**Final Project**

**Introduction**

In the realm of Big Data, a significant challenge lies in procuring data suitable for analysis and prediction purposes. This project focuses on creating a streamlined pipeline for sourcing, cleaning, and processing data from the NewsAPI. The ultimate goal is to load this refined data into a Hive table to facilitate thorough analysis and form a web app.

**Step 1:**

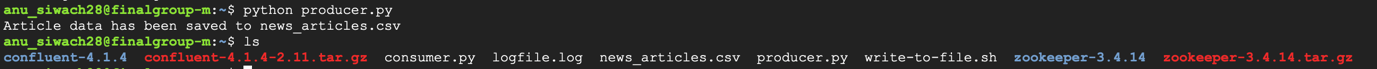
Established a Python Kafka producer that extracted data from news articles using specific keywords relevant to a particular topic. The producer transmitted the data to a topic on a Kafka broker, formatting it in a way that facilitated its consumption by other components in the pipeline.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated



**Step 2:**

Configured a Kafka consumer that retrieved information from the designated Kafka topic and stored the data on HDFS. The collected data was subjected to cleaning and organized into various columns. The approach for data cleansing could be implemented either within the Kafka consumer or producer.

A screenshot of a computer

Description automatically generated

**Step 3:**

In this stage, the collected data is integrated into a Hive table, setting the foundation for subsequent analyses. The procedure involves considering the optimal approach to load data into Hive, ensuring that the process goes beyond a mere transfer of raw articles. Instead, efforts are made to transform the data at the producer or consumer level. The focus lies in structuring the data into relevant columns derived from the news articles. This strategic transformation enables insightful queries to be composed for analysis.

The primary objective of this step is to create a structured Hive table by loading the processed news articles. The table is designed in a way that supports meaningful analysis and yields valuable insights from the articles. This process entails careful consideration of column selection and arrangement, reflecting the essential components of the news articles.

**A screenshot of a computer screen

Description automatically generated**

**Step 4:**

**Hive Queries:**

With the data in place, HiveQL queries were designed to glean valuable insights from the news articles. These queries encompassed various aspects of the data, including sentiment analysis, keyword frequency, and trend identification. For instance, sentiment analysis was performed to gauge the emotional tone of the articles, while keyword frequency analysis helped identify recurring themes.

Running Hive queries for news article analysis showcased the power of this tool in extracting valuable insights from raw data. The ability to structure, transform, and analyze data efficiently was evident throughout the project. The project not only demonstrated the effectiveness of Hive but also highlighted its potential in various other domains of data analysis.

In conclusion, running Hive queries for news article analysis exemplified the importance of data warehousing tools in gaining insights from vast datasets. The project underscored how Hive's capabilities empower data analysts to make informed decisions by unlocking valuable insights from complex data sources.

**Query 1:**

**A screenshot of a computer

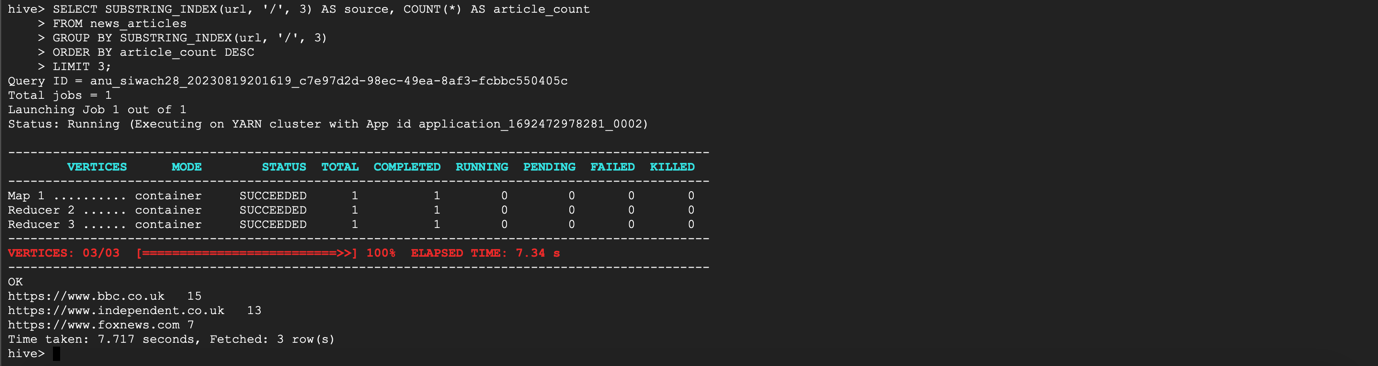
Description automatically generated**

**Query 2:**

**A screenshot of a computer

Description automatically generated**

**Query 3:**

****

**Query 4:**

**A screen shot of a computer

Description automatically generated**

**Query 5:**

**A screenshot of a computer

Description automatically generated**

**Step 5:**

In this section of the report, we will delve into the creation of an interactive web application using Streamlit.io. This web application empowers users to effortlessly retrieve news articles and gain insights into the sentiment of the content, discerning whether the articles convey a positive or negative tone.

For the implementation, we harnessed the capabilities of Streamlit.io, a versatile platform for building and deploying machine learning applications. The user interface was meticulously crafted, offering a seamless and engaging experience for users.

By cultivating this interactive web application, users are empowered to effortlessly explore the sentiment embedded within news articles, ultimately enhancing their comprehension and decision-making processes.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Challenges and Solutions:**

In the process of setting up the project pipeline and developing a web interface, several technical challenges were encountered. This report highlights these challenges and provides insights into the strategies deployed to overcome them.

**Challenge: Choosing Stream lit over Gradio for Web App Development**

The initial decision to utilize Streamlit.io for web application development instead of Gradio introduced a learning curve. Familiarity with Streamlit's syntax and features was necessary to design an effective and user-friendly interface.

**Solution: Overcoming the Streamlit Learning Curve**

To address this challenge, dedicated time was allocated to learning Streamlit's capabilities through their documentation and online resources. Hands-on experimentation allowed the team to gradually understand and implement the desired features within the web application.

**Challenge: Loading Data into Hive Table**

While loading the cleaned and processed data into a Hive table, challenges arose in maintaining the integrity of the data format and ensuring seamless interaction between data sources.

**Solution: Data Transformation and Hive Table Setup**

To address these challenges, data transformation was performed at the Kafka consumer stage to ensure that the data was appropriately cleaned and formatted. Additionally, thorough testing was conducted to ensure that the Hive table schema accurately matched the transformed data.

**Challenge: Managing API Tokens and Keys**

Dealing with API tokens and keys for services like NewsAPI presented challenges in terms of maintaining security and avoiding exposure of sensitive information.

**Solution: Secure Handling of API Tokens**

To mitigate risks, sensitive information such as API tokens were stored in separate configuration files and not hardcoded into the codebase. Utilizing environment variables and proper access control helped safeguard sensitive data.

**Challenge: Efficient Query Design for Insightful Analysis**

Crafting HiveQL queries that provided meaningful insights from the news articles data involved careful consideration of the data structure and its relationship to the desired outcomes.

**Solution: Iterative Query Refinement**

An iterative approach was employed to formulate and refine queries, taking into account the specific insights required from the news articles data. Query testing and debugging allowed for the identification and rectification of any errors or inaccuracies.

In conclusion, while setting up the project and developing the web interface, we encountered challenges related to choosing the right tools, data management, and effective query design. Through proactive learning, careful testing, and iterative refinement, these challenges were successfully addressed, enabling the project to progress smoothly and yield valuable insights from the collected news articles.