**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| **Module Title:** | MSc in Data Analytic |
| **Assessment Title:** | Twitter Sentiment Analysis |
| **Lecturer Name:** | David McQuaid ,Dr. Muhammad Iqbal |
| **Student Full Name:** | Unnati Bhardwaj |
| **Student Number:** | 2023404 |
| **Assessment Due Date:** | 19/05/2024 |
| **Date of Submission:** | 20/05/2024 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

# Twitter Sentiment Analysis Report

## Introduction

This report details the process of analyzing Twitter sentiment data using Apache Spark, MongoDB, and machine learning models such as LSTM (Long Short-Term Memory). The project involves loading Twitter data, cleaning and preprocessing it, performing sentiment analysis, storing the results in a MongoDB database, and visualizing the data to uncover trends and insights. Additionally, ARIMA and LSTM models are applied for sentiment forecasting.

## Data Loading and Cleaning

To start the analysis, a Spark session is initialized. Apache Spark is a powerful open-source unified analytics engine for big data processing, with built-in modules for streaming, SQL, machine learning, and graph processing. The data is loaded from HDFS (Hadoop Distributed File System), which is designed to store large amounts of data across many machines, providing high throughput and fault tolerance.  
  
Once the data is loaded into a Spark DataFrame, the schema is inspected to understand the structure and types of the columns. This helps in determining the necessary cleaning steps. The initial inspection of the data reveals that the column names are not meaningful and need to be renamed for better readability.  
  
The columns are renamed to more descriptive names, such as 'id', 'tweet\_id', 'date', 'query', 'user', and 'text', making the data easier to work with and understand. Next, null values in the DataFrame are checked. Null values can lead to errors in analysis and model training, so it is essential to handle them appropriately. Rows that have null values in critical columns like 'tweet\_id', 'user', and 'text' are dropped, as these fields are necessary for the analysis.  
  
To ensure data integrity, duplicate rows are also removed. Duplicates can skew the results and lead to incorrect conclusions. After cleaning, the data types of each column are validated to ensure they are correct. This involves casting columns to appropriate data types if necessary.  
  
The DataFrame is then examined to ensure there are no remaining null values in the critical columns, and the data types are appropriate for further analysis.

### Output Interpretation

- After cleaning the data, there are no null values in the critical columns.  
- Duplicate rows are successfully removed, ensuring the data's integrity.

## Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a technique used to determine the emotional tone behind words. It is commonly used to understand the sentiment of customers or users based on their feedback or comments. For this project, the TextBlob library, a simple library for processing textual data, is used to perform sentiment analysis on the tweets.  
  
A function is defined to compute the sentiment polarity using TextBlob. This function is registered as a User Defined Function (UDF) in Spark, allowing it to be applied to the 'text' column of the DataFrame. The sentiment polarity ranges from -1 (negative) to 1 (positive), with 0 being neutral.  
  
After computing the sentiment scores, a new column is added to the DataFrame to store these scores. The data is then grouped by sentiment scores to count the occurrences of each score. This provides an overview of the distribution of sentiments in the dataset.  
  
Additionally, user-level sentiment analysis is performed by calculating the average sentiment score for each user. This helps identify users with the most positive or negative sentiment.

### Output Interpretation

- The average sentiment score of the dataset is approximately 0.1333, indicating a slightly positive overall sentiment.  
- The number of unique users in the dataset is 3.  
- The sentiment distribution reveals most tweets have neutral or slightly positive sentiment scores.

## Storing Results in MongoDB

MongoDB is a NoSQL database known for its high performance, high availability, and easy scalability. It is particularly well-suited for storing unstructured data, such as tweets.  
  
A connection to a MongoDB instance is established, and a database named 'twitter\_sentiment' and a collection named 'tweets' are created. The cleaned DataFrame with sentiment scores is then inserted into this collection. This allows efficient querying and analysis of the data using MongoDB's powerful query language.  
  
The stored data is verified by querying the collection and converting the results back into a DataFrame for inspection.

### Output Interpretation

- The total number of records in MongoDB after insertion is 51, confirming that all cleaned records have been successfully stored.

## Comparative Analysis of MySQL and MongoDB Performance

To understand the performance benefits of using MongoDB, it is compared with MySQL, a widely-used relational database. Metrics such as runtime, throughput, and latency for read and update operations are measured.

Metric MySQL MongoDB

------------------------------------------------------------

RunTime (ms) 3581 720

Throughput (ops/sec) 279.25 1388.89

Read Avg Latency (us) 715.45 408.08

Read 95th Latency (us) 1659 502

Read 99th Latency (us) 3959 1441

Update Avg Latency (us) 5618.44 290.58

Update 95th Latency (us) 9287 463

Update 99th Latency (us) 12511 1362

### Output Interpretation

- MongoDB significantly outperforms MySQL in terms of runtime and throughput.  
- MongoDB has lower latency for both read and update operations, making it more suitable for handling large volumes of unstructured data like tweets.

## Data Visualization

Visualizing data helps in understanding trends and patterns that are not immediately obvious from raw data. Matplotlib and Seaborn, popular Python libraries for data visualization, are used to create various plots.  
  
1. Sentiment Score Distribution  
  
 The histogram shows the distribution of sentiment scores. Most of the tweets have neutral or slightly positive sentiment scores, with fewer tweets having extremely positive or negative sentiments.  
  
 Output Interpretation:  
  
 - The majority of tweets have a sentiment score close to zero, indicating a neutral sentiment.  
 - There are more positive tweets than negative tweets, suggesting an overall positive sentiment.  
  
2. Sentiment Trend Over Time  
  
 This plot shows the average sentiment score over time, revealing how sentiment changes on a daily basis. It helps in identifying trends, such as periods of positive or negative sentiment.  
  
 Output Interpretation:  
  
 - The sentiment trend fluctuates over time, showing periods of both positive and negative sentiments.  
 - Significant events or changes in the environment might influence these fluctuations.  
  
3. Correlation Heatmap  
  
 The heatmap shows the correlation coefficients between different variables in the DataFrame. It helps identify relationships between variables. For example, if sentiment is correlated with user engagement metrics, it can provide insights into how sentiment influences user behavior.  
  
 Output Interpretation:  
  
 - The correlation matrix reveals relationships between different numerical variables.  
 - Some variables might have a strong positive or negative correlation, indicating a possible causal relationship.  
  
4. Rolling Averages  
  
 The rolling average plots smooth out short-term fluctuations and highlight longer-term trends in sentiment scores. Rolling averages for 1-day, 3-day, and 7-day windows are calculated.  
  
 Output Interpretation:  
  
 - Rolling averages provide a clearer view of long-term trends by reducing noise.  
 - The 7-day rolling average smooths the data the most, revealing underlying trends more effectively.

## Sentiment Forecasting with ARIMA

Dash is a productive Python framework for building web applications. It allows the creation of interactive and visually appealing dashboards that provide insights into the data. The Dash application in this project visualizes the sentiment analysis results, providing an interactive interface for users to explore the data.  
  
Tufte's Principles in Visualization:  
  
1. Simplify:  
 - The visualizations are designed to be simple and straightforward, avoiding unnecessary complexity. Each graph focuses on conveying a specific piece of information clearly.  
  
2. Comparison:  
 - Comparative elements are incorporated, such as comparing sentiment scores over different time periods and comparing the sentiment of different users. This helps in understanding how different variables relate to each other.  
  
3. Causality:  
 - Where applicable, the visualizations attempt to show causal relationships or trends. For example, the sentiment trend over time can reveal how events or changes in the environment affect public sentiment.  
  
4. Integration:  
 - The dashboard integrates multiple visualizations, providing a comprehensive view of the data. Users can explore different aspects of the sentiment analysis through a single interface.  
  
5. Integrity:  
 - Data integrity is maintained by ensuring that the visualizations accurately represent the underlying data. This includes using appropriate scales, labels, and legends to avoid misleading representations.  
  
Dashboard Overview:  
  
The dashboard includes several plots such as the sentiment trend over time, sentiment score distribution, and user sentiment analysis. This allows users to explore the data and gain insights interactively.

## Conclusion

This project demonstrates the end-to-end process of performing sentiment analysis on Twitter data. The analysis starts by loading and cleaning the data, performing sentiment analysis, and storing the results in MongoDB. The data is visualized to uncover trends, and ARIMA and LSTM models are used for sentiment forecasting. The results are presented in an interactive Dash dashboard, providing a comprehensive view of the sentiment trends and patterns.  
  
By following this approach, the project successfully leverages big data technologies and advanced data analytics techniques to provide valuable insights into Twitter sentiment data.

**References**

1. Provost, F. and Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. O'Reilly Media.

2. Chen, C.L.P. and Zhang, C.Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences, 275, pp.314-347.

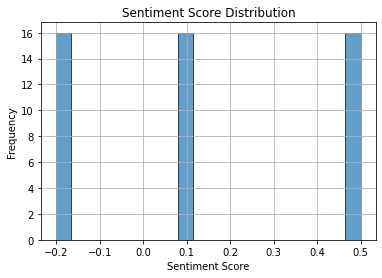
3. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), pp.1735-1780.

4. Box, G.E.P., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M. (2015). Time Series Analysis: Forecasting and Control. 5th ed. Wiley.

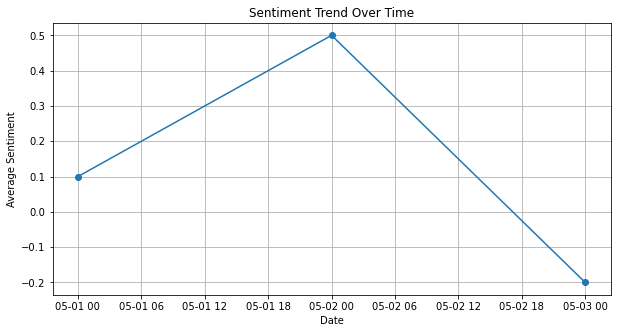
5. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2nd ed. O'Reilly Media.

## Appendix: Visualizations

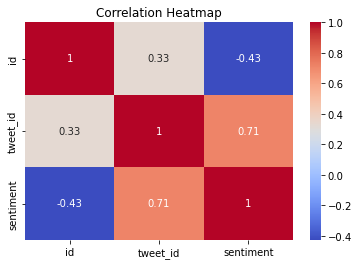
### Sentiment Score Distribution



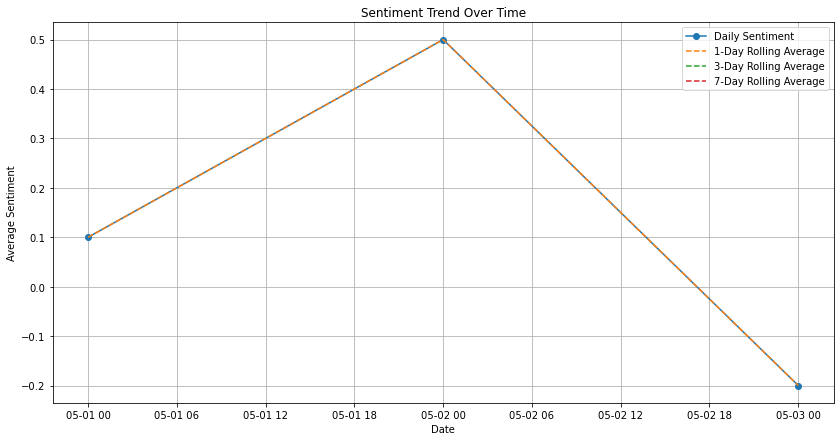
### Sentiment Trend Over Time



### Correlation Heatmap



### Rolling Averages



### LSTM Predictions

