New York City

Taxi Trip Duration

-

Azure (Storage + Databricks)

Aditya Pradip Kulkarni

DTSC 701 - Introduction to Big Data

ID: 1330344

Table of contents:

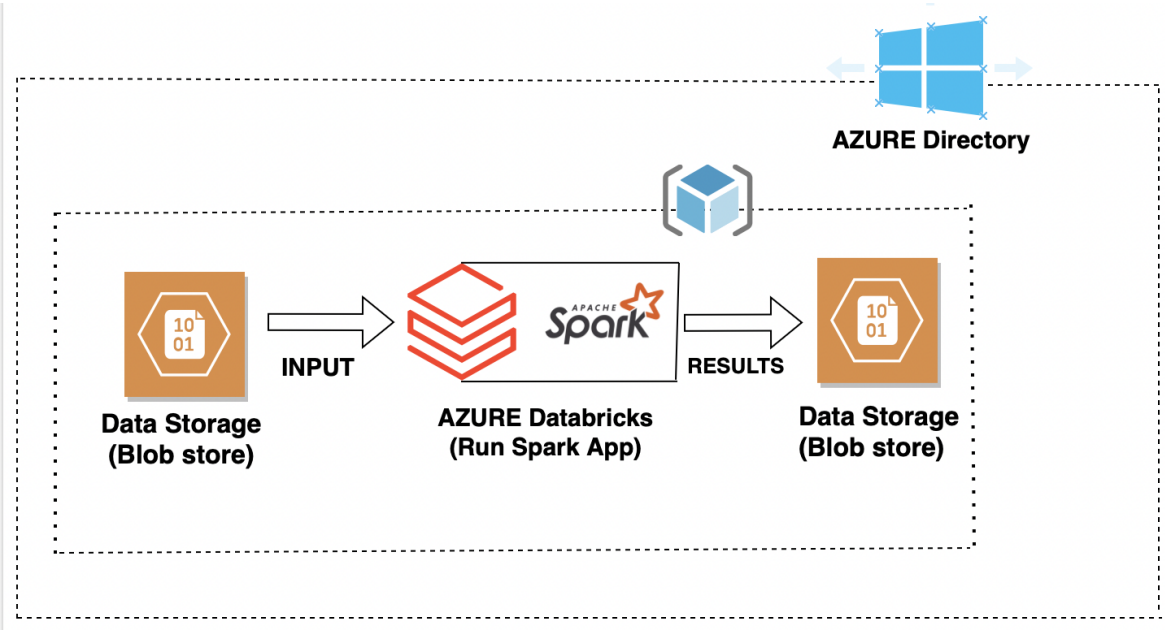
Data Lake:

- I am implementing a data lake solution for performing the data analysis, transformation and later training machine learning model.
- The flow will be azure blob store > store dataset > mount azure blob on azure databricks > perform preprocessing and transformation > train models > store the visualizations in azure blob.
- As my data is not completely transformed and ready to use for training ML models, I need to store the raw data in blob storage and later perform operations on it in azure databricks which will then give me a processed spark dataframe.
- This new spark dataframe is then used for further analysis and training models.
- Unlike data warehouse, which requires storing structured data and performing SQL queries on the data for further analysis.
- Also this solution will provide good scalability in future, when it comes to storing and analyzing large amounts of data.
- Hence, I choose to go with this solution.

G.O.A.L

- Analyze the NYC taxi trip data.
- Store data in Azure blob store.
- Perform the data transformation, preprocessing in Azure Databricks notebook using Apache Spark as processing engine.
- Train machine learning models, and evaluate the models based on certain metrics.
- Store the EDA plots in the blob container for further analysis and use.

Project Architecture



Components:

1. **Azure blob stores (hot store)**: For storing the dataset, Visualization plots, results.
2. **Azure databricks**: Performing Preprocessing, Exploratory data analysis, data transformation, and training machine learning models.
3. **Compute engine**: A compute engine named(ETL Compute), this is the cluster/machine on which the databricks spark notebook will be running and handling the spark jobs. Specifications : 14GB RAM, 4 Core CPU.
4. **Resource group**: A resource group is a logical container for resources deployed in an Azure subscription. It helps you manage and organize resources efficiently.

Dataset

➢ **Data Source:** The provided information pertains to New York City Taxi Cab trips, specifically from the year 2014, sourced from both driver input and GPS coordinates via New York City's Open Data website, the dataset is taken from

Kaggle.

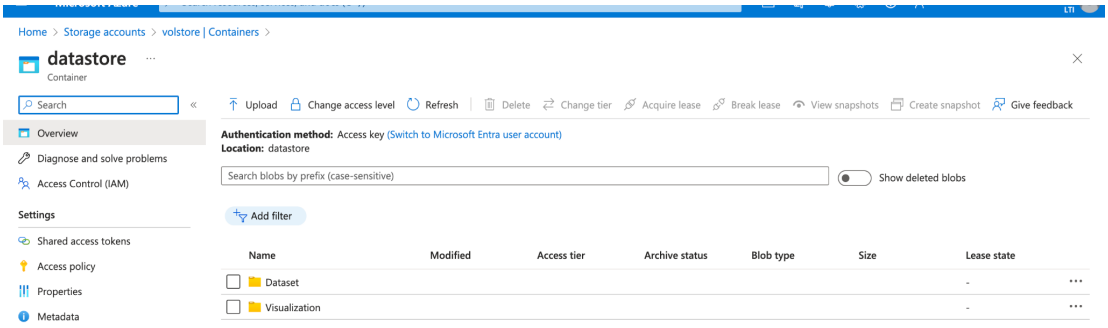
- **Data size**: The data size is 2.36 GB, with shape (14999999, 18)
- **Data Subset:** The shared data represents a subset of the complete dataset, containing the first 15 million records from 2014, compared to the original dataset size of over 165 million trips.
- **Data Format and Fields:** The data is provided in CSV format and includes various fields such as vendor ID, pickup/dropoff datetime, passenger count, trip distance, location coordinates, payment details, fare amount, and additional information collected during the taxi trips.
- **Kaggle**: https://www.kaggle.com/datasets/kentonnlp/2014-new-york-city-taxi-trips/data

Data

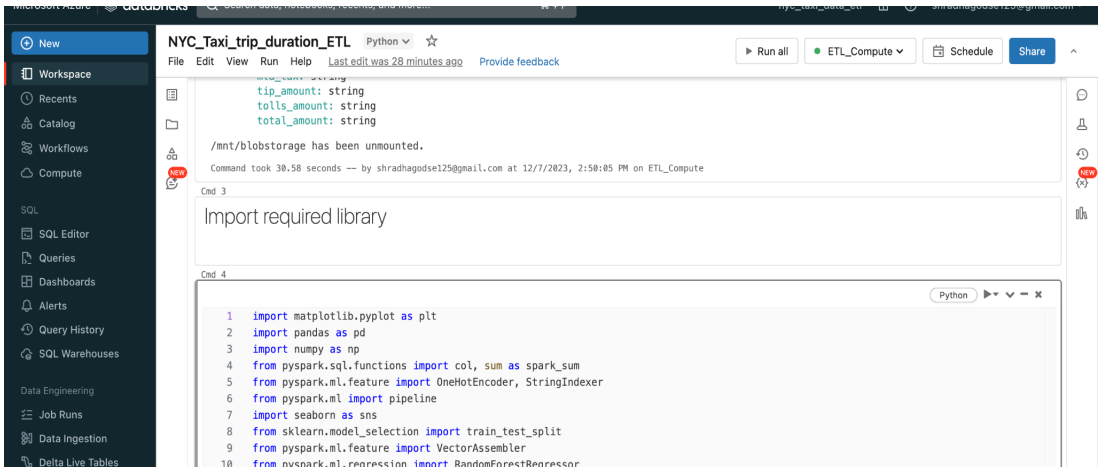
	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dro
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	40.767937	-7
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415	40.738564	-7
2	id3858529	2	2016-01-19 11:35:24	2016-01-19 12:10:48	1	-73.979027	40.763939	-7
3	id3504673	2	2016-04-06 19:32:31	2016-04-06 19:39:40	1	-74.010040	40.719971	-7
4	id2181028	2	2016-03-26 13:30:55	2016-03-26 13:38:10	1	-73.973053	40.793209	-7

Azure services

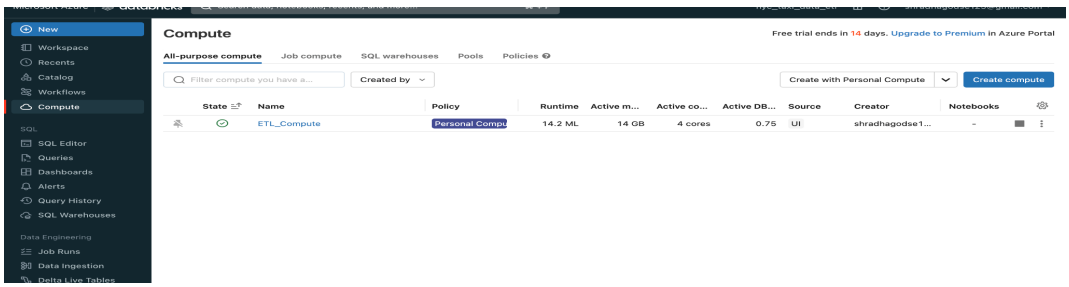
Azure blob store



Azure databricks notebook



Azure compute engine

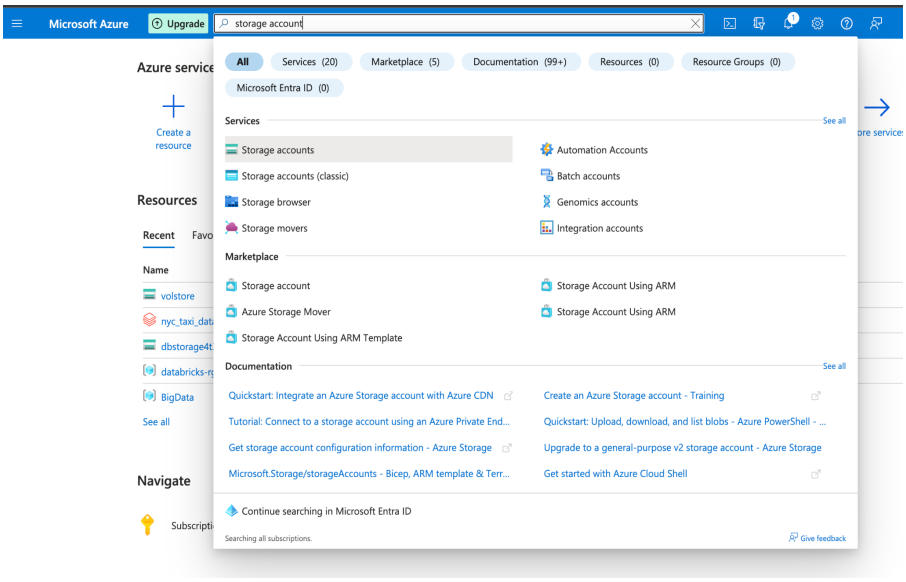


PROCEDURE:

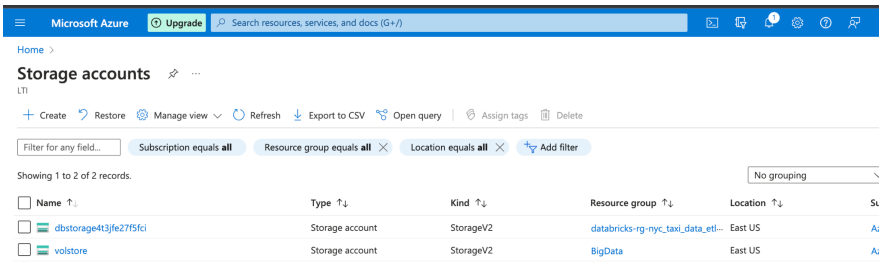
1. Create an Azure account. (Link:https://azure.microsoft.com/en-us/free/).
2. After the account is created, you will get free access to some of the interesting azure tools(Link:https://azure.microsoft.com/en-us/free/#all-free-services).

3. Create a Azure blob storage account:

   i. Type storage account on the search bar situated on the top center part of the azure portal.(*See this written Search resources, service, and docs*)

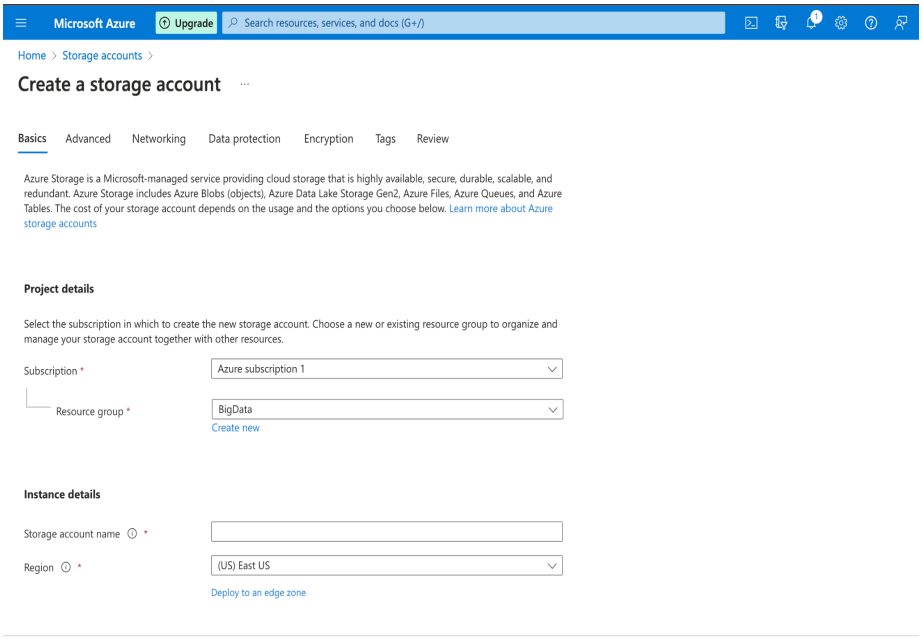


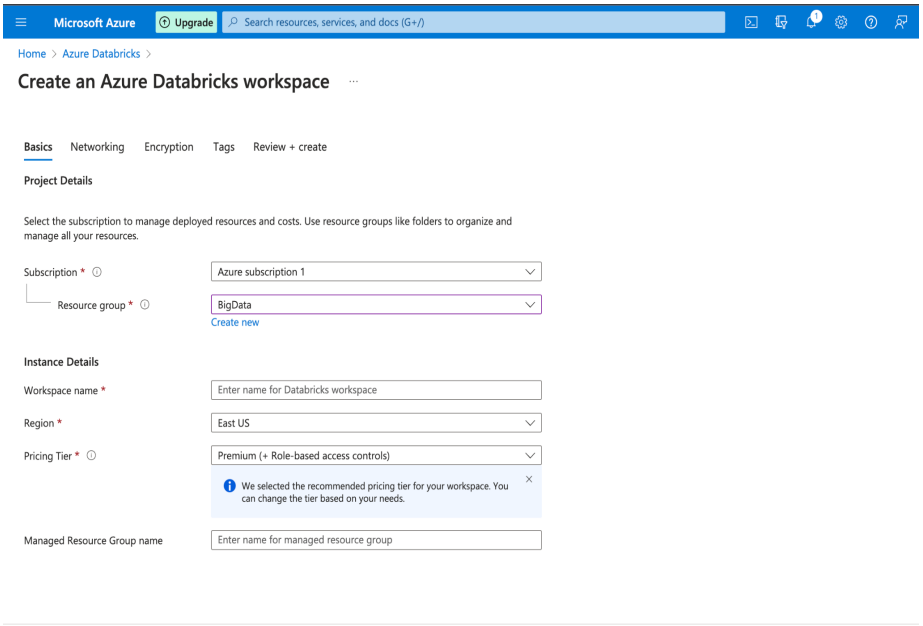
   ii. Click on storage account, and later you will be redirected to the storage account page, click on create option to create a new storage account.



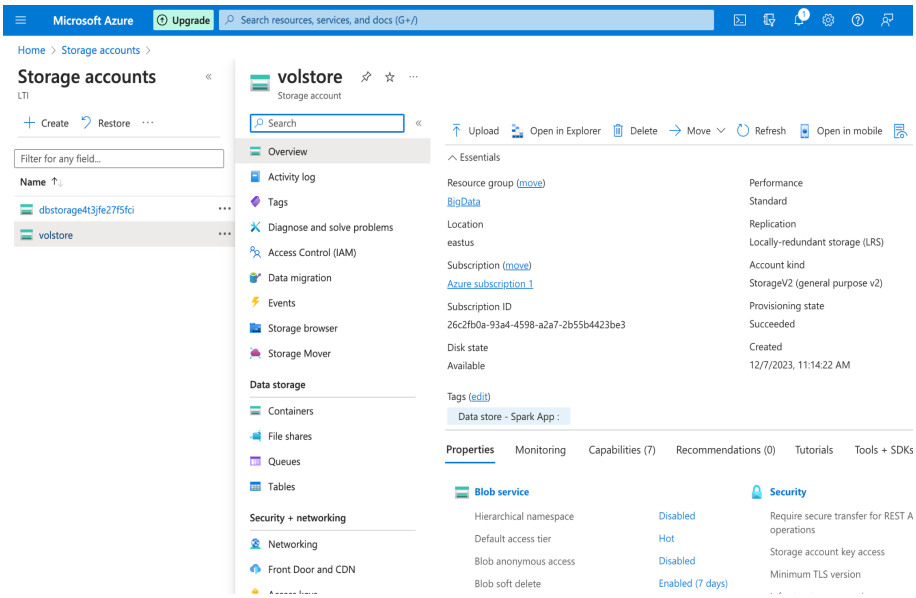
   iii. While creating a storage account, you need to mention the

resource group name you have or wish to create which will hold these resources(blob store, Azure databricks) into a single container.

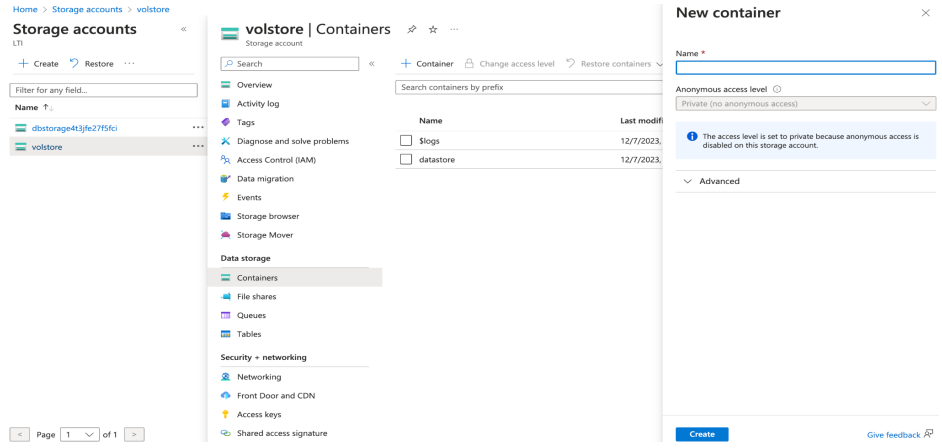




iv. Once you create a storage account, you can see that listed in the storage account landing page, now go to the containers option, which is on the left hand side of the storage account panel.

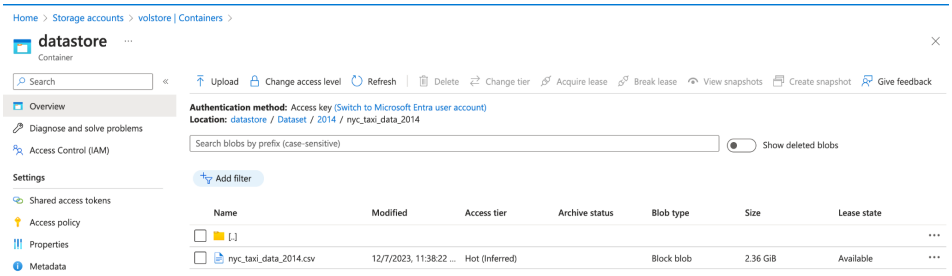


v. Go to the containers page, and create a new container, keep the settings default unless you want to modify some settings as per your requirements.

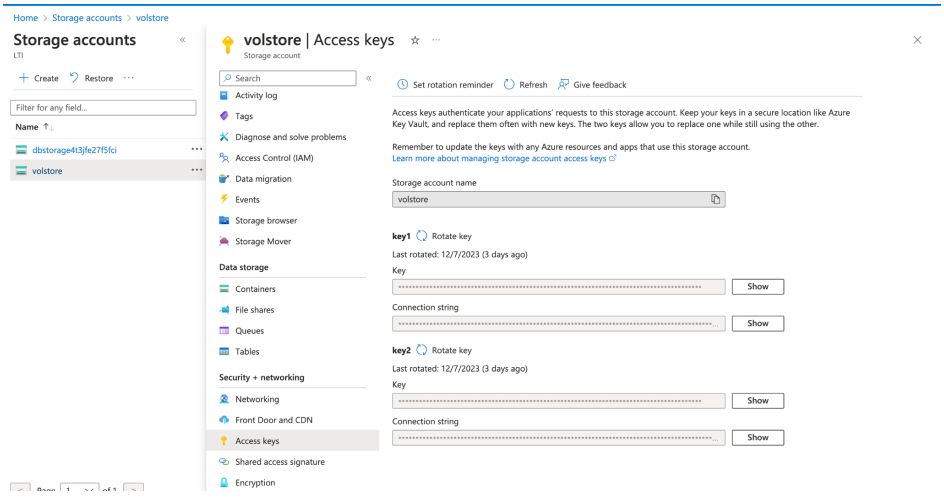


vi. Once the container is created, you will now be able to upload the required data files, any kind of files

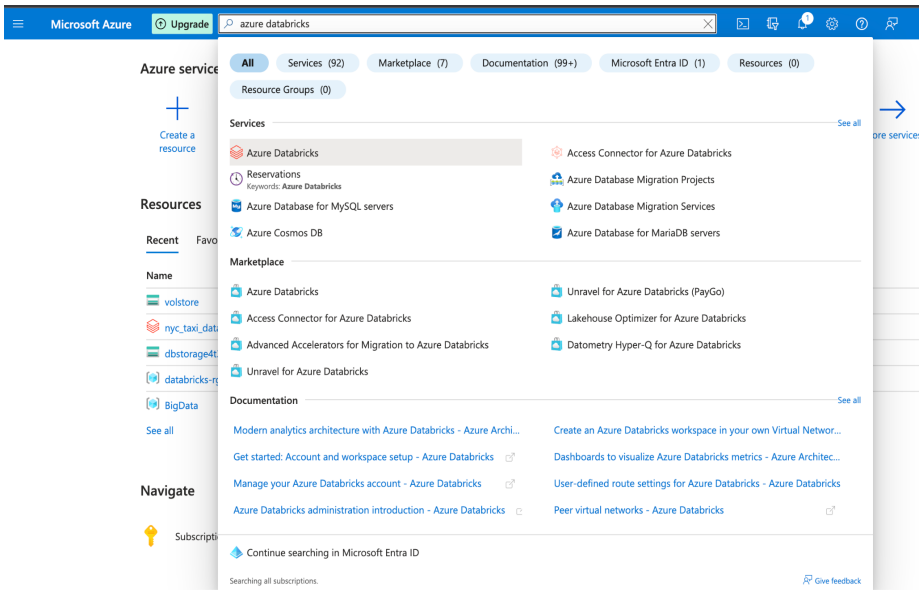
(csv,excel,pdf,parquet, etc.) as this is a storage that supports all kinds and all sizes of formats.



vii. Please kindly copy the container name, storage account name and the storage account key which we will need when we mount the container onto the azure databricks notebook. You can find the storage access key in the access keys option located on the left side of the panel.

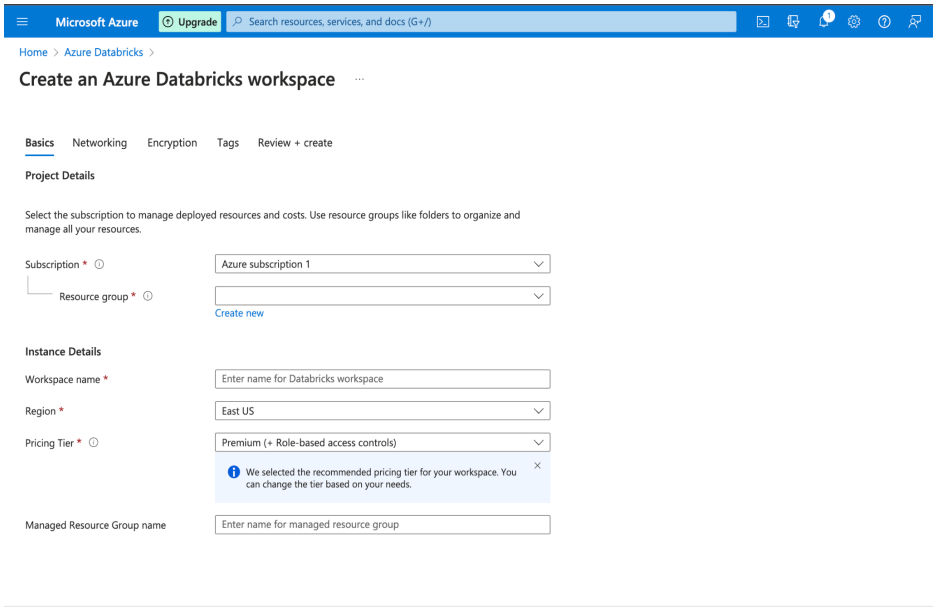


4. Now, on the search bar type databricks, and click on azure databricks option.



5. Once you are on the azure databricks landing page which looks like this:

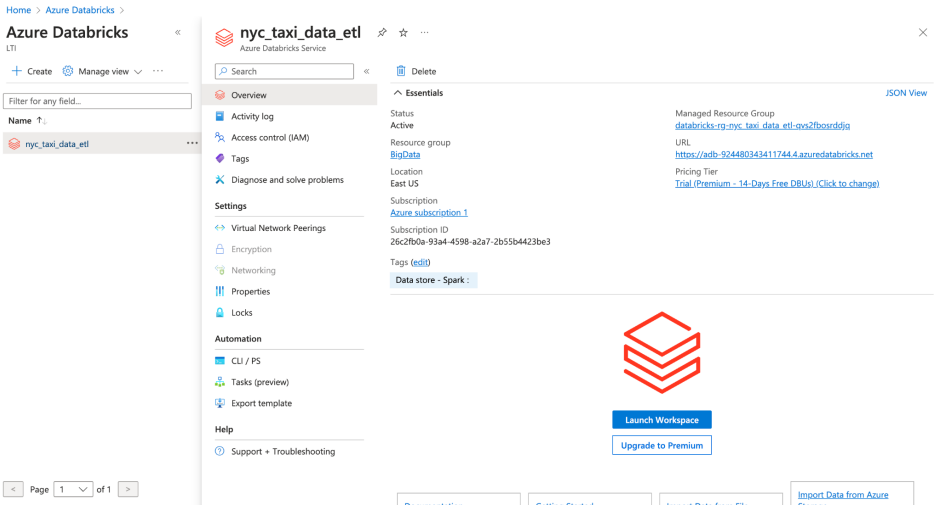
    i. Click on the create workspace button to create a new azure databricks workspace in which you can create notebooks for performing ETL, schedule jobs etc.



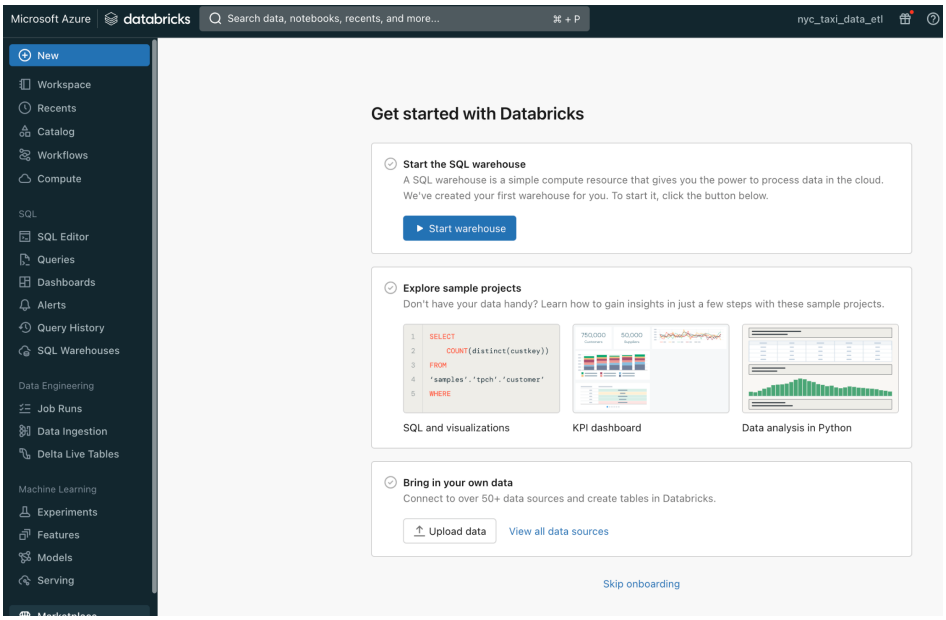
Here, enter the workspace name, select the resource

group where the azure blob storage account is present. Select the pricing as per your requirement(*free trial recommended*).

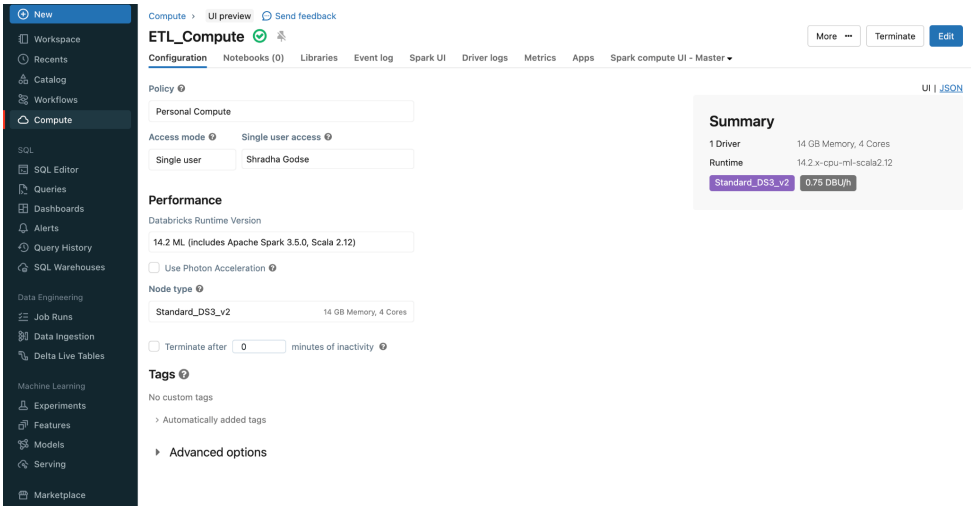
ii. Once the workspace is created you will see the following screen.



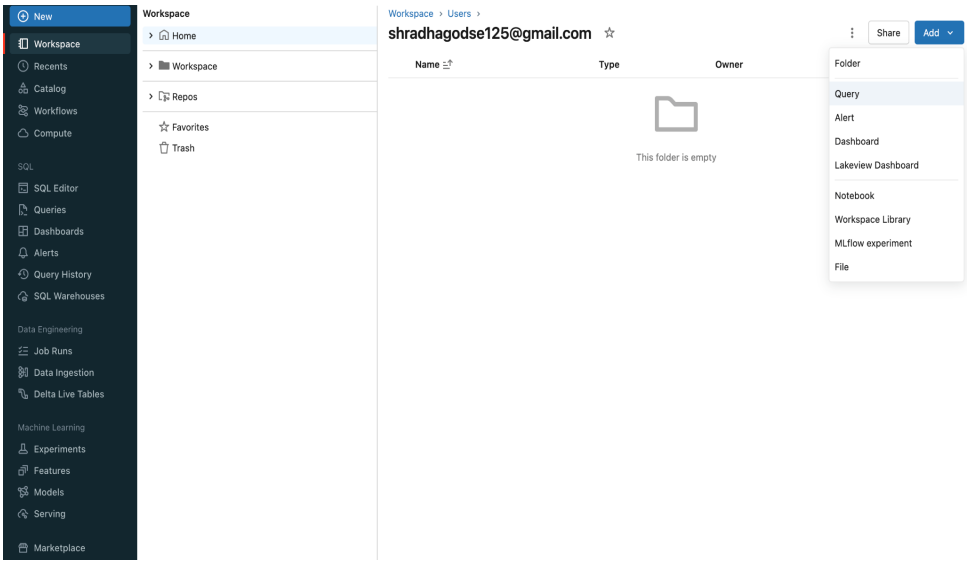
iii. Click on the launch workspace button, which will redirect you to the azure databricks page.



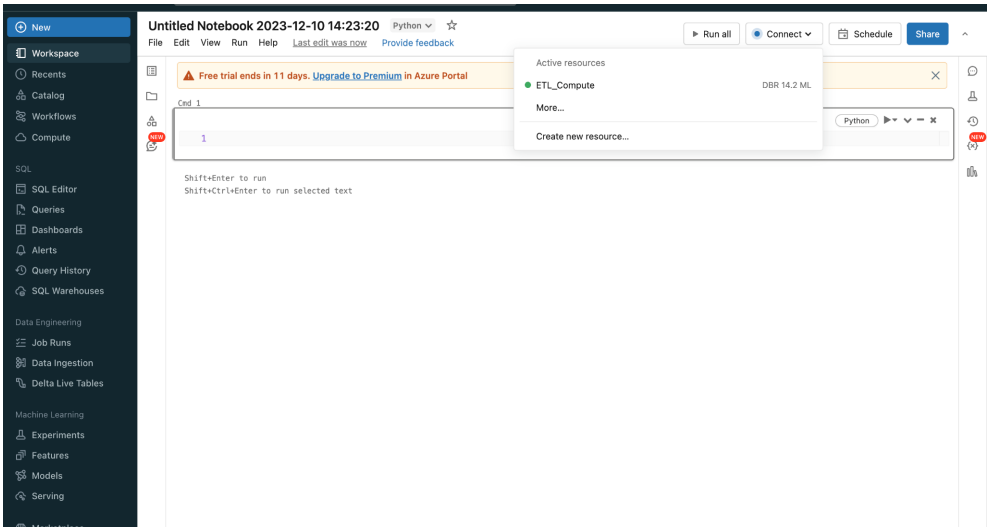
iv. Go to compute option, once you are on the landing page of compute, click on create compute and keep the below settings. *Note: I kept the compute 14GB RAM, 4 Core cpu for handling the workload more efficiently and to improve performance of the overall data processing and training ML model, also it was free for 14 days).*



v. Once the compute is created, go to the workspace option located on the left side of the page. Then click on the Add option on the top right corner of the page. And click on the notebook option.



vi. Once you have created a notebook, now attach the compute we created in Step 2, to the notebook as a compute engine.



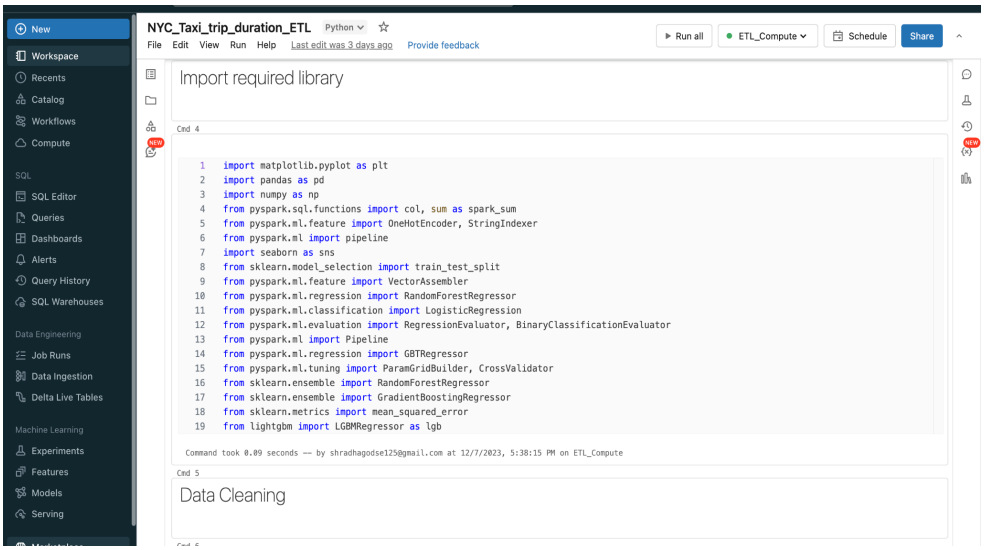
6. Once the notebook is created, you can mount the Azure blob storage on the cluster(ETL Compute) which will act as a local storage for this

notebook. We can do that by using the following code:

```
dbutils.fs.unmount("/mnt/blobstorage")

spark.conf.set(

   "fs.azure.account.key.”storage_account_name”.blob.core.windows.net",

   "Storage_account_access_key"

)

# Define the storage account details

storage_account_name = "storage_account_name"

container_name = "container_name"

storage_account_key = ""

# Mount the Blob Storage container

dbutils.fs.mount(

   source = f"wasbs://{container_name}@{storage_account_name}.blob.core.windows.net/",

   mount_point = "/mnt/blobstorage",

   extra_configs =
{f"fs.azure.account.key.{storage_account_name}.blob.core.windows.net":
storage_account_key}

)

df = spark.read.option("header",
"true").csv("/mnt/blobstorage/Dataset/2014/nyc_taxi_data_2014/nyc_taxi_data_2014.csv")
```

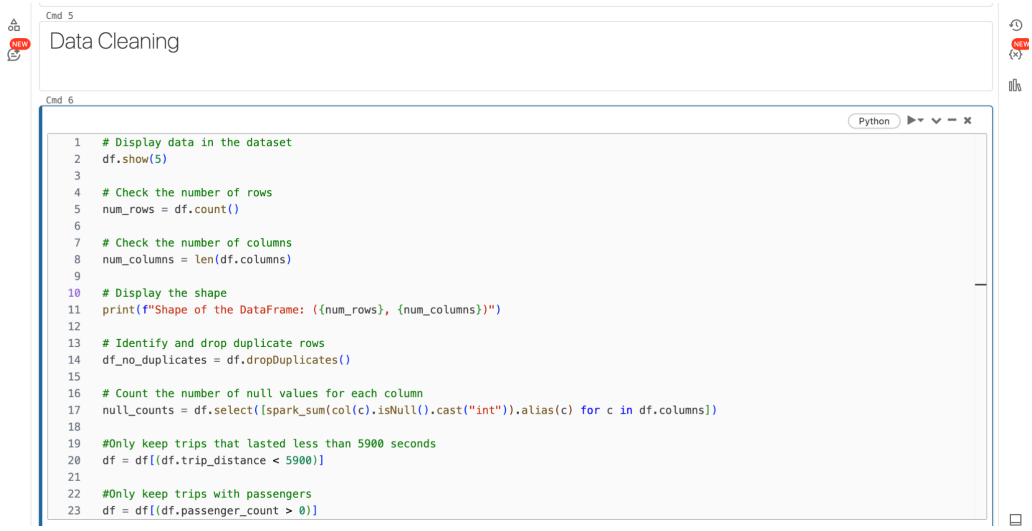
Note: *dbutils.fs.unmount("/mnt/blobstorage") this line is to unmount the existing mounted storage.*

7. Now, we can continue performing the data preprocessing and transformation process inside the databricks notebook. As this is a python + spark notebook all the required libraries are available and we need to use them just by importing into the code.

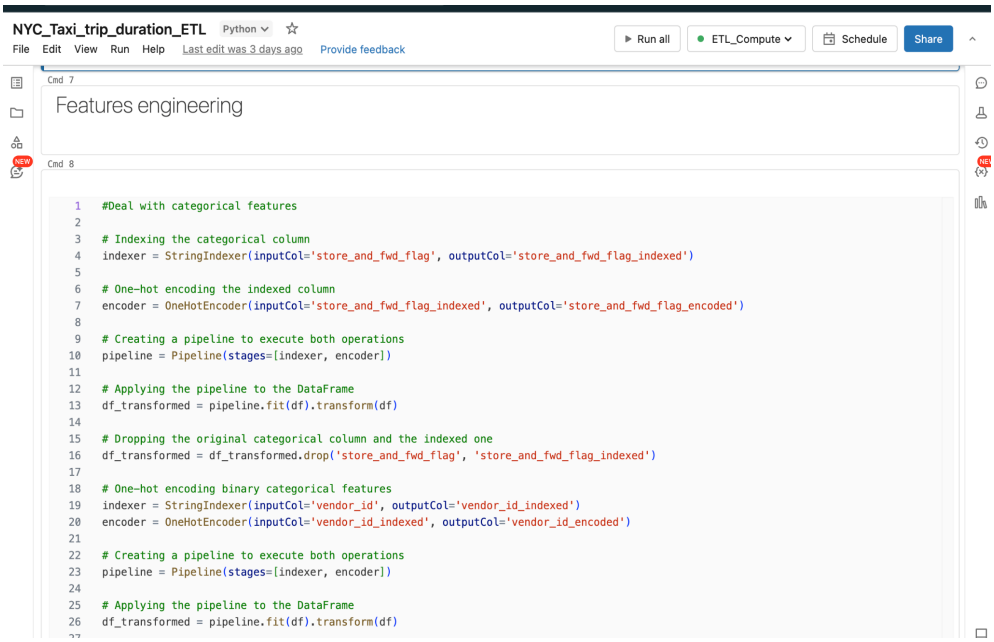


8. The following preprocessing and transformations are performed on the dataset:

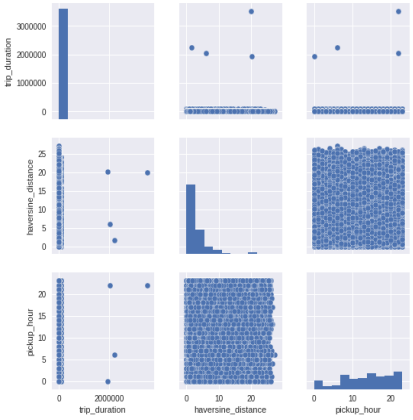
1. **Data cleaning**:
    a. Check the shape of data.
    b. Check for duplicate records.
    c. Check for null values.
    d. Applying filters.

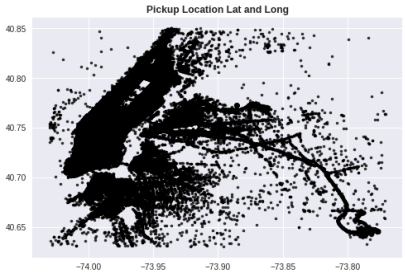


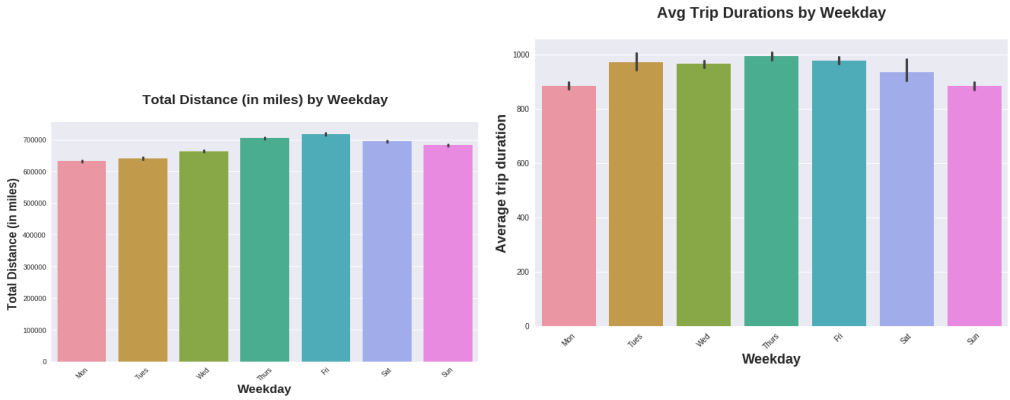
2. **Feature engineering**:
    a. Indexing.
    b. Label encoding categorical columns.
    c. Dropping the columns which are not required.

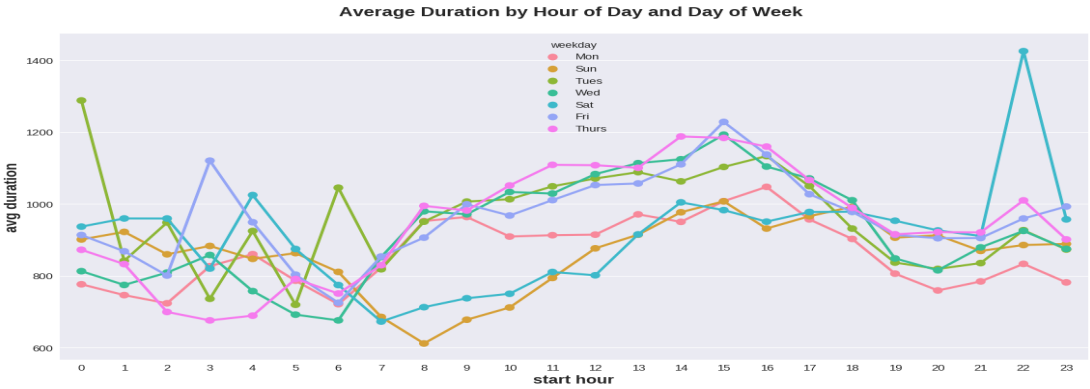


3. **Exploratory Data Analysis**:









4. **Splitting the data**:

Cmd 11

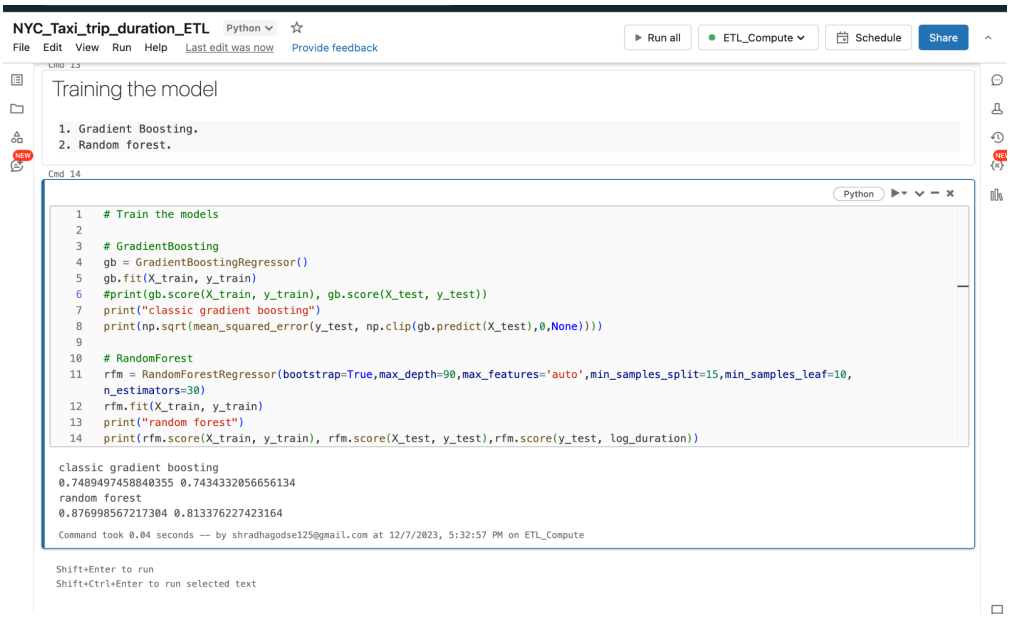
Train & Test Split

Cmd 12

```
df['hour']=subset_train.pickup_hour.astype(str)
df = df[['log_duration','log_haversine_distance', 'weekday','month','hour']]

X = df.drop("log_duration",axis=1)
y = df["log_duration"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

5. **Training the model**:

I have used 2 machine learning models, gradient boosting and random forest model. Where the split size is 80%-20%.



Conclusion:

- Azure distributed cloud service offers a good solution when it comes to data engineering and big data analysis.
- Azure blob helps in storing unstructured data which can help in building a data lake solution.
- Azure databricks provide intensive features which are really beneficial for performing operations on the dataset. Not only ETL, it also provides features to create dashboards, create workflows where we can schedule jobs(notebooks) for execution.
- The dataset is of NYC Taxi trip duration, and the aim was to predict the time duration required for the taxi.

References:

- Azure databricks training.
- Kaggle dataset.
- Azure blob store.