HIGH LEVEL DESIGN

SENTIMENT ANALYSIS

Written By

Rajdeep Sonawane

Contents

Abstract……3

1. Introduction……4

1.1 Why this High-Level Design Document?……4

1.2 Scope……5

2. General Description……5

2.1 Project Perspective……5

2.2 Problem Statement……7

2.3 Proposed Solution……7

3. Tools Used……9

4. Architecture……11

4.1 Scalable pipeline using spark to read customer review from s3 bucket and store it into HDFS……11

4.2 Sentiment analysis Model……11

5. Event Logging……12

6. Error Handling……12

7. Deployment……12

8. Conclusion……13

Abstract

In this project, we propose the design of a scalable pipeline using Apache Spark to ingest customer reviews from an S3 bucket, perform sentiment analysis using Spark Machine Learning, and store the analysed data into HDFS. The pipeline is scheduled to run iteratively after each hour, ensuring timely processing of newly uploaded customer reviews. To facilitate this process, a folder will be created within the S3 bucket to accommodate customer reviews in JSON format.

The pipeline begins with the ingestion of customer reviews from the designated folder in the S3 bucket. Spark's S3 connector will be utilized for efficient parallel processing, enabling scalability to handle large volumes of data. Subsequently, the raw customer review data undergoes preprocessing and transformation as required for sentiment analysis.

Utilizing Spark's Machine Learning library (MLlib), specifically logistic regression, we perform sentiment analysis on the pre-processed customer reviews. This involves training a model to classify customer sentiments into categories such as positive, negative, or neutral.

1. Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2. General Description

2.1 Project Perspective

1 **User-Friendly Interface**:

The pipeline should have a user-friendly interface for configuration, monitoring, and management. This interface should allow users to easily set up the pipeline, schedule its execution, monitor its progress, and access the results.

2 **Customizable Configuration:**

Users should be able to customize various aspects of the pipeline, such as the S3 bucket location for input data, the frequency of pipeline execution, and the HDFS storage path for analyzed data.

3 Automated Scheduling:

The pipeline should support automated scheduling to run iteratively after each hour. Users should have the flexibility to configure the scheduling parameters according to their specific requirements.

4 Robust Error Handling:

The pipeline should include robust error handling mechanisms to handle failures gracefully. It should log errors, retry failed tasks, and alert administrators in case of critical failures.

5 Scalability and Performance:

The pipeline should be designed to scale horizontally to handle large volumes of customer review data efficiently. It should leverage Spark's distributed computing capabilities to achieve high performance and scalability.

6 Real-Time Insights:

Users should have access to real-time insights derived from sentiment analysis of customer reviews. The pipeline should provide timely and actionable insights to help users make informed decisions.

2.2 Problem Statement

Design scalable pipeline using spark to read customer review from s3 bucket and store it into HDFS. Schedule your pipeline to run iteratively after each hour.

Create a folder in the s3 bucket where customer reviews in Json format can be uploaded. The Scheduled big data pipeline will be triggered manually or automatically to read data from The S3 bucket and dump it into HDFS.

Use Spark Machine learning to perform sentiment analysis using customer review stores in HDFS

2.3 Proposed Solution

1 Architecture Overview:

- The solution will utilize a distributed architecture leveraging Apache Spark for data processing and machine learning tasks.
- Customer reviews will be stored in JSON format within a designated folder in an S3 bucket.
- Spark will be deployed on a cluster to handle the data processing tasks efficiently.

2 Data Pipeline:

- The data pipeline will be designed to read customer reviews from the S3 bucket, and store the data into HDFS and perform sentiment analysis using Spark's machine learning capabilities.

- Spark's S3 connector will be used to efficiently read data from the S3 bucket, ensuring parallel processing for scalability.
- The pipeline will consist of multiple stages including data ingestion, transformation, sentiment analysis, and data storage.

3 Sentiment Analysis:

- Sentiment analysis will be performed using Spark's machine learning library (MLlib).
- Logistic Regression will be employed as the machine learning algorithm for sentiment analysis due to its simplicity and effectiveness in binary classification tasks.
- The customer review text will be tokenized, and features will be extracted for input to the logistic regression model.
- The trained model will classify each review into sentiment categories such as positive, negative, or neutral.

4 Iterative Execution and Scheduling:

- The pipeline will be scheduled to run iteratively after each hour to process new customer reviews uploaded to the S3 bucket.
- Scheduling can be implemented using tools like Apache Airflow, Apache Oozie, or cron jobs.

3 Tools Used:

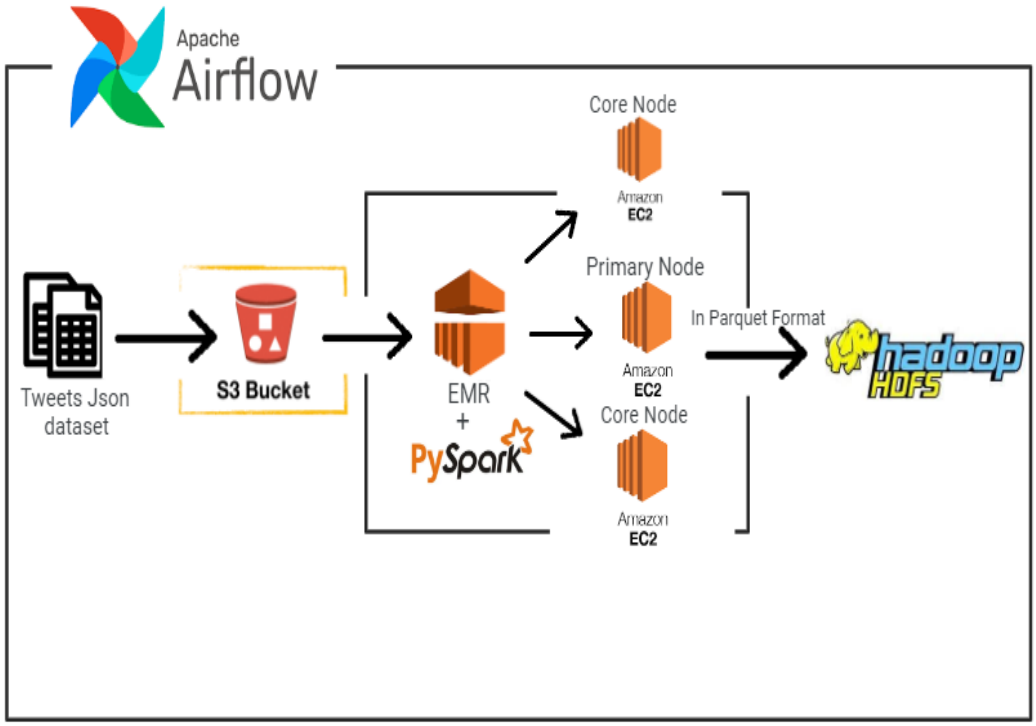
- **Pyspark** is used for Data transformation, Data processing and machine learning.
- **Airflow** is used for Scheduling the job for every one hour.
- **Amazon EMR** is used for creating the Hadoop ecosystem.
- **Amazon S3** is used for storing the customer Review dataset in Json format.
- **Hadoop hdfs** is used for storing the Customer review dataset for analysing purpose.

HIGH LEVEL DESIGN

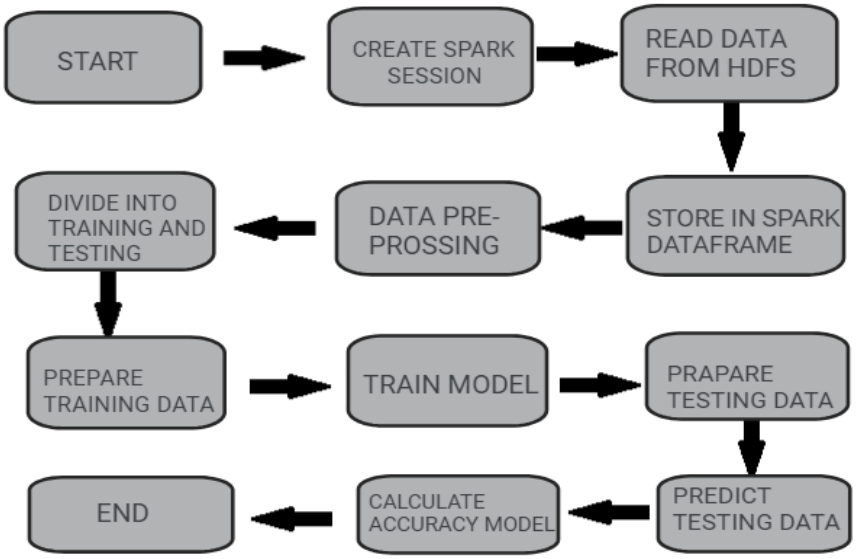
- **Amazon EC2** is used for creating the cluster for Hadoop ecosystem such as primary node and core node.
- **Jupyter Notebook** is used for writing the spark machine learning code.
- **Vscode** is used for creating the environment for spark and airflow.
- **Docker** is used for running the airflow image.

4 Architecture :

4.1 Scalable pipeline using spark to read customer review from s3 bucket and store it into HDFS



4.2 Sentiment analysis Model



5 Event Logging:

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required

2. The System should be able to log each and every system flow.

3. Developer can choose logging method. You can choose database logging/ File logging as well.

4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

6 Error Handling :

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage

7 Deployment :

8 Conclusion :

In conclusion, the proposed project of designing a scalable pipeline using Spark for sentiment analysis on customer reviews stored in an S3 bucket and storing the analyzed data into HDFS offers significant benefits for businesses seeking to gain insights from customer feedback. By implementing this project, organizations can:

- **Efficiently Process and Analyze Customer Feedback**: The scalable pipeline leverages Apache Spark's distributed computing capabilities to efficiently process and analyze large volumes of customer review data.
- **Derive Actionable Insights**: Through sentiment analysis, businesses can gain valuable insights into customer sentiments, allowing them to identify trends, patterns, and areas for improvement.
- **Enhance Decision-Making**: Armed with insights from sentiment analysis, organizations can make data-driven decisions to improve products, services, and overall customer experience.
- **Enable Real-Time Analysis**: The iterative scheduling of the pipeline ensures that customer feedback is analyzed in near real-time, enabling timely responses to emerging trends or issues.
- **Ensure Scalability and Performance**: The use of Spark and HDFS ensures scalability and high performance, allowing the pipeline to handle growing datasets efficiently.