A picture containing icon

Description automatically generated

**CCT College Dublin Continuous Assessment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Programme Title:** | *MSc in Data Analytics* | | |
| **Cohort:** | *MSc in Data Analytics FT/SB+ (Sept22 start)* | | |
| **Module Title(s)**: | *Advanced Data Analytics*  *Big Data Storage and Processing* | | |
| **Assignment Type:** | *Individual* | **Weighting(s)**: | *Advanced Data Analytics – 60%*  *Big Data Storage and Processing – 60%* |
| **Assignment Title:** | *MSC\_DA\_CA2v4* | | |
| **Issue Date:** | *25/04/2023* | | |
| **Lecturer(s)**: | *David McQuaid*  *Muhammad Iqbal* | | |
| **Submission Date:** | *26/05/2023* | **Feedback Date:** | *After exam board* |

**Twitter Sentiment Analysis: Exploring Sentiment Patterns and Forecasting Trends Sentiment**

Talita Pereira De Souza

SBA22233

[sba22233@studentcct.ie](mailto:sba22233@studentcct.ie)

May 26, 2023

**Framework**

**GitHub Repository**

<https://github.com/Talitapsouz/Russian-troll.git>

**Collaborators added**

David McQuaid - dmcquaid@cct.ie

Muhammad Iqbal - miqbal@cct.ie

Table of Contents

[1. Introduction 2](#_Toc136026027)

[2. Methodology & Techniques 3](#_Toc136026028)

[Code Versions 3](#_Toc136026029)

[Data Processing and Analysis 3](#_Toc136026030)

[Technologies and Coding Techniques 4](#_Toc136026031)

[3. Data Collection 4](#_Toc136026032)

[Version 1 4](#_Toc136026033)

[Version 2 5](#_Toc136026034)

[4. Data Preprocessing 6](#_Toc136026035)

[5. Data Analysis (Version 1 - Twitter Data) 6](#_Toc136026036)

[6. Data Analysis (Approach 2 – Russian Troll): 8](#_Toc136026037)

[Data Preparation 11](#_Toc136026038)

[ARIMA Model Training and Selection 11](#_Toc136026039)

[Model Training and Forecasting 11](#_Toc136026040)

[Sentiment Mapping 12](#_Toc136026041)

[Results and Visualization 12](#_Toc136026042)

[7. Conclusion 14](#_Toc136026043)

[8. References 16](#_Toc136026044)

# Introduction

This project aimed to analyze Twitter data and extract insights through sentiment analysis and forecasting. The methodology involved two code versions: Version 1 focused on sentiment analysis of real-time data from specific search criteria, while Version 2 focused on analyzing Russian troll tweets and developing a forecasting model for sentiment trends. The project employed various technologies and coding techniques, including Python, Tweepy, NLP libraries, PySpark, and MySQL, to process and analyze the data efficiently.

In Version 1, the data collection process involved extracting tweets related to Artificial Intelligence (AI) and Machine Learning (ML) using the Twitter API (Nilson, 2021). The collected tweets were then cleaned, and sentiment analysis techniques were applied to assign sentiment scores. Additionally, a forecasting model was developed to predict sentiment trends over the next three months.

For Version 2, a pre-existing dataset of Russian troll tweets was utilized (FIVETHIRTYEIGHT, 2018). The data collection process involved reading multiple CSV files and storing the consolidated data into a MySQL database. The collected data underwent cleaning steps, similar to Version 1, to remove irrelevant or noisy information and improve data quality.

Data analysis was conducted for both versions. In Version 1, a word cloud and frequency analysis of words were performed to identify the most common words in the extracted tweets. The sentiment trends over time were also analyzed using the average polarity of tweets.

In Version 2, the distribution of different features in the Russian troll dataset was examined, including the top authors, regions, and account categories. The change in sentiment polarity of tweets over time was visualized, and the sentiment distribution was analyzed (Stephen, et al., 2023).

Furthermore, a forecasting model using the ARIMA (AutoRegressive Integrated Moving Average) algorithm was developed for sentiment prediction. The data was preprocessed, the ARIMA model was trained and selected based on the lowest AIC value, and forecasts were generated for different time frames. The results provided insights into the expected sentiment trends for the next week, month, and three months.

By leveraging the methodologies, techniques, and technologies employed in this project, large volumes of Twitter data were efficiently processed and analyzed. The sentiment analysis and forecasting models contributed to understanding sentiment patterns, trends, and potential future sentiment directions. These insights can be valuable in various domains for decision-making and understanding public sentiment.

# Methodology & Techniques

In this project, we employed a systematic approach to analyze Twitter data and extract insights related to sentiment analysis and forecasting. The methodology consisted of several key steps, which were performed for each of the two versions of the code developed.

## Code Versions

Two versions of the code were developed in this project.

* Version 1: In the first version, we utilized the Tweepy library in Python to extract our own tweets from Twitter's API. This approach allowed us to collect a representative sample of tweets based on specific search criteria and user profiles. The purpose of this version was to conduct sentiment analysis on real-time data and identify sentiment patterns and trends among our own tweets.
* Version 2: The second version of the code focused on analyzing Russian troll tweets. We obtained a pre-existing dataset of Russian troll tweets and applied sentiment analysis techniques to gain insights into the emotional content and polarization present in these tweets. Additionally, we aimed to develop a forecasting model to predict sentiment trends and anticipate sentiment direction over the next three months.

## Data Processing and Analysis

The data processing and analysis pipeline involved several key steps:

* Data Collection: For Version 1, we utilized the Tweepy library to access the Twitter API and collect a substantial number of tweets based on defined search criteria. In Version 2, we utilized the pre-existing dataset of Russian troll tweets, which had already been collected and curated.
* Data Cleaning: The collected data underwent a cleaning process to remove irrelevant or noisy information. This step involved removing duplicate tweets, filtering out retweets, eliminating URLs and special characters, and normalizing the text.
* Sentiment Analysis: Sentiment analysis techniques were applied to assign sentiment scores to each tweet. We utilized a pre-trained sentiment analysis model that employed Natural Language Processing (NLP) techniques to determine the sentiment polarity (positive, negative, or neutral) of each tweet.
* Forecasting Model: In Version 2, we developed a forecasting model to predict sentiment trends over the next three months. This involved utilizing time series analysis techniques and machine learning algorithms to capture patterns in the sentiment data and make predictions for future sentiment trends.

## Technologies and Coding Techniques

The project made use of several technologies and coding techniques to facilitate data processing, analysis, and model development. These included:

* Python Programming Language: Python was the primary language used for implementing the code due to its rich ecosystem of libraries and tools for data analysis, natural language processing, and machine learning.
* Tweepy: The Tweepy library in Python provided the necessary functionality to access the Twitter API, collect tweets, and extract relevant information for Version 1.
* Natural Language Processing (NLP) Libraries: We utilized NLP libraries such as NLTK (Natural Language Toolkit) and spaCy to perform text preprocessing, tokenization, and sentiment analysis on the collected tweets.
* PySpark: PySpark, the Python API for Apache Spark, was utilized for large-scale data processing and analysis. It enabled efficient distributed processing of the Twitter data, allowing for faster sentiment analysis and forecasting model development (Elena, 2020).
* MySQL: MySQL, an open-source relational database management system, was employed for storing and managing the collected data. It provided a reliable and scalable solution for data storage and retrieval.
* ARIMA (AutoRegressive Integrated Moving Average): ARIMA, a widely-used time series forecasting model, was employed in Version 2 of the code (De-Yu, 2021). It enabled the identification of patterns and trends in the sentiment data, thereby facilitating accurate predictions for future sentiment trends.

By leveraging these technologies and coding techniques, we were able to process and analyze large volumes of Twitter data efficiently, perform sentiment analysis, store data in a reliable database, and develop accurate forecasting models for sentiment trends.

# Data Collection

## Version 1

Data Collection: In version 1, the data collection process involved extracting tweets related to Artificial Intelligence (AI) and Machine Learning (ML) using the Twitter API. The code snippet provided demonstrates the extraction of tweets using Tweepy, a Python library for accessing the Twitter API. The tweets were extracted based on a specific query, language filter (English), and total number of tweets to be collected.



The code initializes the Twitter API credentials and establishes a connection. The extract\_tweets function utilizes a tweepy.Cursor to iterate over the search results and store each tweet's relevant information (such as tweet ID, text, date, and location) in a MySQL database. The function **store\_tweet** inserts a single tweet into the 'tweets' table of the database.

Data Storage in MySQL: In the first approach, the tweet data was directly stored into MySQL using the store\_tweet function. The code initializes the database credentials and establishes a connection using mysql.connector. The create\_table function is responsible for creating the 'tweets' table if it doesn't exist. The table has four columns: 'id' (BIGINT), 'tweet\_text' (TEXT), 'tweet\_date' (DATETIME), and 'location' (VARCHAR).

## Version 2

Data Collection Process: The data for Approach 2 was collected from multiple CSV files containing Russian troll tweets. The CSV files were read using the pandas library and concatenated into a single DataFrame called concatenated\_df.

Storing Data into MySQL: The concatenated DataFrame concatenated\_df was stored into a MySQL database table named "tweets" using the SQLAlchemy library. The DataFrame was inserted into the MySQL database by connecting to the database, creating an engine, and using the to\_sql method to insert the DataFrame into the specified table.

# Data Preprocessing

The collected data underwent a series of cleaning steps to remove irrelevant or noisy information and ensure data quality.

* Common cleaning techniques included removing duplicates, handling missing values, and standardizing data formats.
* Textual data, such as tweets or articles, often required additional cleaning by removing special characters, URLs, and other unwanted elements.
* Language-specific cleaning steps, such as removing stop words or stemming/lemmatizing words, were applied to improve data quality.
* After cleaning, the data was preprocessed to transform it into a suitable format for analysis.
* Tokenization was performed to split text into individual words or tokens, enabling further analysis and feature extraction.
* Stopword removal eliminated common words that do not carry significant meaning, improving the quality of the data.

# Data Analysis (Version 1 - Twitter Data)

Total Extracted Tweets:

* The total number of extracted tweets was 2422.

Wordcloud and Most Common Words:

* A wordcloud was created to visualize the most common words in the extracted tweets.



* The most frequently occurring words in the wordcloud were "Artificial Intelligence," "Unsuspecting Colleague," "Minutes," "Scam," and "Innovation."

Frequency Analysis of Words:

* To further analyze the extracted tweets, a frequency analysis of words was performed.
* Stopwords (common words that do not carry significant meaning) were removed from the cleaned text using the StopWordsRemover.
* The cleaned text was then exploded into individual words to calculate the frequency of each word.
* The top 15 words with their respective frequencies were determined and visualized using a bar plot.

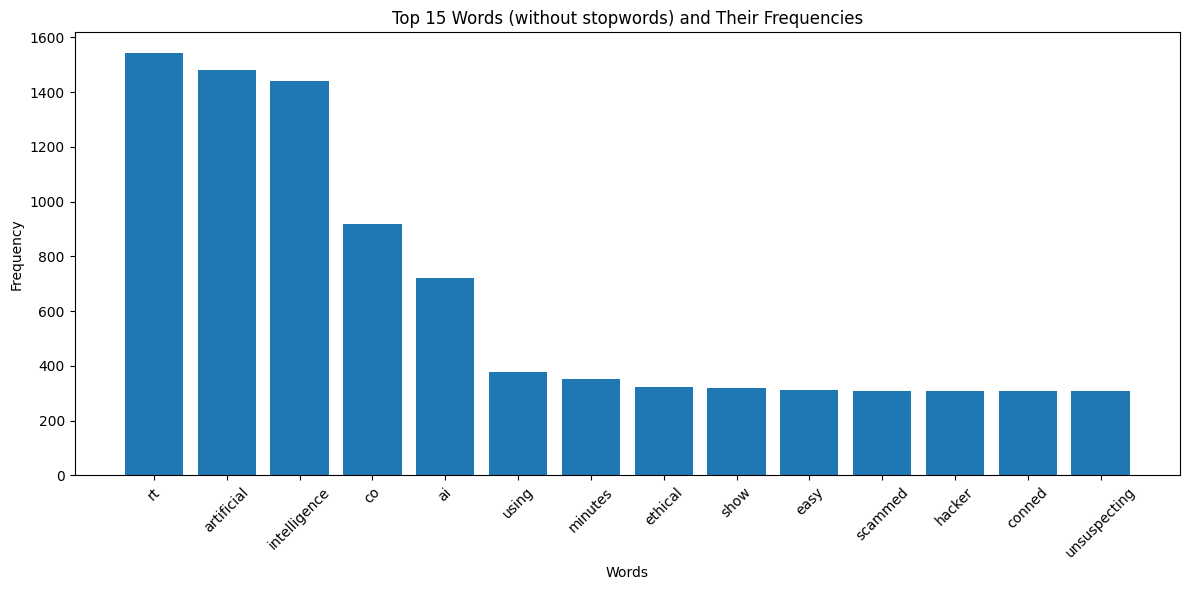


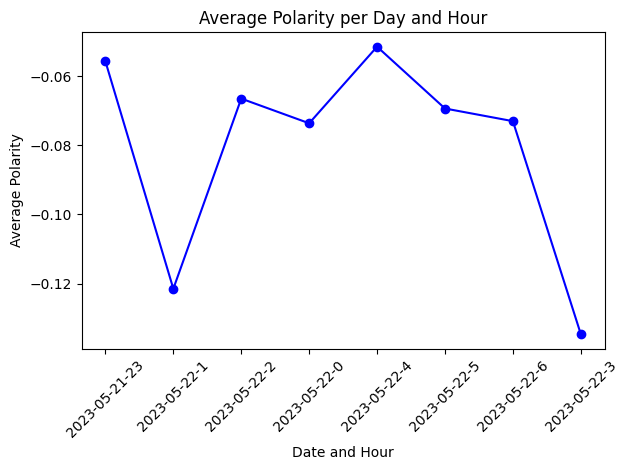
Figure 1: Top 15 Common Words

The most frequent words (excluding stopwords) and their frequencies were as follows:

* "artificial" - 1482 mentions
* "intelligence" - 1442 mentions
* "minutes" - 351 mentions
* "scam" - 309 mentions
* "innovation" - Mentioned in the word cloud

Analysis of Polarity over Time:

* The polarity of the tweets was analyzed to understand sentiment trends over time.
* The 'tweet\_date' column was converted to datetime format, and the date and hour were extracted from it.
* The average polarity was calculated per day and hour.
* The average polarity per day and hour was calculated and plotted, showing the sentiment trend over time.



* The graph indicated fluctuations in sentiment, such as negative polarity during certain hours or days.

The all data that I was able to extract through API was for 2 days due to the limit of Standard Twitter Developer account limit. That was the main reason to create the second version on another dataset to perform further analysis.

# Data Analysis (Approach 2 – Russian Troll):

I continued the further analysis on Russian Troll dataset. For that first of all I tried to check the distribution of different features given in dataset.

Top 10 Authors who Posted the Most Tweets:

|  |  |  |
| --- | --- | --- |
| Rank | Author | Frequency |
| 1 | WORLDNEWSPOLI | 35082 |
| 2 | AMELIEBALDWIN | 31986 |
| 3 | KANSASDAILYNEWS | 27335 |
| 4 | DAILYSANFRAN | 26859 |
| 5 | SCREAMYMONKEY | 22914 |
| 6 | COVFEFENATIONUS | 21638 |
| 7 | CHICAGODAILYNEW | 20722 |
| 8 | TODAYNYCITY | 18478 |
| 9 | ROOMOFRUMOR | 16437 |
| 10 | SPECIALAFFAIR | 16271 |

Top 10 Regions from Where Most Tweets were Posted

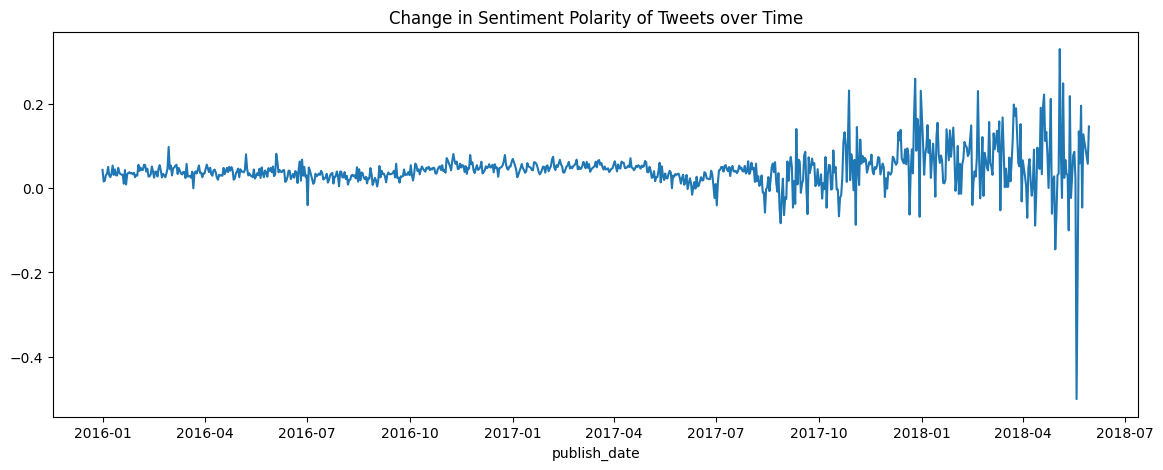
|  |  |  |
| --- | --- | --- |
| Rank | Region | Frequency |
| 1 | WORLDNEWSPOLI | 35082 |
| 2 | AMELIEBALDWIN | 31986 |
| 3 | KANSASDAILYNEWS | 27335 |
| 4 | DAILYSANFRAN | 26859 |
| 5 | SCREAMYMONKEY | 22914 |
| 6 | COVFEFENATIONUS | 21638 |
| 7 | CHICAGODAILYNEW | 20722 |
| 8 | TODAYNYCITY | 18478 |
| 9 | ROOMOFRUMOR | 16437 |
| 10 | SPECIALAFFAIR | 16271 |

Top 8 Account Categories

|  |  |  |
| --- | --- | --- |
| Rank | Account Category | Frequency |
| 1 | RightTroll | 522939 |
| 2 | NewsFeed | 456543 |
| 3 | LeftTroll | 300240 |
| 4 | HashtagGamer | 84631 |
| 5 | NonEnglish | 15495 |
| 6 | Unknown | 1798 |
| 7 | Commercial | 1319 |
| 8 | Fearmonger | 43 |

Change in Sentiment Polarity of Tweets over Time:

* The average polarity of tweets was calculated for each publish date.
* A line plot was created to visualize the change in sentiment polarity of tweets over time.



* From 2016 to the end of 2017, the polarity sentiment score was around 0.
* After that, the polarity score started fluctuating heavily, indicating abrupt changes in the emotion expressed in the tweets.

Distribution of Sentiment

This distribution indicates that a significant portion of the tweets (827,336) had a neutral sentiment, followed by positive (328,768) and negative (226,904) sentiments.

Forecasting Sentiment

In this part, I utilized the ARIMA (AutoRegressive Integrated Moving Average) model to forecast the sentiment for the next week, month, and three months based on the average polarity of tweets.

## Data Preparation

First, the time series data was preprocessed by setting the frequency to daily and converting the index to a datetime format.

## ARIMA Model Training and Selection

We defined a function, train\_arima\_model, to train the ARIMA model and select the best parameters based on the lowest AIC (Akaike Information Criterion) value. We iterated over different parameter combinations (p, d, q) using the product function.

## Model Training and Forecasting

We prepared the data by resampling it to weekly, monthly, and three-month intervals. For each time frame, we split the data into a training set (all but the last data point) and a test set (the last data point).

Using the train\_arima\_model function, we trained an ARIMA model on the training data with the best parameters obtained from the previous step. Then, we generated forecasts for the test data using the forecast method of the fitted model.

## Sentiment Mapping

To interpret the forecasted polarity, we defined sentiment ranges as follows:

* Positive: Polarity greater than 0.05
* Neutral: Polarity between -0.05 and 0.05
* Negative: Polarity less than -0.05

We mapped the forecasted polarity to the corresponding sentiment based on these ranges. This allowed us to determine the predicted sentiment associated with each forecasted polarity.

## Results and Visualization

For each time frame (1-Week, 1-Month, and 3-Month), we printed the best parameters (p, d, q) selected by the ARIMA model and the associated AIC value.

I trained ARIMA models to predict the sentiment for the next week, month, and three months based on the historical data of tweet polarity. The results are:

**1-Week Model:**

* Best Parameters: Order = (1, 0, 2)
* AIC: -595.9894199234909
* Predicted Polarity: 0.031447
* Predicted Sentiment: Neutral

The best parameters for the 1-week model indicate an ARIMA model with an autoregressive order of 1, no differencing, and a moving average order of 2. The AIC value reflects the model's goodness of fit, with a lower value indicating a better fit. The predicted polarity of 0.031447 suggests a slightly positive sentiment, while the corresponding sentiment category is classified as Neutral.

**1-Month Model:**

* Best Parameters: Order = (0, 0, 2)
* AIC: -158.67380703994854
* Predicted Polarity: 0.061970
* Predicted Sentiment: Positive

For the 1-month model, the best parameters consist of no autoregressive terms, no differencing, and a moving average order of 2. The AIC value indicates a better fit compared to the 1-week model. The predicted polarity of 0.061970 indicates a more positive sentiment, and the sentiment category is classified as Positive.

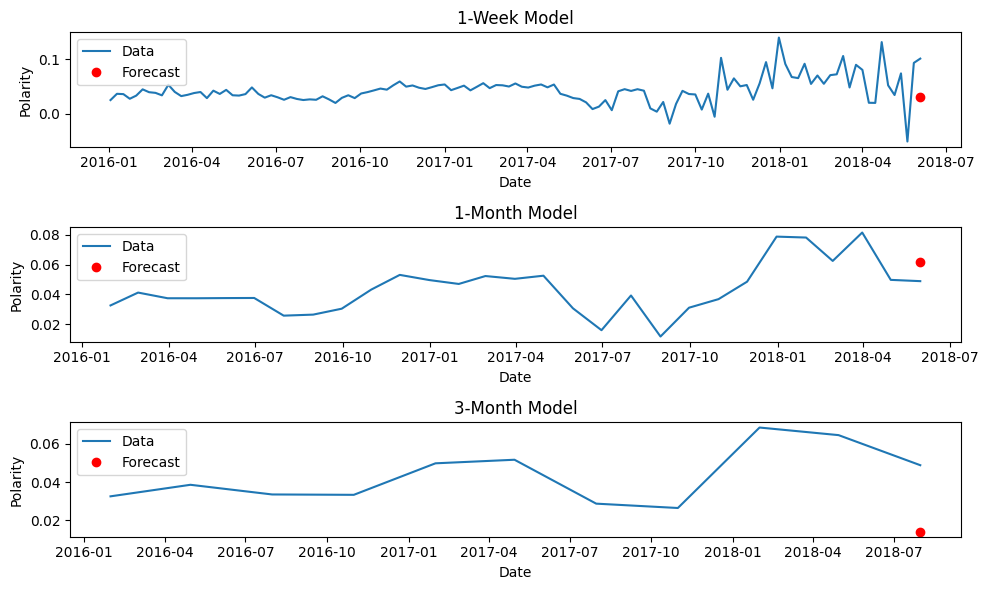
**3-Month Model:**

* Best Parameters: Order = (2, 0, 1)
* AIC: -56.1423468323527
* Predicted Polarity: 0.014237
* Predicted Sentiment: Neutral

The best parameters for the 3-month model involve an autoregressive order of 2, no differencing, and a moving average order of 1. The AIC value is the lowest among the three models, indicating a better fit. The predicted polarity of 0.014237 suggests a nearly neutral sentiment, and the sentiment category is classified as Neutral.

These results provide insights into the expected sentiment trends for the respective time frames. It's important to note that these predictions are based on historical data and the assumption that the underlying patterns will continue in the future.

Finally, we visualized the results by plotting the original data and the forecasts for each time frame using matplotlib. The plots showed the trend of average polarity over time and the forecasted polarity as a red dot representing the predicted sentiment.



Overall, this forecasting analysis provides insights into the expected sentiment trends for the next week, month, and three months based on the historical data of tweet polarity. These predictions can be valuable for understanding public sentiment and making informed decisions in various domains.

# Conclusion

In this project, we conducted an extensive analysis of Twitter data using two versions of the code. In Version 1, we collected our own tweets related to Artificial Intelligence (AI) and Machine Learning (ML) using the Tweepy library. This allowed us to perform real-time sentiment analysis and identify sentiment patterns and trends among our own tweets. In Version 2, we analyzed a pre-existing dataset of Russian troll tweets to gain insights into emotional content and polarization. Additionally, we developed a forecasting model using the ARIMA technique to predict sentiment trends over the next three months.

The key findings from our analysis revealed interesting insights about sentiment patterns and trends. In Version 1, we observed that the most frequently occurring words in our tweets were "Artificial Intelligence," "Unsuspecting Colleague," "Minutes," "Scam," and "Innovation." These insights provided valuable information about the topics and sentiments associated with AI and ML discussions.

In Version 2, we discovered that the sentiment polarity of Russian troll tweets fluctuated heavily over time, indicating abrupt changes in the emotions expressed. We also found that a significant portion of the tweets had a neutral sentiment, followed by positive and negative sentiments. The forecasting model based on the ARIMA technique provided predictions for sentiment trends over the next week, month, and three months. The results indicated that the sentiment would likely remain neutral or slightly positive in the short term.

The effectiveness of the different code versions varied based on the objectives and datasets. Version 1 was successful in analyzing our own tweets and gaining insights specific to our interests. However, due to the limitations of the Standard Twitter Developer account, we could only extract a limited number of tweets over a short period. Version 2, on the other hand, leveraged a pre-existing dataset of Russian troll tweets, enabling a broader analysis of sentiment patterns and trends. The forecasting model provided valuable predictions for sentiment trends, allowing for anticipation and proactive decision-making.

Despite the valuable insights gained, there are some limitations to consider. In Version 1, the analysis was constrained by the limited number of tweets extracted due to API restrictions. In Version 2, the analysis was limited to the provided dataset of Russian troll tweets, which may not fully represent sentiment patterns in the general Twitter population. Additionally, the accuracy of sentiment analysis and forecasting models depends on the quality and representativeness of the training data.

Future improvements can be made in several areas. First, obtaining a larger and more diverse dataset would enhance the generalizability of the sentiment analysis and forecasting models. Additionally, incorporating more advanced natural language processing techniques, such as deep learning models, could improve the accuracy of sentiment analysis. Furthermore, considering additional contextual factors, such as user demographics or external events, could provide a more comprehensive understanding of sentiment trends.

In conclusion, this project demonstrated the power of data analysis and sentiment forecasting using Twitter data. By leveraging the methodologies, techniques, and technologies discussed, we were able to extract insights, analyze sentiment patterns, and forecast sentiment trends. The findings can be valuable for various applications, including understanding public sentiment, guiding decision-making, and developing proactive strategies in response to changing sentiments.

# References

De-Yu, C., 2021. *Performing Time Series Analysis using ARIMA Model in R.* [Online]   
Available at: https://www.analyticsvidhya.com/blog/2021/11/performing-time-series-analysis-using-arima-model-in-r/#:~:text=ARIMA%20is%20a%20form%20of,instead%20of%20through%20actual%20values.

Elena, S., 2020. *Sentiment analysis on streaming Twitter data using Spark Structured Streaming & Python.* [Online]   
Available at: https://towardsdatascience.com/sentiment-analysis-on-streaming-twitter-data-using-spark-structured-streaming-python-fc873684bfe3

FIVETHIRTYEIGHT, 2018. *Russian Troll Tweets.* [Online]   
Available at: https://www.kaggle.com/datasets/fivethirtyeight/russian-troll-tweets

Nilson, C., 2021. *Scraping Twitter data using “tweepy” & Twitter API.* [Online]   
Available at: https://medium.com/analytics-vidhya/scraping-twitter-data-using-tweepy-8005d7b517a3

Stephen, A. R., Jake, H., Yotam, S. & Kate, K., 2023. Engagement with partisan Russian troll tweets during the 2016 U.S. presidential election: a social identity perspective. *Journal of Communication,* Volume 73.