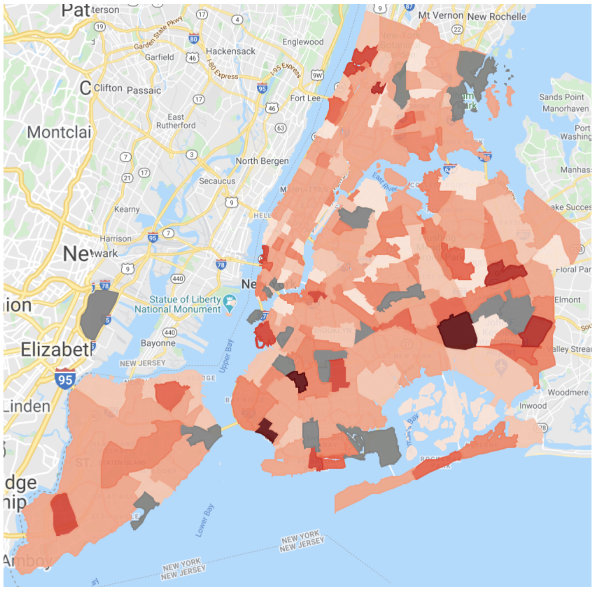


Figure 19: Get New Zone Status

The Dynamic Server is saved at "dbfs:/mnt/group01/LSDE_2020_IV_FINAL_SUBMISSION/ LSDE_Dynamic.tar.gz"

Since the final submission should be a static web page we changed the number of drivers for the static web page. This page shows the top 1000 high incoming routes and profitable areas with red color, the darkest edges are the most profitable edges.The driver can pick those dark road around him to traverse or select the most profitable areas to go. An example can be seen below (select input, get new zone status and top 1000 profitable routes(figure 23), click and view zone's information(figure 24), zoom to view high income routes (figure 25), hover and get route info (figure 26 and figure 27)):

The Static Web Page is saved at "dbfs:/mnt/group01/LSDE_2020_IV_FINAL_SUBMISSION/ LSDE_Static.tar.gz"

5.2 Driver characteristics

This section will answer the final research question from Section 4: What are the driving patterns of successful and unsuccessful drivers?"

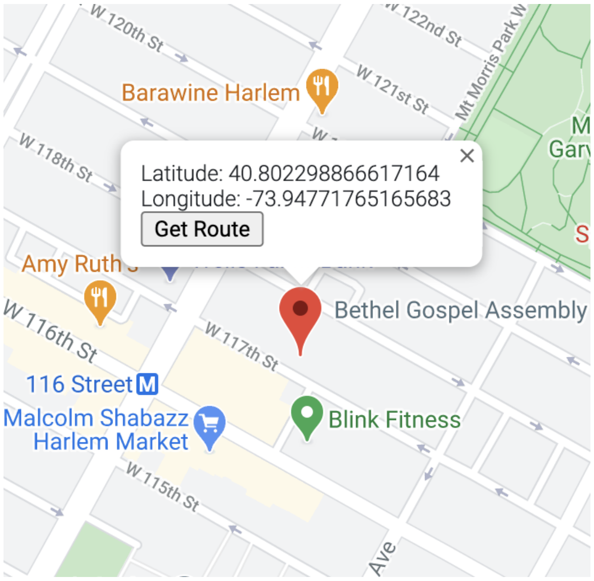

Latitude: 40.802298866617164
Longitude: -73.94771765165683
Get Route

Figure 20: Right Click To Get Route

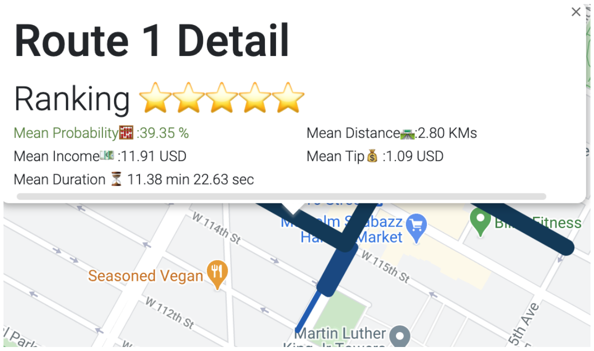
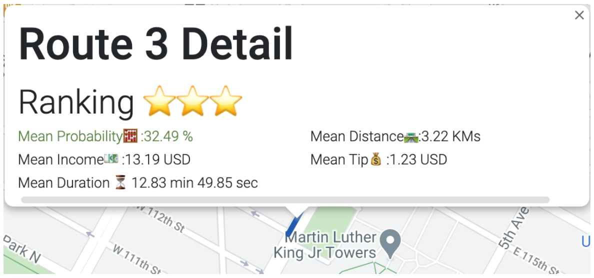

Route 1 Detail
Ranking ★★★★★
Mean Probability :39.35 %
Mean Distance :2.80 KMs
Mean Income :11.91 USD
Mean Tip :1.09 USD
Mean Duration 11.38 min 22.63 sec

Figure 21: View Rank 1 Cruise Route Info

To find commercially successful drivers, we came up with a situation: If we are a new taxi driver in New York, what should we do to make more money? Generating successful drivers' behavior pattern from cleaned data is a good way to show it. From [14], we learned that there are rich feature importance information when we used decision tree and this information can help us find which character influence driver's income most. After the calculation of drivers' income per hour, we label the top 10 percent drivers with 'high income' and the other drivers are labeled with 'low income'. Then we use the feature we generated above to train the model and rank the features by their feature importance.



Route 3 Detail

Ranking ☆☆☆

Mean Probability:32.49 %

Mean Distance:3.22 KMs

Mean Income:13.19 USD

Mean Tip :1.23 USD

Mean Duration 12.83 min 49.85 sec

Figure 22: View Rank 3 Cruise Route Info

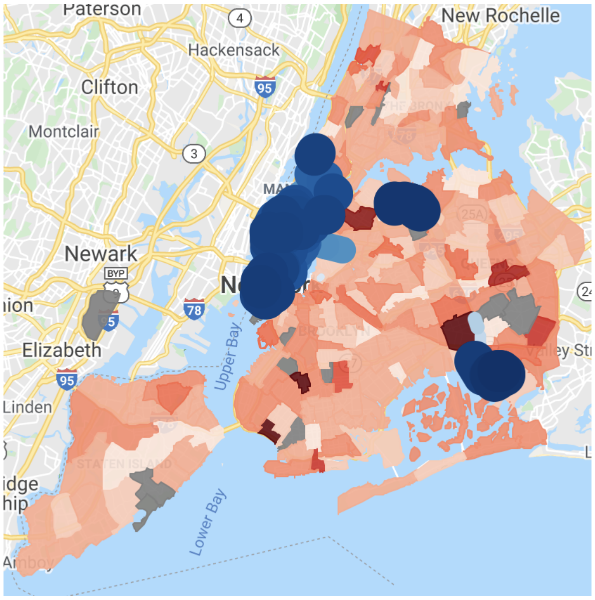


Figure 23: Zone Info and High Income Roads

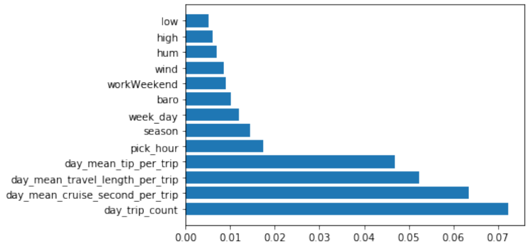
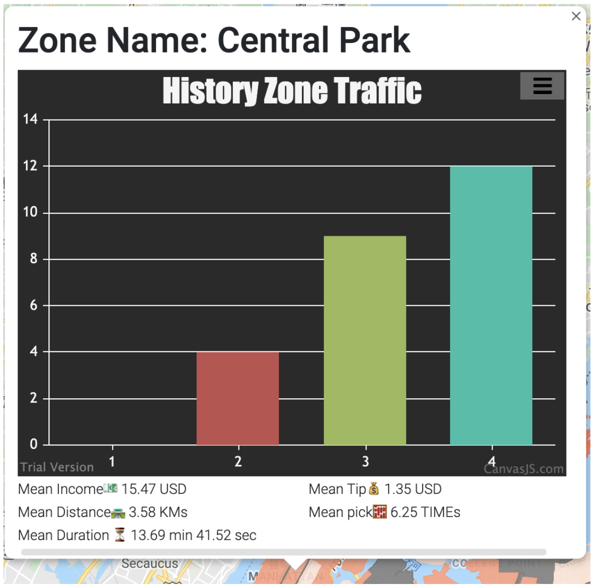


Figure 28: Decision Tree Feature Importance

As can be seen in Figure 28: 'trip count', 'cruise time', 'travel distance' and 'average tip' all strongly influences the final income. After that, 'pick hour', weather related features and 'weekday' also influence the somewhat. As a final step the trip data was analyzed by selecting the features with higher importance as selected by the Decision Tree. Below the driver's behaviour can be seen which we found to influence the average income:

5.2.1 *Time*

In Figure 29 the distribution of taxi trips on each day can be seen. The most popular day for taxi drivers is Saturday with 18% of the trips, the least popular day is Tuesday with 10% of the trips. The distribution of the taxi trips per hour can be seen in Figure 30, as can be seen in this figure the most popular times are between 17:00 and 00:00. This 7 hour time frame provides 68% of the taxi trips while the remaining 17 hours account for 32% of the trips.



Zone Name: Central Park

Mean Income 15.47 USD

Mean Tip 1.35 USD

Mean Distance 3.58 KMs

Mean pick 6.25 TIMEs

Mean Duration 13.69 min 41.52 sec

Figure 24: Click and view zone information

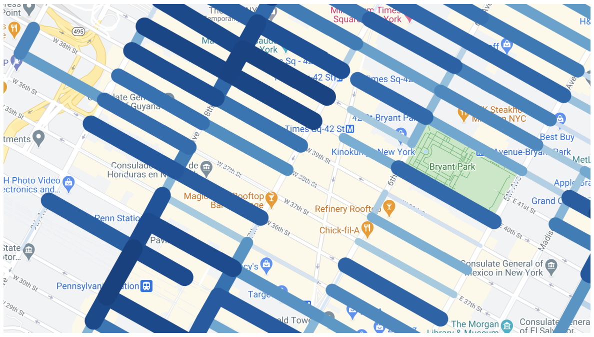


Figure 25: Zoom Map and check Roads

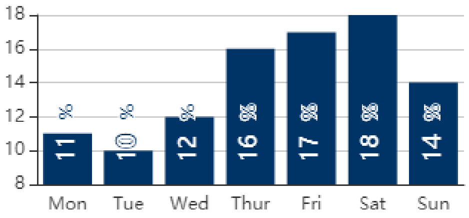
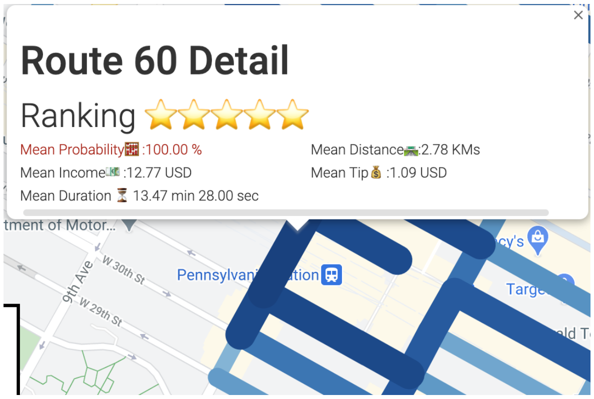


Figure 29: Weekday distribution



Route 60 Detail

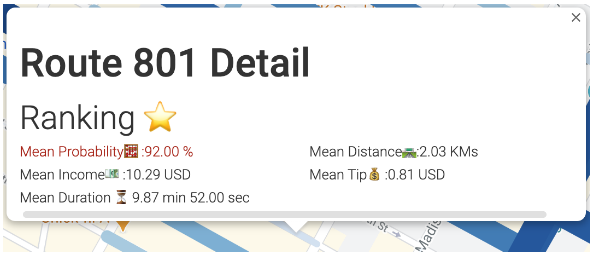
Ranking ★★★★★

Mean Probability :100.00 % Mean Distance :2.78 KMs

Mean Income :12.77 USD Mean Tip :1.09 USD

Mean Duration 13.47 min 28.00 sec

Figure 26: View high rank roads info



Route 801 Detail

Ranking ★

Mean Probability :92.00 % Mean Distance :2.03 KMs

Mean Income :10.29 USD Mean Tip :0.81 USD

Mean Duration 9.87 min 52.00 sec

Figure 27: View low rank roads info

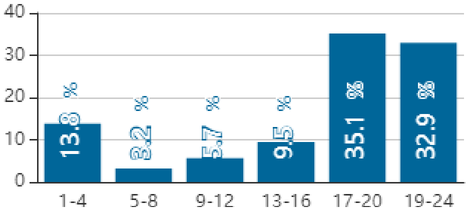


Figure 30: Hour distribution

5.2.2 *Trip distance*

We calculated the relationship between average trip distance of drivers and the average income of these same drivers, the results can be seen in 31. Drivers who perform longer trips on average will earn less on average, this phenomena plateaus at around an average trip length of 3 miles.

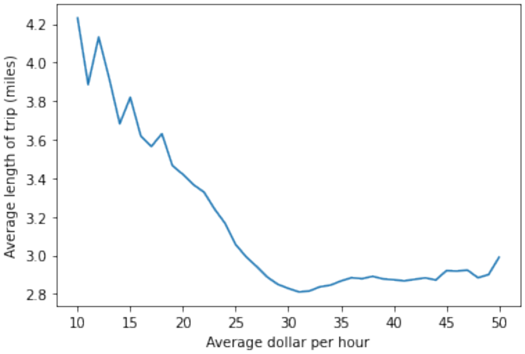


Figure 31: Trip distance

5.2.3 *Cruise time*

The last finding of this project is the difference in waiting time, where a taxi driver drives without a passenger, between junior and senior taxi drivers. On average senior drivers spend 10.1 minutes waiting on a new passenger while junior drivers spend 13.1 minutes waiting. The difference between these two groups could likely be reduced if the junior drivers learned the tactics of the senior drivers.

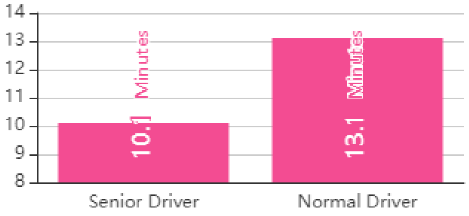


Figure 32: Cruise time

5.3 Technical Research Question

1. Can pandas interact with pyspark? Is it efficient to complete the goal of this project? In the development stage, some data could only be used with *pandas* due to some libraries being limited to only work with *pandas*. Although *pyspark* has a *.toPandas()* function which transforms dataframes to pandas dataframs, this transformation is very inefficient. So we saved the calculation with *pyspark* first, then read it with *pandas* finally save the *pandas* calculation result and read it with *pyspark*. The reason is that if we use *.toPandas()* directly, *pyspark* will transfer spark rdd to numpy array row by row. This procedure can take a very long time. This problem can be solved by arrow function, *df.select("*").toPandas()* with spark configuration: *spark.conf.set("spark.sql.execution.arrow.enabled", "true")*

2. How to store the data and load the visualization efficiently on a disk in order to response to users' requirements?

The taxi drivers' input can have thousands of combinations, if we saved the final result in a single file this file would be more than 1GB. This will take a long time for users to load data. Instead we solved this problem by savinin *json* format and partitioned by "minute", "hour", "season", "weather" and "temperature".

To solve this problem we saved the result file in *json* format and partitioned by "minute", "hour", "season", "weather" and "temperature". This results in the response data for each request being only around 30 KBs which increases the visualization data loading time.

3. What problems should be kept in mind during the process of data storage and use?

During the data exploration and data cleaning stage more than 1 million files were created using using very deeply partitioned delta tables. This is because we encountered out of memory when we did experiment on my own computer in write out stage. To solve this problem we did large partition. We then realized that partitioning too deep and split to many files can cut down on query latency which results in slower queries since there are too many small files. We solved this problem by correct spark memory setting. Furthermore, properly partitioning data can indeed speed up queries, like the one we did in visualization data but too much partitioning can have the opposite effect. So in the following work the spitted number of files was reduced which makes the queries faster.

Contribution

What	**Who**
Project Plan	Kalle Janssen
Non-trivial data	Kalle Janssen
Presentation	Kalle Janssen, Futong Han, Kai Zhang
Paper research	Kai Zhang, Futong Han, Kalle Janssen
Data Pipeline	Kai Zhang, Futong Han
IV Experiment	Kalle Janssen, Kai Zhang
IV Data Generation	Kai Zhang, Futong Han
Visualization site	Futong Han, Kai Zhang
Visualization	Futong Han, Kai Zhang, Kalle Janssen
Report	Futong Han, Kai Zhang, Kalle Janssen

6. REFERENCES

[1]

[2] New york taxis - t1: Taxi business.

[3] Feature importances with forests of trees, 2020.

[4] New york city taxi and limosine commission, 2020.

[5] New york city taxi fare prediction, 2020.

[6] Nyc salary, 2020.

[7] G. Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems 65*, 2017.

[8] B. Donovan and D. Work. New york city taxi trip data (2010-2013), 2016.

[9] Y. Hu, Y. Yang, and B. Huang. A comprehensive survey of recommendation system based on taxi gps trajectory. In *2015 International Conference on Service Science (ICSS)*, pages 99–105, 2015.

[10] S. R. Nandani Garg. Route recommendations for idle taxi drivers: Find me the shortest route to a customer! *kdd*, 2018.

[11] V. N. Paul Jolly, Boxiao Pan. Caesar's taxi prediction services. 2011.

[12] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong. A cost-effective recommender system for taxi drivers. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2014.

[13] L. Richardson. Beautiful soup documentation. *April*, 2007.

[14] A. A. A. S. Jalil Kazemitabar. Variable importance using decision trees. *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.*, 2017.

[15] R. Wade. World taxi prices: What a 3-kilometer ride costs in 88 big cities, 2017.

[16] H. G. Xiangpeng Wan. A generic data-driven recommendation system for large-scale regular and ride-hailing taxi services. *mdpi*, 2020.

[17] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun. Where to find my next passenger. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, page 109–118, New York, NY, USA, 2011. Association for Computing Machinery.

[18] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Trans. on Knowl. and Data Eng.*, 25(10):2390–2403, Oct. 2013.

[19] C. Zhu and B. Prabhakar. Reducing inefficiencies in taxi systems. December 2017.