**Project Report on:**

**Creating a Dashboard visualization interface for Uber Service Data**

**Mentor : Prof. Dadasaheb Hinge**

**Team Member :** Atharva Bharambe, Bhagyashri Yerawar

**Abstract**

The "Creating a Dashboard Visualization Interface for Uber Service Data" project aims to develop an interactive and insightful dashboard for analyzing Uber’s operational data using Azure Databricks. The goal is to provide real-time and historical insights into various key performance metrics, such as ride demand, trip duration, fare trends, and driver performance, by visualizing Uber’s service data. Through the aggregation and transformation of large-scale trip data, the dashboard will offer comprehensive visualizations including heatmaps, time series graphs, and distribution charts to help users understand patterns and make data-driven decisions. The project leverages Azure Databricks for scalable data processing, ensuring efficient handling of large datasets. Real-time data processing capabilities are incorporated to reflect live updates in the visualizations, offering an up-to-the-minute view of Uber’s operations. The dashboard will be designed with user interactivity in mind, allowing users to filter and drill down into data based on geographic regions, timeframes, and other relevant variables. Additionally, role-based access controls will ensure data security and privacy for different user roles. This visualization interface is intended to improve business insights, operational efficiencies, and strategic decision-making for Uber’s service management and growth.

1. **Introduction :**

This project focuses on implementing a data ingestion pipeline to process and analyse Uber ride data using Databricks and Azure Data Lake Storage Gen2 (ADLS Gen2). The pipeline ingests CSV data, performs transformations to calculate key metrics, and organizes the data into distinct layers: staging, raw, silver, and gold.

In the staging layer, data is initially loaded into Delta tables. This raw data is then processed to add timestamps in the raw layer, ensuring accurate versioning. Moving to the silver layer, the data is refined further, adding computed columns such as trip\_date, trip\_hour, and distance. In the gold layer, aggregate metrics like daily total trips, average fare, and total fare are calculated, preparing the data for business insights.

This architecture offers a structured, scalable way to manage data, leveraging Delta Lake's capabilities for consistency and Databricks for efficient processing. The result is a well-organized dataset that enables advanced data analytics and facilitates the creation of dashboards for business intelligence. This project demonstrates the effective use of cloud-based big data tools to solve real-world data challenges.

1. **Objectives**

* **Primary Objectives**

The main goal of this project is to develop a dynamic and interactive dashboard that provides valuable insights into Uber service data. The dashboard will aim to showcase trends, patterns, and key metrics related to Uber's performance, helping users understand the business and make informed decisions.

* **Secondary Objectives**

In addition to the primary objectives, the dashboard will feature several secondary functionalities that enhance its usefulness and flexibility for a broader range of users.

* **Technical Objectives**

The technical implementation of the dashboard will involve several considerations to ensure scalability, security, and efficient data handling.

1. **Layers Used :**
2. **Staging Layer:**

The staging layer serves as the initial point of data ingestion. Raw data, often from CSV or other external sources, is loaded into this layer without any transformations. This layer ensures that the raw data is safely stored in a Delta table, providing a reliable starting point for further processing. The main goal here is to retain the original data in its entirety and ensure it's available for subsequent transformations.

1. **Raw Layer:**

After the data is ingested into the staging layer, the raw layer involves minimal transformations. Here, basic operations such as adding metadata like timestamps (e.g., created\_timestamp) are performed. This layer serves as a backup for the original ingested data, and it’s often used for tracking historical data changes. The raw data is stored with versioning, providing an auditable and consistent record of data changes.

1. **Silver Layer:**

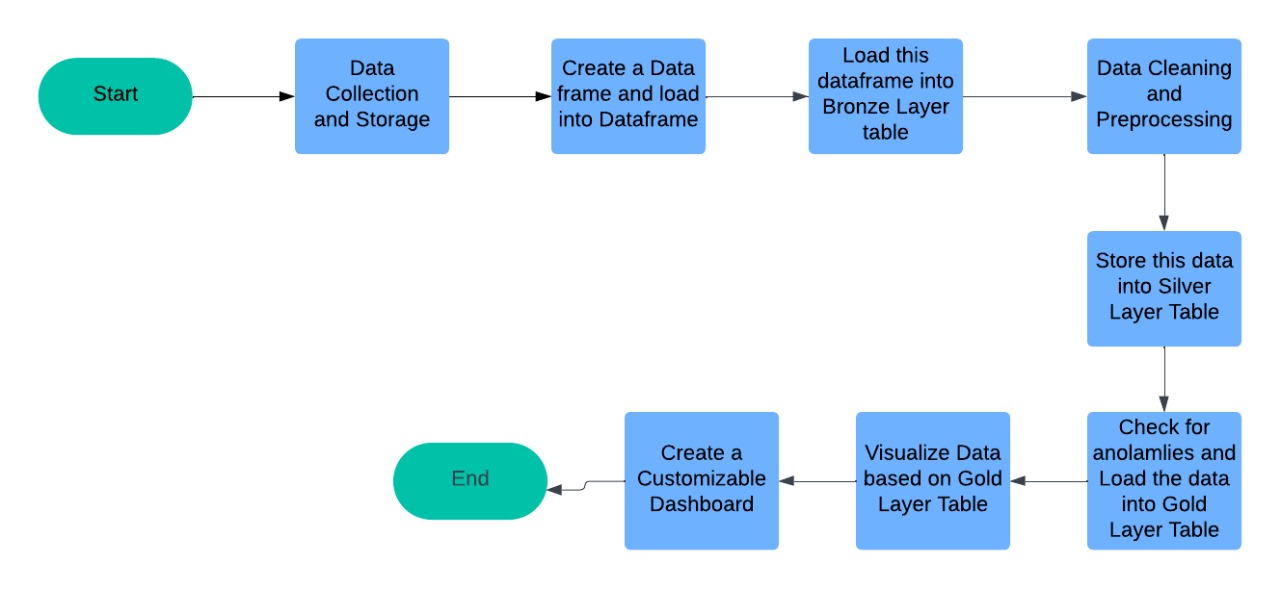
The silver layer is where data transformation begins. In this layer, more advanced transformations are applied to the data. For example, new columns are added (such as trip\_date, trip\_hour, and distance), which are essential for deeper analysis. The goal of the silver layer is to clean and enrich the data by adding meaningful derived attributes, making it ready for analytical and reporting purposes.

1. **Gold Layer:**

The gold layer contains the final, aggregated, and refined dataset, which is ready for business analytics and decision-making. In this layer, data is aggregated, with calculations like total trips, average fare, total fare, and other business metrics performed. The gold layer is typically where data scientists and business analysts would query the data to generate reports or dashboards. It represents the "trusted" version of the data that can be used for insights and visualizations.

Each of these layers builds upon the previous one, adding more structure, cleaning, and transformation to prepare the data for reporting, analysis, and decision-making. This approach ensures that the data pipeline is scalable, efficient, and easy to maintain while supporting advanced analytics.

1. **Methodology :**



1. **Description :**
2. **Data Collection and Storage** :

This step involves collecting raw data and storing it in a centralized location, such as Azure Data Lake Storage (ADLS) Gen2. The data source could be a CSV file or any other structured dataset that will be processed in subsequent steps.

1. **Create a Dataframe and Load into Dataframe** :

After storing the data, it is read into a DataFrame in Apache Spark. This DataFrame format is suitable for applying transformations and analyses within Spark and Databricks.

1. **Load this DataFrame into Bronze Layer Table** :

The raw data is loaded into the "Bronze" layer table. The Bronze layer serves as the initial storage layer, preserving the data in its raw form without much processing, and acts as a source of truth for the original dataset.

1. **Data Cleaning and Preprocessing** :

In this step, data cleaning and preprocessing are conducted to remove inconsistencies, handle missing values, and prepare the data for analysis. This ensures that the data is ready for further transformations and aggregations.

1. **Store this Data into Silver Layer Table** :

The cleaned and preprocessed data is stored in the "Silver" layer, which represents a refined version of the data. The Silver layer includes cleansed and standardized data that is ready for analytical operations.

1. **Check for Anomalies and Load the Data into Gold Layer Table** :

After verification, the data is transformed into a more structured format in the "Gold" layer, where it is enriched with business-specific logic, aggregations, or derived metrics. This layer is used for generating insights and metrics.

1. **Visualize Data Based on Gold Layer Table** : Visualizations and metrics are created based on data in the Gold layer. This step involves using various analytical tools to gain insights from the data, enabling better decision-making.
2. **Create a Customizable Dashboard** :

Finally, the processed data is used to build a dashboard for users to view trends, patterns, and key metrics. This dashboard enables stakeholders to interact with data and monitor performance or other critical indicators in real-time.

1. **Tools**
2. **Databricks** :

An integrated environment for Apache Spark and Delta Lake, used for data engineering and collaborative notebook-based development.

1. **Delta Lake** :

Provides reliable data storage with ACID transactions and schema enforcement, used for data versioning and consistency.

1. **Apache Spark (PySpark)** :

A distributed data processing framework used to handle transformations, aggregations, and SQL-like operations.

1. **Azure Data Lake Storage Gen2 (ADLS Gen2)** :

A cloud-based storage solution for big data, used for scalable and secure storage of data.

1. **Technologies**
2. **Python** :

The main programming language used to develop the project code, including PySpark scripts.

1. **SQL** :

Used within Databricks notebooks to query and manipulate data, especially for aggregation and metrics calculations.

1. **PySpark Functions (pyspark.sql.functions)** :

Provides various functions for data manipulation and transformation, such as creating new columns and calculating derived metrics (e.g., trip distance).

1. **Conclusion :**

In conclusion, this data processing project effectively demonstrates a structured, multi-layered approach to managing and analyzing large datasets using Databricks, Apache Spark, and Delta Lake on Azure Data Lake Storage Gen2. The project begins by ingesting raw data, which is then cleaned, refined, and stored across Bronze, Silver, and Gold layers, each with increasing levels of data transformation and quality. This layer-based approach ensures data reliability, supports efficient querying, and facilitates deeper analysis.

The final output includes a customizable dashboard built from the Gold layer, which provides users with insights into key metrics and trends, aiding in informed decision-making. By utilizing this architecture, the project achieves scalability, data consistency, and an organized data flow that can be applied to various analytical use cases. This pipeline serves as a robust framework for similar data engineering tasks and can be scaled or modified based on different data requirements and business needs.