**Unlocking Insights**

**A Deep Dive into Comprehensive Retail Sales Analysis**

**A Data Engineering Case Study**

**Stream : Spark Azure track**

**Curated by : Anandh Kumar M**

**Technology :**

**Hadoop,Hive,PySpark,Azure Data engg (ADLS/ADF/ADB),Git**

**Project Overview:**

**The Geocart e-commerce ordering and delivery application aims to enhance the shopping experience for its users by leveraging transactional data for insightful analysis. The project focuses on designing and implementing a robust data pipeline to process and analyze a large relational dataset of customer orders. The dataset, containing anonymized information from over a million orders, serves as the foundation for predicting user behavior, including repeat purchases, new product trials, and items added to the cart during a session.**

**What is Expected?**

In the initial step of the exploratory analysis, I have generated a document that outlines the outputs of the preliminary data exploration. This document highlights key distributions, and patterns in the data. It also includes a comprehensive list of potential issues and data anomalies that require further investigation and follow-up. These issues range from missing values and outliers to inconsistencies in data entries. The descriptive analysis section of the document emphasises important outcomes and findings from the data, such as trends in campaign performance metrics, user engagement patterns,and conversion rates.

Moving forward, the next level of analysis will involve a detailed examination of classifying successful and unsuccessful campaigns. I plan to employ various analytical techniques, to identify factors that contribute to campaign success. This analysis will encompass different methods and reports. Through these methods, I aim to uncover insights into the characteristics of successful campaigns and the underlying patterns that differentiate them from unsuccessful ones. The inferences drawn from this

comprehensive analysis will provide actionable insights and guide strategic decisions for optimizing future campaign strategies.

**Efficient Data Engineering Solution:**

Design and implement a scalable data pipeline for processing and analyzing customer order data.

Support real-time model predictions to optimize the user's shopping journey.

**Exploratory Analysis:**

Conduct exploratory analysis to understand key distributions, patterns, and potential issues in the dataset.

Document findings related to missing values, outliers, and inconsistencies.

**Campaign Performance Analysis:**

**Analyze campaign performance metrics, user engagement patterns, and conversion rates.**

**Identify trends that can guide strategic decisions for optimizing future campaign strategies.**

**Expected Deliverables:**

**Preliminary Data Exploration Document:**

**Outlines key distributions, patterns, and potential issues in the dataset.**

**Highlights trends in campaign performance metrics, user engagement patterns, and conversion rates.**

**Detailed Campaign Analysis:**

**Classify successful and unsuccessful campaigns using various analytical techniques.**

**Uncover factors contributing to campaign success and differentiate them from unsuccessful ones.**

**Provide actionable insights for optimizing future campaign strategies.**

**Datasets :**

[**https://github.com/akgeoinsys/retail/blob/main/Retail-20210521T141851Z-001.zip**](https://github.com/akgeoinsys/retail/blob/main/Retail-20210521T141851Z-001.zip)

**Specific Analysis Tasks:**

**Monthly Pricing Summary:**

**Generate a report for line items shipped within the last 60-120 days.**

**Group data by RETURNFLAG and LINESTATUS, calculating totals and averages for various metrics.**

**Quarterly Pricing Analysis:**

**Perform quarterly analysis on pricing data for line items within the specified date range.**

**Group data by RETURNFLAG and LINESTATUS, displaying results in ascending order.**

**Yearly Price Trends:**

**Analyze yearly pricing trends for line items shipped within the specified date range.**

**Product Return Analysis:**

**Analyze pricing information for returned products within the 60–120-day window.**

**Discount Effectiveness Report:**

**Investigate the effectiveness of discounts on online items shipped within the specified timeframe.**

**Group data by RETURNFLAG and LINESTATUS, computing aggregates and presenting results in ascending order.**

**Line-Item Performance Comparison:**

**Compare the performance of line items shipped within the given date range. Group data by RETURNFLAG and LINESTATUS, calculating aggregates and displaying results sorted by attributes.**

**Strategic Implementation Blueprint: A Roadmap for the Execution and Technicalities of the Retail Sales Analysis Data Engineering Project**

**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 Presentatio**n: Completed case study will be presented to SMEs and Mentors.

**Technicalities:**

**Data Pipeline Pattern**:

Adherence to a standard data pipeline pattern for systematic and efficient data processing and transformation.

**Key Stages:**

**Data Ingestion:** Bringing data into the system.

**Data Processing**: Transforming and cleansing the data.

**Data Storage**: Organizing and managing data efficiently.

**Data Visualization and Reporting**: Preparing data for visualization and reporting.

Data Layers:

**Parent Folders:**

Each team has a dedicated parent folder for data processing.

Ensures data isolation and promotes collaboration within the team.

**RAW Sub-folder:**

Stores raw and unprocessed data from various sources.

Includes data ingested through Azure Data Factory or other mechanisms.

**STG (Staging) Sub-folder:**

An intermediate storage location.

Transforms and prepares data from RAW for further processing.

Ensures data quality and consistency before moving to the CURATED sub-folder.

**CURATED Sub-folder:**

Holds processed and curated data ready for visualization and analysis.

Transformed, cleansed, and enriched to meet specific business requirements.

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

**Data Ingestion and Cleansing:**

**Data Engineers' Role:**

Utilize PySpark for data ingestion from Hive, leveraging its distributed processing capabilities.

Create a reusable and secured connection for seamless data extraction from Hive.

Specific Tasks Include:

Leverage PySpark to handle the extraction process efficiently.

Perform initial data cleansing within PySpark for improved data quality.

ETL and Analysis using PySpark:

**Data Engineers' Responsibilities:**

Utilize PySpark extensively for data transformation, taking advantage of its powerful processing capabilities.

Filter out irrelevant or incomplete data within the PySpark framework.

Aggregate data using PySpark functions for calculating summary statistics.

Implement extensive transformations using PySpark to enrich data types and create derived columns.

Leverage PySpark's join operations based on common keys for comprehensive data consolidation.

Apply PySpark-based data partitioning for enhanced query performance.

**Additional Actions with PySpark:**

Leverage PySpark for data deduplication, ensuring enhanced data quality.

Implement validation checks within the PySpark workflow to maintain adherence to business rules.

**Data Storage:**

Azure ADLS (Azure Data Lake Storage):

Leverage PySpark to write transformed data directly into Azure ADLS.

Utilize PySpark's capabilities for efficient and optimized data storage in ADLS.

Benefits of PySpark Integration:

**Distributed Processing:**

Leverage PySpark's distributed processing capabilities for efficient data handling.

Optimize data processing tasks using PySpark's parallelized operations.

**Unified Approach:**

Achieve a unified data engineering approach with extensive use of PySpark for ingestion, transformation, and storage.

Enhance code reusability and maintainability through PySpark's versatile functions.

**Optimized Data Storage:**

Directly write transformed data into Azure ADLS using PySpark for optimized storage.

Leverage PySpark's optimized write operations for enhanced performance.

**Cloud ETL pipelinesAzure Data Factory (ADF) Task After Writing into Azure Data Lake Storage (ADLS):**

**Task: Data Orchestration and Transformation**

**Objective:**

Orchestrate a data transformation pipeline in Azure Data Factory after writing data into Azure Data Lake Storage.

Steps:

**Trigger:**

Set up a trigger to initiate the pipeline based on a schedule or event.

**Data Movement:**

Add activities to move data from ADLS to Azure Data Warehouse/Database (SQL DW/DB) and Azure Databricks.

Utilize ADF connectors for seamless data movement.

**Transformation:**

Implement transformations using Azure Databricks or Azure Data Flow in ADF.

Leverage PySpark for extensive data transformations.

Apply additional business logic, aggregations, or calculations as needed.:

**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.