MCAST

An Analytical Study of Sentiment Analysis Techniques for Evaluating Football-Related Tweets

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) 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 Mr. Carlo Mamo.

....

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

I would like to start by expressing my appreciation to my mentor, Mr. Carlo Mamo, whose unwavering support and knowledge have been instrumental throughout the course of this dissertation.

As well, I wish to convey my sincere appreciation to my family and friends, whose constant encouragement and support have been invaluable along the way. This support has been a source of unending motivation, for which I am profoundly thankful.

Lastly, in honor of my grandpa, who began the story with me but couldn't be there for its closing chapter.

Abstract

This study investigated the performance of different sentiment analysis models applied to football-related data sets, exploring how stop word strategies affected model performance. The investigation revealed differences in accuracy among lexicon-based, traditional machine learning, and deep learning models, with the hybrid CNN-LSTM model achieving the highest accuracy of 83.45% when no stop words were removed. Custom stop words and removal of stop words strategies showed significant lower accuracy to the hybrid model. Further, the research explored the correlation between public sentiment expressed through social media and football match outcomes. The findings suggested a positive correlation between match-winning probabilities and the positivity rate of pre-match tweets. However, the correlation was not strong enough to consider sentiment scores as the only factor in a team's winning probability. Lastly, the study looked into the relationship between the Positive Sentiment Score and the Points Per Game (PPG) achieved by teams throughout a football season. The results showed an increasing correlation between positive sentiment and team performance as the season progressed, although not consistently across all teams. This research contributes to understanding the role of public sentiment in sports analytics and offers implications for enhancing fan engagement strategies.

Keywords: Sentiment Analysis, Stopwords, Lexicon-Based, Traditional Machine Learning, Deep Learning, Match Outcomes, Points Per Game (PPG), Tweets

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BoW	Bag of Words
CNN	Convolutional Neural Network
GloVe	Global Vectors for Word Representation
LSTM	Long Short-Term Memory
ML	Machine Learning
NB	Naive Bayes
NLP	Natural Language Processing
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
VADER	Valence Aware Dictionary and Sentiment Reasoner
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Chapter 1: Introduction

In 2022, it was determined that the digital landscape has undergone a tremendous expansion, with around 4.95 billion people worldwide having a secure access to the internet. Additionally, approximately 4.62 billion individuals also possess an active social media account, representing around 58.4% of the global population[1]. Such figures underscore the role of the internet and social media platforms in the current time, resulting in profound implications for various aspects of human life. The digital discourse on social media platforms has, in fact, become a significant data source for gauging public sentiment.

A subject of increasing interest within this sphere is the sport of football. The sport has seen a noticeable surge in attention on social media platforms, leading to an increase in interest of sentiment analysis in relation to football-related discourse. Researchers have embarked upon explorations to discern the sentiments expressed about football teams, matches, players, and related topics, thereby contributing to the broader understanding of public sentiment in relation to the sport.

The emergence of improvements in machine learning and artificial intelligence coincides with the rising interest in sentiment analysis. These technologies have caused an evolution in research methodology, with a growing interest in training models to give more information regarding public attitude. This advancement in research methodology has opened up intriguing new study opportunities, particu-

[1] `https://datareportal.com/reports/digital-2022-global-overview-report`

larly in terms of leveraging the greatest tools to analyze public mood.

1.1 Problem Definition

The principal problem addressed in this research lies that despite the substantial volume of football-relate tweets and the burgeoning interest in the field, comprehensive exploration of these data sets, especially the one that will be used in this study, has been limited. Another limitation, is the knowledge of utilizing various stop word strategies to football-related data sets. Furthermore, the potential correlation between public sentiment, as reflected in tweets, and the outcomes of football matches still remains unclear.

1.2 Purpose Statement

The purpose of this research can be split in two: primarily, it seeks to investigate and evaluate the efficacy of different sentiment analysis models when applied to football-related data sets. In an effort to ensure a comprehensive analysis, different stop words techniques will be implemented during the sentiment analysis process to investigate if they effect football-related sentiment analysis models. By scrutinizing these models performance, the research aims to establish their utility, and accuracy in capturing and reflecting public sentiment in the realm of football.

Secondarily, this study aims to delve into the potential correlation between public sentiment, as expressed through social media, and football match outcomes. Specifically, it aims to examine the potential interplay between the sentiments expressed in pre-match tweets and the results of the football matches.

Moreover, this study seeks to observe the temporal evolution of public sentiment throughout a month. By analyzing and tracking this progression, the research intends to clarify whether the public sentiment various over time, and how these variations might align with the performances of the team.

As such, this research stands at the intersection of machine learning and social media analysis, which aims to enhance our understanding of football spectators and public engagement in an increasingly digital world.

1.3 Research Questions & Hypothesis

In alignment with the purpose of this study, the following research questions have been proposed:

1. How do lexicon-based, traditional machine learning, and deep learning sentiment analysis models differ in their accuracy when analyzing Malafosse's football related data set?

2. How do different stopword strategies (e.g., NLTK stop words, customized stop words, and no stop words) impact the performance of sentiment analysis football-related tasks?

3. How does the positivity rate of tweets relate with match-winning probabilities, as assessed through Logistic Regression?

4. What is the correlation between the Positive Sentiment Score and the Points Per Game (PPG) achieved by teams, and how does this relationship evolve over the course of a football season?

Based on these research questions, the following hypothesis will be proposed:

1. There are significant differences in accuracy among lexicon-based, traditional machine learning, and deep learning sentiment analysis models when applied to Malafosse's football-related dataset.

2. Different stopword strategies will have a significant impact on the performance of sentiment analysis models in football-related sentiment analysis tasks.

3. The increase in positive percentage is directly proportional to the increase in winning probability.

4. The relationship between positive sentiment on social media, specifically Twitter, and football team performance strengthens as the season progresses, with increased positive sentiment generally corresponding to higher Points Per Game (PPG).

1.4 Motivation

The motivation of this research is driven by the aim to combine the passion for football with the capabilities of machine learning. As a long-standing fan, the discourse and predictions before each match always intrigued my interest. Therefore, this motivation of this study can be split in two, it is an exploration of a lifelong fascination with football and a curiosity about the collective voice of its fans and the motivation by the desire to contribute to explore and improve technologies in the field of sentiment analysis.

1.5 Research Outline

Following the introduction, a literature review in Chapter 2 is carried out to investigate the background to sentiment analysis, natural language processing and machine learning, as well as related studies will be investigated.

In Chapter 3, we present the methodology, which includes the tools used and the pipeline of the systems. The process of training data collection, including data manipulation, stop words filtering, and machine learning techniques will be discussed. Additionally, we explain how the tweets were collected and how the statistical method was applied.

Chapter 4 focuses on the discussion of results. The outcomes of the models trained using various combinations of stop words and hyper-parameters will be presented. Additionally, the correlation between the percentage of positive tweets and teams' performance will be analyzed. Lastly, the research questions will be addressed and provide answers to the hypotheses

Conclusions in Chapter 5 will be then drawn with limitations and offer suggestions for future research.

Chapter 2: Literature Review

2.1 Introduction

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 [1]. 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 [2]. The underlying premise behind this notion is that crowds possess a more extensive base of knowledge [3].

2.2.1 Third-party Observations in Sentiment Analysis

Due to the informalities of a social media site like Twitter, researchers encounter several unique difficulties as social media text differs significantly from detecting sentiment in conventional text. . 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 [4]. 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 [5].

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 [6].

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 [7].

The relationship between social media activity and National Football League (NFL) games was studied by Sinha et al [8], 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 using Logistic Regression, and the technique correctly predicted the winner 63.8% of the time.

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 [9].

During the 2018 World Cup, a study was conducted to explore the potential signals in Twitter data and the impact of sentiment magnitude on successful match prediction. The study involved the creation of a database containing 38,371,358 tweets from 7,876,519 unique users. Nine different machine learning models were trained using data from the 48 matches in the group phase and then tested to predict outcomes for round 16 and beyond. The models took into account specific information about the user, such as the number of followers, location, likes, tweets, and other factors, 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 [10].

A study explored an odds-based methodology that analyzes an odds-maker's match balance sheet in response to wagers. Sentiment was derived by standardizing a specific data model against tweets corresponding to a particular club and match. The obtained sentiment 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 [11].

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 [12].

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 [13].

2.3 Natural Language Processing

Natural Language Processing (NLP) refers to the computational analysis of textual data. The field of NLP is a field that investigates how humans understand and use language. Its ultimate goal is to develop sophisticated methods and tools enabling computers to understand and interact with human languages. The intention is to equip these systems to perform a wide array of tasks according to need [14].

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 [15].

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 [16].

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 [17]. 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. An example of pre-processing components is described as follows:

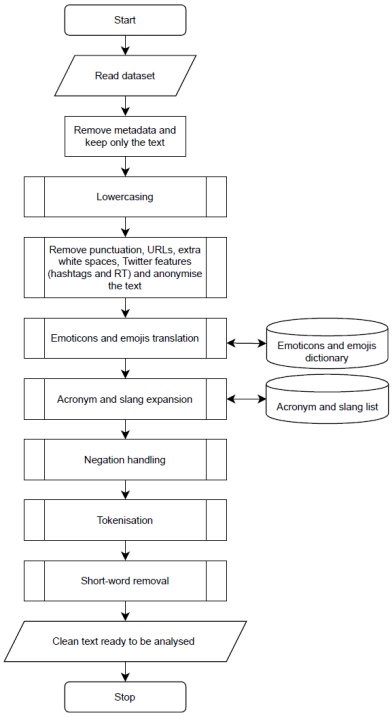

Figure 2.1: *A schematic diagram of the pre-processing steps* [16]

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 [18].

2.4 Lexicon-Based Approach

This section explores lexicon-based sentiment analysis techniques that use predetermined 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 [19]. This approach usually has two types of techniques: the Dictionary-Based Technique and the Corpus-Based Technique [20].

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 [21].

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 [22].

In a comparative study utilizing a lexicon-based approach, the efficacy of VADER was thoroughly evaluated. When gauging accuracy, VADER displayed superior performance by achieving the highest accuracy rate, specifically when

utilized with the Stanford Twitter Sentiment. This data set comprised of 182 tweets labeled as positive, 177 categorized as negative, and 139 designated as neutral. VADER achieved an accuracy of 72%. Conversely, the performance of SentiStrength, AFINN-111, and Liu-Hu lexicons remained relatively stable and unchanged [23].

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 [24].

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 [7]. 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 [25].

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 [7].

2.5.1 Traditional ML Approaches

The process of classification in Machine Learning involves the use of a training set and a test set, and is some cases also a validation test. This is dependent on the scale of the data set, which contain the 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. Some of the most commonly used machine learning techniques for sentiment classifications are Naïve Bayes (NB) and Support Vector Machines (SVM). When working with sentiment classifications, 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 [26].

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 Sentiment140 data set [27].

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 [28].

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 [29].

On the other hand, the Naïve 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,

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

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 [30].

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 [31]:

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

- *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*

In a study implementing the MultinomialNB package in Python, a Naïve Bayes classifier was trained using five-fold cross-validation, in order to avoid over-fitting with bag of words feature extraction. This technique estimate the conditional probability of features belonging to a particular class based on the assumption of a multinomial distribution [32]. The classifier demonstrated high performance in correctly categorizing tweets as positive or negative, achieving high accuracy, recall, precision, and F1-measure for several feature combinations.

Table 2.1: *Performance of Naïve Bayes classifier with different feature combinations on Sentiment140 and MovieReview datasets.*

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 stop words	0.85	0.85

Football events and sentiment analysis were investigated using Naïve Bayes and Bayesian Logistic Regression (BLR) classifiers using Term Frequency-Inverse Document Frequency (TF-IDF) approach. A tweet polarity classifier was trained on N-grams features (N=1 to 2) using WEKA as a machine learning framework, and a 10-fold cross-validation was applied. 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. The BLR model's performance was compared to that of Naïve Bayes, and both models demonstrated optimal performance when utilizing Uni-grams and Bi-grams feature combinations [33].

Table 2.2: *Performance comparison of Bayesian Logistic Regression (BLR) and Naïve Bayes (NB) classifiers with different feature combinations.*

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

Lastly, both Bag of Words (BoW) and TF-IDF were tested as feature ex-

traction methods for different machine learning algorithms. The results showed that Naïve Bayes performed poorly with BoW, however, when using the TF-IDF feature extraction method, Naïve Bayes' accuracy significantly improved to 75.27% [34].

Table 2.3: *Experiment conducted with TF-IDF and the Naïve Bayes Model*

Algorithm	Accuracy	Recall	Precision	F-Measure
Naïve Bayes	75.27%	0.79	0.86	0.69

In conclusion, both SVM and Naïve 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, Naïve 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, 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 [35].

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 avoid-

ing the time-consuming and often incomplete process of manual feature engineering. Moreover, deep learning produces high-quality features and multiple levels of representation, which significantly enhance the efficacy of machine learning models [36].

Moreover, Artificial Neural Networks (ANN) are computational systems inspired by the human brain, capable of modifying their structure to solve nonlinear problems by reconstructing ambiguous rules. They consist of nodes, receiving input and producing output via connections with associated functions and strength denoted by positive or negative values [37].

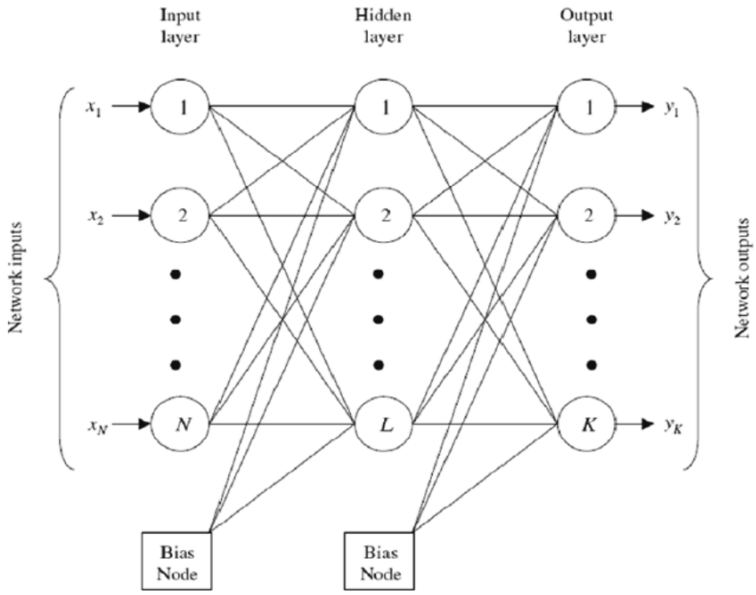

Figure 2.2: *A schematic diagram of artificial neural network and architecture of the feed forward network with one hidden layer*

[38]

These neural networks utilize advanced mathematical modeling techniques to process data in various ways. Moreover, 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 [39]. 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.

Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) are deep learning techniques widely employed for sentiment analysis. LSTM is a specific type of Recurrent Neural Network (RNN) which utilize memory cells with input, forget, and output gates to regulate signal flow, enabling successful RNN training. These models make them suitable for sentiment analysis tasks by their capacity to learn long-term relationships between words and phrases. The forget (f) gate in LSTMs allows for state resets, enabling accurate predictions by looking back over multiple timesteps [40]. 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)

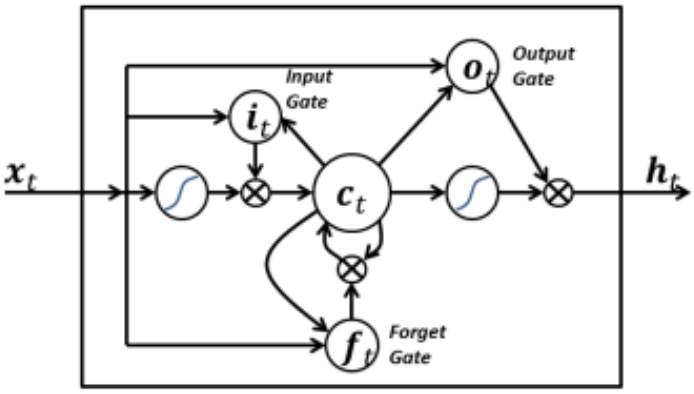

Figure 2.3: *An LSTM unit with input i, output o, and forget f gates*

In a study focusing on sentiment analysis of crypto currency on social media posts, Long Short-Term Memory (LSTM) was presented as a suitable sentiment analyzer, capable of learning long-term relationships between words and phrases through forget and remember gates [41]. The LSTM model was trained on a sequence of crypto word embeddings generated from tokenized social media posts. The model's output was transformed using a fully connected layer and a sigmoid function. The authors evaluated the LSTM sentiment predictor on Sina-Weibo posts from the top 100 crypto investors' accounts and found that it outperformed the time series auto regression (AR) approach by 18.5% in precision and 15.4% in recall, highlighting LSTM's effectiveness in sentiment analysis and its potential in cryptocurrency trading.

Murphy et al. [42] also employed LSTM to assess the effectiveness of deep learning techniques in sentiment analysis using an IMDB and Amazon review

dataset. The dataset comprised 50,000 equally polarized reviews, with 50,000 used for training and 23,500 for model evaluation. The authors found that LSTM-based deep learning techniques outperformed other methods with an accuracy of 85%, using a dropout rate of 0.2, Adam optimizer, sparse categorical cross-entropy loss function, and a batch size of 500. This study emphasizes 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 [43]. The architecture of the network comprises of convolutional and sub sampled 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.

Several studies have explored the application of convolutional neural networks (CNNs) for sentiment analysis tasks in different contexts. For example, Liao et al. [44] proposed a machine learning-based approach to classify the sentiment of Twitter data using a CNN. Their model was trained on the MR and STS Gold datasets, which consists of movie reviews and real tweets, and achieved a development accuracy of 74.5% on the MR dataset and 68% on the STS Gold dataset. The authors detailed various parameter settings, such as filter windows, dropout rate, l2 regularization lambda, and batch size, providing insight into the optimal configuration for this task.

In a separate study, researchers investigated the use of CNNs and Long Short-

Term Memory (LSTM) deep learning neural networks for sentiment analysis on the Lithuanian Internet comment dataset. In this case, the models were tested using both word2vec and FastText word embeddings. The results demonstrated that the CNN model provided accurate results in 70.6% of instances. The authors concluded that deep learning neural networks exhibit favorable performance when applied to smaller datasets, suggesting the potential for further exploration in this area [45].

2.5.3 Hybrid Approach

A hybrid-approach is an approach where certain researchers choose to employ hybrid solutions, wherein they integrate the utilization of machine learning with another technique, for the purpose of constructing a sentiment analysis model.

Rehman et al. [46] and Venkatesh et al. [47] proposed hybrid models that combine Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) techniques. However, a key distinction between the two studies is their choice of word embedding techniques.

The study conducted by Rehman et al. [46] utilized Word2Vec, a technique that translates text into vector values based on word meaning and clusters similar words together. The researchers proposed a hybrid model that combined CNN and LSTM layers, wherein the CNN layers extracted local features while the LSTM layers managed long-term dependencies in word sequences. The final classification layer employed a sigmoid function. The results showed a significant improvement in the F-measure score, ranging from 4-8% compared to individual CNN and LSTM models on the IMDB dataset. Furthermore, the proposed hybrid

model outperformed traditional machine learning techniques such as Naïve Bayes, Support Vector Machine, and Genetic Algorithms in terms of accuracy.

On the other hand, the study made by Venkatesh et al. [47], adapated the CNN-LSTM hybrid model with GloVe model which focuses on capturing both global and local semantic relationships between words by leveraging co-occurrence statistics with a similar architecture. The only difference is that it features a single LSTM layer rather than the multiple LSTM layers in the hybrid model, and the final classification layer employs a softmax function instead of a sigmoid function.

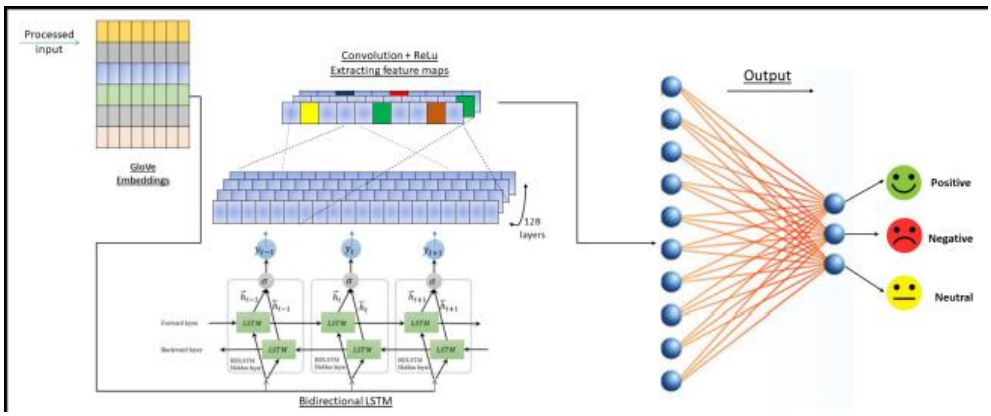

Figure 2.4: *Proposed hybrid CNN+LSTM model with GloVe*

The CNN-LSTM with GloVe model, displayed strong performance in sentiment analysis tasks, with the following reported metrics:

Category	**Precision**	**Recall**	**F1-Score**	**Support**
Positive	0.90	0.89	0.89	560
Negative	0.78	0.63	0.70	115
Neutral	0.78	0.83	0.81	326
			Accuracy:	85%

Table 2.4: *Performance metrics for the CNN-LSTM with GloVe model*

Both the hybrid CNN-LSTM model and the CNN-LSTM with GloVe model

offer promising approaches to sentiment analysis by capitalizing on the strengths of CNN and LSTM networks. Each model exhibits unique advantages concerning word embedding techniques and model architectures.

2.6 Conclusion

This chapter reviewed the use of sentiment analysis in the field of sports and the effectiveness of employing machine learning and deep learning techniques to generate sentiment scores. Based on the research reviewed, it was concluded that VADER, Naïve Bayes, and a Convolutional Neural Networks (CNN) - Long Short-Term Memory (LSTM) with GloVe model are the most accurate approaches for sentiment score prediction. Our study will focus on comparing the performance of VADER, Naïve Bayes, and the CNN-LSTM GloVe model to identify the most suitable approach for sentiment analysis in our context. The following chapter covers the research methodology and implementation of the chosen models.

Chapter 3: Research Methodology

3.1 Introduction

This chapter aims to provide a detailed explanation of the research methodology applied in this dissertation, particularly focusing on the techniques and models deployed for sentiment analysis, data processing, and analysis. This study follows a quantitative approach, leveraging statistical analysis and numerical data to address the research questions.

The research embodies a quantitative nature through its employment of different sentiment analysis models and techniques, and its reliance on statistical tests to determine relationships between sentiment categories, match results, and points per game. It utilizes numerical data and statistical analysis to generate a more nuanced understanding of the study's objectives and hypotheses.

The subsequent sections will further delve into the particular models and techniques implemented in this research, in addition to outlining the data collection and analysis procedures.

3.2 Prototype Pipeline

This prototype pipeline diagram visually represents the key stages and processes involved in this study. This provides a clear overview of the research methodology, illustrating the flow of data and sequential steps taken from data collection to the final analysis. The pipeline serves as a useful guide for understanding the

interconnections between different components of the study and offers a basis for further elaboration on the specific models, techniques, and procedures employed.

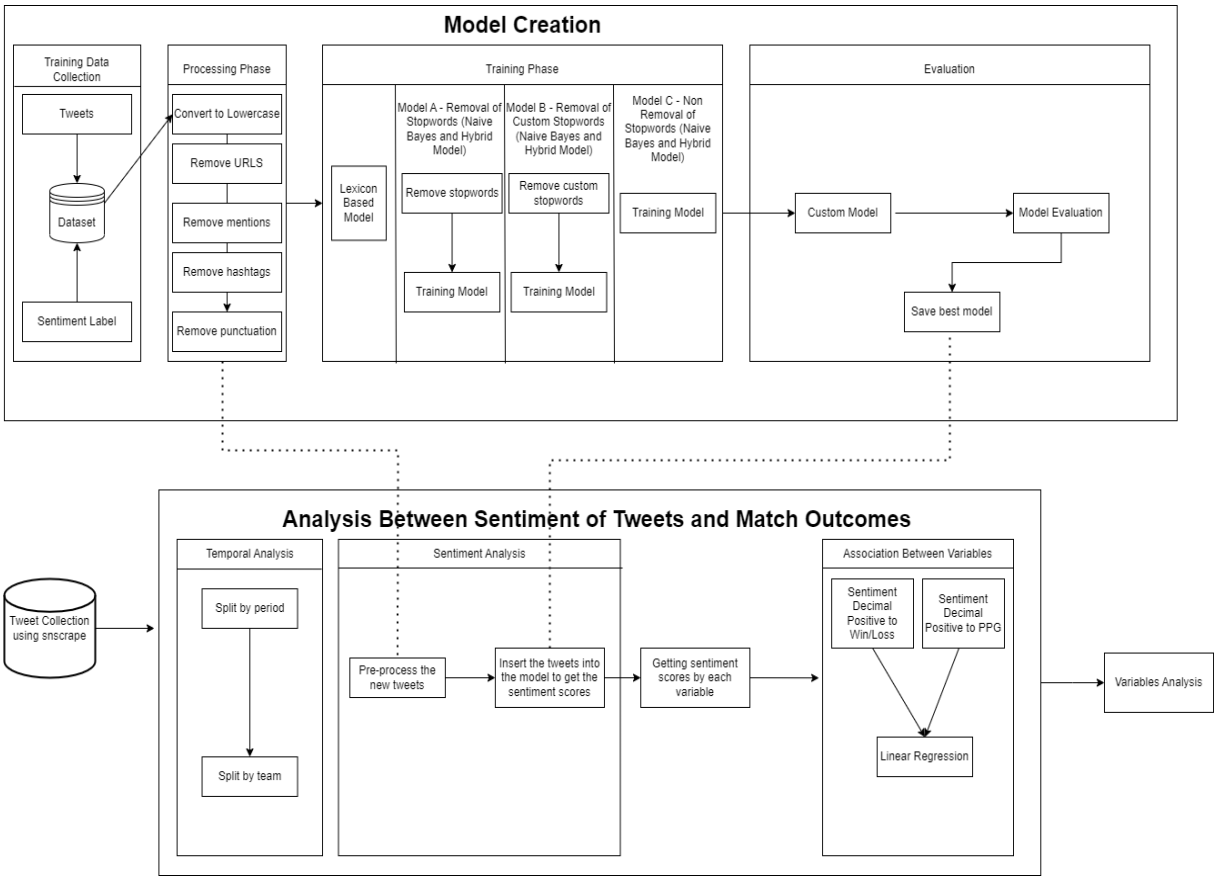

Figure 3.1: *Research Pipeline*

3.3 Frameworks and Libraries

In developing the sentiment analysis system, Python was utilized as the programming language of choice due to its versatility, simplicity, power, and dynamic nature, as well as its proficiency in processing natural language data. Python's extensive collection of open-source libraries and large user community further contributed to its appeal and functionality for various applications. Additionally, to facilitate ease of package management, the virtual environment was created and activated using the Anaconda platform, allowing for seamless installation, updates,

and removals of necessary packages. The Anaconda version 22.9.0 was employed for this research.

The main packages used in this study are as follows:

Package	Description	Sentiment Analysis Applications
pandas	Data manipulation library for structured data	Data preprocessing, cleaning, and feature extraction
nltk	Natural language processing library	Text preprocessing, tokenization, stemming, and sentiment analysis
sklearn	Machine learning library	Building and evaluating sentiment analysis models
scipy	Collection of mathematical and scientific functions for numerical computation	Supporting mathematical operations in sentiment analysis algorithms
torch	Deep learning library with GPU acceleration	Building and training deep learning models for sentiment analysis
numpy	Numerical computing library for array and matrix operations	Data preprocessing and backend support for other libraries
optuna	Hyperparameter optimization framework	Fine-tuning models for improved sentiment analysis performance

Table 3.1: *Summary of Python packages for sentiment analysis*

3.4 Training Dataset

The data set utilized for training the football sentiment models is secondary data sourced from a publicly available GitHub repository [1]. As secondary data, this data set was initially collected and used by others for their own purposes. It comprises three discrete files that pertain to football-related tweets, player tweets, and World Cup tweets, along with the associated sentiments expressed. For the purpose of this research, which solely focuses on the sentiment of football teams, the data sets of both teams and players were merged.

[1] `https://github.com/charlesmalafosse/open-dataset-for-sentiment-analysis`

3.4.1 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 removed, 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 the sentiment analysis model, a language verification was executed to corroborate that all the tweets within the data set were composed in the English Language. Upon inspecting the sentiment analysis categories of the data set, the mixed category was removed due to the relatively small number of tweets assigned to it. Since the tweets were not equally distributed for each category, the categories were down-sampled to a uniform count of 354,501 tweets per classification. This action was deemed necessary to maintain the integrity of our analysis and avoid potential inaccuracies in the model training and evaluation process.

The pre-processing pipeline of the training tweets included removing redundant words, abbreviations, and unwanted patterns such as URLs, usernames, punctuation's and hashtags.

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 3.2: *Data Refinement Process*

This table below provides a clear comparison of tweets before and after the pre-processing tasks and showcasing the cleaned and simplified tweet text.

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

3.4.2 Stop Words Filtering

stop words are common words in a language that are often considered insignificant and removed from text data during natural language processing tasks. They are typically nouns, prepositions and conjunctions which carry little meaning on their own.

However, in the context of analyzing football-related tweets, eliminating stop

words may inadvertently strip away important contextual information. For instance, the tweet ”My team didn’t play well”, removing stop words leaves us with ”team, play, well” which may lead to misinterpretation or loss of sentiment.

To address this issue, three different approaches are proposed for the models in processing football-related tweets.

1. This involves training the models using the standard NLTK stop words list, which is recommended by researchers for normal sentiment analysis tasks.

2. This approach involves creating a customized stop words list by retaining important words from the NLTK list that are relevant to the football domain:

Stop word	Original Tweet	Processed Tweet	Reason for Importance in the context
Off	”The players look off today”	”players look today”	Maintains poor performance context
Over	”The game is over”	”game”	Keeps game over context
Under	”Our team is under performing”	”team performing”	Losing context of superlative degree
Few	”Few supporters showed to the game”	”supporters showed game”	Losing quantity of supporters context

More	”Our team needs more training”	”team training”	Losing superlative degree context
No	”No goals were scored in the match”	”goals scored”	Losing lack of goals context
Not	”Our team did not play well today”	”team play well today”	Losing the negation of the sentiment
Don’t	”Don’t underestimate our team”	”underestimate team”	Losing advice or warning context
Should	”We should have played better”	”played better”	Losing the context and sentiment
Should’ve	”We should’ve won the match”	”won match”	Losing negative result context
Aren’t	”Our players aren’t performing well”	”players performing well”	Losing the context of negation
Couldn’t	”Our team couldn’t score a goal”	”team score goal”	Losing the context of inability
Didn’t	”We didn’t win the match”	”win match”	Losing the context of negation
Doesn’t	”Our team doesn’t give up easily”	”team give easily”	Losing the context of negation

Hadn't	"We hadn't practiced enough before the game"	"practiced enough game"	Losing the context of negation and past event
Haven't	"We haven't lost a game this season"	"lost game season"	Losing the context of negation
Mustn't	"We mustn't lose focus during the match"	"lose focus match"	Losing advice or warning context
Shouldn't	"We shouldn't have underestimated the other team"	"underestimated team"	Losing recommendation context
Wasn't	"The referee wasn't fair in his decisions"	"referee fair decisions"	Losing the context of negation
Weren't	"Our players weren't in good form today"	"players good form today"	Losing the context of negation
Won't	"Those decisions won't help the team"	"decisions help team"	Losing the context of negation
Wouldn't	"Our coach wouldn't make such a decision without reason"	"coach make decision reason"	Losing negation and assumption context

3. Entails not using any stop words, allowing the models to process the

tweets in their entirety.

The NLTK stop words list accounts for words with an apostrophe, such as ”wouldn’t”, but does not include variations without the apostrophe, like ”wouldnt”. Given that many tweets may contain informal language or unconventional spellings, a customized function has been implemented to accommodate these variations and ensure that stop words without apostrophes are also considered during the text processing.

The primary aim of conducting a comparative analysis of the performance of these methodologies is to ascertain the most efficacious approach with regard to the utilization of stop words in the context of football-related tweets.

3.5 Model Selection

The primary objective of this chapter is to provide a systematic approach to selecting the most effective model for sentiment analysis to predict football-related tweets and how these models differ from each other in terms of performance. As the field of sentiment analysis encompasses a diverse range of methods and techniques, it is imperative to identify a model that offers superior performance and reliability when applied to the unique characteristics and nuances of football-related content.

3.5.1 VADER Implementation

In the present study, the Valence Aware Dictionary and Sentiment Reasoner (VADER) lexicon-based model was utilized as an alternative approach to classify the senti-

ment of football-related tweets. VADER constitutes a lexicon and rule-based sentiment analysis technique, explicitly designed for discerning sentiment expressed in social media text. This method considers the intensity of emotions and furnishes a compound score indicative of the sentiment present within a given textual sample.

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 Al Shabi [48], 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.

3.5.2 Naive Bayes Implementation

The chosen machine learning algorithm as described in the previous chapter for sentiment analysis is the Multinomial Naïve Bayes classifier. 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 having a superior classification performance in comparison to a Bernoulli Naive Bayes classifier, which solely takes into account the presence or absence

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

of words.

The study aims to classify the sentiment of football-related tweets using this probabilistic approach. The classifier uses two distinct techniques: Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF).

Bag of Words Approach

The Bag-of-Words (BoW) model representation was used using CountVectorizer [3], which 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. This approach doesn't take into consideration the order of words; it merely focuses on their presence and frequency.

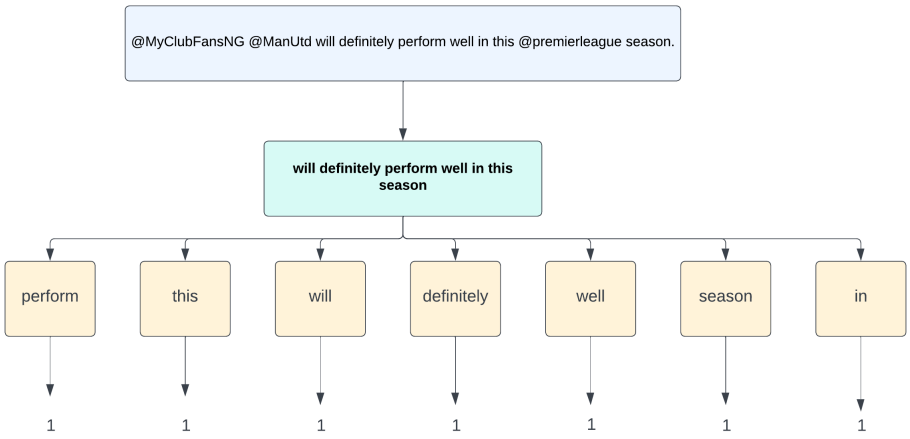

Figure 3.2: *Example of a Training Tweet in Bag of Words Representation*

The following parameters were used to test the best parameters for the BoW model:

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

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.
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 3.5: *Hyper-parameters considered for the BoW Naive Bayes model.*

Term Frequency-Inverse Document Frequency (TF-IDF)

The subsequent prototype that was tested made use of the TF-IDF approach, an alternative to Bag of Words (BoW) representation that takes the importance of words in a given dataset into account. The inverse document frequency (IDF) and word frequency (TF), two separate measurements, are the foundation of this methodology. The former calculates a word's frequency of use within a specific document, whereas the latter calculates a word's relevance across all texts in the collection. As a result, the resulting TF-IDF score gives words that are more significant but less common across all publications more weight.

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 3.6: *Description of Term Frequency and Inverse Document Frequency*

In addition to the parameters that were used to the Bag of Words implementation, the ensuing parameters were also utilized to attain the optimal parameters possible:

Hyperparameter	**Description**
use-idf	A boolean flag that determines whether to use the Inverse Document Frequency (IDF) weighting..
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.
fit-prior	A boolean flag that determines whether to learn class prior probabilities from the training data. If set to True, the classifier will estimate the prior probabilities based on the training data. If set to False, it will use uniform prior probabilities, assuming that each class is equally likely.

Table 3.7: *Hyper-parameters considered for the TF-IDF Naive Bayes model.*

For each sentiment class, the model calculates the probability that the input

text belongs to that class. This is done by multiplying the conditional probabilities of each token (TF-IDF value) given the sentiment class, and then multiplying by the prior probability of the class (based on the proportion of each class in the training dataset).

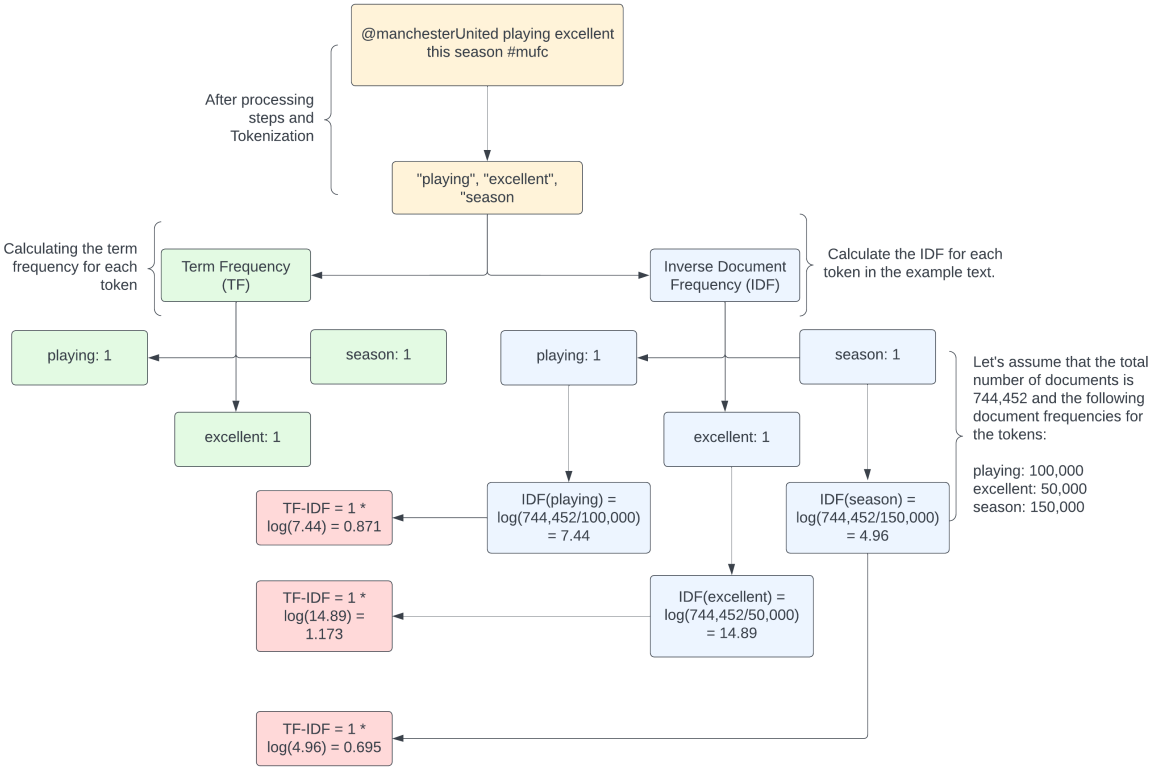

Figure 3.3: *Example of the TF-IDF process*

Training the Model and Hyper-parameter Tuning

The dataset was split into a training, validation and test dataset, with a 70:15:15 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 [4] function from the Scikit-learn library was used. RandomizedSearchCV is an alternative to GridSearchCV, another popular method for hyper-parameter tuning.

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 [49]. This approach is more computationally effective, particularly when working with numerous hyper-parameters or a huge search space. However, it is crucial that the model be trained using the best hyper-parameters available. Consequently, GridSearchCV emerges as the superior solution, given that RandomizedSearchCV encompasses a limited set of hyper-parameters in its pursuit of attaining the highest accuracy.

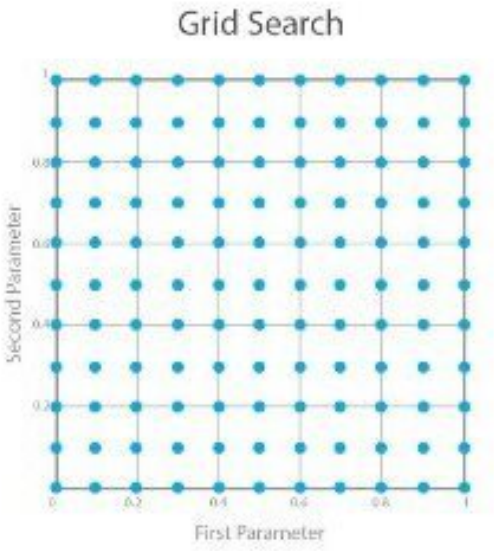

Figure 3.4: *An illustration of a grid search space. A range of the possible parameters and the algorithm makes a complete search over them.*

In an effort to improve upon the methodology employed in a previous Naive Bayes study by Iqbal et al. [32], the GridSearchCV instance employed a 10-fold

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

cross-validation approach to ensure a more thorough evaluation of the model's performance across different hyper-parameter configurations. The optimal parameters, as determined by the validation accuracy, were evaluated using the test dataset to determine if there were any discrepancies in accuracy when stop words were used, including customized stop words, in comparison to the model without stop words. The evaluation metrics included accuracy and a classification report, which provided detailed information on precision, recall, and F1-score for each class.

3.5.3 Hybrid CNN+LSTM Implementation

This part outlines the methodology employed in this study, which combines a hybrid CNN-LSTM model with GloVe word embeddings to perform sentiment analysis. The following sections detail the steps taken to achieve the best possible accuracy, the hyper-parameters tested, the validation and test accuracy, and other relevant information.

Data Preparation

The initial step in model development was data preparation and pre-processing, involving tokenization and padding of input text. Tokenization is a critical step in natural language processing, in which it breaks raw text into smaller units or tokens. In our implementation, tokens represent individual words in the input text, converting unstructured data into a structured format for machine learning models.

Keras Tokenizer class was employed for tokenization, initialized with a 'num-

words' parameter set to a maximum of 30,000. This limits the vocabulary size and reduces computational requirements, considering only the top 30,000 frequent words in the dataset. This is done as less frequent words, often seen as noise, may not significantly impact 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 methodology employed in this study leverages a hybrid CNN-LSTM model with Global Vectors for Word Representation (GloVe) word embeddings to enhance sentiment analysis performance. GloVe is an unsupervised learning algorithm that generates pre-trained word vectors by analyzing the co-occurrence statistics of words in large text corpora. These embeddings capture the semantic meaning of words in a high-dimensional space, allowing for efficient representation of their relationships and contextual similarities.

GloVe embeddings are utilized in this study to provide a rich, context-aware representation of words in football-related tweets. By initializing the hybrid CNN-LSTM model with these pre-trained vectors, the model can benefit from the semantic knowledge already embedded in the vectors, potentially improving the model's ability to understand and analyze the sentiment in the text. The rationale behind using a hybrid CNN-LSTM model for sentiment analysis lies in the strengths of both convolutional and recurrent layers in processing sequential data like text.

Convolutional layers are adept at detecting local patterns or features within

the input text, such as n-grams or phrases, that can be indicative of sentiment. These local features are often translation-invariant, meaning they can be recognized regardless of their position in the text. On the other hand, LSTM layers are designed to capture long-range dependencies and contextual information by maintaining a hidden state that can store information from earlier tokens. By combining both CNN and LSTM layers, the model can effectively identify local patterns as well as their context, resulting in improved sentiment analysis performance.

Hence, the architecture of the model is outlined below:

Layer	**Description**
1. Pre-trained Glove Embedding Layer	Pre-trained word embedding for mapping input tokens to fixed-size vectors
2. 1D convolutional layer	Identifies local patterns or features within the input text.
3. ReLU Activation Function	Introduces non-linearity into the model using ReLU function.
4. 1D max-pooling layer	Reduces the spatial dimensions of feature maps, selecting the most important features and reducing over-fitting.
5. Bidirectional LSTM layer with dropout	Captures context from both past and future tokens in the input text.
6. Fully connected layer	Combines the features extracted by the previous layers to make a final decision
7. Softmax activation function	Converts the logits into probabilities for each sentiment class, allowing for the prediction of the highest probability sentiment.

Table 3.8: *Hybrid CNN-LSTM Architecture*

In the final stage of the model, the fully connected layer combines the features extracted by the previous layers to make a final decision about the senti-

ment. This decision is based on the patterns and contextual information captured by the convolutional and LSTM layers, taking into account the positive, negative, or neutral aspects of the text.

The softmax activation function in the last layer then converts the logits from the fully connected layer into probabilities for each sentiment class (e.g., positive, negative, and neutral). The model predicts the sentiment class with the highest probability as the final sentiment of the given text. This way, the model categorizes the input text into one of the three sentiment classes based on the features and context identified throughout the layers.

Parameter Tuning and Model Training

In order to optimize the model performance, an extensive search for the best hyper-parameter to optimize the models' performance was done using Optuna[5], an optimization framework. The hyperparameters under consideration included the number of filters, filter size, pool size, LSTM output size, and dropout rate. A total of 96 combinations were executed to identify the most suitable set of hyper-parameters. The model was trained and evaluated on the test dataset for each set of hyperparameters, with accuracy scores and hyperparameters recorded for each trial.

The following table presents the hyper-parameters and their respective ranges considered during the optimization process:

[5] `https://optuna.org/`

Hyperparameter	Description	Range
No. of filters	Number of convolutional filters	32 to 256 with a step of 32
Filter size	Size of convolutional filters	3 to 7 with a step of 2
Pool size	Size of max-pooling filters	2 to 4 with a step of 2
LSTM output	Number of LSTM output nodes	64 to 512 with a step of 64
Dropout rate	Dropout rate for regularization	0.1 to 0.5 with a step of 0.1

Table 3.9: *Hyperparameters and their respective ranges considered during tuning*

The model was evaluated on both validation and test data sets with a respective distribution of 75:15:15. The validation data set was used during hyper-parameter optimization, in which the best hyper-parameters was used to train the model. The model was trained for five epochs, and the loss was calculated using the cross-entropy loss function and the adam optimizer was used to update the model parameters. Ultimately, the performance of the final model was evaluated using the test dataset. Furthermore, an analysis was conducted to determine if there were any variations in the model's efficacy when utilizing stop words and customized stop words. The evaluation metrics used were accuracy, confusion matrix, and classification report.

3.6 Data Collection

This section outlines the methodology for acquiring tweets, which constitutes the primary data for this study. This primary data collection process involves employing prominent team-specific hashtags and focusing on discrete intervals within the football season. This approach facilitates the understanding of sentiment and discourse evolution throughout the progression of the season.

Given that our study is divided into different periods within the 2021-2022

season, the Twitter API proved unsuitable due to its 7-day retrieval limit. Consequently, "snscrape"[6] [7] was employed as an alternative to gather the primary data, and a more comprehensive range of tweets throughout 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 2021/2022 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. [11] and Kampakis and Adamides [12]. The hashtags utilized in this research align with those used in studies by Kampakis and Adamides [12] and D. Pacheco et al. [50].

Table 3.10: *Premier League Teams and Hashtags*

Team	Hashtag	Team	Hashtag	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 983,119 tweets was conducted to identify any sarcasm-related hashtags, and such tweets were removed accordingly, in accordance with the guidelines outlined by C. Liebrecht et al. [51]. This task was accomplished by importing all the tweets into SQL tables and subsequently constructing a query designed to identify the presence of any

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

[7] As of the publication of this dissertation in June 2023, all external Twitter APIs have been blocked. Consequently, these resources are no longer functional beyond the aforementioned date. It should be noted, however, that these APIs were actively utilized during the research process prior to their deactivation.

sarcasm-related hashtags within the tweets.

Table 3.11: *Data collection dates for each time period during the 2021/2022 Premier League season*

Dates	**Number of Games**
1st December 2021 - 1st January 2022	44
1st April 2022 - 1st May 2022	43

3.6.1 Sentiment Analysis

Upon the completion of the data collection, a sentiment analysis was executed on the tweets to ascertain the expressed sentiment towards each team, divided on a weekly basis and within two distinct months.

The most accurate model from the models investigated previously was utilized in order for the sentiment of the tweets to be predicted in which these were subsequently preserved as a newly-created CSV file. In order to maintain consistency and comparability between the training data set and the collected data, the same pre-processing methods were applied.

For the analysis of relationships between match results and sentiment, this study will specifically focus on the sentiment expressed in tweets collected within a time frame of three days prior to each match. The rationale behind focusing on the sentiment expressed in tweets collected within a time frame of three days prior to each match is to ensure that the analysis captures the prevailing sentiment specifically related to the upcoming match. Furthermore, the tweets published on the day of the match were solely taken into consideration up until one hour prior to the start of the game.

By narrowing down the time frame, the study aims to minimize the influence of tweets reflecting sentiments about events that may have occurred in the days preceding the match, which could potentially skew the analysis which was a similar approach taken by Sinha et al [8]. Additionally, the assessment of team performance will encompass a more extensive period, taking into account the entire month for the calculations. During this month-long period, Points Per Game (PPG) will be used as the primary measure to assess team performance instead of the league standings changes. This approach accounts for the varying number of games played by different teams within a month, ensuring a more equitable comparison of their performance.

$$PPG = \frac{Total\ \ Points}{Total\ \ Games\ \ Played} \tag{3.1}$$

In order to classify the Points-Per-Game contingency table, a quartile-based stratification method was utilized, wherein the positions of the four quartiles were ascertained using the QUARTILE [8] function in Microsoft Excel. This process was conducted separately for each month under consideration, as combining the values from the two distinct periods might not yield an appropriate representation, given the potential differences between them. By employing this method, the teams were systematically divided into four distinct categories for each month, ensuring an equitable distribution.

Furthermore, the tweets demonstrating neutral sentiment were excluded from the dataset, as they frequently pertained to matters unrelated to the team's perfor-

[8] https://support.microsoft.com/en-us/office/quartile-function-93cf8f62-60cd-4fd

mance. The newly generated file encompassed the original tweet text in conjunction with the predicted sentiment for each respective tweet. This procedure facilitated the expeditious analysis and visualization of sentiment trends throughout the specified periods, thereby enabling a correlation with the team's performance to be established.

3.7 Association Between Variables

In this last section, a comprehensive overview of the methodology will be provided that used to investigate the relationship between sentiment analysis results obtained from the best-performing model and match outcomes, as well as Points per Game (PPG) per month. The objective is to elucidate the process of examining the correlations between the sentiment expressed in football-related tweets and their impact on match results and overall team performance.

3.7.1 Logistic Regression

In this study, a Logistic Regression model is employed to explore the relationship between the predictor variables and the outcome. This choice is motivated by the statistical properties of this model, and its relevance to the binary nature of the outcome in our data (i.e., match win or not).

Logistic Regression is a statistical method that models the log-odds of a binary outcome as a linear function of predictor variables. Its name stems from the logistic function, also known as the sigmoid function, that it uses to transform the output of the linear regression into a probability between 0 and 1. The Logistic Regression model calculates the probability of an event occurring (e.g.,

winning a match) using the logistic function, which is defined as follows:

$$p(x) = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

Where:

- $p(x)$ represents the probability of the event occurring,
- x is the sentiment analysis percentage,
- b_0 is the intercept,
- b_1 is the coefficient associated with the sentiment analysis percentage.

In order to ensure the normalization of tweets, considering that the number of teams may not be evenly distributed, each team's percentage of positive and negative tweets was calculated. Specifically, when examining the positive tweets related to match outcomes, a binary array was generated, where "1" represented a win and "0" denoted a draw or loss and the decimal of the percentage of positive tweets. Likewise, a similar process was applied to assess the loss and negative sentiment. The percentage of negative tweets for each team was determined, and "1" indicated a loss while "0" indicated a draw or win.

While Logistic Regression makes use of a linear equation to estimate the log-odds, the transformation of these log-odds to probabilities via the logistic function allows for a non-linear relationship between the sentiment analysis percentage and the probability of winning a match. Therefore, it is well-suited to handle data sets like ours, where the relationship between the sentiment analysis percentage and the probability of winning a match may not be strictly linear.

3.8 Conclusion

In this chapter, various approaches to sentiment analysis employing distinct pre-processing techniques and configuration have been explored. By utilizing hyper-parameter tuning, an extensive array of combinations was also discussed to ascertain the most effective method for achieving optimal performance in sentiment analysis. Finally, the statistical models employed for evaluating the relationships between football-related tweets and the corresponding sentiment scores, while considering diverse criteria and time frames has also been discussed.

In the forthcoming chapter, a comprehensive presentation and analysis of the results derived from the evaluation of all tested models will be evaluated. This will facilitate the identification of the best-performing model for sentiment analysis in the context of football-related tweets. Additionally, we will scrutinize the hypothesis outlined in Chapter 1, determining the extent to which they hold true in light of the empirical evidence obtained through our analyses.

Chapter 4: Analysis of Results and Discussion

In the present chapter, we shall conduct a comprehensive evaluation of the model's performance by employing a variety of different combination. This evaluation will not only involve scrutinizing the model's effectiveness using various combinations, but will also delve into the analysis of the observed correlation between the positive rate of tweets and the team's performances.

4.1 Model Metrics

A key aspect of understanding and evaluating machine learning models is the assessment of their performance using various evaluation metrics. These metrics allow us to gauge the accuracy, reliability and robustness of the models, and they play an essential role in selected the most suitable model for the given task. In order to compute the aforementioned evaluation metrics, several fundamental measures are employed to establish the mathematical expression for each metric. These measures include True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

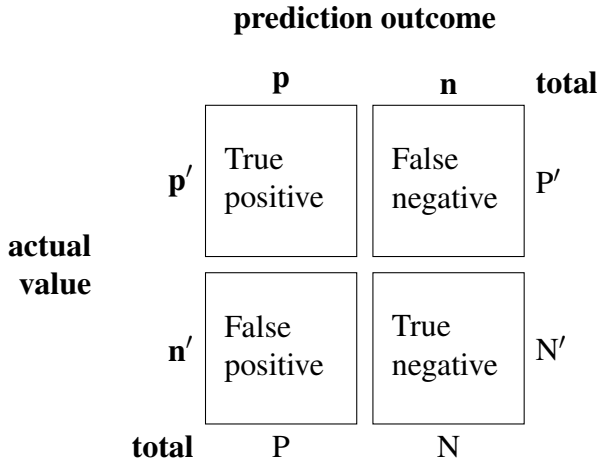

Figure 4.1: *Confusion Matrix Visualization*

The following evaluation metrics serve as an extensive and comprehensive illustration of the performance characteristics of a classifier, as described by Hossin [52].

- **Precision**: The ratio of true positive predictions to the total number of positive predictions made by the classifier. It reflects the model's ability to accurately identify positive instances.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.1}$$

- **Recall**: Recall is employed to quantify the proportion of positive instances that are accurately identified by the classifier.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.2}$$

- **F1-score**: The harmonic mean of precision and recall, which balances the trade-off between precision and recall. It is a commonly used metric to

summarize a model's performance.

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

- **Accuracy**: The ratio of correct predictions to the total number of instances. It is a widely used metric for evaluating the overall performance of a classifier.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{4.4}$$

4.2 Classifier Performance

This section delves into the detailed analysis and evaluation of different sentiment analysis models and techniques using the comprehensive football-related tweet data set curated by Malafosse.

This section aims to provide insightful answers to two fundamental research questions: RQ1: "How do lexicon-based, traditional machine learning, and deep learning sentiment analysis models differ in their accuracy when analyzing Malafosse's football related data set?" and RQ2: "How do different stop word strategies, such as NLTK stop words, customized stop words, and no stop words, influence the performance of machine learning and deep learning models in sentiment analysis football-related tasks?"

By addressing the research questions outlined above, the aim of this study is to provide valuable insights and recommendations for researchers and practitioners seeking to enhance sentiment analysis techniques in the domain of football sentiment analysis.

4.2.1 Lexicon-Based Results

The performance of VADER sentiment analysis model was assessed across three distinct scenarios, encompassing the removal of stop words, the utilization of custom stop words, and an analysis conducted without any stop words.

In the initial scenario, wherein stop words were removed from the data set consisting of football-related tweets, VADER exhibited the lowest level of performance. The precision scores pertaining to the negative sentiment category remained relatively consistent throughout all scenarios, registering values of 0.71, 0.7, and 0.72 for the removal of stop words, custom stop words, and the absence of stop words, respectively. The recall scores exhibited a lower enhancement as stop words were eliminated of 0.61, compared to the implementation of custom stop words of 0.65 and the exclusion of stop words of 0.66 for the negative sentiment category.

The positive sentiment category precision and recall score maintained a consistent level across all scenarios, lingering at a value of 0.54 and 0.86 to 0.87 respectively. However, since the model is showing high recall for the positive class, implies that this is highly effective in correctly retrieving positive instances from the data set. This interesting trade-off between precision and recall suggests that the model may be biased towards classifying tweets as positive.

Similarly, within the neutral sentiment category, the precision scores showcased minimal variance across the diverse scenarios, converging at a value of 0.67. Conversely, the recall scores revealed a marginal enhancement within the stop words removal scenario of 0.33 in contrast to the utilization of custom stop

words of 0.32 and the absence of stop words with 0.32.

In terms of accuracy, stop words removal technique recorded scores of 60.17%, 61.05% for custom stop words, and 61.60% for the absence of stop words. The level of accuracy attained in this study was found to be inferior to that demonstrated by Al-Shabi's [48], which managed to achieve an accuracy of 72%. However, it is important to highlight that the data set utilized in our research was substantially larger compared to the one employed in the aforementioned study.

The increased support exhibited by this particular model, as compared to the support for Naive Bayes and CNN-LSTM models, can be attributed to its utilization of a lexicon-based technique. This technique eliminates the need to split the data into training, validation and test subsets.

Table 4.1: *VADER Results*

	Precision	Recall	F1-Score	Support	Accuracy
No stop words Removed					
Negative	0.72	0.66	0.69	373854	
Positive	0.54	0.87	0.67	373854	
Neutral	0.67	0.32	0.43	373854	
Accuracy					61.60%
Custom stop words					
Negative	0.7	0.65	0.67	373854	
Positive	0.54	0.86	0.66	373854	
Neutral	0.67	0.32	0.43	373854	
Accuracy					61.05%
Stop words Removed					
Negative	0.71	0.61	0.66	373854	
Positive	0.53	0.87	0.66	373854	
Neutral	0.66	0.33	0.44	373854	
Accuracy					60.17%

The findings suggest that the removal of stop words from the football-related tweet data set exerted a marginal negative influence on VADER's performance,

eliciting slightly reduced improvements in recall scores within the negative and neutral sentiment categories. Meanwhile, the implementation of not removing stop words, though not significantly impacting precision and recall, yielded a marginally diminished accuracy when contrasted with alternative scenarios. Notably, the precision scores remained relatively stable throughout all scenarios.

The performance of the best predictive technique, in this example with no stop words removed, is shown visually in the confusion matrix below.

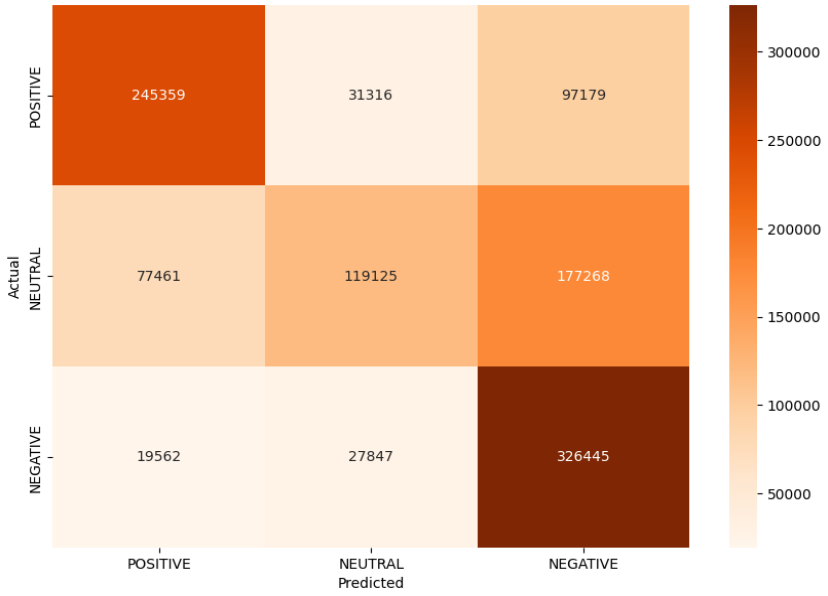

Figure 4.2: *VADER Confusion Matrix*

In conclusion, while the removal of stop words displayed some degree of negative impact on VADER's performance, the observed improvements were not of substantial magnitude. These outcomes imply that the choice of stop words might possess limited influence on VADER's sentiment classification within the football domain.

4.2.2 Naive Bayes Results and Discussion

The experiment entailed the application of the Naive Bayes model, specifically employing BoW and TF-IDF. Additionally, as previously mentioned, the same three scenarios of stop words scenarios was tested: without stop words, with stop words, and with customized stop words.

The Naive Bayes model performed relatively well in terms of TF-IDF technique. The best model achieved an accuracy 75.95% with customized stop words and the following hyperparameters: 'clf-alpha': 0.1, 'clf-fit-prior': False, 'tfidf-max-features': None, 'tfidf-ngram-range': (1, 2), 'tfidf-norm': 'l1', 'tfidf-use-idf': False.

A notable finding when working with Naive Bayes and TF-IDF is the influence of the 'ngram-range' hyper-parameter. The best-performing models were those with ngram ranges of (1,2) and (1,3), utilizing both unigrams, bigrams and trigrams. This result is consistent with the study made by Iqbal et al [32], where unigram with bi-gram features also performed better. The reason behind this could be that bigrams and trigrams capture more context compared to unigrams, which often leads to better model performance.

Table 4.2: *Naive Bayes with TF-IDF Results*

	Precision	Recall	F1-Score	Support	Accuracy
No stop words Removed					
Negative	0.72	0.89	0.80	53,292	
Positive	0.77	0.83	0.80	52,881	
Neutral	0.80	0.54	0.65	53,352	
Accuracy					75.42%
Custom stop words					
Negative	0.73	0.88	0.80	53,292	
Positive	0.79	0.81	0.80	52,881	
Neutral	0.76	0.59	0.67	53,352	
Accuracy					75.95%
Stop words Removed					
Negative	0.74	0.85	0.79	53,292	
Positive	0.77	0.81	0.79	52,881	
Neutral	0.75	0.59	0.66	53,352	
Accuracy					75.1%

On the other hand, the model's accuracy decreased with an ngram range of (2, 2) exclusively featuring bigrams, as observed from the worst-performing models. This outcome suggests that unigrams carry substantial information for sentiment classification in the given data sets and should not be omitted.

The influence of the 'clf-alpha' hyper-parameter, which controls the model's smoothing, was also prominent. Lower alpha values (0.1, 0.5) resulted in better performance than larger ones (1.0, 10.0), indicating that less smoothing was preferable in this data set. This could mean that the distributions of words in the training data are relatively representative of those in the test data, so less compensation for unseen words is required.

The performance of the best predictive model, in this example with custom-stop words, is shown visually in the confusion matrix below.

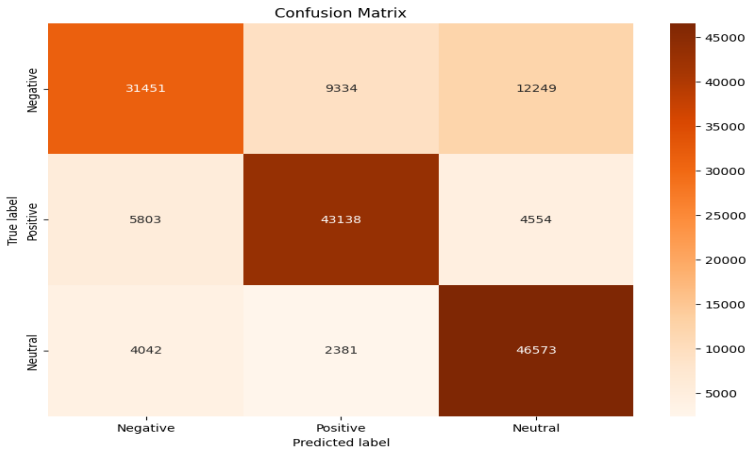

Figure 4.3: *Naive Bayes with TF-IDF Confusion Matrix*

In comparison to the study made by Akuma et al. [34], the employed model outperforms this study which reported an accuracy of 75.27% using TF-IDF and Naive Bayes. In contrast, the research conducted by Barnaghi et al. [33] demonstrated an accuracy of 66.21%, in which the model employed in this study exhibits a significantly improved accuracy. Venkatesh et al.'s model incorporates an external lexicon for sentiment analysis, adding an extra dimension to their feature set. While this approach can improve performance by utilizing predefined sentiment indications, it can also introduce bias based on the lexicon's composition, potentially affecting the model's ability to generalize.

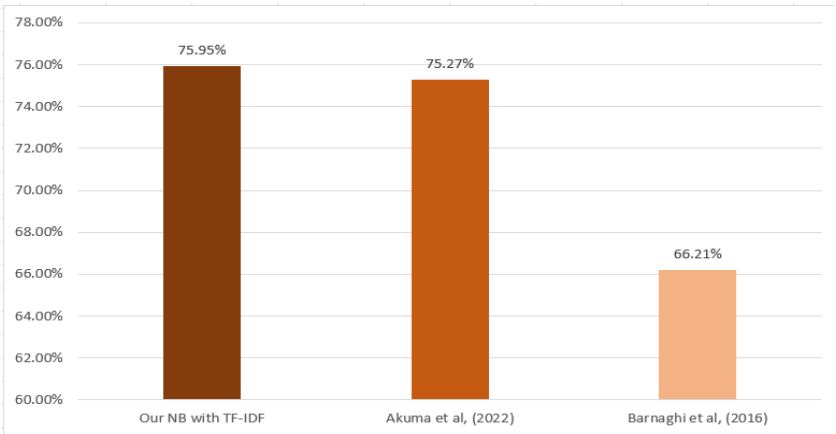

Figure 4.4: *Comparison of our Naive Bayes Model with TF-IDF study*

On the other hand, BoW approach achieved a top accuracy of 74.39% with no stop words and the following hyper-parameters: clf-alpha: 0.01, vect-max-features: 10000, vect-ngram-range: (1,1). While the approach performed worse than the TF-IDF, it still provided reasonable results. A likely explanation is that the BoW approach treats the document as an unordered collection of words and fails to capture the semantic relationship between words, while TF-IDF gives more importance to the words that are more unique to the document which may provide more context for sentiment analysis.

Table 4.3: *Naive Bayes with BoW Results*

	Precision	Recall	F1-Score	Support	Accuracy
No stop words Removed					
Negative	0.72	0.86	0.78	53499	
Positive	0.77	0.8	0.78	53142	
Neutral	0.75	0.57	0.65	52884	
Accuracy					74.39%
Custom stop words					
Negative	0.72	0.88	0.79	53499	
Positive	0.76	0.81	0.78	53142	
Neutral	0.76	0.54	0.63	52884	
Accuracy					74.35%
Stop words Removed					
Negative	0.72	0.87	0.79	53499	
Positive	0.75	0.81	0.78	53142	
Neutral	0.76	0.55	0.64	52884	
Accuracy					74.11%

Furthermore, the alterations in precision, recall and F1-score are indicative of the impact of using customized stop words, which may contain certain words that help in better classifying negative sentiments but contribute to a less accurate classification of neutral sentiments. The alterations might be due to the stop words capability of removing noise and focusing on significant words.

However, it is worth mentioning that the difference in results is relatively marginal, suggesting that while the use of different stop words does have an impact, it may not be as substantial as other aspects of the model, such as feature selection or the underlying algorithm. This is the following confusion matrix for the Naive Bayes using BoW for the best predictive model.

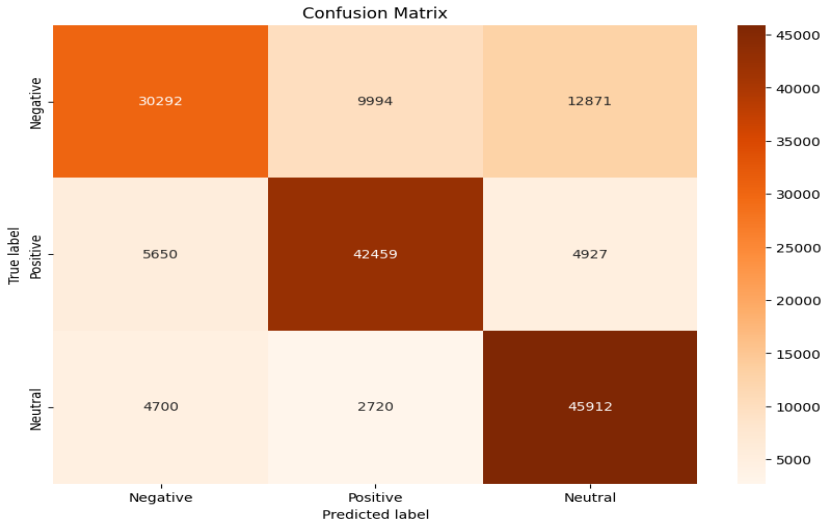

Figure 4.5: *Naive Bayes with BoW Confusion Matrix*

When comparing the results, the study represented by Iqbal et al [32] presented a superior accuracy of 89% when Unigram and Bi-gram features were used in conjunction with a bag-of-words implementation utilizing the Sentiment140 data set. While the architecture constructed is similar, since it is utilizing Naive Bayes and bag-of-words, the architectural divergence between the two studies emphasized the different ways of handling text data and extracting features for sentiment analysis.

Source	Technique	Accuracy
This study	NB with TF-IDF	75.95%
This study	NB with BoW	74.39%
Akuma et al, (2022)	NB with TF-IDF	75.27%
Barnaghi et al, (2016)	NB with TF-IDF	66.21%
Iqbal et al, (2018)	NB with BoW	89%

Table 4.4: *Comparison of results achieved by the Naive Bayes model with different studies.*

4.2.3 CNN-LSTM Results and Discussion

In this section, a comprehensive examination of the results obtained from implementing hybrid CNN-LSTM model. As mentioned in Chapter 3, separate models was built using different hyper-parameters, such as number of filters, pooling size, LSTM output dimension, and drop out rate with different stop words techniques.

The highest accuracy was recorded at 83.45%, when no stop words were removed from the data set. This model employed the following hyper-parameters: number of filters set to 256, pool size to 2, LSTM output dimension to 448, and a dropout rate of 0.2. Furthermore, it was discovered that optimal performance of the model was achieved with a number of filters at 192 or 256, pool size of 2, LSTM output dimensions ranging from 350-512, and a dropout rate ranging between 0.2 to 0.4.

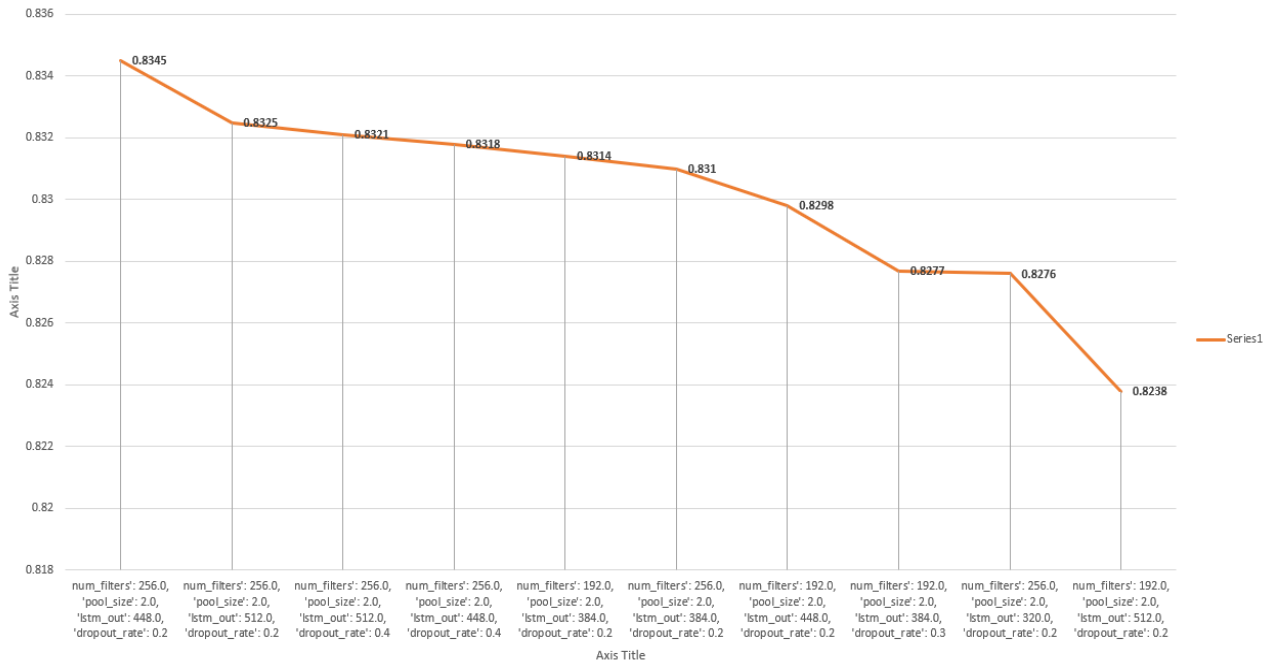

Figure 4.6: *LSTM Results with Different Parameters*

When stop words were introduced into the model, a decline in accuracy was noted. With customized stop words, the accuracy slightly reduced to 81.58% and further dipped to 80.61% when standard stop words were applied. The precision, recall, and F1-score across the three sentiment categories were also affected by the use of stop words. In the case where no stop words were used, better precision and recall were observed for negative and positive sentiments. Consequently, the F1-score, which is the harmonic mean of precision and recall, also saw an improvement for these categories. On the contrary, the inclusion of stop words led to slightly lower precision, recall, and F1-score for the negative and positive sentiments. The reason behind this could be that the inclusion of stop words may have introduced noise, leading to less accurate classification in these categories.

Table 4.5: *Hybrid CNN-LSTM Results*

	Precision	Recall	F1-Score	Support	Accuracy
No stop words Removed					
Negative	0.85	0.89	0.87	52830	
Positive	0.85	0.86	0.85	53172	
Neutral	0.81	0.75	0.77	53524	
Accuracy					83.45%
Custom stop words					
Negative	0.84	0.87	0.85	52830	
Positive	0.86	0.81	0.83	53172	
Neutral	0.75	0.76	0.76	53524	
Accuracy					81.58%
Stop words Removed					
Negative	0.81	0.86	0.84	52830	
Positive	0.84	0.8	0.82	53172	
Neutral	0.76	0.74	0.75	53524	
Accuracy					80.61%

The below confusion matrix provides a visual representation of the performance of the best predictive model, in this case with no-stop words removed. The x-axis represents the predicted classes (Negative, Positive, Neutral), while the y-axis represents the actual classes.

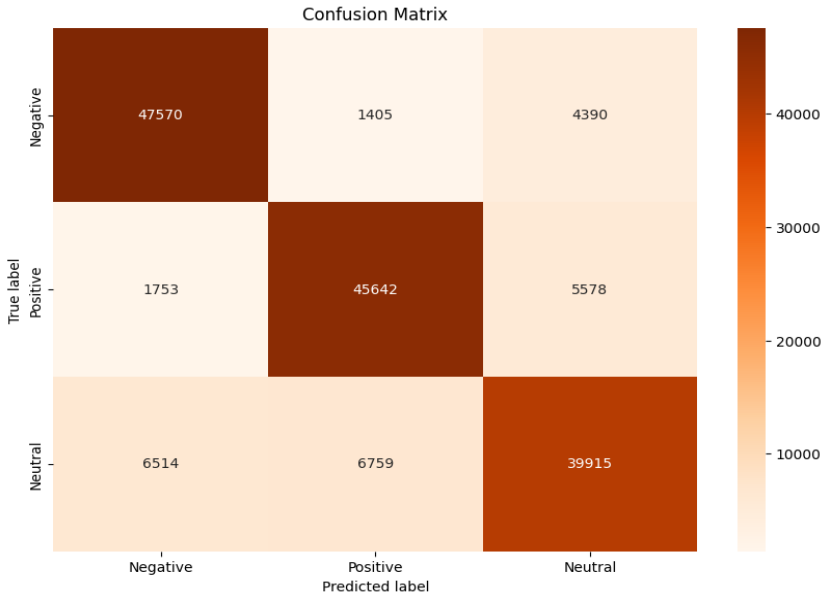

Figure 4.7: *LSTM Confusion Matrix*

The worst-performing parameters revealed a consistent pattern. It was observed that these models typically had higher pool sizes (4), lower LSTM output dimensions (ranging from 64 to 256), and lower number of filters (64 and 128). These models achieved validation accuracy ranging from 77.20% and 78%. The recurrent pattern suggests that high pool size, low LSTM output dimensions, and low number of filters are detrimental to the performance of our CNN-LSTM models.

Upon comparing the best model's performance with the study by Venkatesh et al. [47], it was found that the model achieved slightly lower accuracy which was 85%. However, considering the precision, recall and F1-score, the model demonstrated comparable or better performance.

Moreover, it's essential to mention that the support size in this study was significantly larger compared to the benchmark study. This means that our model was tested on a much larger and more diverse set of data, adding robustness and validity to our results. Therefore, even though our model's accuracy was slightly lower, the results are arguably more dependable due to the increased support size.

Table 4.6: *Comparison of the performance of our model with the study by Venkatesh et al, (2021)*

	Study by Venkatesh et al, 2021				Our Model			
Sentiment	Precision	Recall	F1 Score	Support	Precision	Recall	F1 Score	Support
Positive	0.90	0.89	0.89	560	0.85	0.86	0.85	52830
Negative	0.78	0.63	0.70	115	0.85	0.89	0.85	53172
Neutral	0.78	0.83	0.81	326	0.81	0.75	0.77	53524

4.2.4 Summary

In summary, to answer the first research question, while each model demonstrated the capacity to perform sentiment analysis on Malafosse's football-related tweet data set, the CNN-LSTM model outperformed the Naive Bayes with difference in accuracy of 8.18% and VADER models in terms of accuracy by around 21.85% which effectively demonstrated the potential advantages of utilizing deep learning models which are capable of recognizing intricate patterns and relationship within data.

These findings align with our hypothesis that the performance of different sentiment analysis models will significantly differ between lexicon-based, tradition machine learning, and deep learning models when applied to the tweet data set. The observed differences in model performance not only support our hypothesis but also underscore the significance of selecting the appropriate sentiment analysis models. Nonetheless, the results provide a measure of the performance associated with each degree of model complexity, thereby serving as a potential initial benchmark for other researchers with different model techniques in mind.

Furthermore, in order to answer the second research question, it was observed that the performance of VADER and Naive Bayes models, remained relatively stable, with little to no difference in accuracy when applying different stop words strategies. These models might be less sensitive to the use of stop words, perhaps due to their inherent algorithmic nature and their design principles.

However, a contrasting trend was noted with the deep learning model, which exhibited noticeable variations in performance when different stop word strategies

were employed. Specifically, a difference of up to 3% in accuracy was observed. A reason for this could be that in such models, the sequential nature and context of the words matter. When stop words are removed, an alteration to the sequence pattern could be done, which might affect these sequence-dependent models.

Therefore, the initial hypothesis can be considered partially confirmed. The impact of stop word strategies appears to be model-dependent, underlining the importance of considering the model type and task specificity when deciding upon a stop word strategy in sentiment analysis tasks, particularly in nuanced fields such as football-related sentiment analysis.

4.3 Probability Analysis

This section delves into the detailed analysis and evaluation of the correlation between the percentage of positive tweets and the team performance.

This section aims to provide insightful answers to another two fundamental research questions: RQ3: ”How does the positivity rate of tweets relate with match-winning probabilities, as assessed through Logistic Regression?” and RQ4: ”What is the correlation between the Positive Sentiment Score and the Points Per Game (PPG) achieved by teams, and how does this relationship evolve over the course of a football season?”

By addressing these research questions, a more valuable insights will be shed light on how the public opinion can be a good predictor for football matches.

4.3.1 Period I Results and Discussion

The analysis and discourse based on the findings obtained during the period from the 1st of December and 1st of January will be discussed. The focus shall be on the Premier League teams and their correlation between match outcomes and PPG within that particular month, along with their corresponding sentiment scores.

Throughout this period, a substantial quantity of 440,149 tweets were gathered, encompassing all teams. Notably, the most prominent team amongst them were Manchester United, using the hashtag #mufc. It is worth noting that this alignment coincides with their status as the most followed team on Twitter, as of May 2022, according to statistical data[1].

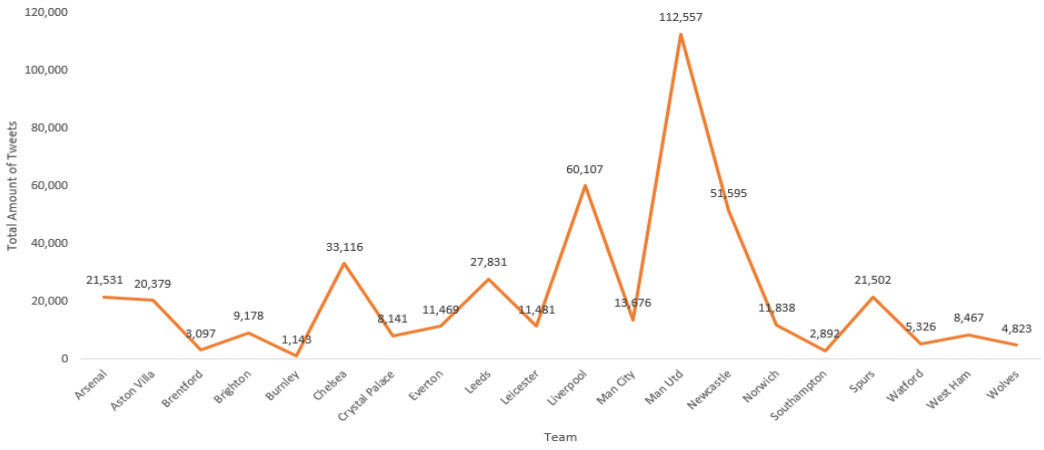

Figure 4.8: *Middle of Season Tweets*

Analysis of Match Win with Positive Sentiment

The relationship between match win and positive sentiment was done using Logistic Regression, where the dependent variable (win/did not win) is binary and the independent variable (positive sentiment) is continuous.

[1]https://www.statista.com/statistics/546991/number-of-twitter-followers-football-clubs-english-premier-league/: :text=The%20statistic%20depicts%20the%20number,followers%20at%20approximately%2031%20millio

The average predicted win probability for sentiment values greater than 60% was calculated to be 42.78%. Furthermore, a consistent increase in the win probability was observed as the score increased from 60% to 100% of the positive tweets.

More specifically, for every increase of 1% in the sentiment score starting from 60%, there was an increase of approximately 0.11% in the predicted win probability. This trend was relatively consistent throughout the range of sentiment scores analyzed.

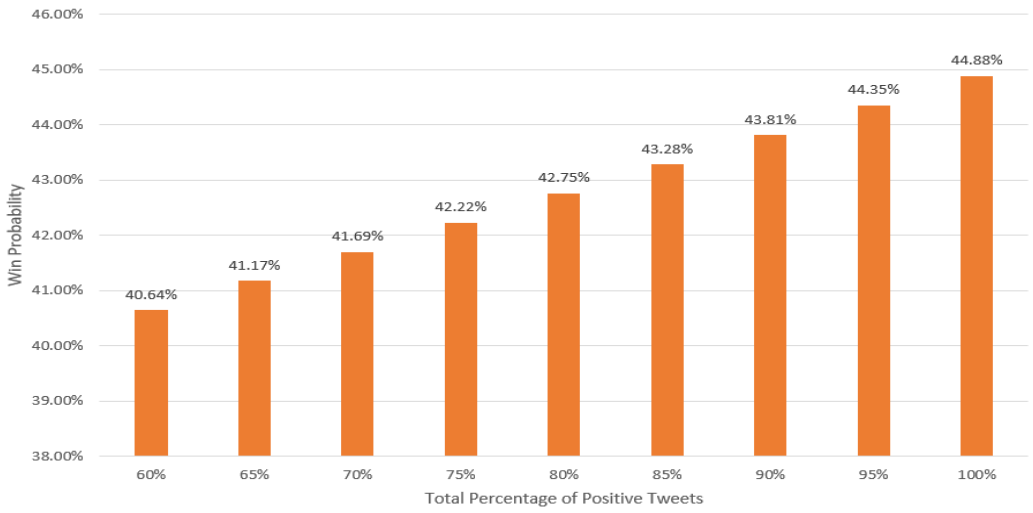

Figure 4.9: *Middle of Season Win Probability Trend*

This analyses demonstrated a positive relationship between sentiment scores and predicted win probability in the middle of the season. However, it is important to note that given the predicted win probability for sentiment values greater than 0.6 is only 42.78%, although sentiment score is a potential contributing factor to winning, it is likely not the only factor at play.

Analysis of PPG with Positive Sentiment

The relationship between PPG during the month of December and the positive sentiment percentage of tweets was also investigated. This was also done using Logistic Regression, and it was measured by the likelihood of each category when the positive sentiment percentage ranges between 60% and 100%.

In the analysis for this month, it's noteworthy to consider the quartile results for PPG. The distribution of scores across the low, medium and high categories is defined by the specific quartile ranges as follows:

1. Very Low PPG: 0.00 to 0.95
2. Low PPG: 0.96 to 1.37
3. Medium PPG: 1.38 to 1.63
4. High PPG: 1.63 to 3.00

The detailed PPG scores of the individual teams are as follows. The positive sentiment score means the total percentage of positive tweets as inserted in a decimal value in Logistic Regression model:

Team	Points Per Game (PPG)	Positive Sentiment Score
Arsenal	2.40	0.51
Aston Villa	1.50	0.6
Brentford	1.00	0.58
Brighton	1.25	0.54
Burnley	0.33	0.48
Chelsea	1.50	0.46
Crystal Palace	1.40	0.6
Everton	1.33	0.43
Leeds Utd	0.25	0.45
Leicester City	1.50	0.45
Liverpool	2.00	0.59
Manchester City	3.00	0.57
Manchester Utd	2.50	0.53
Newcastle Utd	0.80	0.49
Norwich City	0.00	0.37
Southampton	1.20	0.63
Tottenham	2.00	0.52
Watford	0.00	0.39
West Ham Utd	1.40	0.23
Wolverhampton	1.00	0.52

Table 4.7: *Middle of Season PPG*

During this investigation, it was noted that as the percentage of positive sentiment increases, the probability of the team achieving a Very Low PPG decreases gradually. For instance, when the positive sentiment is at 0.6, the probability for Very Low PPG is 24.25%. However, when the positive sentiment increases to 1.00, the probability for Low PPG decreases to 21.44%. A similar trend was found for the Medium PPG , with a decreasing trend was observed with the probability going from 24.64% at 0.55 positive sentiment to 23.17% at 1.00 positive sentiment.

On the other hand, an upward trend was noted in both High PPG and Low PPG. When the positive sentiment is 0.6, the probability of High PPG is 25.57%, which increases to 27.83% at 1.00 positive. A comparable trend was found for

Low PPG, as the probability increases from 25.53%, to 27.55% when the positive percentage increases from 0.6 to 1.0.

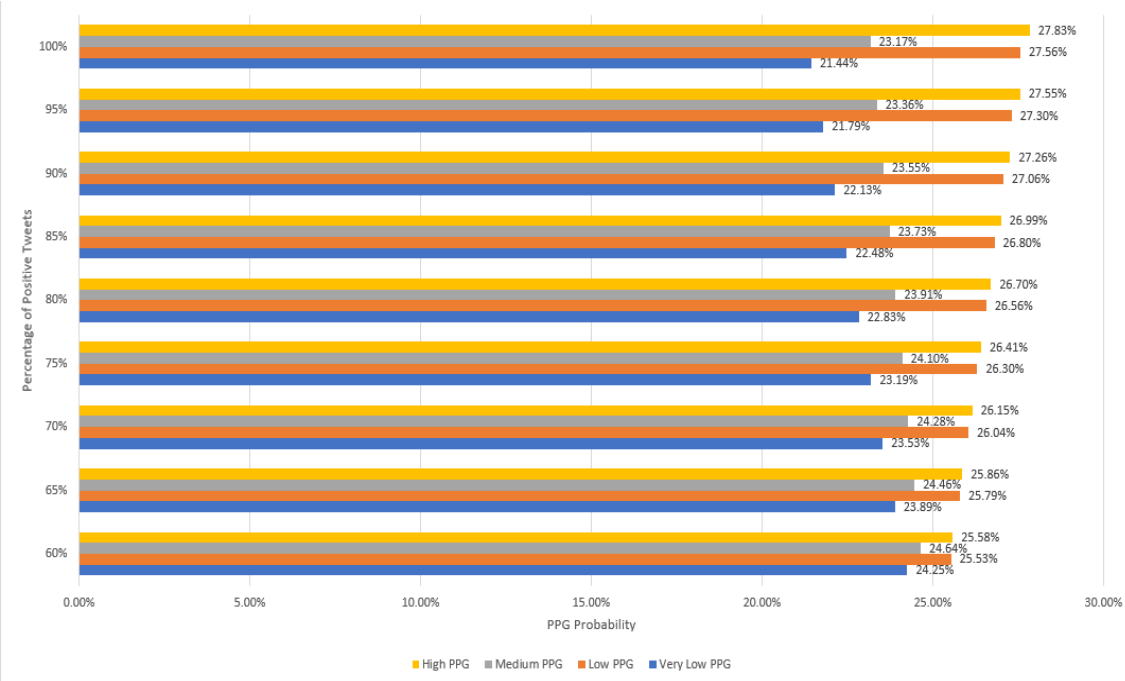

Figure 4.10: *Middle of Season PPG Trend*

The average probabilities were calculates across all sentiment scores, in which it was found that the average probability of Very Low PPG when positive sentiment percentage was greater than 60% was 22.84, Low PPG was approximately 26.54%, Medium PPG was 23.91% and High PPG was approximately 26.7%. These averages provide further insight into the overall trend where the higher average probability for High PPG with 26.5% as compared to Very Low PPG with 23% supports our finding of a positive relationship between sentiment scores and Points Per Game.

4.3.2 Period II Results and Discussion

In this section, a similar examination and discussion will be examined but the centered around the discoveries acquired between April 1st and May 1st.

During this time frame, a total of 489,909 tweets were collected, surpassing the number of tweets obtained during the mid-season period. Furthermore, it is worth noting that Manchester United emerged again as the team generating the highest volume of tweets.

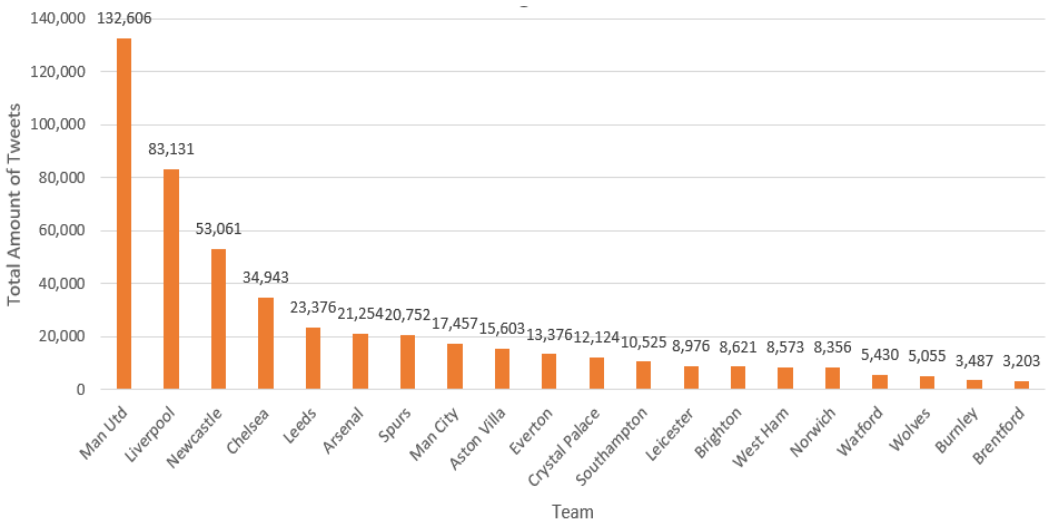

Figure 4.11: *End of Season Tweets*

Analysis of Match Win with Positive Sentiment

In this case, the average predicted win probability for sentiment values greater than 0.6 is 43.36%. As the data sentiment score between 0.6 to 1.0 is examined, it was also noted that the corresponding predicted win probability gradually rises. This begins at a sentiment score of 0.6 with a predicted win probability of 40.86% and ascends to a sentiment score of 1.0, where the win probability peaks at 46.87%.

This upward trend suggests that the higher the sentiment score, the higher the predicted win probability. It reflects a linear relationship between sentiment score and win probability under these conditions, indicating that every increase in sentiment score is associated with a proportional increase in the chance of winning.

In a comparison to the study made by Beal et al [9], which considered sentiment derived from sports journalists and statistical data, this study findings also demonstrated a similar trend of increased predictive accuracy as the season progressed, despite the difference in methodologies. Starting with a sentiment score of 0.6, the predicted win probability at the end of the season is 40.86% compared to 40.64% at the middle of the season. This disparity persists across the sentiment scale, reaching a peak at a sentiment score of 1.0 where in the end of the season predicts a win probability of 46.87% while in the middle of the season, it predicts a slightly lower figure of 44.88%.

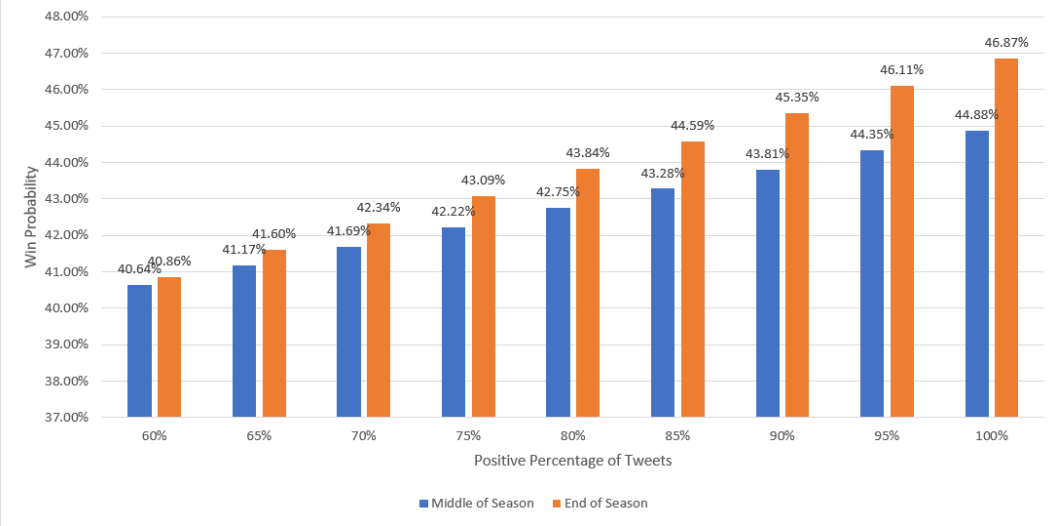

Figure 4.12: *End of Season Win Probability Trend*

Analysis of PPG with Positive Sentiment

In this scenario, the PPG with the percentage of positive sentiment will be investigated between the month of April. In the distribution of the quartile results, there was a slight variation between the ranges of the PPG classifications.

1. Very Low PPG: 0 - 0.95
2. Low PPG: 0.95 - 1.33
3. Medium PPG: 1.34 - 1.99
4. High PPG: 2 - 2.5

The details of the PPG for every individual teams are as follows:

Team	Points Per Game (PPG)	Positive Sentiment Score
Arsenal	1.20	0.55
Aston Villa	1.33	0.48
Brentford	2.33	0.81
Brighton	2.00	0.64
Burnley	2.17	0.49
Chelsea	1.75	0.55
Crystal Palace	1.40	0.68
Everton	0.8	0.46
Leeds Utd	1.33	0.53
Leicester City	1.25	0.59
Liverpool	2.5	0.71
Manchester City	2.5	0.6
Manchester Utd	1.33	0.32
Newcastle Utd	2.00	0.66
Norwich City	0.75	0.5
Southampton	0.8	0.49
Tottenham	1.75	0.55
Watford	0.00	0.47
West Ham Utd	1.00	0.62
Wolverhampton	0.00	0.54

Table 4.8: *End of Season PPG*

In comparing the data from the middle of the season to the end of the season for different teams, drastic changes were observed for some teams.

Brentford showed a dramatic increase in both PPG and positive sentiment score from the middle of the season. Their PPG increased from 1.00 to 2.33, a jump of 133%. Concurrently, their positive sentiment score demonstrated an enhancement from 58% to 81%, a positive deviation amounting to 40%. A similar trend was noted with Newcastle where a dramatic increase in PPG from 0.8 to 2.00 by the end of the season, a 150% increase. The positive sentiment score also saw a significant increase from 49% to 66%, amounting to 35% increase, which indicates a strong positive relationship between public sentiment and team performance.

Conversely, certain teams such as Manchester United were observed to undergo a severe decline in their performance scores. The PPG index for Manchester United dropped significantly from 2.50 mid-season to 1.33 at the end of the season, translating to a decrease of 47%. Along similar lines, their Positive Sentiment Score also depreciated from 53% to 32%, marking a 40% decrease of the total percentage of positive tweets.

Furthermore, a noticeable trend was found when evaluating the PPG categories and positive sentiment score where the probability of achieving a High PPG appears to increase with the proportion of positive sentiment in the tweets. The Medium PPG also achieved a higher accuracy with positive sentiment percentage, however this increase was very marginal. At a positive sentiment of 60%, the probabilities for Very Low, Low, Medium, and High PPG were roughly 25%,

30%, 15%, and 31% respectively. However, as the positive sentiment increased to 1.00, the probabilities altered to approximately 21%, 26%, 15%, and 37% for Very Low, Low, Medium, and High PPG respectively.

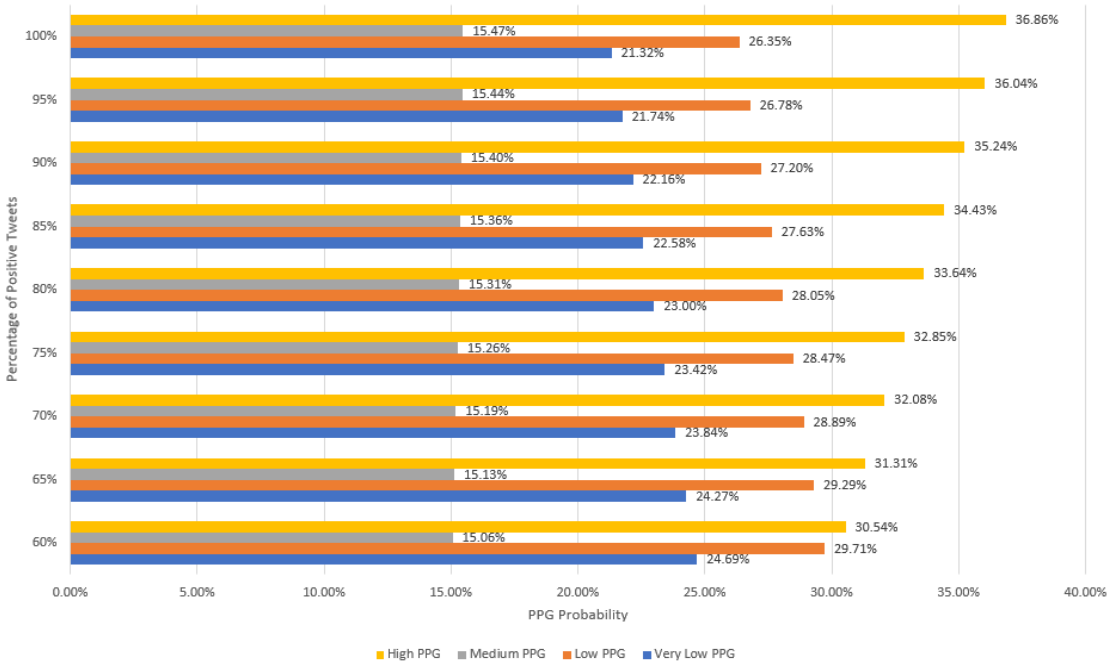

Figure 4.13: *End of Season PPG Trend*

The trends observed in the end-of-season data suggests that higher positive sentiment is associated with an increased likelihood of achieving a High PPG and a decreased probability of a Very Low PPG. Although these results suggest a positive correlation, it is important to take into consideration the substantial percentage of a Low PPG probability. In contrast, the mid-season results reveal a more balanced relationship. These differences between the mid-season and end-of-season could suggest that the public sentiment becomes a better predictor as the season progresses.

4.3.3 Summary

Based on the analyzed data, the answer to the third research question is that there is a positive correlation between the positive rate of pre-match tweets and the probability of a team winning a match, as assessed through logistic regression.

This correlation was observed both in the middle and at the end of the season since the higher the positive percentage, the higher the chances of winning increased with starting from 40.64% and 40.86% when the total positive sentiment of the total tweets is 60% and finishing with 44.88% and 46.87% respectively when the positive sentiment percentage is 100%. Therefore, the hypothesis mentioned in Chapter 1 is supported.

Although there is a correlation, it should be noted that sentiment scores are not the only factor of a team's winning probability. While sentiment scores contribute to the predicted win probability, other factors are also likely to play a part given that the average predicted win probabilities is around 42%-43%.

In response to the fourth research question, the findings reveal a nuanced picture. During the middle of the season, an increase in positive sentiment scores correlated with an increase in High PPG and Low PPG, while probabilities for Very Low and Medium PPG displayed a decreasing trend. However, the relationship between positive sentiment and team performance seemed to strengthen as the season progressed.

By the end of the season, a clearer trend is seen as an increase in positive sentiment scores was associated with higher probabilities of achieving Medium or

High PPG, and lowered the probabilities of achieving Very Low or Low PPG, which could suggests that positive sentiment could be more accurate predictor of team performance later in the season. However, it was noted the persistent effect of a relatively high Low PPG probability throughout the season, even as the positive scores increased. This consistent high Low PPG indicates that positive sentiment does not necessarily alleviate the risk of Low PPG, but increases the chance of High PPG.

Upon conducting an individualized examination of the teams, the correlation was not universally applicable. In fact, for certain teams, such as Brentford, Manchester United, and Newcastle United, major shifts in sentiment score appeared to correspond with significant changes in performance. However, this trend was not consistent across all teams.

Therefore, it was concluded that the hypothesis that the correlation between positive sentiment score and team performance strengthens as the season progresses and increased positive sentiment generally correspond to higher PPG is partially supported. This is because, since for certain teams, substantial changes in sentiment score did indeed align with significant changes in performance, this wasn't a universal pattern. However, it was noted that as the season progresses, an increase positive sentiment generally corresponding to higher PPG during the month. Therefore, while this trend supports the hypothesis, the exceptions seen when examining the teams individually suggest that other factors might be impacting team performance.

Chapter 5: Conclusions and Recommendations

The study embarked on a comprehensive exploration of sentiment analysis models in the context of football-related text. The central goal was to evaluate and compare different models, varying in complexity and stop words techniques, to determine the most appropriate solution for the given problem. The study also aimed to provide valuable guidelines regarding each model's performance when applied to football-related sentiment analysis.

In response to this, several models, including VADER, Naive Bayes using both TF-IDF and BoW, and a hybrid CNN-LSTM, were investigated and compared. These findings indicate that while traditional machine models and lexicon-based approaches yielded commendable results, neural network models, particularly CNN-LSTM, showed superior performance.

The study also examined the impact of different stop word strategies on these models. These results suggest that while lexicon-based and traditional machine learning models were largely stable when different stop word strategies were implemented, the CNN-LSTM model's performance varied, with accuracy fluctuating by up to 3%.

The study further analyzed the correlation between positive tweets and match outcomes. Tweets collected over a month-long period demonstrated that an increased percentage of pre-match tweets with positive sentiment correlated with increased chances of winning of around 44.88% during the middle of the season,

and 46.87% during the end of the season. Moreover, when comparing PPG over a period of a month, positive sentiment appeared to be a more accurate predictor of team performance towards the end of the season.

Nevertheless, this correlation was not consistent across all teams. Notably, teams like Brentford, Manchester United, and Newcastle United showed a significant relationship between sentiment scores and performance changes. This inconsistency suggests that while sentiment analysis can be a useful tool in football team performance over a period of a month, it should be complemented with other predictive factors to enhance accuracy.

In summary, this study provides a basis for future research in this area, particularly regarding the effectiveness of different sentiment analysis models and the influence of sentiment on match outcomes. However, the findings also highlight the complexity of this field, and suggest the need for further exploration and understanding of these relationships.

5.1 Limitations

Notwithstanding the valuable insights derived from the aforementioned study, it is important to acknowledge the potential limitation posed by the presence of class imbalance within the data set. The uneven distribution of classes has the potential to introduce bias in the models, favoring the majority class and consequently impacting the overall performance. In order to tackle this concern, a down-sampling technique was employed to attain a balanced representation of the categories, thereby removing the potential bias resulting from class imbalance. Fu-

ture research could explore the effect of different sampling techniques to handle class imbalance could be examined.

In addition to this, the sudden block of Twitter APIs posed another limitation to the study. This block prevented the collection of data from the beginning of the season, which could have potentially shown information critical to our analysis and thus may have influenced the results of the study.

5.2 Future Improvements

Considering the findings of the current study and the limitations identified, several improvements can be proposed for future research. Firstly, the results of the study indicated that neural networks were the most effective technique for our analysis. Thus, future studies could build on this finding and implement a wider range of neural network models. This could further improve the accuracy of the predictions and offer more robust analysis.

In the rapidly, evolving field of AI, newer models like ChatGPT show immense potential. Therefore, it would be intriguing to compare the performance of the newer models with the ones currently used for sentiment analysis even for a small range of the data set, since at the moment ChatGPT doesn't allow for files to be uploaded. This comparative analysis could shed light on whether these models offer any improvements in predicting sentiment from text data.

Furthermore, an interesting area to explore in the future would be the examination of in-play-tweets. Evaluating whether these tweets have any influence on match win probabilities could add a new dimension to understanding of real-time

sentiment's impact on outcome. Future research could test this hypothesis, which would contribute to a richer and more nuanced understanding of the interplay between social media sentiment and match outcomes.

Lastly, expanding the data sources could also add significant value. The current study analyzed Twitter data, but considering the popularity and wide usage of Facebook, incorporating comments from this platform could offer additional valuable insights.

List of References

[1] 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.

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

[3] 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.

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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

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

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

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

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

[21] 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.

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

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

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

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

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

[31] 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.

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

[33] 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.

[34] Akuma Stephen, Tyosar Lubem, and Isaac Adom. Comparing bag of words and tf-idf with different models for hate speech detection from live tweets. *International Journal of Information Technology*, 14, 09 2022.

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

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

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

[38] 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.

[39] 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.

[40] 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.

[41] 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.

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

[43] 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.

[44] 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.

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

[46] Anwar Ur Rehman, Ahmad Malik, Basit Raza, and Waqar Ali. A hybrid cnn-lstm model for improving accuracy of movie reviews sentiment analysis. *Multimedia Tools and Applications*, 78, 09 2019.

[47] Venkatesh, Siddhanth U Hegde, Zaiba A S, and Nagaraju Y. Hybrid cnn-lstm model with glove word vector for sentiment analysis on football specific tweets. In *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–8, 2021.

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

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

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

[51] 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.

[52] Mohammad Hossin and Sulaiman M.N. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process*, 5:01–11, 03 2015.