Big Data Processing and Applications
Project Report

New York City Taxi Trip Data Analysis

Student 1: Aditya Mehta
Student 2: Andrei Baraian
Student 3: Baptiste Hutteau

14-05-2020

Project Description

In today's fast moving world, a huge amount of pressure is put on being able to rapidly adapt and reconfigure to the latest information possible in order to be more efficient. The vast amount of available data and cutting edge technology makes it possible, only if we know how to efficiently analyze the data. The purpose of our project is to study and analyse possible patterns and insights from the data obtained from a fleet of taxi operating in New York City. The New York City Taxi Limousine Commission regulates all taxis and for-hire vehicles[3]. For this, we have chosen the publicly available dataset offered by *New York City Taxi & Limousine* [4]. The main purpose of choosing this dataset is to explore the traffic patterns in famous New York City Taxi system. We aimed to begin with exploratory data analysis and tried to implement some machine learning algorithm as well for prediction of fare amounts based on different features. There are many use cases that can be formulated for this dataset, like traffic monitoring and optimization so as to be able to provide approximate duration of trip to customers, financial predictions, social analysis, optimization for operations by making data-based decisions etc.

We have decided to focus our research and analysis in the following topics:

- **Traffic Optimization**
  - Analysis of number of trips in December 2019
  - Exploration of average distance per ride over days of week
  - Identify hot-spots and traffic free areas across NYC
- **Financial Analysis**
  - Analysis for distribution of fare amount according to traffic zones
  - Analysis of shared vs. single use of taxi
  - Fare amount prediction based on various features

This list is not exhaustive and we have also tried to find other insightful information from the data (some of them successfully, some not), but they will be described in the below sections. To perform analysis on this amount of data, we needed programming skills in Python and good knowledge of libraries such as PySpark, SQL and data visualization tools (matplotlib). For good code management and collaboration, we used individual notebooks for this project work. As a development IDE, we have used Jupyter Notebook, for its fast prototyping, easy integration with PySpark and the commodity of keeping the variables in memory, so we would not have to read the data every time we modify part of the code.

Data Description

Dataset Collection

The New York City Taxi and Limousine Commission (TLC) is the agency responsible for licensing and regulating New York City's Medallion (Yellow) taxi cabs, street hail livery (SHL) green taxis, for-hire vehicles (FHV), commuter vans, which are used for serving various commuting needs in the city. The data in this dataset were collected and provided to TLC by technology providers authorized under the Taxicab & Livery Passenger Enhancement Programs (TPEP/LPEP). As per terms of use, this dataset can be used for lawful purposes, including but not limited to academic research projects as well as for industry projects, with terms of use available on [1] with dataset released as part of open data initiative [6]

This dataset is collected on a monthly basis and shared in comma separated value (CSV) files to be downloaded from the data page of TLC [4]. There are data available from the year 2009 to 2019 with separate data file for yellow taxi, green taxi, for-hire-vehicles for each month of the year. However, this data volume is quite huge. For this project work, we are focusing on yellow taxi (609 MB) and green taxi (39 MB) for the latest available data, from the month of December in 2019.

Dataset Description

Overview of Taxi System

Yellow Taxi

Also known as Medallion Taxis, they provide transportation exclusively through street-hails. Passengers can access this mode of transportation by hailing an available taxi with hands. The pickups are not prearranged. A fixed number of medallions vehicles can accept street hails and electronic trips (e-hail) in the city. [2]

Green Taxi

Also known as Street Hail Livery (SHL) or boro taxis, these taxis can accept street hails and electronic trips, as well as pre-arranged trips. This program allows livery vehicle owners to license and outfit their vehicles with green borough taxi branding, meters, credit card machines, and ultimately the right to accept street hails in addition to prearranged rides. [2]

Dataset

There is one file for yellow taxi, one for green taxi, and one for zone information. The yellow and green taxi trip records include fields capturing pick-up and drop-off dates/times, locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts. It contains information on the geolocation and collected fares of individual taxi trips. Using this data, operations can be planned in a better way and optimized by analyzing the gathered data and making data-based decisions.

The variables in the dataset consists of variables with similar datatypes. Only one additional variable is present for green taxis. These two dataset description has been provided in the tables 1 and 2:

Table 1: Dataset description common for two taxi types

Field Name	Description	Datatype	Data Example
VendorID	TPEP provider identification record (1 or 2)	int	1
tpep_pickup_datetime	Date and time of engaging meter	timestamp	2019-12-01 00:26:58
tpep_dropoff_datetime	Date and time of disengaging meter	timestamp	2019-12-01 00:41:45
passenger_count	Number of passengers in the vehicle	int	1
trip_distance	Miles reported by the taximeter	double	4.2
RateCodeID	Type of rate for fair applicable (1 to 6)	int	1
store_and_fwd_flag	In server unavailability, record was saved in vehicle memory (Y or N)	string	N
PULocationID	Taxi Zone of engagement	int	142
DOLocationID	Taxi Zone of disengagement	int	116
payment_type	Payment mode for trip (card, cash, others) (1 to 6)	int	2
fare_amount	Fare calculated by the meter	double	14.5
extra	Miscellaneous surcharges such as rush hour and overnight	double	3.0
mta_tax	Based on the metered rate in use	double	0.5
tip_amount	Populated for credit card tips, cash tips exclusive	double	0.0
tolls_amount	Total amount of all tolls paid in trip	double	0.0
improvement_surcharge	Assessed trips at the flag drop	double	0.3
total_amount	Amount charged to passenger. cash tip exclusive.	double	18.3
congestion_surcharge	Charges based on traffic congestion	double	2.5

Table 2: Additional dataset description for green taxi

Field Name	Description	Datatype	Data Example
Trip_type	street-hail or a dispatch trip (1 or 2)	int	1

Data Pre-processing

For beginning with data analysis, we started with creating spark application on the assigned cluster by SparkSession then used read csv methods of spark, with inferring schema as well.

The Yellow and Green taxi file contains 6896317 and 450627 rows respectively. However, upon inspection of data, we begin with cleaning and pre-processing of this data as this contained many null values, invalid data and outliers, for instance, pick up time after drop off time. For better analysis, we added one column named as duration, i.e. timestamp difference between pick-up and drop-off. This step was found out to be most time-consuming and it consisted of two parts, first detecting the outliers and secondly removing them from records. The steps followed for data pre-processing has been summarized in table 3.

Table 3: Data PreProcessing Steps for Yellow and Green Taxis

Description	Yellow Taxi	Green Taxi
Number of records in raw file	6896317	450627
Are there any missing values for any columns?	Yes	Yes
Records after null values removal for all columns	6845299	359900
Are there any duplicate records?	No	No
Are there any invalid data? (incompatible values)	Yes	Yes
Records after removal of invalid data	6381864	326968
Are there any outliers?	Yes	Yes
Records after removal of all outliers	6381860	326965

For outlier detection, we begin with plotting scatter plot for different variables in context of frequency of occurrence. This has been shown in 1, 2, 3, and 4. The visual representation of variables helped us to narrow down the range of outlier values as depicted in below figures. Later on, by filtering dataframe according to lower or upper limit permissible values, we removed such outliers.

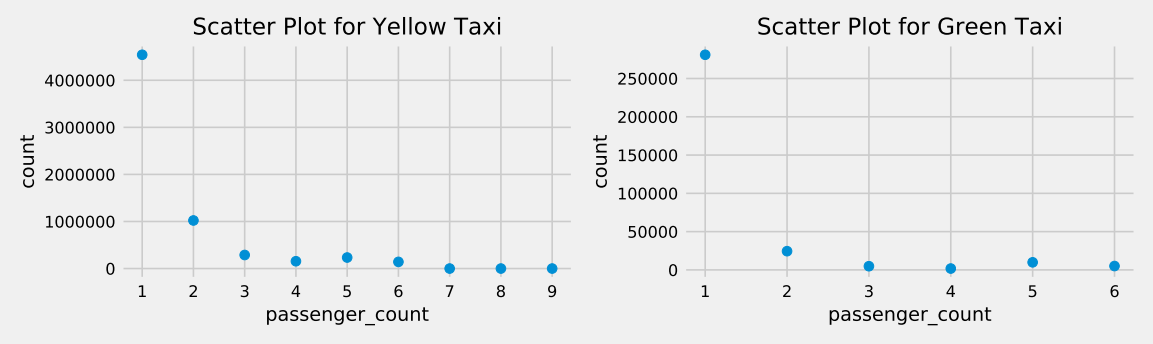

Figure 1: Scatter plot for passenger count

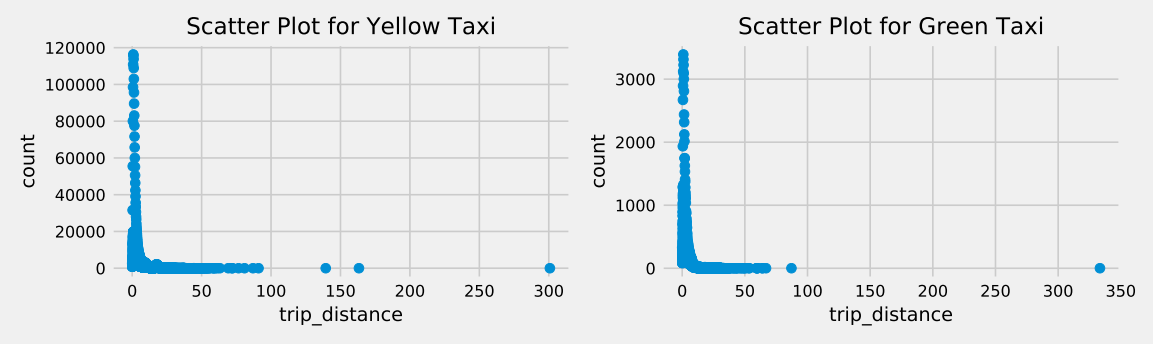

Figure 2: Scatter plot for trip distance

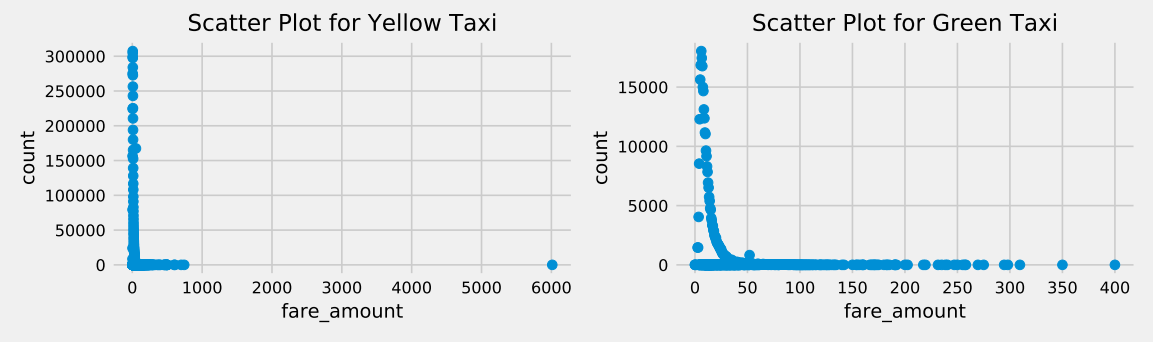

Figure 3: Scatter plot for trip fare amount

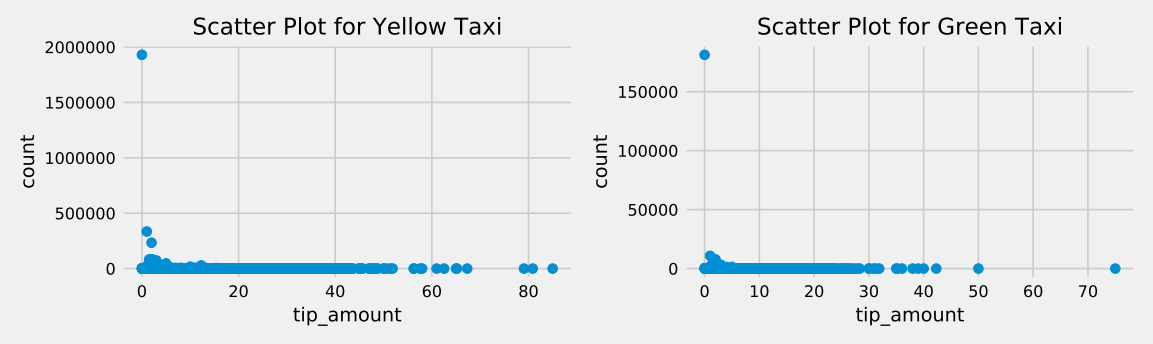

Figure 4: Scatter plot for trip tip amount

As outliers, we find trip duration as high as 100 hours and even negative trip duration. As an example, trip duration should be positive and not longer than 12 hours [2]. For this, we needed to filter the rows that do not match the criteria. Since we are not familiar with all aspects of New York transportation, we could not find outliers regarding tolls amount, mta_tax, congestion surcharge. We focused on few parameters after observation of scatter plots as given in previous figures. Another step in the data cleaning was to verify that the entries for categorical variables are actually correct and are not outside the specified range. We were in fact surprised that we did find some erroneous entries, in the case of pickup and drop-off location IDs and rating codes. So, we proceeded with removing such records as well.

After cleaning both files, the summary statistics were obtained, which have been summarized briefly in table 4 and 5 only for numerical variables of interest.

Table 4: Summary Statistics for Yellow Taxi

summary	passenger_count	trip_distance	tip_amount	fare_amount
count	6381860	6381860	6381860	6381860
mean	1.58	2.94	2.24	13.16
stddev	1.17	3.77	2.59	11.03
min	1	0.01	0.0	0.0
max	9	163.3	79.06	743

Table 5: Summary Statistics for Green Taxi

summary	passenger_count	trip_distance	tip_amount	fare_amount
count	326965	326965	326965	326965
mean	1.32	2.67	1.20	11.87
stddev	0.97	3.02	1.89	6.61
min	1	0.01	0.0	0.01
max	6	87.14	75	309.5

Methods and Tools

For this project data analysis, we divided tasks in data cleaning and preprocessing, data manipulation and data visualization. As the dataset primarily consists of multiple variables, we focused on exploration of two datasets and their similarity and/or differences. Given the amount of limited data, i.e. only for one month time period, we decided not to implement advanced machine learning algorithms and to work with exploratory data analysis.

Exploratory data analysis (EDA) is an approach of analysis of data for summarizing main characteristics, often finding relations between variables of interest through using visualization techniques and tools. This consists of using statistical methods, which helps to understand distribution of data so that further analysis can be implemented to find correlation or causation patterns. Spark provided useful features such as interactive spark shell, dataframe abstraction to support structured analysis, local mode computation such that processing can be done with minimal latency. Also, the Notebook interface is

smoothly integrated for EDA. The tools of exploratory analysis are mainly graphical representations, such as: histograms, bar charts, box plots, mean plots, etc.). With Spark, one can easily find outliers based on drawing scatter plot, box plot, and/or histogram. From mathematical view, interquartile range is used to consider values only within a range of normally occurring values.

For this project, we decided to use the exploratory analysis technique concept for getting initial insights into our data. The reason for choosing this method is that given limited time span, we wanted to spend enough time to clean the data first and to explore various patterns. Data cleaning and pre-processing took most of the time as expected. As a step-wise approach, we need to perform data cleaning to ensure that we work with correct data. First we create a spark session on the given cluster by creating an application for analysis. This session runs on the cluster with multiple (4 in our case) worker nodes who have their own designated memory (3 GB in our cluster) and each having 2 cores. The running application and processes can be monitored on the Spark UI dashboard. We use high-level spark APIs for the analysis.

For exploring relations between various variables of interest, we worked with statistical analysis approaches such as by counting sum, average, minimum, maximum values of trips per day, fares collected, passenger counts etc. For analysis, we primarily worked with PySpark module, dataframes for analysis, occassional SQL query operation for specific purposes, then to convert to Pandas DataFrame so that we can work with visualization of analysis. In sum, libraries such as PySpark, SQL API were used for pre-processing, cleaning and analysis part. The results after analysis in Spark DataFrame were converted to Pandas DataFrame for visual representation. For visualization, we worked with Matplotlib, Seaborn libraries in Python.

The reason for primarily choosing these tools is that we had some prior experience in Pandas Dataframe for data exploration and Matplotlib for visualization. However, using Spark cluster, processing our programs in a distributed fashion was learning experience. The exercises were helpful to get the basic environment setup. During this project, we got to learn about Spark, PySpark, distributed processing, APIs of Spark such as SQL and streaming API. We used SQL API in our analysis to perform relational database style queries. For data visualization, Seaborn and Matplotlib libraries were used. How they can be used in combination was a learning experience.

In order to perform regression, we have decided to use the machine learning APIs of Spark and use the Gradient-Boosted Tree (GBTRregressor) to fit a regression model. To evaluate this model, we have used the Root Mean Squared Error (RMSE), which is a way of measuring the error of a model in predicting quantitative data.

$$RMSE = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} (y_i - x_i)^2}$$

Data Analysis

In this section, we present the answers to our research questions. This section contains separate sections for different questions, the analysis, and implications found out by this analysis.

Total trips per day for two taxi types

To analyse the similarity and/or differences in total daily trips for two taxi types, we performed analysis of dataset by extracting pick up date from all trips, then to group them as per the pick up date and to visually represent the patterns for two taxi types usage. This has been depicted in 5.

From this analysis, we identify that for green taxi types, Sunday contains the minimum demand for taxis. Interesting here is that the taxi usage increases monotonically usually along weekdays. The taxi consumption becomes highest on Saturdays and falls again on Sunday. This pattern is somewhat similar for yellow taxis. However, the rise and fall of the taxi demand is not monotonically increasing and decreasing. One common finding is that first few weeks of December 2019 had significantly higher total trips per day as compared with that in the last weeks. Also, the date (25th December 2019) consists of the minimum trips in the month. One main reason is that the number of yellow taxis deployed are comparatively higher than green taxis, which can be interpreted in different number of total trips, i.e. on the y axis.

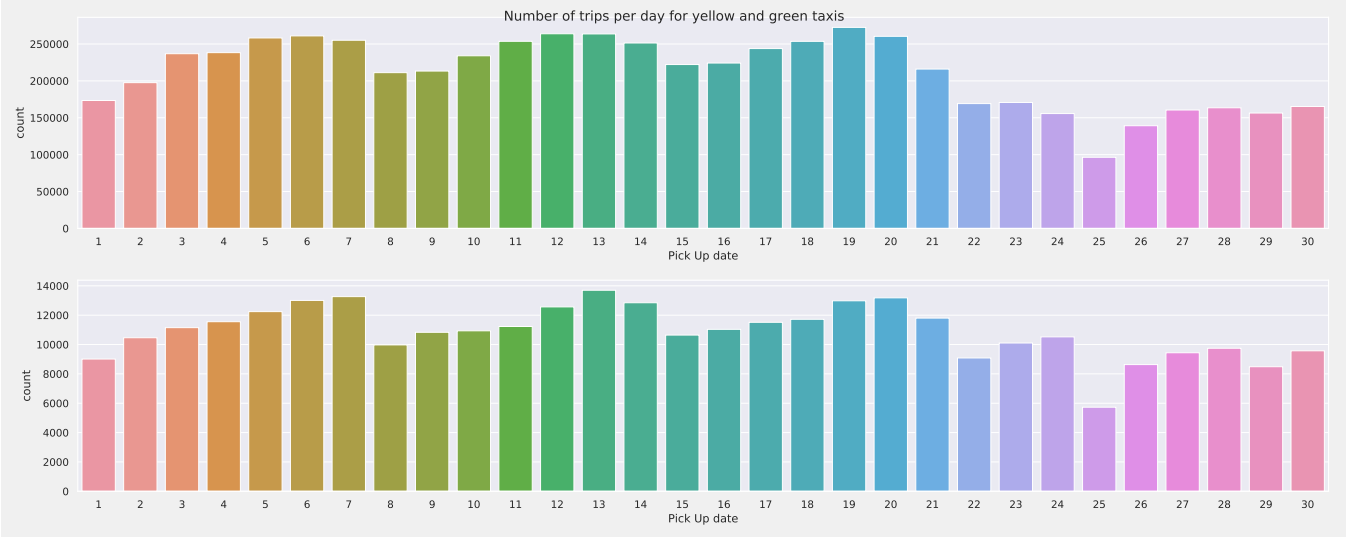

Figure 5: Total trips per day for both taxi types

Average distance per ride for two taxi types

For the initial exploratory analysis, we began to identify the days of the week with the highest distances covered in a day. The results have been depicted in 6. Tuesday and Saturday for Yellow taxis while Monday is the average distance per ride for green taxis. One common pattern is that for both taxi types, Sunday has the highest average distance per ride. In relation with our previous analysis, we found that Sunday has the lowest number of total trips in the week. Combining that results in perspective with this result, we can identify that although less number of people go out on Sunday, however the users taking a ride on Sunday take trips for long distances.

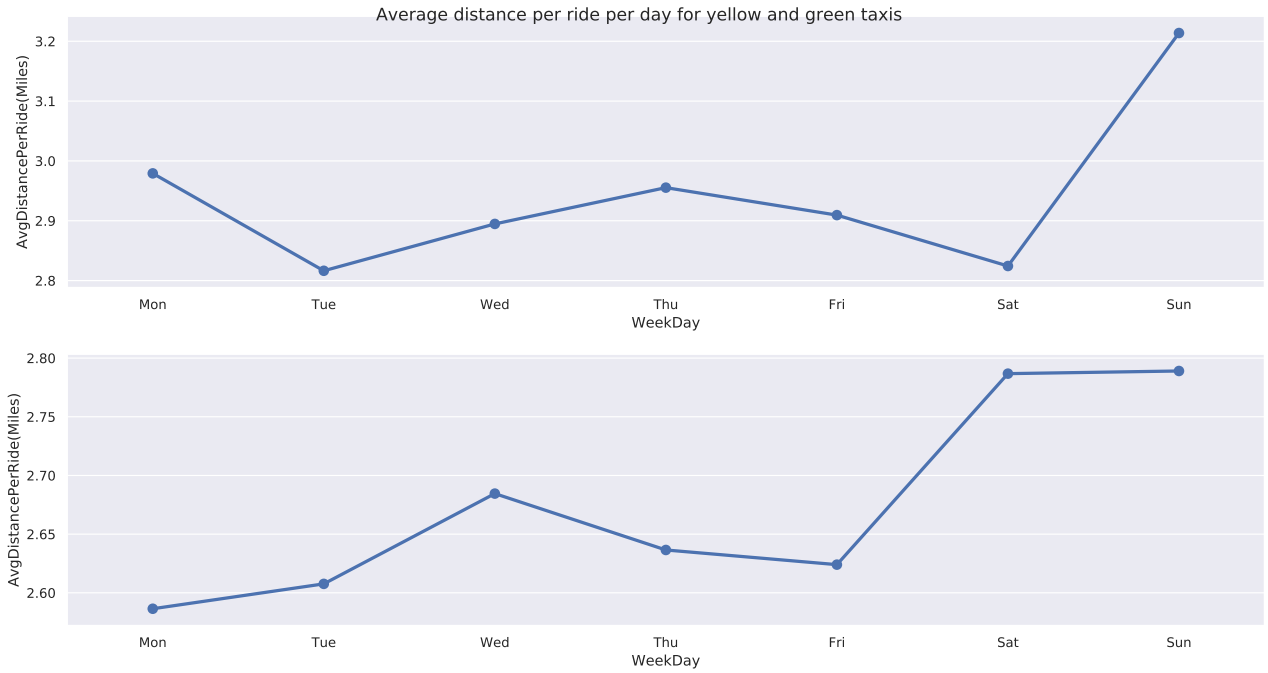

Figure 6: Average distance covered per ride for both taxi types

Analysis on fare amounts and distributions on boroughs

We can try to derive some social and financial insights by analyzing the taxi rides that happen in various boroughs. This could be a method of ranking the boroughs according to how willing are its citizen to pay for long trips or what is the average amount that people in that borough usually pay. The results can be depicted in Fig. 7 and 8, where we plotted the total

amount of sums as well as the average sum paid. For analysis, we have used the data only from the green taxi, as we are not interested necessarily in a comparison of the two companies, but rather a comparison of the boroughs.

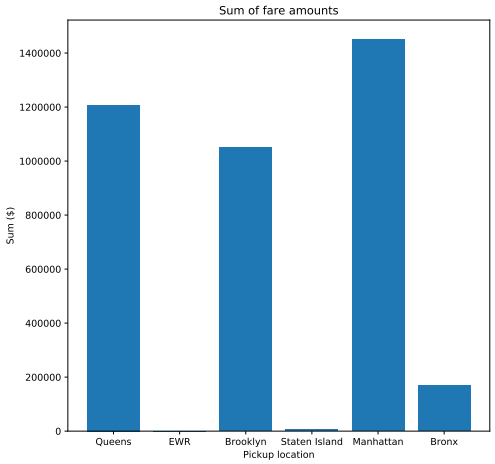

Figure 7: Sum of fare amounts

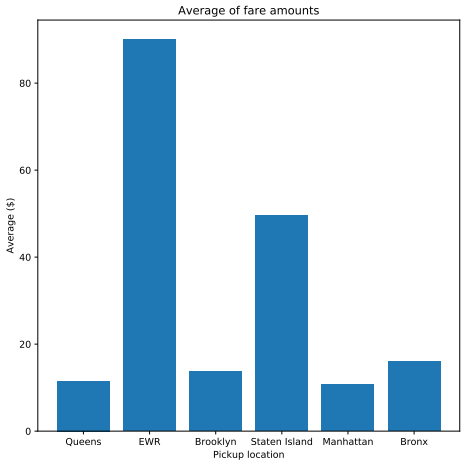

Figure 8: Average of fare amounts

From the above figures, we can deduce some interesting aspects. Although Manhattan appears to have the highest sum of fares, it may seem surprising that the average amount is the lowest. This is also because the majority of the trips are happening in Manhattan and they are of short distance. On the other hand, Staten Island has the lowest sum of fares, but the second average. This is also because Staten Island is actually an island and people need to take longer trips to reach mainland.

Prediction of fare amounts

We have decided to use the machine learning capabilities of Apache Spark as well and we have tried to predict the fare amounts based on selected features. First of all, we have chosen to use regression for this task since the fare amounts variable is a numerical one. Otherwise, we would need to perform a classification task, which would have other evaluation metrics.

When selecting the features, we have decided to use the pickup date, passenger count and trip distance. Even before performing the actual analysis, it is expected to get a somewhat linear function, since the fare amount is usually calculated based on the trip distance, so this can be used as evaluation as well. Before that, we have also analyzed the passenger count variable, by plotting the histogram Fig. 9. It comes as no surprise that the majority of trips are dominated by single customers, which of course influences the prediction to be linear. If we would have an increase in ride sharing, then maybe the predicted function would look different.

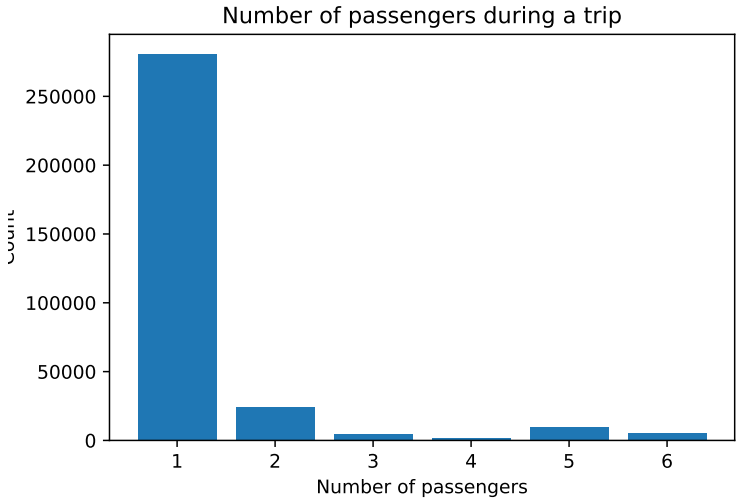

Figure 9: Histogram of passengers

In Fig. 10, we can see the predicted regression model. Again, it is quite normal for it to be a linear function. Data has been split as 70% training data and 30% test data. For the model, we have used a (Gradient-Boosted Trees) GBTRegressor, which is perfect for a regression task. We enforced maximum 30 iterations and a maximum depth of 11 for the tree. To evaluate the model, we used the Root Mean Squared Error (RMSE) and got a result of 0.85 on the test data.

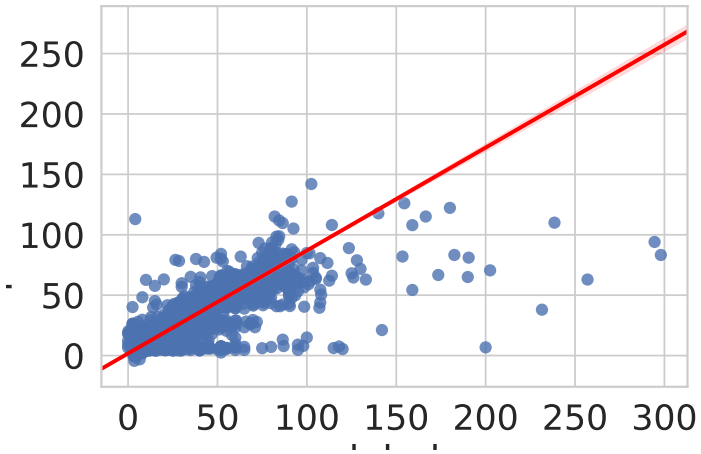

Figure 10: Regression result

Discussion about results

In this project, we performed an exploratory data analysis of two taxi types in the New York City. Also we tried to learn and implement machine learning algorithm for prediction of fare amounts. The results highlight that the most of the rides in the New York City are taken by single passengers. The implication can be to introduce shared taxi rides on such routes with high demand. This can help the company to save the cost of fuel. Also, this can be helpful for customers to get ride services at a lower price. Two taxi service types depicted different usage patterns for the days of the week. One thing common to both taxi types is that Sunday is the day with the lowest number of trips in the week, however the trips taken on Sunday have found to be having highest average distance per ride for both taxi types. The sum of fare amounts is found to be highest for Manhattan borough whereas average of fare amounts is found to be the highest for EWR.

Contribution Report

- Aditya Mehta: Project Description, Data description, cleaning, pre-processing, summary statistics, methods and tools, data analysis for total trips per day, average distance per ride, Discussion about results, conclusion. Overleaf formatting
- Andrei Baraian: Project Description, Data description, cleaning, prediction, analysis of fare amounts, discussion, conclusion
- Baptise Hauteau: Data analysis

Conclusion

Enabled with the knowledge taught in the lectures and exercises, we performed an exploratory study and a prediction task by using the data of New York City Taxi Services.

The challenges we faced were:

- When working with large files in the current cluster, specially when converting to Pandas DataFrame, the cluster kernel runs in out-of-memory issue.

- If two members are working on the cluster simultaneously, the Spark session/application run by one user eats up all the memory and for other person to run the program, they need to either wait for first execution to complete or kill the process.

Although the exercises were helpful, however they introduced concepts at basic level and we had to go beyond that and design our own pipeline to achieve our objectives. Starting from data cleaning, to pre-processing and then using SQL-like queries to group data in meaningful ways. Lastly, we have decided to try to predict the fare amounts by taking in consideration the number of passengers, trip distance and the time of pickup, resulting in a regression task.

This analysis can be used to support various initiatives for TLC such as [5]: accessibility, driver fatique, driver pay, FHV Wheelchair accessibility, FHV trip record data, language access, pilot programs, vision zero.

Bibliography

[1] City of New York. Terms of use. https://www1.nyc.gov/home/terms-of-use.page, urldate = 2020-04-13.

[2] City of New York. Tlc factbook. https://www1.nyc.gov/assets/tlc/downloads/pdf/2018_tlc_factbook.pdf, urldate = 2020-05-13.

[3] City of New York. Tlc trip record data. https://www1.nyc.gov/site/tlc/about/about-tlc.page, urldate = 2020-04-13.

[4] City of New York. Tlc trip record data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page, urldate = 2020-04-13.

[5] City of New York. Tlc trip record data. https://www1.nyc.gov/site/tlc/about/tlc-initiatives.page, urldate = 2020-04-13.

[6] City of New York. Tlc trip record open data. https://www1.nyc.gov/site/tlc/businesses/opendata-current-licensees.page, urldate = 2020-04-13.