Big data & Social networks Final report

Danish Cheema
University of Trento
196267
danishasghar.cheema *at* studenti.unitn.it

Nicolae Puica
University of Trento
204735
nicolaerazvan.puica *at* studenti.unitn.it

Nicola Gilberti
University of Trento
198739
nicola.gilberti *at* studenti.unitn.it

ABSTRACT

The NYC Taxi & Limousine commission has online all the record of the trips to enable everyone to analyze data and find interesting information.
Thanks to this open dataset, everyone can find valuable information, to let the T&L improve the reliability and safety of its system.
This project aim to find interesting data from a small dataset, from January to June 2018, given by the NYC government.
The report start with an introduction of what are the valuable information inside the dataset, and how we can query them.
After that, there is a brief description about how to collect data from the dataset, with some program choices.
It follow a description of the data recovered, with a graphical visualization and a description of the results.

1. INTRODUCTION

In this section it is explained our first analysis of the data and what we think are the queries that can be performed on them to find helpful information. This first approach lead us to the second part with an idea of what our operation has to recover.
By looking at the different fields of the data and how it is composed we came to the conclusion that it would be a nice way to look at the data analysis if we cluster the data from different field perceptive. The complete list of the field elements and how we divide the intervals is given in the Table1 Appendix section.

1.1 Analysis

In the sub section we explore what kind of possible information can be extracted from the data which can be useful from NYC management perceptive and can be relevant to future studies:

1.1.1 Vendors

The Data on which analysis was performed mainly have 2 different Vendor Ids which indicates the TPEP provider that provided the record.

1. Creative Mobile Technologies, LLC
2. VeriFone Inc

Given the two different vendors we try to answer the following queries, how a vendor performs as compare to the other, in different categories.

- Which vendor have recorded more number or records.
- Average Trip distance for different vendors.
- which vendor have more store and forward flag, where flag indicates whether the trip record was held in vehicle memory before sending to the vendor.
- Vendor which save more records with different payment types
- With different trip distances how different vendor record the data
- At different time of the they how the vendor stores the information.
- making a comparison between vendors and there ability of gathering the data between weekdays and weekends.
- Ability of different vendors to provide service in different borough of New York by clustering for pickup and drop off locations.

1.1.2 Day Time

The data contains a field with pickup date and time of the ride. keeping in mind we divided the day in different time sections to observe the traffic flow of the taxes. We try to cluster the data in different categories and try to answer the following queries.

- Given the different time interval when it is that more passenger are traveling together.
- At which time of the day people are traveling longer distances.
- Number of pickups and drop offs for different borough at different times of the day.
- By which mean people like to pay given different time interval of the day.
- Average Tip, Extra and Total Amount payed at different time of the day.

1.1.3 Trip Distance

Data also provides the length of the ride in miles. Given that we divided the distance in different intervals and try to get the meaningful information out of it. We try to cluster the data in different categories and try to answer the following queries.

- Given different distance intervals what is the average a person has paid.
- Number of rides in different distance intervals
- Number of passenger travel together for different distance intervals.
- Average time it took for different trip distances.

1.1.4 Pick-up And Drop-off Locations

Pickup and drop off location is also given for each record and this locations represent different zone of the city. Each zone belong to one of the 5 borough of the city. We try to cluster the data in different categories and try to answer the following queries.

- Number of rides between different boroughs
- Average cost and time it takes to travel from one borough to the other.
- Number of rides in same boroughs.
- High density nodes for different Pickup and Drop-Offs

1.1.5 Other Fields

Different rate codes are used for different rides. Which manifest the final rate code in effect at the end of the trip. Data also contains a field which indicates whether the trip record was held in vehicle memory before sending to the vendor, aka store and forward, because the vehicle did not have a connection to the server. different type of payment modes have been used to pay for the ride. Then we also consider the Fare Amount and explore which kind of interesting stats we can come up with.

- Average and Maximum values and frequencies grouping in the rate code ids.
- Frequency of rides with store and forward flag.
- Number of Rides and averages amount payed with different Payment types.
- Maximum and average fare amount for different intervals
- Number of trips with negative and same number of positive total amount.

2. THE *MAP-REDUCE* PHASE

The raw data given by the NYC government require some reduction to find valuable information.
Our approach started, as explained in Sec. 1, with the definition of the element we wanted to find out. Then, an automatic approach is required to run all the possible map-reduce executable on the trip record data.
Our choice is a Java project, structured in such a way that, for each analysed case, a set of possible reduction is defined. The structure is given by the first key element our reduction has to deal with.
Starting data is a JavaRDD dataFrame (JavaRDD<Row>). Custom classes are used to manage specific type of data like enumerators for fixed options, or ad-hoc classes to manage in an easier way some calculus, like the *MaxValueManager* class. Now the focus shift on the map-reduce phase, in the next sub-sections a detailed explanation of each class and map-reduction set will be explained.

Structured Row

Through all the map-reduce operations, the program generate a structured row similar for each cases in which only specific element are inserted, with some variation from the initial data. The data we want to analyze for each case is most of the time the same, like the passenger count, distances and the costs. Also, for each of them, the research focus often on max values, with their frequency, and averages.
So, we decide to generate a formatted intermediate Row during each map phase to simplify future reduces and to standardize results. To manage the max element we defined a class, MaxValueManager, that is in charge of saving the 3 max values and their frequency. While, for the averages the single element is enough to have the total sum during the reduce and the Row counter let to calculate the mean required.

2.1 Vendor

This sub-section define all the reduction and maps done with a vendor discrimination in mind as first key-mapping.

2.1.1 Vendor ID

The first simple operation is an analysis of max values/ averages/ frequency of all the information grouping only for the *vendor ID*. Starting from the JavaRDD we *map* all the data in a JavaPairRDD in which the first element is the vendorID, while in the second we generate the intermediate Row [2].

$$RDD < VendorID, Row > res \leftarrow \\ data.mapToPair(a- > mapVID(a))$$

The output of the *map* phase became the input of the *reduce by key* phase.

$$RDD < VendorID, Row > output \leftarrow \\ res.reduceByKey((a,b)- > counts(a,b))$$

The output can now be used for analytic, see next sections.

2.1.2 Vendor ID & Distance interval

This analysis search more specific information, with an extra discrimination using *distance Interval*. This element is an ad-hoc *Enum* that define range of distances, like TWO_FIVE meaning from 2 to 5 miles trips. With this map-reduce we can understand if, different vendors have differences or not in the range intervals. For each vendor is also possible to understand their capability/ effort per distance class. [2].

Require: key IS_A$(VendorID, Distance)$

$$RDD < key, Row > res \leftarrow \\ data.mapToPair(a- > mapVIDDI(a))$$

The output of the *map* phase became the input of the *reduce by key* phase. As before, after the reduce, there are data usable for analytics.

2.1.3 Vendor ID & Time interval

This analysis search more specific information, with an extra discrimination using *Time Interval*. This element is an ad-hoc *Enum* that define range of times, like *MORNING* meaning the hour range from 4am to 12am. With this map-reduce we can understand if, different vendors have differences in those hour intervals. For each vendor is also possible to understand their capability/ effort per time class. [2].

Require: key IS_A$(VendorID, Time\ interval)$

$RDD < key, Row > res \leftarrow$
$data.mapToPair(a- > mapVIDTI(a))$

2.1.4 Vendor ID &

As the map-reduces before, and thinking on the queries we find in the section before, a lot of different discrimination can be done, from the *Payment type* to the *Pick up/Drop off Borough*, searching for valuable data that could help each vendor to improve.

2.2 Time interval

This sub-section define all the reduction and maps done with a time discrimination in mind as first key-mapping. Differently from before, the time can be divided in more ways; As cited before intervals can be the division of the time through each day, morning afternoon and night, but it can be also a division of different days. The project allow to differentiate between weekend and weekdays.

2.2.1 Time interval

The first simple operation is an analysis of max values/ averages/ frequency of all the information grouping only for the *Time interval*. Starting from the JavaRDD we *map* all the data in a JavaPairRDD in which the first element is the vendorID, while in the second we generate the intermediate Row [2].

$RDD < Time\ interval, Row > res \leftarrow$
$data.mapToPair(a- > mapTI(a))$

Or

$RDD < WeekendWeekday, Row > res \leftarrow$
$data.mapToPair(a- > mapWW(a))$

The output of each *map* phase became the input of the *reduce by key* phase.

$RDD < VendorID, Row > output \leftarrow$
$res.reduceByKey((a, b)- > counts(a, b))$

The output can now be used for analytic, see next sections.

2.2.2 Time interval & Payment type

This analysis search more specific information, looking for the method used to pay. This element is an ad-hoc *Enum* that define all the payment type accepted, how is defined in the documentation given by the NYC T&L commission.

Require: key IS_A$(Time\ interval, Payment\ type)$

$RDD < key, Row > res \leftarrow$
$data.mapToPair(a- > mapTIPT(a))$

As before, this map can be done on the weekend/weekdays, but also on both the two time discrimination.

Require: key IS_A$(Time\ interval,$
$WeekendWeekday,$
$Payment\ type)$

$RDD < key, Row > res \leftarrow$
$data.mapToPair(a- > mapTIWWPT(a))$

The output of the *map* phase became the input of the *reduce by key* phase. As before, after the reduce, there are data usable for analytics.

2.2.3 Time interval & Borough

This analysis search more specific information, with an extra discrimination using *Borough*. With this map-reduce we can understand how traffic is during time.

Require: key IS_A$(Time\ interval, Pick\ up\ Borough)$

$RDD < key, Row > res \leftarrow$
$data.mapToPair(a- > mapTIPUB(a))$

This case can also be analyzed for the *WeekendWeekday* differentiation.

2.2.4 Time intervals &

As the map-reduces before, and thinking on the queries we find in the first section, a lot of other different discrimination are not shown there, but the code let you find valuable data, from the single zone perspective to the payment option.

2.3 Fare amount

This sub-section define all the reduction and maps done with a discrimination on the fare paid in mind as first key-mapping.
As a Double value, we choose to create an Enumerator to manage classes of the fare paid.

2.3.1 Fare amount

The first simple operation is an analysis of max values/ averages/ frequency of all the information grouping only for the single key element. The result will be a definition of what a user can do with a fixed budget, finding how much far he can go or how the fare is directly connected, or not, with the number of passenger.

$RDD < Fare\ interval, Row > res \leftarrow$
$data.mapToPair(a- > mapFA(a))$

The output of the *map* phase became the input of the *reduce by key* phase.

$RDD < Fare\ interval, Row > output \leftarrow$
$res.reduceByKey((a, b)- > counts(a, b))$

The output can now be used for analytic, see next sections.

2.3.2 Fare amount &

As the map-reduces before, and thinking on the queries we find in the section before, a lot of different discrimination can be done, from the *Time intervals* to the *weekend weekday* distinction. This analysis could help people to organize trips, searching for the cheapest moment.

2.4 Pick up & Drop off location

This sub-section define all the reduction and maps done with a discrimination on where the trip started and ended.
Thus, we could eventually find if there are high dense area of taxi's users or which area is expensive/cheap.

$$RDD < Fare\ interval, Row > res \leftarrow data.mapToPair(a- > mapSamePUDO(a))$$

This operation can be done also to find each route from one borough to the others, to find specification for all type of trips. The output of the *map* phase became the input of the *reduce by key* phase. The output can now be used for analytic, see next sections.

2.5 Other Maps

As we have defined in the section before, there are a lot of map-reduce that can be performed other then the ones defined before. The algorithm perform, in fact, map-reduce operation on Payment-types, on the storeAndForward variable, on the RateCode and on some combination of those with the previous. The format is the same as above, where a map is performed, defining the key and initializing the Row [2] for the next reduce.

3. RESULT DESCRIPTION

After running our algorithm against the provided data we find some interesting stats so we cluster the data with respect to different parameters and the results are saved in the csv files but in order to increase the readability we used graph representation of the data. This section will explain what data represents in the output csv files and how it is represented in the graphs and histograms. Also the interesting facts and trends we found while looking at the results.

3.1 CSV output

As cited in the first part of the section, our program generate a *CSV* file for each map-reduce sequence we are interested in. The csv results are generated starting from a dataset composed by the first half of the 2018 data. Even if they are still hard to read, the data they contain is dense and more accessible in respect of the starting Data Frame. In terms of memory load, the starting Data use more than 4GB, while the CSVs outputs use approximately 100KB, a great reduction. The second point in favor of that output is that they are easily presentable with graphics techniques, like diagrams. Graphics convey essential meaning in a simplified way, in fact we use them to show our result in the next sub-section.
The data on which we run the algorithm, if we talk about number of records recorded by each vendor there are 23 million records provided by vendor one and 30 million by Vendor two. Average trip distance for Vendor one is 2.78 miles and 3.0 miles for vendor two. Vendor one has 0.22 million store and forward records and vendor two only 237. so vendor two is performing way better than vendor one. By looking at the payment type for different vendors both of them have same ration between credit card vs cash. other finding are either not really interesting or discussed in the graph section below.
In the morning 14 million, afternoon 21 million and evening 18 million records are being evaluated. From the Time interval finding it is obvious that more people are likely to travel together in the span of 20 to 4. In the morning and night more people like to pay with with credit card. Average Tip, Extra and Total Amount payed at different time of the day are almost identical.
there are 32.2 million records with zero to two miles, 14.2 million with two to five, 7 million for five to twenty and so on. Average paid is 9, 16 and 40 dollar respectively average time is 9, 20 and 36 minutes respectively. by looking at the stats it we found out that for shorter distance more passenger like to travel together.
If we talk about rate codes there are 52 million rides with Standard rates and 2nd highest is JFK airport with 1.2 million. 37 million rides where customer payed with credit card and 16 million with cash. Out of 54 million records 29859 have negative with same amount of positive records in the data.

3.2 Graphical representation

e In this sub-section we will show the results followed by an analysis. The element shown in this sub-section are for illustrative purpose of what can be assessed with our output, but there are a lot of other analysis executable on them.

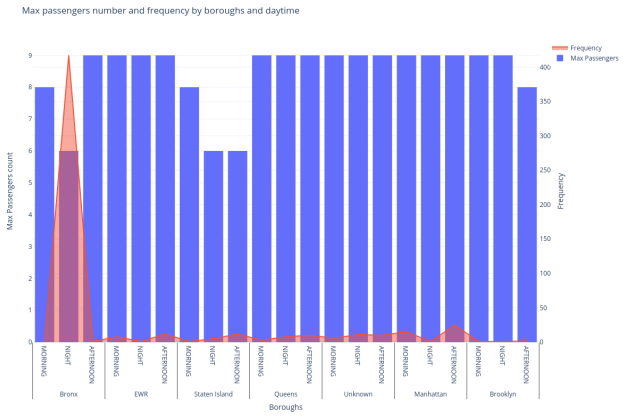

Figure 1: Max passenger & frequency for borough and daytime

As we can observe in the above figure, among the boroughs we notice that usually the max passengers count is 9. But in some cases this number never overtake the 6 max passengers. In this last case we find out that the frequency is much greater. In the case of Bronx in the period from January-June the frequency of the max passengers number (that is 6) is much higher compare to the other. In fig.1 we focus more on the max passengers number, because by this study we can see how many seats a car usually needs. So the goal can have cars in specific boroughs with less spots that implies smaller cars with minor weight and minor gasoline consumption. By combining all this points we can save money in the result. In the Bronx specific case, we find out that is useless to have cars with over 6 sports during the night time.
Another study can be, is the distance from departure spot to the arrival spot related to proceeds?

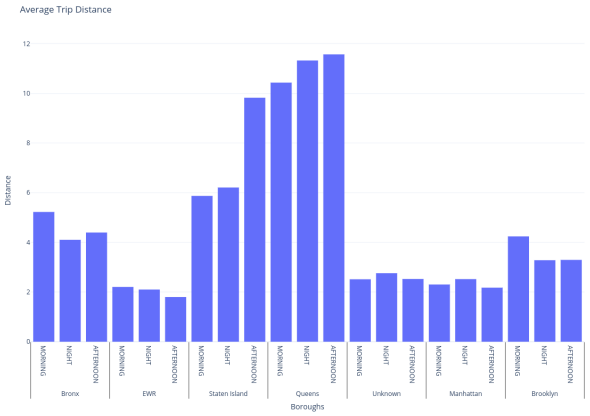

Figure 2: Average trip distance

By observing on fig.2 is instant that Queens borough is the one with the higher distance avg. But lets see the total takings:

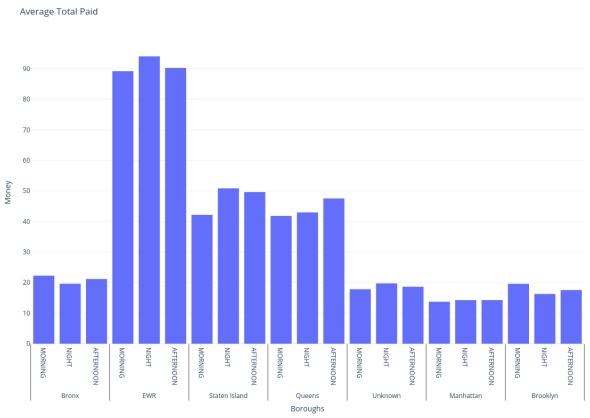

Figure 3: Average total paid for borough and daytime

EWR is the highest money taking. Why? This area is famous for the Newark Liberty International Airport. So even if the distance is less, the presence the airport increase the money collection. So a study to EWR can suggest to have big cars that can host a large number of passengers and baggages. Even if big cars use more gasoline, doesnt matter because the distance will not be long.

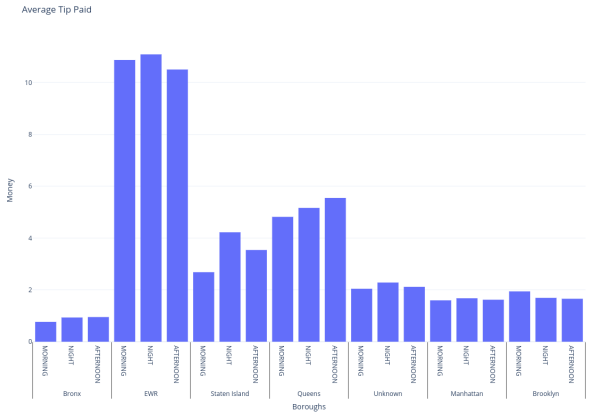

Figure 4: Average tip paid for borough and daytime

In fig.4 we can notice that tips are very frequent on airport shuttles. And even in borough like Queens high tips are not missing.

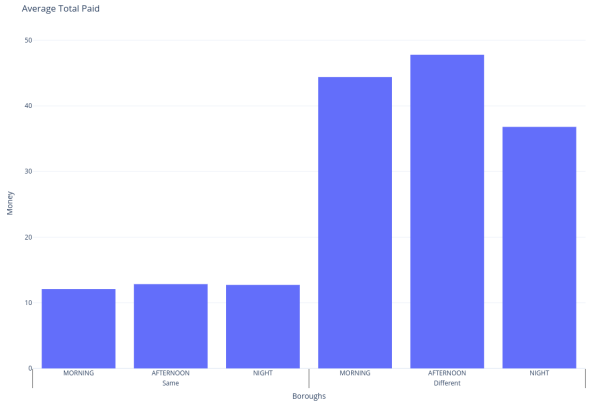

Figure 5: Average total paid & frequency for same/ diff borough

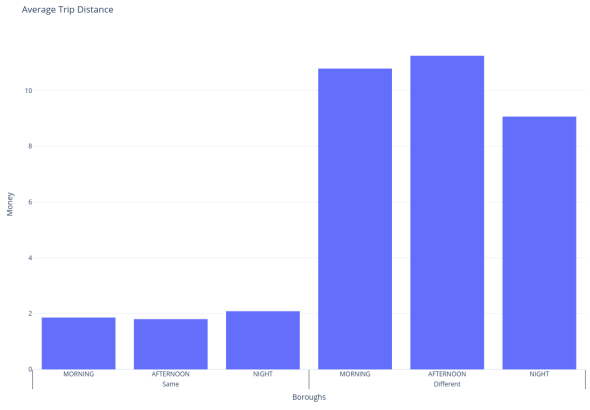

Figure 6: Average trip distance & frequency for same/ diff borough

The above plots (fig.5 and fig.6) gives us a suggestion that is: people that start from a borough and end up in another one, travel a more distance and consequently spend more money. This figure is another source of debate because some data is more obvious than we think. So this is an example of data that needs just a shallow analysis.
People usually prefer spend money by credit card, cash, no charge or dispute?

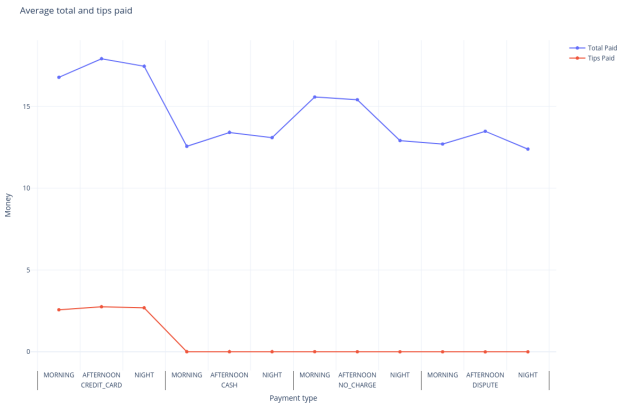

Figure 7: Average total & tips paid

Fig.7 tells us that people that in average spend a large amount of money, they prefer to pay and to give tips with a credit card payment method.

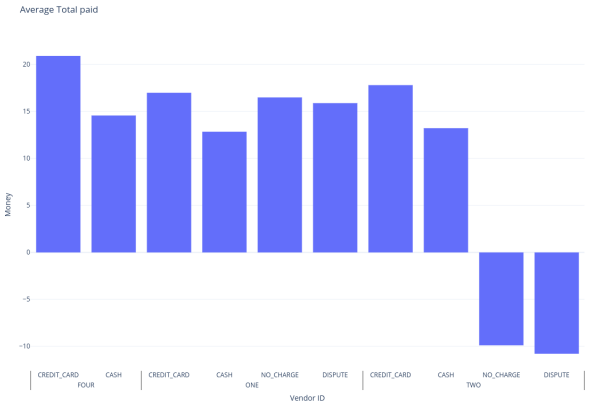

Figure 8: Average total paid to Vendor

We can visualize another interesting fact through fig.8. That is, vendor with id 4 dont accept a no charge and dispute payment method. Maybe is their business method they adopt, luxurious cars or head towards a niche people where spend more money through credit cards (as the above fig. suggest). Instead, the negative numbers can be explained by: vendor with id 2 change money to the passengers just through no charge and dispute types

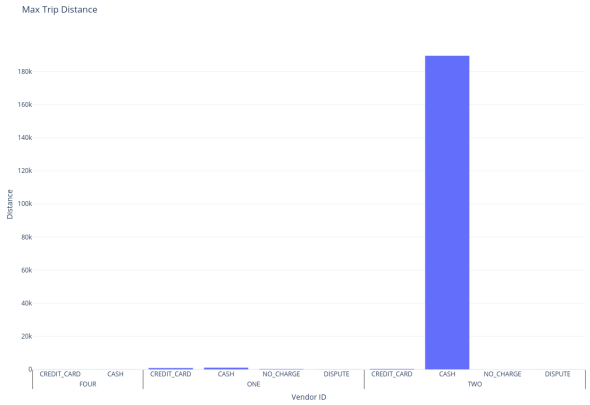

Figure 9: Max trip distance payment

Here we want to show that sometimes it could be wrong and unusual numbers that represents data. If we consider the second highest distance made by vendor 2, it follows:

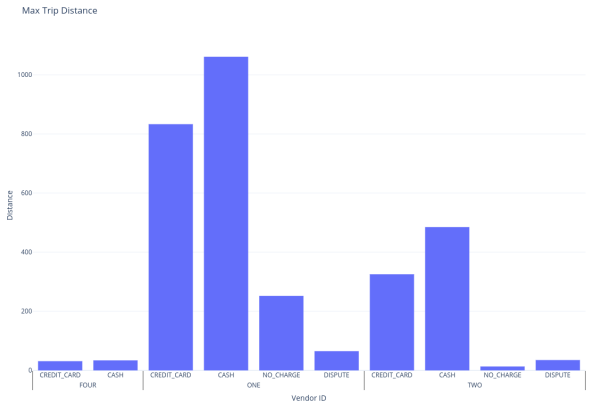

Figure 10: Max trip distance payment V2

Following by the top 3 max trip distance:

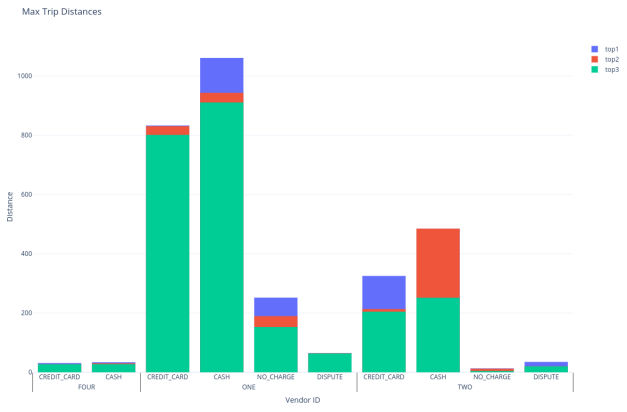

Figure 11: 3 max trip distances payment

So here the schema tells us that vendor one is the most used for long movements inside the city. But we can go deeply inside this diagram by combining the suggestions from fig.8. This join states that vendor 1 is the most used for long trip re-locations. Why? Payment amount and distance advice that people prefer to travel with vendor 1 because they have some discounts for long trips.
Now lets have a look at the number of payments by interval amount payment:

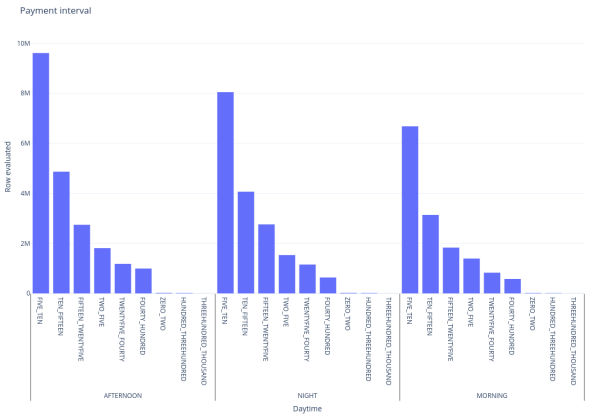

Figure 12: Revenue ranges for daytime

Most common money amount interval is five-ten. And doesnt matter the if it is morning, afternoon or night, because this interval it repeats more than 20M times if we sum them. We can also observe that tiny interval are avoided, maybe people prefer to move themself by feet for really short distances.

4. CONCLUSIONS

The code let us to find in an automatic way a lot of information, that can be shown through graphical tecniques. Java let us to use the Spark environment without difficulties and with Python, and the support of the Plotly library, we generate diagrams accordingly to our ideas. Although queries of the first section find a response with our outputs, there are also other details that can be founded and represented. The size of the dataset is enough to find some interesting information, like the Fig.1, but maybe with a larger dataset, everything could tend towards a defined limit or some data could became more useful.

APPENDIX

Table 1: This data dictionary describes yellow taxi trip data And how we consider the different intervals for different fields

Field Name	Description	Notes and Comments
VendorID	A code indicating the TPEP provider that provided the record. **1= Creative Mobile Technologies, LLC;** **2= VeriFone Inc.**	In the data we observe that there can be up to 4 different vendors
tpep_pickup_datetime	The date and time when the meter was engaged.	morning (4, 12) afternoon (12, 20) night (20, 4)
tpep_dropoff_datetime	The date and time when the meter was disengaged.	morning (4, 12) afternoon (12, 20) night (20, 4)
Passenger_count	The number of passengers in the vehicle. This is a driver-entered value.	
Trip_distance	The elapsed trip distance in miles reported by the taximeter.	zero to two (0, 2) two to five (2, 5) five to twenty (5, 20) twenty to two hundred (20, 200) two hundred plus (200, 100000000);
PULocationID	TLC Taxi Zone in which the taximeter was engaged	In total 265 different zones and 5 different borough
DOLocationID	TLC Taxi Zone in which the taximeter was disengaged	In total 265 different zones and 5 different borough
RateCodeID	The final rate code in effect at the end of the trip. **1= Standard rate** **2=JFK** **3=Newark** **4=Nassau or Westchester** **5=Negotiated fare** **6=Group ride**	There exist some records with id 99 we consider them as false.
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka store and forward, because the vehicle did not have a connection to the server. **Y= store and forward trip** **N= not a store and forward trip**	
Payment_type	A numeric code signifying how the passenger paid for the trip. **1= Credit card** **2= Cash** **3= No charge** **4= Dispute** **5= Unknown** **6= Voided trip**	
Fare_amount	The time-and-distance fare calculated by the meter.	zero to two (0, 2), two to five (2, 5), five to ten (5, 10), ten to fifteen (10, 15), fifteen to twenty five (15, 25), twenty five to forty (25, 40), forty to hundred (40, 100), hundred to three hundred (100, 300), three hundred plus (300, 1000)
Extra	Miscellaneous extras and surcharges. Currently, this only includes the $0.50 and $1 rush hour and overnight charges.	
MTA_tax	$0.50 MTA tax that is automatically triggered based on the metered rate in use.	
Improvement_surcharge	$0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015.	
Tip_amount	This field is automatically populated for credit card tips. Cash tips are not included.	
Tolls_amount	Total amount of all tolls paid in trip.	
Total_amount	The total amount charged to passengers. Does not include cash tips.	