

Analysis of New York City Taxi Trips

Big Data Processing with Databricks Spark



Name: Amy Yang

Date: 02.10.2022

Contents

Project Overview	3
Data Understanding	3
Data Ingestion and Preparation	3
Business Questions	6
Machine Learning Models	9
Reflection on the Project	11
Conclusion	12
Appendix	13
Appendix A - Data Dictionary	13
Appendix B – Separate folders in Azure Storage for Yellow Taxi Dataset	15
Appendix C - Data Quality Check and Imputation	16
Appendix D - Correction Analysis	17

Project Overview

The New York City Taxi and Limousine Commission (TLC) is the agency responsible for licensing and regulating New York City's taxi cabs. It has been collecting trip records for both yellow and green taxi cabs since 2009. This project focuses on the trips in recent 4 years and performs data processing, transformation and analysis with Databricks Spark to provide business findings and insights. A regression model is also built to predict the total fare amount of taxi trips based on several features including trip duration, taxi speed, tips and so on. Detailed discussion is in the sections below followed by the conclusion. All the scripts are saved in the file 'NYC_Taxi.ipynb'.

- Data Understanding
- Data Ingestion and Preparation
- Business Questions
- Machine Learning Models
- Reflection on the project

Data Understanding

The dataset used in this project includes the trip records for both yellow and green taxi cabs during the period from January 2019 to April 2022, which was downloaded from the website of [TLC](#). The data for each month is saved in a separate parquet file for both yellow and green taxi trips. The records include fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts. Based on the initial data exploration, the following differences are found in the fields of yellow and green taxi trip records:

- The field of 'airport_fee' only exists in the yellow taxi dataset
- The fields of 'ehail_fee' and 'trip_type' are only captured by the green taxi dataset
- The field names of pick-up and drop-off dates/times in yellow and green taxi datasets are different.

The data dictionary for both yellow and green taxi trips is included in [Appendix A](#).

Data Ingestion and Preparation

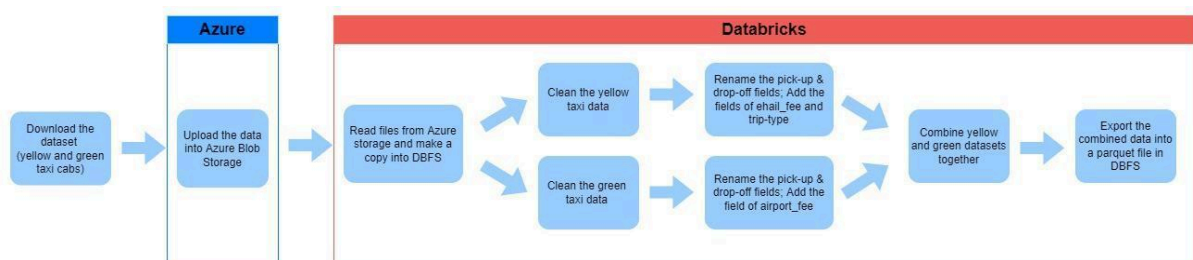


Figure 1. Flow chart for Data Ingestion and Preparation

With a better understanding of data, the next action is to ingest the data on Databricks and perform data cleaning before analysis is carried out. Figure 1 above summaries the steps of data ingestion and

preparation, starting from the downloaded dataset to the final parquet file used to create a table or view. Further explanation of each step is described below.

1. Download the dataset and upload the data into Azure Storage

Download the dataset for yellow and green taxi cabs from January 2019 to April 2022 on the [TLC](#) website. Then create a new container called 'bde-assignment-2' in the storage account on the Azure Portal and upload the downloaded parquet files into this container. As there is a data type issue with the 'airport_fee' field in the yellow dataset, the parquet files of yellow taxi data need to be uploaded into two folders – double and integer. The parquet files under the folders of double and integer are listed in [Appendix B](#). The data type issue is not found in the green taxi dataset. Thus no separate folders are needed.

2. Read the files from Azure storage and make a copy into DBFS

Mount the blog container into the Databricks Files System with the storage account name, storage account access key and blob container name as shown in Figure 2.

[illegible]

Figure 2. Mount the blob container into Databricks Files System

Then read the parquet files in folders of double and integer, append them together and save a parquet file for yellow taxi trips in DBFS. Likewise, read the parquet files of the green taxi dataset and save a parquet file in DBFS accordingly.

3. Count the total number of rows for each taxi colour

Read the parquet files stored on DBFS for yellow and green taxi datasets respectively and count the total number of rows in both datasets. As Figures 3 and 4 show, the yellow and green taxi dataset has total rows of 152,823,008 and 9,390,483 respectively.

```
1 # Count the total numbers of rows for yellow taxi
2 yellow.count()
```

- ▶ (2) Spark Jobs

Out[5]: 152823008

Figure 3. Total number of rows in yellow taxi data

```
1 # Count the total numbers of rows for green taxi
2 green.count()
```

- ▶ (2) Spark Jobs

Out[8]: 9390483

Figure 4. Total number of rows in green taxi data

4. Convert the “Yellow” April 2022 parquet file into a CSV file and send it to Azure Storage

Read the parquet file 'yellow_tripdata_2022-04.parquet' from the mnt folder and write a CSV file 'yellow_tripdata_2022-04.csv' under the mnt folder. After this, it can be seen from Azure Storage that there are two files generated accordingly. One CSV file named 'yellow_tripdata_2022-04.csv' has a size of 0B (Figure 5) and another CSV file named starting with 'part-00000' has a size of 393.83MiB (Figure 6). Thus the total size of generated CSV files is 393.83MiB. Compared to the

parquet file 'yellow_tripdata_2022-04.parquet' with a size of 52.66MiB, the CSV files are significantly larger and take much more storage in Azure. This explains why the data is preferred to be stored in parquet files rather than CSV files.

Figure 5. CSV file generated for yellow April 2022 data

The screenshot shows the Azure Storage Explorer interface. At the top, a file named 'yellow_tripdata_2022-04.csv' is selected, with details: 9/22/2022, 2:01:01 PM, Hot (Inferred), Block blob, 0 B, Available. Below this, the location is 'bde-assignment-2 / yellow_tripdata_2022-04.csv'. A search bar is present with the text 'Search blobs by prefix (case-sensitive)'. A table of blobs is displayed with columns: Name, Modified, Access tier, Archive status, Blob type, and Size. The table contains five rows: a folder '[.]', a file '_committed_2654202827336990849' (112 B), a file '_started_2654202827336990849' (0 B), a file '_SUCCESS' (0 B), and a file 'part-00000-tid-2654202827336990849-bd93c5d4-e306-4a60-a45b-2d56ba62b39f-96-1-c000.csv' (393.83 MiB).

Name	Modified	Access tier	Archive status	Blob type	Size
[.]					
_committed_2654202827336990849	9/22/2022, 2:00:57 PM	Hot (Inferred)		Block blob	112 B
_started_2654202827336990849	9/22/2022, 1:59:07 PM	Hot (Inferred)		Block blob	0 B
_SUCCESS	9/22/2022, 2:01:01 PM	Hot (Inferred)		Block blob	0 B
part-00000-tid-2654202827336990849-bd93c5d4-e306-4a60-a45b-2d56ba62b39f-96-1-c000.csv	9/22/2022, 2:00:55 PM	Hot (Inferred)		Block blob	393.83 MiB

Figure 6. CSV file generated for yellow April 2022 data

5. Clean the datasets of both yellow and green taxi trips

In this step, data cleaning is performed for both yellow and green taxi datasets by removing the unrealistic trips. Two new columns 'trip_duration' (seconds) and 'speed' (mile per hour) are created. Trip duration is defined as the gap between drop-off time and pick-up time and speed is calculated as trip distance divided by trip duration. The criteria in Table 1 are taken into consideration when unrealistic trips are removed.

Table 1 Criteria for removing the unrealistic trips

Criteria	Justification
Drop-off datetime is earlier than pick-up datetime	self-explanatory
Speed is less than 0	self-explanatory
Speed is over 65 miles per hour (mph)	Based on the research , the speed limit in New York City is 30mph and the speed limit outside New York City is 65mph.
PULocationID is not between 1-263 or DOLocationID is not between 1-263	Based on the TLC Trip Records User Guide , the location ID should range from 1-263. As suggested by Wiki , New York City (NYC) is made up of 5 boroughs – the Bronx, Brooklyn, Manhattan, Queens and Staten Island. According to the lookup table CSV file in the TLC Trip Records User Guide, the location IDs for these 5 boroughs range from 2-263 and location ID=1 represents Newark Airport. Thus, location IDs 2-263 are assumed to be within NYC and location ID 1 is assumed to be outside NYC.
Speed within New York City is over 30 mph	Based on the research , the speed limit in New York City is 30mph.
Trip duration is no less than 2 mins and no more than 3 hours	It is assumed that a trip shorter than 2 minutes can be replaced by a walk. Based on the research, it can take 1.5 hours to drive

	the longest distance within NYC. Considering the traffic during the peak hours, the trip duration is limited to 3 hours.
Trip distance is no shorter than 0.2 miles (320m) and no longer than 195 miles	It is assumed that a trip shorter than 320m can be replaced by a walk. As the trip duration is no more than 3 hours and the speed limit is 65mph, the trip distance should be limited to 195 miles.
Passenger count is over 6	Based on the ruling in 54-15(g) , a Driver must not permit more than 4 Passengers to ride in a four-passenger Vehicle, nor more than 5 Passengers in a five-passenger Vehicle, except that an additional Passenger must be accepted if the Passenger is under the age of seven (7) and is held on the lap of an adult Passenger seated in the rear. Thus the maximum passenger count is 6.
The total amount is negative when the payment type is 1 (credit card) or 2(cash)	self-explanatory
Pick-up time is earlier than 2019-01-01 and later than 2022-05-01	The date range of the dataset is from 2019-01-01 to 2022-04-30
The total amount is over \$300	It is assumed that passengers will not pay for a trip with total fare amount over \$300. Instead, they will choose other transport such as airplane.
Any duplicate rows	Remove the rows when the value of each field is the same

After the data is cleansed, for both yellow and green taxi datasets, the total number of rows is around 92% of the row number for their original datasets.

6. Combine the yellow and green taxi datasets

Before the yellow and green taxi datasets are appended, the steps below are taken:

- Rename the pick-up and drop-off fields for yellow taxi data to 'pickup_datetime' and 'dropoff_datetime'
- Rename the pick-up and drop-off fields for green taxi data to 'pickup_datetime' and 'dropoff_datetime'
- Add the column of 'color' in the yellow taxi dataset and label the column as 'yellow'
- Add the column of 'color' in the green taxi dataset and label the column as 'green'
- Add the columns of 'trip_type' and 'ehail_fee' for yellow taxi data with null values and double data type
- Add the columns of 'airport_fee' for green taxi data with null values and double data type

After the data processing, the yellow and green taxi datasets are combined using 'unionByName' function.

7. Export the combined data into a parquet file

Save the combined dataset and write it into a parquet file on DBFS. The parquet file then can be read and loaded as a view or table.

Business Questions

Once the data is ingested and cleaned, it is ready to be analysed. Six questions are looked at in depth as below.

- As shown in Figure 7, for each year and month, the total number of trips varied significantly over the last 4 years. Before March 2020, the total number of trips ranged from 6 to 8 million. Then the number of trips reduced dramatically to below 1 million from March to August 2020 and gradually increased to 2-3 million in the year 2022. During the week, it is commonly seen that Thursday and Friday had the most trips. During the day, 3pm and 6pm are most likely to be the busiest time for taxi drivers. In each month, the average passenger number per trip is between 1 and 2 and the average amount paid per trip is around 15-20 dollars compared with the average

	year_month	numberoftrips	dayofweek	hourofday	average_passengers	average_amount_paid_per_trip	average_amount_paid_per_passenger
1	2019-01-01	7739553	Thursday	18	1.55	14.86	9.62
2	2019-02-01	7202004	Friday	18	1.55	17.82	11.52
3	2019-03-01	8031031	Friday	18	1.56	18.34	11.82
4	2019-04-01	7563635	Tuesday	18	1.56	18.45	11.92
5	2019-05-01	7682470	Thursday	18	1.55	18.81	12.16
6	2019-06-01	7039764	Saturday	18	1.55	18.89	12.21
7	2019-07-01	6347126	Wednesday	18	1.56	18.54	11.97
8	2019-08-01	6092175	Thursday	18	1.56	18.56	11.99
9	2019-09-01	6583009	Thursday	18	1.54	18.95	12.41
10	2019-10-01	7231464	Thursday	18	1.53	18.85	12.46
11	2019-11-01	6860643	Friday	18	1.53	18.48	12.16
12	2019-12-01	6880891	Tuesday	18	1.54	18.64	12.24
13	2020-01-01	6360748	Friday	18	1.51	17.52	11.81
14	2020-02-01	6273517	Saturday	18	1.50	17.66	11.93
15	2020-03-01	2982461	Tuesday	18	1.46	17.18	11.92
16	2020-04-01	221742	Wednesday	15	1.29	14.83	11.99
17	2020-05-01	304539	Friday	15	1.31	15.60	12.75
18	2020-06-01	502011	Tuesday	15	1.36	16.46	12.65
19	2020-07-01	729430	Thursday	15	1.38	16.50	12.37
20	2020-08-01	926242	Monday	15	1.41	16.54	12.12
21	2020-09-01	1248028	Wednesday	15	1.42	16.45	11.94
22	2020-10-01	1569632	Thursday	15	1.43	16.39	11.78
23	2020-11-01	1406618	Monday	14	1.41	16.23	11.78
24	2020-12-01	1356882	Tuesday	15	1.42	16.13	11.68
25	2021-01-01	1244915	Friday	15	1.41	15.81	11.52
26	2021-02-01	1252621	Friday	15	1.41	16.19	11.76
27	2021-03-01	1764323	Wednesday	15	1.40	16.37	11.93
28	2021-04-01	1994348	Friday	15	1.41	17.01	12.32
29	2021-05-01	2315006	Saturday	15	1.42	17.35	12.53
30	2021-06-01	2637707	Wednesday	18	1.44	18.23	12.82
31	2021-07-01	2590135	Thursday	18	1.46	18.37	12.71
32	2021-08-01	2549586	Tuesday	18	1.44	18.58	13.06
33	2021-09-01	2720999	Thursday	18	1.43	19.22	13.61
34	2021-10-01	3234109	Friday	18	1.43	19.08	13.53
35	2021-11-01	3244435	Tuesday	18	1.42	19.39	13.80
36	2021-12-01	2993772	Thursday	15	1.44	19.43	13.62
37	2022-01-01	2265355	Monday	17	1.39	17.26	12.48
38	2022-02-01	2768712	Saturday	18	1.39	18.31	13.20
39	2022-03-01	3358147	Thursday	18	1.39	19.19	13.85
40	2022-04-01	3329488	Friday	18	1.41	19.72	14.00

amount paid per passenger at approximately 10-15 dollars.

- As presented in in Figure 8, the values of average, median, minimum and maximum are calculated for three variables – trip duration, trip distance and speed.

	color	average_duration	median_duration	min_duration	max_duration	average_distance	min_distance	max_distance	average_speed	median_speed	min_speed	max_speed
1	green	16.85	12.58	2.00	179.87	5.89	3.54	134.38	19.63	17.70	0.16	90.28
2	yellow	13.98	10.97	2.00	180.00	4.46	2.72	191.03	17.88	16.25	0.16	104.61

Figure 8. Trip duration, distance and speed by taxi color

It is found that:

- The average and median trip duration is 13-17 minutes and 10-13 minutes respectively
- The average and median trip distance is 4-6 km and 2-4 km respectively

- The average and median speed is 17-20 km/h and 16-18km/h respectively
 - Compared to yellow taxi trips, a green taxi had a higher average and median values for all three variables
 - The maximum values for all three variables are higher during yellow taxi trips than in green taxi trips
3. The percentage of trips where drivers received tips is around 69.82% (Figure 9).
 4. For trips where the driver received tips, there are around 2.63% of trips where the driver received tips of at least \$10 (Figure 10).

	percentage_with_tips ▲
1	69.82

Figure 9. Percentage of trips where drivers received tips

	percentage_with_tips_over10dollars ▲
1	2.63

Figure 10. Percentage of trips where drivers received tips of at least \$10

5. To explore the impact of trip duration on speed (km/h) and distance per dollar (km/\$), the trip duration is separated into 6 bins as shown in Figure 11. On average, the speed varied from 16 to 25 km/h among different bins of trip durations. The average speed was highest when the trip duration was between 30 and 60 minutes and lowest when a trip lasted 5-20 minutes. For each dollar, the average distance ranged from 130m to 400m. It is interesting to discover that the average distance per dollar increased with the length of trip durations. When the trip duration was under 5 minutes, drivers only drove 130m on average to earn a dollar in contrast to 400m when the trip duration was over 1 hour.

	trip_duration_bins ▲	average_speed ▲	average_distance_per_dollar ▲
1	a. Under 5 mins	18.98	0.13
2	b. From 5 mins to 10 mins	16.72	0.17
3	c. From 10 mins to 20 mins	16.87	0.23
4	d. From 20 mins to 30 mins	19.39	0.29
5	e. From 30 mins to 60 mins	24.51	0.36
6	f. At least 60 mins	22.86	0.40

Figure 11. Average speed and distance per dollar for bins of durations

6. As discussed in point 5, when the trip duration was under 5 minutes, the distance per dollar on average is the shortest. As the longer distance means that drivers have to pay more for petrol costs and servicing costs, the drivers will end up with less money when the trip duration is longer than 5 minutes. Another way to look at this is to calculate the average dollars per hour within different bins of trip durations (Figure 12). It is discovered that on average the drivers received the most money per hour (around \$145) when trips lasted less than 5 minutes.

Considering both the average distance per dollar and average dollar per hour, it is suggested that drivers should take more short trips (under 5 minutes) to maximise their income.

	trip_duration_bins ▲	average_dollar_per_hour ▲
1	a. Under 5 mins	145.21
2	b. From 5 mins to 10 mins	95.49
3	c. From 10 mins to 20 mins	73.16
4	d. From 20 mins to 30 mins	67.45
5	e. From 30 mins to 60 mins	69.91
6	f. At least 60 mins	55.88

Figure 12. Average dollar per hour for bins of durations

Machine Learning Models

With the data ingested and prepared in the previous section, we can now use this data to build a regression model to predict the total fare amount. Our intent here is to use all the data except April 2022 to train two different machine learning models and choose the best model to predict the total fare amount for taxi trips in April 2022.

Data Quality Check and Imputation

Based on the check, the number of missing values for main fields is listed in Table 2. To minimise the impact on the performance of machine learning models, the imputation is carried out in two methods:

- For fields related to fees and surcharges (including airport_fee, ehail_fee, improvement_surcharge and congestion_surcharge), the null values are replaced with 0.
- For other fields with limited unique values (including passenger_count, RatecodeID, payment_type and trip_type), the most frequent value is used to replace the missing values, which is 1 based on the summary in [Appendix C](#).

Table 2 Count of null values for main variables

Fields	Count of Null Values
pickup_datetime	0
dropoff_datetime	0
passenger_count	1,506,715
trip_distance	0
RatecodeID	1,506,715
payment_type	1,506,715
fare_amount	0
extra	0
mta_tax	0
tip_amount	0
tolls_amount	0
improvement_surcharge	1
total_amount	0
congestion_surcharge	6,515,722
airport_fee	114,252,815
trip_duration	0
speed	0
ehail_fee	149,397,212
trip_type	142,239,033
color	0

Feature Extraction

Features extracted from the initial columns are as follows:

- year_month: value from 2019-01 to 2022-04 to capture each year and month of the records
- dayofweek: value from 1 for a Sunday through to 7 for a Saturday
- hour: value from 1 to 24 to capture the hour of the day in the records

Features for Modelling

Based on the correlation analysis in [Appendix D](#), the following variables (excluding fare_amount) have a stronger relationship with the total amount compared to other variables.

- trip_distance
- trip_duration
- toll_amount
- tip_amount
- airport_fee
- speed

However, since the variable trip_distance is highly correlated to trip_duration (0.84) and speed (0.64), this variable is not included as a feature in the model. The features and the target are summarised in Table 3.

Table 3 Summary of features and target

Features/Target	Variable names
Numerical Feature	trip_duration
Numerical Feature	toll_amount
Numerical Feature	tip_amount
Numerical Feature	airport_fee
Numerical Feature	speed
Target	total_amount

Model Development and Evaluation

The taxi trips from January 2019 to March 2022 are used and split into training and testing sets with the 80-20 ratio, followed by the data processing with VectorAssembler and ML Pipelines. Two algorithms - Multiple Linear Regression and Decision Tree are deployed to build machine learning models aimed at predicting the total fare amount in April 2022. The Root Mean Square Error (RMSE) score for each model is presented in Table 4.

Table 4 RMSE score for ML models

ML Model	RMSE - training set	RMSE - testing set
Multiple Linear Regression (MLR)	3.0517	3.0577
Decision Tree (DT)	3.8250	3.8257

RMSE is a metric used for model evaluation. The lower the RMSE, the better the model and its predictions. Compared to the DT model, the MLR model gives a lower value of RMSE for both training and testing sets. Therefore, the MLR model is regarded as a better model and utilized to predict the total fare amount for taxi trips in April 2022.

Prediction for Taxi Trips in April 2022

Using the MLR model, we predict the total fare amount for taxi trips in April 2022 (Figure 13). The RMSE on prediction is calculated as 3.3701 (Figure 14).

features	total_amount	prediction
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [613, 8.2]}	9.3	10.916584552815822
▶ {"vectorType": "dense", "length": 5, "values": [1019, 0, 3.11, 0, 7.5]}	18.66	18.944743029420792
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [536, 8.9]}	8.3	10.369598719464546
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [352, 9.2]}	6.8	8.421307346543871
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [492, 10.2]}	11.05	10.477430259729585
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [468, 8.7]}	7.8	9.504369137380255
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [845, 17.5]}	15.8	17.89389391023175
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [773, 6.5]}	10.3	11.938525344899235
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [902, 10.9]}	15.55	15.457955294647348
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [777, 12.5]}	13.3	14.786558335568118
▶ {"vectorType": "dense", "length": 5, "values": [1103, 0, 1, 0, 6.3]}	15.3	16.795462717217898
▶ {"vectorType": "sparse", "length": 5, "indices": [0, 4], "values": [995, 9.6]}	17.55	15.906279878705066

Figure 13. Prediction sample for total fare amount of April 2022 trips

```

1 # Calculate the RMSE on April 2022 prediction
2 print(f"RMSE for testing set is {lr_evaluator.evaluate(pred_apr):0.4f}")

```

▶ (1) Spark Jobs

RMSE for testing set is 3.3701

Figure 14. RMSE on prediction for total fare amount of April 2022 trips

Reflection on the Project

There are several issues worth reflecting on for this project. The details are discussed below.

1. Dealing with the data type issue for 'airport_fee'. It is found that the field 'airport_fee' in the yellow taxi dataset has different data types (double or integer) in different parquet files. The inconsistency in data type triggered the error when all the yellow taxi data was read together and saved into DBFS. The workaround solution is to separate the parquet files of the yellow taxi into two groups – double and integer, read them separately and append the records together. To be more proactive in future projects, it would be better to check the schema in all the parquet files first to identify any data type issues.
2. The efficient way to perform data cleaning and model building. When working with a big dataset like NYC taxi trips, it takes quite a while to clean the whole dataset and build a machine learning model considering the iterative process involved. One of the recommended solutions is to run a sub-dataset first and set up the steps for data processing and model fitting before running the whole data and getting the final result. Other things that could help to reduce the processing time in Databricks include minimizing the number of cells in the ipynb file, clearing the Garage Collection, and starting with fewer features to fit the model before using more.
3. Limitations of the current machine learning models. One of the limitations is that the models have not considered the impact of seasonality or covid on the total fare amount. For example, during the months with public holidays such as December, the total fare amount may be greater

than in other months due to the surcharges. Also, based on the data exploration, the number of taxi trips dropped significantly during the covid times. There are likely to be any discounts for the trips to encourage the public to get a taxi.

Conclusion

In this project, we applied Databricks Spark to process and analyse the data of New York City taxi trips in the past 4 years and built two machine learning models to predict the total fare amount. The features used in the model include trip duration, taxi speed, tip amount, toll amount and airport fee. By using the metric RMSE, the Multiple Linear Regression model was evaluated to have better performance and used to predict the total fare amount for April 2022 taxi trips. Some potential improvements can be worked on in the future, such as tuning the hyperparameters in Decision Tree model, exploring other machine learning algorithms and including the feature of seasonality or the impact of covid in the model.

Appendix

Appendix A - Data Dictionary

Data Dictionary for Yellow Taxi Dataset

Field Name	Description
VendorID	A code indicating the TPEP provider that provided the record. 1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.
tpep_pickup_datetime	The date and time when the meter was engaged.
tpep_dropoff_datetime	The date and time when the meter was disengaged.
Passenger_count	The number of passengers in the vehicle. This is a driver-entered value.
Trip_distance	The elapsed trip distance in miles reported by the taximeter.
PULocationID	TLC Taxi Zone in which the taximeter was engaged
DOLocationID	TLC Taxi Zone in which the taximeter was disengaged
RateCodeID	The final rate code in effect at the end of the trip. 1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server. Y= store and forward trip N= not a store and forward trip
Payment_type	A numeric code signifying how the passenger paid for the trip. 1= Credit card 2= Cash 3= No charge 4= Dispute 5= Unknown 6= Voided trip
Fare_amount	The time-and-distance fare calculated by the meter.
Extra	Miscellaneous extras and surcharges. Currently, this only includes the \$0.50 and \$1 rush hour and overnight charges.
MTA_tax	\$0.50 MTA tax that is automatically triggered based on the metered rate in use.
Improvement_surcharge	\$0.30 improvement surcharge assessed trips at the flag drop. The improvement surcharge began being levied in 2015.
Tip_amount	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.
Tolls_amount	Total amount of all tolls paid in trip.
Total_amount	The total amount charged to passengers. Does not include cash tips.
Congestion_Surcharge	Total amount collected in trip for NYS congestion surcharge.
Airport_fee	\$1.25 for pick up only at LaGuardia and John F. Kennedy Airports





















Data Dictionary for Green Taxi Dataset

Field Name	Description
VendorID	A code indicating the LPEP provider that provided the record. 1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.
lpep_pickup_datetime	The date and time when the meter was engaged.
lpep_dropoff_datetime	The date and time when the meter was disengaged.
Passenger_count	The number of passengers in the vehicle. This is a driver-entered value.
Trip_distance	The elapsed trip distance in miles reported by the taximeter.
PULocationID	TLC Taxi Zone in which the taximeter was engaged
DOLocationID	TLC Taxi Zone in which the taximeter was disengaged
RateCodeID	The final rate code in effect at the end of the trip. 1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride
Store_and_fwd_flag	This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server. Y= store and forward trip N= not a store and forward trip
Payment_type	A numeric code signifying how the passenger paid for the trip. 1= Credit card 2= Cash 3= No charge 4= Dispute 5= Unknown 6= Voided trip
Fare_amount	The time-and-distance fare calculated by the meter.
Extra	Miscellaneous extras and surcharges. Currently, this only includes the \$0.50 and \$1 rush hour and overnight charges.
MTA_tax	\$0.50 MTA tax that is automatically triggered based on the metered rate in use.
Improvement_surcharge	\$0.30 improvement surcharge assessed on hailed trips at the flag drop. The improvement surcharge began being levied in 2015.
Tip_amount	Tip amount – This field is automatically populated for credit card tips. Cash tips are not included.
Tolls_amount	Total amount of all tolls paid in trip.
Total_amount	The total amount charged to passengers. Does not include cash tips.
Trip_type	A code indicating whether the trip was a street-hail or a dispatch that is automatically assigned based on the metered rate in use but can be altered by the driver. 1= Street-hail 2= Dispatch





















Appendix B – Separate folders in Azure Storage for Yellow Taxi Dataset

Upload the parquet files into two folders in the azure blob storage:

double ("bde-assignment-2/yellow/double")

<input type="checkbox"/>	 yellow_tripdata_2020-08.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-09.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-11.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-12.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-01.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-02.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-03.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-04.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-05.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-06.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-07.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-08.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-09.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-10.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-11.parquet
<input type="checkbox"/>	 yellow_tripdata_2021-12.parquet
<input type="checkbox"/>	 yellow_tripdata_2022-01.parquet
<input type="checkbox"/>	 yellow_tripdata_2022-02.parquet
<input type="checkbox"/>	 yellow_tripdata_2022-03.parquet
<input type="checkbox"/>	 yellow_tripdata_2022-04.parquet

integer ("bde-assignment-2/yellow/integer")

<input type="checkbox"/>	 yellow_tripdata_2019-01.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-02.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-03.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-04.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-05.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-06.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-07.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-08.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-09.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-10.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-11.parquet
<input type="checkbox"/>	 yellow_tripdata_2019-12.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-01.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-02.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-03.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-04.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-05.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-06.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-07.parquet
<input type="checkbox"/>	 yellow_tripdata_2020-10.parquet

Appendix C - Data Quality Check and Imputation

Count the number of records for each value in 'passenger_count' field

	passenger_count ▲	count ▲
1	1	106860892
2	2	21636426
3	3	5858865
4	5	5039976
5	6	3084806
6	0	2770730
7	4	2641345
8	null	1506715

Count the number of records for each value in 'RatecodeID' field

	RatecodeID ▲	count ▲
1	1	144736046
2	2	2437126
3	null	1506715
4	5	406573
5	3	259258
6	99	44286
7	4	9621
8	6	130

Count the number of records for each value in 'payment_type' field

	payment_type ▲	count ▲
1	1	108724417
2	2	38305386
3	null	1506715
4	3	534211
5	4	328845
6	5	181

Count the number of records for each value in 'trip_type' field

	trip_type ▲	count ▲
1	null	142239484
2	1	6939199
3	2	221072

Appendix D - Correction Analysis

Note: The correlation analysis is based on the sub-dataset from January 2022 to March 2022.

	passenger_count	trip_distance	RatecodeID	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge	total_amount	congestion_surcharge	airport_fee	trip_duration	speed	ehail_fee	trip_type	dayofweek	hourofday
passenger_count	1.000000	0.021675	-0.020922	0.008408	0.001329	-0.105778	0.009157	0.008734	0.018217	0.004368	0.001821	0.015302	0.013875	0.016850	0.016006	NaN	-0.002832	0.011239	0.019579
trip_distance	0.021675	1.000000	0.104931	-0.001720	0.066892	0.040766	-0.067741	0.545162	0.645190	0.007909	0.084270	-0.201054	0.620014	0.843463	0.636668	NaN	0.007126	-0.019288	-0.010040
RatecodeID	-0.020922	0.104931	1.000000	-0.026086	0.009823	-0.045268	-0.005209	-0.039300	0.073482	0.002832	0.008836	-0.193320	-0.001016	0.144564	0.017236	NaN	0.013683	-0.003539	-0.030015
payment_type	0.008408	-0.001720	-0.026086	1.000000	-0.000686	-0.026923	-0.257264	-0.455372	-0.016522	-0.318926	-0.010524	-0.217191	0.015765	-0.000170	-0.018459	NaN	0.016289	-0.009035	-0.031154
fare_amount	0.001329	0.066892	0.009823	-0.000686	1.000000	0.003241	-0.000096	0.038626	0.044270	0.009077	0.999647	-0.011130	0.039727	0.062912	0.036333	NaN	0.001585	-0.000136	-0.000987
extra	-0.105778	0.040766	-0.045268	-0.026923	0.003241	1.000000	0.076490	0.049731	0.039508	0.062839	0.006788	0.151782	0.037691	0.043430	-0.012974	NaN	-0.016998	0.005474	0.126543
mta_tax	0.009157	-0.067741	-0.005209	-0.257264	-0.000096	0.076490	1.000000	-0.017603	-0.123106	0.820306	0.001251	0.434098	0.019469	-0.037213	-0.055888	NaN	-0.148618	0.001479	0.012325
tip_amount	0.008734	0.545162	-0.039300	-0.455372	0.038626	0.049731	-0.017603	1.000000	0.433525	0.050286	0.062410	0.001730	0.343907	0.491678	0.314311	NaN	-0.005971	0.000340	0.023669
tolls_amount	0.018217	0.645190	0.073482	-0.016522	0.044270	0.039508	-0.123106	0.433525	1.000000	0.017487	0.063516	-0.103633	0.472785	0.473307	0.430299	NaN	-0.002070	-0.015402	-0.008517
improvement_surcharge	0.004368	0.007909	0.002832	-0.318926	0.009077	0.062839	0.820306	0.050286	0.017487	1.000000	0.012939	0.378469	0.022235	0.012529	0.001403	NaN	-0.004243	0.000531	0.004210
total_amount	0.001821	0.084270	0.008836	-0.010524	0.999647	0.006788	0.001251	0.062410	0.063516	0.012939	1.000000	-0.007516	0.051741	0.077387	0.046855	NaN	0.001028	-0.000196	0.000671
congestion_surcharge	0.015302	-0.201054	-0.193320	-0.217191	-0.011130	0.151782	0.434098	0.001730	-0.103633	0.378469	-0.007516	1.000000	-0.301402	-0.135134	-0.179551	NaN	-0.061017	0.017918	0.020588
airport_fee	0.013875	0.620014	-0.001016	0.015765	0.039727	0.037691	0.019469	0.343907	0.472785	0.022235	0.051741	-0.301402	1.000000	0.455517	0.437692	NaN	-0.004348	-0.029343	0.012564
trip_duration	0.016850	0.843463	0.144564	-0.000170	0.062912	0.043430	-0.037213	0.491678	0.473307	0.012529	0.077387	-0.135134	0.455517	1.000000	0.247032	NaN	0.006771	0.018307	-0.004062
speed	0.016006	0.636668	0.017236	-0.018459	0.036333	-0.012974	-0.055888	0.314311	0.430299	0.001403	0.046855	-0.179551	0.437692	0.247032	1.000000	NaN	0.008955	-0.070131	-0.037941
ehail_fee	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
trip_type	-0.002832	0.007126	0.013683	0.016289	0.001585	-0.016998	-0.148618	-0.005971	-0.002070	-0.004243	0.001028	-0.061017	-0.004348	0.006771	0.008955	NaN	1.000000	0.000122	-0.004822
dayofweek	0.011239	-0.019288	-0.003539	-0.009035	-0.000136	0.005474	0.001479	0.000340	-0.015402	0.000531	-0.000196	0.017918	-0.029343	0.018307	-0.070131	NaN	0.000122	1.000000	0.043268
hourofday	0.019579	-0.010040	-0.030015	-0.031154	-0.000987	0.126543	0.012325	0.023669	-0.008517	0.004210	0.000671	0.020588	0.012564	-0.004062	-0.037941	NaN	-0.004822	0.043268	1.000000