**18. End-to-End Streaming Data Pipeline with Azure Data Factory and Databricks.   
Data Engineering Project 2.**

-Vikas Reddy Gorantla (Maverick)

[gorantlavikas24@gmail.com](mailto:gorantlavikas24@gmail.com) (+91) 8688603921

**Title: *Set up an end-to-end pipeline with Azure Data Factory for ingesting streaming data and Azure Databricks for real-time processing and analysis of the streaming data.***

**1. Project Overview:**

I built an end-to-end data pipeline using Azure tools to automate the process of handling streaming data. The raw data was first stored in Azure Blob Storage and then transferred to Azure Data Lake Storage using the Copy Data feature in Azure Data Factory. After the data was moved, a Databricks notebook was connected to the pipeline to perform data transformations on the stored data. The transformed data was then saved back to Data Lake Storage for further use. This entire workflow: data ingestion, transformation, and storage was automated within the Azure Data Factory pipeline, ensuring a seamless and efficient process.

**2. Data Overview:**

The e-commerce dataset contains information about 3,660 online purchases, with each purchase uniquely identified by a User\_ID and Product\_ID. It provides details about the products, such as their category (like Sports, Clothing, Toys, and Beauty), their original price, and the discount percentage applied. The final price customers paid after the discount is also recorded. Additionally, the dataset includes information about the payment methods used, such as Net Banking, Credit Card, and UPI, along with the purchase date for each transaction. Importantly, there are no missing values in the dataset, making it reliable for analysis. Overall, this dataset gives a clear picture of customer buying habits, product popularity, and pricing patterns on the e-commerce platform.

**3. Architecture diagram –ER diagram:**

A screenshot of a computer

Description automatically generated

**4. Source Data File:**

The source data for this project was obtained from Kaggle, where the e-commerce dataset is available. The dataset is provided in CSV (Comma Separated Values) format, which is commonly used for structured data storage and analysis.

[Download Ecommerce Dataset](https://www.kaggle.com/datasets/steve1215rogg/e-commerce-dataset?resource=download)

**5. Execution Overview:**

This project utilizes Apache Spark in combination with Azure Data Lake Storage Gen2 to process real-time e-commerce transaction data. The process begins by configuring Spark to connect to the Azure Data Lake Storage, where the raw e-commerce data is stored. The dataset includes key fields such as Product ID, User ID, Category, Price, Discount, Final Price, Payment Method, and Purchase Date.

The data is ingested, with several transformations applied:

* **Total Sales Calculation:** The total sales for each transaction is calculated by applying the discount to the original price.
* **Data Filtering:** Transactions with missing product categories are filtered out to ensure the dataset is clean and complete.
* **Date Formatting:** The Purchase Date is standardized to the yyyy-MM-dd format for consistency in future analysis.

The transformed data is then written back to Azure Data Lake in CSV format, ensuring that new data is appended as it arrives. A checkpointing mechanism is implemented to track the progress of the streaming job, enabling the system to recover from any failures. The job continues to run, processing and updating the data in real-time, making it available for further analysis or reporting.

**6. Azure Resources Used for this Project:**

* Azure Data Lake Storage Gen2
* Azure Databricks
* Azure Storage Account
* Azure Spark Service
* Azure Blob Storage



**7. Project Requirements:**

* **Azure Subscription:** An active Azure subscription to create and manage Azure resources such as Data Lake Storage, Databricks, and other necessary services.
* **Azure Data Lake Storage Gen2:** A provisioned Azure Data Lake Storage Gen2 account for storing raw and processed data.
* **Azure Databricks Workspace:** A Databricks workspace to set up and execute Spark-based transformations on the streaming data.
* **Apache Spark:** A Spark environment within Databricks for processing large-scale data in real-time.
* **Data Source:** A CSV-formatted e-commerce dataset containing transaction details like User\_ID, Product\_ID, Category, Price, Discount, Payment\_Method, and Purchase\_Date.
* **Python & PySpark Libraries:** Python environment with PySpark libraries installed for accessing and processing data from Azure Data Lake Storage and Databricks.
* **Azure Storage Accounts:** Storage accounts for raw data input and processed output, with appropriate keys and permissions to access the data.
* **Checkpointing:** Checkpoint locations for streaming jobs to ensure recovery and fault tolerance during data processing.

**8. Tasks Performed:**

* **Data Ingestion from Azure Data Lake**

1. Configured the Spark environment to access Azure Data Lake Storage Gen2 using the storage account key.
2. Set up a streaming read process to ingest the e-commerce transaction data stored in CSV format.

* **Data Transformation**

1. Defined the schema for the e-commerce dataset to properly interpret the columns during ingestion.
2. Applied transformations to calculate total sales by considering the final price and discount.
3. Filtered data to exclude rows with missing or null values in specific columns (e.g., Product Category).
4. Reformatted the Purchase\_Date field from the original format into yyyy-MM-dd format for consistency.

* **Real-Time Streaming Processing**

1. Set up a real-time streaming process using Spark Structured Streaming to continuously process and transform incoming data from the Data Lake.
2. Added custom transformations like calculating total sales and formatting the date to create a clean, processed dataset.

* **Data Storage in Azure Data Lake**

1. Wrote the processed data to a separate container in Azure Data Lake Storage Gen2 in CSV format, ensuring that the data is accessible for future analysis or reporting.
2. Implemented checkpointing to ensure that the stream can resume processing in case of failures or restarts.

**9. Analysis Result:**

A screenshot of a computer

Description automatically generated

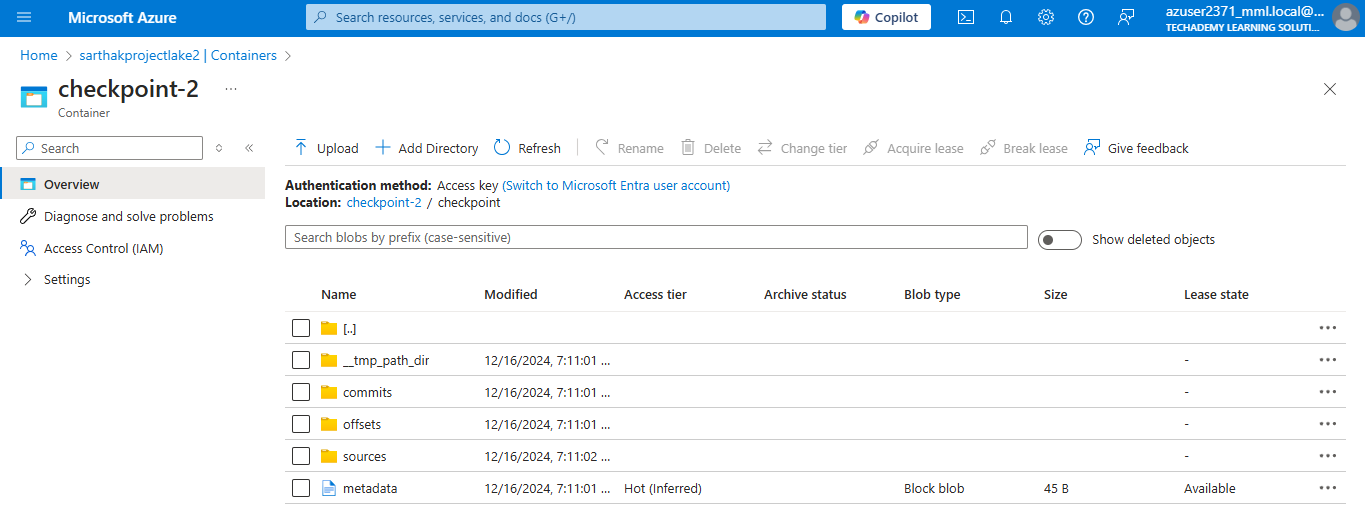
A computer screen with a blue background

Description automatically generated with medium confidence

*Note: - In the picture above, since the code is set up for streaming live data, the status of the notebook will remain as "In Progress" for up to 10 minutes, as the timeout for the streaming job is configured to 10 minutes.*

A screenshot of a computer

Description automatically generated



*Note: - The image above shows the checkpoint directory created for the streaming job. In this directory, Spark stores the metadata and progress information of the streaming job, ensuring that the job can resume from the last processed state in case of a failure or restart.*

**10. Project Code and Output:**

The output CSV file showcases the successful implementation of a real-time streaming pipeline, where data ingestion, transformation, and storage were seamlessly achieved using Azure Data Factory (ADF) and Azure Databricks (ADB). Initially, the dataset was uploaded to Azure Blob Storage, simulating streaming data. Using ADF, a pipeline was created with activities to copy and process the data in chunks, triggering a Databricks Notebook for advanced transformations. The Databricks Structured Streaming logic calculated the Final\_Price (Rs.) by applying discounts to the original Price (Rs.), standardized purchase dates into Formatted\_Purchase\_Date, and aggregated the Total\_Sales for each record. The pipeline ensured smooth orchestration between data ingestion and processing, while the output file reflects key insights such as product categories, payment methods, and sales metrics. This end-to-end process validates the pipeline's ability to process and transform large-scale datasets efficiently, delivering clean and structured data ready for analysis and reporting.



Link to whole project: <https://drive.google.com/drive/folders/1bM5PvRzd_-SXzDxJCb0HQbMP68IOKXCv?usp=drive_link>