Big Data Engineering

Spring 2022

Assessment – 2

Data Processing on a Big Dataset with Databricks Spark

Rajat Singh

14330397

rajat.i.singh@student.uts.edu.au

1. Introduction

The aim of this project is to analyse New York Taxicab dataset from January 2019 to April 2022, to determine answers to critical business problem statements. Moreover, this project also creates a Machine Learning model to predict the travel fare prices of a taxi ride in New York City.

2. Problem Statement

The following problem statements are given to do and analyse: -

- Storing data in Azure Blob Storage
- Perform data cleaning
- Combining separate datasets for yellow and green taxis and saving the parquet file in DBFS
- Answer critical business questions, such as: -
    - For each year and month (e.g., January 2020 => “2020-01-01” or “2020-01” or “Jan 2020”:
        - What was the total number of trips?
        - Which day of week (e.g., Monday, Tuesday, etc..) had the most trips?
        - Which hour of the day had the most trips?
        - What was the average number of passengers?
        - What was the average amount paid per trip (using total_amount)?
        - What was the average amount paid per passenger (using total_amount)?
    - For each taxi colour (yellow and green):
        - What was the average, median, minimum and maximum trip duration in minutes (with 2 decimals, e.g., 90 seconds = 1.50 min)?
        - What was the average, median, minimum and maximum trip distance in km?
        - What was the average, median, minimum and maximum speed in km per hour?
    - What was the percentage of trips where drivers received tips?
    - For trips where the driver received tips, what was the percentage where the driver received tips of at least $10.
    - Classify each trip into bins of durations and then for each bin, calculate:
        - Average speed (km per hour)
        - Average distance per dollar (km per $)
    - Which duration bin will you advise a taxi driver to target to maximise his income?
- Create a Machine Learning Model to predict total fare amount of a trip and apply that Machine Learning model to April 2022 data.

3. Dataset

Since 1971, the New York City Taxi and Limousine Commission (TLC) has been in charge of issuing licences and enforcing regulations for the city's taxi cabs. TLC made millions of trip data from both yellow and green taxi cabs available to the public.

Each entry contains fields that record the pick-up and drop-off times and locations, trip distances, itemised rates, rate kinds, payment methods, and driver-reported passenger counts.

Yellow taxi cabs are the city's most recognisable taxis and are authorised to pick up passengers who hail them on the street anywhere in New York. Each of the approximately 13,600 authorised taxis in New York City is required to have a yellow medallion attached to it.

To enhance taxi service and availability in the boroughs, green cabs were introduced in August 2013.

3.1 Data dictionary

3.1.2 Yellow

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

3.1.2Green

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

4. Architecture

The project uses three major technologies and SaaS platforms and they are:-

- ## Azure Blob Storage

  Azure Blob Storage is a scalable and secured data management solution by Microsoft that helps to create data lakes.

  Like a directory in a file system, a container organises a group of blobs. An infinite number of containers may be included in a storage account, and an infinite number of blobs may be kept in a container.

- ## Apache Spark

  Apache Spark is a framework for data processing that can swiftly conduct operations on very large data sets and distribute operations across several computers, either alone or in conjunction with other tools for distributed computing. These two characteristics are essential to the fields of big data and machine learning, which both call for immense computing resources to be mobilised in order to process enormous data warehouses.

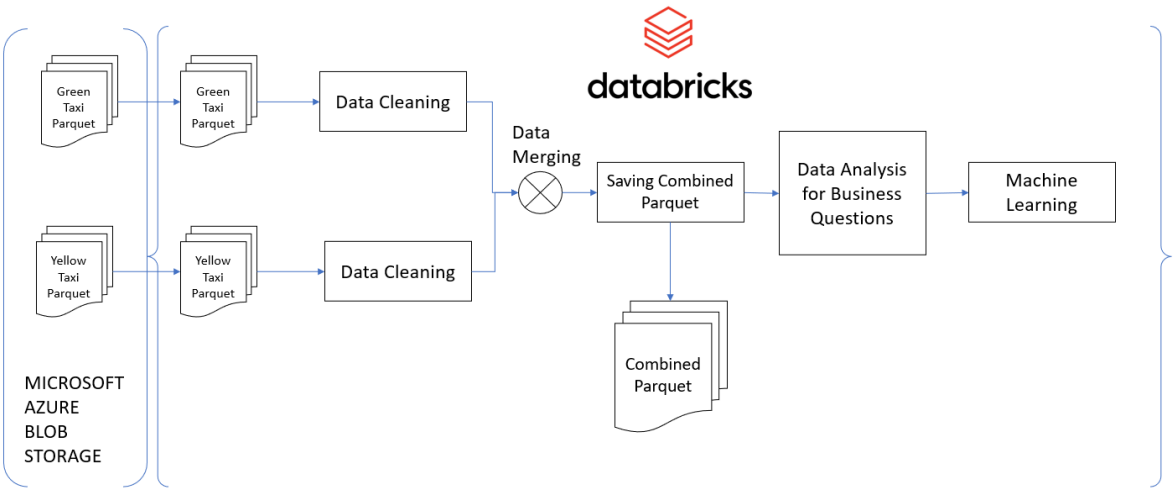
- ## Databricks

  DataBricks was established as a MapReduce substitute and offers big data processing clients a just-in-time cloud-based platform.

  For the purpose of integrating data science, engineering, and the business that supports it across the machine learning lifecycle, DataBricks was developed for data scientists,

engineers, and analysts. The procedures from data preparation to experimentation and the deployment of machine learning applications are made easier by this connection.

5. Project Structure and its Processes



The following process were done to achieve the desired objective for the project.

- The New York Taxi files for green and yellow taxis were uploaded to Microsoft Azure Blob storage.

Azure Resource	*Name*
Storage Account	rajatsinghuts
Storage - Container	bde-at-2

- Then a new cluster was created in Databricks named my cluster. The Blob storage was mounted to Databricks to read the files from blob storage.
- ***Data Cleaning***: After loading these files in databricks the following data cleaning measures were taken: -
    - Trips finishing before the starting time were removed.
    - Trips with negative speeds were removed.
    - Trips with very high speed were removed.
    - Trips that are travelling too short or too long as per their distance were removed.
    - Trips that are travelling too short or too long as per their duration were removed.
    - Trips with more than 5 passengers were removed.
    - Trips that were less than 30 seconds were removed.
    - Trips that had pickup and drop times before 2019-01-01 and after 2022-05-01 were removed.
- ***Data Transformation***: Two new columns were added for data cleaning and answering business questions. These two columns are: -
    - Time_duration : This column has the trip time duration in hours.
    - Speed: This column has the speed in miles per hour
- ***Data Merging***: Since, the schema of the data of both taxi types is not same, the following steps were taken.

- Some columns were renamed. Please see the table below

DATASET	COLUMN NAME	RENAMED COLUMN NAME
YELLOW TAXI	tpep_pickup_datetime	pickup_datetime
YELLOW TAXI	tpep_dropoff_datetime	dropoff_datetime
GREEN TAXI	lpep_pickup_datetime	pickup_datetime
GREEN TAXI	lpep_dropoff_datetime	dropoff_datetime

- A new column named taxi_type is added.
- These two different data frames for yellow and green taxis were merged by setting allow missing column to True, since yellow data frame doesn't have airport fee column and green taxi data frame doesn't have ehail_fee and trip_type column.

- The Combined data frame was saved into 'dbfs/taxis_combined_dataset.parquet'.
- ***Data Analytics:*** To answer critical business questions Spark SQL was used. Please check the section for Business Problems and its Solutions section for more information.
- ***Machine Learning :*** A linear regression model and a decision tree model are trained on the combined dataset using only 3 features, i.e. trip_distance, time_duration, and taxi_type to predict total_amount of a trip.

6. Details of New Files Created

The combined dataset, which has the data for both yellow and green taxis is merged, is stored in dbfs with a filename taxis_combined_dataset.parquet'. This file is used further for answering business questions using spark SQL.

6.1 Data Dictionary

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
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
Speed	Speed in Miles per hour
Time_duration	Time taken during the trip in hours.

7. Business Problems and Its Solutions

The following business problems were answered: -

- *For each year and month (e.g January 2020 => "2020-01-01" or "2020-01" or "Jan 2020":*
  - What was the total number of trips?
  - Which day of week (e.g. monday, tuesday, etc..) had the most trips?
  - Which hour of the day had the most trips?
  - What was the average number of passengers?
  - What was the average amount paid per trip (using total_amount)?
  - What was the average amount paid per passenger (using total_amount)?

	Year	Month	total_number_of_trips	Day_of_week_with_most_trips	Hour_of_week_with_most_trips	average_passenger_count	average_amount_paid_per_trip	average_amo
1	2019	1	7978315	Thursday	18	1.4333164333571689	15.524905926296611	13.225030227
2	2019	2	7305926	Friday	18	1.4384017850714612	18.35393096297747	15.580704977
3	2019	3	8120243	Friday	18	1.4450882812250816	18.782751747061365	15.876046452
4	2019	4	7653491	Tuesday	18	1.4458713023899812	18.8450560919991	15.861384024
5	2019	5	7763616	Thursday	18	1.4418001869232069	19.177839803592015	16.154371680
6	2019	6	7122833	Saturday	18	1.4443142216025562	19.301711478340685	16.232172559
7	2019	7	6435696	Wednesday	18	1.4494663825015974	19.052903144356133	15.964738576

Showing all 40 rows.

- *For each taxi colour (yellow and green):*
  - What was the average, median, minimum and maximum trip duration in minutes (with 2 decimals, eg. 90 seconds = 1.50 min)?
  - What was the average, median, minimum and maximum trip distance in km?
  - What was the average, median, minimum and maximum speed in km per hour?

	taxi_type	average_trip_duration	minimum_trip_duration	maximum_trip_duration	average_trip_distance_in_km	minimum_trip_distance_in_km	maximum_trip_distance_in_km	ave
1	yellow	14.117781233358475	0.5166666666666666	599.9666666666667	4.8314691398944465	0.0160934	311.085422	18.
2	green	14.97293480622997	0.5166666666666666	599.9833333333333	5.069373324396097	0.0160934	275.2132334	19.

- *What was the percentage of trips where drivers received tips?*

	percentage_of_trips_where_drivers_received_tips
1	69.80101303544603

- *For trips where the driver received tips, what was the percentage where the driver received tips of at least $10.*

	percentage_where_the_driver_received_tips_of_at_least
1	3.502919485116748

- *Classify each trip into bins of durations:*
  - Under 5 Mins
  - From 5 mins to 10 mins
  - From 10 mins to 20 mins
  - From 20 mins to 30 mins
  - From 30 mins to 60 mins
  - At least 60 mins

  Then for each bins, calculate:

  - Average speed (km per hour)
  - Average distance per dollar (km per $)

	time_duration_bins	average_speed_kmph	average_distance_per_dollar
1	From 5 mins to 10 mins	16.845600696104185	0.10813943767242058
2	Under 5 Mins	19.325810334383682	0.07755030761602254
3	From 10 mins to 20 mins	17.5059765104798	0.1416167167859619
4	From 20 mins to 30 mins	21.55589617174349	0.17819641743830686
5	At least 60 mins	23.287228914720007	0.28598271394345604
6	From 30 mins to 60 mins	27.289337370953152	0.22561459884388496

- *Which duration bin will you advise a taxi driver to target to maximise his income?*

  From 5 mins to 10 mins bin should be choosen to maximise profit, since average distance per dollar is very low leading to increased profits due to less expense on fuel cost moreover the number of trips is substainally high resulting in more trips.

	time_duration_bins	average_speed_kmph	average_distance_per_dollar	number_of_trips
1	From 5 mins to 10 mins	16.845600696104185	0.10813943767242058	45823417
2	Under 5 Mins	19.325810334383682	0.07755030761602254	21865221
3	From 10 mins to 20 mins	17.5059765104798	0.1416167167859619	53384399
4	From 20 mins to 30 mins	21.55589617174349	0.17819641743830686	18449513
5	At least 60 mins	23.287228914720007	0.28598271394345604	1272877
6	From 30 mins to 60 mins	27.289337370953152	0.22561459884388496	10759472

8. Machine Learning

The following steps were taken to implement machine learning to predict total_amount of a trip:

- The dataset is divided into testing and training dataset. Dates before April 2022 are in the training dataset and on and after April 2022 in testing dataset.
- Trip_distance, time_duration, taxi_type and total_amount features are selected.
- Categorical features are indexed using StringIndexer and then OneHot encoded.
- All the features are assembled in a vector named features.
- Linear Regression and Decision Tree is applied to the selected dataset.
- RMSE score is use to evaluate the model.

Model	**Training RMSE Score**
Linear Regression	154.46254572600506
Decision Tree	154.47532920782268

The Linear Regression model is chosen since, it had better RMSE score.

The testing RMSE score for linear model is `7.081949243051425.`

9. Conclusion

As per the business questions, a Linear regression machine learning model is created with a RMSE score of around 7.08. Moreover, all the business questions are answered using spark sql.