**Enhancing Loan Management Efficiency for the International Bank for Reconstruction and Development (IBRD)**

**A Data Engineering Case Study**

**Stream : Spark Azure track**

**Curated by : Anandh Kumar M**

**Technology :**

**Hadoop,Hive,PySpark,Spark Streaming Azure SQL ,Git**

**Problem Statement:**

The International Bank for Reconstruction and Development (IBRD), in collaboration with the World Bank, plays a pivotal role in providing loans to middle and low-income countries for diverse projects. Despite a history of successful loan disbursements and repayments, the occurrence of loan cancellations poses a significant challenge. These cancellations, loans withdrawn before disbursement, result in substantial time and financial costs. The objective of this project is to analyze the patterns of cancelled loans and develop a predictive model to prevent or proactively address such instances.

**Project Overview:**

As a Big Data Engineer, the focus is on managing and processing a large amount of data to support the data science team's analysis and modeling efforts. The project involves multiple tasks spanning data integration, cleansing, transformation, optimization, integration with Azure services, and incorporating real-time data using Spark Streaming.

**Datasets:**

**Batch Processing 👍**

<https://finances.worldbank.org/Loans-and-Credits/IBRD-Statement-Of-Loans-Historical-Data/zucq-nrc3>

**Spark Streaming Datasets: Latest**

<https://github.com/akgeoinsys/finance-ibrd>

**Key Objectives:**

Case Study: Enhancing Loan Management Efficiency for the International Bank for Reconstruction and Development (IBRD)

Problem Statement:

The International Bank for Reconstruction and Development (IBRD), in collaboration with the World Bank, plays a pivotal role in providing loans to middle and low-income countries for diverse projects. Despite a history of successful loan disbursements and repayments, the occurrence of loan cancellations poses a significant challenge. These cancellations, loans withdrawn before disbursement, result in substantial time and financial costs. The objective of this project is to analyze the patterns of cancelled loans and develop a predictive model to prevent or proactively address such instances.

Project Overview:

As a Big Data Engineer, the focus is on managing and processing a large amount of data to support the data science team's analysis and modeling efforts. The project involves multiple tasks spanning data integration, cleansing, transformation, optimization, integration with Azure services, and incorporating real-time data using Spark Streaming.

**Business Questions:**

**Loan Processing Time:**

Determine the number of days taken for loan signing and repayment.

**Top Three Loan Recipients:**

Identify the top three countries with the highest loan amounts and observe any sudden drops.

**Countries with Loan Cancellations:**

Identify countries that experienced at least one loan cancellation, assuming they had borrowed a sufficient number of loans.

**Average Repayment Period:**

Calculate the average repayment period for each country.

**Tasks:**

**Database Creation and Data Loading:**

Build an Azure SQL database and load data from disparate sources using Sqoop or Spark DataFrames.

**Cleansing and Scrubbing:**

Remove bad records and replace missing data by reading data from JDBC or file connections.

**Spark DataFrame Creation and ETL:**

Create Spark DataFrames and perform Extract, Transform, Load (ETL) operations to address analytical requirements.

**Data Transformation:**

Perform transformations such as joins, sorting, aggregation, adding new columns, dropping unnecessary columns, and handling duplicates.

**Optimization Techniques:**

Apply optimization techniques like partitioning, bucketing, and compression to enhance data processing efficiency.

**Integration with Azure Services:**

Integrate with Azure services, using Azure Data Lake Storage or Blob Storage to build external tables for efficient data storage.

**Real-time Data Incorporation with Spark Streaming:**

Develop a Spark Streaming application to ingest and process the latest loan data in real-time.

**Storage Performance Evaluation:**

Experiment with different storage options in Azure to evaluate performance and processing efficiency.

**Visualization for Business Queries:**

Create visualizations for addressing business questions related to the retail industry in different perspectives.

**Strategic Implementation Blueprint: A Roadmap for the Execution and Technicalities of the Loan Management Efficiency 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 visualisation 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.