NYC Taxi Data Analysis Final Project Report

Mathew Pelletier
Schulich School of Engineering
University of Calgary
10103415

Alexandra Glodzinski
Schulich School of Engineering
University of Calgary
30038541

Marco Arias
Deparmtment of Science
University of Calgary
30079108

Maram Elsayed
Schulich School of Engineering
University of Calgary
30071200

***Abstract* — In New York City (NYC), Taxis are a common mode of transportation. With each taxi trip taken through NYC, a cluster of data is collected with the trip's information. This paper explores the aforesaid cluster of data and utilizes it in predicting trip duration, as well as trip fare, by using a Machine Learning (ML) approach, i.e., Linear Regression. By performing Exploratory Data Analysis, data pre-processing, data scaling, and building a ML model, our solution was able to achieve a test accuracy score of 0.57 when predicting Trip Duration, and a score of 0.87 when predicting Trip Fare.**

I. Introduction

The following project explores the FOILing NYC's Taxi Trip Data [1]. This data was a combination of two datasets; Trip data and Fare Data. The problem we are addressing is regarding the Trip Fare and Trip Times of Taxi rides throughout the City. Using the aforementioned data, we will be predicting the Taxi Trip Time (in seconds) and the Taxi Trip Fare (in USD) by using Trip Distance, Day of Week, Hour of Day, Pickup Location and Drop-off Location as our features.

The data portrayed in this dataset shows descriptive and important information regarding taxi trips in New York City. With the observations made from the dataset, we would be able to forecast adequate times to take a taxi and when it might be better to find another form of transportation. Additionally, it would be beneficial to clients by providing them with an estimate of how much their trip fare would cost depending on desired factors.

After research, we have discovered several other projects tackling similar problems, though most use different datasets than the one we chose. Of the projects examined, only a few were using Spark, and none were using PySpark specifically. A few projects were simply visualizing the data, others however were trying to accomplish something similar to our goal generating a model to predict the fare and trip duration.

For this project, as mentioned, we are using Apache Spark as our framework so we can resolve the problem as a Big Data problem. By using Apache Spark, we are attempting to create a more scalable solution since the dataset used in this problem is vast and will continue to grow further.

While analyzing the data, the main questions that we were eager to answer were: "Which day is the busiest day of the week?", "Which hour is the most common to take a cab?" and "How can distance affect our observation?". Alongside these questions, we wanted to display the statistical properties of each attribute to understand our data better, process it, tune it, and make our own assumptions.

This project proposes a strategy for NYC residents to plan their journey out in a more efficient and timely manner by anticipating what to expect when choosing taxis as their primary mode of transportation throughout specific city blocks. From our experiments we were able to predict the trip fare relatively successfully with an R-squared of 0.86, however the trip duration predictions were lacking with an R-squared of 0.57.

II. Background and Related Work

For the topic of our project (predicting Taxi Trip Times and Fares), no technical background is necessary to understand frequently used terms. However, a basic understanding of Machine Learning techniques and algorithms will be needed to understand the model used and the process of using it.

Both the trip fare and duration prediction for NYC taxis were past Kaggle competitions hosted by Google Cloud and Kaggle respectively. As such there are several sources of existing work related to this problem [2, 3, 4, 5], albeit with different datasets. Most of these projects had a process that resembled our own, with similar choices in data exploration and pre-processing as well as the machine learning tools used to generate their model. However, none of the projects on Kaggle used Spark as a framework, so we decided to do some more research and found a few projects using Spark however they were written in Scala and [6, 7] R rather than using PySpark as we intended to.

III. Methodology

A. Experiment Setup

After exploring different platforms, out of Docker, Databricks, and Deepnote, as well as choosing to implement our solution in either Hadoop MapReduce or Apache Spark, we decided to settle with Deepnote using PySpark as it is an easy tool to use and allows members to collaborate smoothly. We also decided to select Python as our programming language, as it is relatively simple to work with and it has numerous Machine Learning libraries to utilize in our implementation. In case there would be any trouble arising from this platform or this programming

language, we can choose to convert to a different platform or language with little complexities.

After analyzing the data files for NYC taxi data, we have combined both the trip and fare data. We only used one of the 12 files provided for each trip and fare data since each file was very large in size and would greatly increase our execution time if all files were used. Data entries from the chosen file have been reduced to approximately 10% of the available dataset to avoid long wait times. After analyzing the combined data, we noticed a lot of irrelevant features to our solution, such as "Vendor ID" and "Hack License". Therefore, we only kept necessary information, such as Trip Distance, Pickup Location, Drop-off Location, etc.

One main event during pre-processing was filtering out anomalies, these included Null values and data points that did not fit into certain parameters (such as Latitude and Longitude Coordinates that were out of the range of NYC). Once all the irrelevant data had been removed, we then decided to simplify the pickup and drop off coordinates into a grid of "blocks" spanning NYC. This allowed us to examine trips from one "block" to another rather than having to deal with the specific coordinates. One thing to note here is that the actual range of blocks that is used is substantially smaller than we originally anticipated only ranging from 0-7593 rather than 0-10000 as we originally planned. This discrepancy is explained by the irregular shape of NYC vs the rectangular bounding box coordinates we used to assign the blocks. As such many of these blocks are likely underwater and as such will not be represented in the data.

B. Experimentation Factors

The machine learning algorithm used is Linear Regression which is the transpose of w on x where w is weights and x is features.

The training and test data was split into a 75%, 25% split respectively. This was decided after taking into account the recommendation given of a 60%, 20%, 20% split for training testing and validation. Taking this into account we decided to leave 25% for testing and the remainder for training.

Our hyper parameters were tuned using the Standard Processor from the sklearn library (sklearn.preprocessing.StandardScaler). This executes the feature scaling we want by using the formula result = (value - mean) / standard deviation.

C. Experiment Process

The following sequence of actions were followed to complete our Experimentation Process:

1. Pick a valuable dataset to carry experimentation on..
2. Perform Exploratory Data Analysis to understand the content of the dataset better.
    a) Print the schema of our PySpark DataFrame.
    b) Show the first five rows of our DataFrame.
    c) Print the shape of the DataFrame.
    d) Print the statistical properties of our features.
    e) Visualizing Frequency of 'day_of_week' and 'hour' to look at trends.
3. Pre-process the data to remove unnecessary data points and features discovered when performing EDA. A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
    a) Removing unnecessary spaces from headers to join Trip Data and Fare Data.
    b) Join Trip Data and Fare Data.
    c) Filter the dataset to approximately 10 percent of data.
    d) Remove rows that don't have type 1 for 'rate_code'.
    e) Filter out unwanted features.
    f) Changing all data types to designated types.
    g) Removing Invalid Location Coordinates.
    h) Turning 'date' from string to date format.
    i) Creating a 'day_of_week' column where certain days of week correspond to numbers.
        - Sunday: 1
        - Monday: 2
        - Tuesday: 3
        - Wednesday: 4
        - Thursday: 5
        - Friday: 6
        - Sunday: 7
4. Scaled our features.
5. Built models.
6. Reported Scores.

D. Performance Metrics

In the below image we are given the trip time scores for accuracy in the training data, test data, as well as error values. The accuracy for both train and test data is around 57-58%

```
Model Train Accuracy (score): 0.5765169328187789
Model Test Accuracy (score): 0.5701033823538049
Mean Absolute Error: 193.4752193358585
Mean Squared Error: 83525.65859334059
Root Mean Squared Error: 289.00805973768377
R2: 0.5701033823538049
```

Figure 1: Trip Time Accuracy and Errors

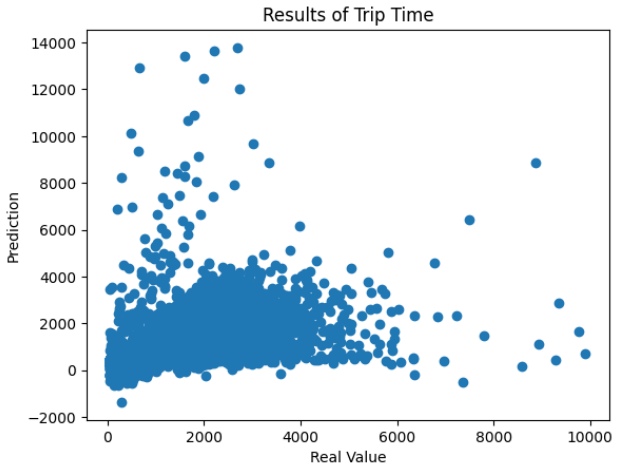

Figure 2: Trip Time Visualization

In the below image we are given the trip fare scores for accuracy in the training data, test data, as well as error values. The accuracy for both train and test data is around 86-87%

```
Model Train Accuracy (score): 0.871361787058128
Model Test Accuracy (score): 0.8684295807308309
Mean Absolute Error: 1.2825764309115004
Mean Squared Error: 6.818901423691923
Root Mean Squared Error: 2.611302629664345
R2: 0.8684295807308309
```

Figure 33: Trip Fare Accuracy and Errors

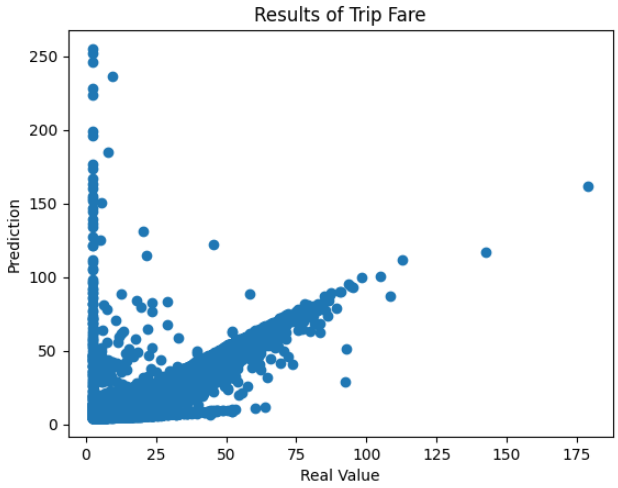

Figure 4: Trip Duration Visualization

We notice that the trip time data is much less accurate than trip fare. There can be multiple reasons for this. Trip time may not be very accurate in general due to factors like special events happening, time of day like rush hour, and weather which is especially unpredictable in winter. Our data is from the month of January which is winter in NYC which can explain some of the lack in accuracy. Trip fare can be argued to be more accurate since it is dependent on both time and distance which is usually how taxi fare meters work.

We weren't able to report Precision, Recall and the F-score of our model due to it being a regression model and not a classifier. This is because regression models give us a continuous output (not a classification). Therefore, we cannot use the metrics mentioned since they use predictions that were either classified correctly or incorrectly.

IV. Results

A. Key Findings

Our exploratory data analysis yielded a few interesting points, namely that the records had several wild inconsistencies. Several trips had a zero distance, whether this is from poor record keeping, a bug, or user error is unknown. Further, the trip coordinates had several values that were well outside the possible range indicating a possible error with the mechanism for GPS tracking. We also discovered some statistical properties from the data including that Thursdays had the highest frequency of trips compared to any other day as seen below in figure 3, and that 6:00 PM is the most frequent time of day for a trip as seen in figure 4. From our machine learning models we generated a few metrics to use in analysis, which can be found in section 4.0 above. Perhaps the most relevant metric here is R-Squared value which is approximately 0.57 trip duration prediction and 0.86 for the trip fare prediction.

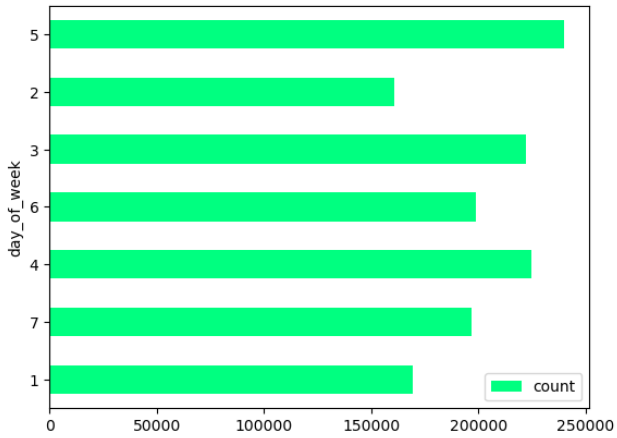

Figure 5: Counts Per Day of Week

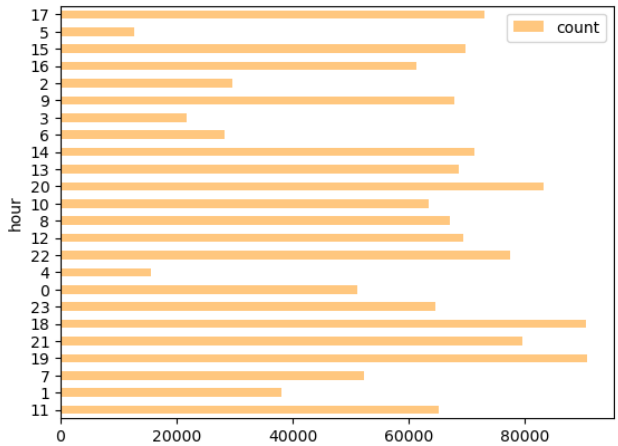

Figure 6: Counts Per Hour of Travel

B. *Conclusion*

The objectives of this project were to predict travel time and fares for taxis in NYC. After obtaining data, completing exploratory data analysis, and applying machine learning, we have successfully been able to predict times and fares. Along the way we were also able to determine the data accuracy as well as determine the model built through linear regression. Overall, we determine this project to be a success, however there is more that can be done with this project.

C. *Future Work*

To build onto our project in the future, we would implement a more scalable solution that allows us to use significantly more than 10% of our data, this can be done by utilizing parallelism to split the execution time by the number of partitions we define. Another improvement that should be implemented in future work is the application of more complex models (like neural network models or any type of classifiers), this might guarantee us a better prediction of the Taxi Trip Time and hopefully provide us with an accuracy of above 0.5. Lastly, we would visualize our data more during EDA by using heatmaps and other visualization methods, this will improve the layout of our project and help readers understand the data better.

V. Team Member Contributions

For the project, everyone contributed almost equally, therefore, we are saying that each individual: Mathew, Marco, Maram, and Alexandra, contributed approximately 25% towards the project. Mathew set up the environment. As for the rest of the project, we all would mob on it during the class's lab times (Wednesdays from 3pm - 5pm). This includes EDA and machine learning.

By signing this declaration each team member agrees that everyone contributed approximately 25% towards the group project. By not signing an individual does not agree with this estimate.

I, Alexandra Glodzinski, hereby sign this declaration on December 15th, 2022.

I, Mathew Pelletier, hereby sign this declaration on December 15th, 2022.

I, Marco Arias, hereby sign this declaration on December 15th, 2022.

I, Maram Elsayed, hereby sign this declaration on December 15th, 2022.

References

[1] C. Whong, "Foiling NYC's Taxi Trip Data," *Chris Whong*, 18-Mar-2014. [Online]. Available: https://chriswhong.com/open-data/foil_nyc_taxi/.

[2] V. Atal, "NYC taxi trip duration," *Kaggle*, 09-Dec-2022. [Online]. Available: https://www.kaggle.com/code/vivekatal/nyc-taxi-trip-duration. [Accessed: 15-Dec-2022].

[3] A. Van Breemen, "NYC taxi fare - data exploration," *Kaggle*, 27-Aug-2018. [Online]. Available: https://www.kaggle.com/code/breemen/nyc-taxi-fare-data-exploration/notebook. [Accessed: 15-Dec-2022].

[4] Sara, "NYC taxi duration - regression LIGHTGBM ML model," *Kaggle*, 13-Dec-2022. [Online]. Available: https://www.kaggle.com/code/salvarezgonz/nyc-taxi-duration-regression-lightgbm-ml-model. [Accessed: 15-Dec-2022].

[5] R. Christy, "New York City taxi fare prediction," *Kaggle*, 01-Dec-2022. [Online]. Available: https://www.kaggle.com/code/rinichristy/new-york-city-taxi-fare-prediction. [Accessed: 15-Dec-2022].

[6] "Analyzing a billion NYC taxi trips in Spark," *rstudioconnect*. [Online]. Available: https://beta.rstudioconnect.com/content/1705/taxiDemo.nb.html. [Accessed: 15-Dec-2022].

[7] Yinshangyi, "Yinshangyi/NYC-Taxi-rides-Spark: Predicting the fare of taxi rides (with Spark-Scala)," *GitHub*. [Online]. Available: https://github.com/Yinshangyi/nyc-taxi-rides-spark. [Accessed: 15-Dec-2022].