**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Advanced Data Analytics  Big Data Storage and Processing |
| **Assessment Title:** | twitter sentimization |
| **Lecturer Name:** | Dr. Muhammad Iqbal, David McQuaid |
| **Student Full Name:** | Goksel Sezgın |
| **Student Number:** | 2023419 |
| **Assessment Due Date:** | *17/04/2024* |
| **Date of Submission:** | *19/05/2024* |

**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. |

**Long-Term Time Series Analyses: Twitter Sentiment Analysis**

# **Abstract**

Modern data management has entered an era where old data storage and management approaches are increasingly inadequate. The limitations of legacy systems force a shift to innovative and scalable solutions to meet modern data requirements. This transition enables organizations to gain a competitive advantage by optimizing big data analysis processes. Features such as scalability, speed, security and cost effectiveness, which are especially required for big data, can be achieved with innovative and flexible solutions.

In this study, sentiment analysis was performed on a subset of tweet data using the VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analysis tool. VADER was chosen because it is particularly attuned to emotions expressed on social media and works well on short texts such as tweets.

In the first part of this study, a comparative analysis was made through CRUD operations using MySQL and MongoDB databases in another study conducted on the same data set. This analysis allowed both database systems to be evaluated in terms of performance, scalability and ease of management. The results have guided the selection of data management strategies, especially in large-scale and distributed systems.

In the second part, emotional scores of tweets were calculated by performing sentiment analysis from the texts in the data set. These scores were entered into the 'sentiment\_score' column, thus quantifying the degree to which each tweet was positive, negative or neutral. This analysis was accomplished through algorithms and techniques used to determine the emotional content of text-based data.

**CRUD Analysis: Examining the Performance of Database Systems**

CRUD analysis comprehensively evaluates the basic functions in database systems — Create, Read, Update and Delete operations. This analysis examines the key performance indicators of a database management system and provides detailed information about the system's efficiency, effectiveness, speed and usability. CRUD operations are at the heart of a database system, and optimization of each of these operations directly affects overall system performance.

**Table 1.** CRUD Analysis of MySQL and MongoDB Databases

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **MySQL** | **MongoDB** |
| **Create** | Code | data.to\_sql('info', con=engine, if\_exists='replace', index=False) | for index, row in data.iterrows():  document = { "id": row["id"],  "date": row["date"], "flag": row["flag"],  "user": row["user"], "text": row["text"] }  collection.insert\_one(document) |
| Time | 30 Seconds | 18 Hours (333.448 Data) |
| **Read** | Code | query = text("SELECT \* FROM info") | results = collection.find().limit(160000) for document in results:  print(document) |
| Time | 18 Seconds | 34 Seconds (160.000 Data) |
| **Update** | Code | update\_query = UPDATE info  SET id = id + 1  WHERE MOD(id, 2) = 1 | result = collection.update\_one({'\_id': document['\_id']}, {'$set': {'date': new\_date\_iso}}) |
| Time | 28 Seconds | 27 Seconds (160.000 Data) |
| **Delete** | Code | delete\_query = "DELETE FROM info" | delete\_result = collection.delete\_many({'\_id': {'$in': document\_ids}}) |
| Time | 13 Seconds | 31 Seconds (333.448 Data) |

**MongoDB Transaction Times and Efficiency**

1. **Creating**

Code Description: The command data.to\_sql('info', con=engine, if\_exists='replace', index=False) transfers a Pandas DataFrame object to the 'info' table in a MySQL database. This operation replaces the existing table with new data and opts not to include DataFrame indices in the table. Performance Evaluation: The approximately 30-second data writing process is effective and fast for transferring a large data set, such as 1.6 million tweets, to the database. While direct reading from a CSV format takes 3.2 seconds, the 30 seconds required for database reading remains acceptable when considering the volume of data and the processing capacity of the database server.

1. **Reading:**

Code Description: The query text("SELECT \* FROM info") retrieves all records from the 'info' table. Performance Evaluation: Reading 1.6 million records from the database in 18 seconds indicates that the database querying process is well-optimized and the data infrastructure is high-performing. This demonstrates effective process management for large-volume data readings.

1. **Update:**

Code Description: The command UPDATE info SET id = id + 1 WHERE MOD(id, 2) = 1 increments the IDs of all records with an odd ID by one, making them even. Performance Evaluation: Completing this bulk update in 28 seconds, considering the size of the data set, offers remarkable speed. This operation signifies the effective use of query optimization and database management systems.

1. **Deletion:**

Code Description: The command DELETE FROM info removes all data from the 'info' table. Performance Evaluation: Deleting all data in 13 seconds demonstrates that deletion operations are efficiently managed by the database. This duration is a significant advantage when large data sets need to be quickly cleaned. General Evaluation: The performance timings indicate that MySQL can effectively and swiftly manage a large data set containing 1.6 million tweets. The operations performed demonstrate that the database design and query optimization are well-planned and implemented to handle data at this scale successfully.

**MongoDB Transaction Times and Efficiency**

1. **Adding Data (Insert):**

Duration: 18 Hours (333,448 Data)

Evaluation: The fact that data addition takes so long may be due to the high amount of data being processed. However, 18 hours is quite a long time for 333,448 records. This may be due to inserting each document one by one with insert\_one.

1. **Data Reading (Read):**

Duration: 34 Seconds (160,000 Data)

Evaluation: Reading 160,000 records in 34 seconds shows that MongoDB is not very efficient. In fact, it can be considered as appropriate performance for an internet-based database. It can be improved with appropriate optimization techniques.

**3. Data Update:**

Duration: 27 Seconds (160,000 Data)

Compared to MySQL, this time is still not acceptable for such a large amount of data. However, MongoDB's update time can be improved with appropriate optimization techniques.

**4. Data Deletion (Deletion):**

Duration: 31 Seconds (333,448 Data)

Evaluation: The time required to delete such a large number of documents using MongoDB's delete\_many function is quite effective. This indicates that MongoDB can effectively manage large **amounts of data and provide high performance in deletion operations.**

**Data Deletion (Delete):**

Duration: 31 Seconds (333,448 Data)

Evaluation: The time required to delete such a large number of documents using MongoDB's delete\_many function is quite effective. This is an indication that MongoDB can effectively manage large amounts of data and provide high performance on deletions as well.

**General evaluation:**

MongoDB is suitable for working with large data sets, but data insertion times need to be optimized. Its performance for read, update and delete operations indicates an impressive capacity for data management and query answering. Using batch operations for inserts or reviewing the database configuration can improve overall performance.

1. Sentiment Analysis

Sentiment Analysis involves interpreting and classifying sentiments (positive, negative and neutral) found in text data using text analysis techniques. Sentiment analysis helps organizations measure public sentiment towards certain words or topics.

Sentiment Analysis has become a significant focus of research in recent years, serving as a method to categorize human emotions related to specific events, products, or services. It is a critical issue, particularly for organizations or businesses that wish to understand consumer perceptions of their offerings. With the growth of social media platforms, Sentiment Analysis has gained prominence as an intriguing field of study. On social media, individuals frequently share their feelings, whether joy or sorrow, daily. Consequently, the sheer volume of data now available renders traditional solutions inadequate, making the development of automated analysis techniques essential [1].

Natural Language Processing (NLP) combines the disciplines of computer science, artificial intelligence, and linguistics to develop computational models that can process and understand natural language. This includes enabling computers to recognize semantic expressions (for example, understanding the closer semantic relationship between 'cat' and 'dog' compared to 'cat' and 'spoon'), converting text to speech, translating languages, and much more [2].

**EDA (Exploratory Data Analysis )**

The chart titled Figure 1: Number of Hourly Tweets Viewed in a 15-Day Period visualizes the hourly tweet frequency for fifteen days. Each point in this time series graph corresponds to the volume of tweets sent at a particular hour. Visible fluctuations with peaks and valleys on the chart reflect the variation in tweeting activity throughout the day for the analyzed period.

Peaks in the chart represent periods of increased tweeting activity, typically ranging from 2,000 to 5,000 tweets per hour, and indicate times when users are more active on the platform. Conversely, lower points indicate quieter hours with fewer tweets. The presence of significant gaps in the data set may indicate missing data or periods of exceptionally low activity.

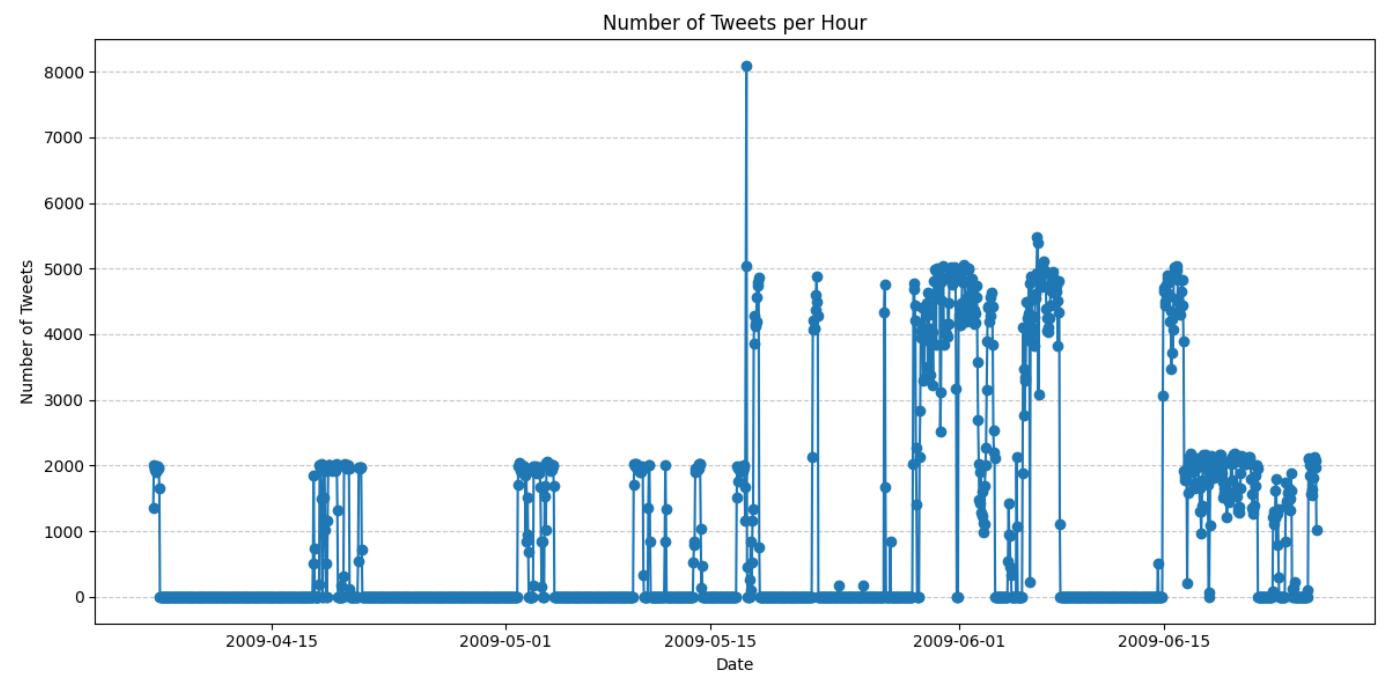


Fig 1. Number of tweets sent on an hourly basis, displayed over a 15-day period

Figure 1 shows the content of the dataset and sentiment scores. The dataset includes the timestamp ('date'), a unique identifier ('id'), the status flag ('flag'), the user's identifier ('user'), the tweet content ('text') and a calculated 'sentiment\_score'. The timestamp of each tweet is recorded with precision down to the second, allowing for accurate chronological analysis.

The "sentiment\_score" assigned to each tweet is a numerical value that measures the emotional value of the text. This score is derived from sentiment analysis, a computational process that evaluates the emotional tone behind a set of words. Scores in this dataset range from -1 to 1; Here, negative values represent negative emotions, positive values represent positive emotions, and scores around zero represent neutral sentiment.



Fig 2. Calculated Sentiment Scores

Figure 2, titled “Number of Tweets per Hour,” visually shows the distribution of tweet volume by different hours of the day. The x-axis of the chart represents the hours from 0 (midnight) to 23 (23:00) in a day, and the y-axis represents the number of tweets posted in each hour.

It is observed from the graph that tweet activity varies significantly throughout the day. Lower tweet volumes are evident in the early morning hours and gradually increase as the day progresses. There is a noticeable peak in the late evening hours, indicating that this is when Twitter users are most active. The highest tweet volumes occur between 9pm and 11pm, indicating a potential prime time for user engagement on the platform.

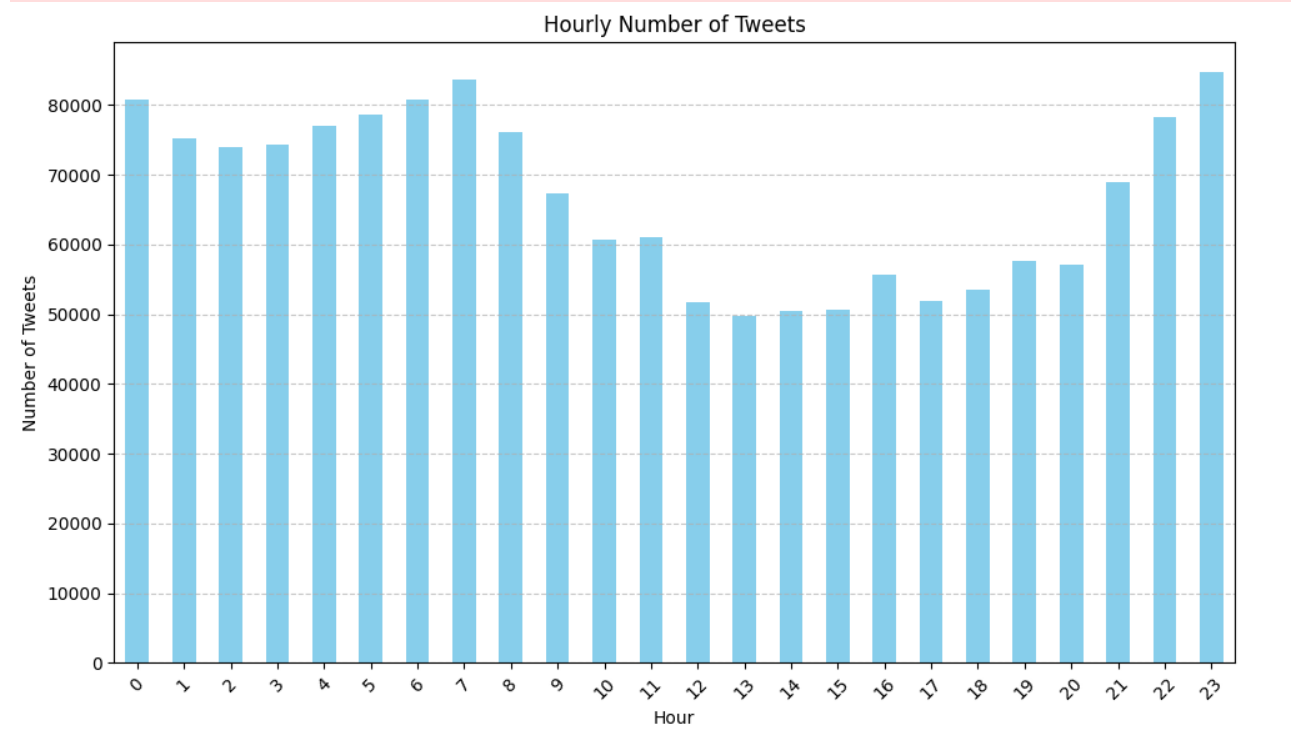
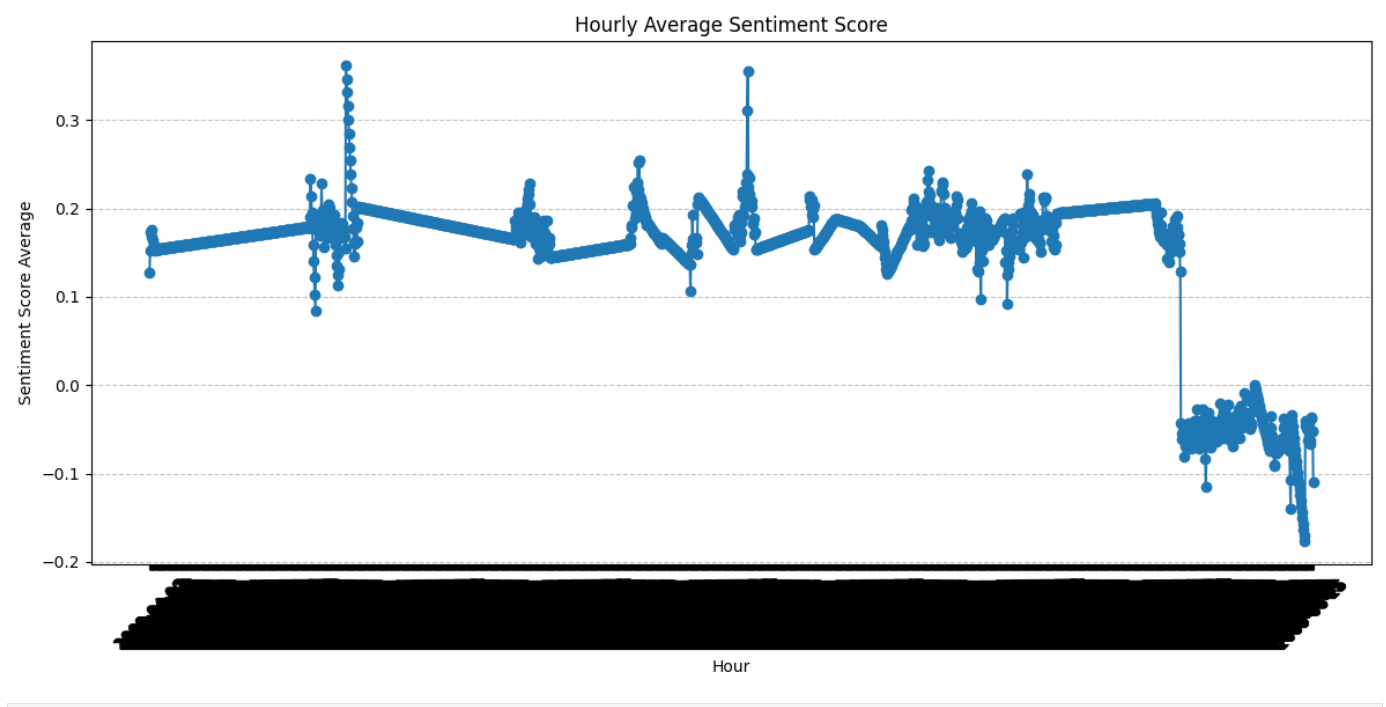


Fig 3. Hourly Number of Tweets

Figure 3, titled "Average Hourly Sentiment Score Across Time Series" provides an overview of the average sentiment scores of tweets across a range of all dataset hours, shown along the x-axis. The y-axis represents sentiment scores, ranging from negative values indicative of negative emotions to positive values indicative of positive emotions.

This visualization shows fluctuations in tweet sentiment scores across various days. The overall trend shows that sentiment is mostly positive, with most scores above zero. However, there are also noticeable differences, such as sharp drops in sentiment on certain days and more negative sentiment reflected in tweets.

Towards the end of the chart, there is a noticeable drop in sentiment scores, almost touching the lower ends of the scale. This could indicate a period of extremely negative emotions or potential problems with data integrity or collection methods these days.



**Fig.4** Average Hourly Sentiment Score Across Time Series"Trend Analizi

**Dealing with Short-Term Time Series**

In 2014, Huanfei and colleagues developed a methodology to address short-term time series. Their study proposed a method for transforming information from multivariate interactions into the time domain, thereby converting short-term high-dimensional time series into corresponding longer, lower-dimensional time series. The researchers indicated that their proposed method effectively overcame limitations due to the lack of temporal information in short-term datasets [4].

However, when the method proposed by Huanfei and his team was applied to a Twitter dataset, it did not yield successful results. The objective of "transforming information from multivariate interactions into the time domain" could not be realized in this application.

On another note, Calver (1996) introduced a study that employed a simple flow modeling approach to incrementally decompose data from daily to hourly time intervals. Furthermore, a study by Güntner and colleagues in 2001 focused on converting daily time series into hourly resolutions, thereby extending the time series. These studies demonstrated that the scale-invariant assumptions of the model used provided similarly satisfactory results for both datasets examined [5, 6].

**Trend Analysis**

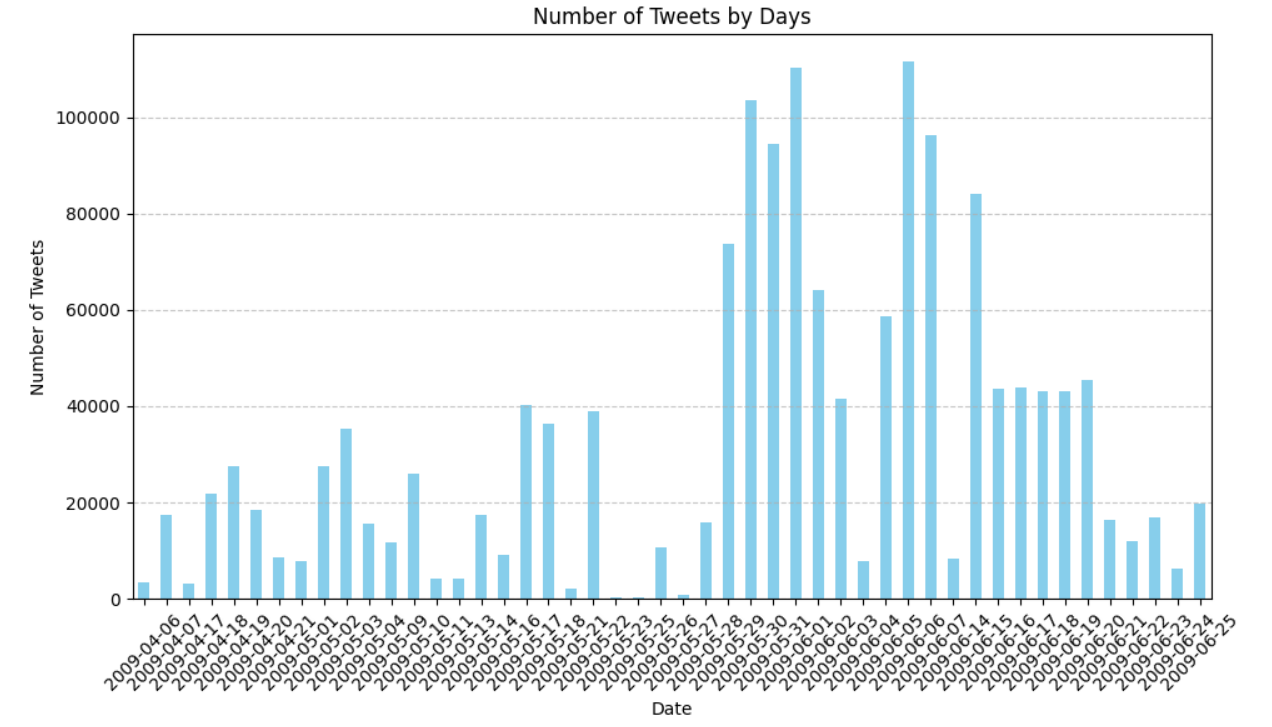
We used the following methods to determine whether there were trends in the data set.

**1. Visual Check**

The presence of a trend can be checked by visually examining the data set with a time series chart. A trend usually appears as a smooth long-term increase or decrease in the data set.

When Figure 5 is examined, it can be seen that there is no visible trend in the frequency of tweets during the observed period. The fluctuations observed in the data do not indicate a consistent upward or downward trend. Instead, differences in tweet volume are significant; This shows that the number of tweets can be quite variable on a daily basis.

This variability is potentially attributable to the impact of certain events or news items that temporarily impact user engagement on the platform. On certain days, tweet volumes exceed 100,000; This indicates high user activity or interest, possibly resulting from major news events or social media trends. Conversely, on other days, tweet volume drops below 20,000, indicating less engagement or a lack of engaging content that would drive user engagement.



**Fig 5.** Number of tweets sent by day.

**2. Augmented Dickey-Fuller (ADF) Testi:**

ADF test is used to test the existence of unit root of a time series. Unit root means that a time series is not constant in a statistical sense. If there is no unit root in the series, this may indicate the existence of a trend [3].

**ADF Test Results Evaluation**

The Augmented Dickey-Fuller (ADF) test applied to the Sentiment Analysis data provided significant evidence concerning the stationarity of the series. The test statistic calculated was -13.81, which is substantially lower than the critical values at 1% (-3.43), 5% (-2.86), and 10% (-2.56) confidence levels. This substantial difference indicates a strong rejection of the null hypothesis, suggesting that the time series is indeed stationary.

The exceedingly small p-value of \(8.02 \times 10^{-26}\) further supports this conclusion, denoting an extremely low probability that the observed results are due to chance. This level of significance implies that the series does not exhibit unit root characteristics and thus does not have a time-dependent structure that would make it non-stationary.

These results are critical as they validate the appropriateness of using techniques that assume stationarity in subsequent analyses of the Sentiment Analysis data. The absence of non-stationarity implies that the properties of the series such as mean and variance are constant over time, which is beneficial for modeling and forecasting purposes. This stationarity assumption can simplify the modeling process and increase the reliability of any inferences made from statistical models applied to the data.

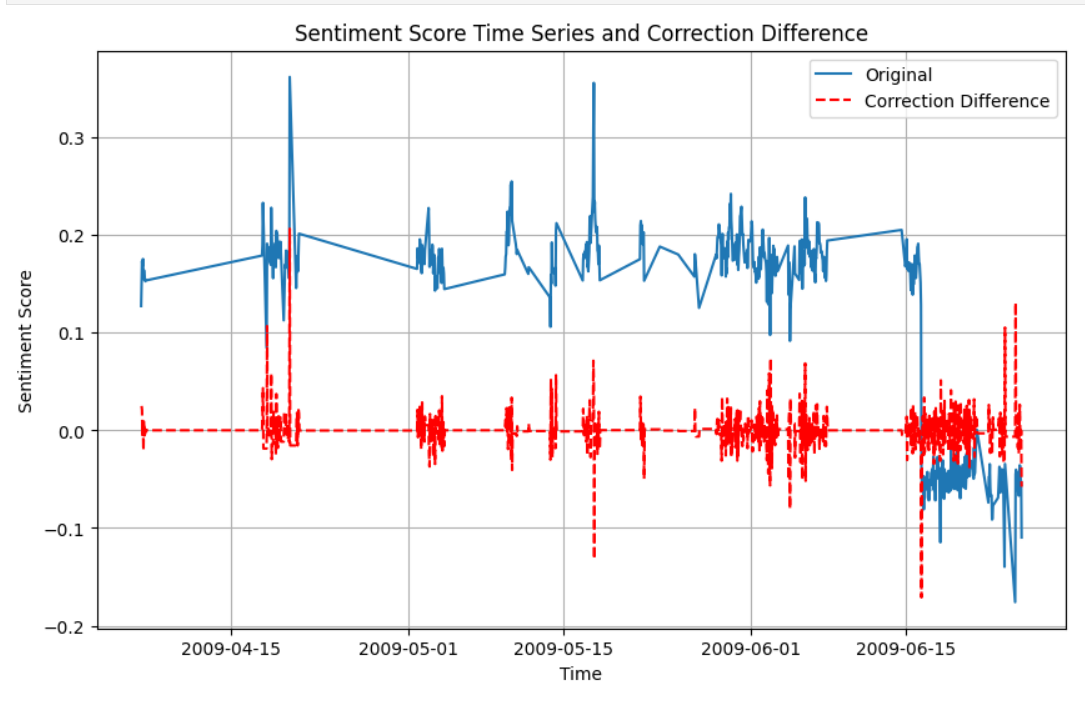
****

Fig 6. Sentiment Score and Correction Difference

Figure 6 provides a graphical representation of sentiment scores over time, shown along with a correction difference. The blue line shows the original sentiment scores, which indicate the emotional tone of the tweets on a scale ranging from -0.1 to 0.3. This range indicates fluctuations between mildly negative and moderately positive emotions during the observed period.

The red dashed line represents the "correction difference", which appears to indicate adjustments or anomalies in the sentiment scoring, accounting for errors or external factors affecting the sentiment analysis. These adjustments are mostly negative, suggesting a systematic adjustment downward from the original sentiment scores.

Significant spikes and declines on both lines indicate moments of high volatility, possibly in response to specific events or news that cause bursts of positive or negative emotion. For example, notable peaks in sentiment may correspond to positive news or events, while troughs may relate to negative events.

**Evaluation and Justification of Hyperparameter Tuning Techniques Used**

**LSTM**

Various configurations were tried during the process of designing the model and determining its parameters, and the parameter combination that achieved the lowest error rate was found in this model. Comparisons between the model's final configuration and other tested parameters are summarized below.

**Model Configuration and Selection**

• Number of Cells: 50 LSTM cells are used in the model. Other trials have tested higher cell numbers, such as 100 and 150 cells. However, in these cases, the training time of the model increased significantly and overfitting problems were observed. In models with fewer cells (e.g. 20 and 30 cells), insufficient learning and higher error rates were observed.

• Activation Function: relu activation function was used because this function helped to train the model quickly and effectively. Other activation functions such as tanh and sigmoid have also been tried; However, slower training processes and higher error rates were achieved compared to relu.

• A single density layer was used and this layer served directly as the output layer. Adding multiple density layers during experiments increased the complexity of the model and again led to overfitting.

• Batch Size: 1 was used as the batch size in training the model. Although larger batch sizes (e.g. 10 or 20) offered faster training processes, this negatively affected the stability of the gradient and the overall performance of the model.

In addition to these parameters, various experiments have been carried out on other hyperparameters such as learning rate and loss function. In this process, man optimization algorithm and mse (mean square error) loss function were determined as the structure that gave the best results. All these tests and comparisons showed that the current model configuration is the optimal solution for temporal emotion score predictions.

**Model Configuration and Selection**

**ARIMA**

ARIMA(2,1,2)

This model is a very flexible and powerful model used in time series analysis. This model often includes autoregressive (AR) terms, differencing (I) processes, and moving average (MA) terms. In the ARIMA(2,1,2) specification, p=2, d=1 and q=2 parameters are used. The meanings of the parameters used in this model:

AR Term (p=2): The autoregressive term expresses the dependence of the model on two past data points (`p=2`). This means that the values in the previous two time periods affect the current value. This is an important feature for continuously changing time series such as sentiment analysis, where previous sentiments can influence the next.

I (d=1): The integrated component is the differencing process required to make the non-stationary structure of the data stationary. d=1 means taking the data to first difference; That is, the differences between the current values and the previous values are taken. This is used to eliminate trends or random walk features in the series.

MA Term (q=2): The moving average term expresses the dependence of the error terms of the model on the two past values. This helps model how random shocks or noise in the series affect the last two time periods. This feature is useful for capturing the short-term effects of unexpected events or news on sentiment scores.

The use of the ARIMA(2,1,2) model offers the lowest error rates among other model configurations tried during the analysis process. One of the main factors in choosing this model is that the model successfully captures the structural features of the data set and produces effective predictions. Using these parameters in the model means that it best represents the structural information and predictability of the data set and produces reliable predictions by minimizing error rates.

As a result of this comprehensive analysis and testing process, it was decided to continue with the ARIMA(2,1,2) model. This decision is based on the model's ability to make effective predictions using historical information in the time series and its success in sentiment analysis.

**Table 1.** A time series forecast of the sentiment of the entire dataset after 1 day, 3 days and 7 days

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Day 1 | Day 3 | | | Day 7 | | | | | | |
| Arima | -0.0007 | -0.0004 | -0.0002 | 0 | -0.0004 | 0 | 0 | 0 | 0 | 0.0002 | -0.0007 |
| ANN |  | -0.1174 | -0.1445 | -0.0402 | 0.05 | 0.1441 | 0.1405 | 0.1441 | 0.1441 | 0.1441 | 0.1441 |

Table 2 presents the results of the sentiment analysis on Twitter using ARIMA and Artificial Neural Networks (ANN) models for 1, 3 and 7 day periods. The main points to consider when interpreting these results may be:

**ARIMA Model:**

Day 1: Sentiment score shows little change. This may indicate a generally stable emotional state.

Day 3: There is a small increase in emotion score. Perhaps an event or trend triggered this change.

7 Days: Emotion score remained almost constant or changed very little. The ARIMA model predicts that emotions will remain stable in the long run.

**Artificial Neural Networks (ANN):**

Day 1: There is a significant decrease in the emotion score. This can perhaps be attributed to the impact of negative news.

Day 3: A significant improvement is seen in emotion scores. A positive event or news may have triggered this change.

Day 7: Emotion scores are rising again. ANN predicts that the increase in emotional reactions will continue.

**General evaluation:**

Reactivity: The ANN model seems to react more quickly and noticeably to certain events or news. The ARIMA model, on the other hand, offers a more stable forecast.

Consequently, the results of both models provide valuable information for understanding specific susceptibility situations and possible influencing factors.

**Source**

[1] Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang. 2011. Topic Sentiment Analysis in Twitter: A Graph-based Hashtag Sentiment Classication Approach. In ACM International Conference on Information and Knowledge Management (CIKM). 1031–1040.

[2] Dritsas, E., Livieris, I. E., Giotopoulos, K., & Theodorakopoulos, L. (2018, November). An apache spark implementation for graph-based hashtag sentiment classification on twitter. In *Proceedings of the 22nd Pan-Hellenic Conference on Informatics* (pp. 255-260).

[3] Baum, C. F., & Otero, J. (2021). Unit-root tests for explosive behavior. *The Stata Journal*, *21*(4), 999-1020.

[4] Ma, Huanfei, et al. "Predicting time series from short-term high-dimensional data." *International Journal of Bifurcation and Chaos* 24.12 (2014): 1430033.

[5] Calver, A. 1996. Development and experience of the Tate rainfallrunoff model. Proc. Inst. Civil Engrs: Water, Maritime and Energy, 118, 168–176.

[6] Güntner, Andreas, et al. "Cascade-based disaggregation of continuous rainfall time series: the influence of climate." *Hydrology and Earth System Sciences* 5.2 (2001): 145-164.