

**C O V E N T R Y**  
**U N I V E R S I T Y**

Faculty of Engineering, Environment and Computing  
School of Computing, Mathematics and Data Science

MSc. Data Science

7150CEM

Data Science Project

Tweets and Emotions: Exploring Sentiment Analysis on Twitter

Author: Vinay Narendra Gurrup

SID: 13475101

1<sup>st</sup> Supervisor: Dr. Lakhvir Singh

2<sup>nd</sup> Supervisor: Dr. Tarjana Yagnik

Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Data Science

**Academic Year: 2023/24**

## Declaration of Originality

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

## Statement of copyright

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialise products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see [www.coventry.ac.uk/ipr](http://www.coventry.ac.uk/ipr) or contact [ipr@coventry.ac.uk](mailto:ipr@coventry.ac.uk).

## Statement of ethical engagement

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (<https://ethics.coventry.ac.uk/>) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed: Vinay Narendra Gurrup

Date: 08/12/2023

Please complete all fields.

|                                 |                    |
|---------------------------------|--------------------|
| First Name:                     | Vinay Narendra     |
| Last Name:                      | Gurrup             |
| Student ID number               | 13475101           |
| Ethics Application Number       | P165426            |
| 1 <sup>st</sup> Supervisor Name | Dr. Lakhvir Singh  |
| 2 <sup>nd</sup> Supervisor Name | Dr. Tarjana Yagnik |

This form must be completed, scanned and included with your project submission to Turnitin. Failure to append these declarations may result in your project being rejected for marking

## ABSTRACT

Microblogging sites and social media platforms are now an important sources of information sharing. Twitter is a popular micro-blogging platform where people exchange their ideas and opinions in the form of tweets which contains huge amount of raw data for opinion mining and sentiment analysis in areas such as reviews, elections, predictions, marketing, etc. Sentiment analysis is the process of analyzing large amounts of data and classifying it into different sentiment categories. In this thesis, relevant techniques have been applied to classify the polarity of the sentiments and compared their results using the experiments conducted on Sentiment140 dataset. Two types of macro-categories of models are considered: Traditional models (Naïve Bayes, RF, DT, LR) and Neural models (CNN, LSTM, and a Hybrid Approach). The approach of the test models was very similar, with the best performing model represented by Logistic Regression. Traditional models turned out to be the best trade-off and provided the best performances despite the potential of Neural models.

**Keywords:** *Machine Learning, Sentiment Analysis, Twitter Data, Deep Learning, Twitter Sentiment Analysis, Naïve Bayes, Random Forest, CNN, LSTM, Bi-LSTM.*

## TABLE OF CONTENTS

### Table of Contents

|  |           |
|--|-----------|
| <b>ABSTRACT .....</b>                          | <b>2</b>  |
| <b>TABLE OF CONTENTS .....</b>                 | <b>3</b>  |
| <b>ACKNOWLEDGEMENT .....</b>                   | <b>9</b>  |
| <b>1. INTRODUCTION .....</b>                   | <b>10</b> |
| 1.1 BACKGROUND .....                           | 12        |
| 1.1.1 DATA MINING .....                        | 12        |
| 1.1.2 TEXT MINING .....                        | 12        |
| 1.1.3 INTRODUCTION TO SENTIMENT ANALYSIS ..... | 12        |
| 1.2 PROJECT AIM AND OBJECTIVES .....           | 14        |
| 1.2.1 AIM .....                                | 14        |
| 1.2.2 OBJECTIVES .....                         | 14        |
| 1.3 OVERVIEW OF THE REPORT .....               | 15        |
| <b>2. LITERATURE REVIEW .....</b>              | <b>16</b> |
| <b>3. METHODOLOGY .....</b>                    | <b>23</b> |
| 3.1 COUNT-VECTORIZER .....                     | 23        |
| 3.2 TF-IDF VECTORIZER .....                    | 23        |
| 3.3 TENSORFLOW VECTORIZATION .....             | 24        |
| 3.4 EVALUATION METRICS .....                   | 26        |
| 3.5 ACCURACY .....                             | 26        |
| 3.6 PRECISION .....                            | 27        |
| 3.7 RECALL .....                               | 27        |
| 3.8 F1 SCORE .....                             | 27        |
| 3.9 POS TAGGING .....                          | 28        |
| 3.10 WORD EMBEDDINGS .....                     | 28        |
| 3.11 K-FOLD CV .....                           | 29        |
| 3.12 APPROACHES TO SENTIMENT ANALYSIS .....    | 29        |
| 3.12.1 LEXICON-BASED APPROACH .....            | 30        |
| 3.12.2 DICTIONARY BASED APPROACH .....         | 31        |
| 3.12.3 CORPUS BASED APPROACH .....             | 31        |
| 3.12.4 MACHINE LEARNING APPROACH .....         | 32        |
| 3.12.5 SUPERVISED LEARNING .....               | 33        |
| 3.12.6 UNSUPERVISED LEARNING .....             | 33        |
| 3.12.7 SEMI-SUPERVISED LEARNING .....          | 33        |
| 3.12.8 HYBRID APPROACH .....                   | 34        |
| 3.12.9 DEEP LEARNING APPROACH .....            | 34        |
| 3.12.10 NAÏVE BAYES .....                      | 35        |
| 3.12.11 LOGISTIC REGRESSION .....              | 36        |

|   |           |
|---|-----------|
| 3.12.12 DECISION TREES .....  | 37        |
| 3.12.13 RANDOM FOREST .....   | 38        |
| 3.12.14 XGBoost (eXtreme Gradient Boosting) .....                   | 39        |
| 3.12.15 CNN (Convolutional Neural Network) .....                    | 40        |
| 3.12.16 LSTM (Long Short-Term Memory) .....                         | 41        |
| 3.12.17 Bi-LSTM (Bidirectional Long Short-Term Memory) .....        | 42        |
| 3.12.18 HYBRID NEURAL NETWORK .....                                 | 43        |
| <b>4. DESIGN .....</b>  | <b>44</b> |
| 4.1 CRISP-DM .....  | 44        |
| <b>5. IMPLEMENTATION .....</b>                                      | <b>46</b> |
| 5.1 DATASET .....   | 46        |
| 5.1.1 Changing to lower case of text .....                          | 48        |
| 5.1.2 Tokenization .....  | 48        |
| 5.1.3 Stemming .....  | 48        |
| 5.1.4 Lemmatization .....   | 49        |
| 5.1.5 WordCloud .....   | 49        |
| 5.1.6 Hardware and System Requirements .....                        | 50        |
| 5.1.7 Python .....  | 50        |
| 5.1.8 NLTK .....  | 51        |
| 5.1.9 Pandas .....  | 51        |
| 5.1.10 NumPy .....  | 51        |
| 5.1.11 Matplotlib .....   | 51        |
| 5.1.12 Sklearn .....  | 52        |
| 5.1.13 Keras .....  | 52        |
| 5.2 ANALYSIS OF THE DATASET AND PRE-PROCESSING .....                | 52        |
| <b>6. RESULTS AND DISCUSSIONS .....</b>                             | <b>59</b> |
| 6.1 TRADITIONAL MODELS .....  | 59        |
| 6.1.1 Experiment 1 – Baseline Models .....                          | 59        |
| 6.1.2 Experiment 2 – Hyperparameter Tuning of Baseline Models ..... | 60        |
| 6.2 NEURAL MODELS .....   | 61        |
| <b>7. PROJECT MANAGEMENT .....</b>                                  | <b>64</b> |
| 7.1 PROJECT SCHEDULE .....  | 64        |
| 7.2 RISK MANAGEMENT .....   | 64        |
| 7.3 QUALITY MANAGEMENT .....  | 65        |
| 7.4 SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL CONSIDERATIONS .....    | 66        |
| <b>8. CONCLUSIONS AND FUTURE WORKS .....</b>                        | <b>67</b> |
| 8.1 CONCLUSION .....  | 67        |
| 8.2 FUTURE WORKS .....  | 68        |
| <b>9. STUDENT REFLECTIONS .....</b>                                 | <b>69</b> |

|   |           |
|---|-----------|
| <b>BIBLIOGRAPHY AND REFERENCES .....</b>              | <b>70</b> |
| <b>APPENDIX A: MEETING RECORDS .....</b>              | <b>77</b> |
| <b>APPENDIX B: HYPERPARAMETERS OF ML MODELS .....</b> | <b>79</b> |
| <b>APPENDIX C: ETHICS APPROVAL CERTIFICATE .....</b>  | <b>80</b> |

**TABLE OF FIGURES**

|   |    |
|---|----|
| Figure 1: Count-Vectorizer .....                          | 23 |
| Figure 2: TensorFlow Vectorization .....                  | 25 |
| Figure 3: Overview of the whole process.....              | 25 |
| Figure 4: Example of Confusion Matrix.....                | 26 |
| Figure 5: Example of POS Tagging.....                     | 28 |
| Figure 6: Sentiment Analysis Approach .....               | 30 |
| Figure 7: Example of Lexicon-Based Approach .....         | 32 |
| Figure 8: Example of Decision Trees .....                 | 38 |
| Figure 9: Example of Random Forest.....                   | 39 |
| Figure 10: Example of CNN Process .....                   | 41 |
| Figure 11: Example of Bi-LSTM Process.....                | 43 |
| Figure 12: Example of Hybrid Neural Network Process ..... | 43 |
| Figure 13: CRISP-DM Flow Process .....                    | 44 |
| Figure 14: Example of WordCloud.....                      | 50 |
| Figure 15: Raw Sentiment140 dataset .....                 | 52 |
| Figure 16: Clean Sentiment140 dataset .....               | 53 |
| Figure 17: Distribution of data .....                     | 53 |
| Figure 18: Key Statistics before pre-processing.....      | 54 |
| Figure 19: Key Statistics after pre-processing.....       | 54 |
| Figure 20: WordCloud of Positive Tweets.....              | 55 |
| Figure 21: WordCloud of Negative Tweets .....             | 55 |
| Figure 22: Wordlist of Positive and Negative Tweets ..... | 56 |
| Figure 23: Bag-of-Words Representation .....              | 57 |
| Figure 24: Data Visualization of Tweets .....             | 58 |
| Figure 25: Gantt Chart.....                               | 64 |

**TABLE OF TABLES**

|  |    |
|--|----|
| Table 1: Pre-Processing Techniques.....  | 47 |
| Table 2: Example of Stemming .....   | 48 |
| Table 3: Difference between Lemmatization and Stemming.....                    | 49 |
| Table 4: Traditional Model Performance with CountVectorizer .....              | 59 |
| Table 5: Traditional Model Performance with TF-IDF Vectorizer .....            | 60 |
| Table 6: Hypertuned Traditional Model Performance with CountVectorizer .....   | 61 |
| Table 7: Hypertuned Traditional Model Performance with TF-IDF Vectorizer ..... | 61 |
| Table 8: CNN Model Performance .....   | 62 |
| Table 9: LSTM Model Performance .....  | 62 |
| Table 10: Hybrid Model Performance .....                                       | 63 |
| Table 11: Risk Management.....   | 65 |



**ABBREVIATIONS**

NLP – Natural Language Processing

TF-IDF – Term Frequency – Inverse Document Frequency

TP – True Positives

TN – True Negatives

FP – False Positives

FN – False Negatives

POS – Part of Speech

BoW – Bag of Words

CNN – Convolutional Neural Network

LSTM – Long Short-Term Memory

Bi-LSTM – Bidirectional Long Short-Term Memory

CV – Cross Validation

SVM – Support Vector Machine

CRISP-DM – Cross Industry Standard Process for Data Mining

NB – Naïve Bayes

RF – Random Forest

LR – Logistic Regression

DT – Decision Tree

XGBoost – eXtreme Gradient Boosting

NLTK (Natural Language Toolkit)

NumPy – Numerical Python

Sklearn – Scikit-learn

RNN – Recurrent Neural Networks

GloVe – Global Vectors

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor Dr. Lakhvir Singh for his continuous support and guidance throughout my entire research process. He was always available to provide feedback at all stages of the project and continuously monitored my progress without delay. His feedback has enabled me to further enhance my work.

I also want to express my gratitude and appreciation to all the faculty members who have guided me throughout this course and who have shared their valuable experience with me which has enabled me to finish this project.

## 1. INTRODUCTION

Internet users now have a place to express and share their thoughts and opinions on a variety of topics and events thanks to the rise of social media. One of the most broadly utilized informal communication stages is Twitter. It permits clients to post messages communicating their inclinations, convictions, considerations, and opinions on different subjects and issues they experience consistently. The messages are called tweets, which are real-time and at most 280 characters. The monitoring and analysis of these tweets provide valuable insights to individual users, both in the private and public sectors (Siddiqua et al., 2016). The massive amount of data is the reason why sentiment analysis is selected as a technique to analyze this data due to the ease in identifying the user generated opinions without going through millions of tweets manually (Mertiya & Singh, 2016). Sentiment Analysis is one of the NLP techniques which determines whether a text is positive, negative, or neutral. Sentiment analysis is often used to discover how people feel about a topic (Gautam & Yadav, 2014). It also helps in forming the opinion or a point of view of a person and hence is also known as opinion mining.

Sentiment Analysis is carried out in three ways: Machine learning-based approach, Sentiment lexicon-based approach, and the Hybrid approach. Although classifying a tweet into either a positive or a negative sentence is a natural process for human, handling the large amount of data requires a more automated approach. Machine learning approaches were therefore developed to overcome these issues. Machine learning based approach (ML) uses machine learning classifiers in classifying the data. Sentiment lexicon-based approaches identify the polarity of the text document or text sentence by using lexicons (dictionary) that contains vocabulary of positive and negative words while analyzing the data. On the off chance that the text contains more number of positive words, it is given a positive score and assuming the text contains a more noteworthy number of negative words a negative score is given. The hindrance of the dictionary-based approach in the nostalgic examination is that it relies upon the size of the vocabulary. As the dictionary size builds, the intricacy of the examination increments. The hybrid approach uses both Machine learning and Sentiment lexicon approach for classification (Jain & Dandannavar, 2016).

Many of the companies have gone online, making it easier for readers to express their sentiments. This data is not only important for the companies but also to others which may include governmental issues, politicians and even advertising companies. Given the fundamentals of sentiment analysis, it is, therefore, important to identify techniques to automatically classify the user's opinions. Most of the research on sentiment analysis have been focusing on building an efficient machine learning and deep learning models.

## 1.1 BACKGROUND

### 1.1.1 DATA MINING

The rise of data management is always closely linked to the development of information technology. The idea of data mining has evolved with the growth of web-based data involving the knowledge and patterns discovery from huge amounts of data. Data mining techniques are used to make informed and proactive decisions by predicting future trends and behaviors. The prospective analysis provided by data mining are automated, moving away from the retrospective analysis provided by decision support system. Data mining tools makes it easier to answer business questions that have traditionally taken a long time to solve (Kroeze et al., 2003).

### 1.1.2 TEXT MINING

Opposing standard data mining practices, Text mining uses computational linguistics and NLP to extract meaningful and unknown information and patterns by analyzing large amounts of unstructured text data (Kroeze et al., 2003).

### 1.1.3 INTRODUCTION TO SENTIMENT ANALYSIS

Sentiment Analysis (Liu, 2012) also known as opinion mining, is a subfield of text mining that analyses individual's attitudes, feeling, opinions and emotions towards an entity or a subject. Most of the available text is unstructured which makes it hard for a machine to detect the proper polarity of the sentiment. Due to its practical applications which enables decision making and provide targeted insights to domain analysts, the research domain of Sentiment Analysis has become a hot topic among academics and practitioners because of the ever-increasing amount of information available in terms of size, velocity, and opinion-driven content (Kumar & Jaiswal, 2019). According to (Thormundsson, 2022) the market value of NLP will rise from 3 billion US dollars in 2017 to more than 43 billion in 2025.

Sentiment analysis is a classification task. Models that focus on polarity are used as 1 class, for example, positive, negative, and neutral. Models that are feelings-focused

may use emotions such as anger, joy, sadness, etc. When polarity precision is necessary to the application scenario, it may be relevant to expand the categories to include more shades. This approach is called the fine-grained sentiment analysis.

Document-level sentiment analysis aims to determine the sentiment of the entire text, allowing more data to be leveraged while generalizing about the content. An entity-level approach helps in identifying the sentiment in relation to the entity causing it. This is useful, for example, in situations such as product reviews where the product owner is able to understand what the weaknesses and the strengths of the product are and take measures to adapt it to the customer's expectations and needs. Sentence level stands in the middle.

Social media has permitted people to express their opinions, emotions, beliefs and feelings more openly than ever before. They express their opinions on products, companies, services, political issues, science, events, and so on. Sentiment Analysis is extremely beneficial to businesses for this very reason. It makes it possible to identify the sentiment of the customers towards products, brands or services in online conversations and feedback.

Accurate Sentiment Analysis can be used to predict product sales success, validate marketing and strategic decisions, brand monitoring, improve customer service and conduct market research. Companies can use the service to understand the tone of their customers' conversations and respond to each customer properly, or to understand and improve their customer engagements.

## 1.2 PROJECT AIM AND OBJECTIVES

### 1.2.1 AIM

The main aim of this thesis is to analyze the effectiveness of sentiment analysis algorithms in classifying tweets as positive or negative by exploring the impact of different feature extraction and text pre-processing techniques on sentiment analysis performance and investigating and evaluate the performance of the machine learning algorithms like Naïve Bayes, Random Forest, XGBoost and deep learning algorithm like CNN-LSTM model.

### 1.2.2 OBJECTIVES

- Prepare the dataset for analysis by cleaning and pre-processing the text data which involves tasks like tokenization, stop word removal, and stemming or lemmatization.
- Build and train sentiment analysis models using various machine learning or deep learning algorithms, such as Naive Bayes, Random Forest, Decision Trees, etc.
- Evaluate the performance of the sentiment analysis models using appropriate metrics like accuracy, precision, recall, and F1-score. Compare the results and determine which model(s) perform best.
- Create visualizations and plots to represent sentiment trends within the dataset.
- Analyze the results and draw conclusions about the sentiment patterns in the dataset. Discuss the practical implications of the findings and any limitations of the dataset or methods used.
- Provide recommendations for future research in sentiment analysis, considering any challenges or opportunities identified during research.

An experiment is carried out to achieve the goal and aims of the thesis, for the identified algorithms the performance metrics are evaluated, the comparison of performance is done for the different algorithms. The results of the experiment are used to identify the best fit classification algorithm for Sentiment140 Twitter dataset sentiment analysis.

## 1.3 OVERVIEW OF THE REPORT

The following chapters of the research project is organized as follows:

- Chapter 2 (Literature Review) presents the past and current research study in the field of Sentiment Analysis.
- Chapter 3 (Methodology) describes the methodological approach utilized to achieve the research objectives.
- Chapter 4 (Design) provides the reader with an overview of the project architecture and experimental setup.
- Chapter 5 (Implementation) discusses the project implementation steps.
- Chapter 6 (Results and Discussions) details the results and evaluation metrics used in the experiments.
- Chapter 7 (Project Management) focuses on the schedule, risks, and quality management aspects of the project.
- Chapter 8 (Conclusion and Future Work) concludes the project and highlights ways this project can be extended.



## 2. LITERATURE REVIEW

Much research has been conducted about sentimental analysis in the past. The most recent study in this area is to perform sentiment analysis on user data collected from many social networking platforms such as Facebook, Amazon, Twitter, etc. Majority of the research on sentiment analysis is based on machine learning and deep learning algorithms, whose main purpose is to identify whether the given text is in favor or against and to identify the text polarity.

Information exchanges, in today's times, through social media has become popular with most users actively sharing their personal thoughts and opinions publicly. For an analyst or researcher, this information is a gold mine to uncover important and valuable insights which will aid in strategic decision making (Younis, 2015). Now-a-days, most individuals consider other people's opinion, and openly express their support or disagreement with the point of view. For example, asking relatives for their opinion on a new play in the theatre, reading online product reviews before buying it, voting in an election, and considering political parties and candidate who promises the best for the society.

According to (Alves et al., 2014), sentiment analysis has been one of the most actively researched topics in the domain of NLP since the early 2000s. It helps in identifying the responses of people towards a specific topic and helps in classifying whether it is positive, negative, or neutral.

(Hemalatha et al., 2014) argues that the Twitter has more meaningful content on particular events with hashtags which has been widely shared and accepted by many celebrities.

In their experiment, (Neri et al., 2012) classified that positive or negative polarity is not the only concept of sentiment analysis. Sentiment analysis is a data structure that analyzes the words in a sentence from root to parent node. Sentiment analysis is also a system for sentence structure which analyzes the word meaning, synonyms, expression and changes polarity in the case of negative word. It also changes and modifies the polarity of words based on adverbs, nouns, and adjective.

In their research, (Isah et al., 2015) suggest that the goal of sentiment analysis is to identify and capture the moods, sentiments and attitudes of individuals and groups.

(Turney, 2002) predicts review using an unsupervised learning method by calculating the average semantic orientation of a phrase that includes adjective and adverb. This helps them in determining whether the given phrase is positive or negative and classifies it as a thumbs up or down review. (Elbagir & Yang, 2018).

(Pagolu et al., 2016) conducted sentiment analysis of twitter data for stock market movement predictions using Word2Vec. They have used N-gram to analyze the sentiments in tweets and compared the stock movement with the company sentiment across 16 tweets. This is a good example of price vs sentiment correlation analysis. The accuracy obtained with Word2vec, and N-gram applied to Random Forest classifier is same.

Alsaeedi researched on different approaches used in sentiment analysis while Dubey used a lexicon-based technique to classify sentiments. His research was primarily focused on showing the word count for each country. (Alsaeedi & Khan, 2019; Dubey, 2020).

Emojis and emoticons are a great way to express emotional status and meaning of a message. Emojis have recently become popular and may be useful to enhance the quality of models. (Wankhede et al., 2018) states that many models are inaccurate because they do not take into consideration the presence and importance of emoticons. Furthermore, they provide 3 sentiment polarity categories with different granularity. After noticing that the majority of Twitter messages fell far short of the character limit, Eisentein published an article in collaboration with Pavalanathan (Pavalanathan & Eisentein, 2015) highlighting the rise of emoji culture and its importance in text communication.

Before the development of neural models, which are possible due to the increased availability of computational power and data, NLP. Therefore, the traditional approaches were mainly based on Sentiment Analysis.

(Pang et al., 2002) were the first to work on sentiment analysis. Their main aim was to categorize text by total sentiment rather than a single topic, for example classifying the review of a play either positive or negative. They used machine learning algorithm on movie review database and the results showed that these algorithms outperform human produced algorithms. The machine learning algorithms they use are NB, maximum entropy, and SVM. In their conclusion, they looked at various factors that made it difficult to classify sentiment. They showed that supervised machine learning algorithms were the foundation for sentiment analysis.

(Gurkhe et al., 2014) explored the processing of the data. The first step was to collect data from different sources and remove those features that do not contribute to finding the polarity and then they sent this data into the sentiment classification engine i.e. naïve bayes classification algorithm. This algorithm will calculate the probability, i.e., the amount of corrected data, and it will predict the sentiment for that query.

(Gautam & Yadav, 2014) have discussed about classifying customer reviews by using the labelled Twitter dataset. They have used machine learning based algorithm i.e. naïve bayes, SVM, maximum entropy. They have worked with NLTK, Python, and SVM, naïve bayes, and maximum entropy training. When it comes to accuracy, Naïve Bayes outperforms Maximum Entropy in terms of results. We can get the better results by SVM with unigram model as compared to the SVM alone. The accuracy can be further improved by semantic analytic followed by WordNet.

(Shi & Li, 2011) created a supervised machine learning method by using unigram features to analyse the sentiment of English online hotel reviews and then they used term frequency and TF-IDF to determine the polarity of the document. The SVM classifier was selected because it was reported to perform better than other classifiers ((Pang et al., 2002), although (Tong & Koller, 2001) show that Naive Bayes and SVM are the most effective classifiers among machine learning techniques. There were 4,000 reviews in the hotel review corpus; all of the reviews had been pre-processed and labelled as positive or negative. The obtained sentiment classification model was then used to classify real-time information into positive and negative documents. The TF-IDF feature performed better than simple term frequency (Shi & Li, 2011).

Another study (Boiy & Moens, 2009) used supervised classification to determine the sentiment in documents particularly from blogs, forums and reviews (Ponte & Croft, 1998). Various features such as unigrams, stems, negation, and discourse features were conducted after pre-processing. Machine learning algorithms such as SVM, maximum entropy and Naïve Bayes classifiers were used. English-language corpora were collected from blogs, reviews, and forum sites such as livejournal.com or skyrock.com. The Maximum Entropy classifier provided 83% accuracy, which was better than SVM and Naïve Bayes (Shi & Li, 2011).

(Prabowo & Thelwall, 2009) suggested an ensemble method to identify the polarity in the documents. Different classifiers such as SVM and Naïve Bayes are trained to improve the performance. Results showed that the ensemble approach (89.23%) was better than SVM (78.78%) and Naïve Bayes (74.23%).

(Thet et al., 2010) proposed polarity detection method for movie reviews. SentiWordNet is used to determine the polarity of extracted features (ngram features). Finally, a film reviews dataset (manually collected) is used to evaluate the polarity of the dataset using SVM classifier achieving an accuracy of 67.54%.

A rule-based approach for extracting keywords from product reviews was proposed by (Poria et al., 2014). Using the Amazon product reviews dataset, the SVM achieves 91.2% accuracy compared to Naïve Bayes at 89.25%.

(Rout et al., 2017) used supervised and unsupervised approaches on different datasets obtained from Twitter. For an unsupervised approach, it achieved 80.68% accuracy on tweets. They were able to achieve an accuracy of 67% for supervised approach by combining unigram, bigram, and POS as features.

(Pak & Paraoubek, 2010) trained the training data by assuming that the emoticons in the text represented the overall sentiment in that text. Based on this assumption, a large amount of data was collected. An ensemble of two different Naïve Bayes classifiers were used: one trained using unigrams while the other trained using POS tagging. When the two classifiers were combined, they accuracy was 74%

Another important study (Melville, 2009) in the field used lexical knowledge with Naïve Bayes model to create a classifier that performed better as compared to the performance of the individual models. The model was evaluated on several different datasets with the most significant result being an accuracy of 91.21% on a Lotus dataset.

The main differences between random forest (RF), support vector machine (SVM), and random forest support vector machine (RFSVM), which are effective in generating appropriate rules for classification technique, have been clearly explained by (Amrani et al., 2018). Based on the results of this experiment, Random Forest Support Vector Machine algorithms (RFSVM) performed better for classification of Amazon Product reviews (dataset is provided by Amazon). The use of all classification methods, SVM and RF, is the reason why the hybrid classification yielded better results.

For the movie reviews data set, (Wankhede & Thakare, 2017) proposed the Random Forest method for the prediction of sentiment. They compared Random Forest model with SVM model, Hybrid SVM model, Max Entropy model, NB model, and found that Random Forest model performed better with 90% accuracy compared to other algorithms.

(Aramaki et al., 2011) focused on classifying tweets including keywords into two categories by using Machine Learning and Twitter to detect influenza epidemics and a number of machine learning algorithms were compared. The random forest performed best with an accuracy of 72.9% on the test dataset.

(Samuel et al., 2020) presented sentiment analysis of Tweets about Coronavirus using Naive Bayes and Logistic Regression. Different lengths of tweets were analyzed; less than 77 characters (small to medium) and less than 120 characters (longer). Naive Bayes proved to be better with an accuracy of 91% at classifying small to medium size Coronavirus Tweets sentiments. With an accuracy of less than 57%, both approaches performed poorly for longer tweets.

(Mukku et al., 2017) used a smaller dataset to solve the issue of label annotation for Telugu data that were not labelled. To overcome this issue, hybrid approach is

suggested that combines different query selection strategy framework in order to increase the accuracy of training data. The author used Gradient boosted trees (GBT), SVM, XG boost to conclude his analysis. XG boost performed better as compared to the selected algorithms with a precision of 79%.

Due to the drawbacks of traditional techniques, researchers have begun to look for more advanced approaches to processing text data. Deep learning is well-proven in NLP tasks and is being used more frequently and successfully in NLP tasks.

Using Natural Language Processing (NLP) techniques (Vishwakarma et al., 2019) analyze based on their mental health. Using DL models, they classified each text with the following emotions: angry, anticipation, disgust, frighten, delight, sadness, surprise, and confidence.

In their paper, (Fitri et al., 2019) used sentiment analysis to identify human emotions by using NLP and Machine Learning models to train and validate the dataset by comparing the results using various ML classifiers (Naïve Bayes; Random Forest; Support Vector Machine) to determine whether the unconstructed text was negative, positive or neutral.

A study was conducted by (Back & Ha, 2019) comparing the Naive Bayes algorithm with Natural Language Processing (NLP) on the Twitter dataset. Accuracy and speed were the two categories for their comparison. Their results showed that the Naïve Bayes algorithm got 63.5% accuracy, which is less than that achieved by the NLP method. However, processing speed analysis showed that machine learning method performs 5.4 % faster than NLP.

(Makhadmeh & Tolba, 2019) presented a hybrid method which is a combination of Machine Learning and NLP technique to identify and predict hateful speech from social network platforms. After the hate speech was captured, it was stemmed, tokenized, and unnecessary characters removed.. The text is then classified into neutral language, offensive language, and hate language (in our study the tweets were classified as positive and negative). The performance of the system is then evaluated

using overall accuracy, f1 score, and precision and recall metrics. The system achieved an accuracy of 98.71%.

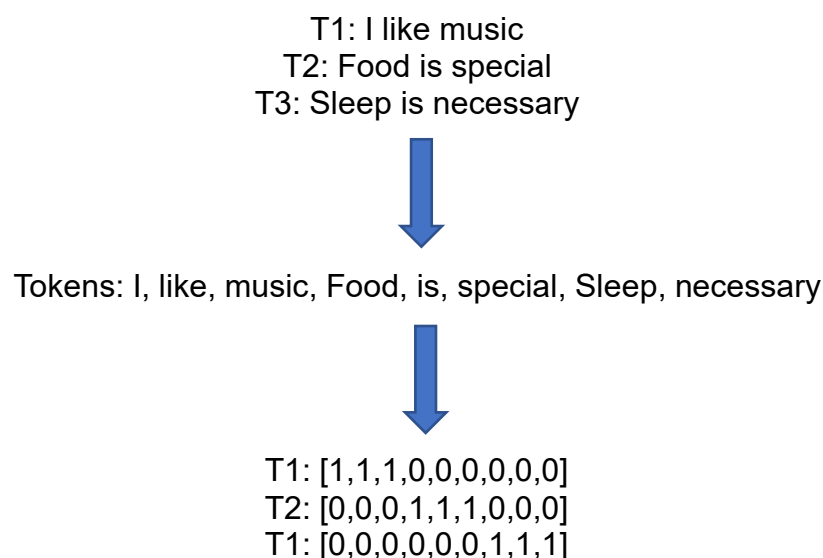
(Jain et al., 2019) used four small datasets and applied sentiment analysis and then the dropout was used to control overfitting. A hybrid approach combining CNN and Bi-LSTM is presented. The experimental results showed that the proposed methods performed much better than the existing methods over the considered datasets.

(Ouyang et al., 2015) suggested a framework which using a deep learning classifier in identifying polarity in English movie reviews. The movie reviews were collected from rottentomatoes.com. The dataset consisted of five labels: positive, somewhat positive, neutral, somewhat negative, and negative. The results showed that the RNN performed better with an accuracy of 78.34% as compared to SVM (62%).

### 3. METHODOLOGY

#### 3.1 COUNT-VECTORIZER

Count-Vectorizer is a part of Scikit-learn library (Pedregosa et al., 2012). It allows the creation of a matrix of token counts from a set of text documents. A sparse representation of the counts is the end result. The number of features in the updated data representation matches the vocabulary size discovered during data analysis. Due to this the messages consist of a small number of words compared to the total number of tokens; the use of sparse matrices reduces the amount of storage space needed. It is a simple approach that does not consider the order of the words as well as the importance of each word in a sentence. Here is an example of how it works:



*Figure 1: Count-Vectorizer*

#### 3.2 TF-IDF VECTORIZER

Another method to count vectorization is to calculate the frequency of the words where the basic idea is that a word that appears frequently in the encoded vectors has less information as compared to the words that appear less frequently. The most common method of calculating word frequency is called “Term Frequency – Inverse Document Frequency”. Each word is assigned a value that indicates the importance of the word



for a document in a collection of documents (Leskovec et al., 2010). This value, called TF-IDF value, is a product of inverse document frequency and term frequency.

If we define the frequency of the term  $t$  in a document as  $n_t$  and the total number of terms in the document as  $N_t$ , then the frequency of the term  $t$  can be defined as follows:

$$TF(t) = \frac{n_t}{N_t}$$

Similarly, inverse document frequency can be defined by defining  $N_d$  at the total number of documents, or messages in the context of this thesis, and  $n_{d,t}$  and the number of documents containing the term  $t$ :

$$IDF(t) = \log_e \left( \frac{N_d}{n_{d,t}} \right)$$

The final value is nothing but the product of these two factors:

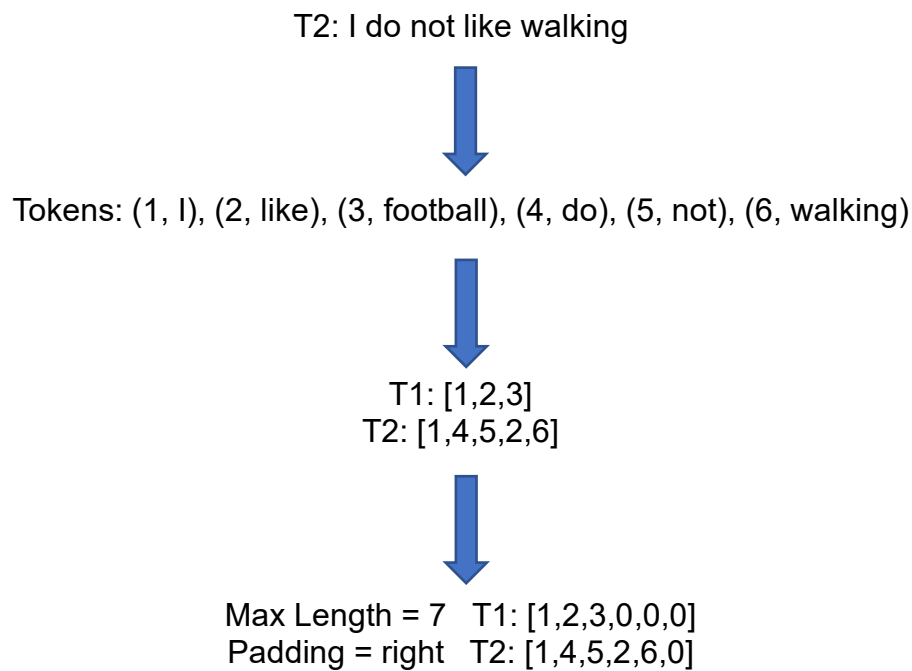
$$TF-IDF(t) = TF(t) * IDF(t)$$

It increases with the number of times a word occurs in the document while also decreasing with the number of documents in the corpus that contain the word, helping to adjust as required as some words appear more frequently in general.

### 3.3 TENSORFLOW VECTORIZATION

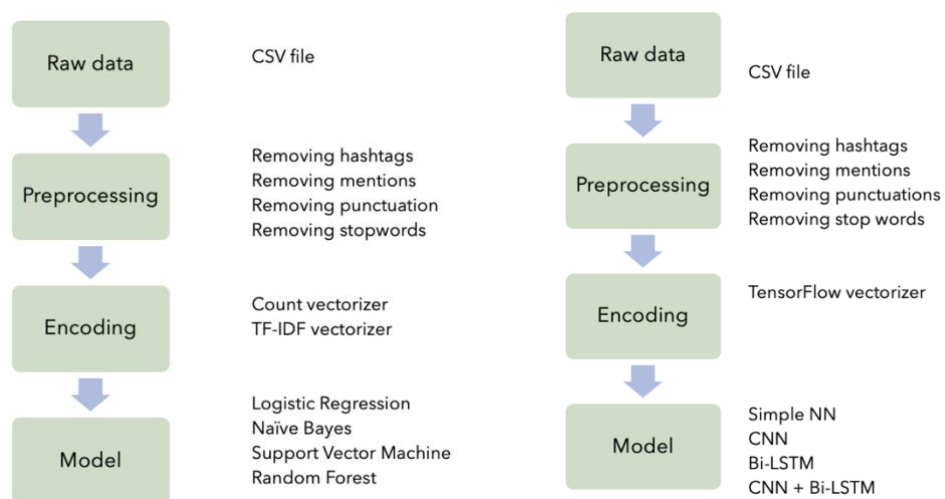
TensorFlow's built-in Tokenizer starts the vectorization process. The vectorization process has three steps. Tokens that will be utilized in the next phases are created in the first step. Here, each word is replaced with its token representation, changing the sentences which can be thought as a sequence of words to a sequence of tokens. each token sequence is padded by adding zeros to the right until the maximum length is reached, as the neural network requires inputs of equal size. This allows the data to be processed as integer numbers, but it also reduces the total memory to store the data because each word, which is composed of different characters is mapped to an integer.

T1: I like football



*Figure 2: TensorFlow Vectorization*

The complete process, from data collection to the final model can be schematized as follows, highlighting the major phases and essential information that is important for each stage.

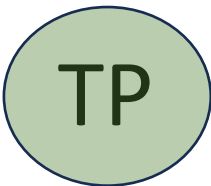
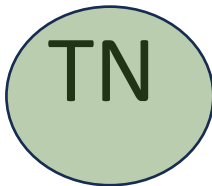


*Figure 3: Overview of the whole process*

### 3.4 EVALUATION METRICS

Every machine learning research project can be evaluated based on its performance results. The confusion matrix and the classification report, which includes accuracy, precision, recall, and F1 Score, are commonly used as measures to assess the performance of the models. While classification report gives the outcome of the classifications' method to see whether the suggested model fits the actual data or not, confusion matrix shows the number of input elements that have been correctly classified and then number of input elements that have not been classified accurately. The rows of the matrix represent the instances in the predicted class, while the columns represent the instances in a real class. Classified elements fall into following categories:

- True Positive (TP): Case was positive, and it predicted positive.
- True Negative (TN): Case was negative, and it predicted negative.
- False Positive (FP): Case was negative, and it predicted positive.
- False Negative (FN): Case was positive, and it predicted negative.

|  |              | Predicted Class  |  |
|--|--------------|--|--|
|  |              | Positive   | Negative   |
|  | Actual Class |  TP | FN   |
|  | Negative     | FP   |  TN |

*Figure 4: Example of Confusion Matrix*

### 3.5 ACCURACY

Accuracy helps us to measure how many tweets were predicted correctly out of all the tweets in the dataset. The percentage of a test set that the classifier successfully

classifies is known as the classifier accuracy of the test set. This is also known as the classifier's total recognition rate in the research on pattern recognition. Using training data to estimate the accuracy of a learned model results in overfitting of the learning algorithm. It is better to measure accuracy on a test set made up of class-labeled tuples that have not been trained to train the model. (Visa et al., 2011).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

### 3.6 PRECISION

The precision metric allows us to count the number of tweets that were correctly predicted to belong to a particular category out of all the texts that were predicted correctly or incorrectly. Precision (P) is the percentage of the positive cases that have been predicted correctly, by calculating using the equation:

$$Precision = \frac{TP}{TP+FP}$$

### 3.7 RECALL

Recall allows you to measure the number of tweets that are correctly predicted to belong to a certain category out of all the tweets that should belong to that category. Recall which is also known as the true positive rate (TP) shows the percentage of positive cases that are accurately identified.

$$Recall = \frac{TP}{TP+FN}$$

### 3.8 F1 SCORE

F1 score which is also known as F-Score or F-measure is the weighted average measure of precision and recall. The F1 score ranges from 0 to 1 and, it is considered perfect when it is equal to 1 indicating that the model has a low number of false positives and a low number of false negatives.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

### 3.9 POS TAGGING

POS Tagging is extremely valuable in Opinion Mining procedure (Davidov et al., 2010). When examining a sentence or document, we must focus on the subjective information included in that sentence or record. POS Tagging can help in finding the word parts of speech. We can perform different activities on these words after we have extracted them and come to a decision. POS Tagging is performed using the HMM model that tokenizes and tags the words additionally for naming element. The words within the texts are tagged using a POS-tagger and the purpose of this is to assign a name to each word, allowing the machine to do something about it. The POS-tagging looks like this:

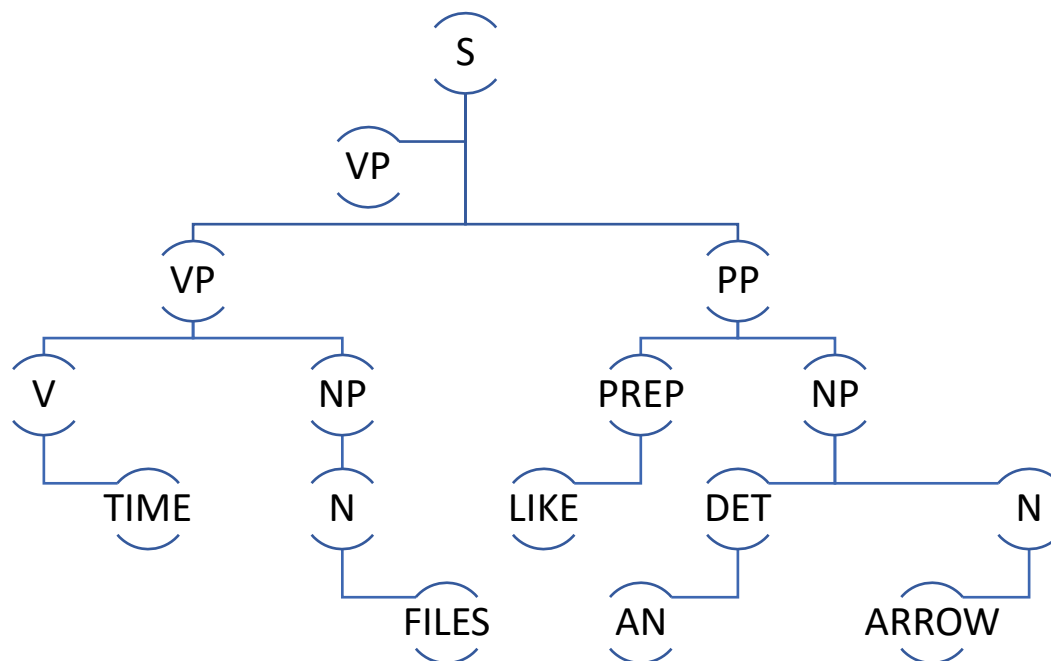


Figure 5: Example of POS Tagging

### 3.10 WORD EMBEDDINGS

The last transformation that allows the neural networks to process the input data is the vectorial representation. It's called embedding, and it's essentially a mapping of input data to a vector of real numbers. The goal of word embeddings is to translate the semantic meaning to a geometrical space. To do this, each word in a dictionary is

given a numeric vector, such that any distance between two vectors may be used to partially represent the semantic link between the two related words. The geometrical space that these two vectors form is called the embedding space.

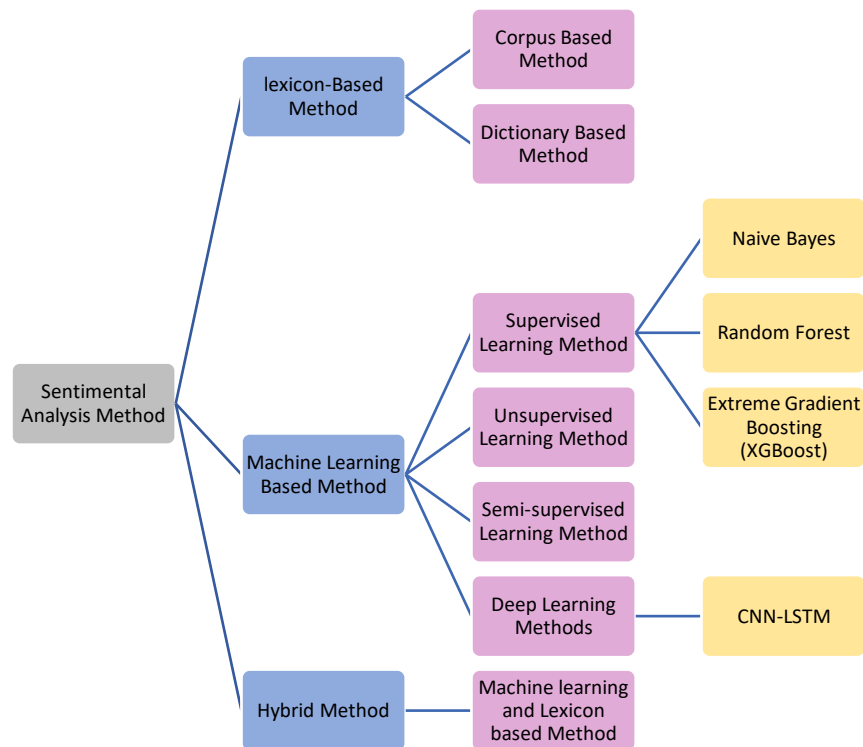
The best embedding space is one where words with similar semantic are placed closed to each other in the space. This allows you to classify a sentence's overall sentiment by looking at how the embedding represents the words in the sentence. (Chollet, 2016) Keras provide a simple way to create the embedding space. The Embedding layer takes the sequence of integers which has been produced through the vectorization process as inputs and produces a new representation that is used by the following layers.

### 3.11 k-Fold CV

In order to prevent issues like overfitting and underfitting and to determine how well a model will perform to a separate dataset, cross-validation is used to evaluate the model throughout the training phase. Cross-validation helps in choosing the model that works better on unseen data. Underfitting is when the model does not capture enough patterns in the data and performs poorly because of it. When a model performs well on the training set but incompetently on the test set, it is said to be overfitting. The data is divided into two sets for cross-validation: the test set and the train set, here the sets don't overlap. However, cross-validations are not commonly used to validate deep learning models due to the higher computational costs.

### 3.12 APPROACHES TO SENTIMENT ANALYSIS

There are many approaches used for sentiment analysis, and which approach to be used depends on the nature of the data and the platform you are working on. The majority of sentiment analysis research uses machine learning or lexicon-based analytic methods. The Machine learning techniques controls the data processing using machine learning algorithms and by classifying the linguistic data representing them into vector form (Olsson, 2009). On the other hand, the dictionary-based (or lexicon-based) technique uses a dictionary lookup database to classify the linguistic input. In the classification process, lexicon databases such as WordNet, SentiWordNet, and treebanks are used to compute sentiment polarity at the sentence or document level.



*Figure 6: Sentiment Analysis Approach*

### 3.12.1 LEXICON-BASED APPROACH

In this method, a pre-defined dictionary of pre-defined lexicons is used however, the dictionary can be different for different use cases. This works on simple principle: Using a predetermined token sequence (uni-gram, bi-gram, word-level, etc.), the input text is divided into tokens, and each token is compared to the dictionary's entries. Token scoring will be performed if a match is found, otherwise the token will not be scored. Lexical analysis based on polarity is also an option. This method assigns a sentiment score to the incoming token sequence based on the number of matches found in the text by simply looking for a token match positive list or the negative list. This simple approach has the capability to produce very high-quality sentiment classification results. One of the original methods for classifying sentiment, it could achieve 80% accuracy on individual sentences including adjectives (Sadia et al., 2018).

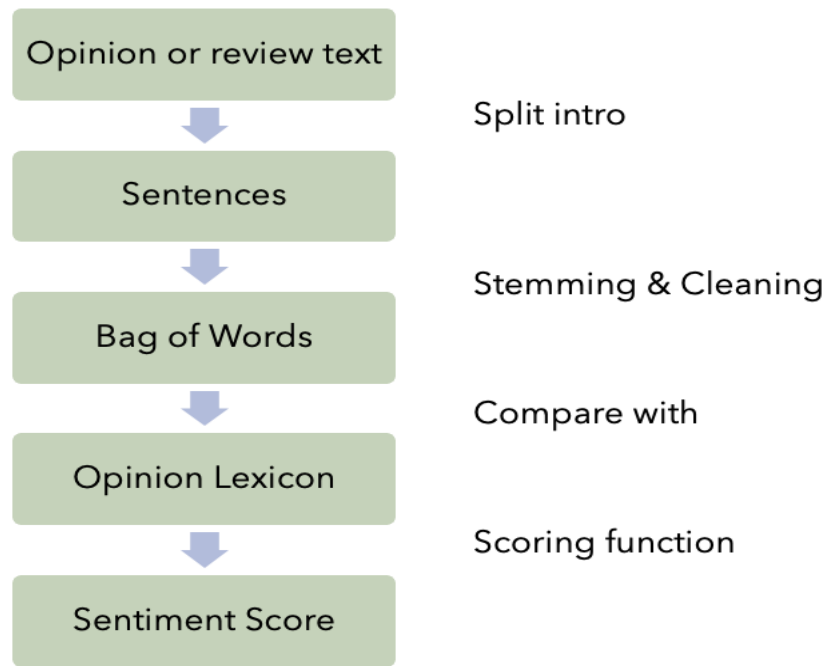
### 3.12.2 DICTIONARY BASED APPROACH

The dictionary approach involves manually collecting a small set of words with known orientations, and the number of words is then increased by looking up for synonyms and antonyms in a well-known corpora. (Hu & Liu., n.d.). The words with strong opinions are compiled using it. Usually, there is a list of negative and positive sentiments in a dictionary and the process is very simple for the dictionary-based approach. First, we collect a list of the known positive or negative words manually. Then, the algorithm looks for synonyms and antonyms from the WordNet dictionary or online dictionary to grow this dataset. Later the wordlist is updated if found and then it goes through several iterations. The process continues until there are no more words to update the dictionaries and then the list is cleaned manually to finish the process.

### 3.12.3 CORPUS BASED APPROACH

The corpus-based technique looks for opinions words that have the orientations related to the context. Looking for other opinion words in the large corpus is based on syntactic patterns that appear together with the starting list of opinion words. This method was used in (Hatzivassiloglou & McKeown, n.d.), in which they identified more adjective opinion terms and their orientations by first using a small set of opinion adjectives and then a set of linguistic limitations. (Medhat et al., 2014). The disadvantage of the corpus-based approach is that it is not as powerful as a dictionary-based approach as it is difficult to create a large corpus of all English words. Corpus based approach can be further divided into two techniques: semantically, and statically.





*Figure 7: Example of Lexicon-Based Approach*

#### 3.12.4 MACHINE LEARNING APPROACH

This method is capable of producing high level of accuracy and a good domain-flexibility. This might be one of the reasons why this technique is more preferred. For labeled sentiments datasets, machine learning algorithms are one of the most suitable methods for sentiment analysis. Uni-grams, bi-grams and tr-gram sequences can be used as feature vectors to represent single-word, two-word, and three-word phrases, respectively. N-grams of higher order are useful when you need additional adjectives or adverbs. The meaning of bigrams is also increased when negation and indirect word references are used. For example, when using unigram, the statement 'This is not good' may be classified as positive due to the word 'good,' but when bigrams are used, 'not good' is classified as negative attitude. Supervised ML models can be used for classifying sentiment analysis such as SVM (Support Vector Machine), Nave Bayes, and Random Forest. When these supervised algorithms are used, accuracy ranges between 60%-80% for classification (Elbagir & Yang, 2018).

### 3.12.5 SUPERVISED LEARNING

A well-labeled corpus is needed for supervised learning in order to train a classifier. There are a number of supervised learning algorithms. The main challenge with supervised learning methods is a well-defined labelled data must be used in order to function, otherwise it is difficult to perform training and testing of the models. There are two categories of supervised learning: regression and classification. Regression provides a trained, labelled data set and attempts to predict and improve the model through iterative runs using available solutions. This classification helps in finding the appropriate class labels that can be used to predict positive, negative, and neutral sentiment. Supervised learning involves building machine learning models that uses labelled data to train and classify tweets and tries to predict their sentiment. (Ghahramani & Jordan, 1996).

### 3.12.6 UNSUPERVISED LEARNING

Unsupervised techniques can be machine learning-based or lexicon-based. Unsupervised learning does not always need the labelled corpus. In this method, the raw data is provided to the machine and the model does not require labelling. Another name for unsupervised learning is pattern recognition. Clustering is a classic example of unsupervised learning. A Sentiment Lexicon is typically used in an unsupervised method to sentiment analysis (Kharde & Sonawane, 2016; Go et al., 2009). Semantic orientation has been used to classify text where various algorithms are used to extract phrases which contain adjectives and adverbs to determine the semantic orientation of a phrase.

### 3.12.7 SEMI-SUPERVISED LEARNING

Semi supervised learning is a type of supervised learning model that falls between unsupervised and supervised learning models. A set of labelled and unlabeled data is used in the semi-supervised learning model. The objective of the semi-supervised learning process is to classify some of the unlabeled data using labelled information set. There are few difficulties in predicting the sentiment of the Twitter datasets, such as the size of the unlabelled dataset should be larger than labelled data, the input-

output symmetry, relatively simple labelling, and the low dimensionality of the problem. This model is most commonly used in stock trend analysis and is not suitable for sentiment analysis of datasets. (Felix et al., 2014)

### 3.12.8 HYBRID APPROACH

Hybrid approach combines the best features of both the lexicon approach and the machine learning-based approach to improve the accuracy and speed of classifiers. To create a hybrid sentiment analysis method, take any basic classifier (e.g., Random Forest, SVM, or Naïve Bayes) and combine it with a lexical component. Several algorithmic approaches have been tried and tested in Twitter to conduct sentiment analysis.

### 3.12.9 DEEP LEARNING APPROACH

Deep learning is a form of machine learning that uses more than one layers of neural networks to generate high-level features and it is based on the fundamental principles of artificial intelligence (Majumder et al., 2017). It is a powerful learning technique that uses neural networks to solve problems. Neural Networks are similar to the neurons in the human brain. Artificial neural networks consist of three layers namely an input layer, an output layer, and an optional hidden layer. Each node in a neural network has an input value, and each edge has a weight. The weights are initially random values, and then a bias is applied that is always fixed. The way the neural networks work is by calculating a weighted sum.

$$\text{weighted sum} = \sum w_i x_i + b$$

The weighted sum is used as an activation function to improve the output. Activation Functions are used to make the output into a non-linear so that classification can be achieved. Relu is a rectified linear unit in order to achieve only positive values and zeros.

$$\text{Relu} = \max(0, x)$$

The sigmoid function is a special S-shaped curve which limits the values between 0 and 1.

$$\text{sigmoid} = 1/(1+e^{-x})$$

The training data is used as a set of examples to train the neural network in a deep learning model. Since the output of the input is already known, the neural network learns from these examples to predict the outputs. The ratio of the number of correctly classified samples to the number of samples used in the training data is known as training accuracy.

A test data set is a set of examples that is used to see how well a neural network has learned to classify from the training data set. Since the output to these inputs are already known, the neural network is tested to see if it generates the expected predictions while being tested on a new data that is different from the training data set. The test accuracy is determined by the relationship between the number of correctly classified examples and the number of examples that are used as testing data. In supervised learning, data fed to the algorithm includes the desired solutions call labels whereas in unsupervised learning, the data is not labelled. The deep learning model tries to learn without any supervision (Majumder et al., 2017).

### 3.12.10 NAÏVE BAYES

In order to classify input data, The Naïve Bayes classifier is a simple probabilistic model that is based on the assumption that the features are independent on each other. The Naïve Bayes algorithm is one of the most commonly used for text classification in opinion mining (Pak & Paraoubek, 2010; Go et al., 2009). A large part of its success is due to its simplicity, low computational costs, and high precision.

The technique is referred to as "naïve". It makes the assumption that every characteristic in the input data is independent of every other feature's existence or absence. In reality, the words in the sentence are highly correlated, and their position and presence in the sentence have a significant influence on the meaning and sentiment of the sentence. Despite this simple assumption, the classifier can achieve high classification accuracy with high quality training and in particular domains. A recent study (Zhang et al., n.d.) addressed this assumption and provided strong

evidence of how the algorithm can achieve high accuracy while relying on this assumption.

The algorithm itself is derived from Bayes Theorem:

$$P_{NB}(c|d) = \frac{(P(c) \sum_{i=1}^m P(f_i|c)^{n_i(d)})}{P(d)}$$

Where  $P(c)$  and  $P(f_i|c)$  are computed by calculating the relative frequency of a feature “ $f_i$ ” extracted from the training data corpus and where  $n_i(d)$  is the number of these features. There are ‘ $m$ ’ features in the entire training data corpus. The documents ‘ $d$ ’ contains the training data or the input data to be classified.

The algorithm takes each word in the training set as a feature and determines the likelihood of it to be in a positive or negative class. Once the probabilities of each feature have been calculated accurately, the algorithm is ready to classify the new data. For example, when classifying a new sentence, it will be split into individual word features. The model used the conditional probabilities that were calculated in the training phase to predict the class of the combined features.

### 3.12.11 LOGISTIC REGRESSION

Linear regression is the process of finding a relationship between a dependent variable (class)  $Y$  and an independent variable (features)  $X$  by using a linear equation on the observed data (James et al., 2013). A hyperplane linear regression has an equation as follows:

$$Y = \alpha + \beta X$$

where  $X$  is an independent variable and  $Y$  is the dependent variable. The slope of the hyperplane is  $\beta$ , and  $\alpha$  is the intercept (the value of  $Y$  when  $X = 0$ ).

Linear Regression can generate any result. To classify an independent variable, we need a model that predicts the probability that the independent variable falls within the interval [0..1]. Logistic regression model was created to predict the probability that an outcome can have only two values. That means it models  $P(x) = P(y=1|x)$  where we suppose  $y=1$  is the positive class and  $y=0$  is the negative class. This can be written as:

$$P(x) = p(y=1|x) = \frac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}} = \frac{1}{1 + e^{-(\alpha+\beta x)}}$$

where  $P(x)$  is the probability of the independent variables,  $p(y=1|x)$  is the probability of the positive class given the independent variables,  $e$  is the base of the natural logarithm with  $\beta$ , and  $\alpha$  as the parameters of the model.

### 3.12.12 DECISION TREES

A decision tree classifier uses a diagram-like structure to create a classification model in the form of a tree. Each node inside the tree represents a test for a feature (such as whether a word is subjective or objective), and the individual branches represent the features that lead to each class label. The paths from the root to the single leaf represent classification rules. A DT uses a set of if-then rules, which are mutually exclusive and comprehensive for the classifier. Using the training data, the rules that partition the data sequentially are learnt one after the other. The tuples that a rule covers are eliminated after each addition, and this process continues until no more data is left (Quinlan, 1986). Because the tree is constructed from top to bottom, each attribute must be either categorical or pre-determined. Attributes at the top of the tree have a large impact on classification. Because DT classifiers are simpler models to represent, they are easier to understand, and because they are outliers, they require less pre-processing of data. On the other hand, such classifiers have a tendency to overfit and generate a large number of unnecessary branches, so the researcher needs to perform pre-pruning, which stops tree growth at an early stage, or post pruning, which takes out branches from a fully grown tree at the end (Bramer, 2007).

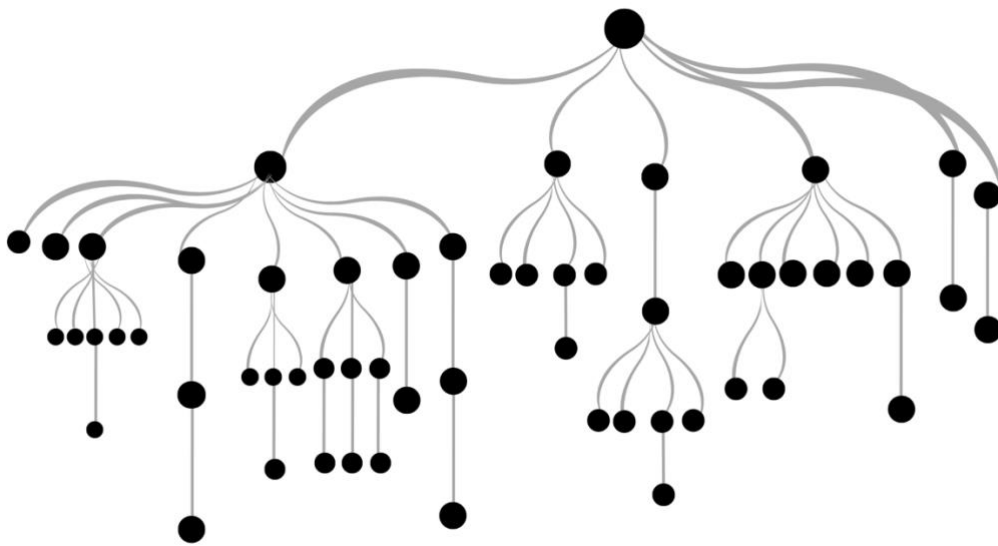
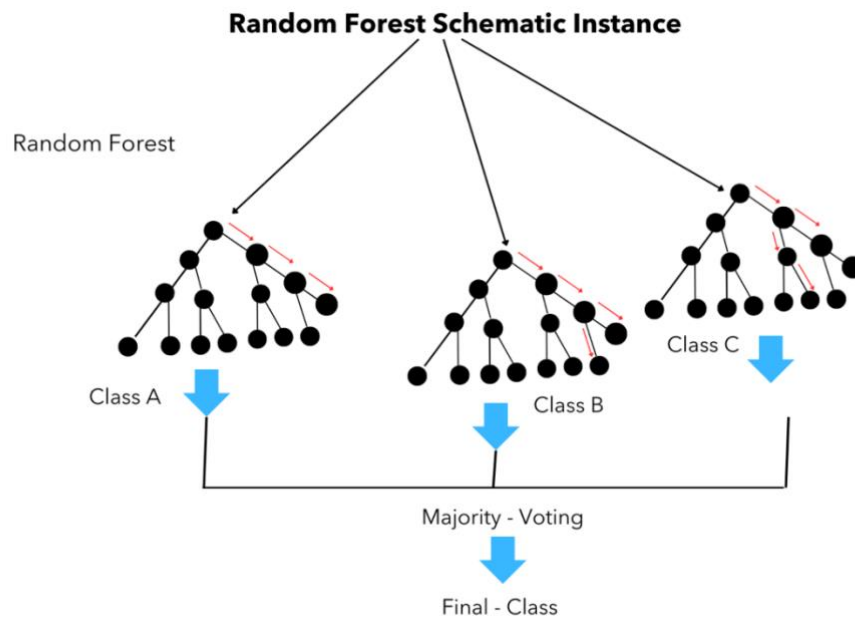


Figure 8: Example of Decision Trees

### 3.12.13 RANDOM FOREST

Ensemble learning focuses on methods to combine the output of different trained models to create a more functionally correct classifier. Ensemble models generally show significant performance improvements over the performance of a single model.. The random forest algorithm was first introduced by (Breiman, 2001). This is a very simple algorithm, and it can provide state-of-the-art performance when it comes to classification. The basic structure of the random forest can be seen below:



*Figure 9: Example of Random Forest*

Random forests are created by combining several decision tree classifiers, and each tree is trained on bootstrapped subset learning data. At each decision node, a random subset of the features is chosen, and the algorithm only considers splits on those features. The main disadvantage of using a single tree is that it has a high degree of variance, which means that the way the training data and the features are arranged can affect its performance. The variance of the overall classification can be decreased by averaging the values of an ensemble of trees, even when each tree has a high variation on its own. According to the central limit theory, if all the trees are not highly correlated and have accuracy greater than pure chance, then their average will result in a Gaussian distribution. The more decisions that are averaged the lower the variance becomes. Lowering the variance will generally improve the model's overall performance by reducing the overall error.

#### 3.12.14 XGBoost (eXtreme Gradient Boosting)

In the paper (Mukku, 2017), the author addressed the problem for annotating labels for unlabeled Telugu data using limited dataset. In order to address this, a hybrid solution that combines many query selection strategy frameworks is suggested in order to improve training data instances accuracy with smaller datasets. To achieve this, the author uses SVM as well as XGboost. XGboost is a Gradient boosted tree



(GBT) algorithm. The accuracy of XG boost is 79%, which is higher than the other algorithms. Since XGBoost is highly accurate in delivering better results for unlabelled data, it was used in this study to find out whether it performs similarly on Twitter dataset and to know its accuracy value.

XGBoost stands for eXtreme Gradient Boosting. “The name XGBoost, though, actually refers to the engineering goal to push the limit of computations resources for boosted tree algorithms which is the reason why XGBoost is used” (Chen and Guestrin, 2016). It is an ensemble machine learning technique that uses decision trees to build the gradient boosting framework. The XGBoost incorporates the gradient boosting decision tree algorithm. Boosting is the most common way of adding new models to the blunders of existing models. New models are added each in turn until there are no more models that can be gotten to the next level. Gradient boosting is a approach in which new models are created to predict the residuals or errors of older models. These new models are then combined to produce the final prediction. As it minimizes the loss while adding new models by using a gradient descent approach, it is also known as gradient boosting. This method works for both, regression, and classification predictive modelling problems (Ren et al., 2017).

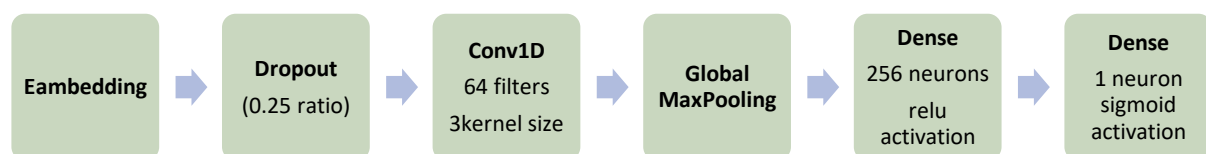
### 3.12.15 CNN (Convolutional Neural Network)

CNNs are made up of several convolutional layers with the outputs subjected to nonlinear activation functions like ReLU or tanh. In a typical feedforward neural network, each neuron in the input layer is connected to each neuron in the output layer. This network is also referred to as an affine layer or a completely linked layer (Zhang et al., 2018).

Convolutions across the input layer are used in CNNs to compute the output. This results in local connections, where a neuron in the output is connected to every input area. As can be seen above, each layer applies hundreds or thousands of filters, and mixes the resulting images. During the training phase, a CNN learns the values of the filters based on the given task.

For example, in order to classify images, a CNN might learn to recognize the edges of raw pixels in layer 1, use these edges to recognize simple forms in layer 2, use those simple forms to identify higher-level features such as facial shapes in layer 3, and the last layer is the classifier that uses those high-level features. The input to most NLP jobs isn't picture pixels. Instead, it's a matrix of sentences or text. Each line in the matrix represents a token, usually a word but sometimes a character. In other words, every row is a vector that represents a word. Word vectors are usually low-dimensional word representations (Word2Vec, GloVe, etc.), but they can be 1-dimensional word indexing vectors that index a word to dictionary.

CNN performs well when faced with the dimensionality reduction problem (Goularas & Kamis, 2019), extract features at different positions and capture short-range and long-range relationships (Toutanova & Wu, 2014). This demonstrates the usefulness of CNN in text analysis as they are able to identify key characteristics of text data.



*Figure 10: Example of CNN Process*

### 3.12.16 LSTM (Long Short-Term Memory)

LSTMs were first developed by Hochreiter and Schmidhuber in 1997 (Jeenanunta et al., 2018). They are able to capture context-specific temporal dependence over a long period of time (Nandakumar et al., 2018). LSTM combats the problems of vanishing and exploding gradients and long-term dependence by introducing hidden units (Hochreiter & Schmidhuber, 1997). LSTM works similarly to RNN: information about previous inputs is used to process the next input; however, each LSTM has gates that decide whether information should be retained or deleted from memory (Hochreiter & Schmidhuber, 1997). Memory operations are based on the importance of this information. LSTM has three information control gates: input gate, output gate and forget gate. Additionally, Cn cell status is used to convey information (colah's, 2015).

The state of a cell is affected by vector operations, such as addition or multiplication. (colah's, 2015).

The forget gate decides which Information should be deleted. The current input and the previous input are used to determine this, and the Sigmoid function returns a value between 0 and 1, where 0 indicates that the information should be deleted and 1 indicates that it should be retained (colah's, 2015).

The input gate determines what new data should be retained. The Tanh function creates a vector of new values that may be added to memory, while the sigmoid function is responsible for updating the associated values that describe the sequence (colah's, 2015). The value produced by the forget gate is then multiplied by the old cell state, and the result is added to the value produced by the input gate. This updates the cell state and carries the new information to the next iteration.

At the last step, the output gate makes a decision about what to do with that input. First, the sigmoid function is used for the current input and previous output (colah's, 2015). Then, tanh function is used for the updated cell state to multiply that with the output of sigmoid function. (colah's, 2015).

### 3.12.17 Bi-LSTM (Bidirectional Long Short-Term Memory)

LSTM is an example of a recurrent neural network (RNN). A RNN has circular connections. The network can reuse the output of the neuron as input back to the other neurons. In addition to the input data, the network can build relationships between the data. However, the RNN cannot connect the relevant information if there is a large difference in the input data. Furthermore, the RNN is only able to process previous context. New information cannot directly affect the past. LSTM addresses the problem of connecting distant information by introducing the concept of "storage" of relevant information.

The Bidirectional property overcomes the second weakness: training in both direction at the same time, with different hidden layers (Yu et al., 2019). It requires two LSTMs,

one acting forward and the other acting backward. Due to the higher number of operations than other neural models, the training time is much slower.



*Figure 11: Example of Bi-LSTM Process*

### 3.12.18 HYBRID NEURAL NETWORK

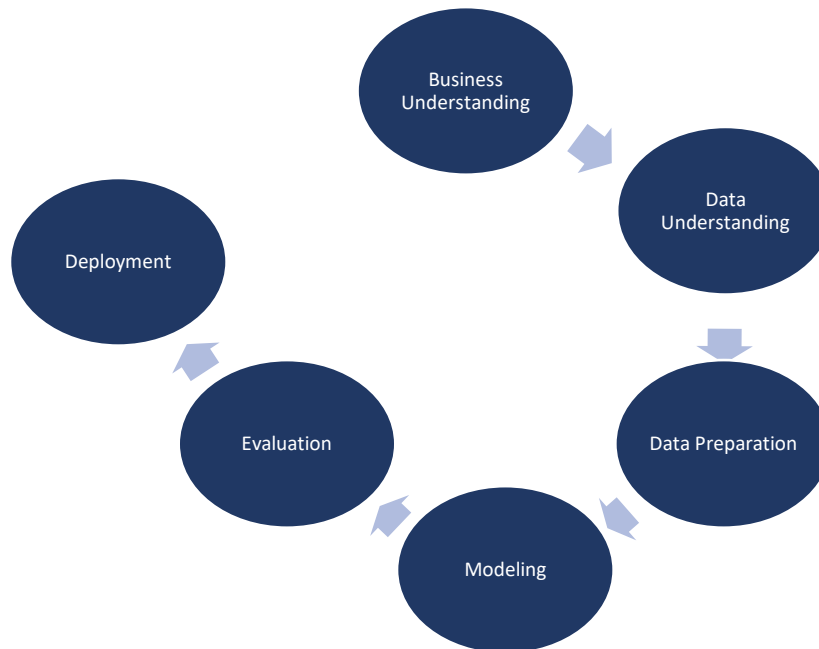
Following the approach provided in (Jain et al., 2019), a hybrid solution has been created combining both, a CNN and Bi-LSTM. Despite following its institution, the models have been combined in a manner different from the one outlined in the above research paper. Rather than using an ensemble approach, the CNN output is fed into the Bi-LSTM in an effort to capture more semantic information about the sentences.



*Figure 12:Example of Hybrid Neural Network Process*

## 4. DESIGN

### 4.1 CRISP-DM



*Figure 13: CRISP-DM Flow Process*

- The first stage is business understanding: which is about analyzing the business, objectives, and requirements.
- The second stage is the Data Understanding step: This includes data collection, data visualization, data summarization, and deriving insights from the data.
- The third stage is the Data preparation: Also known as the data pre-processing phase, the final data set is created from the original data set. Various pre-processing steps are performed during this phase. Examples: Removal of stop words, removal of numbers, removal of special characters
- The fourth stage is the modelling stage: different machine learning algorithms are applied, and their parameters are tuned to equip the dataset to be investigated. We used TFIDF-Vectorizer to embed the text and split the data in a ratio of 80: 20 for training and testing and then, we used Naïve Bayes, CNN, LSTM, etc.

- The fifth stage is the Evaluation stage: The model adopted in the previous step will be evaluated to assess its accuracy and performance. Based on the performance evaluation, the previous step is repeated for choosing a different model.
- The sixth stage is the deployment stage: The final phase is to deploy the model for use by public users.

## 5. IMPLEMENTATION

### 5.1 Dataset

The dataset considered for the analysis is the famous Sentiment140 dataset. It is a publicly available dataset that contains 1,600,000 tweets collected by Stanford graduate students (<http://help.sentiment140.com/for-students>). The tweets are labelled as “positive” (4) or “negative” (0). While documentation online says the dataset contains “neutral” (2) tweets as well, upon observation, we found very few (only about 150) tweets that were labelled as “neutral,” so we did not consider them in our model.

Each data entry has the following features:

- target: the polarity of the tweet (negative or positive).
- ids: an integer that represents the id of the tweet;
- date: the timestamp of the tweet;
- flag: the query, if there is no query, then this value is NO\_QUERY;
- user: the username of the user who published the message;
- text: the actual content of the tweet;

For the purpose of this project, the dataset has been adjusted so that the “negative” tweets were given the label “0” while the “positive” tweets were given the label “1”. This decision was made to simplify the tasks of calculating training loss and model accuracy. Due to computational limitations, the models were trained on a subset of the Sentiment140 dataset containing 100,000 positive and negative tweets.

Twitter data may be in an unstructured format that is not suitable for feature extraction. Tweets may include blank spaces, stop words, slang, unique characters, hashtags, emojis, timestamps, abbreviations, URLs, and more. To extract this, we first need to pre-process the data using NLTK's functions. The first goal of pre-processing is to extract messages and remove all hashtags (#), spaces, repeated words, stop words (he, she, they, etc.), emojis, and abbreviations which will be replaced with their corresponding meaning such as :-), =D, LOL. They will be replaced with happiness, laughter, or laughing out loud.

Python code has been written in which the function is defined to obtain pre-processed tweets. The code is used to achieve the following functions:

- Remove quotes: Allow the user to remove the quotes from the tweet.
- Remove @ - Allow users to choose whether to remove the @ symbol, delete @ along with the username, or replace @ and the username with the term 'AT\_USER' and attach it to the stop words.
- Remove URL's - URL stands for uniform resource locator. It allows you to delete URLs or replace them with the term 'URL' and attach to stop words.
- Removal of RT (Re-Tweet) - it removes the RT word from the text.
- Removal of emoticons - Remove emoticons and replace them with their right meanings.
- Removal of duplicates - remove all duplicate words from the tweet.
- Removal of hashtag (#) - delete hashtags from the tweet.
- Removal of stopwords - remove all stopwords from the tweet, such as he, she, and them, because they do not carry meaning in classification.

| Function                          | Action                              |
|-----------------------------------|-------------------------------------|
| Punctuation                       | Deleted                             |
| #word                             | Deleted                             |
| @any_user                         | Deleted                             |
| URLs and web links                | Deleted URLs and added in stopwords |
| All word                          | Lemmatized all words                |
| Number                            | Deleted                             |
| Stop Words                        | Deleted                             |
| Whitespaces                       | Deleted                             |
| Uppercase characters              | Lowercase all words                 |
| Words not starting with alphabets | Deleted                             |
| Emoticons                         | Replaced with respective meaning    |

| Raw Data                                    | Clean Data         |
|---|--------------------|
| @abc123 I love sports and it is the best :) | love, sports, best |

*Table 1: Pre-Processing Techniques*



### 5.1.1 Changing to lower case of text.

This is the most important step because Python is a case-sensitive programming language, so it is imperative to check whether the text is lowercase or uppercase.

This means that the training model interprets uppercase information and lowercase text differently.

### 5.1.2 Tokenization

The text will be divided into tokens in the next phase. To feed the text into our training model, we provide two options: word tokenization or sentence tokenization. We have selected word tokenization.

### 5.1.3 Stemming

Stemming (another name for text standardization) is the next step after stop words removal, where words are returned to their main stem/base form. For example, "Base", "Basement", and "Based" all forward to "Base". The only negative impact of this step is that the sentence/content may lose its meaning once the stemming is complete. Below are some examples of stemming.

| No | Original Word  | Stemmed Word |
|----|----------------|--------------|
| 1  | Establish      | Establish    |
| 2  | Established    | Establish    |
| 3  | Establishment  | Establish    |
| 4  | Establishments | Establish    |
| 5  | Establishing   | Establish    |

| No | Original Word | Stemmed Word |
|----|---------------|--------------|
| 1  | Go            | Go           |
| 2  | Gone          | Go           |
| 3  | Going         | Go           |

*Table 2: Example of Stemming*

### 5.1.4 Lemmatization

In contrast to stemming, this pre-processing step ensures that the meaning of the word is not lost. In the lemmatization step, words are pre-stored in a library and cross-checked during reduction. The following example illustrates the difference between stemming and lemmatization.

| Input Word | Stemming | Lemmatization |
|------------|----------|---------------|
| Change     | Chang    | Change        |
| Changer    | Chang    | Change        |
| Changing   | Chang    | Change        |
| Changed    | Chang    | Change        |
| Changes    | Chang    | Change        |

*Table 3: Difference between Lemmatization and Stemming*

### 5.1.5 WordCloud

An effective approach to convey information is through data visualization, which includes graphs, charts, infographics but what if the raw data is text-based? Using a WordCloud, which is available in the Python programming language, is the answer to this. WordCloud or TagCloud is a cloud-like structure filled with many words in various shapes and sizes. The size of each word indicates the frequency or importance of each word. A larger size indicates more repeated words.



To implement the practical part of this study, Python programming language version 3.6 and open-source libraries including Torch and Keras have been employed. Furthermore, all the experiments have been performed on the Google Colaboratory environment which is a free cloud service that provide free access to resources such as T4 GPU which has 12 GB RAM.

Guido van Rossum created the potent programming language Python in the Netherlands National Institute of Mathematics and Computer Science in the late 1980s. Programmers may write code more effectively and efficiently by using data structures and object-oriented programming techniques, which need less lines of code. There are two major Python versions available till date- Python 2 and Python 3. For the majority of machine learning research and development, Python is the primary programming language. A few highlights that made Python so well-known and the most ideal for machine learning are as follows:

- 50

- Python has community and corporate support.
- Python is portable and extensible.

During this study, NLTK was used to process the linguistic data in Python 3, which is the most Python-compatible language for NLTK.

#### 5.1.8 NLTK

It is a python package that handles human language data. It provides a user-friendly interface to various lexical resources such as WordNet, text processing libraries, etc. These are used to classify, tokenize, stem, tag, and parse human language data. (Loper & Bird, 2002).

#### 5.1.9 Pandas

For data pre-processing and data handling, panda's package is used (pandas, 2023). Pandas provides fast and flexible data structure for the scraped data. The advantage of pandas is its multipurpose function for handling data. All the scraped data for the thesis work will be converted into a data frame, ready for analysis and prediction (pandas, 2023)

#### 5.1.10 NumPy

A NumPy stands for "Numerical Python". This library is used to perform numerical calculations for vector and matrix. It is 50 times faster than list data. This library is used for data analysis and numerical calculations in thesis. (NumPy, n.d.).

#### 5.1.11 Matplotlib

A python library that can be used to create plots, histogram, power spectrum, bar chart, etc. In this study, matplotlib.pyplot module is used for plotting the metrics (matplotlib, n.d.).

### 5.1.12 Sklearn

Scikit-learn provides machine learning and statistical modeling tools and capabilities for classification, clustering, and other predictions. For example, the data is split into train, validation and test subset to create features and tokens for text inputs, and count vectors like frequency count for TF-IDF. For classification, task data is split into train and test (scikitlearn, n.d.).

### 5.1.13 Keras

Like TensorFlow, Keras is an extraordinary, open-source software Application Programming Interface (API) that serves as an interface for the TensorFlow library. It also offers a Python interface for artificial neural networks. When compared to TensorFlow, it is somewhat quicker and more user-friendly. (Géron, 2019).

## 5.2 Analysis of the dataset and pre-processing

### Distribution of data

Before and after the pre-processing, descriptive analysis has been done to identify relevant insights about the length of the tweets and the number of data points used to power the models.

|   | Sentiment | ID         | Date                         | Query    | User_ID         | Tweet   |
|---|-----------|------------|------------------------------|----------|-----------------|---|
| 0 | 0         | 1467810369 | Mon Apr 06 22:19:45 PDT 2009 | NO_QUERY | _TheSpecialOne_ | @switchfoot http://twitpic.com/2y1zl - Awww, t... |
| 1 | 0         | 1467810672 | Mon Apr 06 22:19:49 PDT 2009 | NO_QUERY | scotthamilton   | is upset that he can't update his Facebook by ... |
| 2 | 0         | 1467810917 | Mon Apr 06 22:19:53 PDT 2009 | NO_QUERY | mattycus        | @Kenichan I dived many times for the ball. Man... |
| 3 | 0         | 1467811184 | Mon Apr 06 22:19:57 PDT 2009 | NO_QUERY | ElleCTF         | my whole body feels itchy and like its on fire    |
| 4 | 0         | 1467811193 | Mon Apr 06 22:19:57 PDT 2009 | NO_QUERY | Karoli          | @nationwideclass no, it's not behaving at all...  |

*Figure 15: Raw Sentiment140 dataset*

There is no reason to look for correlation among the features. Therefore, only sentiment and cleaned text are kept for further analysis.

| Sentiment |   | Clean_Tweet  |
|-----------|---|--|
| 0         | 0 | switchfoot http:twitpic.com/yZl awww thats bummer... |
| 1         | 0 | upset cant update facebook texting might cry r...    |
| 2         | 0 | kenichan dived many time ball managed save res...    |
| 3         | 0 | whole body feel itchy like fire                      |
| 4         | 0 | nationwideclass behaving im mad cant see             |

Figure 16: Clean Sentiment140 dataset

As can be seen from the figure, the dataset is perfectly balanced. This has been done by the creators to simplify the further analysis.

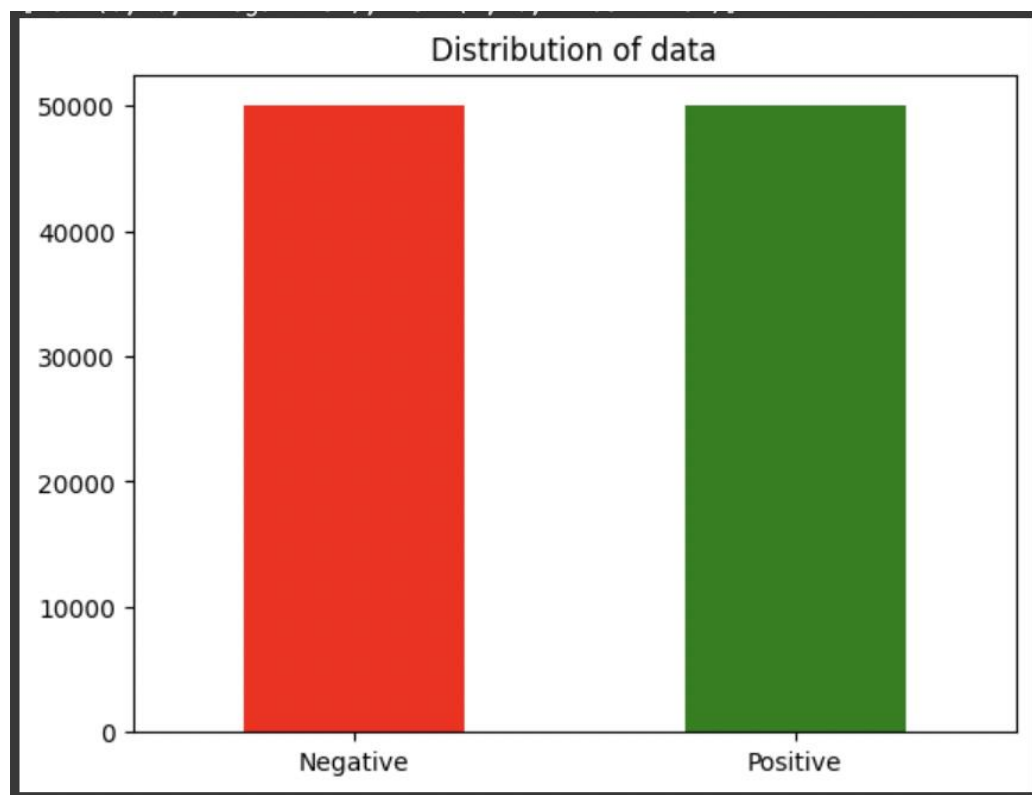


Figure 17: Distribution of data

### Key Statistics about the tweet

To understand the effect of pre-processing, a quantitative analysis has been run before taking any action on the corpus. Table shows the key statistics about the Tweet before pre-processing.

```
Average number of characters: 73.96
Longest tweet characters: 186
Shortest tweet characters: 6
Number of characters of quantile 0.99: 141.00
Average number of words: 13.18
Number of words in the longest tweet: 41
Number of words in the shortest tweet: 1
Number of words of quantile 0.99: 28.00
Number of unique words: 1260915
```

*Figure 18: Key Statistics before pre-processing*

```
Average number of characters: 45.93
Longest tweet characters: 142
Shortest tweet characters: 0
Number of characters of quantile 0.99: 105.00
Average number of words: 7.83
Number of words in the longest tweet: 40
Number of words in the shortest tweet: 0
Number of words of quantile 0.99: 18.00
Number of unique words: 765600
```

*Figure 19: Key Statistics after pre-processing*

As can be seen from above, 99% of the Tweets has used only half of the characters despite the maximum characters length being 280. In the below table, same key statistics are shown after pre-processing of the data. Steps like removal of hashtags, URLs, punctuation, etc has been used to reduce the noise in the data.



[illegible]

Figure 20: WordCloud of Positive Tweets

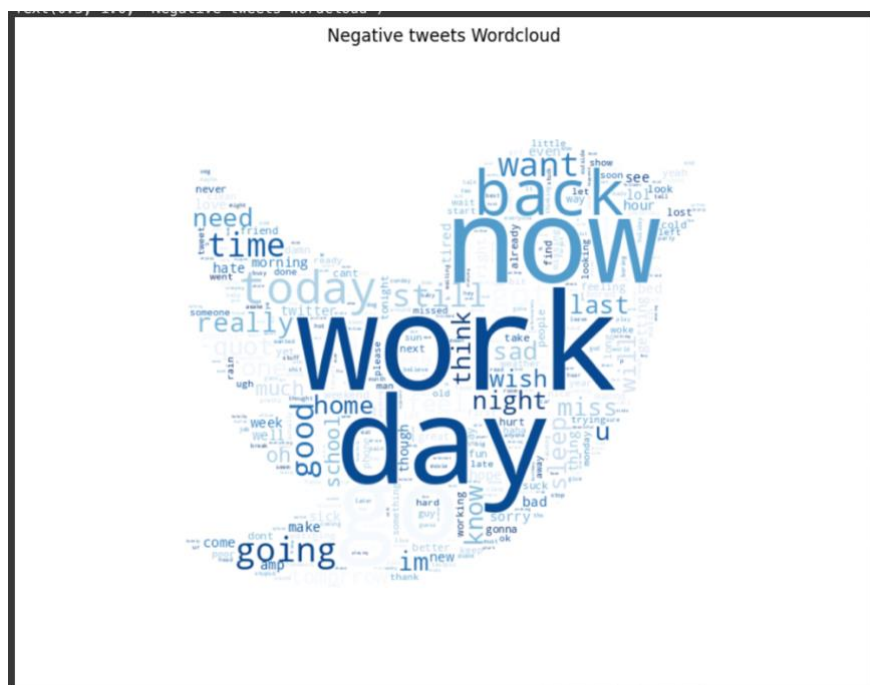


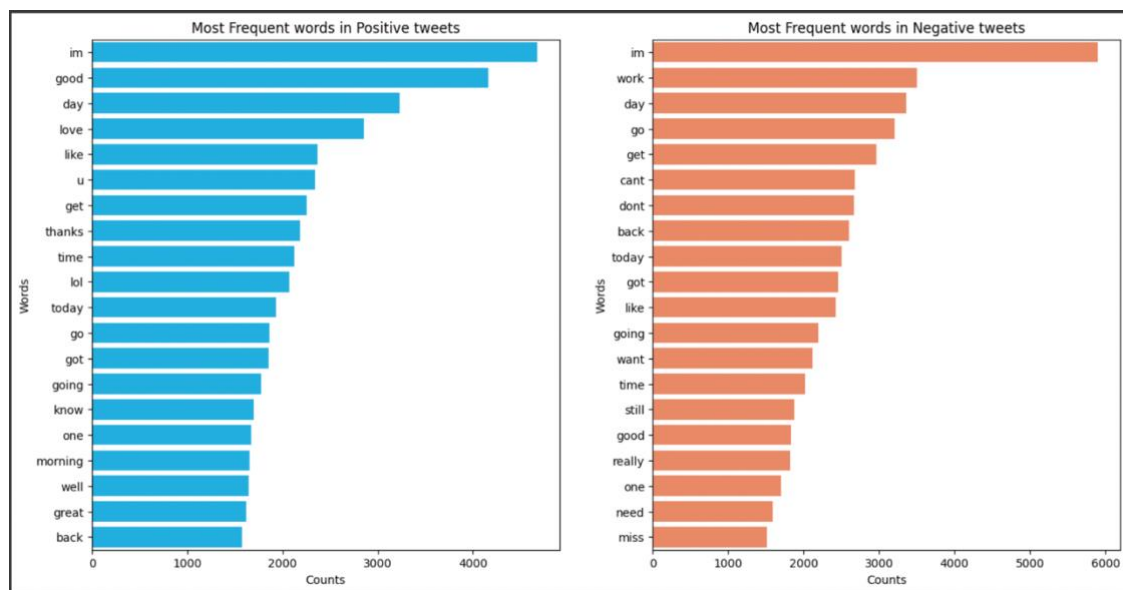
Figure 21: WordCloud of Negative Tweets

The most common words that can be seen from above WordClouds are good, day, time which can be cross-verified from the Bag-of-Words representation as well.



## Wordlist

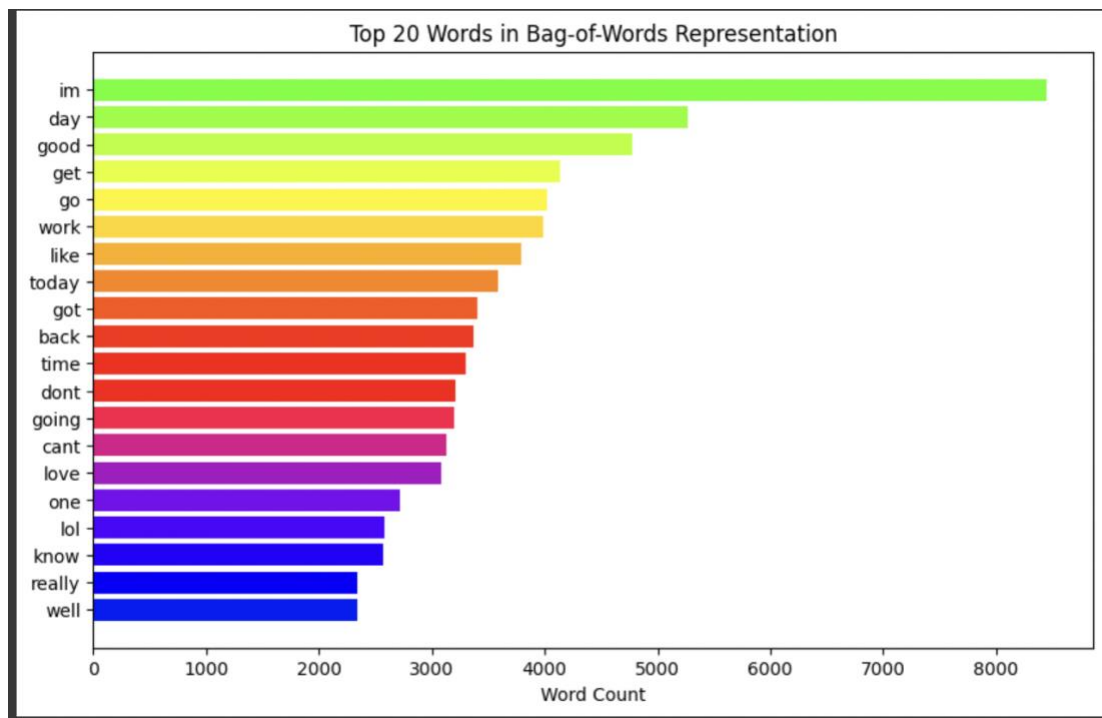
A wordlist is created after text pre-processing and the number of occurrences of each word in the entire training data set is calculated. The most common words that appear in the wordlist are usually the stop words that do not affect the sentiment in the tweet. These stop words are removed, and some words like “not” and “don't” explain the sentiment of the tweet.



*Figure 22: Wordlist of Positive and Negative Tweets*

## Bag of Word (BoW)

The data is prepared to be represented as a bag of words. In a bag of words representation, some common words display the high degree of distinction between themselves such as 'got' and 'back' and other occurring words.



*Figure 23: Bag-of-Words Representation*

## Trends

It seems like people tweet more on average after the usual business hours probably during the end of their day and late into the night. Surprisingly, it seems like the most positive tweets occur at night and when people aren't working, explaining that user behavior and how they seem grumpier during work hours.

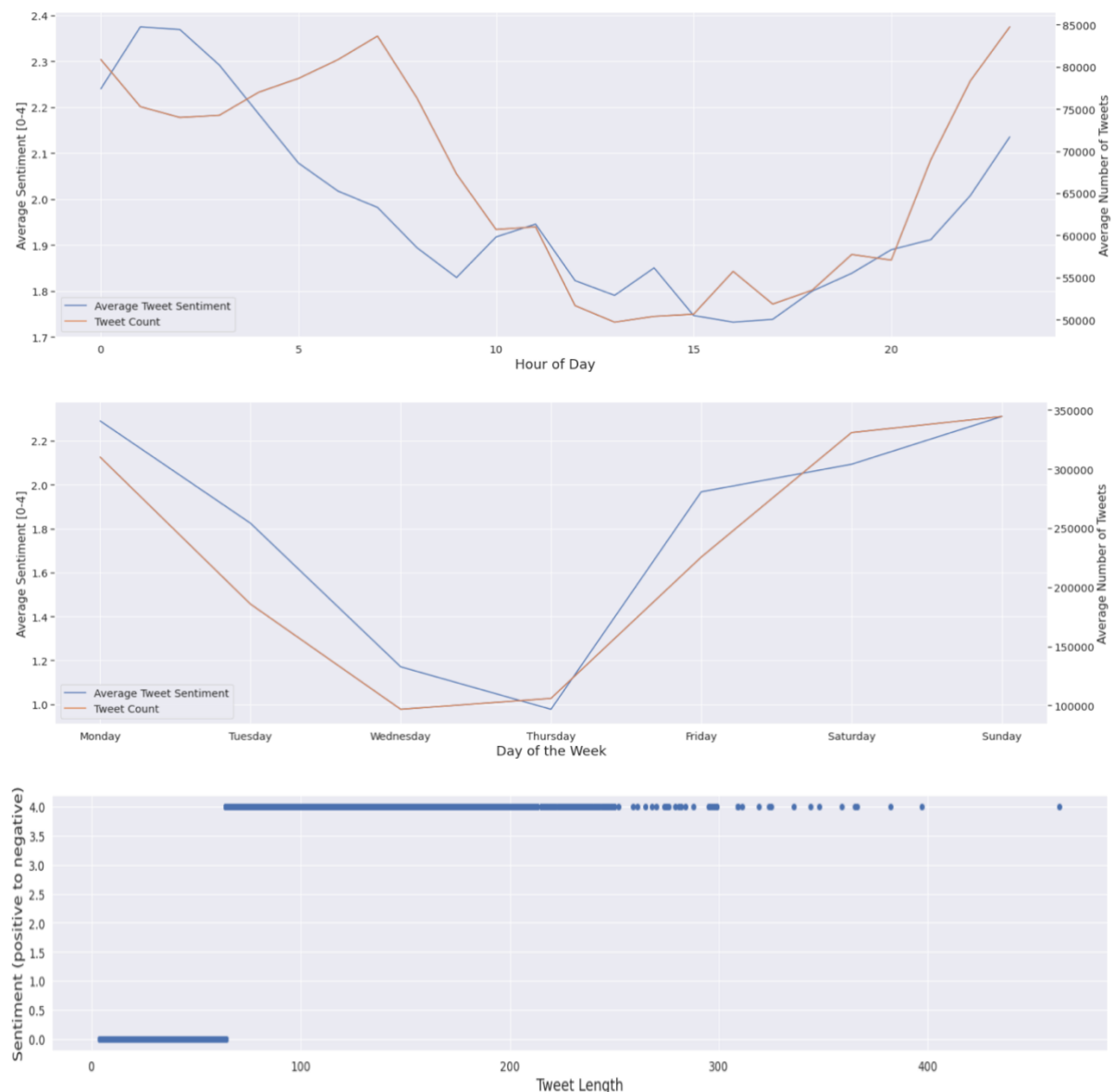


Figure 24: Data Visualization of Tweets

The average amount of tweets trends downwards as people progress into the weekday, and they start tweeting more again as the weekend starts it seems, they're even more positive! Is this because a certain group of positive tweeters come into the mix which are freer on the weekends? Or is it the same set of people but grumpier? A much more thorough analysis is needed to answer that.

## 6. RESULTS and DISCUSSIONS

### 6.1 Traditional Models

As mentioned above, due to computational limitations, the models were trained on a subset of the Sentiment140 dataset containing 100,000 positive and negative tweets. The models were trained on 80% of the sample data and 20% of the training data.

The models were subjected to two rounds of testing to investigate any potential effect of different representation of the data. The data was encoded with CountVectorizer in the first one and the TF-IDF Vectorizer in the second one.

#### 6.1.1 Experiment 1 – Baseline Models

##### Encoding with a CountVectorizer

Table gives information of the results from the baseline models. In this experiment, the models are trained with CountVectorizer. The Logistic Regression model obtains an Accuracy of 77.88%, with a Precision of 78.96%, and Recall Score of 78.57%. Naïve Bayes model obtains a less similar accuracy by 0.81% with the Decision Tree model performing the worst with 71.05% accuracy.

| Models              | Accuracy | Precision | Recall |
|---------------------|----------|-----------|--------|
| Naïve Bayes         | 77.07%   | 77.17%    | 76.16% |
| Random Forest       | 75.55%   | 76.82%    | 75.76% |
| Decision Tree       | 71.05%   | 71.12%    | 71.09% |
| Logistic Regression | 77.88%   | 78.96%    | 78.57% |
| XGBoost             | 74.20%   | 75.16%    | 75.44% |

*Table 4: Traditional Model Performance with CountVectorizer*

Encoding with a TF-IDF Vectorizer

Table gives information of the results from the baseline models. In this experiment, the models are trained with TF-IDF Vectorizer. Similar to the above experiment, The Logistic Regression model performs the best with an Accuracy of 77.57%, with a Precision of 78.14%, and Recall Score of 77.96% followed very closely by Naïve Bayes model which obtains a very similar accuracy of 77.51% with the Decision Tree model again performing the worst with 69.15% accuracy.

| Models              | Accuracy | Precision | Recall |
|---------------------|----------|-----------|--------|
| Naïve Bayes         | 77.51%   | 75.71%    | 74.52% |
| Random Forest       | 76.01%   | 74.12%    | 74.01% |
| Decision Tree       | 69.15%   | 70.44%    | 70.23% |
| Logistic Regression | 77.57%   | 78.14%    | 77.96% |
| XGBoost             | 74.07%   | 74.95%    | 73.76% |

*Table 5: Traditional Model Performance with TF-IDF Vectorizer*

### 6.1.2 Experiment 2 – Hyperparameter Tuning of Baseline Models

Encoding with a CountVectorizer

Table gives information of the results from the baseline models with their respective hypertuning values which can be seen in Appendix. In this experiment, the models are trained with CountVectorizer. The Logistic Regression model obtains an Accuracy of 77.94%, with a Precision of 77.97%, and Recall Score of 78.23%. Naïve Bayes model obtains a less similar accuracy by 0.58% with the Decision Tree model performing the worst with 71.19% accuracy.

| Models              | Accuracy | Precision | Recall |
|---------------------|----------|-----------|--------|
| Naïve Bayes         | 77.36%   | 77.44%    | 76.23% |
| Random Forest       | 75.36%   | 75.12%    | 76.88% |
| Decision Tree       | 71.19%   | 70.44%    | 71.67% |
| Logistic Regression | 77.94%   | 77.97%    | 78.23% |
| XGBoost             | 76.31%   | 76.35%    | 77.01% |

*Table 6: Hypertuned Traditional Model Performance with CountVectorizer*

### Encoding with a TF-IDF Vectorizer

Table gives information of the results from the baseline models with their respective hypertuning values which can be seen in Appendix. In this experiment, the models are trained with TF-IDF Vectorizer. The Logistic Regression model performs the best with an Accuracy of 78.02%, with a Precision of 79.25%, and Recall Score of 78.44%. The worst performing model was Decision Tree with an accuracy of 70.00%

| Models              | Accuracy | Precision | Recall |
|---------------------|----------|-----------|--------|
| Naïve Bayes         | 76.37%   | 74.22%    | 74.46% |
| Random Forest       | 76.34%   | 75.87%    | 76.45% |
| Decision Tree       | 70.00%   | 69.42%    | 71.57% |
| Logistic Regression | 78.04%   | 79.25%    | 78.44% |
| XGBoost             | 75.22%   | 74.27%    | 75.35% |

*Table 7: Hypertuned Traditional Model Performance with TF-IDF Vectorizer*

## 6.2 Neural Models

The models were trained on 60% training set, 20% validation set and 20% training set, training on 60000 samples, validating and testing on 20000 samples each. While

running the models for neural network, it was made sure that the test data is similar to that of traditional models.

### CNN

The CNN model was trained on 5 epochs and we can see that the model performs similar with hypertuning to the above traditional models though it did require a considerable time to build.

| CNN            | Statistics |
|----------------|------------|
| Accuracy       | 77.62%     |
| Precision      | 76.82%     |
| Execution Time | 1061.16 s  |
| Time/epoch     | 212.44 s   |

*Table 8: CNN Model Performance*

### LSTM

The same experiment has been repeated with the LSTM and the performance turned out to be slightly better. It can be seen that the precision rate is better than the CNN model although the LSTM model has taken a significant amount of time to train.

| LSTM           | Statistics |
|----------------|------------|
| Accuracy       | 77.76%     |
| Precision      | 77.16%     |
| Execution Time | 2542.46 s  |
| Time/epoch     | 509.88 s   |

*Table 9: LSTM Model Performance*

### HYBRID MODEL

The model is composed of CNN-BiLSTM and is a more complex than LSTM model. Though it required less time, the performance of the model was worse indicating that it does not make sense to complicate the model any further.

| CNN-LSTM       | Statistics |
|----------------|------------|
| Accuracy       | 76.04%     |
| Precision      | 74.10%     |
| Execution Time | 1861.77 s  |
| Time/epoch     | 373.46 s   |

*Table 10: Hybrid Model Performance*



## 7. PROJECT MANAGEMENT

### 7.1 PROJECT SCHEDULE

Project timeline and clarity on the work items are very important for success of any project. To be able to achieve the primary goal of this project, which was implementing and delivering the outcome for the designated target, a project schedule was created in the early stages, in the form of famous Gantt chart. Detailed project schedule with work breakdown and schedule timeline is provided.

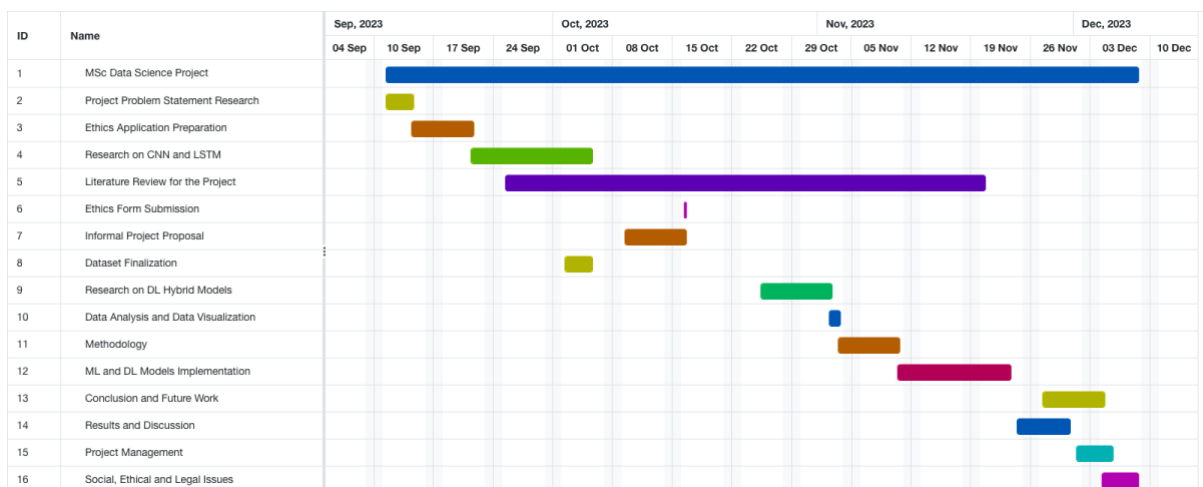


Figure 25: Gantt Chart

### 7.2 RISK MANAGEMENT

Risk Management is the process of project management that mainly focuses on managing risks occurred during the project execution. Identifying such risks in advance and having an appropriate solution for them is an important factor for a successful project management. For that reason, during this project some risks have been identified and possible solutions for them is suggested, which are listed in Table.

| No | Risk      | Mitigation                | Solution  | Impact |
|----|-----------|---------------------------|---|--------|
| 1  | Data Loss | Regular back up on Google | Since the information is gotten from openly available | Medium |

|   |   |  |  |      |
|---|---|--|--|------|
|   |   | Drive and External Hard                                | sources, it is feasible to download it once more. Adjusted variants, then again, should be reproduced. |      |
| 2 | Need of computation power to process huge dataset | Using small and simple sets of data if possible        | Utilizing cloud conditions, for example, Google Collab to acquire free GPU access.                     | High |
| 3 | System Crash                                      | Regular backup of data in SharePoint and local machine | Due to the loss of the report and source code, delivery times were significantly impacted.             | High |

*Table 11: Risk Management*

### 7.3 QUALITY MANAGEMENT

To get a satisfactory result at the project's conclusion, quality control is yet another crucial component of project management operations. Weekly meetings with the supervisor and regular feedback on completed tasks, together with a discussion of the next steps to guarantee that the output is of a satisfactory quality as the project's outcome, were one of the techniques used to keep the project on track. The meeting minutes are available in Appendix B. The dissertation project schedule served as a roadmap for work activities and deadlines, with consistent weekly reviews to make sure all of the tasks were completed on time.

## 7.4 SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL CONSIDERATIONS

As data is the most crucial part of any Machine Learning study, there is always scope for some considerations about the same. The data used in this project have been collected from online source that is publicly available at no cost. The work carried out in this project is in accordance with the UK rules and regulations. This project is created in accordance with criminal laws in terms of theft, privacy, fraud and sabotage. All materials and content contained in this project from adaption to citation content has been created in accordance with the UK and international copyright regulations.

The project is created from a scientific perspective to serve mankind and improve the utilization of technology. The project is not created for commercial purposes, personal gain, or any other business purpose. It is not based on a specific set of values, beliefs, or practices that a person, group, or nation adopts. This project is accessible to everyone from anywhere, regardless of ethical differences between cultures. In summary, the data used in this study will not be processed to cause discrimination of any kind and does not violate any laws such as Human Rights Act 1998 (Article 1, UN 1948)

## 8. CONCLUSIONS AND FUTURE WORKS

### 8.1 CONCLUSION

The main aim of this thesis was to analyse and identify the best fit algorithm for sentiment analysis by using the Sentiment140 dataset. It is only appropriate to conclude this thesis from the results obtained through pre-processing and data analysis and then with the performance of the models thereby addressing the achievements of the research aims.

It can be seen from the data analysis that there is no correlation between the features thereby eliminating the irrelevant columns to improve the data quality and model's performance. The insights from the data analysis shows that 99% of the tweets uses only half of the allowed characters even if the maximum length of the characters is 280. Additionally, pre-processing reduced the number of unique words by 60.71% and the total words decreased by 52.73% which ensured in faster training of the models and more algorithms to be conducted in spite of the low computational resources available.

It can also be seen from that data visualizations plot that more people tend to tweet after working hours and more positively during nighttime which might indicate that people are more relaxed at the end of the data as compared to stressed and frustrated during the day due to personal and professional reasons. The average amount of tweets reduces during the mid-week and increases again as the weekend approaches which again gives insights into the emotions of the people that they are happier and more proactive towards the end of the week as compared to the mid-week when they are less active and might be feeling grumpy. It was also seen that the lengthier the tweet was, the more the tweet was positive which came as a surprise.

Coming to the performance of the models, it can be derived that the traditional models outperformed the neural models by a very small margin. Logistic Regression performed the best amongst all the traditional models with an accuracy of 78.04% and required less time to train. It can also be noticed that the performance of the Logistic Regression model kept on improving across all the experiments while the accuracy of

Decision Trees was the worst with its wavy performance. From the experiments, it can be concluded that encoding with TF-IDF or CountVectorizer did not significantly affect the classification performance or abilities of the models. Regarding the neural models, LSTM with its accuracy of 77.76% was the best model, followed closely by CNN mode, although it requires more time to train the LSTM model.

Lastly, keeping in mind the time needed for the experiments and unavailability of the computational power, it can be concluded that the models with the best potential is Logistic Regression and LSTM.

## 8.2 FUTURE WORKS

According to this thesis, it seems that the traditional models are still one of the best classification algorithms. The neural models, however, have the potential to surpass traditional model, though it requires considerable skills and expertise.

Some of the work that can be included in the future:

- Build models on large data with the help of computational power.
- Work on multi language to provide sentiment analysis to the local.
- Hyperparameter tuning of neural model to increase its efficiency.
- Including emoji labels to figure out if it improves the performance of the models.
- Use more time of text cleaning of pre-processing section which can lead to the better performance oof the model.

## 9. STUDENT REFLECTIONS

One of the major challenges I came across was understanding and implementing the numerous deep learning algorithms used in sentiment analysis. The process required a deep understanding of the algorithms and their application in real-world scenarios. However, I overcame this obstacle by dedicating time to conduct thorough research study about the algorithms while also practicing them, using resources such as online tutorials, textbooks, and peer discussions. This not only improved my understanding about the subject but also enhanced my problem-solving skills.

Project management and in particular time management has been another important learning outcome from this research project. It was difficult to keep self-motivated and at the same time keep the progress of the project as per the plans. I had some deviation during the middle of project due to unforeseen challenge related with computational power need to train the model with huge dataset. But this problem gave me an opportunity to find an alternate way to manage model training with smaller subsets of data and helped me in demonstrating the skill to manage unfavorable conditions during the project execution.

The thesis also required a high level of critical thinking and reflection. I had to constantly evaluate the performance of my models, identify areas of improvement, and be flexible towards necessary change while iterating my approach. This consistent activity taught me the importance of being adaptable and open to changes and constructive feedback, which are valuable skills in any research or professional setting.

Overall, this journey of master's dissertation has been a rewarding and enlightening experience. It has not only enhanced my technical skills but also fostered a growth mindset, resilience, and critical thinking.

## BIBLIOGRAPHY AND REFERENCES

Amrani, Y.A., Lazaar, M., & Kadiri, K. (2018). Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis. *Procedia Computer Science*, 127, 511-520. <https://doi.org/10.1016/j.procs.2018.01.150>

Alsaeedi, A., & Khan, M.Z. (2019). A Study on Sentiment Analysis Techniques of Twitter Data. *International Journal of Advanced Computer Science and Applications*, 10(2), 361-374. DOI: 10.14569/IJACSA.2019.0100248

Alves, A.L., Baptista, C., & Firmino, A.A. (2014). A comparison of SVM Versus Naïve-Bayes Techniques for Sentiment Analysis in Tweets: A Case Study with the 2013 FIFA Confederations Cup. *ACM Digital Library*, 123-130. <https://doi.org/10.1145/2664551.2664561>

Aramaki, E., Maskawa, S., & Morita, M. (2011). Twitter Catches the Flu: Detecting Influenza Epidemics using Twitter. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1568-1576. [https://www.researchgate.net/publication/221012847\\_Twitter\\_Catches\\_The\\_Flu\\_Detecting\\_Influenza\\_Epidemics\\_using\\_Twitter](https://www.researchgate.net/publication/221012847_Twitter_Catches_The_Flu_Detecting_Influenza_Epidemics_using_Twitter)

Back, B., & Ha, I. (2019). Comparison of Sentiment Analysis from Large Twitter Datasets by Naïve Bayes and Natural Language Processing Methods. *Journal of Information & Communication Convergence Engineering*, 17(4), 239-245. <https://web.s.ebscohost.com/abstract?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=22348255&AN=143167735&h=fTBQnzOFCL2YkKvuYGb63OIb5bR1jriP2vm5zG8r%2fmTnnadCdOGY3U%2f8UerOlauawqreYQSAqWILtMUelkk8kA%3d%3d&crl=f&resultNs=AdminWebAuth&resultLocal=ErrCrlNotAuth&crlhashurl=login.aspx%3fdirect%3dtrue%26profile%3dehost%26scope%3dsite%26authtype%3dcrawler%26jrnl%3d22348255%26AN%3d143167735#:~:text=Based%20on%20sentiment%20accuracy%20analysis,yielded%20by%20the%20NLP%20method>

Boiy, E., & Moens, M. (2009). A Machine Learning Approach to Sentiment Analysis in Multilingual Web Texts. *Information Retrieval Journal*, 12(5), 526-558. <https://link.springer.com/article/10.1007/s10791-008-9070-z>

Bramer, M. (2007). Principles of Data Mining. *SpringerLink*, 119-134. [https://www.researchgate.net/publication/220688376\\_Principles\\_of\\_Data\\_Mining](https://www.researchgate.net/publication/220688376_Principles_of_Data_Mining)

Breiman, L. (2004). Random Forests. *SpringerLink*, 45, 5-32. <https://link.springer.com/article/10.1023/A:1010933404324>

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *KDD*, 785-794. <https://doi.org/10.1145/2939672.2939785>

Chollet, F. (2016). Using pre-trained word embeddings in a keras model. *The Keras Blog*. <https://blog.keras.io/using-pre-trained-word-embeddings-in-a-keras-model.html>

colah. (2015). Understanding LSTM Networks. *colah's blog*.  
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

Davidov, D., Tsur, O., & Rappoport, A. (2010). Enhanced sentiment learning using Twitter hashtags and smileys. *Proceedings of the 23<sup>rd</sup> International Conference on Computational Linguistics*, 241-249. <https://dl.acm.org/doi/10.5555/1944566.1944594>

Dubey, A.D. (2020). Twitter Sentiment Analysis during COVID19 Outbreak. *SSRN Electronic Journal*. <http://dx.doi.org/10.2139/ssrn.3572023>

Elbagir, S., & Yang, J. (2018). Sentiment Analysis of Twitter Data Using Machine Learning Techniques and Scikit-learn. *Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence*, 1-5. <https://doi.org/10.1145/3302425.3302492>

Fitri, V.A., Andreswari, R., & Hasibuan, M.A. (2019). Sentiment Analysis of Social Media Twitter with case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm. *Procedia Computer Science*, 161, 765-772. <https://doi.org/10.1016/j.procs.2019.11.181>

Gautam, G., & Yadav, D. (2014). Sentiment Analysis of Twitter Data using Machine Learning Approaches and Semantic Analysis. *7<sup>th</sup> International Conference on Contemporary Computing*, 437-442. DOI: 10.1109/IC3.2014.6897213

Géron, A. (2019). Hands-on Machine Learning with Scikit-learn, Keras and Tensorflow. *O'Reilly Media*.  
[https://powerunit-ju.com/wp-content/uploads/2021/04/Aurelien-Geron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow\\_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-OReilly-Media-2019.pdf](https://powerunit-ju.com/wp-content/uploads/2021/04/Aurelien-Geron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-OReilly-Media-2019.pdf)

Ghahramani, Z., & Jordan, M. (1996). Supervised learning from incomplete data via an EM approach. *Advances in the Neural Information Processing Systems*, 120-127. [https://www.researchgate.net/publication/2648965\\_Supervised\\_learning\\_from\\_incomplete\\_data\\_via\\_an\\_EM\\_approach](https://www.researchgate.net/publication/2648965_Supervised_learning_from_incomplete_data_via_an_EM_approach)

Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *Stanford University*.  
<https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf>

Goularas, D., & Kamis, S. (2019). Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data. *International Conference on Deep Learning and Machine Learning in Emerging Applications*. DOI: 10.1109/Deep-ML.2019.00011

Gurkhe, D., Pal, N., & Bhatia, R. (2014). Effective Sentiment Analysis of Social Media Datasets using Naïve Bayesian Classification. *International Journal of Computer Application*, 99(13).  
<https://research.ijcaonline.org/volume99/number13/pxc3898274.pdf>

Hatzivassiloglou, V., & McKeown, K.R. (1997). Predicting the semantic orientation of adjectives. *Proceedings of the 35<sup>th</sup> Annual Meeting of the Association for*



*Computational Linguistics*, 174-181.  
<https://dl.acm.org/doi/pdf/10.3115/976909.979640>

Hemalatha, I., Varma, G.P., & Govardhan, A. (2014). Case Study on Online Reviews Sentiment Analysis Using Machine Learning Algorithms. *IJIRCCE*, 2(2), 3182-3188.  
[https://www.researchgate.net/publication/334672115\\_Case\\_Study\\_on\\_Online\\_Reviews\\_Sentiment\\_Analysis\\_Using\\_Machine\\_Learning\\_Algorithms](https://www.researchgate.net/publication/334672115_Case_Study_on_Online_Reviews_Sentiment_Analysis_Using_Machine_Learning_Algorithms)

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780.  
[https://www.researchgate.net/publication/13853244\\_Long\\_Short-term\\_Memory](https://www.researchgate.net/publication/13853244_Long_Short-term_Memory)

Hu, M., & Liu, B. (2004). Mining and Summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168-177. <https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf>

Hunter, J., & Dale, D. (n.d.). Matplotlib Documentation. *matplotlib*.  
<https://matplotlib.org/stable/>

Isah, H., Neagu, D., & Trundle, P. (2015). Social Media Analysis for Product Safety using Text Mining and Sentiment Analysis. *UKCI*. DOI: 10.1109/UKCI.2014.6930158

Jain, A.P. & Dandannavar, P. (2016). Application of machine learning techniques to sentiment analysis. *iCATct*. <https://ieeexplore.ieee.org/document/7912076>

Jain, D., Garg, A., & Saraswat, M. (2019). Sentiment Analysis using Few Short Learning. *ICIIP*. DOI: 10.1109/ICIIP47207.2019.8985855

James, G., Witten, D., & Hastie, T. (2013). An Introduction to Statistical Learning with Applications in R. *SpringerLink*.  
<https://link.springer.com/book/10.1007/978-1-4614-7138-7>

Jeenanunta, C., Chaysiri, R., & Thong, L. (2018). Stock Price prediction with Long Short-Term Memory Recurrent Neural Network. *ICESIT-ICICTES*. DOI: 10.1109/ICESIT-ICICTES.2018.8442069

Kharde, V., & Sonawane, S. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications*, 139(11), 5-15.  
<https://www.ijcaonline.org/research/volume139/number11/kharde-2016-ijca-908625.pdf>

Kroeze, J.H., Matthee, M.C., & Bothma, T. (2003). Differentiating data-and-text-mining methodology. *Proceedings of SAICSIT 2003*, 93-101.  
[https://www.researchgate.net/publication/228541509\\_Differentiating\\_data-and\\_text-mining\\_terminology](https://www.researchgate.net/publication/228541509_Differentiating_data-and_text-mining_terminology)

Kumar, A., & Jaiswal, A. (2019). Systematic literature review of sentiment analysis on Twitter using soft computing techniques. *Concurrency and Computation Practice and Experience*, 32(4). <https://onlinelibrary.wiley.com/doi/10.1002/cpe.5107>

Leskovec, J., Rajaraman, A., & Ullman, J.D. (2010). Mining of Massive Datasets. <http://infolab.stanford.edu/~ullman/mmds/book.pdf>

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 1-167.

Loper, E., & Bird, S. (2002). NLTK: The Natural Language Toolkit. *arXiv*. <https://doi.org/10.48550/arXiv.cs/0205028>

Majumder, N., Poria, S., & Gelbukh, A. (2017). Deep Learning-Based Document Modeling for Personality Detection from text. *IEEE Intelligent Systems*, 32(2), 74-79. <https://ieeexplore.ieee.org/document/7887639>

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093-1113. <https://www.sciencedirect.com/science/article/pii/S2090447914000550>

Mertiya, M. & Singh, A. (2016). Combining naïve bayes and adjective analysis for sentiment detection on Twitter. *ICICT*. <https://ieeexplore.ieee.org/document/7824847>

Meville, P., Gryc, W., & Lawrence, R.D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. *Proceedings of the 15<sup>th</sup> AGM SIGKDD*, 1275-1284. <https://doi.org/10.1145/1557019.1557156>

Mukku, S.S., Oota, S.R., & Mamidi, R. (2017). Tag Me a Label with Multi-Arm: Active Learning for Telugu Sentiment Analysis. *Big Data Analytics and Knowledge Discovery*, 355-367. [https://www.researchgate.net/publication/318487690\\_Tag\\_me\\_a\\_Label\\_with\\_Multi-Arm\\_Active\\_Learning\\_for\\_Telugu\\_Sentiment\\_Analysis](https://www.researchgate.net/publication/318487690_Tag_me_a_Label_with_Multi-Arm_Active_Learning_for_Telugu_Sentiment_Analysis)

Nandakumar, R., Uttamraj, K.R., & Lokeshwari, Y.V. (2018). Stock Price prediction using Long Short-Term Memory. *IRJET*. <https://www.irjet.net/archives/V5/i3/IRJET-V5I3788.pdf>

Neri, F., Aliprandi, C., & Capeci, F. (2012). Sentiment Analysis on Social Media. *IEEE*. <https://ieeexplore.ieee.org/document/6425642>

NumPy Developers (n.d.). NumPy Documentation. *NumPy*. <https://numpy.org/doc/stable/>

Olsson, F. (2009). A literature survey of active machine learning in the context of natural language processing. [https://www.researchgate.net/publication/228682097\\_A\\_literature\\_survey\\_of\\_active\\_machine\\_learning\\_in\\_the\\_context\\_of\\_natural\\_language\\_processing](https://www.researchgate.net/publication/228682097_A_literature_survey_of_active_machine_learning_in_the_context_of_natural_language_processing)

Ouyang, X., Zhou, P., & Li, C. (2015). Sentiment Analysis using Convolutional Neural Networks. *IEEE International Conference on Computer and Information Technology*. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.349

Pagolu, V.S., Reddy, K.N., & Panda, G. (2016). Sentiment Analysis of Twitter Data for predicting stock market movements. *SCOPES*. DOI: 10.1109/SCOPES.2016.7955659

Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010*, 17-23. [http://www.lrec-conf.org/proceedings/lrec2010/pdf/385\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2010/pdf/385_Paper.pdf)

pandas. (2023). Pandas Documentation. *pandas*. <https://pandas.pydata.org/docs/>

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical Methods in natural language processing*, 10, 79-86. <https://doi.org/10.3115/1118693.1118704>

Pavalanathan, U., & Eisentein, J. (2016). Emoticons vs Emojis on Twitter: A Casual Inference Approach. *arXiv*. <https://doi.org/10.48550/arXiv.1510.08480>

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2012). Scikit-learn. Machine Learning in Python. *Journal of Machine Learning Research*, 2825-2830. [https://www.researchgate.net/publication/51969319\\_Scikit-learn\\_Machine\\_Learning\\_in\\_Python](https://www.researchgate.net/publication/51969319_Scikit-learn_Machine_Learning_in_Python)

Ponte, J.M., & Croft, W.B. (1998). A language modeling approach to information retrieval. *Proceedings of the 21<sup>st</sup> annual international ACM SIGIR conference on Research and development in information retrieval*, 275-281. <https://doi.org/10.1145/290941.291008>

Poria, S., Cambria, E., & Ku, L. (2014). A Rule-Based Approach to Aspect Extraction from Product Reviews. *Association for Computational Linguistics and Dublin City University*, 28-37. DOI:10.3115/v1/W14-5905

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2), 143-157. <https://doi.org/10.1016/j.joi.2009.01.003>

Quinlan, J.R. (1986). Induction of Decision Trees. *SpringerLink*, 1, 81-106. <https://link.springer.com/article/10.1023/A:1022643204877>

Ren, X., Guo, H., & Li, S. (2017). A Novel Image Classification Method with CNN-XGBoost Model. *International Workshop on Digital Watermarking*, 378-390. DOI:10.1007/978-3-319-64185-0\_28

Rout, J.K., Choo, K.R., & Dash, A.K. (2018). A model for sentiment and emotion analysis of unstructured social media text. *Electronic Commerce Research*, 18, 181-199. <https://link.springer.com/article/10.1007/s10660-017-9257-8>

Sadia, A., Khan, F., & Bashir, F. (2018). An Overview of Lexicon-based Approach for Sentiment Analysis. *IEEC*. [https://ieec.neduet.edu.pk/2018/Papers\\_2018/15.pdf](https://ieec.neduet.edu.pk/2018/Papers_2018/15.pdf)

Samuel, J., Ali, G.G., & Rahman, M. (2020). COVID-19 Public Sentiment Insights and Machine Learning for Tweets Classification. *MDPI*. <https://www.mdpi.com/2078-2489/11/6/314>

scikit-learn (n.d.). Scikit-learn Documentation. *scikit-learn*. <https://scikit-learn.org/stable/>

Shi, H., & Li, S. (2011). A Sentiment analysis model for hotel reviews based on supervised learning. *International Conference on Machine Learning and Cybernetics*. DOI: 10.1109/ICMLC.2011.6016866

Siddiqua, U.A., Ahsan, T., & Chy, A.N. (2016). Combining a rule-based classifier with ensemble of feature sets and machine learning techniques for sentiment analysis on microblog. *2016 19<sup>th</sup> International Conference on Computer and Information Technology*. <https://ieeexplore.ieee.org/document/7860214>

Silva, N., Hruschka, E.R., & Hruschka.Jr, E.R. (2014). Tweet Sentiment Analysis with Classifier Ensembles. *Decision Support Systems*, 66, 170-179. <https://doi.org/10.1016/j.dss.2014.07.003>

Thet, T.T., Na, J., & Khoo, C.S. (2010). Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science*, 36(6), 823-848. DOI:10.1177/0165551510388123

Thormundsson, B. (2022). Revenues from the natural language processing (NLP) market worldwide from 2017 to 2025. *statista*. <https://www.statista.com/statistics/607891/worldwide-natural-language-processing-market-revenues/>

Tolba, A., & Makhadmeh, Z. (2020). Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. *SpringerLink*, 501-522. <https://link.springer.com/article/10.1007/s00607-019-00745-0>

Tong, S., & Koller, D. (2001). Support Vector Machine Active Learning with Applications to Text Classification. *Journal of Machine Learning Research*, 2(1), 45-66. <https://www.jmlr.org/papers/volume2/tong01a/tong01a.pdf>

Turney, P.D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40<sup>th</sup> Annual Meeting on Association for Computational Linguistics*, 417-424. <https://doi.org/10.3115/1073083.1073153>

United Nations. (n.d.). Universal Declaration of Human Rights. *United Nations*. <https://www.un.org/en/about-us/universal-declaration-of-human-rights>

Visa, S., Ramsay, B., & Ralescu, A. (2011). Confusion Matrix-based Feature Selection. *Proceedings of the 22<sup>nd</sup> Midwest Artificial Intelligence and Cognitive Science Conference*. [https://www.researchgate.net/publication/220833270\\_Confusion\\_Matrix-based\\_Feature\\_Selection](https://www.researchgate.net/publication/220833270_Confusion_Matrix-based_Feature_Selection)

Vishwakarma, D.K., Varshney, D., & Yadav, A. (2019). Detection and veracity analysis of fake news via scrapping and authenticating the web search. *Cognitive Systems Research*, 58, 217-229. <https://doi.org/10.1016/j.cogsys.2019.07.004>

Wankhede, R., & Thakare, A.N. (2017). Design approach for accuracy in movies reviews using sentiment analysis. *ICECA*. DOI: 10.1109/ICECA.2017.8203652

Wankhede, S., Patil, R., & Sonawane, S. (2018). Data Preprocessing for Efficient Sentimental Analysis. *ICICCT*. DOI: 10.1109/ICICCT.2018.8473277

Wu, H., & Toutanova, K. (2014). A Convolutional Neural Network for Modelling Sentences. *Proceeding of the 52<sup>nd</sup> Annual Meeting of the Association for Computational Linguistics*, 655-665.  
<https://aclanthology.org/P14-1062/>

Younis, E.M. (2015). Sentiment Analysis and Text Mining for Social Media Microblogs using Open Source Tools: An Empirical Study. *International Journal of Computer Applications*, 112.  
[https://www.researchgate.net/publication/272463313\\_Sentiment\\_Analysis\\_and\\_Text\\_Mining\\_for\\_Social\\_Media\\_Microblogs\\_using\\_Open\\_Source\\_Tools\\_An\\_Empirical\\_Study](https://www.researchgate.net/publication/272463313_Sentiment_Analysis_and_Text_Mining_for_Social_Media_Microblogs_using_Open_Source_Tools_An_Empirical_Study)

Yu, Y., Si, X., & Hu, C. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architecture. *MIT Press*, 31(7), 1235-1270.  
<https://ieeexplore.ieee.org/document/8737887>

Zhang, H. (2004). The optimality of Naïve Bayes. *FLAIRS*, 562-567.  
[https://courses.ischool.berkeley.edu/i290-dm/s11/SECURE/Optimality\\_of\\_Naive\\_Bayes.pdf](https://courses.ischool.berkeley.edu/i290-dm/s11/SECURE/Optimality_of_Naive_Bayes.pdf)

Zhang, L., Wang, S., & Liu, B. (2018). Deep Learning for Sentiment Analysis: A Survey. *Computation and Language*.  
<https://doi.org/10.48550/arXiv.1801.07883>

## APPENDIX A: MEETING RECORDS

| Meeting No. | Topics Discussed  | Date       | Time                |
|-------------|---|------------|---------------------|
| 1           | <ul style="list-style-type: none"> <li>Introducing the Project Subject and discussing about the project objectives and expected outcome with Prof. Long Chen (Temporary Supervisor).</li> <li>Supervisor suggested new techniques such as Zero-shot learning to explore and familiarize with its application.</li> </ul>  | 02/10/2023 | 10:30 AM – 11:00 AM |
| 2           | <ul style="list-style-type: none"> <li>Introduction with Dr. Lakhvir Singh as my new supervisor.</li> <li>Provided updates regarding the discussion with Prof. Long Chen and latest developments with the project.</li> <li>Sought approval for the selected dataset to be used in the project.</li> <li>Discussion regarding the ethics application process and the contents to submit for approval</li> </ul> | 11/10/2023 | 10:00 AM – 10:30 AM |
| 3           | <ul style="list-style-type: none"> <li>Supervisor suggested to prepare an informal research proposal which will serve as a tool to monitor the research progress.</li> <li>Discussion regarding the classification algorithms to be used in the project.</li> <li>Conversation on the background researches related to the project.</li> </ul>  | 17/10/2023 | 10:00 AM – 10:30 AM |
| 4           | <ul style="list-style-type: none"> <li>Discussion and decide to use CNN and LSTM models as the algorithms for the project.</li> <li>Sought feedback and review regarding the research proposals</li> </ul>  | 30/10/2023 | 11:00 AM – 11:30 AM |

|   |  |            |                     |
|---|--|------------|---------------------|
|   | <ul style="list-style-type: none"> <li>• Discussion regarding the pre-processing techniques to be used for the project.</li> </ul>   |            |                     |
| 5 | <ul style="list-style-type: none"> <li>• Discussion on the performance of the models.</li> <li>• Discussion regarding the solutions to adopt due to longer time in training the models.</li> <li>• Have been advised to reach out to the HPC facility for resources to complete the project</li> </ul> | 22/11/2023 | 10:00 AM – 10:30 AM |
| 6 | <ul style="list-style-type: none"> <li>• Discussion on what steps to be taken due to the complexity in using HPC.</li> <li>• Have been advised to use reduced dataset and run the models on local host.</li> <li>• Sought clarification on each weightage section of the project.</li> </ul>           | 27/11/2023 | 11:00 AM – 11:30 AM |
| 7 | <ul style="list-style-type: none"> <li>• Discussion on the results obtained from the models.</li> <li>• Suggested to compare the results and observe the behaviour of all the models.</li> </ul>   | 1/12/2023  | 3:00 PM – 3:30 PM   |

## APPENDIX B: HYPERPARAMETERS OF ML MODELS

| Model               | Hyperparameters  | Hyperparameter Search Space  | Optimal Value   |
|---------------------|--|--|---|
| Random Forest       | <ul style="list-style-type: none"> <li>• N_estimators</li> <li>• Criterion</li> <li>• Min_Samples_Split</li> <li>• Min_Samples_Leaf</li> </ul> | <ul style="list-style-type: none"> <li>• 100, 200, 500</li> <li>• Gini, Entropy</li> <li>• 1, 2, 5</li> <li>• 1, 2, 5</li> </ul> | <ul style="list-style-type: none"> <li>• 100</li> <li>• Gini</li> <li>• 2</li> <li>• 1</li> </ul> |
| Decision Tree       | <ul style="list-style-type: none"> <li>• Criterion</li> <li>• Max_depth</li> <li>• Min_Samples_Split</li> </ul>                                | <ul style="list-style-type: none"> <li>• Gini, Entropy</li> <li>• 1, 2, 4</li> <li>• 1, 2, 5</li> </ul>                          | <ul style="list-style-type: none"> <li>• Gini</li> <li>• 2</li> <li>• 2</li> </ul>                |
| Logistic Regression | <ul style="list-style-type: none"> <li>• Solver</li> <li>• C</li> <li>• Max_iter</li> </ul>  | <ul style="list-style-type: none"> <li>• Lbfgs, Liblinear, Saga</li> <li>• 1, 2, 3</li> <li>• 100, 500, 1000</li> </ul>          | <ul style="list-style-type: none"> <li>• Saga</li> <li>• 2</li> <li>• 1000</li> </ul>             |
| XGBoost             | <ul style="list-style-type: none"> <li>• Learning_rate</li> <li>• N_estimators</li> <li>• Max_Depth</li> </ul>                                 | <ul style="list-style-type: none"> <li>• 0.001, 0.01, 1</li> <li>• 100, 200, 500</li> <li>• 2, 4, 6</li> </ul>                   | <ul style="list-style-type: none"> <li>• 1</li> <li>• 200</li> <li>• 6</li> </ul>                 |



## APPENDIX C: ETHICS APPROVAL CERTIFICATE

|   |  |             |
|---|--|-------------|
| Tweets and Emotions: Exploring Sentiment Analysis on Twitter  |  | P165426     |
|   |  |             |
| <h3>Certificate of Ethical Approval</h3>  |  |             |
| Applicant:  | Vinay Gurrapp  |             |
| Project Title:  | Tweets and Emotions: Exploring Sentiment Analysis on Twitter |             |
| <p>This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk</p> |  |             |
| Date of approval:   | 15 Oct 2023  |             |
| Project Reference Number:   | P165426  |             |
| <hr/>   |  |             |
| Vinay Gurrapp (7150CEM)   | Page 1   | 15 Oct 2023 |