Study of Yellow Taxi in NYC

Ashutosh Mahajan
CIMS, NYU
New York, USA
abm523@nyu.edu

Shubham Divekar
CIMS, NYU
New York, USA
sjd451@nyu.edu

Abstract

The paper proposes a system for travelling efficiently in any major metropolitan area (New York City, in this case) by studying various features of a widely used transport system, Yellow Taxi. This is done using publicly available historical data containing crucial information relating the Taxi trips such as start and end locations of a trip, distance covered, fare required, number of passengers, etc. and insights in the form of direct relationship between the distances, displacements, fares, passenger density, vehicle density, etc. is drawn. These insights can be used by the end users to select and manage their transportation styles; efficient routes (in terms of time, fare, availability, distance to fare ratio) can be provided to the user for better planning. A second phase of this project incorporates restaurant data in form of their location and rating, along with the transport data to provide insights such as the availability of similar restaurants based on user choice (distance, rating, cuisine type). A brief study about the traffic pattern is done in the project to better understand the overall volume flow of passengers, vehicles and money. A comparison between three factors namely, number of trips, average fare, and average speed is done, and the results conform to the pattern that can be usually seen in NYC. Fare prediction is then applied to the feasible restaurant alternatives using random forest regression and a list of least expensive restaurants is provided to the user.

Keywords—analytics, taxi, regression, clustering, density estimation, fare prediction.

I. INTRODUCTION

New York City (or any other metropolitan location) usually face unreliable and traffic ridden transport system. The following paper presents ideas for developing solutions for better transport facilities and software by studying the historical and current transport data. Data in form of various travel information ranging from the source location, destination location, length of commute, time of commute, fare, etc. will be collected and analyzed to gain relevant information like the traffic density along different routes, on a day of week and during a period during the day, economically viable routes and transport options. The study further makes a comparison of different factors such as average fare during the entire day, average speed of transport during the day, and the number of trips in a 24-hour period. This data then in turn can be used by various end users, ranging from commuters, transport department, public and private transport providers to calculate better commute routes thus optimizing the transport in the city.

A second phase of this paper focuses on restaurant data to find a correlation between similar restaurants. This data includes the business tags, cuisines, locations, ratings, etc. which is useful to provide useful insights such as the popularity of the restaurant, number of people using the restaurant, potential location for the growth of the business and providing similar alternatives in case of the unavailability/inaccessibility of a location.

A simple prototype experiment is done taking user input in form of restaurant name. This restaurant is then geocoded (geocoding is also used on the entire restaurant dataset to get the latitudes and longitudes needed to process the dataset with Taxi data) to find its location and then it is compared with the cluster centers found using the taxi data. Alternate restaurants are suggested to the user based on a factor suited to the user, viz., rating of the restaurant, distance from other cluster points (centroids of traffic density), type of cuisine. In the interest of the prototype cuisine factor is selected and alternate restaurant from similar cuisine category are suggested as alternate restaurants to the user. There exists a lot of work done previously in this field of interest thus making it an important issue to be studied and better computational solutions to be provided.

Big data tools like Spark Scala, Scala MlLib are used for the processing of data whereas the visualization is done using Tableau and python. The comparison as discussed above is done using simple python graph plotting, the data for which is the cleaning processes on Spark. Tableau is used for plotting of heatmaps, the data for which is calculated from the big data and is narrowed down to a smaller scale using the pre-defined functionality of K-Means clustering provided by Scala. Fare prediction models are generated using 4 different models, viz., random forest regression, linear regression, ridged linear regression, gradient boosted tree regressor of which only one model gives satisfactory results, details about which are described in further sections.

II. MOTIVATION

This project derives motivation from multiple works previously done. Taxi is a major source of transportation in metropolitan areas and thus has an ever-increasing demand; to accommodate the growing traffic we need to carefully study the patterns of the trips, which can help in improving the infrastructure and resources. Another motivation for the second phase of the project is google maps and its feature of suggesting multiple location choices, in this case restaurants based on factors like

cuisines, star rating, reviews and distance. Fare estimation [4], fare comparison, average speed comparison derived motivation from the need to study traffic pattern and passenger volume movement to understand the basic urban population movement [11].

III. Related Work

[4] The above paper does travel time estimation using historical data based on the GPS in Taxis. The GPS provides dataset in form of locations and route followed with additional information of the distance. They further use this data to develop a model that estimates the link travel time and calculates hourly average of the *urban link travel time*. Further it establishes a relation between the expected path (displacement in our project which is approximately calculated using the latitude and longitude of the source and destination) and the actual path taken by the taxi. The above information is used to derive insights such as the status of Taxi network in the city. They have established that the results achieved using this method are much cheaper to calculate than to use the traditional sensor data. The above paper related directly to our proposed paper idea of calculating the traffic density in New York City (similar to the network status in the above paper); of establishing a relationship between the distance and displacement between two points (source and destination) and then relating it to the general fare between the two locations. This will help us derive insights such as fare estimation and distance to fare ratio of a particular route. A similar approach of Big Data Analysis is used by both the papers though the technologies and methodology used differ widely. The above paper inspires us to design a scalable model using which we can expand the idea to multiple datasets like Uber, MTA Subway and Bus route data, City Bike Data, etc. The hourly average proposed in the summarized paper helped us devise new insight such as passenger density calculation in specific zones or on specific routes which in turn can be directly related to the yelp business classification done as a second phase of our project. Using passenger density instead of vehicle density on a particular route makes more sense when it comes to calculating how much the region and businesses around a location is used[4].

There exists a lot of research on Taxi density and Vehicle density calculations, it sure is helpful when we consider aspects related to traffic of resources used but this is not always directly related to the actual number of passengers moved from one place to another. Thus, it becomes an important procedure to analyze the actual passenger mass movement and calculate its density when it comes to insights like the crowd at a location or the popularity of a destination/venue. Calculation of the passenger density will help us directly get information about *urban human activity and mobility* on daily basis or over a fixed period. Taxi/Vehicle data can be easily converted and applied to calculation of passenger density by attaching additional attributes to the dataset like number of passengers and type of rides (individual rides, pool rides, exclusive rides, etc.). This paper[12] talks about maximizing the driver's profit based on time varying matrices and using dynamic programming to maximize the earnings based on the policy. The paper is based upon the combination of NYC Taxi and Uber datasets. The proposal produces contingency plans to maximize earnings as well as optimizations based on real time data that is gathered from the Uber API. The approach uses two portfolios one of the city which constitutes : Empirical Transition Matrix (F), which gives the probability that a user travels from zone i to zone j, Travel time matrix(T) which gives the times to reach zone j from zone I and a Reward Matrix(R) which gives the reward for the same, which is of the form : earning(i,j) – cost(i,j). All the three matrices are time variant and depend on the time at which they are calculated. The second is mapping the driver. Each driver has a home i0 from which he starts and returns, time frame B which gives the max number of hours of work. At any point the diver can take actions like: Get passenger(a0), Go Home(a1) or Relocate(a2(j)). Based on this a policy can be created using the 3 tuples: (zone, time and action). This optimization can be represented by calculating the Expectation Φ (i0, B, N), for a driver starting from i0, who has B hours of work to do in a duration of N hours. Then use dynamic algorithm for the recurrence. The idea is to simply store the results of subproblems, so that we do not have to re-comupute them when needed later. This simple optimization reduces time complexities from exponential to polynomial. Hence we compute the earnings for all the walks and return the maximum among them which represents the profit of the driver of the day given his budget. These policies can also be changed by using the real time data instead of predicted transition matrix to calculate the earnings dynamically.

[11]Density can be calculated by various method and algorithms. Clustering can be performed to obtain quantized data patterns or grouping can be performed to define a set of density zones depending on the application. In the above papers the authors have used Density based Spatial Clustering Applications with Noise, popularly known as DBSCAN algorithm by feeding it, pick-up and drop-off locations as input. (What difference the additional attributes make? Consider a simple situation where out of 10 taxis in an area zone, 5 are un-occupied and 5 are occupied with single passenger each; whereas in another location we have 3 taxis traveling with their full capacity of 4 passengers, which makes the passenger density higher in Zone2 as compared to Zone1 despite the traffic density being exactly the opposite). As mentioned above this can be easily regulated to work with passenger density matrix by adding tie, distance, speed, occupied status of a taxi, etc. to the input stream. The authors have further classified the entire area of the city into zones with area of 0.015(longitude) × 0.005(latitude), and then have calculated how many people are attracted towards the zone location based on the number of incoming and outgoing taxis. They have classified taxis into two categories, occupied and un-occupied (analogous to our idea of implementing passenger density). As a part of case study, "*the authors have used shopping malls which is an inspiration for our idea of doing the same for restaurants in New York City".* The trip distribution is done using the equation [11] $R^o = (k, l^o, \tau^o)$, where k is the taxi, l^o is the location vector denoted as latitude and longitude and τ is the time period (15mins for our project)

As opposed to the DBSCAN [11] algorithm used in the paper we plan on using K-means clustering as we aim to divide taxi trip into partitions and cluster them to the nearest mean value thus driving the point on density. The partitions will be done based on period of 15minutes. Furthermore, the shopping mall idea of the author is used as an inspiration for the second phase of our project where we center the traffic density around select restaurants in New York City and then based on the density around the location and the identity (factors like rating, type of cuisine, etc.) of the restaurant we provide alternate destinations and fare approximation to the user.

[10]A predictive model is generated using the customer rating to design a recommendation system which works on collaborative and content-based features that are built upon customer and restaurant profiles. The various algorithms used for implementing the composite program are singular value decomposition, hybrid cascade of K-nearest neighbor clustering, weighted bi-partite graph projection, and several other learning algorithms. Using these algorithms, we can identify which restaurant profiles are most suitable to a particular user. Using the same approach, we will be able to identify user profiles and generate a recommendation of restaurants that may be suitable as an alternative to the desired restaurant [10].

IV. Application Design

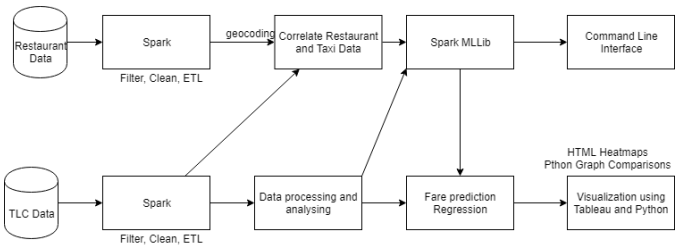

Figure 1 Project Architecture

The application uses Scala Spark, Spark MLLib for the data processing and analysis and uses Tableau and Python for visualization. The application does not have a specific Graphical User Interface but uses the Command Line Interface for actuation.

Interesting properties of the NYC traffic can be obtained and visualized using parameters such as average speed, average fare, and the number of trips. We observe that as the number of trips decreases the average speed increases and vice versa. This can also help in deciding when to begin the journey for lowest travel time. We also observed that the average fare increases during the peak hours (5pm – 7pm). This can be correlated to the increased traffic during this time

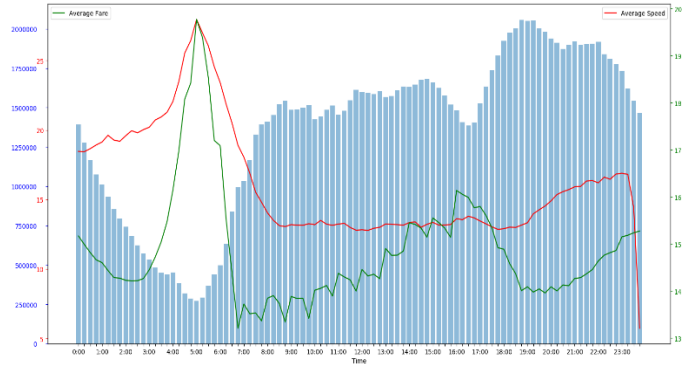

Figure 2: Analysis of different parameters vs Time

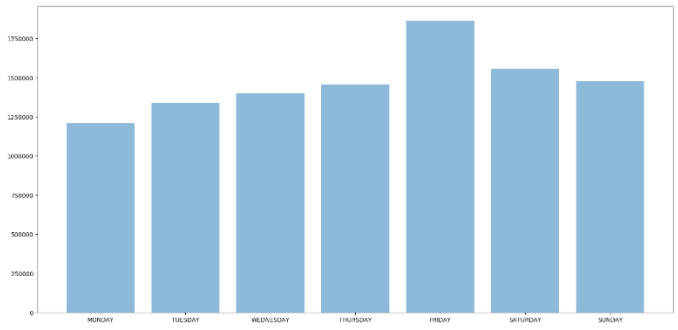

Figure 3: Number of Trips per Weekday

Both datasets are stored on HDFS and passed through Spark for its cleaning filtering and the overall ETL process. Geocoding was performed in python to get the latitude and longitude for restaurants based on their addresses. The output files are stored on HDFS if they are *truly* bigdata or are exported to local system for further analysis. Regression models are used for fare prediction, and KMeans Clustering is used to scale down the data for visualization. Heatmaps are plotted using Tableau on a map of NYC, whereas python is used to plot the graphs showing comparison between different data factors.

Command Line Interface takes input from user in form of the restaurant they want to visit, and the application processes the user input to suggest them alternate restaurants if the selected restaurant is difficult to reach, i.e., lies near a cluster point, thus denoting trafficked area. The user can also input his choice for alternate restaurant, rating or cuisine. For the purpose of this project we are selecting cuisine as a factor and then applying the grade filter as well as the distance filter. The distance filter ensures that the alternate restaurant that is suggested is located in a less trafficked location and the grade filter ensures that the alternate suggested restaurant is similar in quality to the user input restaurant. Then the fare estimation model is applied to get the top 10 least expensive trips to a restaurant among the filtered alternatives. We can use Tableau to visualize the results obtained as the suggestions by applying filters on 2 locations namely User location(Green) and User restaurant(Red).

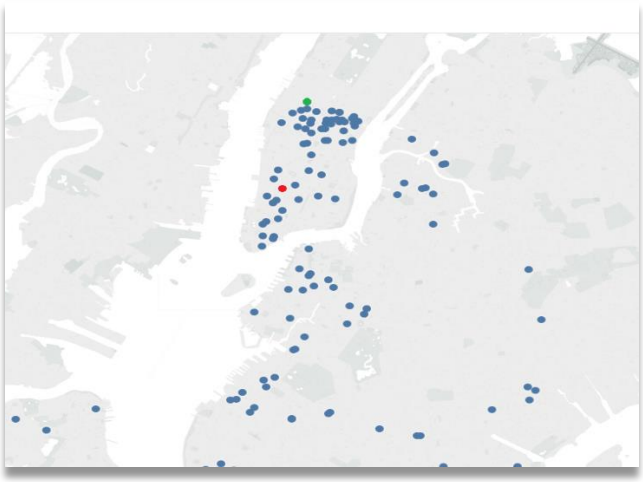

Figure 4: Visualizing User and Suggested restaurants

V. Datasets

Data was collected from publicly available repositories, TLC dataset and NYC Restaurant Inspection data from NYC OpenData. It was exported in tsv and csv format for processing on Spark using RDDs. The datasets used for the experiments of this paper are described below. Both sets were collected in a single batch and moved to the Hadoop data store for its analysis.

Schema Structure:

Column Name (Range): Type (Nullable?)

TLC: Taxi Dataset (Cleaned data schema described)

The data contains collection of trip data. The trip data will have information like source, destination, estimated travel time, actual

travel time, estimated travel route distance, actual travel route distance, fare (including tolls, base fares, over-charge, etc.)

Pickup DateTime (): DateTime (Nullable = False)

Dropoff DateTime (): DateTime (Nullable = False)

Number of Passengers (): Int (Nullable = False)

Distance (0 to 120): Float (Nullable = False)

Pickup Latitude (40.498707 to 40.914389): Float (Nullable = False)

Pickup Longitude (-73.700617 to -74.254881): Float (Nullable = False)

Dropoff Latitude (40.498707 to 40.914389): Float (Nullable = False)

Dropoff Longitude (-73.700617 to -74.254881): Float (Nullable = False)

Total Fare (): Float (Nullable = False)

NYC Restaurant Inspection Dataset (Cleaned data schema described)

Restaurant ID (): String (Nullable = False)

Name (): String (Nullable = False)

Address (): String (Nullable = False)

City (NYC): String (Nullable = False)

Latitude (40.498707 to 40.914389): Float (Nullable = False)

Longitude (-73.700617 to -74.254881): Float (Nullable = False)

Cuisine (): String (Nullable: True)

Grade (A, B, C): String (Nullable = True)

VI. Remediation

Different insights are drawn ranging from traffic density, average fare, average speed, fare spread out across the day, number of trips and their comparison. Actuations such as route suggestions avoiding the trafficked locations or alternate times for travel based on the fare and average speed can be done. The actuation that we are implementing is to suggest alternate restaurant based on cuisine, grade and location. Based on the fare actuations like redirecting the user to alternate Taxi service such as Uber can be done. It is possible to automate the entire processing in future work but the current application required user input from command line and returns results on Console. Insight visualization in reflected as diagrams that describe the data patterns.

The application can be completely automated to read the search results of users based on their restaurant search and then take this as an input to the application and further suggest alternate restaurants. Multiple filters apart from cuisine, user distance and cluster density can be applied, such as star ratings, user reviews, price range of food items, options available and many more. Fare estimation based on regression can also be further modified and converted into real-time by updating the regression model using real-time data and then predicting fare for a specific time of day for a specific route. The results in form of remediation can be seen for this specific actuation by creating a form which accepts the source and destination and then calculated the best route between them and then predict the fare for it using the regression model. For restaurant recommendation system the location of the user can be retrieved automatically by the GPS location of the device the user is using. After a set of alternates is generated, the fare estimation model is run on all the values and the trips from the user to the restaurant which have the least cost are then displayed to the user.

```
sjd451@login-1-1:~
scala> :load AlternateRestaurant.scala
Loading AlternateRestaurant.scala...
defined module AlternateRestaurant

scala> AlternateRestaurant.main(Array())
Enter the restaurant name you wan to go to:
You Entered: MERCI MARKET
Enter your current Latitude
You Entered: 40.638431
Enter your current Longitude
You Entered: -74.034668
The restaurant you are trying to reach may be busy
Enter the desired restaurant rating/grade
You Entered: A
The least expensive trips to similar restaurants :
+--------------------+------------------+
|                name|        prediction|
+--------------------+------------------+
|  BARI SANDWICH SHOP|15.105455635233772|
|    PICKLES & OLIVES|15.147713260099787|
|       SUNSET BAGELS|15.561197585644143|
|THE LUTHERAN HALA...|15.561197585644143|
|   SUNSET RIDGE DELI|15.561197585644143|
|        DELI & GRILL|15.561197585644143|
|            GOUSTARO|15.902992653388315|
|         SUNSET DELI|15.902992653388315|
|CHARLIE'S SANDWIC...|15.902992653388315|
|   DYKER PARK BAGELS|15.902992653388315|
+--------------------+------------------+
only showing top 10 rows
```

Figure 5: CLI for suggesting alternate restaurants

VII. Experiments

The data was prepared using Spark-Scala. For the taxi data the outliers were identified as trips with their starting or ending location outside of NYC, with the same pickup and drop-off locations or time, with fare or number of passengers as zero. The taxi dataset also contained variation in terms of the locations. In the recent dataset the locations are in the form of zones which require lookup to determine the locations; These values were discarded. The pickup and drop off time were then split into weekday and time (to the nearest 15-minute mark) format. Next the trips were grouped according to the time and/or weekday to derive insights from the dataset. Using the time grouping the data was passed through the K-Means clustering algorithm with number of cluster centers as 50 for 100 iterations. Hence 96 sets of centroids were calculated and stored as a comma separated file, indexed according to time. These were visualized using Tableau by applying a filter on the time.

Figure 6: Visualizing cluster centroids

We also experimented with numerous Regression models and selected Random Forest Regression model based on its accuracy. RMSE was used to calculate the efficiency of each model. The regression tree ensemble was built with 20 trees. From this model we were able to determine the most important features for fare estimation; Distance, pickup latitude, drop off latitude, pickup longitude and drop off longitude. Friday and Monday also have a minor impact on the results

```
scala> var importance = m.stages(1).asInstanceOf[RandomForestRegressionModel].featureImportances
importance: org.apache.spark.mllib.linalg.Vector = (106,[0,1,2,3,4,16,27,28,31,32,37,38,44,48,50,51,52,5
3,54,55,58,60,64,66,67,68,69,70,71,72,73,74,76,77,78,79,80,81,82,84,85,86,87,88,90,91,92,93,95,96,98,100
,101,102,103,104,105],[0.5815684279013195,0.20159856302571738,0.07390057593168761,0.06304880332787784,0.
07914689263492525,2.0048249523486 92E-7,3.3387412115391343E-5,2.8250060558670136E-7,4.817148537242719E-7,
5.487272283499667E-7,2.485498829855576E-7,5.3806770619229 89E-8,1.8978765241855513E-6,6.934846121634408E-
7,6.6986898809925545E-6,4.254204911494399E-7,1.3328981828375 66E-6,2.8419291843218 76E-6,1.1692343472791748E-6,1.3660517164385 01E-7,1.3265529414637412E-7,2.4027345346053 17E-7,5.7941866222412 01E-6,1.7163138102984558E-6,3.3241397499270 24E-7,3.8720821328052 79E-6,6.4367307238716 97E-...
scala> importance.toArray.zipWithIndex.map(_.swap).sortBy(-_._2).take(10).foreach(x => println(x._1 + "
-> " + x._2))
0 -> 0.5815684279013195
1 -> 0.20159856302571738
4 -> 0.07914689263492525
2 -> 0.07390057593168761
3 -> 0.06304880332787784
101 -> 2.4562640468339284E-4
102 -> 1.7995736017152318E-4
84 -> 4.221425028866497E-5
27 -> 3.3387412115391343E-5
80 -> 2.8530338062927593E-5
```

Figure 7: Feature Importance

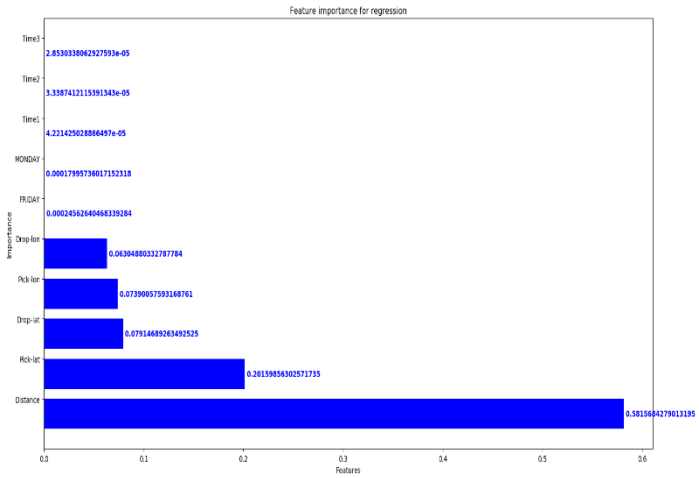

Figure 8: Visualizing feature importance

For restaurant analysis we tried to use the Yelp Dataset but after the cleaning process, it was discovered that it contained only 22 restaurants that were in NYC. The DOHMH New York City Restaurant Inspection Results dataset did not contain the location in form of global coordinates. Hence, we used the open source ArcGIS API to geocode the address of the restaurants and stored the derived latitude and longitude into the dataset. The dataset contained multiple entries for a particular restaurant corresponding to multiple inspections. Such entries were combined to reflect the most recent inspection. Dataset also contained different formats for storing multi-cuisines, which was unified to / separated cuisines.

During the course we tried to calculate the percentage of efficient trips. This can be done by calculating the 'city block' distance between the end points and then comparing it with actual distance. While calculating the city block distance, the inclination of Manhattan with the equator must be considered this proposed a difficulty as we could not calculate this distance correctly, which can be improved in the future. The project could benefit from real-time traffic data which is currently not available. We also observed the number of filters available to the users for selecting an alternate restaurant are also limited. Hence a user ratings or star- reviews could greatly benefit the algorithms and provide the user with robust alternatives.

Through this course and project, we learnt various big data tools such as Spark-Scala, Tableau and visualizations in Python and Geocoding. We were also able to test out and evaluate different machine learning models used for clustering and regression. We also got familiar with various operations like filtering, joining, mapping etc. on RDDs and dataframes in the process. The efficiency of Spark as a big data tool could be seen by the distribution and efficiency of the tasks on the cluster.

VIII. Conclusion

The above experiments provide us with some interesting results that depict the pattern of Taxi traffic in NYC. The comparison between average fare, average speed and the number of trips in a day conform with the usual traffic pattern seen in NYC. With lower number of trips per hour the fare increases because of lower demand and the average travel is decreased owing to increased speed. On the other hand as the number of trips increase the average speed increases despite an increase in the average volume movement of passenger; and because of higher demand the average fare value goes down to make the taxi trips more affordable to a higher portion of population. These results can directly be used by regular passenger to carefully plan their daily trips, by taxi companies to further improve their services and by the transport authorities to provide the city with essential infrastructure which can help decrease the congestion in densely trafficked locations. These insights can then be directly used for automated route planning, real-time fare and speed calculations, calculate the estimated time of arrival, etc. Fare predition model gives an approximate accuracy of 82% achieved by improving the regression model and selecting an appropriate type of model. Restaurant recommendation system provides with accurate results based on the filters which conforms with manually visualized data. Alternate locations are recommended that are near the original location of the user but further away from the traffic cluster centers thus allowing an easy but similar access time and fare. The map results showed above clearly denoted higher density near the original location. Fare prediction for a trip from the user location to the suggested alternate locations show a fair correlation between the different values calculated thus confirming the desired result.

The entire application when constructed as an end-to-end usable product provides an useful tool that will help passengers and customers of restaurant to paln their desired trip to their favorite restaurant with minimal time and money spent. Apart for the above-mentioned portion of the end users, restaurant owners, taxi drivers and transport authorities can carefully

study the result achieved in our project to monitor their businesses and improve their service and provide the customers with essential infrastructure better suited to their needs.

IX. Future Work

The project has a vast scope in terms of expansion. The primary future work involved automating the entire application and combining all the functionalities into one single end-to-end program that inputs the processed data, applies the business logic and then output the desired results.

For fare prediction the accuracy of the regression model can be improved to get better fare result. Moreover, additional parameters can be included such as the route taken by the trip, the traffic condition on the trip route, historical fare patterns, locations, passengers, type of trip, etc. Additional filters can be applied for restaurant recommendation system like user rating, user reviews, star rating, location accessibility, seat availability, etc. This will narrow the alternate options provided to the user. An additional feature of allowing the user to choose the filters can be incorporated to provide user with a friendly interface. Multiple data-sets such as weather, pricing, safety can be joined with the existing dataset to further refine the results and expand the scope of the project.

Acknowledgment

We would like to take this opportunity to thank our professor Prof. Suzanne McIntosh to provide us with insightful information and guidance throughout the project-work. We would like to extend our gratitude towards Rajat Agarwal for providing us with crucial feedback for improvising the project insights. Finally, we would like to thank the NYU HPC team for providing us with a platform for data storage and data processing.

References

1. T. White. Hadoop: The Definitive Guide. O'Reilly Media Inc., Sebastopol, CA, May 2012.
2. High-Performance Spatial Query Processing on Big Taxi Trip Data Using GPGPUs <https://ieeexplore.ieee.org/abstract/document/6906763
3. A big data driven model for taxi drivers' airport pick-up decisions in New York City <https://ieeexplore.ieee.org/abstract/document/6691775>
4. Urban link travel time estimation using large-scale taxi data with partial information <https://www.sciencedirect.com/science/article/pii/S0968090X13000740>
5. Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips <https://ieeexplore.ieee.org/abstract/document/6634127>
6. Taxi data in New York city: A network perspective <https://ieeexplore.ieee.org/abstract/document/7421468>
7. Visualizing Yelp Ratings: Interactive Analysis and Comparison of Businesses <http://vis.berkeley.edu/courses/cs294-10-fa14/wiki/images/f/f4/Datavis.pdf>
8. Forecasting Ratings and Review Counts for Yelp Businesses <http://snap.stanford.edu/class/cs224w-2017/projects/cs224w-60-final.pdf>
9. CompRec-Trip: A composite recommendation system for travel planning <https://ieeexplore.ieee.org/abstract/document/5767954>
10. Yelp food recommendation System <https://pdfs.semanticscholar.org/8b2b/ada22181916196116f1711d456ea212f2b3b.pdf>
11. Uncovering urban human mobility from large scale taxi GPS data; Jinjun Tang, Fang Liu, Yinhai Wang, Hua Wang <https://www.sciencedirect.com/science/article/pii/S0378437115005853>
12. Putting Data in the Driver's Seat: Optimizing Earnings for On-Demand Ride-Hailing <https://dl.acm.org/citation.cfm?id=3159721>