Optimizing Aviation Insights: A Data Engineering Approach for Analyzing Flight Delays and Cancellations
A Data Engineering Case Study

Stream : **Spark Azure track**

Curated by : **Anandh Kumar M**

Technology :

Hadoop,Hive,PySpark,Spark Streaming Azure Data factory ,ADLS

Problem Statement:

The problem at hand revolves around the assessment and analysis of the on-time performance of domestic flights operated by major air carriers within the United States. The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) systematically tracks and documents crucial information pertaining to these flights. This comprehensive dataset specifically focuses on the events and occurrences during the year 2015, providing a detailed account of flight delays, cancellations, and diversions.

The DOT releases summary information derived from this dataset in its monthly Air Travel Consumer Report, offering a consolidated view of the aviation landscape's operational efficiency and challenges. The dataset itself serves as a valuable resource for understanding the dynamics of domestic air travel, allowing for a nuanced exploration of the factors influencing flight punctuality and potential disruptions.

Key metrics captured in the dataset include the number of flights that adhered to their scheduled times, those that experienced delays, flights that were cancelled, and instances where flights were diverted from their original routes. Each of these metrics contributes to a holistic understanding of the operational challenges faced by large air carriers and serves as a foundation for identifying patterns, trends, and potential areas for improvement in the broader context of domestic air travel.

What is expected :

The expected outcome of the data engineering efforts for the analysis of 2015 flight delays and cancellations using PySpark, Azure Data Lake Storage (ADLS), and Azure Data Factory (ADF) is a seamlessly integrated, scalable, and efficient data processing pipeline. The PySpark-based data ingestion process should successfully retrieve the historical flight dataset from the U.S. Department of Transportation's Bureau of Transportation Statistics, while Spark Streaming continuously fetches and processes real-time flight data. The integrated dataset should undergo transformations for standardization, outlier handling, and feature derivation, resulting in a unified and comprehensive dataset. Robust data quality

checks within PySpark ensure a clean and reliable dataset with handled missing data, maintaining the integrity of the subsequent analysis and predictions. The storage strategy in ADLS should be optimized, incorporating considerations such as partitioning, file formats, and compression for enhanced performance. The entire process should be orchestrated using Azure Data Factory, facilitating scheduled batch processing for historical data and Spark Streaming for real-time updates. The outcome is a well-documented, collaborative data engineering solution that seamlessly integrates with the data science workflow, ultimately empowering stakeholders with actionable insights into flight trends and contributing to informed decision-making in the aviation industry.

Datasets:

Batch Processing

Airlines :
https://github.com/akgeoinsys/ltimindtree2023/blob/main/casestudies/datasets/airlines.csv

Airports:

https://github.com/akgeoinsys/ltimindtree2023/blob/main/casestudies/datasets/airports.csv

Streaming datasets :

Flights: (historical)
https://github.com/akgeoinsys/ltimindtree2023/blob/main/casestudies/datasets/flights_sample.csv

Flights: (latest)
https://github.com/akgeoinsys/ltimindtree2023/blob/main/casestudies/datasets/flights_latest.csv

Key Objectives for Flight Cancellation Dataset Analysis:

Data Ingestion and Streaming:

Develop a PySpark-based data ingestion process to collect detailed information on flight cancellations from the U.S. Department of Transportation's Bureau of Transportation Statistics.
Implement real-time Spark Streaming to continuously fetch and process the latest data on flight cancellations, enabling immediate insights and decision-making.

Data Integration and Transformation:

Conduct data integration by merging historical records of flight cancellations with real-time streaming data to create a unified and comprehensive dataset.
Execute essential data transformations to standardize schemas, handle outliers, and derive features conducive to predictive modeling for identifying patterns in flight cancellations.

Data Quality and Imputation:

Implement robust data quality checks within the PySpark framework to identify and rectify anomalies or inconsistencies in the combined historical and real-time flight cancellation datasets.
Develop advanced strategies for handling missing data, ensuring that imputation techniques are applied judiciously to maintain the integrity of the analysis and predictions.

Efficient Data Storage and Management:

Design an efficient data storage strategy utilizing Azure Data Lake Storage (ADLS) to store both historical and real-time flight cancellation datasets.
Optimize storage considerations, including partitioning, choice of file formats (e.g., Parquet), and compression techniques, to enhance performance and cost-effectiveness.

Scalability and Performance:

Ensure the scalability of PySpark data engineering processes to effectively handle large volumes of historical and real-time flight cancellation data.
Fine-tune Spark configurations and implement effective partitioning strategies to optimize processing performance for timely analysis.

Comprehensive Workflow Orchestration:

Develop a comprehensive workflow orchestration system using Azure Data Factory to seamlessly coordinate and automate the execution of data engineering tasks.
Schedule both batch processing for historical flight cancellation data and Spark Streaming for real-time updates, providing a holistic view of cancellation trends.

Case Study Execution Plan:

Team Structure: A group of 4 or 5 members will execute the case study.
Task Assignment: Each member will have specific tasks aligned with project objectives.
Concurrent Work: Team members will work concurrently, ensuring parallel progress.
Integration: Individual contributions will integrate during the final project stage.
Final Presentation: Completed case study will be presented to SMEs and Mentors.

Data Engineering Approach in the Retail Sales Analysis Case Study with PySpark:

Tasks:

Data Ingestion with Azure Data Factory:

Utilize Azure Data Factory to ingest the 2015 flight delays and cancellations dataset from its source into Azure Data Lake Storage (ADLS).
Schedule periodic data ingestion to ensure the dataset remains up-to-date for ongoing analysis.

Data Transformation with PySpark:

Develop PySpark scripts to read the raw data from ADLS, transform it, and create structured DataFrames.
Handle missing values, standardize formats, and create derived features that can enhance analysis (e.g., calculate delays in minutes).

Exploratory Data Analysis (EDA):

Utilize PySpark for exploratory data analysis to identify patterns, trends, and statistical summaries.
Generate visualizations using PySpark to illustrate key insights, such as the distribution of delays, cancellation rates, and busiest airports.

Advanced Analytics with PySpark:

Apply PySpark machine learning capabilities to build predictive models for flight delays and cancellations.
Evaluate the performance of the models and identify factors contributing to delays or cancellations.

Data Aggregation and Summary Statistics:

Leverage PySpark for aggregating data to calculate summary statistics, such as the total number of delayed or canceled flights, average delay times, and monthly trends.

Integration with External Data Sources:

Integrate external data sources, such as weather or holiday calendars, using Azure Data Factory, to enrich the analysis and identify potential external factors affecting flight delays.

Data Visualization and Reporting:

Use PySpark and Azure Data Factory to create visualizations or reports summarizing key findings.
Schedule the generation of periodic reports to keep stakeholders informed about ongoing trends.

Optimization and Scalability:

Optimize PySpark scripts and Azure Data Factory pipelines for scalability, ensuring efficient processing of large volumes of flight data.
Consider partitioning strategies and cluster configurations to improve performance.

Enhanced Data Engineering Approach with Additional Tasks(Optional):

Data Ingestion and Cleansing:

PowerBI Report Creation:

Participants can create simple PowerBI reports showcasing key metrics during data ingestion.
Visualize the data quality metrics, such as the count of missing values or duplicate records.

Machine Learning for Data Quality:

Implement a basic machine learning model using PySpark to identify and handle missing values more intelligently.
Leverage PySpark's MLlib for this task and integrate it into the data ingestion pipeline.