CIS 5200: Machine Learning, Fall 2022
Final Project

Rut Vyas, Irene Grace Karot Polson, Ruchika Singh

December 14, 2022

1 Abstract

Yellow Taxi Cabs are one of the significant modes of transportation in New York City. The number of daily taxi trips in the city that never sleeps amounts to almost half a million. This gives us plenty of information to understand what features (besides the quality of the driver) actually affect the tip received by a cab driver.

After examining quite an extensive data set, we have come up with a set of features that affect the tipping amount which in turn would help the drivers redefine their business model. Using this feature set, we try and predict the amount of the tip. Multiple models are trained and compared using the RMSE values to choose the best model for our task.

According to our findings the best model for our task is Decision tree regressor which gives a RMSE value of 0.74 and a list of features that were best for our task has been found in the following sections.

2 Motivation

The predictive task that is being analyzed is the tip amount paid for taxi trips in New York City (NYC). The motivation behind this predictive task was chosen to determine what factors cause people to tip higher than others. This analysis can be useful for taxi drivers in considering these factors in order to maximize the tip received as well as reconsider their business model that can generate higher tips.

The reason we have used machine learning to solve the problem is because we wish to identify the tipping patterns of passengers using the available data to see which features motivate them. This could lead to a better estimate of the suggested amount to be tipped making it more realistic and acceptable to both the driver and passengers.

3 Related work

Some of the works related to our project are as follows:

1. In this project [2] a variety of visualizations are performed to understand the distribution of values across the categorical variables and correlation of tip amount with other features. Initially, logistic regression with only numerical features was performed and the model gave an AUC of 0.51.

   Then, a non linear algorithm Random Forest was implemented and this performed better providing the team with an AUC of 0.64. Finally, time related features and categorical variables (through one-hot encoding) were added and the model was run again which brought the AUC up to 0.668.

2. The teams goal was to predict the tip amount. [3] Pre-processing was carried out to handle time data, categorical variables and eliminating outliers.

Subsequently, feature selection was carried out using the correlation matrix. A linear regression model was implemented using 4 features to predict tip amount. It yielded a RMSE of 1.84 and R-square value of 0.49.

3. This blog [4] performed an in depth analysis of the dataset to understand the dependance of features with one another and used a wide range of visualizations to analyze and comprehend the data.

4 DataSet

Our dataset is from the the New York City Taxi and Limousine Commission (TLC) where thye have provided TLC Trip Record Data that gives data from the 2009 to 2022. We have chosen 2021 data from this repository for our analysis as it has a large amount of data in itself.

4.1 Description of Dataset:

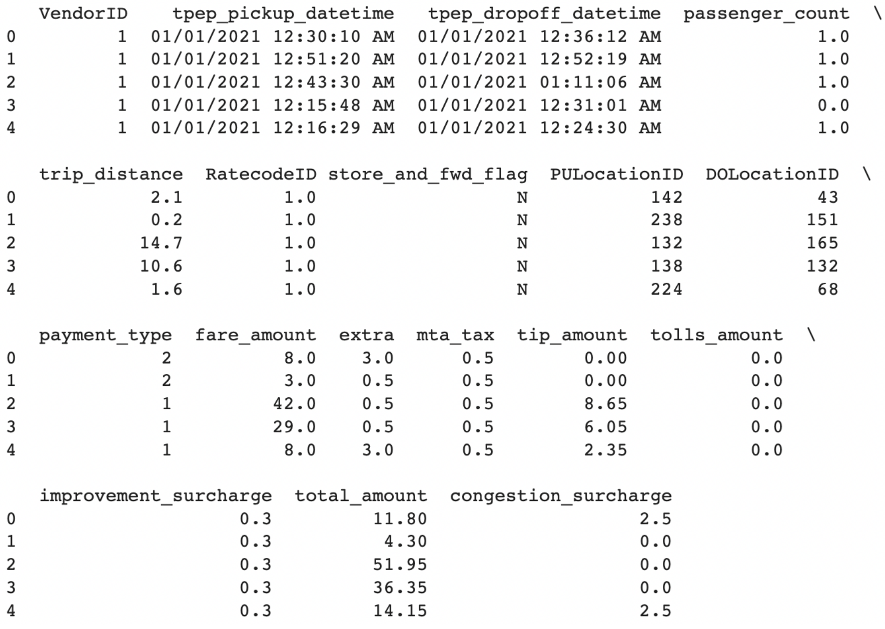
Our data set has 9214761 data points which correspond to n and 18 features which correspond to p. The data can be found at the NYC open data repository [1].

4.2 Data Pre-Processing:

In this section we walk you through the data pre-proccessing we carried out for the task.

4.2.1 Reducing Data Points

There are two vendors that provided the data for the yellow taxi lines. To have a daatset that is still large enough to retain the tipping patterns but not so large that our platform crashes, we have chosen the data from vendor 1 (Creative Mobile Technologies) in our dataset. We used vendor ID filter to filter out the data in order for the colab to handle the magnitude of data. In Fig 1 we have shown the dataset before feature engineering.



```
   VendorID    tpep_pickup_datetime   tpep_dropoff_datetime  passenger_count  \
0         1  01/01/2021 12:30:10 AM  01/01/2021 12:36:12 AM              1.0
1         1  01/01/2021 12:51:20 AM  01/01/2021 12:52:19 AM              1.0
2         1  01/01/2021 12:43:30 AM  01/01/2021 01:11:06 AM              1.0
3         1  01/01/2021 12:15:48 AM  01/01/2021 12:31:01 AM              0.0
4         1  01/01/2021 12:16:29 AM  01/01/2021 12:24:30 AM              1.0

   trip_distance  RatecodeID store_and_fwd_flag  PULocationID  DOLocationID  \
0            2.1         1.0                  N           142            43
1            0.2         1.0                  N           238           151
2           14.7         1.0                  N           132           165
3           10.6         1.0                  N           138           132
4            1.6         1.0                  N           224            68

   payment_type  fare_amount  extra  mta_tax  tip_amount  tolls_amount  \
0             2          8.0    3.0      0.5        0.00           0.0
1             2          3.0    0.5      0.5        0.00           0.0
2             1         42.0    0.5      0.5        8.65           0.0
3             1         29.0    0.5      0.5        6.05           0.0
4             1          8.0    3.0      0.5        2.35           0.0

   improvement_surcharge  total_amount  congestion_surcharge
0                    0.3         11.80                   2.5
1                    0.3          4.30                   0.0
2                    0.3         51.95                   0.0
3                    0.3         36.35                   0.0
4                    0.3         14.15                   2.5
```

Figure 1: Preview of the Dataset

4.2.2 Dropping Outliers

Through the box and whisker plot in Fig. 2, we realised that there were certain customers who tipped way too much. Further, we calculated the Z score of the tip amount column and removed all the datapoints that lay beyond the 3rd standard deviation point. This was done because we are more interested in predicting what most of the public tips and not certain NYC elites.

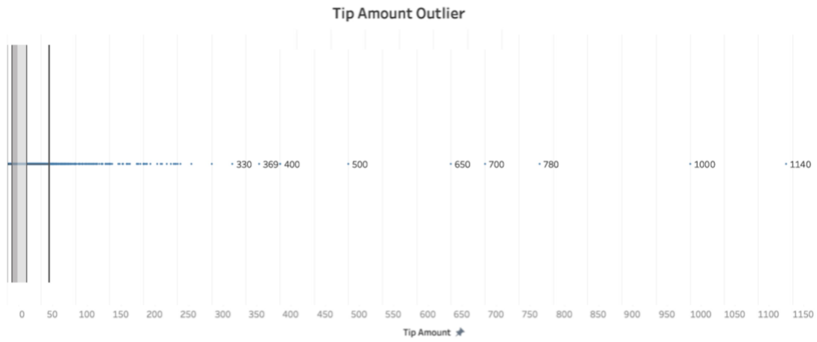
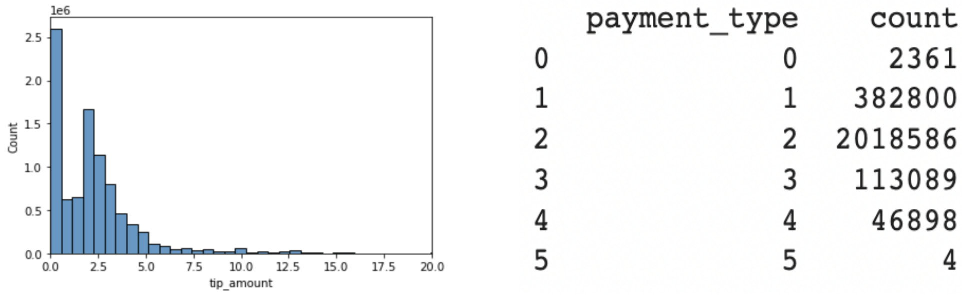


Figure 2: Outliers in our trip data

4.2.3 Zero Tipping and Cash Tipping

After analyzing the tip amount column we were able to identify that a majority of the population was tipping 0 dollars. In order to find the reason behind this, we went ahead and explored the data more. On analyzing the payment type and its relation to the tipping amount, we were able to observe the following:



```
   payment_type    count
0             0     2361
1             1   382800
2             2  2018586
3             3   113089
4             4    46898
5             5        4
```

(a) (b)

Figure 3: Tipping amount 0 grouped by by payment type

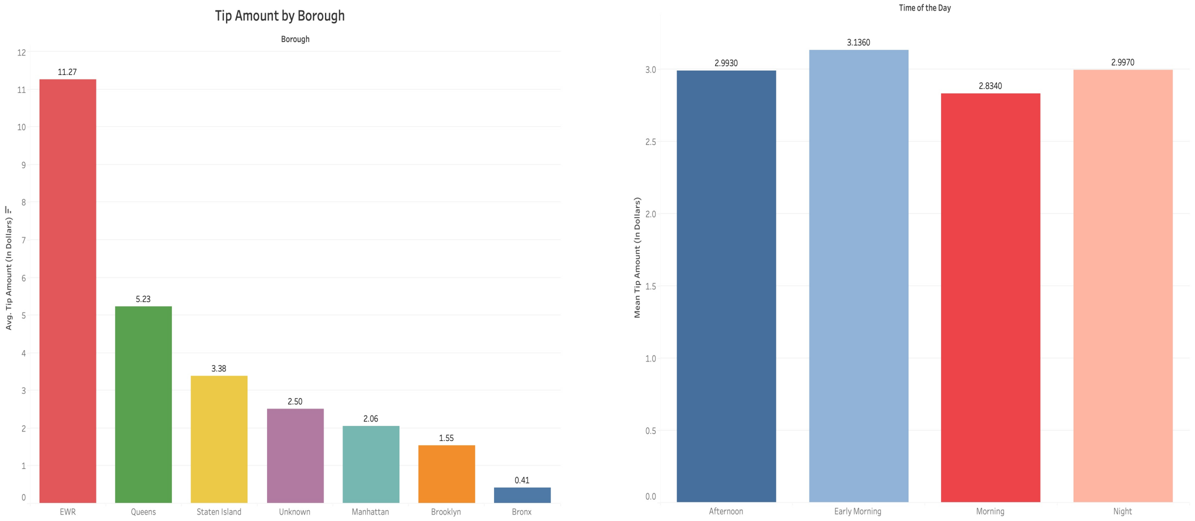
From this analysis, we were able to infer that 99% of the customers paying using the payment type: 2(cash), 3(no charge), 4(dispute), and 5(unknown) are tipping 0 dollars. One of the reasons behind this could be that cash tips are not recorded in the dataset. Other payment-type tips are also either not recorded or have faulty data collection. In order to reduce the noise, we will drop these columns.

4.2.4 Time Component

In order to use the time component in our models, we will go ahead and extract the pickup hour and pickup day from the tpep_pickup_datetime column after converting it to a suitable format according to our use. We then further grouped the hours of the day into four categories: early morning, morning, afternoon and night. This was done to reduce the number of categorical variables in that column in an attempt to improve the feature being fed to the models.

4.2.5 Geospatial Component

We divided the city of New York into 6 boroughs and grouped all the areas within those boroughs. A single numerical value was assigned to each borough. This was done for both pickup location and drop off location and these features were then added back into the dataset.



(a) (b)

Figure 4: Time and Spatial Component of Data

4.2.6 Feature Engineering

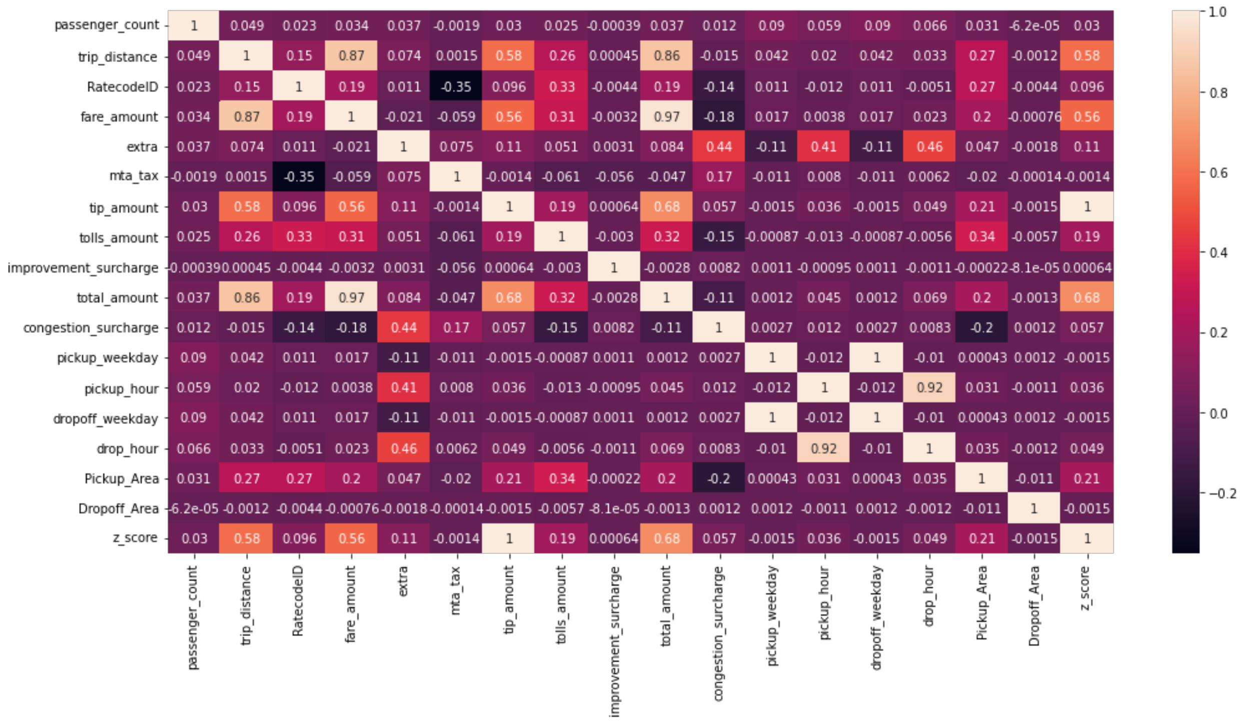
Based on the analysis above as well as the correlation matrix we have shown these features were identified as the features that would add the most value to our predictive models:

'trip_distance, 'RatecodeID', 'fare_amount', 'tolls_amount',' extra',' pickup_hour', 'pickup_weekday', 'Pickup_Area', 'Dropoff_Area'

5 Problem Formulation

Through comprehensive data analysis and pre-processing, we were able to find the features that affected the tip amount the most. This identification of features is especially helpful as they can be used by drivers and taxi companies for formulating business models that can help maximize tips. Moreover, this tip prediction model helps come up with more realistic tip suggestions that are beneficial to both the driver and the passenger. This way we have identified it as a machine learning task making use of the abundant data available for training.

We used L2 loss to calculate the Root Mean Square Error. The RMSE gives us a very quantifiable metric for evaluating how well the model is predicting tips. As we are doing regressions, RMSE gives us an error estimate in dollars for how much our tip prediction could be off from the actual tip given for a ride. We have used the same features that were engineered as illustrated in the previous section.



	passenger_count	trip_distance	RatecodeID	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge	total_amount	congestion_surcharge	pickup_weekday	pickup_hour	dropoff_weekday	drop_hour	Pickup_Area	Dropoff_Area	z_score
passenger_count	1	0.049	0.023	0.034	0.037	-0.0019	0.03	0.025	-0.00039	0.037	0.012	0.09	0.059	0.09	0.066	0.031	-6.2e-05	0.03
trip_distance	0.049	1	0.15	0.87	0.074	0.0015	0.58	0.26	0.00045	0.86	-0.015	0.042	0.02	0.042	0.033	0.27	-0.0012	0.58
RatecodeID	0.023	0.15	1	0.19	0.011	-0.35	0.096	0.33	-0.0044	0.19	-0.14	0.011	-0.012	0.011	-0.0051	0.27	-0.0044	0.096
fare_amount	0.034	0.87	0.19	1	-0.021	-0.059	0.56	0.31	-0.0032	0.97	-0.18	0.017	0.0038	0.017	0.023	0.2	-0.00076	0.56
extra	0.037	0.074	0.011	-0.021	1	0.075	0.11	0.051	0.0031	0.084	0.44	-0.11	0.41	-0.11	0.46	0.047	-0.0018	0.11
mta_tax	-0.0019	0.0015	-0.35	-0.059	0.075	1	-0.0014	-0.061	-0.056	-0.047	0.17	-0.011	0.008	-0.011	0.0062	-0.02	-0.00014	-0.0014
tip_amount	0.03	0.58	0.096	0.56	0.11	-0.0014	1	0.19	0.00064	0.68	0.057	-0.0015	0.036	-0.0015	0.049	0.21	-0.0015	1
tolls_amount	0.025	0.26	0.33	0.31	0.051	-0.061	0.19	1	-0.003	0.32	-0.15	-0.00087	-0.013	-0.00087	-0.0056	0.34	-0.0057	0.19
improvement_surcharge	-0.00039	0.00045	-0.0044	-0.0032	0.0031	-0.056	0.00064	-0.003	1	-0.0028	0.0082	0.0011	-0.00095	0.0011	-0.0011	-0.00022	-8.1e-05	0.00064
total_amount	0.037	0.86	0.19	0.97	0.084	-0.047	0.68	0.32	-0.0028	1	-0.11	0.0012	0.045	0.0012	0.069	0.2	-0.0013	0.68
congestion_surcharge	0.012	-0.015	-0.14	-0.18	0.44	0.17	0.057	-0.15	0.0082	-0.11	1	0.0027	0.012	0.0027	0.0083	-0.2	0.0012	0.057
pickup_weekday	0.09	0.042	0.011	0.017	-0.11	-0.011	-0.0015	-0.00087	0.0011	0.0012	0.0027	1	-0.012	1	-0.01	0.00043	0.0012	-0.0015
pickup_hour	0.059	0.02	-0.012	0.0038	0.41	0.008	0.036	-0.013	-0.00095	0.045	0.012	-0.012	1	-0.012	0.92	0.031	-0.0011	0.036
dropoff_weekday	0.09	0.042	0.011	0.017	-0.11	-0.011	-0.0015	-0.00087	0.0011	0.0012	0.0027	1	-0.012	1	-0.01	0.00043	0.0012	-0.0015
drop_hour	0.066	0.033	-0.0051	0.023	0.46	0.0062	0.049	-0.0056	-0.0011	0.069	0.0083	-0.01	0.92	-0.01	1	0.035	-0.0012	0.049
Pickup_Area	0.031	0.27	0.27	0.2	0.047	-0.02	0.21	0.34	-0.00022	0.2	-0.2	0.00043	0.031	0.00043	0.035	1	-0.011	0.21
Dropoff_Area	-6.2e-05	-0.0012	-0.0044	-0.00076	-0.0018	-0.00014	-0.0015	-0.0057	-8.1e-05	-0.0013	0.0012	0.0012	-0.0011	0.0012	-0.0012	-0.011	1	-0.0015
z_score	0.03	0.58	0.096	0.56	0.11	-0.0014	1	0.19	0.00064	0.68	0.057	-0.0015	0.036	-0.0015	0.049	0.21	-0.0015	1

Figure 5: Correlation Matric for our dataset

6 Methods

In this section, we look at the different models that were considered and trained to fit the data above. In the end we compare the different models and their performance.

6.1 Baseline Model

We use multivariate linear regression as our model. We wanted to see if the dependent variable was linearly separable based on the 9 features that we had shortlisted in feature engineering and check to see the extent of correlation between the dependent and independent variables. Based on the RMSE value of 1.28 and R-squared value of 0.51, we can conclude that only 51% variance of the dependent variable is being explained by the independent variables.

6.2 Decision Tree Regressor

The decision tree regressor splits the training data into smaller and smaller subsets such that the sum of squared residuals is minimized. Finally, the output is predicted by taking the average of all values that fall in that category. This model was implemented as it is not affected by non-linearity and we wanted to explore if a single decision tree can explore the complex relationship between our tipping amount and its features. To avoid overfitting, a hyperparameter search was performed over the maximum depth and the optimal depth of 20 was chosen.

6.3 Random Forest Regressor

The random forest regressor is an ensemble learning algorithm that creates several random decision trees from random data points in the dataset, combines predictions from all the trees and averages their output to give a stronger prediction. It is much better at handling the problem of overfitting that our previous model (decision tree) suffers from. In order to optimize our model and keeping in mind that this model is computationally very slow, a hyperparameter search was performed using the RandomizedSearchCV package of sklearn. The parameters that were finalized were as below:

- 'n_estimators': 150
- 'min_samples_split': 10
- 'min_samples_leaf': 4,
- 'max_features': 'sqrt',
- 'max_depth': 15

6.4 K Neighbours Regressor

We wanted to explore if this problem could be solved through clustering in a higher dimensional space. We implemented K neighbors regressor that calculates the average of the numerical target value of the K nearest neighbor. This model gives higher weightage to features that have higher magnitude as it is a distance based model and hence we scale the data before feeding it to the model. The biggest disadvantage with this model is that the time and cost required to calculate the distance between a new point and each existing point is huge.

6.5 XGBoost

In extreme gradient tree boosting, trees are grown/added sequentially to the ensemble and more weight is given to weak learners to convert them into stronger learners. Each new decision tree is fit on the residual of the previou learner and makes use of arbitrary differentiable loss function and gradient descent optimization algorithm. It carries out parallel tree boosting and is thus much faster than other ensemble tree models like Random forests. Hence, this model was implemented to explore the tradeoff between training time and accuracy. A hyperparameter search was performed using the RandomizedSearchCV package of sklearn and the ones that were finalized are:

- 'subsample': 0.75
- 'n_estimators': 200
- 'max_depth': 8
- 'learning_rate': 0.3

6.6 Neural Nets

Our data has a complex non-linear relationship and in order to explore this avenue, we decided to implement the neural network using the Keras deep learning library. It has an activation function in each layer that helps it understand this relationship. The mean squared error was used as the loss function because this the same function we are using to evaluate the model and taking square root of it will give us the accuracy in terms of dollars. The adam optimizer was used because it converges faster, can handle sparser gradients and is known to perform generally well on noisy problems. The depth (number of layers) and width (number of neurons in each layer) of the network were chosen through experimentation and intuition.

7 Experiments and Results

7.1 Evaluating Success:

We will use an evaluation metric to compare how our different predictive models perform on the given dataset. This evaluation metric helps us identify whether a model can actually be considered an optimization. We are specifically using mean squared error while evaluating the model. We have selected linear regression as our base model and we have incorporated models to compare and try to beat our baseline model.

We have made use of root mean squared error for two purposes:

- RMSE squares the errors before averaging and assigns higher weights to larger errors. This is particularly useful in our scenario as we are working with money and higher deviation from the actual dollar amount should be punished more.
- Our predicted amount is in dollars.Thus, by using RMSE, we are getting our error in dollars too. This is really helpful in terms of understanding and analyzing the performance of each model.

Model	RMSE	Training Time
Linear Regression	1.28	1.1 sec
Decision Tree Regressor	0.74	16.7 sec
Random Forest Regressor	0.83	15 mins
K Neighbours Regressor	0.94	2hr 10mins
XGBoost	0.94	37mins
Neural Nets	0.88	257 secs

From our analysis, we have come to the conclusion that Decision Tree Regressor works the best. One of the possible reasons for this could be that it most closely mimicks the though process a passenger goes through when contemplating how much tip. Another reason we choose Decision Tree Regressor as the best model is from the trade of between training error and test error. Moreover it has a more realistic training time for the model allowing scale it more with larger datasets in the future.

8 Conclusion and Discussion

This project was quite a journey and we learned a number of things. We can divide this into three categories:

Handling Large Amount of Data: The data had 30.9 million rows and our platform kept crashing while performing even the most basic operations. To overcome this, we initially set up the pyspark environment on Collab and carried out stratified sampling. However, the collab still kept crashing when we tried to convert the reduced size pyspark dataframe to a pandas dataframe. Thus, we later went ahead with a separate strategy of taking the data only from a single vendor to maintain uniformity, reducing the size to roughly 10 million rows.

Geospatial Data: We had a feature in the dataset that specified from and to what location does a passenger travel. We constructed a strategy wherein we divided the entire city of New York into 6 boroughs and assigned a variable to each of the boroughs. This feature was then added back into the models.

Time Data: Our dataset had a feature that specified the time of boarding and getting off from the taxi. We extracted the hour of the day and further grouped it into four different periods (morning, afternoon, etc) and also extracted the week day (Monday, Tuesday, etc). Both of these features were assigned variables and added back into the model to see if time of the day or the weekday attribute had any effect on our predictions.

We realized that there are certain features absent from the dataset that could further help in improving our predictions. Possible extension to this project could be:

- Scrapping the weather data and merging that to our current dataset with time as the key. Then, explore if this feature can improve the accuracy of our predictions.
- Carrying out an extensive research on the per capita income of all the areas of New York and grouping them based on this income rather than through the 6 boroughs.
- Creating an ensemble machine learning model that stacks different modeling algorithms to give the final prediction.

When we think of how a tip is evaluated, we generally assess the person for the quality of their service. This for the most part is intuitively true, but when we look deeper we find that it is not the only factor in play. In the case of New York Citys taxi drivers, location and time both play a role in determining how much of a tip they receive. It can be said that during peak hours, higher percentage of tips will be paid to any taxi driver. Hopefully, with this information, taxi drivers can better plan their routines for the day in order to increase their own earnings.

References

[1] TLC Trip Record Data. `https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page`, 2022. [Online; accessed 19-July-2008].

[2] Joseph W. Richards Henrik Brink and Mark Fetherolf. *Real-World Machine Learning*. Manning, 2016.

[3] Pamphile Roy. Machine learning series: Nyc yellow taxi tips prediction, 2022.

[4] Todd W. Schneider. Analyzing 1.1 billion nyc taxi and uber trips, with a vengeance, 2015.