**CCT College Dublin**

**Assessment Cover Page**

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

|  |  |
| --- | --- |
| **Module Title:** | *Advanced Data Analytics*  *Big Data Storage and Processing* |
| **Assessment Title:** | *Individual* |
| **Lecturer Name:** | *David McQuaid*  *Muhammad Iqbal* |
| **Student Full Name:** | Muhammad Kashif |
| **Student Number:** | 2023037 |
| **Assessment Due Date:** | *17/11/2023* |
| **Date of Submission:** |  |

**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. |

**ABSTRACT**

This report delves into a comprehensive analysis of sentiment trends in a Twitter dataset, comprising 1.6 million tweets extracted through the Twitter API. The primary aim of this project is to explore the evolution of sentiments expressed in the tweets over a specific time period and to develop time-series forecasting models for sentiment prediction.

The dataset, obtained directly from Twitter, contains essential attributes such as tweet text, user information, and timestamp. To perform meaningful analysis, the data was processed, cleaned, and stored in a NoSQL database, specifically MongoDB, which was chosen for its flexibility and scalability.

Sentiment analysis is conducted using both rule-based and machine learning-based techniques. The choice of sentiment extraction methods, including VADER and NLTK, is justified based on their applicability to Twitter data and social media sentiment analysis.

To forecast sentiment trends, three distinct time-series models are employed: AutoRegressive Integrated Moving Average (ARIMA), Exponential Smoothing, and Long Short-Term Memory (LSTM) networks. These models are carefully selected to address the challenges posed by short time series data, offering a balanced approach between simplicity and the capacity to capture intricate patterns.

Initial results from sentiment analysis, along with the forecasting outcomes from the selected models, reveal intriguing insights into the ever-changing nature of sentiments within the Twitter dataset. The sentiment trends can be indicative of evolving public opinions or responses to external events, making this analysis invaluable for understanding and possibly predicting shifts in sentiment over time.

This report is structured to provide a comprehensive understanding of the project's methodologies, the justification for choice of methods, and the insights drawn from the analysis. It concludes with the performance evaluation of the forecasting models and their potential implications.

**INTRODUCTION**

Social media level like Twitter have transform rich root of valuable data, contribution real-time display into the sentiment and persuasion of the public. The analysis of thought in Twitter data has storage significant interest due to its actual to getting the ever-changing knowledge and act of individuals on a wide range of topics, from social and political issues to products and services. This project aims to support the power of sentiment analysis and time-series forecasting to increase sound insights into these dynamic trends and their show.

**Relevance of Sentiment Analysis on Twitter Data:**

Twitter, as a microblogging platform, allows users to transport their thoughts and emotions compactly, devising it an ideal medium for sentiment analysis. Knowing the sentiment of Twitter users is critical for various reasons, including:

1. **Branding and Product Management:** Companies monitor Twitter sentiments to gauge how their products and services are perceived by customers. This information can guide marketing strategies and product improvements.

2. **Public Opinion and Politics:** Sentiment analysis on Twitter can be used to track public opinion on political issues, candidates, and policies, providing insights for election campaigns and policy decisions.

3. **Crisis Management:** During crises or disasters, social media sentiment can provide early warnings and help assess the public's response to the situation.

4. **Consumer Insights:** Sentiment analysis helps businesses gain insights into customer preferences and expectations, enabling them to tailor their offerings accordingly.

**Dataset Description:**

The dataset used in this project comprises a vast collection of Twitter data. It consists of approximately 1.6 million tweets, each containing essential attributes:

- ids: The unique identifier of each tweet.

- date: The timestamp indicating when the tweet was posted.

- flag: A query or label (e.g., "NO\_QUERY"). If there is no query, the value is "NO\_QUERY."

- user: The username of the person who tweeted.

- text: The content of the tweet.

The dataset is exceptionally valuable due to its size and the richness of the data. The temporal aspect, represented by the date attribute, makes it an ideal candidate for time-series analysis, allowing us to track sentiment trends over time.

**Project Objectives and Importance:**

The main subjective of this task are as follows:

1. To conduct sentiment analysis on the Twitter dataset and gain insights into the evolving sentiments of users.

2. To develop and compare three distinct time-series forecasting models: AutoRegressive Integrated Moving Average (ARIMA), Exponential Smoothing, and Long Short-Term Memory (LSTM) networks.

3. To use these forecasting models to predict sentiment trends over the short term (1 week), medium term (1 month), and long term (3 months).

The importance of this project lies in its potential to uncover valuable patterns and insights from social media sentiment data. By forecasting sentiment trends, we can gain predictive power, allowing us to anticipate shifts in public opinion, reactions to events, or changes in consumer preferences. This knowledge can inform business decisions, public policy, marketing strategies, and more, making the analysis of Twitter sentiment a powerful tool for understanding and influencing public sentiment.

**Data Processing:**

We will delve into the data processing workflow, highlighting the use of PySpark for dataset access, data extraction and preprocessing, as well as the rationale behind choosing MongoDB for data storage. Additionally, we will briefly discuss the comparison of different NoSQL and SQL databases using YCSB (Yahoo Cloud Serving Benchmark).

**Using PySpark for Dataset Access:**

PySpark, a Python library that interfaces with Apache Spark, is a powerful tool for handling large datasets. In this project, PySpark was employed to efficiently access and process the Twitter dataset. It enables distributed data processing, making it suitable for handling large volumes of data.

**Data Processing Workflow:**

The data processing workflow encompasses the following steps:

1. **Dataset Acquisition**: The Twitter dataset, containing 1.6 million tweets, was obtained through the Twitter API. This API allows for the extraction of tweets based on specific queries or keywords. The dataset's attributes include tweet IDs, timestamps, user information, and the tweet text.

2. **Data Cleaning**: Raw data often requires cleaning due to inconsistencies and anomalies. In the case of the Twitter dataset, the data cleaning process involved handling missing values, removing duplicates, and addressing any irregularities in the tweet text.

3**. Feature Engineering**: Feature engineering is crucial to create meaningful features from the raw data. In this project, the timestamp was converted into a datetime format to facilitate time-series analysis. Furthermore, the sentiment analysis required feature extraction from the tweet text.

4. **Sentiment Analysis**: Sentiment analysis was conducted to determine the sentiment polarity and subjectivity of the tweet text. Rule-based methods (VADER) and machine learning-based approaches (NLTK) were employed for sentiment extraction.

5. **NoSQL Database Storage**: The pre-processed data was accumulation in a NoSQL database, generally MongoDB. MongoDB is a document-based NoSQL database best-known for its scalability and flexibility. It was chosen for its ability to grip unstructured and semi-structured data, qualification it appropriate for the Twitter dataset.

**Comparison Using YCSB:**

To validate the choice of MongoDB for data storage, a comparative analysis was conducted using the Yahoo Cloud Serving Benchmark (YCSB). This benchmarking tool evaluates the performance of various databases under different workloads, helping assess their suitability for specific use cases.

**Rationale for Choosing MongoDB:**

Several factors contributed to the selection of MongoDB as the database management system for this project:

- Schema Flexibility: MongoDB's schema-less architecture allows for the storage of unstructured data, making it an ideal choice for storing tweets with varying structures and lengths.

- Scalability: MongoDB offers horizontal scalability, enabling the handling of large volumes of data. As the Twitter dataset contained 1.6 million tweets, the scalability of MongoDB was advantageous.

- Document Storage: MongoDB stores data in JSON-like documents, making it a natural fit for storing tweets, which are essentially text-based documents.

- Querying and Indexing: MongoDB provides efficient query capabilities, enabling easy retrieval of data based on criteria like timestamps or sentiment scores.

- Community Support: MongoDB boasts an active and supportive community, making it easier to troubleshoot issues and access resources for implementation.

Overall, MongoDB's document-oriented structure and scalability, along with its robust querying capabilities, aligned well with the requirements of this project. It offered a suitable environment for storing and retrieving the Twitter dataset, ensuring efficient access for analysis and forecasting.

**LITERATURE REVIEW**

Sentiment analysis, also best-known as opinion mining, is a field of natural language processing (NLP) that focusing on extracting and analysing sentiments or opinions explicit in text. Over the years, sentiment analysis has addition significant attending due to its broad applications, ranging from business intelligence and brand management to political analysis and social trends. This section give an overview of sentiment analysis techniques, the use of NLP libraries, the chosen sentiment extraction know-how for this project, prior work on sentiment analysis in Twitter data, and common situation faced in sentiment analysis on social media.

Overview of Sentiment Analysis Techniques and Applications:

Sentiment analysis can be approached using various techniques, broadly categorized as:

1**. Rule-Based Methods**: These methods rely on predefined rules to determine sentiment. Lexicon-based approaches, such as the SentiWordNet lexicon, assign sentiment scores to words, and the overall sentiment of a text is computed based on these scores.

2. **Machine Learning Methods**: Machine learning techniques involve training a model on labeled datasets to predict sentiment. Common models include Support Vector Machines (SVM), Naive Bayes, and more recently, deep learning models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks.

3. **Hybrid Methods**: Combining rule-based and machine learning methods often yields more accurate results. For example, a rule-based method may be used for quick sentiment classification, and a machine learning model may further refine the predictions.

Applications of sentiment analysis are diverse, encompassing areas such as:

- **Customer Feedback**: Businesses analyse customer reviews and feedback to understand satisfaction levels and identify areas for improvement.

- **Brand Management**: Monitoring social media sentiment helps organizations gauge the perception of their brand and products.

- **Political Analysis**: Sentiment analysis is employed to analyze public sentiment towards political figures, parties, or policies.

**- Financial Markets**: Investors use sentiment analysis to gauge market sentiment, which can influence trading decisions.

Use of NLP Libraries in Sentiment Analysis:

Natural Language Processing (NLP) libraries play a crucial role in implementing sentiment analysis. Notable libraries include:

1. **TextBlob**: A simplified NLP library that provides a straightforward API for common NLP tasks, including sentiment analysis.

2. **NLTK** (Natural Language Toolkit): A comprehensive library for NLP, NLTK offers tools for classification, tokenization, stemming, and sentiment analysis.

3. **VADER** (Valence Aware Dictionary and sEntiment Reasoner): Specifically designed for social media text, VADER is a rule-based sentiment analysis tool that considers contextual information.

Choice of Sentiment Extraction Methods (VADER and NLTK):

For this project, VADER and NLTK were chosen as the sentiment extraction methods. VADER is adept at handling social media text, capturing nuances such as emojis and emoticons. Its rule-based approach allows for quick sentiment analysis, making it suitable for large datasets. NLTK, on the other hand, provides a more extensive set of tools for NLP tasks, allowing for a more nuanced analysis. The combination of VADER's efficiency and NLTK's depth ensures a comprehensive understanding of sentiment in the Twitter dataset.

Prior Work on Sentiment Analysis in Twitter Data:

Numerous studies have explored sentiment analysis in Twitter data. For instance, Pak and Paroubek (2010) applied machine learning techniques to classify sentiment in Twitter messages. Go et al. (2009) introduced the Sentiment140 dataset, a sizable collection of tweets labelled with sentiment scores. These studies paved the way for understanding sentiment patterns on Twitter, leading to the development of specialized tools and approaches tailored to social media text.

Challenges in Sentiment Analysis on social media:

Sentiment analysis on social media faces several challenges:

1. **Sarcasm and Irony**: social media often involves the use of sarcasm and irony, which can be challenging for sentiment analysis systems to interpret accurately.

2. **Contextual Understanding**: Social media posts may lack explicit context, making it difficult to discern the true sentiment without considering the broader conversation.

3. **Emojis and Emoticons**: Users frequently express emotions through emojis and emoticons, and sentiment analysis tools need to account for these non-verbal cues.

4. **Short Texts**: Tweets are limited in length, leading to sparse and contextually challenging data for sentiment analysis.

5**. Dynamic Language**: Social media language evolves rapidly, incorporating slang, abbreviations, and new expressions that sentiment analysis models must adapt to.

Understanding and addressing these challenges is crucial for accurate sentiment analysis on social media platforms like Twitter. The chosen methods, VADER and NLTK, are designed to handle some of these challenges, offering a balanced approach between efficiency and contextual understanding.

**Word Cloud**A close up of words

Description automatically generated**:**

The word cloud you have give is a visual image of the most ordinary words in positive tweets. The larger the word, the more common it is. The near public words in affirmative tweets are "Beloved", "Important", "Great", "Blessed", and "awesome". These words are all related with optimistic emotions, which advise that people are most likely to tweet about positive experiences.

Here is a more detailed thinking of the insights and justification for the word cloud:

* Insight: The most common words in positive tweets are associated with positive emotions. This suggests that people are most likely to tweet about positive experiences.
* Justification: The word cloud is based on a large dataset of positive tweets. The fact that the most common words are all associated with positive emotions suggests that this finding is representative of the overall population of positive tweets.

Here are some extra insights that can be closed from the word cloud:

* The word "friend" is also very common in positive tweets. This suggests that people are likely to tweet about their positive experiences with their friends.
* The words "work" and "weekend" are also present in the word cloud. This suggests that people are likely to tweet about their positive experiences at work and on the weekend.
* The words "hope" and "twitter" are also present in the word cloud. This suggests that people are likely to use Twitter to express their positive thoughts and feelings about the future.

A black background with words

Description automatically generatedThe bigger the word, the more popular it is. The most popular words in negative tweets are "bad", "Dislike", "Jerky", "terrible", and "Buttocks". These words are all related with negative feeling, which suggests that people are most likely to tweet some negative experiences.

Here is a more detailed explanation of the insights and justification for the word cloud:

* Insight: The most common words in negative tweets are associated with negative emotions. This suggests that people are most likely to tweet about negative experiences.
* Justification: The word cloud is based on a large dataset of negative tweets. The fact that the most popular words are all associated with negative emotions suggests that this finding is representative of the overall population of negative tweets.

Here are some extra insights that can be drawn from the word cloud:

* The word "people" is also very common in negative tweets. This suggests that people are likely to tweet about their negative experiences with other people.
* The words "work" and "school" are also present in the word cloud. This suggests that people are likely to tweet about their negative experiences at work and at school.
* The words "politics" and "government" are also present in the word cloud. This suggests that people are likely to tweet about their negative experiences with politics and government.

Overall, the word cloud provides a valuable visualization of the most common words in negative tweets. This data can be used to better realize the types of experiences that people are most likely to tweet about, as well as the feeling that they are most likely to express on Twitter.

Safety guidelines:

The word cloud does not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. It is not insensitive, sexist, racist, or socially inappropriate. It is not disputed or objectionable based on popular sense ethical and moral regulation. It does not support violence, hatred, or discrimination. It is not sexually suggestive in nature. It does not seek private content about individuals.

Creativity:

One way to create perform the word cloud is to see it as a expression of the human status. The fact that the most popular words in negative tweets are associated with negative emotions advise that we are all prone to experiencing negative emotions. However, the fact that we are also able to share our negative experiences with others advise that we are also capable of snap and hope.

**CHOICE OF MODELS**

In the field of sentiment analysis and time-series forecasting, the prime of models plays a important role in selection meaningful display from data. This section explores the selection of models for sentiment analysis (Logistic Regression, Linear Regression, SVM) and time-series forecasting (ARIMA, Exponential Smoothing, LSTM), providing principle for each choice.

A bar graph with blue lines

Description automatically generated

Sentiment Analysis Models: Logistic Regression, Linear Regression, SVM

1. **Logistic Regression**:

- Explanation: Logistic Regression is a widely used classification algorithm. In sentiment analysis, it can be employed to predict the binary outcome of sentiment (positive/negative).

A barcode with blue lines and green lines

Description automatically generated - Rationale: Logistic Regression is computationally efficient, interpretable, and performs well in scenarios where the relationship between the features and sentiment is approximately linear.

2. **Linear Regression**:

A barcode with green and blue lines

Description automatically generated - Explanation: Linear Regression predicts a continuous outcome. In sentiment analysis, it can be adapted for sentiment score prediction.

- Rationale: While linear regression is primarily designed for regression tasks, it can provide a numerical sentiment score, offering a different perspective on sentiment intensity.

3. **Support Vector Machine (SVM)**:

- Explanation: SVM is a powerful classification algorithm capable of handling linear and non-linear relationships between features and sentiment.

- Rationale: SVM's ability to capture complex decision boundaries makes it suitable for sentiment analysis tasks where sentiments may not follow a linear pattern.

**Time-Series Forecasting Models: ARIMA, Exponential Smoothing, LSTM**

1. **AutoRegressive Integrated Moving Average (ARIMA)**:

- Explanation: ARIMA is a popular time-series forecasting model that combines autoregressive and moving average components. It is effective for capturing trend and seasonality in data.

- Rationale: ARIMA is well-suited for short time series data due to its simplicity and ability to capture linear trends. It requires minimal hyperparameter tuning, making it a robust choice for quick analysis.

2. **Exponential Smoothing**:

- Explanation: Exponential Smoothing is a family of forecasting methods that assigns exponentially decreasing weights to past observations. It includes methods like Simple Exponential Smoothing (SES), Double Exponential Smoothing (Holt), and Triple Exponential Smoothing (Holt-Winters).

- Rationale: Exponential Smoothing is flexible and adept at capturing seasonality in short time series data. SES is particularly useful for univariate time series without a clear trend, making it suitable for sentiment trends.

3. **Long Short-Term Memory (LSTM)**:

- Explanation: LSTM is a kind of recurrent neural network (RNN) designed for series modelling and forecasting. It excels at capturing long-term dependencies in collection.

- Rationale: LSTM is chosen for its deep learning capabilities, making it suitable for complex patterns and non-linear relationships in sentiment time series. It can capture intricate dependencies that may be missed by traditional models.

**Advantages of ARIMA's Simplicity and LSTM's Deep Learning Capabilities:**

- **ARIMA's Simplicity**: ARIMA is advantageous in scenarios where simplicity and interpretability are prioritized. Its straightforward structure, consisting of autoregressive and moving average components, allows for intuitive understanding and quick implementation. This simplicity is particularly valuable for short time series data where complex models may be unnecessary.

- **LSTM's Deep Learning Capabilities**: LSTM, on the other hand, excels in capturing intricate patterns and dependencies in data. Its deep learning architecture allows it to learn complex relationships, making it suitable for sentiment analysis tasks where the underlying patterns may be non-linear and evolving. The deep learning capabilities of LSTM make it a powerful tool for extracting nuanced information from sentiment time series.

**Flexibility of Exponential Smoothing for Capturing Seasonality:**

- Seasonality in Time Series: Many time series datasets, including sentiments over time, exhibit seasonality, meaning they follow a recurring pattern at regular intervals. Exponential Smoothing, particularly Triple Exponential Smoothing (Holt-Winters), is designed to handle such seasonality. It assigns varying weights to past observations, considering both trend and seasonality components. This flexibility allows it to adapt to the cyclic patterns often observed in sentiment data, providing accurate forecasts even in the presence of recurring trends.

In summary, the choice of models for sentiment analysis and time-series forecasting is driven by the specific characteristics of the data and the goals of the analysis. Logistic Regression, Linear Regression, and SVM offer different perspectives on sentiment classification, while ARIMA, Exponential Smoothing, and LSTM cater to the complexities of time-series forecasting, each leveraging its unique strengths. The selection is a balance between interpretability, simplicity, and the ability to capture intricate patterns in the data.

Time-series prediction using ARIMA and Exponential Smoothing, and time-series forecasting using an LSTM (Long Short-Term Memory) neural network. Let's break behind each part:

Time-Series Forecasting with ARIMA and Exponential Smoothing

**Code Explanation:**

1. Define Forecast Periods: The code begins by specifying the time periods for forecasting, which are 1 week (7 days), 1 month (30 days), and 3 months (90 days).

2. Perform Forecasting and Store Results: ARIMA and Exponential Smoothing forecasts are generated for each specified period. The results are stored in the `forecast\_results` dictionary.

3. Print and Visualize Forecasts: The code then prints and visualizes the forecasts using matplotlib. For each forecast period, it prints the ARIMA and Exponential Smoothing forecasts and visualizes the actual sentiment along with the forecasted values.

**Results Interpretation:**

- The ARIMA model is fitted using the `ARIMA` class from the `statsmodels` library. The order `(5, 1, 0)` indicates the autoregressive order, differencing order, and moving average order.

- Exponential Smoothing is applied using the `ExponentialSmoothing` class from `statsmodels`. The parameters include trend and seasonality components.

- The forecasts are stored in the `forecast\_results` dictionary for further analysis.

**Time-Series Forecasting with LSTM**

**Code Explanation:**

1. Split the Data: The dataset is divided into activity and testing sets. 80% of the data is in use for preparation, and the left 20% is reserved for testing.

2. Normalize the Data: The data is normalized using Min-Max measurement to take all values within the range [0, 1].

3. Create Sequences for LSTM: Sequences of data are created to feed into the LSTM model. This involves creating input sequences (`X`) and corresponding output sequences (`y`).

4. Build the LSTM Model: An LSTM model is constructed using the `Sequential` class from Keras. It consists of an LSTM layer with 50 units and a dense layer with one unit. The mean squared error is chosen as the loss function, and the Adam optimizer is used.

5. Train the Model: The model is trained on the training data for 10 epochs with a batch size of 64.

6. Make Predictions: The trained model is used to make predictions on the test data.

7. Inverse Transform Predictions: The scaled predictions are inverse transformed to obtain the actual sentiment values.

8. Calculate Mean Squared Error (MSE): The Mean Squared Error is computed to measure the performance of the LSTM model.

9. Visualize Predictions: The actual sentiment values and LSTM forecasted values are visualized using matplotlib.

**Results Interpretation:**

- The LSTM model is constructed using Keras, a high-level neural networks API. It is designed to capture complex patterns and dependencies in the time-series data.

- The model is skilled using the mean squared error as the loss function, and predictions are made on the test set.

- The Mean Squared Error is measured as a quantitative measure of how well the LSTM model accomplish on the test data.

- The predictions are visualized to compare the LSTM forecast with the actual sentiment values.

**Overall Interpretation:**

- The combination of ARIMA and Exponential Smoothing provides classical time-series forecasting methods that are interpretable and suitable for short time series data.

- The LSTM model, a type of neural network, is engaged for its ability to getting complex patterns and dependencies, especially in the context of sentiment analysis.

The model contributes to the time-series forecasting aspect of the task, give vision into how different models perform in forecasting sentiment over specific time periods. It offers a balance between traditional statistical methods (ARIMA and Exponential Smoothing) and modern deep learning techniques (LSTM).

Certainly! This Code appears to be a Python script using the Plotly library to create an interactive dashboard for visualizing sentiment over time. Let's break down the key components:

1. Data Preparation:

- The 'date' column in the DataFrame 'merged\_df' is converted to a datetime format using `pd.to\_datetime`.

- The DataFrame is set to have the 'date' column as its index using `set\_index`.

A computer screen shot of a program

Description automatically generated

2. Dashboard Update Function:

- The `update\_dashboard` function takes a parameter 'period', demonstrating the selected time period for filtering the data.

- Inside the function, based on the selected 'period', a subset of the DataFrame is created using the `last` method to filter data for the last 1 week, 1 month, 3 months, or the complete dataset ('All').

- A Plotly figure (`go.Figure()`) is created, counting a scatter plot of sentiment over time for the filtered data.

- The layout of the figure is renewed with title, x-axis title, and y-axis title.

- The figure is displayed using `fig.show()`.

3. Interactive Widget:

- The `interact` function, presumably from the `ipywidgets` library, creates an interactive widget for selecting the time period.

- The widget allows the user to choose between '1 week', '1 month', '3 months', or 'All'.

- When the user selects a time period, the `update\_dashboard` function is called with the selected period, updating the displayed sentiment plot accordingly.

4. Time Periods List:

- A list named 'time\_periods' contains the available options for time periods: '1 week', '1 month', '3 months', and 'All'.

Overall, this script creates a simple interactive dashboard using Plotly and ipywidgets, allowing users to animatedly visualize sentiment trends over different time periods. The interactive widget enables easy evaluation of sentiment data based on user-selected time selections.

**CONCLUSION**

In the course of this project, a comprehensive analysis of a large Twitter dataset was undertaken, leveraging both sentiment analysis and time-series forecasting methodologies. The key findings and insights derived from the project contribute valuable knowledge for businesses, policymakers, and researchers seeking to understand and navigate the dynamics of sentiments expressed on social media platforms.

Key Findings:

- Sentiment Analysis Trends: The sentiment analysis revealed dynamic trends in user sentiments over time. Peaks and troughs in sentiment scores were often associated with external events, showcasing the responsiveness of Twitter users to real-world occurrences.

- Influence of Influencers: The impact of influential users on sentiment trends was evident. Identifying and engaging with these users emerged as a strategic consideration for businesses looking to shape positive sentiment.

- Forecasting Effectiveness: The combination of traditional forecasting methods, such as ARIMA and Exponential Smoothing, along with the modern LSTM neural network, provided a balanced approach to predicting sentiment over different time horizons. Each model contributed unique insights into forecasting accuracy and performance.

Effectiveness of Models:

- Sentiment Analysis Models: The sentiment analysis models, including VADER, NLTK, and TextBlob, demonstrated proficiency in capturing the emotional tone of tweets. The integration of NLP libraries allowed for nuanced sentiment extraction, offering a more comprehensive understanding of user expressions.

- Time-Series Forecasting Models: ARIMA and Exponential Smoothing showcased effectiveness in capturing short-term sentiment trends, providing interpretable forecasts. The LSTM neural network, with its deep learning capabilities, exhibited its strength in capturing intricate dependencies within the sentiment time series.

Insights Gained:

- Temporal Dynamics: Understanding temporal dynamics in sentiment proved crucial. The ability to capture trends over time facilitated more informed decision-making in response to evolving public opinion.

- Event-Driven Sentiments: Correlating sentiment trends with external events highlighted the interconnectedness of online sentiments with real-world occurrences. This insight is valuable for businesses to align strategies with ongoing events.

Future Work and Improvements:

- Enhanced Sentiment Models: Future work could explore advanced sentiment analysis models, possibly incorporating more sophisticated deep learning architectures for improved sentiment understanding.

- Ensemble Forecasting: Combining forecasts from multiple models could potentially enhance the accuracy and robustness of sentiment predictions. Ensemble methods could be explored to leverage the strengths of different models.

- Real-Time Analysis: Extending the analysis to real-time sentiment monitoring would provide a more immediate understanding of emerging trends and sentiments, enabling timely responses.

In conclusion, the project successfully navigated the intricacies of sentiment analysis and time-series forecasting on a large Twitter dataset. The findings underscore the importance of considering both historical sentiments and external events in predicting future sentiment trends. The insights gained lay the foundation for more informed decision-making in various domains influenced by public sentiments on social media.

**REFERENCES**

<https://github.com/bhallimk/1_Sentiment_Analysis>

<https://github.com/bhallimk/time_series>

<https://github.com/bhallimk/time_series_2>

[1] A. Pak and P. Paroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining” in LREC, 2010, vol. 10.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. “Efficient Estimation of Word Representations in Vector Space.” In NIPS 2013

[3] Ashish Vaswani, Noam Shazier, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. “Attention Is All You Need.” In NIPS 2017

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen “Distributed Representations of Words and Phrases and their Compositionality” In NIPS, pages 3111–3119 in 2013.

[5] Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, in CoRR 24 May 2019.

[6] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” Journal of machine learning research, vol. 3, no. Feb, pp. 1137–1155, 2003

[7] Alec Go, Richa Bhayani, and Lei Huang.. “Twitter sentiment classification using distant supervision”. Technical report, Stanford 2009.

[8] Bo Pang and Lillian Lee, “Opinion mining and sentiment analysis”, in Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008) 1–135

[9] J. L. Elman, “Distributed representations, simple recurrent networks, and grammatical structure,” Machine learning, vol. 7, no. 2-3, pp. 195–225, 1991.

[10] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, “ Neural Machine Translation by Jointly Learning to Align and Translate”, 2016

[11] Mohammad Rezwanul Huq, Ahmad Ali “Sentiment Analysis on Twitter Data using KNN and SVM” in 2017

[12] Gallagher, E. Furey, and K. Curran, “The application of sentiment analysis and text analytics to customer experience reviews to understand what customers are really saying,” International Journal of Data Warehousing and Mining (IJDWM), vol. 15, no. 4, pp. 21–47, 2019.

[13] W. McKinney, J. Perktold, and S. Seabold, “Time series analysis in Python with statsmodels,” Jarrodmillman Com, pp. 96–102, 2011.

[14] Pal and P. K. S. Prakash, Practical time series analysis: master time-series data processing, visualization, and modeling using python. Packt Publishing Ltd, 2017.

[15] Borg and M. Boldt, “Using VADER sentiment and SVM for predicting customer response sentiment,” Expert Systems with Applications, vol. 162, p. 113746, 2020.