**FORMAL ETL REPORT**

**ETL Setup Report**

**Introduction:**

The purpose of this project was to use Python, Apache Hive, and Apache Kafka to develop an ETL (Extract, Transform, Load) pipeline. The pipeline feeds data about news stories into HDFS after extracting it from NewsAPI and transforming it with Kafka. The next phase is analyzing data with Apache Hive to extract insights.

**ETL Pipeline Overview:**

**The ETL pipeline consists of the following key steps:**

* **Data Extraction (Kafka Producer):** Here, in this step we developed a Python script to act as a Kafka producer which Utilized the NewsAPI token to fetch news articles based on specified keywords. Afterwards, we formatted the data to ensure compatibility with downstream components and sent the data to a Kafka topic for further processing.

**Kafka Producer Output**

A screenshot of a computer

Description automatically generated

**Kafka Producer Output**

A screenshot of a computer screen

Description automatically generated

* **Data Transformation (Kafka Consumer):** In this step we then implemented a Python script to act as a Kafka consumer and further retrieved data from the Kafka topic which enabled an option to save data locally or directly ingest into HDFS.

**Kafka Consumer Output**

A screen shot of a computer program

Description automatically generated

* **Data Storage and Analysis (Apache Hive):** We then Utilized Apache Hive to create a table for storing news article data. Then, executed insightful aggregations on the data to derive meaningful insights to check the following:

Examples of aggregations include counts per source, average article length, and keyword frequency.

**Hive Insights:**

1. **Count of Articles Published by Day:** This query helps us fetch Counts of the daily articles, grouping by publication day.

A computer screen shot of a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

2. **Articles with Short Descriptions but Long Titles:** We retrieved articles with long titles (>50 words) and short descriptions (<20 words).

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

3. **Articles with the Most Images:** This query fetches titles and image URLs, prioritizing articles with images.

A computer screen with white text

Description automatically generated

A black screen with white text

Description automatically generated

4. **Temporal Analysis of Article Publication (by Hour):** Over here we analyze the articles published per hour, sorted by publication time.

A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

5. **Articles Mentioning Popular Entities:** This query grabs the titles and content mentioning Microsoft, Apple, or Google (limited to 10)

A screen shot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

6. **Top Authors with Average Article Length:** In this we find the top 10 authors with the longest average articles, excluding null content entries.

A computer screen with text and numbers

Description automatically generated

7. **Word Count in Articles:** This counts the occurrence of each word in articles, presenting the top 10 words with the highest counts.

A computer screen shot of a computer

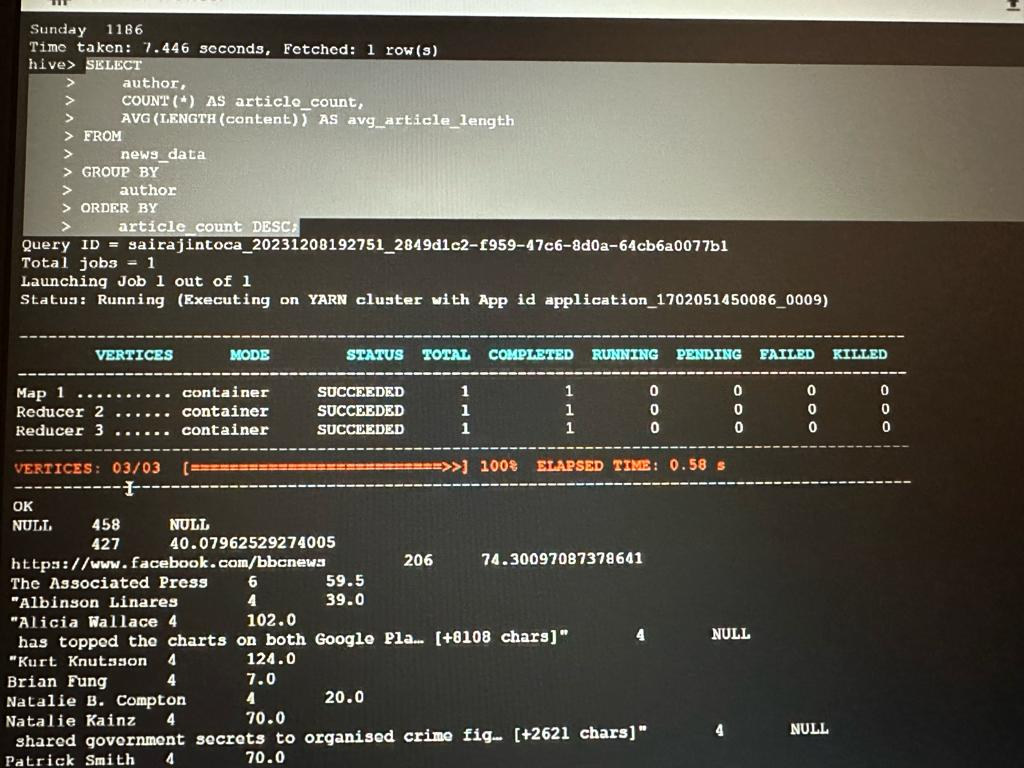
Description automatically generated

8. **Weekday-wise Article Distribution:** This query displays the distribution of articles published on each weekday.

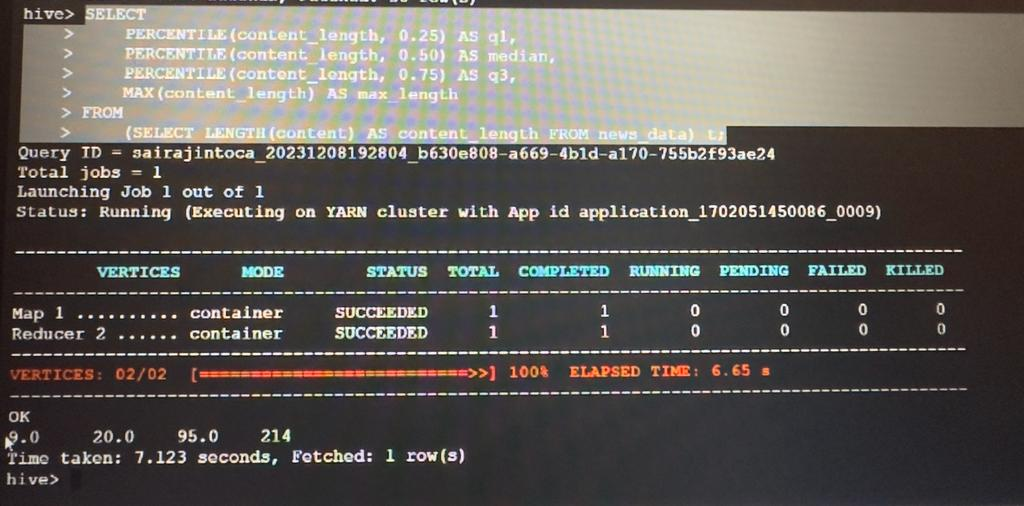
A computer screen with text and numbers

Description automatically generated

9. **Author Contribution Analysis:** In this query we analyzed author contributions by counting articles and calculating average article length, sorted by article count.



10. **Article Length Distribution:** This query calculates the first quartile (q1), median, third quartile (q3), and maximum article length. It uses the PERCENTILE function to find the specified percentiles based on the length of the articles in the dataset.



**Conclusion:**

This ETL pipeline extracts, transforms, and loads news article data in a streamlined manner by effectively integrating several technologies. Apache Hive enables organized storage and perceptive analysis, while Apache Kafka guarantees effective communication between components. The smooth orchestration of the entire pipeline makes it possible to derive important insights from the ingested data.