Ride Data Insights – UBER

Submitted in partial fulfillment of the requirements of

PG-DIPLOMA IN BIG DATA ANALYTICS

By

Sachin Dalal	230310125016
Amol Gupta	230310125001
Ashish Kumar	230310125003
Deependra Kureel	230310125004

Project Guide

Mr. Pappu Kapgate

(Faculty, C-DAC, ACTS, New Delhi)

CENTER FOR DEVELOPMENT OF ADVANCED COMPUTING

New Delhi.

March 2023 – September 2023

TABLE OF CONTENTS

CERTIFICATE

This is to certify that the project entitled “Ride Data Insights - UBER” is a teamwork work of “Sachin Dalal (230310125016), Ashish Kumar (230310125003), Amol Gupta (230310125001) & Deependra Kureel (230210125004).” Submitted to C-DAC New Delhi in partial fulfillment of the requirement for the PG- Diploma in Big Data Analytics.

Mr. Pappu Kapagate

(Faculty Supervisor/Guide)

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Sachin Dalal)

(Ashish Kumar)

(Amol Gupta)

(Deependra Kureel)

Date: 04-09-2023

ABSTRACT

The "Ride Data Predictive Insights for Uber" project is a data-driven initiative designed to leverage Uber's extensive dataset for actionable insights. This project encompasses data collection, data pre-processing, exploratory data analysis, and statistical modeling to uncover patterns, trends, and correlations within Uber's operational data. The expected impact of this project is to enabling the company to make informed decisions, streamline its operations, and ultimately provide a safer and more efficient transportation platform for both drivers and Company. By harnessing the power of data analytics, this project is poised to revolutionize Uber's decision-making processes, optimize operational efficiencies, and, most importantly, enhance the safety and efficiency of its transportation platform, benefiting both drivers and passengers alike. This transformative initiative exemplifies Uber's commitment to data-driven excellence in the ride-sharing industry.

1. INTRODUCTION

The Ride Data Insights-UBER project aims to use data science and machine learning techniques to develop models that can help Uber improve its operations. The "Ride Data Insights-UBER" project represents a pioneering endeavor at the intersection of data science, with a paramount objective to proper Uber's operations into a new era of efficiency and customer satisfaction. In today's fast-paced, technology-driven world, the transportation industry is constantly evolving, and Uber is at the forefront of this revolution. This project is a testament to Uber's commitment to innovation and its dedication to providing exceptional service to riders and drivers alike.

The fundamental essence of this project lies in the application of cutting-edge data analysis techniques to Uber's extensive dataset, extracting invaluable dashboard that will revolutionize the ride-sharing experience. By harnessing the power of data, the project aims to achieve several key objectives that are poised to redefine the way Uber operates.

Some of the specific goals of the project include:

- **Predicting rider demand :-** This can help Uber to better match riders with drivers, reduce wait times, and improve the overall customer experience.
- **Estimating surge prices :-** This can help Uber to dynamically adjust prices based on demand and supply, ensuring that riders always have a ride available at a fair price.
- **Estimating total revenue:-** This can help riders to better plan for their future goals & implementation.

2. DESCRIPTION

Rider data predictive insights refer to the analysis and utilization of data collected from Uber riders to make informed predictions and drive strategic decisions within the company. This process involves the application of advanced data analytics, machine learning, and predictive modeling techniques to extract valuable information from rider-related data sets. Here is a more detailed description of rider data predictive insights in the context of Uber:

Data Collection: Uber collects a vast amount of data related to rider behavior and interactions with its platform. This data includes ride history, pick-up and drop-off locations, ride durations, route choices, payment preferences, and feedback provided by riders.

Data Integration: Rider data is integrated from various sources, including mobile apps, payment systems, and customer support interactions. This integrated dataset forms the foundation for predictive analysis.

3. OBJECTIVE

To provide valuable insights and recommendations to Uber by analysing its data in-depth by primarily focus on demand forecasting, route optimization, user behaviour analysis, pricing strategy, and safety enhancements.

The specific purposes of the project include :-

- Demand & Supply Analysis.
- Pricing Strategy.
- Driver Performance.
- Route Optimization.
- User Behaviour Analysis.

4. METHODOLOGY

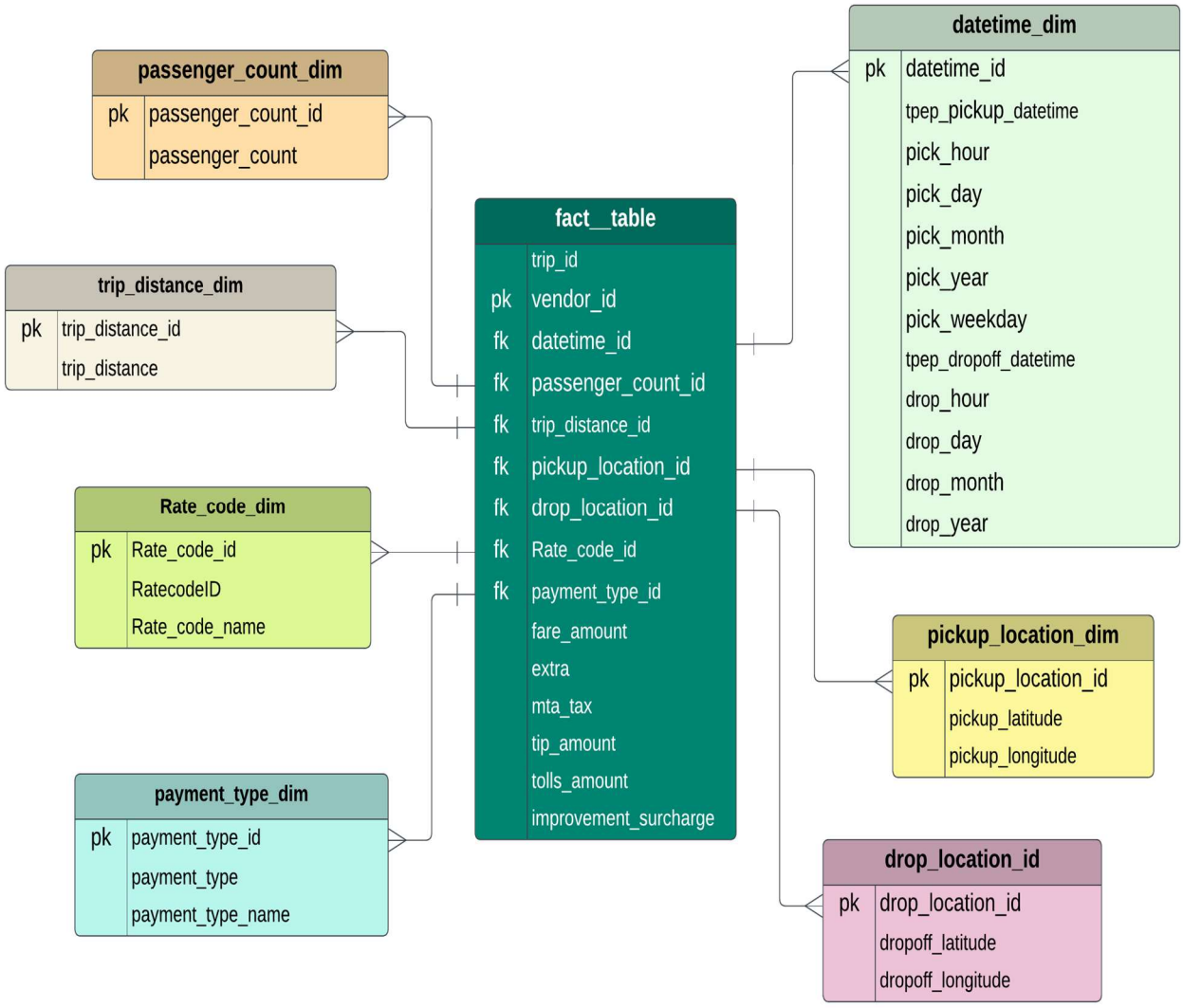
4.1 Data Collection

- Data collection from the NYC.gov: The Uber app generates a lot of data about rider demand, driver availability, and ride prices. This data is collected in real time and stored in a centralized database.
- Data cleaning: The enriched data is then cleaned to remove any errors or inconsistencies. This includes correcting types, and filling in missing values.
- Data transformation: The cleaned data is then transformed into a format that is suitable for Data Pipeline Technology. This may involve converting categorical variables into numerical variables, scaling the data, or creating new features.
- Data storage: The transformed data is then stored in a data warehouse or data lake for future use.

4.2 Data Pre-processing

- Data cleaning: This step involves removing any errors or inconsistencies in the data. This may include removing duplicate records, correcting typos, and filling in missing values.
- Data normalization: This step involves transforming the data so that all of the values are on the same scale. Which often require the data to be normalized.
- Feature engineering: This step involves creating new features from the existing data. This can be done by combining existing features, or by creating new features based on domain knowledge.
- Data sampling: This step involves selecting a subset of the data for analysis. This is often done to reduce the size of the data, or to make the data more manageable

4.3 Uber Data Schema



5. SYSTEM REQUIREMENTS

5.1. Hardware Requirements:

- Processor: Intel Core i5 and above
- Disk Space 2 GB Minimum
- RAM: 8 GB or more
- Cloud Setup

5.2 Software Requirements:

- **Google Cloud Platform:** As the foundation of our Data Analysis infrastructure.
- **Data Pipeline [MAGE]:** for Combining Data from multiple Source.
- **Data Warehouse [GCP Big Query]:** Enable to run SQL queries on massive Data
- **Data Visualization Platform [Looker]:** To create interactive Dashboard and Reports.

5.3 Technologies used:

- **Python :** Python is used for data cleaning, loading, transforming & big query
- **SQL Query:** SQL Query is used to write big sql query in GCP

5.4 . Installation dependencies and set up of project:

- Google Cloud Platform Setup
- MAGE.ai pipeline tool install in VM

6. PROJECT WORKFLOW

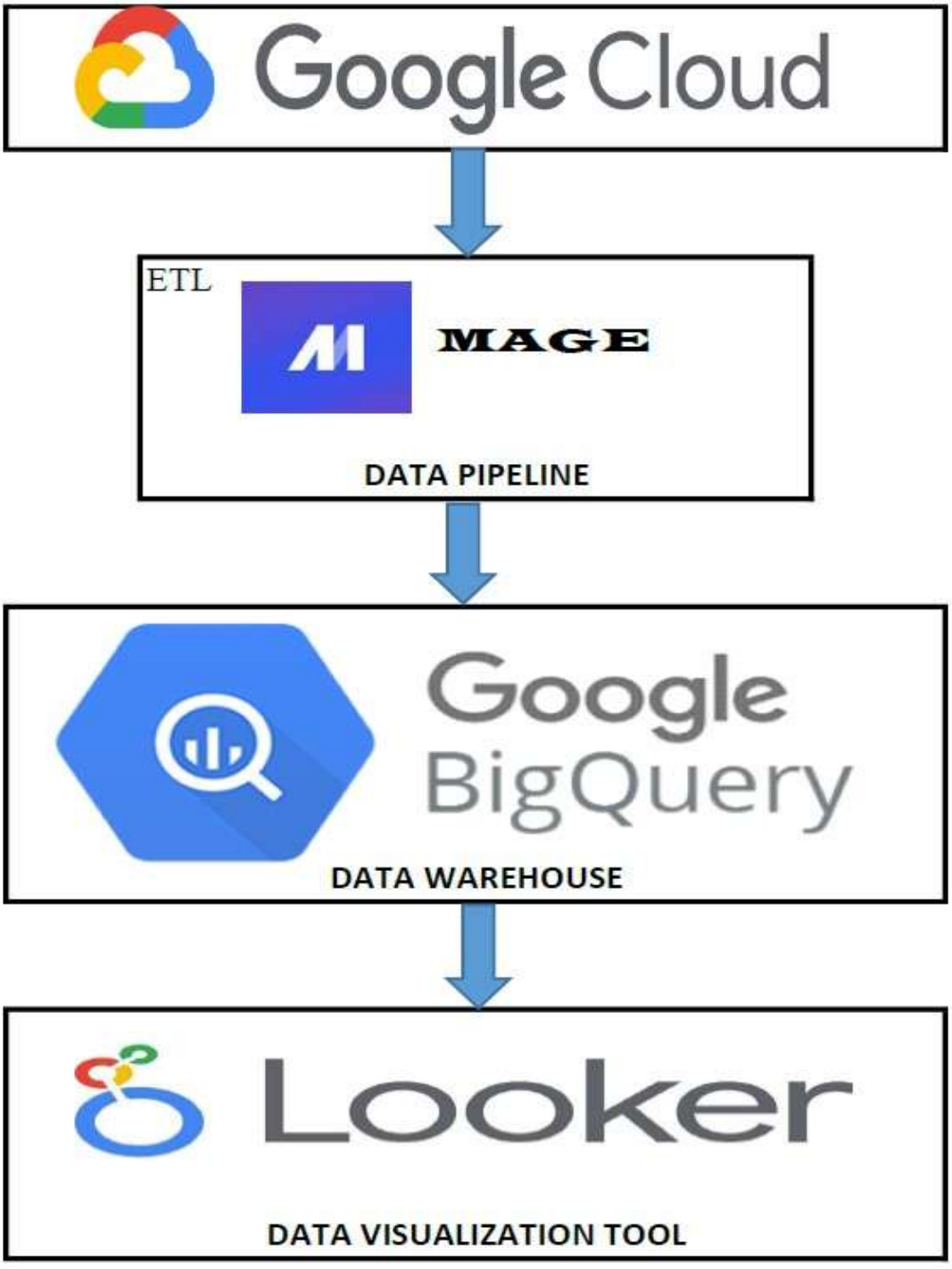


Fig.6.1 Workflow Dia.

7. TECHNICAL ANALYSIS

7.1 DATA ANALYSIS

- Data collection :- The first step is to collect data from Uber rides. This data can be collected from a variety of sources, including the Uber app, third-party data providers, and sensors installed in Uber vehicles.
- Data cleaning :- The next step is to clean the data to remove any errors or inconsistencies. This may involve removing duplicate records, correcting typos, and filling in missing values.
- Data transformation :- The cleaned data is then transformed into a format that is suitable for analysis. This may involve converting categorical variables into numerical variables, scaling the data, or creating new features.
- Data analysis :- The transformed data is then analyzed to gain insights into rider behavior, optimize pricing, and improve the overall user experience. This may involve using statistical methods, machine learning algorithms, or natural language processing techniques.
- Data visualization :- The insights gained from the data analysis can be visualized to communicate them to stakeholders. This may involve creating charts, graphs, or dashboards.

7.2 DATA DICTONARY:

Field Name	Description
Vendor Id	A Code Indicating The Trip Provider That Provided The Record. 1= Creative Mobile Technologies, Llc; 2= Verifone Inc.
Trip_Pickup_Datetime	The Date And Time When The Meter Was Engaged.

Trip_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.

Table 7.2.1 Ref. from NYC.gov.com

7.3 GCP SETUP

GCP is a public cloud vendor — like competitors Amazon Web Services (AWS) and Microsoft Azure. With GCP and other cloud vendors, customers are able to access computer GCP is a public cloud vendor — like competitors Amazon Web Services (AWS) and Microsoft Azure. With GCP and other cloud vendors, customers are able to access computer resources housed in Google's data centers around the world for free or on a pay-per-use basis. GCP offers a suite of computing services to do everything from GCP cost management to data management to delivering web and video over the web to AI and machine learning tools.

GCP Setup Steps:

Create an account: Sign up for a gmail account and you are ready to go.

GCP Signup for Free Trial: Go TRY IT FREE. At first try, you have to agree with GCP's terms and conditions, and provide necessary details, etc. Continue and finish the setup part.

7.4 DATA PIPELINE WITH MAGE FRAMEWORK

Mage is a cutting-edge data science tool that uses machine learning and AI to accelerate and improve data engineering procedures. It is an easy-to-use yet powerful open-source data pipeline tool for data integration and transformation that can make a strong competition for well-known products like Airflow. Mage

transforms the data processing workflow and the way data is handled and processed by fusing the power of automation and intelligence. With its unrivalled features and user-friendly interface, Mage aims to optimize and simplify the data engineering process.

Data Pipeline Mage Steps

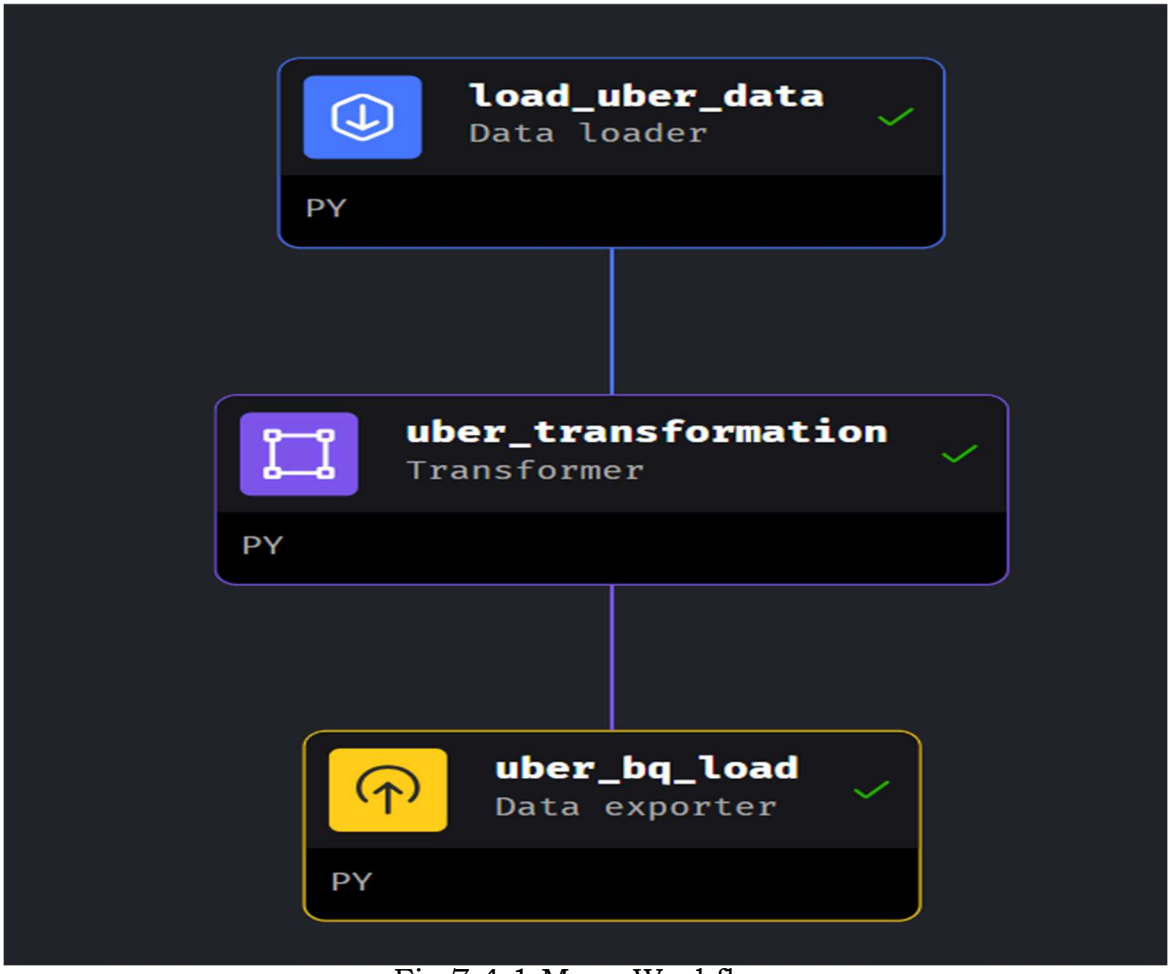


Fig 7.4.1 Mage Workflow

7.5 Create a new pipeline in Mage.ai:

In order to create a new pipeline, we must establish a new project and initiate a tool launch.

The command "**mage start [project_name]**" can be utilized to initiate the creation of a new project.

```
~$ mage start new_project
```

By executing the aforementioned command, a new project can be created, and subsequently, a new interface will open in your browser at **localhost:6789/pipelines**.

On clicking the icon new, we can create a new pipeline. Our objective is to establish a pipeline connecting MySQL and BigQuery while incorporating transformer filters.

Add a data loader task to the pipeline:

To include the data loader, I have already set up a table in MySQL. To create the source pipeline, which is MySQL, follow these steps:

1. Access mage.ai and click on the "Data loader" option.
2. Select "Python" and then choose "MySQL" from the available options.
3. The system will generate the necessary code automatically once you select MySQL as the source.
4. So where this is taking the SQL configuration is from this io.config.yaml.The configurations for this pipeline are set by default in the io.config.yaml file. You can modify this file according to your specific requirements.
5. Refer to the highlighted section in the screenshot provided, and make the necessary changes accordingly.

We had replaced 'Your MySQL query' with the query "SELECT * FROM products" for the selected table. Now, let's proceed with editing the YAML file. To locate and edit the YAML file, follow these steps:

1. Look for the YAML file in the left corner of the interface.
2. Select the file and make the necessary edits according to your requirements.

The io_config.yaml file in Mage AI is used to configure the input and output of data for your pipelines. It contains a list of all the data sources and sinks you wish to utilize in your pipelines, along with their respective configuration settings. After entering the details and saving them, you can now proceed to check if your data is being fetched correctly. To do this, click on the "Run" button. Once the process is completed, you will be able to see the fetched details from the database. Please refer to the screenshot below for reference.

7.6 Add a data transformer task to the pipeline:

Mage.ai offers a range of transformers that can be applied to our pipeline, providing various types of functionality. We will see about a few transformers.

To apply filters on top of the data source, follow these steps in mage.ai:

1. Click on "Transformer" in the menu.
2. Select "Python" and then choose "Row actions" from the available options.
3. Click on the "Filter" option to create the transformer for applying filters to the data.

   In the action_code section, I have added a filter condition to filter the data based on the "price_inr" column. The condition specifies that the value of "price_inr" should be greater than 5000.

When you apply this filter and run the test connection, the following output will be displayed:

1. Only the rows where the "price_inr" column value is greater than 5000 will be included.
2. The filtered data will be shown as the output of the test connection.

7.7 Add a data exporter task to the pipeline:

Now the source looks fine. To transfer the filtered data to the destination, which is BigQuery, follow these steps in mage.ai:

1. Click on “Data exporter” in the menu.
2. Select “Python” and then choose “BigQuery” from the available options.
3. Click on the “BigQuery” option to create the destination pipeline for transferring the data to BigQuery.

To configure the table_id and the io.config.yaml file for BigQuery, please make the following modifications:

1. In the io.config.yaml file, locate the section related to the BigQuery destination.
2. Edit the “table_id” field to specify the desired table where you want to store the data in BigQuery.
3. Enter the required details in the corresponding fields such as project_id, dataset_id, table_id, etc.
4. Save the changes made to the io.config.yaml file.
5. Test the run to ensure that the data is successfully transferred to the specified table in BigQuery.

To fill in the required details for the BigQuery configuration, please follow these steps:

1. Create a **service account** in the **Google Cloud Platform** (GCP) console.
2. Ensure that you grant the necessary access to your project and resources for the service account.
3. Assign a suitable role to the service account, granting it the required permissions.
4. Once the service account is created, navigate to “**IAM & Admin**” in the GCP console.
5. Select “**Service Accounts**” and locate the specific service account you created.
6. Under the “**Keys**” section, click on “**Add Key**” and choose the option to create a new key.

7. Select the **JSON format** for the key and proceed to create the new key.
8. Download the JSON file that contains the required fields for the Google Service Account key.
9. You can now either fill in the corresponding fields in the **io.config.yaml** file with the contents of the JSON file or provide the path to the downloaded Google Service Account key.
10. After entering the necessary details, save the changes made to the io.config.yaml file.
11. Finally, you can **run** the pipeline to **test the connectivity** and ensure the successful transfer of data to BigQuery.
12. Please make sure to carefully follow these steps and provide accurate information in order to establish the required connectivity with BigQuery.

To save this pipeline, click on the File → Save Pipeline. This will save the current state of your pipeline, including all configurations and settings.

7.8. To Schedule the Pipeline:

Scheduling pipelines in Mage.ai is an effective method to ensure regular execution of pipelines, even when you are not actively working on them.

To schedule a pipeline in Mage.ai, follow these steps:

1. Navigate to the "**Pipelines**" section.
2. Click on the specific pipeline you wish to schedule.
3. Create a new trigger by selecting the "**Create New Trigger**" option.
4. Choose the "**Schedule**" option to create a scheduled trigger.
5. Configure the schedule according to your desired frequency and timing.
6. Save the trigger configuration.
7. By following these steps, you can schedule the selected pipeline to run automatically based on the defined schedule.

8. USER INTERFACE

The below screenshot illustrates the output generated from the pipeline that was created.

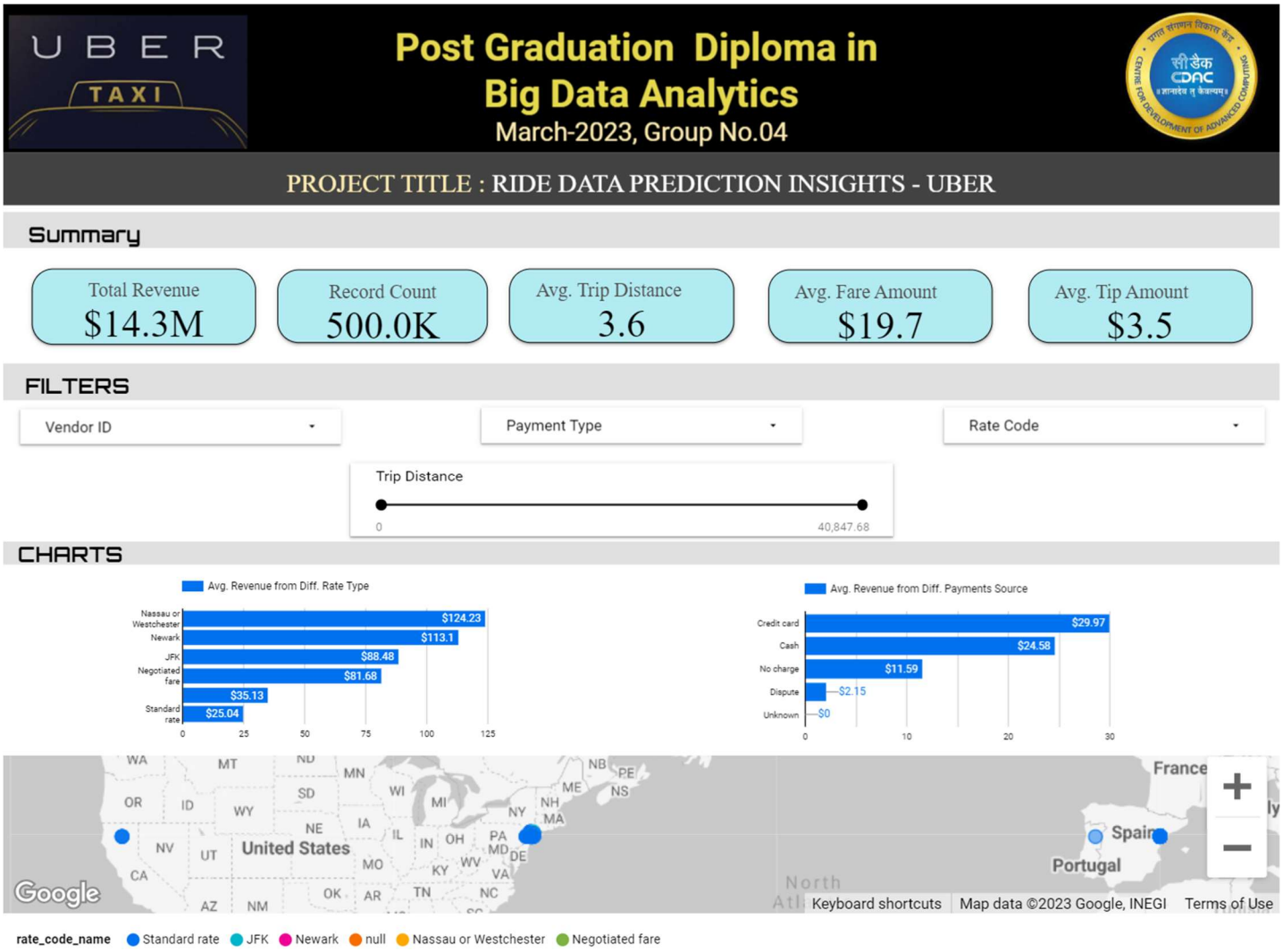


Fig. 8.1 Uber Insights Dashboard

Webpage Link:-

https://dataanalysis10.durable.co/?pt=NjRmNjRiNzU3ZmUwYWJmZmY5NjFjODJkOjE2OTM4OTQ0MzEuNTEyOnByZXZpZXc=

9.RESULT & CONCLUSION

By leveraging the power of data science, this project represents a significant milestone towards transforming Uber into a data-driven, effective, and focused on user's transportation platform. Uber can maintain its position as the market leader in the ridesharing sector and continue to add benefits to both drivers and company by utilizing predictive information to optimize its operations, enhance user experience, and boost safety.

REFERENCES:

1.https://medium.com/@viviennediegoencarnacion/step-by-step-guide-to-get-
2. started-with-google-cloud-platform-for-data-scientists-76e0f5834650
3.https://www.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf
4. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
5. https://youtu.be/3gXsFEC3aYA?si=Zek697JhrKOpbXRv
6. https://youtu.be/OzwSBbuHY-0?si=U5b9ZkjKWT1CWMJy
7. https://youtu.be/ERv2fRGnZTQ?si=OmL9ciGbtfp8VZCv