Is Lyft really cheaper than taxi?

Yahui Ke
kyahui3@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

Xueyang Zhang
xzhang870@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

Sikai Zhao
sikaizhao@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

Hao Wang
hwang794@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

Yaoxu Xiao
yxiao356@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

ABSTRACT

A taxi fare prediction model was build based on New York City yellow taxi trip open data provided by TLC. An interactive web application is built for dynamic taxi fare prediction. Data visualization were performed for taxi fare analysis and for taxi fare prediction and Lyft price comparison with the effects of time or distance.

KEYWORDS

taxi, Lyft, big data analysis, prediction model, visualization

1 INTRODUCTION

The NYC Taxi and Limousine Commission (TLC) provides open data of yellow taxi trip records. Behind those trip data, there must be a lot of interesting stories to tell. How much more expensive would it be to take a taxi during rush hours? Is it cheaper to take a taxi on weekends or on weekdays? And what potential factors might affect the trip price fluctuation? To find out the answers, the New York yellow taxi data of the year 2018 was acquired. A taxi fare prediction model was developed and trained using the data. A large number of trips categorized into peek hours versus non-peak hours or workdays versus weekends were called and the corresponding taxi fare was rendered using our prediction model. Besides, a New York City taxi fare prediction web application was developed, allowing users to get the taxi fare prediction by selecting the pick-up and drop-off locations.

In addition, nowadays, peer-to-peer car hailing offered by ride-hailing companies like Uber and Lyft has become an increasingly popular approach of transportation services in cities. As alternative transportation choices, are they cheaper than Taxi? The trips fares of sample trips from Lyft versus taxi (with our taxi fare prediction model) in New York were acquired and compared.

This project is interesting and essential as we, all as customers, care about how much we need to pay for the daily transportation.

2 LITERATURE SURVEY

Apache Spark is one unified engine for diverse big data processing [1]. It was developed using Scala but can support multi-language APIs including Scala, Java, Python and R[2]. Spark uses Resilient Distributed Datasets (RDDs) to store data, which can be stored in memory between queries. This enables a large increase in data processing speed compared with MapReduce and DAG engines[3].

In chapter 10 of *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*[4], the author introduced a Python library which have calendar-related functionality. By using this datetime library, we can get week number by combine the string into date in Python. Also, we can use the build in function to convert between string and datetime.In Online Map Application Development Using Google Maps API, SQL Database, and ASP .NET, the author[5] shows how to get longitude and latitude using JavaScript which can be used to generate a location ID. In Point in polygon strategies[6], the author provides three strategies to decide whether a point is in a polygon or not, crossing test, angle summation and triangle fan. The crossing test is the simplest way which will be implemented in our algorithm to convert longitude and latitude into location ID.

A deep neural network was built by Simon[7] using six different features as input to predict the taxi fare using six different features such as time, distance. Rangapuram presents a forecasting method that parametrizes a particular linear State space models(SSM)[8] using a recurrent neural network (RNN) for time series problem. Ahmed provides a systematic review of the current time series forecasting model using neural networks[9]. Ahmed tries to define the most important theoretical contributions in the development of artificial neural network models for the forecasting of non-linear time series.

Cullen Schaffer used experiments to prove that cross-validation[10] leads to better performance that it guards against the chance of catastrophic performance. The popular method, 10-folder cross-validation, would be a suitable way to test the prediction model of our project with low variation. Vafeiadis also performed and evaluated five machine learning algorithms based on cross-validation and the Monte Carlo simulation[11] was used to determined the most efficient parameter combinations. Monte Carlo simulation can be used to find the influence of the parameters to the prediction results and model efficiency. Vora tested prediction models based on large scale dataset[12] and used Precision and Recall methods to evaluate the results. The parameters such as TP, FN can also be defined in our model to evade the performance of the algorithms. Traditional and modern clustering algorithms are evaluated by Xu[13] based on parameters of distance or similarity measurement and evaluation indicators.

Integrative Genomics Viewer (IGV) could efficiently interact data with different views[14]. Also, a zero-drift localization system[15] could help to localize and reuse the map in the already mapped areas with more accuracy. This zero-drift system will achieve in our NYC map with high precision in specific pick up and drop off location. It can also be used to match an object or a scene with obstacles and outliers from the data sets images. The approach of polygonal mesh reconstruction errors to obtain higher accuracy of this approach could apply in our visualization process[16].

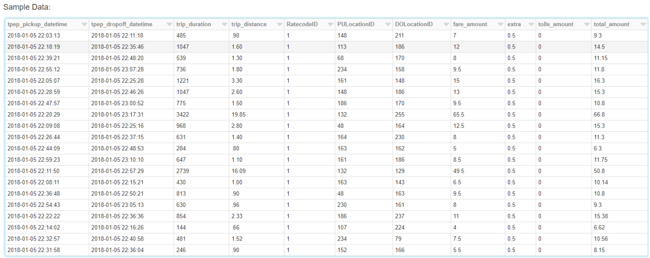
3 METHODOLOGY

Data Cleaning

The New York yellow taxi data of the year 2018 was acquired downloaded from TLC open data [17]. Data cleaning and processing were performed with Spark/Scala on Databricks. Unnecessary columns like payment_type and tip_amount were removed and invalid rows were filtered out. Invalid rows are for example, the rows with RateCodeID == 5 as the fare for those trips are negotiated fare and the rows with 0 or negative fare_amount/total_amount as those data might indicates a refunded trip and thus is not useful for our purposes. Besides, an extra column 'trip_duration' was created by calculating the difference between tpep_dropoff_datetime and tpep_pickup_datetime. Spark is a unified engine for diverse big data processing and is suitable for our purpose of analyzing a whole year of yellow taxi trip data. A sample of the processed data is shown in Fig_1:

Date&Time Clustering

Used python to import the modified data and used split datetime function to generate a week id for



Sample Data:

tpep_pickup_datetime	tpep_dropoff_datetime	trip_duration	trip_distance	RatecodeID	PULocationID	DOLocationID	fare_amount	extra	tolls_amount	total_amount
2018-01-05 22:03:13	2018-01-05 22:11:18	485	.90	1	148	211	7	0.5	0	9.3
2018-01-05 22:18:19	2018-01-05 22:35:46	1047	1.60	1	113	186	12	0.5	0	14.5
2018-01-05 22:39:21	2018-01-05 22:48:20	539	1.30	1	68	170	8	0.5	0	11.15
2018-01-05 22:55:12	2018-01-05 23:07:28	736	1.80	1	234	158	9.5	0.5	0	11.8
2018-01-05 22:05:07	2018-01-05 22:25:28	1221	3.30	1	161	148	15	0.5	0	16.3
2018-01-05 22:28:59	2018-01-05 22:46:26	1047	2.60	1	148	186	13	0.5	0	15.3
2018-01-05 22:47:57	2018-01-05 23:00:52	775	1.50	1	186	170	9.5	0.5	0	10.8
2018-01-05 22:20:29	2018-01-05 23:17:31	3422	19.85	1	132	255	65.5	0.5	0	66.8
2018-01-05 22:09:08	2018-01-05 22:25:16	968	2.80	1	48	164	12.5	0.5	0	15.3
2018-01-05 22:26:44	2018-01-05 22:37:15	631	1.40	1	164	230	8	0.5	0	11.3
2018-01-05 22:44:09	2018-01-05 22:48:53	284	.80	1	163	162	5	0.5	0	6.3
2018-01-05 22:59:23	2018-01-05 23:10:10	647	1.10	1	161	186	8.5	0.5	0	11.75
2018-01-05 22:11:50	2018-01-05 22:57:29	2739	16.09	1	132	129	49.5	0.5	0	50.8
2018-01-05 22:08:11	2018-01-05 22:15:21	430	1.00	1	163	143	6.5	0.5	0	10.14
2018-01-05 22:36:48	2018-01-05 22:50:21	813	.90	1	48	163	9.5	0.5	0	10.8
2018-01-05 22:54:43	2018-01-05 23:05:13	630	.96	1	230	161	8	0.5	0	9.3
2018-01-05 22:22:22	2018-01-05 22:36:36	854	2.33	1	186	237	11	0.5	0	15.38
2018-01-05 22:14:02	2018-01-05 22:16:26	144	.66	1	107	224	4	0.5	0	6.62
2018-01-05 22:32:57	2018-01-05 22:40:58	481	1.52	1	234	79	7.5	0.5	0	10.56
2018-01-05 22:31:58	2018-01-05 22:36:04	246	.90	1	152	166	5.5	0.5	0	8.15

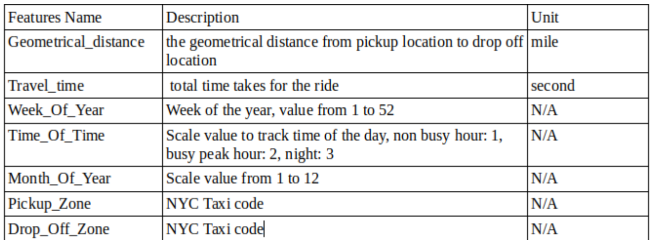
Figure 1: Sample data after processing

training. Trip time was extracted and divided into different groups by 10 minutes.

Model Building

1. Prediction model Features

To predict the taxi fare, we used geometrical distance and travel time. According to NYC taxi fare model[18], the time of day, whether it's peak hour or it's a holiday also impact the fare. Also, rides to airport typically have different prices than normal. We introduce a feature called airport. Fig_2 shows the description all the features we will use for the taxi fare prediction:

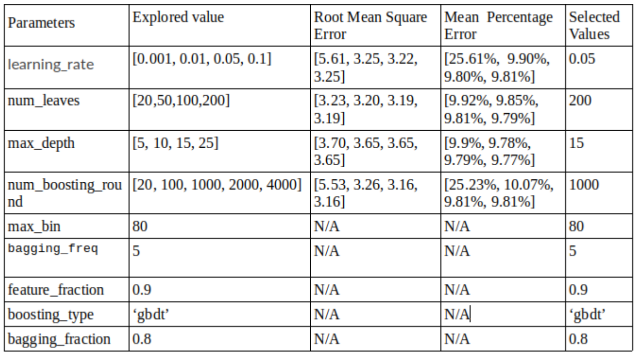


Features Name	Description	Unit
Geometrical_distance	the geometrical distance from pickup location to drop off location	mile
Travel_time	total time takes for the ride	second
Week_Of_Year	Week of the year, value from 1 to 52	N/A
Time_Of_Time	Scale value to track time of the day, non busy hour: 1, busy peak hour: 2, night: 3	N/A
Month_Of_Year	Scale value from 1 to 12	N/A
Pickup_Zone	NYC Taxi code	N/A
Drop_Off_Zone	NYC Taxi code	N/A

Figure 2: the features for taxi fare prediction

2. Training model

A specific Machine Learning library from Microsoft called LightGBM was used to prediction model training. The LightGBM has faster training speed and higher efficiency, lower memory usage, better accuracy. It also supports parallel and GPU learning which is capable of handling large-scale data[19]. The training was performed in Microsoft Azure Cloud Computing Platform. Specifically, a LGBMRegressor will be used. Fig_3 shows the parameters need to be explored and its best values being selected when we are doing parameter tuning for the prediction model. 80% of the data are used for training and validation, 20% if the data are used for testing. Using the selected parameters, we have the prediction root mean square error 2.98 USD and mean percentage error 9.77% for the testing dataset. The model was saved as binary file for later usage in website backend.



Parameters	Explored value	Root Mean Square Error	Mean Percentage Error	Selected Values
learning_rate	[0.001, 0.01, 0.05, 0.1]	[5.61, 3.25, 3.22, 3.25]	[25.61%, 9.90%, 9.80%, 9.81%]	0.05
num_leaves	[20,50,100,200]	[3.23, 3.20, 3.19, 3.19]	[9.92%, 9.85%, 9.81%, 9.79%]	200
max_depth	[5, 10, 15, 25]	[3.70, 3.65, 3.65, 3.65]	[9.9%, 9.78%, 9.79%, 9.77%]	15
num_boosting_round	[20, 100, 1000, 2000, 4000]	[5.53, 3.26, 3.16, 3.16]	[25.23%, 10.07%, 9.81%, 9.81%]	1000
max_bin	80	N/A	N/A	80
bagging_freq	5	N/A	N/A	5
feature_fraction	0.9	N/A	N/A	0.9
boosting_type	'gbdt'	N/A	N/A	'gbdt'
bagging_fraction	0.8	N/A	N/A	0.8

Figure 3: parameters and initial values

3. Prediction

User's input will be pickup and drop off location. The first step is to generate the prediction features for the model. It consists a few steps: converting input location into NYC Taxi Zone Id, getting the trip estimated time and geometrical distance using Google Map API, and then put all the input data into our trained model. Prediction work flow is shown in Fig_4.

Visualization

A interactive website (Fig_6) was designed to provide interactive experience to customers. A google map location-marking function was implemented based on Google map API, the data such as start location ID, destination ID and time can be collected and passed to the prediction model. The estimated price can then be showed on the website.

Framework

FLASK is used to design the framework of the website. Flask is a lightweight WSGI web application framework, providing developers with more

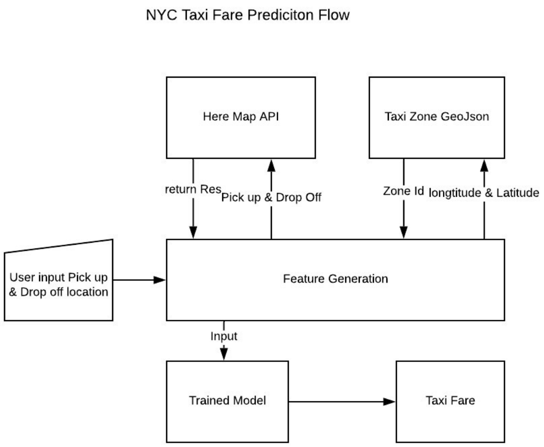


Figure 4: workflow

freedom and convenience. It is much easier to get started and scale up to complex applications compared with other frameworks like Django as an example, which is extremely all-inclusive and is relatively more difficult to implement. Django offers many more functions that we do not need, causing it over-weighted. Therefore, flask is suitable to demand of our website.

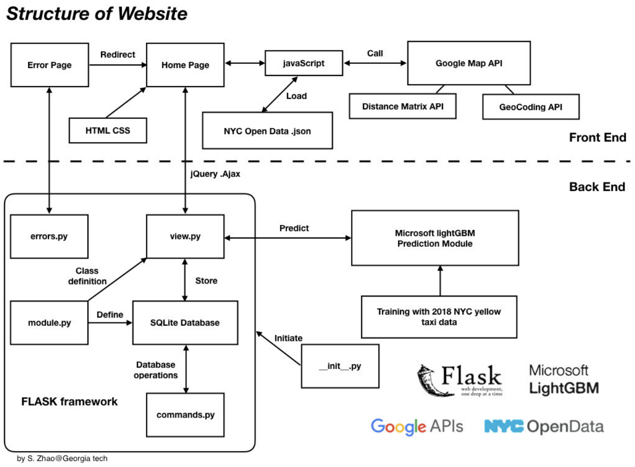
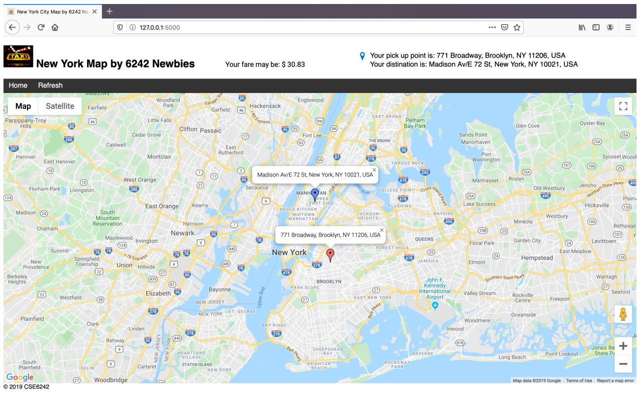


Figure 5: Web application structure

Front end

For the frontend were built based on HTML and CSS. A New York map based on Google map is ingrained in the page. Google map API is called inside JavaScript to get location information of longitude and latitude. And D3.js is used to transfer the longitude and latitude into New York City taxi zone, which is provided by NYC Open Data in the form of .JSON file. Google Distance Matrix Service API is used to obtain the estimated trip duration and distance. All the trip related information and current time are sent to the backend through jQuery .Ajax POST and GET as the input parameters of production model. Same parameters were used to get Lyft's estimated price.



New York Map by 6242 Newbies

Your fare may be: $ 30.83

Your pick up point is: 771 Broadway, Brooklyn, NY 11206, USA
Your distination is: Madison Av/E 72 St, New York, NY 10021, USA

Madison Av/E 72 St, New York, NY 10021, USA

771 Broadway, Brooklyn, NY 11206, USA

Figure 6: Web application demo screenshot

Backend

The current time is transferred to three parts, namely "weekdays", "months" and "time of the day", which describe the features of the time. "Weekdays" feature can help evaluate the difference between weekdays and weekends, "months" feature can help evaluate the difference among seasons in one year. And "time of the day" can help evaluate the difference between rush hour and normal hour. The regression prediction model written in Python is migrated to the FLASK structure and take in the parameters transferred through jQuery to do prediction. The result of estimated price is returned to the front end to display, which is clear and eye-catching on the home page.

The database models of the website are Trip and User class. The Trip class is important to our website because it consists of all the information useful to the prediction model, such as trip information and predicted price. And the User class include id and username, which makes the design of login function convenient. The database of SQLite is ingrained in the website to store all the search history of trips. Initiate and drop database command are provided.

Besides the interactive website, we also design two experiments to analysis the data after processed by selected the statistical significance variables (e.g., distance, time). Furthermore, Tableau, a powerful data visualization software with excellent overall performance and highly security, was used to for charts generation. Different types of visualization with appealing color format and layout can be created directly.

4 EXPERIMENTS

Two problems are expected to be solved through the experiments. First, which factors of a trip will effect the fare obviously. Second, in which condition, Lyft is cheaper than taxi in NYC or vice versa. Two parts of experiments are designed for the questions.

Characteristic of taxi fare

The first part of experiment was implemented to evaluate the taxi fare difference between rush hour versus non-peak hour and workday versus weekend. To compare the effect of rush hour, we set four different time periods: 4:00-6:00(early morning), 8:00-10:00(morning rush hour), 16:00-19:00(night rush hour) and 21:00-23:00(midnight). In order not to affect the accuracy of the experiment, we will randomly generate location id, month and weekday, then trip distance and trip duration are computed by Google Distance Matrix API. Total 800 rows random trips were generated, then were put in our prediction model. In those trip data, the only difference will be the time zone. The result is that early morning and midnight the fare are almost same. And the comparison of morning and afternoon rush hour is shown below as the Fig_7:

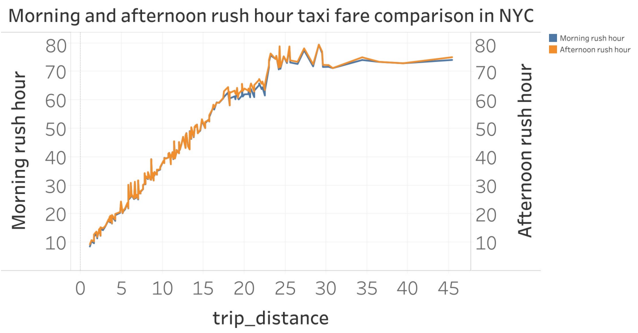


Figure 7: Morning and afternoon rush hour taxi fare comparison in NYC

Data indicates that during the morning rush hour, the traffic has the similar trip patterns as people go to work. Afternoon rush hour traffic is more diverse as people different trip purposes with different destinations. That's why the fare of afternoon trip which have the same trip distance is a little bit higher than the price in the morning rush hour in the mid range distance.

The weekday and weekend taxi fare was compared. In the weekday group, the parameter "weekday" will be a random integer between 1 and 5. In the weekend group, the same parameter will be set to 6 or 7 to represent weekend. Location id, month and time zone were generated randomly. 400 rows random trips were generated and put into our model. The comparison of weekday and weekend is shown in Fig_8:

The data suggests that weekday generally has more uniform fare prices for the same distance. It might be explained by the fact that that weekday traffic has the same trip patterns as people go to work and back home, while weekend traffic is more diverse as people have different trip destinations. Weekend, weekday along has no big effect to taxi fare.

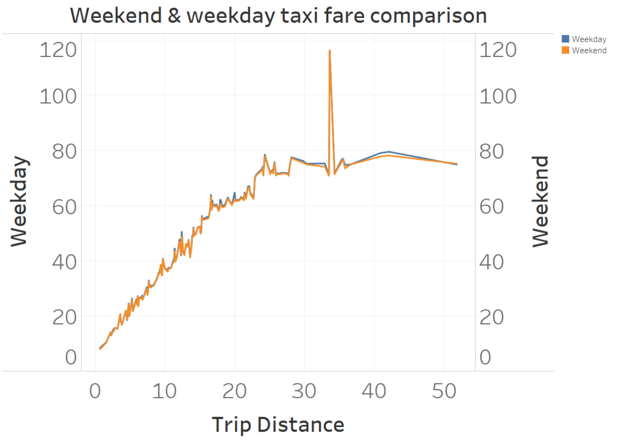


Figure 8: Weekend and weekday taxi fare compression

However, due to the lack of rush hour in weekend, the fare in the same distance in weekend is lower than the trip in weekday.

Lyft and taxi fare comparison

To compare Lyft and taxi fare, two factors are considered important in the experiment. First, different time periods in a day is important and thus time was divided into morning, noon, afternoon and midnight. Second, trip distances are divide into short distance(neighboring location zone), mediate distance(2-5 location zone) and long distance(>5 location zone). Hundreds of rows of hypothetical trips were used to get the predicted price from Lyft application and our taxi fare prediction model.
The result of time effects on Lyft and taxi is shown in Fig_9:
As shown in the figure, the overall trends of price changes according to different time period of a day are similar between Lyft and taxi. However, Lyft seems to be slightly cheaper than taxi in New York during rush hours in the morning before 11 am and in the afternoon after 3 pm. Lyft might be a more cost-efficient approach for transportation during rush hours.
The effects of distance on Lyft and taxi is shown in the Fig_10:

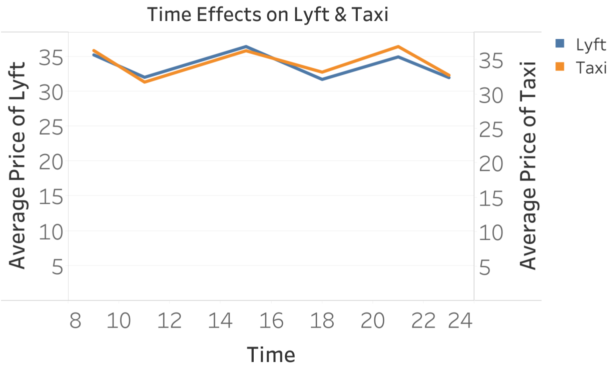


Figure 9: Time effects on Lyft and Taxi

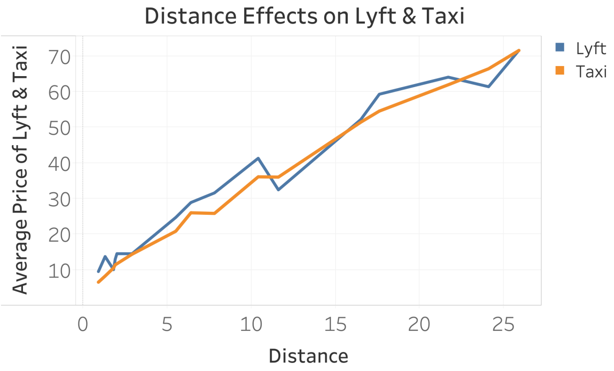


Figure 10: Distance effects on Lyft and Taxi

The data indicates that when distance is less than 10 miles, taxi is less expensive than Lyft. When the distance is longer, Lyft might be a better choice to save money. However, taxi is cheaper than Lyft between 17 and 22 miles, then they become almost uniform. With the increasing of distance, taxi fare seems to continually and smoothly increase; while Lyft fare shows some fluctuation.

Innovation:

1. Developed predication model of taxi fare prediction.
2. Compared fare cost between peek versus non_peek hours and weekdays versus weekends.
3. Converted Pick up and Drop off longitude and latitude into Taxi Zone using NYC Taxi Zone GeoJson Data.
4. Built interactive visualization methods.

5 CONCLUSIONS AND DISCUSSION

7.85 gigabytes data was used. After data cleaning and processing, a taxi fare prediction model was built to predict taxi fare. Thousands of hypothetical trips were used to compare price of taxi and Lyft. Data analysis and visualization were performed to show comparisons between Lyft and taxi.

Taxi is more expensive in afternoon rush hour than in morning rush hour. And in the same trip distance, the taxi price in weekend is less expensive than that in weekday.

Taxi is a better choice for short distance trips, while Lyft might be cheaper for medium distance trips. When a long ride is needed like from Bronx to JFK airport, there is no significant price difference between Taxi and Lyft.

6 DISTRIBUTION OF TEAM MEMBER EFFORT

All team members have contributed similar amount of effort.

Xueyang Zhang: Data cleaning and processing & experiment & poster design
Hao Wang: Data and time processing & map building & experiment
Yahui Ke: Prediction model building & experiment
Sikai Zhao: Model algorithm testing & visualization & website construction
Yaoxu Xiao: Model algorithm testing & visualization & analysis

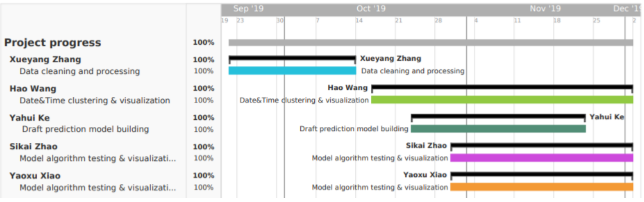


Figure 11: Project Timeline

REFERENCES

[1] Zaharia, Matei, et al. "Apache spark: a unified engine for big data processing." Communications of the ACM 59.11 (2016): 56-65.

[2] Armbrust, Michael, et al. "Scaling spark in the real world: performance and usability." Proceedings of the VLDB Endowment 8.12 (2015): 1840-1843.

[3] Gopalani, Satish, and Rohan Arora. "Comparing apache spark and map reduce with performance analysis using k-means." International journal of computer applications 113.1 (2015).

[4] McKinney, Wes. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* " O'Reilly Media, Inc.", 2012: chapter 10 Time Series

[5] Haines, Eric. "Point in polygon strategies." *Graphics gems IV* 994 (1994): 24-34.

[6] Hu, Shunfu, and Ting Dai. "Online Map Application Development Using Google Maps API, SQL Database, and ASP .NET." *International Journal of Information and Communication Technology Research* 3.3 (2013)

[7] Upadhyay, Rishabh Lui, Simon. (2017). Taxi Fare Rate Classification Using Deep Networks.

[8] Rangapuram, Syama Sundar and Seeger, Matthias W and Gasthaus, Jan and Stella, Lorenzo and Wang,

[9] Ahmed Tealab,Time series forecasting using artificial neural networks methodologies: A systematic review, Future Computing and Informatics Journal, Volume 3, Issue 2, 2018, Pages 334-340,ISSN 2314-7288.

[10] Schaffer, Cullen. "Selecting a classification method by cross-validation." Machine Learning 13.1 (1993): 135-143.

[11] Vafeiadis, Thanasis, et al. "A comparison of machine learn- ing techniques for customer churn prediction." Simulation Modelling Practice and Theory 55 (2015): 1-9.

[12] Vora, Deepali, and Kamatchi Iyer. "Evaluating the Effective- ness of Machine Learning Algorithms in Predictive Mod- elling." International Journal of Engineering Technology 7.3.4 (2018): 197-199.

[13] Xu, Dongkuan, and Yingjie Tian. "A comprehensive survey of clustering algorithms." Annals of Data Science 2.2 (2015): 165-193.

[14] Helga Thorvaldsdóttir, et al. "Integrative Genomics Viewer (IGV):high-performance genomics data." Briefings in Bioinformatics, Volume 14, NO 2.(2013):178-192.

[15] Raúl Mur-Artal, Juan D.Tardós. "Visual-Inertial Monocular SLAM With Map Reuse." IEEE Robotics and Automation Letters, Volume 2, Issue 2 (2017):796-803.

[16] Yasutaka Furukawa, Jean Ponce. "Accurate, Dense, and Robust Multiview Stereopsis." IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 32, Issue 8 (2010)

[17] <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>, accessed 10 Nov 2019

[18] <https://www1.nyc.gov/site/tlc/passengers/taxi-fare.page>, accessed 10 Nov 2019

[19] <https://github.com/microsoft/LightGBM>, accessed 10 Nov 2019