***High-Level Document Design for MovieLens Dataset Analysis Project***

**1. Introduction**

This document outlines the high-level design for the MovieLens dataset analysis project, which aims to answer analytical questions on a semi-structured dataset containing a million records. The project will be implemented using Spark and Scala, leveraging various Spark APIs such as RDD, Spark SQL, and Spark DataFrames. The goal is to draw useful insights about users and movies by performing data analysis and utilizing different forms of Spark APIs.

**2. System Architecture**

The system architecture for the MovieLens dataset analysis project will consist of the following components:

* **Data Source:**

The MovieLens dataset, which contains a million records, will serve as the data source for analysis.

* **Spark Cluster:**

A cluster of machines will be utilized to distribute the data and computations using Spark's distributed processing capabilities.

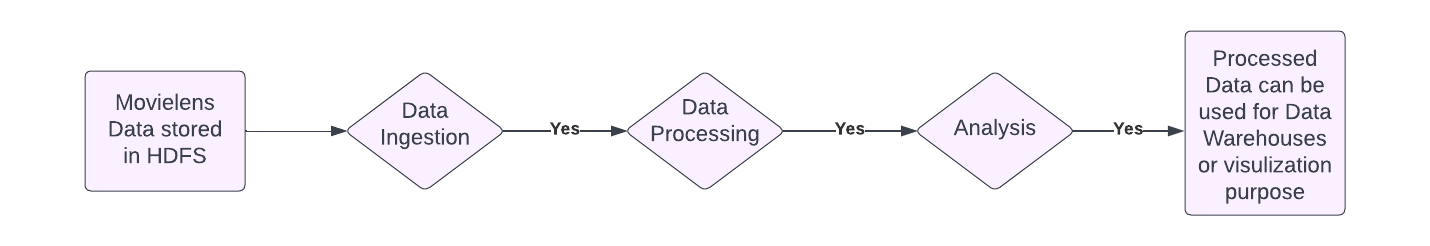
* **PySpark Shell:**

The Spark shell will be used as the interactive development environment for executing Spark operations and running analytical queries.

**3. Data Ingestion and Pre-processing**

The data ingestion and pre-processing tasks will involve the following steps:

* Load the MovieLens dataset into Spark RDDs or DataFrames, using appropriate methods provided by Spark.
* Specify the file format and configuration options for reading the dataset, such as CSV, Parquet, or any other suitable format.
* Perform data validation and pre-processing steps to handle missing values, remove irrelevant columns, and transform the data into a suitable format for analysis.



**4. Analytical Queries using Spark RDD, Spark SQL, and Spark DataFrames**

The project will address several analytical questions using different forms of Spark APIs:

***4.1. Spark RDD***

* Identify the top 10 most viewed movies by counting the number of ratings for each movie.
* Find the distinct list of genres available in the dataset.
* Calculate the number of movies for each genre by grouping the movies based on genre.

***4.2. Spark SQL***

* Create tables for movies.dat, users.dat, and ratings.dat using Spark SQL, enabling easier data querying.
* Retrieve the list of the oldest released movies by sorting them based on their release date.
* Determine the number of movies released each year by grouping the movies based on the release year.
* Calculate the number of movies for each rating by aggregating the ratings data.
* Count the number of users who have rated each movie.
* Compute the total rating for each movie by aggregating the ratings.

***4.3. Spark DataFrames***

* Prepare the Movies data by extracting the year and genre from the text.
* Load the Users data from a double-delimited CSV file.
* Specify a schema programmatically for the Ratings data frame.
* Import data from a URL using Scala and Spark DataFrames.
* Save tables without defining Data Definition Language (DDL) in Hive.

**5. Additional Spark Features**

The project will also utilize the following additional Spark features:

* ***Broadcast Variables:*** Utilize broadcast variables to efficiently distribute reference data, such as lookup tables, to all nodes in the Spark cluster.
* ***Accumulators:*** Use accumulators to track specific metrics or statistics during computations across the Spark cluster.

**6. Technology Stack**

The technology stack for the MovieLens dataset analysis project includes:

* PySpark: Primary framework for processing and analyzing the dataset.
* Spark RDD: Used for low-level data processing and transformations.
* Spark SQL: Utilized for executing SQL-like queries on the dataset.
* Spark DataFrames: Employed for high-level, structured data processing and analysis.
* Hadoop Distributed File System (HDFS): Storage system for the MovieLens dataset.

**7. Resultant Output**

The resultant output from the analysis tasks can be utilized in various ways:

* + Generate reports or visualizations to present the findings and insights derived from the dataset analysis.
  + Integrate the output with other systems or applications for decision-making processes.
  + Utilize the output for further data analysis or machine learning tasks.

**8. Conclusion**

This high-level design document provides an overview of the system architecture, data ingestion and pre-processing, analytical queries using Spark RDD, Spark SQL, and Spark DataFrames, as well as the utilization of additional Spark features such as broadcast variables and accumulators. The design document serves as a roadmap for the implementation phase, guiding the development of the project to efficiently analyze the MovieLens dataset and draw meaningful insights about users and movies using Spark and Scala.