MCAST

An Empirical Study of Sentiment Analysis Techniques for Evaluating the Sentiment of Football-Related Tweets on Social Media

Clayton Borda

Supervisor: Carlo Mamo

June - 2023

A dissertation submitted to the Institute of Information and Communication Technology in partial fulfilment of the requirements for the degree of BSc (Hons) Multimedia in Software Development

Authorship Statement

This dissertation is based on the results of research carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

The research was carried out under the supervision of (name of dissertation tutor –Title, Name and surname)

. . . .

Date Signature

Copyright Statement

In submitting this dissertation to the MCAST Institute of Information and Communication Technology, I understand that I am giving permission for it to be made available for use in accordance with the regulations of MCAST and the Library and Learning Resource Centre. I accept that my dissertation may be made publicly available at MCAST's discretion.

....

Date Signature

Acknowledgements

The list of people that the Student would like to thank on the completion of the dissertation. For example ‘Mr Name Surname, who supported me during my dissertation work as my tutor’.

Abstract

This section should clearly state what the study is about, summarizing how it was carried out and what the results were. References are not to be included in the abstract. It should present only the essentials of the work in general.

Keywords: Dissertation, keywords.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

NN	Neural Network
ML	Machine Learning
DL	Deep Learning
FCN	Fully Convolutional Network
CNN	Convolutional Neural Network

Chapter 1: Introduction

In this section, **you**, the Student, are expected to state clearly:

(a) the 'problem' or 'question' being researched;

(b) why this topic was chosen;

(c) what motivated the you to choose this topic;

(d) why did you investigate the topic the way you did;

(e) what problem did the you wish to explore;

(f) what is the context for the research?

Percentage amount of words in section: 10 % of Dissertation*

1.1 Sub-chapter One

Background goes here. Also you can put in some references [**?**].

Another example of citations [**?**, **?**].

Here is a sample of table in Table 1.1

Age Groups	Frequency	Percent	Valid Percent	Cumulative Percent
16-20 years	**100**	**98.2**	**98.2**	**95.2**
21-25 years	**5**	**4.8**	**4.8**	**100.0**
Total	**105**	**100.0**	**100.0**	

Table 1.1: *Age of Participants*

Use `\newpage` to force start a new page.

A very quick way to create tables in a point & click environment is to use an online table generator[1]

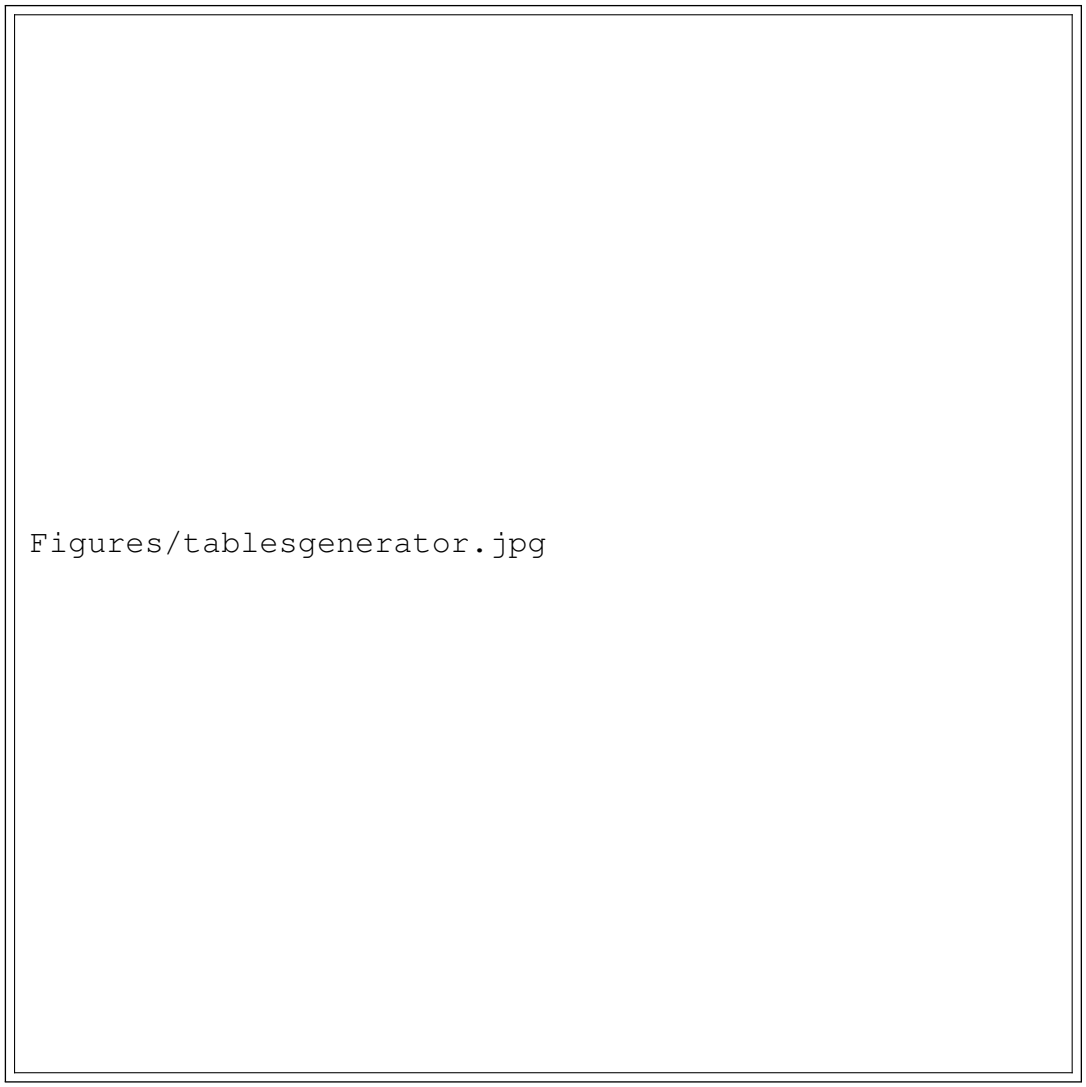

Figure 1.1: *Create a Booktabs style table*

Use `\enquote` for double-quotes. “This is a sample quote.”

Also can try to refer to this image in Figure 1.2. Notice that the `.eps` and `.pdf` format vector graphs are favoured, because:

1. they can be zoomed-in to check the detail.

2. text in such formats are search-able.

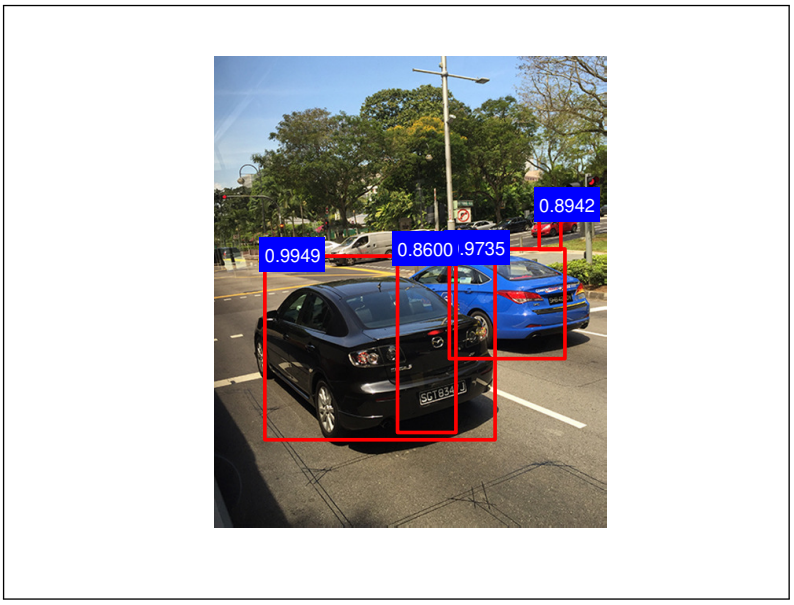

Figure 1.2: *Bounding-box example of cars.*

Try to insert a math equation as in Equation 1.1. If you wanna try the in-line mathematical, here is a sample $\alpha = \pi \cdot \frac{1}{\Theta}$.

$$e^{ix} = \cos x + i \sin x \tag{1.1}$$

When mention some file formats can use `music.mp3`, `latex.pdf`, etc.

1.2 Sub-chapter Two

This is a todo note which appears in the margin

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections

[1] https://www.tablesgenerator.com

1.10.32 and 1.10.33 of de Finibus Bonorum et Malorum (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, “Lorem ipsum dolor sit amet”.., comes from a line in section 1.10.32.

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using ’Content here, content here’, making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for ’lorem ipsum’ will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

1.3 Sub-chapter Three

This todo note appears in line with the text. Todo notes are a great tool to leave comments or notes for self and can then be simply commented out.

Vivamus eget odio tellus. Nam libero augue, eleifend molestie est eu, tempus vehicula magna. Sed dignissim imperdiet urna, non viverra risus blandit in. In lacinia aliquet leo, ut interdum ipsum ornare sed. Praesent ac fringilla justo. Vivamus pharetra non ipsum eget semper. In hac habitasse platea dictumst. Donec neque lectus, ultricies id massa posuere, sodales interdum ante. Sed et nunc vitae urna ornare porttitor nec vitae urna. Curabitur sit amet luctus urna. Nulla porta malesuada rutrum. Pellentesque vitae velit sed odio tincidunt iaculis. Nulla

facilisi.

1.4 Sub-chapter Four

To create a bulleted list you need to make use of the ”itemize” environment. A LaTeXenvironment is a special section within a page where items as placed. An environment always has a “\begin” and a “\end”. Items you can place in such an environment include:

- enumerate – used to create numbered/letter lists
- itemize – creates a bullet list
- figures – to add images or figures (.eps is the recommended format
- tables – you can create them in tablesgenerator.com then copy the LaTeX code and paste it.
- equations – for those really neat looking mathematical formulae
- this list is not exhaustive...

1.5 Sub-chapter Five

The following are some of the most common commands used during your writing. Some characters are reserved and cannot be directly written as in a normal word processor but need to be "escaped". Then there are other useful commands such as adding a footnote, inserting a new line (to begin a new paragraph after a blank line, adding labels to items for cross-referencing, etc.

- The following are special characters which require a \ before each character so the character is reproduced on the output. The tilde and exponent are even more special.
  - \#
  - \$
  - \% – comment
  - \&
  - _
  - ~– type \textasciitilde
  - ^– type \textasciicircum
- \textbf – bold font
- \textit – italics
- \underline{words to underline}

- \emph – emphasized text. If used within an italicized text, the output is normal font. If used by itself, text in its argument is italicized.

An example highlighting the use of \emph.

- An example of emphasized text with *these words emphasized* within normal font.

- *This is an example where* these three words *are emphasized within italicized font*

Chapter 2: Literature Review

2.1 Introduction

In the late 2000s, the introduction and subsequent widespread use of social media, created a trustworthy forum for the public and organizations to disseminate their opinions and facts. Moreover, the sheer size and accessibility of platforms such as Twitter enables a broader public perspective that encompasses a vast array of subject areas, making it a rich resource for text mining and sentiment analysis, in which it provides the capacity to use Twitter-generated data to generate future event insights [1].

In machine learning, a variety of methodologies are employed to forecast football matches and generate a sentiment score that reflects the tenor of a given text. Researchers leverages the disciplines of sentiment analysis and machine learning to predict the tine of a text or tweet and the eventual outcomes of the event. This section will concentrate on the procedures and conclusion derived from academic pursuits aimed at making these predictions.

2.2 Sentiment Analysis

Sentiment analysis has been one of the most prominent and intriguing research topics in Natural Language Processing in the 21st century. It is crucial in comprehending individuals' responses to a particular subject, and in determining whether the feedback received is positive, negative, or neutral [2]. However, some critical

parameters may prove challenging to detect or quantify due to their ever-evolving nature. The inability to accurately identify these key elements can sometimes result in systems relying on extraneous data and, therefore, being hindered in their functioning.

The utilization of Crowd Sourcing as a forecasting tool aims to mitigate the difficulties in selecting and assigning weight to criteria. James Surowiecki's book says that large groups of individuals are more effective than domain experts in making forecasts under uncertainty [3]. The underlying premise behind this notion is that crowds possess a more extensive base of knowledge [4].

2.2.1 Third-party Observations in Sentiment Analysis

Detecting sentiment analysis in Tweets differs significantly from detecting sentiment in conventional text. Due to the informalities of a social media site like Twitter, researchers encounter several unique difficulties. The generated material is constantly developing and incredibly dynamic, with Tweets having a set character restriction in which the limit has been increased from 140 to 280 characters as of November 2017.

This area has garnered significant attention from academics, who view it as a crucial venue for data mining. Data mining is a logical process that is used to search through large amount of data in order to find useful information [5]. The context of the German federal election to assess the utilization of Twitter as a forum for political discourse was studied to determine if online statements on the platform accurately reflect offline political expressions. The results indicated that the volume of tweets mentioning a political party can serve as a reliable

indicator of the party's vote share, and its predictive power is comparable to that of traditional electoral polls [6].

A recent study conducted on the relationship between social media attention and movie sales has revealed that Twitter tweets serve as a reliable predictor of movie sales. The research found that positive sentiment expressed in tweets corresponded with an uptick in movie sales, while negative sentiment was associated with decreased movie sales. Furthermore, the findings revealed that tweets explicitly expressing the desire to watch a particular movie were the most accurate indicators of future sales, as they revealed the viewer's intention to watch the movie [7].

2.2.2 Sentiment Analysis in Sports

As social media grew rapidly, fans write tweets to communicate their own feelings, primarily regarding the club they support and its upcoming opponent's qualities, shortcomings, and prospects. As the prevalence of social media continues to increase, along with the prevalence of platforms like Twitter that enable individuals to provide instantaneous reactions to live sporting events, opportunities exist for future research endeavors to incorporate more extensive fan involvement [8].

The study focused on investigating the relationship between social media activity and National Football League (NFL) games, with a particular emphasis on the timing of the tweets. To accomplish this, a high-precision technique was developed that relied on hashtags in tweets. This technique allowed weekly tweets to be classified based on when they occurred relative to the start of the previous game. Specifically, tweets that occurred at least 12 hours after the start of the

previous game were classified as weekly tweets. Additionally, pre-game tweets were defined as those that occurred between 1 hour to 24 hours before the start of the game. The study analyzed over 177 games in the 2012 season, and the technique correctly predicted the winner 63.8% of the time [9].

The utilization of public sentiment has been explored in numerous attempts to predict football match outcomes. A methodology that incorporates natural language processing techniques, statistical data, and contextual articles from sports journalists was employed to forecast match results. Specifically, the baseline model for the English Premier League was analyzed over the course of three seasons, and it was observed that the model achieved a predictive accuracy of 63.19%. Furthermore, as the season progressed, accuracy improved by 6.9%, indicating that the influence of human variables became more prominent in determining the match outcome [10].

A study was conducted during the 2018 World Cup to investigate the potential signals present in Twitter data and the influence of sentiment magnitude on successful match prediction. The study involved the construction of a database comprising 38,371,358 tweets and 7,876,519 unique users. Nine different machine learning models were trained on the data from the 48 matches in the group phase and tested to predict round 16 and beyond. The models considered specific information about the person, such as the number of followers, location, likes, tweets, and so on, as well as details about the tweet itself, including whether it was a retweet or a reply, retweet and like count, and more. The best-performing model was the Multilayer Perceptron method, which achieved an accuracy of 81.25%

after 30,000 epochs [11].

An odds-based methodology was studied that compiles an odds-maker's match balance sheet in response to wagers. By standardizing a certain data model against tweets for a given club and match, sentiment was derived, which was found to be a reliable source of information.

The equation of the Normalize polarity is:

$$Max(\frac{(Tweets|Model_n, Club_1, Match_m)}{(Tweets|Club_1, Match_m)}, \frac{(Tweets|Model_n, Club_2, Match_m)}{(Tweets|Club_2, Match_m})$$

The tweets were gathered using one team-specific hashtag per club, but only tweets with negative or positive sentiment were used in the models, with the expectation that the team with more positive tweets would win. The pay-out of five out of the eight sentiment models exceeded those of the crowd source odds-only model. Although using only one hashtag per club is a study limitation, the large volume of tweets collected outweighs this restriction. This approach is adopted in the event of non-football occurrences that may influence the data set. For instance, an incident of racist abuse involving supporters of Chelsea FC in Paris could potentially impact the dataset [12].

A comparable methodology of employing hashtags as opposed to keywords to aggregate tweets was similarly observed in another study, whereby a compilation of hashtags closely linked with each of the twenty teams in the Premier League was devised. The development of this list was largely predicated on a multitude of online resources that delineate the official hashtags for the teams, as well as

any nicknames that may be associated with them [13].

Upon analysis, the outcomes of these studies appear favorable, notwithstanding the discovery of a further limitation. Observing Twitter data for prediction, it is noticeable that different implementations are unable to identify sarcasm in text, making it difficult to determine whether the user is being serious or not [14].

2.3 Natural Language Processing

Natural Language Processing (NLP) refers to the computational analysis of textual data. The field of NLP is concerned with acquiring insights into how humans comprehend and employ language, with the ultimate goal of devising effective tools and techniques that enable computer systems to comprehend and manipulate natural languages in order to perform diverse tasks as required [15].

The aim of Natural Language Processing is to accommodate one or more specialities of an algorithm or a system. The NLP metric evaluates an algorithmic system that permits the combination of language comprehension and language production [16].

2.3.1 Pre-Processing Text

The analysis of sentiment necessitates pre-processing components, which serve to organize the text and isolate distinctive features that can subsequently be leveraged by machine learning algorithms and text mining heuristics. In essence, the goal of pre-processing is to segregate a sequence of characters from a text stream into categories, with movements from one state to the next being triggered by the presence of specific characters [17].

However, the intricacies and diversity of natural language pose significant challenges in pre-processing textual data. It can include multiple characters, words, and structures that can be influenced by elements such as misspellings, capitalization, and punctuation. Pre-processing techniques address these challenges by preparing the text for subsequent processing and simplifying its categorization [18].

After a significant quantity of data has been amassed, it undergoes a pre-processing stage. This stage involves eliminating superfluous and uninformative data from the collected text, and preparing it for subsequent classification. Once all data has been pre-processed as desired, it can be passed on for feature extraction, depending on the type of system being used, and eventually annotation. The pre-processing components are described as follows:

Overall, pre-processing text data is a crucial step in many NLP tasks, and involves a range of techniques for cleaning and normalizing text data. These techniques help to improve the performance of downstream NLP tasks, and enable more effective analysis of the text data [19].

2.3.2 Sarcasm in Natural Language Processing

Sarcasm is a complex communication form where the speaker implies their meaning rather than explicitly stating it. Detecting sarcasm is challenging for Natural Language Processing (NLP) systems, despite the use of various techniques such as statistical models, pattern recognition, and machine learning. A study used a Semi-Supervised Sarcasm Identification (SASI) algorithm to classify sarcastic tweets and Amazon reviews [20]. The algorithm was trained using #sarcasm

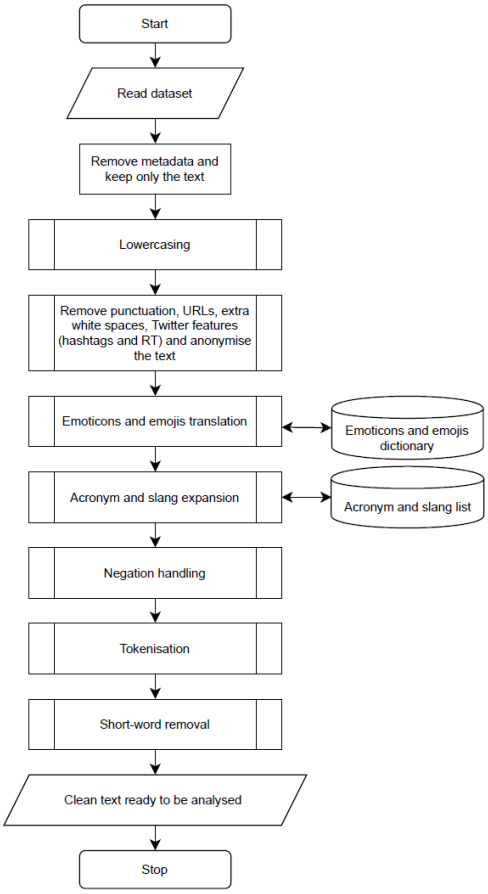

Figure 2.1: *A schematic diagram of the pre-processing steps* [17]

tweets, specifically tagged as sarcastic by users.

However, non-sarcastic tweets with the hashtag and its usage in another context posed a challenge. The study analyzed 5.9 million unique tweets, with only 6.9% of them containing at least one hashtag. Hashtags serve as a digital representation of acerbic non-verbal emotions in human communication, with synonyms for #sarcasm being the most prominent linguistic indicators of sarcasm. Only 86% of tweets explicitly marked with the hashtag were truly sarcastic. The decrease of the number of sarcastic tweets from the final dataset by was also analyzed by merging the tweets based on hashtag markers like #irony, #cynisme, #humor, #joke and punctuation occurring in the same tweet [21].

2.4 Lexicon-Based Approach

This section explores lexicon-based sentiment analysis techniques that use pre-determined dictionaries of sentiment rich words. Although effective in natural language processing, their application in sports, particularly football, has not been thoroughly examined due to varying emotional meanings of words in different contexts and the presence of emotional terms that do not indicate sentiment [22]. This approach usually has two types of techniques: the Dictionary-Based Technique and the Corpus-Based Technique [23].

Dictionary-Based methods involve creating a seed list of words with known polarity, expanding this list by extracting synonyms or antonyms iteratively from online dictionaries like WordNet, and assigning the sentiment strength of words based on their interaction with the seed list while the Corpus-based approach centers on dictionaries that pertain to a particular domain, in which words are associated with semantic and statistical techniques like LSA [24].

When utilizing this approach, there exist several libraries that are available for use. One commonly used library in this regard is referred to as VADER. VADER is open source Python code and uses three pre-defined dictionaries, including LIWC, social media slang and abbreviations, facial expressions, and emoticons. LIWC is text analysis software designed for studying the various emotional, cognitive, structural, and process components present in text samples. LIWC uses a proprietary dictionary of almost 4,500 words organized into one (or more) of 76 categories, including 905 words in two categories especially related to sentiment analysis [25]. In a comparative lexicon-based study, VADER achieved the

best performance in terms of accuracy, while the accuracy of the SentiStrength, AFINN-111, and Liu-Hu lexicons was relatively the same. [26].

TextBlob is also a widely adopted sentiment analysis tool that operates as a Python library designed to handle textual data. The model builds upon the Natural Language Toolkit (NLTK) library, which presents a consistent API for executing fundamental NLP tasks. The TextBlob sentiment analysis approach is based on a combination of a lexicon and rule-based method, similar to the VADER analyzer. Besides, TextBlob offers several functions that include noun phrase extraction, part-of-speech tagging, language detection and translation, n-grams, and spelling correction. In the context of sentiment analysis, the tool computes the polarity and subjectivity of the text under scrutiny [27].

2.5 Machine Learning Approaches

The aim of this section is to analyse various sentiment analysis systems which have been built based on machine learning approaches. Machine Learning (ML) is an implementation of artificial intelligence that employs algorithms to endow system with the capacity to learn automatically from experience, without requiring explicit programming [8]. Machine learning algorithms function by constructing a mathematical model that identifies features and patterns within the data, and draws insights from a designated set of pre-annoted documents, commonly referred to as ”training data”. The model then generates an automated text classifier, which can be employed to make predictions or decisions [28].

One classification of such algorithms is that of supervised learning. In this

approach, the learning algorithms begins by evaluating a set of data that has been previously annoted with corresponding labels, and utilizes this information to construct a theoretical function that can be employed to predict output values. Through this iterative process of analyzing and learning from the labeled data, the system is able to produce outputs for novel inputs with sufficient accuracy and reliability [8].

2.5.1 Traditional ML Approaches

The process of classification in Machine Learning involves the use of a training set and a test set, which contain input feature vectors and their corresponding class labels. The goal is to develop a classification model that can predict the class labels of unseen feature vectors. Naive Bayes (NB) and Support Vector Machines (SVM) are some of the machine learning techniques commonly used for sentiment classification. Various features, including Term Presence, Term Frequency, negation, n-grams, and Part-of-Speech, can be utilized to determine the semantic orientation of words, phrases, sentences, and documents [29].

The SVM algorithm works by determining an optimal hyperplane in a high-dimensional feature space that separates classes effectively. A study was performed to assess the efficacy of SVM by utilizing the Sentiment140 Twitter data set from Stanford University [1]. The Sentiment140 corpus encompasses 1.6 million tweets, evenly divided between two classes, Positive and Negative. The outcome of the sentiment analysis indicated that the Support Vector Machines (SVM) exhibited the optimal performance with an accuracy rate of 83% on the Senti-

[1] `http://help.sentiment140.com/for-students/`

ment140 data set [30].

However, it is important to note that SVM has certain limitations and challenges associated with using this algorithm. One potential issue is the selection of appropriate features, as SVM's performance can be heavily influenced by the quality and relevance of the features used. To address these challenges, researchers have proposed various approaches to improve the performance of SVM in sentiment analysis. For instance, some studies have suggested using more sophisticated feature representations, such as word embeddings, to capture the semantic relationships between words [31].

Following to this, another significant challenges faced by SVMs is their computational complexity, which grows linearly with the number of training examples, thereby resulting in substantial computational costs during the training phase. As such, SVMs may encounter difficulties when applied to high-dimensional problems that necessitate a vast amount of training data [32].

On the other hand, the Naive Bayes algorithm is a probabilistic classifier based on the Bayes' theorem, which is commonly employed in NLP tasks like sentiment analysis. The primary objective of the Naïve Bayes classifier is to identify the most appropriate category for a given text, as well as to classify text documents based on the statistical distribution of term occurrences. These terms, referred to as features, are pivotal in enabling Naïve Bayes to execute topic classification, with frequencies surpassing a particular threshold serving as an example of the features that may be utilized [33].

The Bayes' theorem is used to determine the probability of a hypothesis when

prior knowledge is available, depending on conditional probabilities. The formula is given below [34]:

$$P(CX) = F(XC)P(C)/P(X)$$

where P(CX) is posterior probability i.e. the probability of a hypothesis A given the event B occurs. P(XC) is likelihood probability i.e. the probability of the evidence given that hypothesis A is true. P(C) is prior probability i.e. the probability of the hypothesis before observing the evidence P(X) is marginal probability i.e. the probability of the evidence

The MultinomialNB package in Python was used to estimate the conditional probability of features belonging to a particular class based on the assumption that the occurrence of each feature follows a multinomial distribution. To avoid overfitting, five-fold cross-validation was used to train the Naive Bayes classifier. This method divides the data into five equally sized subsets, utilizing four of them to train the classifier and the remaining subset to test it. This process is repeated five times, with each subset serving as the test set once. Subsequently, the trained classifier was applied to the test dataset to categorize new tweets as either positive or negative. The results of the experiment indicated the high performance of the Naive Bayes classifier, with it achieving high accuracy, recall, precision, and F1-measure for several feature combinations [35].

Experimental Results and ML Method		
Correctly classified instances	Sentiment140 Data set [2]	MovieReview Data set [3]
Stopword filtered word features	0.79	0.81
Unigram with Bi-gram features	0.86	0.89
Bigram with Stopwords	0.85	0.85

The correlation between football events and sentiment analysis was studied using Naive Bayes and Bayesian Logistic Regression (BLR). The tweet polarity classifier was trained based on N-grams features (N=1 to 2) using WEKA as a machine learning framework. A cross validation was used as repeated holdout method used on the data set by splitting it into 10 sections. This method selects 90% for the training set, and 10% for the testing set, repeating it on 10 different sections of the data set. The goal of using this method is to test the model in the training phase, hence, the following steps were taken for machine learning classification: 1. Pre-Processing the data; 2. Feature generation; 3. Feature selection; 4. Training the model and validation.

Furthermore, this study assesses the efficacy of the BLR classifier in improving the performance of a simple two-class sentiment analyzer (positive/negative). The BLR model's performance was compared to that of Naive Bayes, and both models demonstrated optimal performance when utilizing Uni-grams and Bi-grams feature combinations [36].

Experimental Results and ML Method		
Correctly classified instances	BLR	NB
Uni-grams	71.35	66.21
Bi-grams	67.44	63.62
Uni-grams and Bi-grams	74.84	66.24

In conclusion, both SVM and Naive Bayes algorithms have their unique advantages and limitations in sentiment analysis. SVM is effective in high-dimensional feature spaces, particularly when utilizing unigram-based features and feature presence information. On the other hand, Naive Bayes algorithm is a probabilistic classifier based on the Bayes' theorem, which works well with several feature combinations.

2.5.2 Deep Learning Solutions

Deep learning represents a class of machine learning algorithms that leverages complex structures or multiple nonlinear transforms, embodied within multiple processing layers, in order to extract high-level abstractions from data. This algorithmic paradigm, rooted in machine learning, is predicated on characterizing the learning data [37].

It is also a powerful technique that facilitates the representation of words from textual data and generates word embeddings, which can be utilized by various machine learning methods. Unlike the traditional approach of handcrafting features, deep learning enables the automated learning of features, thus avoiding the time-consuming and often incomplete process of manual feature engineer-

ing. Moreover, deep learning produces high-quality features and multiple levels of representation, which significantly enhance the efficacy of machine learning models [38].

Moreover, artificial neural network (ANNs) are computational systems that are inspired by the functioning process of the human brain. They are capable of modifying their internal structure in relation to a function objective and are particularly well-suited for solving non-linear problems by reconstructing the ambiguous rules that govern the optimal solution for these problems. The basic components of ANNs are nodes, also referred to as processing elements (PE), and connections. Each node receives input from other nodes or from the environment and produces an output, which communicates with other nodes or the environment. Each node is associated with a function f that transforms its input into output. The connections between nodes are characterized by the strength of their influence, which can either be excitatory or inhibitory, denoted by positive and negative values, respectively [39].

Deep neural networks utilize advanced mathematical modeling techniques to process data in various ways. A neural network is a flexible model that maps inputs to outputs, and is composed of multiple layers, including an input layer that contains input data, hidden layers that contain processing nodes called neurons, and an output layer that contains one or more neurons, whose outputs constitute the final output of the network [41]. Overall, DNNs are able to capture the complex patterns and nuances in language that are essential for accurately identifying sentiment, resulting in high classification accuracy.

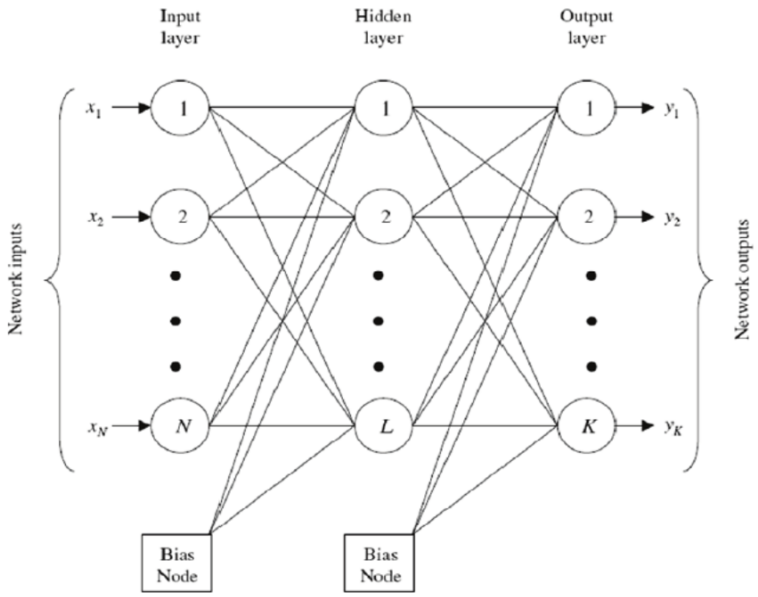

Figure 2.2: *A schematic diagram of artificial neural network and architecture of the feed forward network with one hidden layer*

[40]

Long Short-Term Memory (LSTM) with Recurrent Neural Networks and Convolutional Neural Networks (CNNs) are two types of deep learning methods that have been widely used for sentiment analysis. LSTM is a specific type of RNN that incorporates a memory cell with input, forget, and output gates to regulate the flow of signals and achieve successful RNN training. LSTM's ability to learn long-term relationships between words and phrases makes it suitable for sentiment analysis tasks. The forget (f) gate in LSTMs allows the LSTM to reset its own state. This enables the LSTM to look back many timesteps in order to create more accurate predictions [42]. The gates regulate the flow of signals into and out of the cell to adjust long-term dependencies effectively and achieve successful RNN training. The standard equations for LSTM memory blocks are given as follows:

$$i_t = (w_i[h_{t-1}, x_t] + b_i)$$

$$f_t = (w_f[h_{t-1}, x_t] + b_f$$

$$o_t = (w_o[h_{t-1}, x_t] + b_o)$$

where it - represents input gate.

ft - represents forget gate

ot - represents output gate

- represents sigmoid function

wx - weight for the respective gate (x) neurons

ht − 1 *- output of the previous lstm block(at timestamp t-1*

xt - input at current timestamp

bx - biases for the respective gates (x)

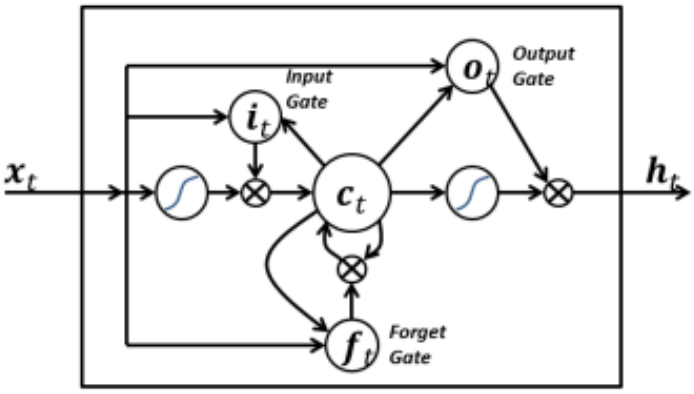

Figure 2.3: *An LSTM unit with input i, output o, and forget f gates*

The study presented Long Short-Term Memory (LSTM) as a suitable sentiment analyzer for cryptocurrency social media posts. LSTM's ability to learn long-term relationships between words and phrases was emphasized by the au-

thors, accomplished through forget and remember gates. The analyzer tokenizes social media posts according to a crypto word vocabulary, converts them into crypto word embeddings, and trains the LSTM on the sequence of embedding feature vectors. The output is transformed using a fully connected layer and a sigmoid function. The authors evaluated their LSTM sentiment predictor using precision and recall metrics on the top 100 crypto investors' accounts' most recent 7 days' Sina-Weibo posts as training data and the following day's posts as testing data. Compared to the time series auto regression (AR) approach, their method outperformed by 18.5% in precision and 15.4% in recall. The study showcases LSTM's effectiveness in sentiment analysis and highlights its potential for cryptocurrency trading and other applications [43].

The study by Murphy et al. [44] employed LSTM to assess the effectiveness of deep learning techniques in sentiment analysis using an IMDB and Amazon review data set. The set comprised 50,000 reviews, equally divided between positive and negative polarizations. Out of these, 50,000 were used for training and 23,500 for model evaluation. The authors reported that a dropout rate of 0.2, Adam optimizer, and sparse categorical cross-entropy loss function achieved the best performance during training, with a batch size of 500. The study found that LSTM-based deep learning techniques outperformed other methods, with an accuracy of 85%. The study highlights the importance of selecting appropriate hyper-parameters for deep learning models in sentiment analysis tasks.

On the other hand, Convolutional Neural Networks (CNNs) are also widely utilized in various domains, such as computer vision, natural language processing,

and recommender systems [45]. The architecture of the network comprises of convolutional and subsampled layers, which extract features from the inputs and reduce their resolution to enhance their resistance against noise and distortion. The final classification is performed by the fully-connected layers.

CNN was outlined in a paper made by Liao et al. [46] in which a machine learning-based approach was done to classify the sentiment of Twitter data using a convolutional neural network (CNN). The approach involves training the CNN model with the MR and STS Gold datasets, consisting of movie reviews and real tweets. The CNN model involves several steps, including converting the sentence into a matrix of word vector representations, performing convolution on the matrix using linear filters, and applying a pooling function to obtain a fixed-length vector. The datasets were divided into two groups, namely the train set (around 90% of the data) and the development set (around 10%) used to verify the accuracy of the CNN model checkpoint. To test the CNN model, it was trained multiple times, and the filter windows (h) 4, 5, 6 with 100 feature maps each were selected based on the highest average development accuracy. The dropout rate (p) was set at 0.5, the l2 regularization lambda was set at 0.001, and the batch size was set at 64. The MR training results had a development accuracy of 74.5%, and the STS Gold Dataset training results had a development accuracy of 68%.

An additional investigation conducted by researchers involved the utilization of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) deep learning neural networks for sentiment analysis on the Lithuanian Internet

comment data set. The models were subjected to testing using both word2vec and FastText word embedding. The outcomes indicated that the CNN model efficiently provided accurate results in 70.6% of the instances. From these results, the authors inferred that deep learning neural networks offer favorable performance when employed to smaller data sets [47].

2.6 Conclusion

The chapter reviewed the use of sentiment analysis in the field of sports and the effectiveness of using machine learning and deep learning to generate sentiment scores. Based on the research reviewed, it was concluded that VADER, Naive Bayes, and Long Short-Term Memory (LSTM) are the most accurate approaches for sentiment score prediction. The following chapter covers the research methodology and implementation of the model.

Chapter 3: Research Methodology

3.1 Introduction

(1,500 – 3,000 words)

The purpose of this chapter is to investigate the relationship between sentiment and changes in the league standings by describing the methodology used to conduct a comparative sentiment analysis on a football related data set.

In terms of methodology, this study uses a mixed-methods approach to investigate the relationship between sentiment and changes in the league table. Quantitative data will be gathered through sentiment scores calculated by the three different sentiment analysis models, as well as changes in the league table. Qualitative data will be gathered through the analysis of the tweets themselves. This mixed-methods approach allows for a more comprehensive understanding of the relationship between tweet sentiment and changes in the league table and match results.

3.2 Research Objectives

This study aims to conduct sentiment analysis on tweets related to a football team, with four research objectives to achieve this goal. The objectives of this study are as follows:

1. To evaluate the performance of different sentiment analysis models on Twitter football data and identify the best model based on accuracy, precision,

recall, F1-score, and other evaluation metrics.

2. To analyze the sentiment of tweets related to football matches and determine if there is a relationship between the sentiment of tweets and changes in the league table.

3. To analyze the sentiment of tweets related to football matches and determine if there is a relationship between the sentiment of tweets and match results.

4. To demonstrate whether Twitter can still provide adequate amounts of tweets for teams who have a smaller fan base that can be enough to conduct a sentiment analysis study.

By accomplishing these objectives, this study aims to advance sentiment analysis in the field of sports, providing a comparative analysis of different approaches and exploring their potential applications in predicting football outcomes.

3.3 Significance of the Study

The significance of this study lies in its contribution to the field of sentiment analysis and its potential impact on football teams, fans, and analysts. By comparing three different approaches to sentiment analysis on a data set of tweets related to a football team, this study will identify the most accurate approach for predicting the sentiment of the tweets. Furthermore, its findings will have broader implications for the field of sentiment analysis. By identifying the most accurate approach to sentiment analysis, this study can inform future research in this area

and help to improve the accuracy of sentiment analysis models in general.

In addition to the mentioned contributions, this study has the potential to offer valuable insights to bettors and betting companies. By identifying the best model for Twitter football sentiment analysis and analyzing the relationship between sentiment and changes in the league table and match results, this study can enhance our understanding of how social media data can be used to predict performance and inform decision-making. Ultimately, this study's findings hold significant implications for both the field of sentiment analysis and the betting industry, underscoring its relevance and importance.

3.4 Prototype Pipeline

3.5 Frameworks and Libraries

In the development of such systems, a variety of resources were utilized, namely programming languages, a virtual environment, and diverse packages that were directly downloaded into said environment.

3.5.1 Programming Language

In the course of this research, Python [1] was utilized for various reasons. As highlighted by Garg and Bassi [48] in his work on sentiment analysis, Python is a versatile programming language that is widely acknowledged for its simplicity, power, and dynamic nature. It is widely acknowledged for its proficiency in processing natural language data, especially spoken English, using the Natural

[1] `https://www.python.org/`

Language Toolkit (NLTK). Python's vast collection of open-source libraries and large user community also contribute to its appeal and functionality for various applications.

3.5.2 Package Installation and Management with Anaconda VE

Although Python packages can be installed via the terminal for use within a project it is generally more advisable to install all necessary packages within a virtual environment to facilitate ease of management. To this end, the virtual environment was created and activated using the Anaconda platform, which not only enabled seamless installation of required packages but also facilitated their subsequent updates and removals. Specifically, for the purposes of this research, the Anaconda version 22.9.0 was employed.

3.5.3 Main Packages

The main packages used in this study are as follows:

3.6 Training Dataset

3.6.1 Overview

The data set utilized for training the football sentiment models was sourced from a publicly available Github repository [49]. This repository comprises three discrete files that pertain to football-related tweets, player tweets and World Cup tweets, along with the associated sentiments expressed. Since our research, solely focuses on the sentiment of football team, the data sets of both teams and play-

ers were merged.

3.6.2 Data Manipulation

The data set under examination featured various fields, including Tweet Creation Date, Tweet ID, Tweet Language and Sentiment Score. Both the date of tweet creation and sentiment rating attributes were subsequently omitted, as they were deemed inapplicable for the purposes of training the model. Upon the compilation of the data sets, a total of 762,643 redundant tweets were discerned and subsequently deleted. The tweet identifier column was ultimately discarded, as it no longer held any relevance to the analytical investigation being conducted.

In order to ensure the effectiveness of our sentiment analysis model, a language verification was executed to corroborate that all the tweets within the dataset were composed in the English Language. Upon inspecting the sentiment analysis categories of the dataset, the mixed category was removed due to the relatively small number of tweets assigned to it. This action was deemed necessary to maintain the integrity of our analysis and avoid potential inaccuracies in the model training and evaluation process.

Dataset Sizes		
Pre-processing Tasks	**File Size (MB)**	**Processing Time (s)**
Before pre-processing	734.2	NA
After removing URLs	699.5	42.3
Removing of mentions ”@”	585.6	57.7
Filtering # from tweets	544.0	50.2
Removing punctuation’s	523.3	62.7

Table 3.1: *Data Refinement Process*

The pre-processing pipeline included removing redundant words, abbreviations, and unwanted patterns such as URLs, usernames, punctuation's, hashtags and more. To facilitate the removal of stop words, the stop words parameter within the CountVectorizer object was set to 'English'.

Before pre-processing Tasks	**After pre-processing Tasks**
I have to agree with #Lovren he has become one of the best defenders in the world but can he keep it up I hope he can #LFC DejanLovren	i have to agree with he has become one of the best defenders in the world but can he keep it up i hope he can
I hate supporting NUFC while Mike Ashley owns the club. It has ruined football and my love of the sport year after year. #NUFC	i hate supporting while mike ashley owns the club it has ruined football and my love of the sport year after year
Why haven't we signed anyone in almost 24 hours. This is a shambles. #cpfc	why havent we signed anyone in almost 24 hours this is a shambles

Table 3.2: *Table to show the tweets before and after processing*

This table provides a clear comparison of tweets before and after the pre-processing tasks. It highlights the removal of hashtags, mentions, and other unnecessary elements, showcasing the cleaned and simplified tweet text that will be used for training the sentiment analysis model.

3.6.3 Data Exploration

Data exploration is a crucial stage in our research, enabling us to comprehend the attributes, recurrent patterns and potential issues of the training dataset. Since the tweets were not equally distributed for each category, the categories was down-sampled to a uniform count of 354,501 tweets per classification in order to ensure precise model training.

A word-frequency analysis was conducted to identify salient terms, subject matter, and unusual patterns in the data. An unusual presence of words like ”amp” was found and removed accordingly. This may be occurring because of technical errors or typographical mistakes, therefore, it is essential to pre-process the data and remove such extraneous words from the data set.

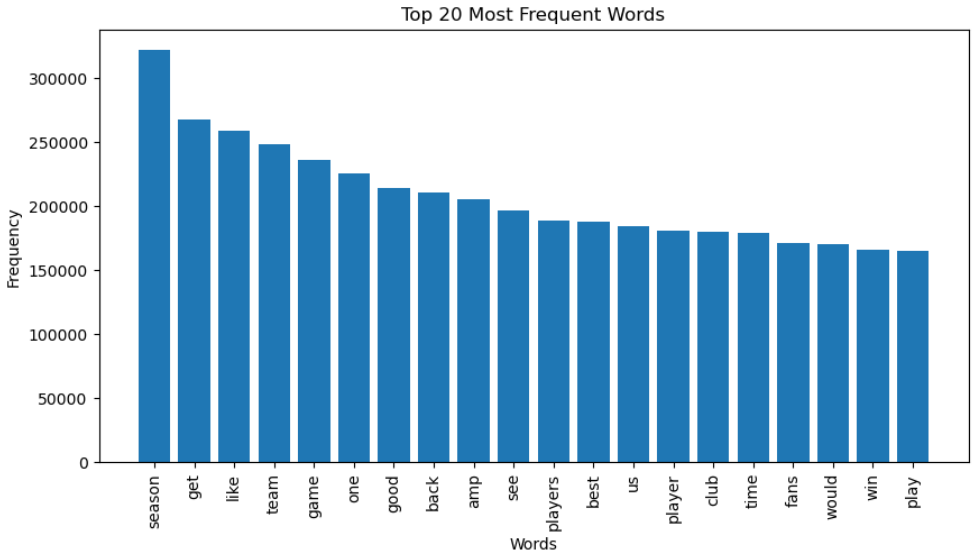

Figure 3.1: *A visualization representing the 20 most frequently occurring words within the trained dataset.*

3.7 Model Selection

3.7.1 VADER Implementation

In this study, we employed the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis method as an alternative approach for classifying the sentiments of football-related tweets. VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to the sentiment expressed in social media text. It takes into account the intensity of emotions and provides a compound score that represents the overall sentiment of a given text.

In order to perform sentiment analysis utilizing VADER, the SentimentIntensityAnalyzer [2] class from the NLTK library was employed. A custom function, named 'predict-sentiment,' was defined to accept textual input and return the corresponding sentiment polarity, which is determined by the compound score generated by VADER. Adhering to the approach outlined by [50], a threshold of ±0.05 was established for the compound score. Consequently, a score greater than or equal to 0.05 was classified as positive, while a score less than or equal to -0.05 was deemed negative; scores falling in between these values were considered neutral.

The dataset employed for generating predictions underwent pre-processing, as described in the preceding chapter. The 'text-without-punctuation' column was converted into string values, and the custom 'predict-sentiment' function was applied to each row to yield VADER's sentiment predictions.

3.7.2 Naive Bayes Model

In this present research, the chosen machine learning algorithm for sentiment analysis is the Multinomial Naïve Bayes classifier. The study aims to classify the sentiment of football-related tweets using this probabilistic approach. The classifier employed a pipeline composed of three primary components: CountVectorizer, TfidfTransformer, and MultinomialNB.

[2] `https://www.nltk.org/api/nltk.sentiment.html`

Bag of Words (BoW) Approach

The first step in the pipeline is the creation of Bag of Words (BoW) representation using CountVectorizer [3]. This process tokenizes the text data and constructs a vocabulary containing all unique words present in the corpus. The text data is then transformed into a matrix where each row represents a document, and each column corresponds to a word in the vocabulary. The value in each cell of the matrix indicates the number of times the word appears in the corresponding document.

The BoW representation generated doesn't take into account the order of words, and it merely focuses on their presence and frequency. This method captures the word distribution in the documents but might give equal importance to common and rare words.

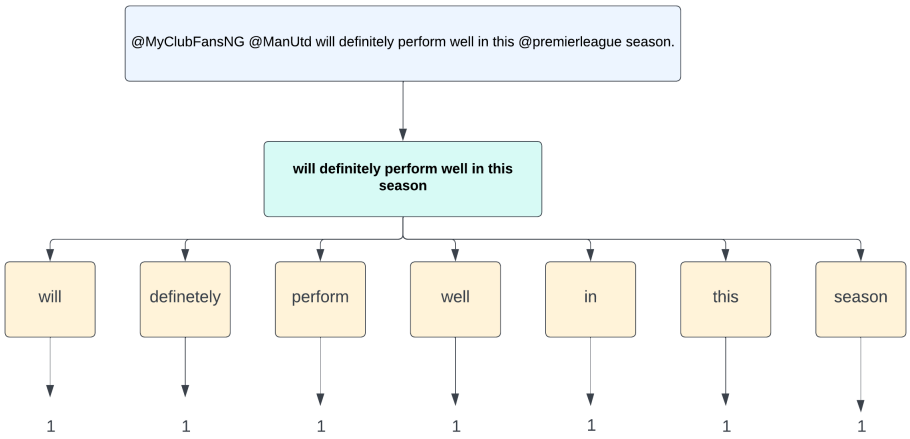

Figure 3.2: *Example of a Training Tweet in Bag of Words Representation*

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html`

Term Frequency-Inverse Document Frequency (TF-IDF)

The next step in the pipeline is to convert the BoW representation into a TF-IDF representation using TfidfTransformer [4]. The TfidfTransformer in the pipeline is responsible for transforming the BoW matrix, into a TF-IDF respresentation.

The rationale behind using TF-IDF with the BoW matrix is to emphasize the importance of words not just based on their frequency within a single document, but also in relation to their occurrence across the entire document set.

Metric	**Description**
Term Frequency (TF)	The raw frequency of a word in a document is normalized by dividing it by the total number of words in that document. This adjustment accounts for varying document lengths and provides a proportional measure of each word's contribution to the document.
Inverse Document Frequency (IDF)	This metric aims to quantify the significance of a word in the entire corpus. By calculating the logarithm of the total number of documents divided by the number of documents containing the word, common words that appear in many documents are assigned lower weights, while unique and context-specific words are assigned higher weights.

Table 3.3: *Description of Term Frequency and Inverse Document Frequency*

The final TF-IDF representation is computed by multiplying the term frequency (TF) with the inverse document frequency (IDF) for each word in the BoW matrix.

The approach of combining Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) in a sentiment analysis pipeline is particularly useful for analyzing football-related tweets, as it provides a robust and effec-

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html`

tive method to represent the domain-specific text data for classification tasks. By initially using Bag of Words, the model captures the frequency distribution of words, which is essential in identifying football-related terms and expressions. Building upon this foundation, the pipeline applies TfidfTransformer to transform the raw term frequency matrix into a TF-IDF representation, which assigns higher weights to words that are more specific and relevant to football discussions, while downplaying the importance of common words that occur frequently across documents. This combination ensures that the model captures not only the word frequency information but also the significance of each football-related word within the corpus. As a result, the Naive Bayes classifier benefits from a richer feature set that better represents the underlying football-related text data, potentially leading to improved classification performance in sentiment analysis tasks.

Multinomial Naive Bayes Classifier

The Multinomial Naive Bayes classifier [5] is used as the final step in the text classification pipeline to classify tweets as positive, negative or neutral. he Multinomial Naive Bayes algorithm is a variant of the Naive Bayes classifier, specifically designed to handle discrete data, such as word frequencies in text documents. It is based on the Bayes theorem, which calculates the probability of a given class (in this case, sentiment) given the observed features (word frequencies). It makes the assumption that the features (words) are conditionally independent given the class, which simplifies the computation and allows the model

[5] `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html`

to scale well with large datasets.

The employment of the Multinomial Naive Bayes classifier, as opposed to the Bernoulli Naive Bayes classifier, is justified by the nature of the text data. This classifier is capable of capitalizing on the wealth of information furnished by word frequencies, potentially culminating in superior classification performance in comparison to a Bernoulli Naive Bayes classifier, which solely takes into account the presence or absence of words. Empirical evidence from numerous studies has demonstrated that the Multinomial Naive Bayes classifier outperforms its Bernoulli counterpart. Furthermore, prior research delineated within the literature review corroborates the utilization of the Multinomial Naive Bayes classifier in the given context [35, 51].

In the training phase, the Multinomial Naive Bayes classifier comprehends the patterns and interconnections amongst the words in tweets and their corresponding sentiment labels (positive, negative, or neutral). Subsequently, when the classifier is provided with a new, untagged tweet, it ascertains the probabilities of the tweet being assigned to each sentiment label based on the word frequencies in the tweet. Consequently, the predicted sentiment of the tweet is assigned based on the sentiment label with the highest probability. The specific decision rules and thresholds for designating a sentiment label may differ depending on the specific execution of the classifier and the task at hand.

Training the Model and Hyper-parameter Tuning

The dataset was split into a training and test dataset, with a 70% and 30% allocation, respectively. This allowed the model to be trained on a large portion

of the data and subsequently evaluated on a separate, unseen dataset to assess its performance and generalizability.

In order to optimize the model's performance and identify the best hyper-parameters, a search cross-validation using the GridSearchCV [6] function from the Scikit-learn library was used. RandomizedSearchCV is an alternative to GridSearchCV, another popular method for hyper-parameter tuning.

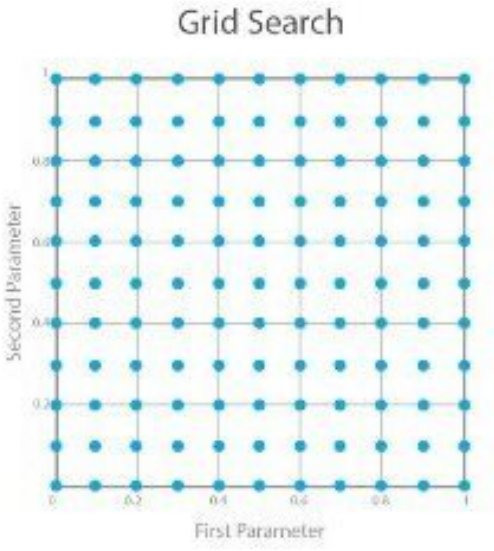

Figure 3.3: *An illustration of a grid search space. A range of the possible parameters and the algorithm makes a complete search over them.*

In contrast to GridSearchCV, which searches all possible combinations of hyper-parameters, RandomizedSearchCV randomly selects a pre-defined number of combinations from the specified hyper-parameter space [52]. This approach is more computationally efficient, especially when dealing with a large number of hyper-parameters or when the search space is vast. Nevertheless, it is imperative for the model to be trained utilizing the most optimal hyper-parameters available. Consequently, GridSearchCV emerges as the superior solution, given that Ran-

[6] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

domizedSearchCV encompasses a limited set of hyper-parameters in its pursuit of attaining the highest accuracy.

A set of hyper-parameters were tuned using GridSearchCV processes. The set of hyper-parameters included the following:

Hyperparameter	**Description**
ngram-range	Determines the size of word combinations (uni-grams, bi-grams, tri-grams) to be considered.
max-features	Limits the number of features to be extracted from the text. The search space included 1000, 5000, 10000, and None, where None indicates no limit on the number of features.
use-idf	Controls whether to apply inverse document frequency (IDF) weighting in the TfidfTransformer. Both True and False values were explored.
norm	Specifies the normalization method for the TF-IDF scores. L1 normalization (Manhattan or Taxicab) calculates the sum of the absolute values of vector components. L2 normalization (Euclidean) calculates the square root of the sum of the squared vector components.
alpha	Represents the smoothing parameter for the Multinomial Naive Bayes classifier. Different values (0.01, 0.1, 1.0, 10.0, 100.0) reflect different assumptions about the training data's representativeness and the influence of unseen words on the classification.

Table 3.4: *Hyperparameters considered for the Naive Bayes model.*

In an effort to improve upon the methodology employed in a previous Naive Bayes study by Iqbal et al. [35], the GridSearchCV instance employed a 10-fold cross-validation approach to ensure a more thorough evaluation of the model's performance across different hyper-parameter configurations. The evaluation metrics included accuracy and a classification report, which provided detailed information on precision, recall, and F1-score for each class.

3.7.3 LSTM Model

This part outlines the methodology adopted for implementing the LSTM model for sentiment analysis. The model development process involved data preparation, model architecture design, parameter tuning and model evaluation.

Data Preparation

The first step in the model development process was data preparation and pre-processing, which involved tokenization and padding of input text. Tokenization is a fundamental step in natural language processing, where raw text is broken down into smaller units called tokens. In the context of the our implementation, tokens refer to individual words in the input text. The purpose of tokenization is to convert unstructured text data into a structured format that can be easily processed and analyzed by machine learning models.

In this implementation, the Keras Tokenizer class is used for tokenization. Since the focus of the model is to build a model on a specific domain, the Tokenizer class is initialized with a parameter 'num-words', which is set to a maximum of 20,000. This means that the tokenizer will only consider the top 20,000 most frequent words in the dataset when creating its internal vocabulary. This is done to limit the size of the vocabulary and reduce the computational requirements for training the model. Less frequent words are often considered as noise and may not contribute significantly to the model's performance. Furthermore, to ensure uniform length of input sequences, padding was applied using the pad-sequence function from Keras, setting the maximum length to 100.

Model Architecture

The LSTM model was designed using the Keras Sequential API, and its architecture consists of three main layers. The first layer, called the Embedding layer, is responsible for converting the words in the input text into fixed-size numerical vectors. These vectors capture the semantic relationships between words and help the model to understand better the text.

The second layer is the Bidirectional LSTM layer, which is designed to capture both past and future context from the input sequences. This is achieved by processing the text in both forward and backward directions, enabling the model to better understand the overall meaning of the text. The number of LSTM units and dropout rates for both input and recurrent connections in this layer are selected to balance model complexity and the risk of over fitting, ensuring a good performance on unseen data.

Finally, the Dense output layer is used to produce the final sentiment predictions. A softmax activation function is applied to this layer, which ensures that the model's predictions form a valid probability distribution over the target classes, such as positive, negative, and neutral sentiments. The number of output units in the Dense layer is set equal to the number of target classes, allowing the model to produce predictions for each sentiment category. This layered architecture enables the LSTM model to effectively analyze text data and predict the sentiment expressed within it.

Parameter Tuning and Model Training

In order to optimize the model performance, various parameters were tested during the model development process. The optimizer, number of LSTM units, and dropout rates were among the parameters considered for tuning. The create-lstm-model function was designed to accept these parameters as inputs, allowing for easy adjustments during experimentation.

The model was trained on the training set using a batch size of 500 and a validation split of 0.1 for three epochs. During training, the model's performance was continually assessed on the validation set to monitor its progress and make any necessary adjustments. After training, the model's performance was evaluated on the test set, with the loss and accuracy reported.

3.8 Data Collection

This chapter aims to outline the methodology of acquiring tweets by employing prominent team-specific hashtags, with a focus on three discrete intervals within the football season. This approach will allow to facilitates the understanding of sentiment and discourse evolution throughout the progression of the season.

In order to retrieve the data that will be utilized for the correlation aspect of this study, a set of hashtags representing each team in the Premier League throughout the 2021/20222 season were retrieved. The adoption of these hashtags aimed to minimize the inclusion of irrelevant tweets in our analysis, as suggested by Schumaker et al. [12] and Kampakis and Adamides [13]. Given that our study is divided into three distinct time periods—beginning, middle, and end of

the season—the Twitter API proved unsuitable due to its 7-day retrieval limit. Consequently, we employed ”snscrape” [7] as an alternative to gather a more comprehensive range of tweets throughout the season.

The hashtags utilized in this research align with those used in studies by Kampakis and Adamides [13] and D. Pacheco et al. [53].

Table 3.5: *Premier League Teams and Hashtags*

Team	Hashtag
Manchester City	#mcfc
Liverpool	#lfc
Chelsea	#cfc
Tottenham	#thfc
Arsenal	#afc
Manchester United	#mufc
West Ham	#whufc
Leicester City	#lcfc
Brighton	#bhafc
Wolves	#wolves
Newcastle	#nufc
Crystal Palace	#cpfc
Brentford	#brentfordfc
Aston Villa	#avfc
Southampton	#southampton
Everton	#efc
Leeds United	#lufc
Burnley	#burnley
Watford	#watfordfc
Norwich City	#ncfc

As a precaution against sarcasm, a manual check of the tweets was conducted to identify any sarcasm-related hashtags, and such tweets were removed accordingly, in accordance with the guidelines outlined in C. Liebrecht et al. [21].

The table below delineates the information about the data collection process for each time period during the 2021/2022 Premier League season.

[7] `https://github.com/JustAnotherArchivist/snscrape`

Table 3.6: *Data collection process for each time period during the 2021/2022 Premier League season*

Time period	Number of games	Standings differences observed
Beginning of season	240	Yes
Middle of season	240	Yes
End of season	240	Yes

3.8.1 Sentiment Analysis

Once the data was collected, the sentiment analysis was performed on the tweets to predict the sentiment expressed towards each team during each time period. The sentiment analysis was conducted using the most accurate model from the training dataset, and the predicted sentiments were saved as a new column in a new CSV file. In order to maintain consistency and comparability between the training dataset and the collected data, the same pre-processing methods were applied. This new file contained the original tweet text along with the predicted sentiment for each tweet. This process enabled us to efficiently analyze and visualize the sentiment trends across the three time periods and correlate it with the team's performance.

3.8.2 Temporal Analysis

In our temporal analysis, three distinct periods within the 2021/2022 Premier League season were focused - the beginning, middle and end of the season. These periods were carefully chosen to capture the dynamic nature of sentiment and its potential impact on the relationships difference as the season progresses. The beginning of the season is characterized by high fan expectations and opti-

mism, whereas the middle phase may see fluctuations in team performance and shifts in sentiment. The end of the season is a critical period marked by the culmination of performance outcomes, with teams striving to secure their positions in the league or avoid relegation. The analysis of these three periods provides a holistic comprehension of the interrelation between sentiment manifested in tweets and match results, taking into account the progressive context and emotions affiliated with distinct stages of the season.

Table 3.7: *Data collection dates for each time period during the 2021/2022 Premier League season*

Time period	**Dates**
Beginning of season	13th August 2021 - 13th September 2021
Middle of season	1st December 2021 - 1st January 2022
End of season	1st April 2022 - 1st May 2022

3.9 Association Between Variables

Our analysis will center on the utilization of the Chi-Squared Test and Cramér's V as the principal tools for evaluating the association between tweet sentiment, characterized as positive or negative tweets, and football outcomes. The chapter will furnish a comprehensive review of these two statistical methods, expounding on their importance in studying categorical data and the reasoning behind selecting them over correlation analysis.

3.9.1 Chi-Squared and Hypothesis

The Chi-Squared Test is a non-parametric statistical method that determines if there is a significant relationship between two categorical variables. It compares observed frequencies (O) in a contingency table with the frequencies that would be expected (E) under the assumption of no association between the variables. A contingency table displays the frequency distribution of categorical variables in a matrix format, showing the observed frequencies of different combinations of categories of two or more variables.

The Chi-Squared Test employs a mathematical formula to calculate the deviation between the observed and expected data, resulting in a Chi-Squared test statistic that quantifies the discrepancy between these frequencies, as follows:

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{3.1}$$

Where:

- $O_i j$ represents the observed frequency in each cell of the contingency table,
- $E_i j$ represents the expected frequency under the assumption of independence, and the summation ($\sum$) is over all cells in the table.

Another essential aspect of the Chi-Squared Test is the Degrees of Freedom (DF), which plays a critical role in determining the distribution of the test statistic and the significance of the test. In the context of the Chi-Squared Test, the degrees of freedom are calculated using the formula:

$$df = (r-1)(c-1) \tag{3.2}$$

Where:

- r: is the number of rows in the contingency table.
- c: is the columns in the contingency table.

In essence, the degrees of freedom indicate the number of values that have the ability to change independently, without impacting the other values in the table. The degrees of freedom are used to determine the critical value for the Chi-Squared Test, which is then compared to the calculated test statistic. This comparison provides a critical foundation for interpreting the results of the Chi-Squared Test and evaluating the independence of categorical variables and conclude the decision to accept or reject the null hypothesis.

In scientific research, a hypothesis is a proposition put forth for consideration that a particular condition or assertion might be valid. The hypothesis is subsequently subjected to testing to determine its validity or falsity [54].

In the context of our investigation, it is noteworthy that the sentiment of tweets is deemed to be the independent variable, whereas the match outcomes constitute to the dependent variable. Accordingly, we shall formulate the subsequent Chi-Square hypothesis:

- Null Hypothesis ($H0$): There is no significant association between tweet sentiment and football outcomes (match results or league standings changes).

- Alternative Hypothesis ($H1$): There is a significant association between tweet sentiment and football outcomes (match results or league standings changes).

Apart from the Chi-Square test statistic, the p-value is also a crucial component of the Chi-Squared Test. The p-value represents the probability of obtaining a test statistic that is equal to or more extreme than the one calculated from the observed data, assuming that the null hypothesis is true. It is compared to a significance level, commonly set at 0.05, to determine whether to reject or fail to reject the null hypothesis.

In order to evaluate the validity of the null hypothesis, it is necessary to compare the calculated Chi-Square test statistic with the corresponding critical value, and also compare the derived P-value with the selected level of significance. If the computed Chi-Square test statistic exceeds the critical value, or if the P-value is below the predetermined level of significance, the null hypothesis is deemed invalid and rejected. Conversely, if the Chi-Square test statistic falls below the critical value, or if the P-value exceeds the significance level, the null hypothesis is accepted as valid.

It is important to note that while the Chi-Squared Test allows us to determine the existence of an association between the variables, it does not provide information about the strength or direction of the relationship. To assess the strength of the association, Cramers V will be used, which is a measure derived from the Chi-Squared statistic.

3.9.2 Cramér's V

Cramér's V is a necessary step after the Chi-Square test to determine the presence of an association to evaluate the strength of the relationship. Cramér's V is a statistic measure that ranges from 0 to 1, where 0 indicates no association between the two variables, and 1 represents a perfect association.

Cramer's V is derived from the Chi-Squared statistic by taking the square root of the Chi-Squared value divided by the total number of observations and the minimum of the number of rows minus 1 or the number of columns minus 1. The formula for calculating Cramér's V is as follows:

$$V = \sqrt{\frac{\chi^2}{n \min(r-1, c-1)}} \tag{3.3}$$

Where:

- V: is the value of Cramér's V
- χ^2: is the result of the Chi-Squared Test
- n: is the total number of observations
- r: is the number of rows in the contingency table
- c: is the number of columns in the contingency table

Once Cramer's has been calculated, it can be interpreted to determine the strength and direction of the relationship between the positive and negative sentiment of tweets and match outcomes. A value of 0 indicates that there is no

relationship between the two variables, while a value of 1 indicates a perfect relationship. In practice, however, values closer to 0.1 or 0.2 are more common, indicating a weak relationship, while values closer to 0.9 or 1 indicate a strong relationship.

3.9.3 Choice of Chi-Squared and Cramer's V over Correlation Analysis

Within the framework of our research, a deliberate choice was made to utilize the Chi-Squared Test and Cramers V instead of correlation analysis methods. The reasoning for this is decision is based on the categorical nature of the data under investigation. It is well-established that correlation analysis, such as Pearson or Spearman correlation, is more applicable for continuous or ordinal data. Given that our dataset comprises counts of positive and negative tweets, as well as categorical outcomes such as match results or league standings changes, the Chi-Squared Test and Cramers V are deemed to be more appropriate techniques for examining the associations between these variables.

3.9.4 Application of the Statistical Model

In order to conduct the Chi-Squared Test, a contingency table was first created for each of the two analyses (match results and standings changes), with rows representing the sentiment categories (positive or negative tweets) and columns representing the outcome categories (match results or standings changes).

	Match Won	Match Drawn	Match Lost
Positive Tweets	45,000	30,000	25,000
Negative Tweets	15,000	20,000	40,000

Table 3.8: *A sample contingency table of tweet sentiment and match outcomes*

	Increase	Decrease	No Change
Positive Tweets	139,000	21,000	45,000
Negative Tweets	24,000	73,000	64,000

Table 3.9: *A sample contingency table of tweet sentiment and changes in standings*

The contingency table was created in Python using the pandas library, and expected frequencies were calculated assuming independence between variables. The chi2-contingency [8] function from the scipy.stats library was used to find degrees of freedom and expected frequencies. The Chi-Square test statistic (X^2) and its corresponding p-value were calculated, with a manual calculation performed to validate the accuracy of the code. If the p-value is below the chosen significance level (e.g., 0.05), we can reject the null hypothesis and conclude a significant association between tweet sentiment and match results or league standings changes.

After determining the significance of the association using the Chi-Squared Test, Cramers V will be computed to measure the strength of the association. Through the calculation of Cramer's V, an evaluation will be conducted to discern the degree to which tweet sentiment is correlated with match outcomes or alterations in league standings. This process will facilitate comprehension not solely of the existence of an association but also of its magnitude.

[8] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html`

3.10 Conclusion

Chapter 4: Analysis of Results and Discussion

In the present chapter, we shall conduct a comprehensive evaluation of the system's performance by employing a variety of assessments. This meticulous examination will enable us to draw the requisite conclusions pertaining to the efficacy of our approach.

4.1 One

This section includes critical discussion about the Student's findings and shows how these findings support the original objectives laid out for the dissertation, which may be partially or fully achieved, or even exceeded. The Student may also include new areas of an investigation prompted by developments in the research dissertation. Above all, it is required to present strong arguments which show how findings may offer a valid contribution to the development of the subject of the selected research area or issues related to it. Percentage amount of words in section: 25 % of Dissertation

4.2 Two

4.3 Three

Chapter 5: Conclusions and Recommendations

5.1 One

In this chapter, the Student has to evaluate the significance of the work done and give recommendations for any further investigations. Percentage amount of words in section: 20 % of Dissertation.

5.2 Two

5.3 Three

List of References

[1] Eman MG Younis. Sentiment Analysis and Text Mining for Social Media Microblogs using Open-Source Tools: An Empirical Study. *International Journal of Computer Applications 112.5*, 2015.

[2] André Luiz Firmino Alves, Cláudio de Souza Baptista, Anderson Almeida Firmino, Maxwell Guimarães de Oliveira, and Anselmo Cardoso de Paiva. A comparison of svm versus naive-bayes techniques for sentiment analysis in tweets: A case study with the 2013 fifa confederations cup. In *Brazilian Symposium on Multimedia and the Web*, 2014.

[3] J. Surowiecki. *The Wisdom of Crowds*. Knopf Doubleday Publishing Group, 2005.

[4] Jennifer Watts-Englert and David Woods. Cooperative advocacy: An approach for integrating diverse perspectives in anomaly response. *Computer Supported Cooperative Work*, 18:175–198, 06 2009.

[5] M. Bharati and Bharati Ramageri. Data mining techniques and applications. *Indian Journal of Computer Science and Engineering*, 1, 12 2010.

[6] Andranik Tumasjan, Timm Sprenger, Philipp Sandner, and Isabell Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. volume 10, 01 2010.

[7] Huaxia Rui, Yizao Liu, and Andrew Whinston. Whose and what chatter matters? the impact of tweets on movie sales. *Decision Support Systems*, 55, 10 2011.

[8] Melanie Knopp. Machine learning and lexicon-based sentiment analysis of twitter responses to video assistant referees in the premier league during the 2019-2020 season. Master's thesis, Technische Universität München, 2020.

[9] Shiladitya Sinha, Chris Dyer, Kevin Gimpel, and Noah Smith. Predicting the nfl using twitter. *Proc. ECML/PKDD Workshop on Machine Learning and Data Mining for Sports Analytics*, 10 2013.

[10] Ryan Beal, Stuart Middleton, Timothy Norman, and Sarvapali Ramchurn. Combining machine learning and human experts to predict match outcomes in football: A baseline model. 02 2021.

[11] Abdullah Talha Kabakus, Mehmet Şimşek, and Ibrahim Belenli. The wisdom of the silent crowd: Predicting the match results of world cup 2018 through twitter. *International Journal of Computer Applications*, 182:40–45, 11 2018.

[12] Rob Schumaker, Tom Jarmoszko, and Chester Labedz. Predicting wins and spread in the premier league using a sentiment analysis of twitter. *Decision Support Systems*, 88, 06 2016.

[13] Stylianos Kampakis and Andreas Adamides. Using twitter to predict football outcomes. 11 2014.

[14] Daniel Gayo-Avello. ”i wanted to predict elections with twitter and all i got was this lousy paper” - a balanced survey on election prediction using twitter data. *ArXiv*, abs/1204.6441, 2012.

[15] Sethunya Joseph, Kutlwano Sedimo, Freeson Kaniwa, Hlomani Hlomani, and Keletso Letsholo. Natural language processing: A review. *Natural Language Processing: A Review*, 6:207–210, 03 2016.

[16] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, 82, 07 2022.

[17] Marco Palomino and Farida Aider. Evaluating the effectiveness of text preprocessing in sentiment analysis. *Applied Sciences*, 12:8765, 08 2022.

[18] Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. Feature subsumption for opinion analysis. pages 440–448, 01 2006.

[19] Bhumika Pahwa, Taruna S., and Neeti Kasliwal. Sentiment analysis- strategy for text pre-processing. *International Journal of Computer Applications*, 180:15–18, 04 2018.

[20] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcasm in Twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[21] Christine Liebrecht, Florian Kunneman, and Antal van den Bosch. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 29–37, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[22] Bing Liu. Sentiment analysis and opinion mining. volume 5, 05 2012.

[23] Nikil T and Aloysius Amalanathan. A comparative study of lexicon based and machine learning based classifications in sentiment analysis. 8:43–47, 12 2019.

[24] Akshi Kumar and Teeja Sebastian. Sentiment analysis: A perspective on its past, present and future. *International Journal of Intelligent Systems and Applications*, 4, 09 2012.

[25] C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01 2015.

[26] Mohammed Al-Shabi. Evaluating the performance of the most important lexicons used to sentiment analysis and opinions mining. 08 2020.

[27] Venkateswarlu Bonta, Nandhini Kumaresh, and Janardhan Naulegari. A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, 8:1–6, 03 2019.

[28] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[29] M S Neethu and R Rajasree. Sentiment analysis in twitter using machine learning techniques. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–5, 2013.

[30] Sai Vikram Kolasani and Rida Assaf. Predicting stock movement using sentiment analysis of twitter feed with neural networks. *Journal of Data Analysis and Information Processing*, 08:309–319, 2020.

[31] Prajakta P. Shelke and Ankita N. Korde. Support vector machine based word embedding and feature reduction for sentiment analysis-a study. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pages 176–179, 2020.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[33] Gintautas Garšva and Konstantinas Korovkinas. Svm and naïve bayes classification ensemble method for sentiment analysis. *Baltic J. Modern Computing*, 01 2018.

[34] Huma Parveen and Shikha Pandey. Sentiment analysis on twitter data-set using naive bayes algorithm. In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 416–419, 2016.

[35] Nazma Iqbal, Afifa Chowdhury, and Tanveer Ahsan. Enhancing the performance of sentiment analysis by using different feature combinations. pages 1–4, 02 2018.

[36] Peiman Barnaghi, Parsa Ghaffari, and John G. Breslin. Opinion mining and sentiment polarity on twitter and correlation between events and sentiment. In *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 52–57, 2016.

[37] Zhiying Hao. Deep learning review and discussion of its future development. *MATEC Web of Conferences*, 277:02035, 01 2019.

[38] Wael Etaiwi, Dima Suleiman, and Arafat Awajan. Deep learning based techniques for sentiment analysis: A survey. *Informatica*, 45:89–96, 08 2021.

[39] Enzo Grossi and Massimo Buscema. Introduction to artificial neural networks. *European journal of gastroenterology hepatology*, 19:1046–54, 01 2008.

[40] Sandip Lahiri and Kartik Ghanta. Artificial neural network model with the parameter tuning assisted by a differential evolution technique: The study of the hold up of the slurry flow in a pipeline. *Chemical Industry and Chemical Engineering Quarterly*, 15, 04 2009.

[41] Nhan Cach Dang, María N. Moreno-García, and Fernando De la Prieta. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3), 2020.

[42] Subarno Pal, Soumadip Ghosh, and Amitava Nag. Sentiment analysis in the light of lstm recurrent neural networks. *International Journal of Synthetic Emotions*, 9:33–39, 01 2018.

[43] Xin Huang, Wenbin Zhang, Yiyi Huang, Xuejiao Tang, Mingli Zhang, Jayachander Surbiryala, Vasileios Iosifidis, Liu Zhen, and Ji Zhang. Lstm based sentiment analysis for cryptocurrency prediction, 03 2021.

[44] Dr Murthy, Shanmukha Allu, Bhargavi Andhavarapu, and Mounika Bagadi. Text based sentiment analysis using lstm. *International Journal of Engineering Research and*, V9, 05 2020.

[45] Nan Chen and Peikang Wang. Advanced combined lstm-cnn model for twitter sentiment analysis. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 684–687, 2018.

[46] Shiyang Liao, Junbo Wang, Ruiyun Yu, Koichi Sato, and Zixue Cheng. Cnn for situations understanding based on sentiment analysis of twitter data. *Procedia Computer Science*, 111:376–381, 2017. The 8th International Conference on Advances in Information Technology.

[47] Jurgita Kapočiūtė-Dzikienė, Robertas Damaševičius, and Marcin Woźniak. *Sentiment Analysis of Lithuanian Texts Using Deep Learning Methods*, pages 521–532. 08 2018.

[48] Prateek Garg and Vineeta Bassi. Sentiment analysis of twitter data using nltk in python. 2016.

[49] Charles Malafosse. Open datasets for sentiment analysis, 02 2019.

[50] Mohammed Al-Shabi. Evaluating the performance of the most important lexicons used to sentiment analysis and opinions mining. 08 2020.

[51] Gurinder Singh, Bhawna Kumar, Loveleen Gaur, and Akriti Tyagi. Comparison between multinomial and bernoulli naïve bayes for text classification. pages 593–596, 04 2019.

[52] Arindam Banerjee. Hyperparameter tuning using randomized search, Nov 2022.

[53] Diogo Pacheco, Marcos Oliveira, and Ronaldo Menezes. Using social media to assess neighborhood social disorganization: a case study in the united kingdom. 05 2017.

[54] A Ugoni and Bruce Walker. The chi square test: an introduction. *COMSIG review / COMSIG, Chiropractors and Osteopaths Musculo-Skeletal Interest Group*, 4:61–4, 12 1995.

Chapter A: Introduction of Appendix

Interview summaries, sample questionnaires, and references should be placed in this section. For easier referencing, figures, tables, graphs, photos, diagrams, etc., should be inserted within the main text such as the literature review, the experimental process or procedure, the results and discussion chapters. Appendices are usually used to present further details about the results. Appendices may be a compulsory part of a dissertation, but they are not treated as part of the dissertation for purposes of assessing the dissertation. So any material which is significant to judging the quality of the dissertation or of the project as a whole should be in the main body of the dissertation (main text), and not in appendices.

Chapter B: Sample Code

You can share your GitHub link. Below shows how to insert highlighted source code from the source file.

```
# This function takes an integer n as input and returns the sum of numbers from 1 to n
def find_sum(n):
    sum = 0
    for i in range(1, n+1):
        sum += i
    return sum

# Ask the user for input and call the function
n = int(input("Enter an integer: "))
result = find_sum(n)

# Output the result
print("The sum of numbers from 1 to", n, "is", result)
```