TWEETS CLASSIFICATION REPORT

INTRODUCTION

In today's digital age, where vast amounts of textual data are generated daily on social media platforms, sentiment analysis has become increasingly important for understanding public opinion, customer feedback, and market trends. This project aims to develop a sentiment analysis model using PySpark, a powerful tool for distributed data processing, to classify tweets into positive and negative sentiments.

OBJECTIVE

The primary objective of this project is to develop a sentiment analysis model using PySpark to classify tweets into positive and negative sentiments. The project involves several key steps: collecting a comprehensive dataset of tweets with sentiment labels, performing extensive data preprocessing to clean and prepare the text data, and extracting meaningful features that can be utilized for machine learning algorithms. Following this, the project aims to train and fine-tune a sentiment classification model using PySpark's MLlib library, employing suitable classification algorithms such as logistic regression or decision trees. The effectiveness of the trained model will be evaluated using appropriate performance metrics to ensure its accuracy and reliability.

DATA SOURCE

Data source link - https://www.kaggle.com/datasets/kazanova/sentiment140
Data size - 238 MB

For this project, **Sentiment140** dataset is chosen as the primary data source. This dataset contains **1,600,000** tweets, each annotated with sentiment labels indicating whether the sentiment expressed in the tweet is positive or negative. The dataset includes six fields:

target: the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)
ids: The id of the tweet (2087)
date: the date of the tweet (Sat May 16 23:58:44 UTC 2009)
flag: The query (lyx). If there is no query, then this value is NO_QUERY.
user: the user that tweeted (robotickilldozr)
text: the text of the tweet (Lyx is cool)

Reasoning for Choosing this Dataset

With 1.6 million tweets, the dataset offers a rich diversity of text examples, ensuring that the model can generalize well to various linguistic nuances and expressions typical of social media.

On a personal note, I have always been fascinated by the dynamic nature of social media interactions and the challenge of interpreting sentiments in this informal and often noisy data. Tweets, with their brevity and spontaneity, present a unique challenge for natural language processing.

DATA PROCESSING AND TRANSFORMATION PROCESSING

The raw tweets often contain noise such as mentions, hashtags, URLs, and special characters. These elements do not contribute meaningful information for sentiment analysis and can degrade model performance. Therefore, the following preprocessing steps were applied:

- **Removing Unnecessary Columns**: The columns id, date, flag, and user were dropped as they do not contribute to the sentiment analysis task.
- **Removing Mentions and URLs**: All mentions (@usernames) and URLs were removed using regular expressions.
- **Lowercasing**: All text was converted to lowercase to ensure uniformity.
- **Removing Special Characters**: Special characters, punctuations, and extra whitespaces were removed.
- **Tokenization**: The text was split into individual words (tokens).
- Stemming was applied to reduce inflectional forms of words to their base forms. For example, "writing", "writes", and "wrote" were reduced to "write". This step helps in normalizing the text data.
- Transformation of Target Variable for Binary Classification
- The preprocessed data was then split into training and test sets to evaluate the model's performance. Typically, an 70-30 split was used, where 70% of the data was used for training and 30% for testing.
- The transformed data is then loaded to Athena

MACHINE LEARNING MODEL DEVELOPMENT

The machine learning model development and pipeline orchestration were crucial components of this project. By leveraging PySpark's MLlib, we efficiently processed the data and built a robust sentiment analysis model. Below, I describe the steps and design choices involved in the machine learning pipeline.

Pipeline Stages

The machine learning pipeline was designed to include several key stages, each responsible for specific tasks in the data processing and model training workflow. The stages included tokenization, feature extraction, and the application of a logistic regression model. Here is a detailed breakdown of each stage:

1. **Tokenizer** - Converts the input text column into a column of tokenized words.
2. **HashingTF -** Applies the hashing trick to map tokens to their term frequencies.
3. **IDF (Inverse Document Frequency) -** Scales the term frequencies by the inverse document frequency to down-weight common terms.
4. **Logistic Regression -** A machine learning algorithm for binary classification tasks.

After defining the individual stages, the pipeline was constructed and used to train the model on the preprocessed training data. The trained model was then used to make predictions on the test data.

EVALUATION AND RESULTS

The model's performance was evaluated using a variety of metrics to ensure a comprehensive understanding of its effectiveness. Below are the details of the evaluation results:

Area Under the Curve (AUC)

The AUC score, which measures the ability of the model to distinguish between the positive and negative classes, was found to be 0.7641. This score indicates a good level of discrimination, meaning the model is relatively effective at predicting the correct sentiment for the tweets.

- **AUC**: 0.7795042732782639

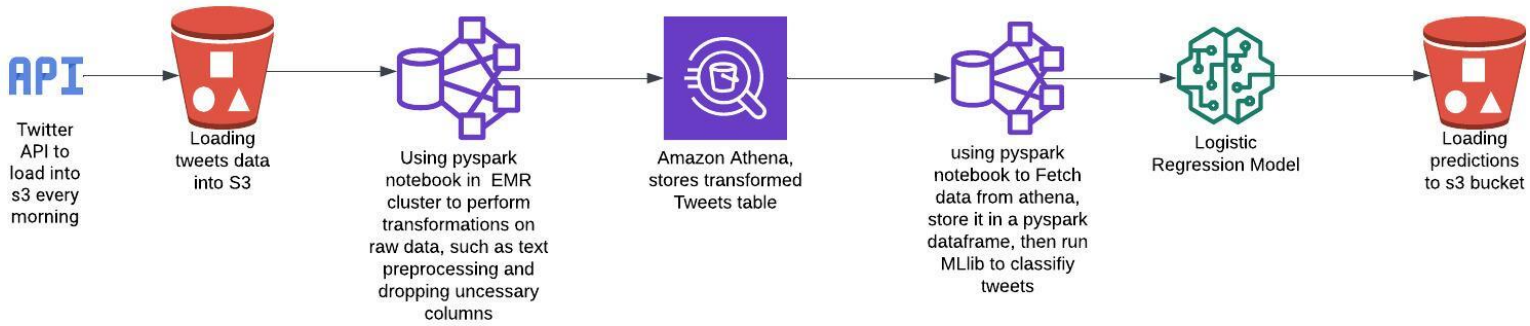
Precision, Recall, F1-Score, and Accuracy

While multiple metrics were evaluated to provide a comprehensive understanding of the model's performance, accuracy was chosen as the primary measure for this project. Accuracy represents the overall correctness of the model's predictions and is a straightforward metric that aligns well with the objective of correctly classifying tweet sentiments.

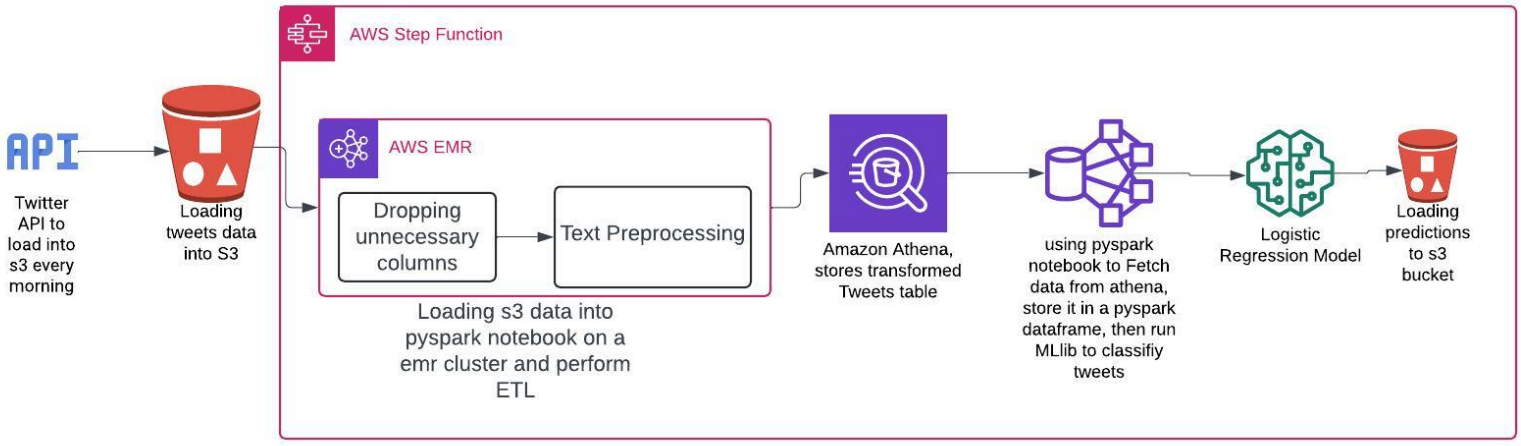
Here are the values for these metrics:

- **Precision**: 0.7478654424625483
- **Recall**: 0.7476235339779083
- **F1-Score**: 0.7475744319527502
- **Accuracy**: 0.7476235339779084

ARCHITECTURE DIAGRAM



FULL ARCHITECTURE DIAGRAM



CHALLENGES FACED:

During the project, a significant challenge arose from AWS sessions timing out after four hours of inactivity. This meant that any inactive sessions would require rerunning the entire pipeline from the beginning. This not only disrupted workflow continuity but also resulted in substantial time losses due to repeated setup and execution.

BENEFITS OF DISTRIBUTED COMPUTING:

Distributed computing was crucial in making project efficient and scalable. By using tools like Apache Spark on AWS, I could easily adjust our computing power based on what we needed at the time. This flexibility meant I could handle large amounts of data without slowing down. Being able to process data in parallel across multiple servers significantly cut down on the time it took to get results.

CONCLUSION AND FUTURE WORK:

This project has showcased the immense benefits of using distributed computing, particularly Apache Spark on AWS, to streamline our data analysis efforts. This approach has not only made our data processing faster and more reliable but also scalable enough to handle large volumes of information effortlessly. Achieving an impressive AUC score of 0.779 validates the effectiveness of our machine learning models in sentiment analysis.

Looking ahead, there are exciting opportunities for further improvements. We aim to boost our model's accuracy and resilience by fine-tuning it with advanced algorithms. Introducing real-time data streaming capabilities, possibly using technologies like Apache Kafka, would enable us to perform sentiment analysis continuously, making our insights more relevant and timely. Expanding our dataset to include diverse sources and languages will broaden the applicability of our sentiment analysis model. Additionally, automating deployment processes using Docker and Kubernetes could optimize our operations, ensuring we can scale up seamlessly as our needs grow.

There's still much to explore and refine.