Sentiment Analysis in Social Networks during the 2021 and 2022 Formula 1 Seasons: A Study Using Natural Language Processing on Twitter

Matheus Mileski
matheuslopes@alunos.utfpr.edu.br
Federal University of Technology – Paraná (UTFPR)
Apucarana, Paraná, Brazil

Daniel Prado Campos
danielcampos@utfpr.edu.br
Federal University of Technology – Paraná (UTFPR)
Apucarana, Paraná, Brazil

Luiz Fernando Carvalho
luizfcarvalho@utfpr.edu.br
Federal University of Technology – Paraná (UTFPR)
Apucarana, Paraná, Brazil

Rafael Gomes Mantovani
rafaelmantovani@utfpr.edu.br
Federal University of Technology – Paraná (UTFPR)
Apucarana, Paraná, Brazil

ABSTRACT

Sentiment Analysis is an emerging research area that focuses on extracting semantic and emotional inferences from natural language, paving the way for analyses that deal with a high volume of textual data. The growing importance of data in strategic decision-making and the recognition of social networks as vast repositories of public opinion have propelled this study, which aimed to explore the interaction between human emotions and motorsport events. Thus, this study focused on applying Natural Language Processing to extract and analyze sentiments expressed in tweets about Formula 1. Advanced machine learning and deep learning techniques were employed to train various models in the sentiment classification task. Among these, Logistic Regression and LSTMs stood out, achieving accuracies of 78.21% and 78.08%, respectively. The LSTM model, in particular, was implemented on a public dataset of tweets collected during the 2021 and 2022 Formula 1 seasons. The model was used to classify the sentiments expressed by fans, allowing for an exploratory analysis of data correlated to specific events of the races. The findings revealed significant engagement patterns, with notable spikes in emotional reactions coinciding with critical moments of the seasons. These discoveries illustrate how particular events can profoundly influence the emotions and behavior of fans. From a detailed analysis of expressed sentiments, valuable data can be obtained that may be leveraged for developing more effective marketing and communication strategies in the sport.

KEYWORDS

data mining, machine learning, public opinion, linguistic analysis

1 INTRODUCTION

In the past decade, Machine Learning (ML) has continuously reshaped our understanding and interaction with data [1]. Deep Learning (DL), a subset of ML characterized by family of neural networks [2] that has significantly enhanced our capabilities in developing robust and complex applications. Its integration with Natural Language Processing (NLP) has been a pivotal advancement in how we process and interpret human language, marking a revolutionary step in the realm of computational linguistics and Artificial Intelligence (AI) [3].

Among the most consolidated applications of the combination of ML and NLP are Opinion Mining and Sentiment Analysis [4]. This area of study focuses on extracting semantic and emotional inferences from natural language, often without a deep understanding of the text, paving the way for analyses that deal with the substantial volume of data generated in social networks [5]. Social networks, in particular, have become rich real-time data sources where users share opinions, emotions, and insights on various topics. Analyzing these sentiments provides a valuable understanding of the public's opinions, which can be applied in various areas, including marketing, politics, entertainment, and sport [6].

In this context, this study aims to perform a sentiment analysis of social media users, specifically on Twitter, during the 2021 and 2022 Formula 1 seasons to identify engagement patterns and feelings and understand the motivations behind these emotions. These predictions can, firstly, offer a new lens of analysis on the public's perception of teams, drivers, and races, identifying patterns and trends that may be strategic for those involved in the competition; and, secondly, contribute to the understanding of the relationship between public emotions, the performance of teams and drivers, and the impact of the organizers' strategic decisions.

This paper is organized as follows: Section 2 presents some of the necessary concepts of Sentimental Analysis and Pattern Recognition and discusses some recent studies combining these two research areas. The experimental methodology is presented in Section 3. The results are discussed in Section 4, while the conclusions and final considerations of the study are presented in Section 5.

2 RELATED WORKS

Sentiment analysis on Twitter has been a prominent research area, with key studies employing various methodologies and machine learning techniques. Go et al. [7] and Pak et al. [8] were early contributors, focusing on automatic sentiment classification using distant supervision and linguistic analysis of tweets, respectively. Both studies utilized Naive Bayes classifiers and emphasized the importance of preprocessing and feature extraction techniques such as unigrams, bigrams, and trigrams.

Agarwal et al. [9] and Wang et al. [10] furthered this research by incorporating tweet characteristics and hashtag analysis into sentiment classification using Support Vector Machines (SVM). These

studies highlighted the significance of integrating elements like hashtags, URLs, and mentions, and the relevance of classifying sentiments at the hashtag level.

In dataset analysis, Saif et al. [11] and Koto and Adriani [12] provided insights into the performance correlations between dataset characteristics and sentiment classification. They explored various textual attributes, including opinion and sentiment lexicons such as AFINN and Senti-Strength, and found that some widely-known lexicons like SentiWordNet were less effective for Twitter sentiment analysis.

Advancements in deep learning were marked by the studies of Sosa [13] and Goularas and Kamis [14], which explored hybrid neural network models such as CNN-LSTM and LSTM-CNN. These studies demonstrated the superiority of hybrid and combined approaches over single models in sentiment classification.

Recent contributions, such as those by Swathi et al. [15], have broadened the application of Twitter sentiment analysis to include domains like stock price forecasting. In their study, Teaching-Learning-Based Optimization (TLBO) is integrated with Long Short-Term Memory (LSTM) networks to analyze sentiments on Twitter for predicting stock prices. Additionally, Saranya and Usha [16] and Aslan et al. [17] have employed advanced machine learning techniques such as Random Forest and Convolutional Neural Networks (CNN) combined with Arithmetic Optimization Algorithm (AOA). These methodologies have been applied to multi-class sentiment classification, yielding high accuracies in their respective studies.

Overall, these studies collectively contribute to the understanding of sentiment analysis on Twitter. However, there is a notable gap in the literature regarding applying these techniques in sports data analysis, particularly for sports audiences such as Formula 1. This gap presents a potential area for future research, exploring the intersection of sentiment analysis, text data mining, and sports data analysis.

3 EXPERIMENTAL METHODOLOGY

This section presents the experimental methodology employed in the experiments with traditional and DL algorithms. Figure 1 depicts the whole process, including its sub-steps. The following subsections will explain each step in detail.

3.1 Datasets

This study explored two different public textual datasets: the first, entitled `Sentiment140`, is currently available on Kaggle[1]. This dataset, developed originally by [18] is composed by approximately 1.6 million tweets and created with an automated label generation approach, assuming that tweets containing positive emoticons, such as ':)', were positive, and those with negative emoticons, such as ':(', were negative.

Table 1 displays all the features available in `Sentiment140` dataset. The original dataset contains a third and neutral category but with few instances. Thus, we removed these examples, keeping only the *tweets* classified as negative (800k examples) or positive (800k examples). Only two features were used to train ML and DL models: the *target* (polarity of the tweet) and the *text* (content). This choice aims to focus exclusively on the textual content of *tweets* and their sentimental polarity, essential for the training and validation of the proposed models.

Table 1: Sentiment140 dataset's features

Feature	Description
target	Polarity of *tweet*, with labels 0 for negative, 2 for neutral, and 4 for positive
id	tweet identifier
date	tweet date
flag	a search term to find the *tweet*. If there is no term, the value is set with *"NO_QUERY"*
user	username
text	text from tweet

Table 2: F1 dataset's features

Feature	Description
user_name	Username as defined by the user
user_location	User defined location for profile
user_description	User-defined description for your account
user_created	Date and time the user account was created
user_followers	Number of user followers
user_friends	Number of user friends
user_favorites	Number of user favorites
user_verified	Whether the account is verified or not
date	UTC date and time of posting the tweet
text	Text from tweet
hashtags	Additional hashtags in tweet besides #f1
source	Tool used to post tweet
is_retweet	Indicates whether the tweet was reposted by the authenticated user

In addition, a second and unlabeled dataset was used as a test set. This *dataset* entitled "`Formula 1 (F1) trending tweets`" [2] covers tweets posted with the *hashtag* "#f1" in the period from July 24, 2021, on August 20, 2022. It has a total of 628,360 tweets, but unlike `Sentiment140` dataset, it does not present a specific methodology used to collect the data. Table 2 displays the features available in the `F1` dataset. Again, only the *text* and *date* features were used for this study. The *text* column is essential for applying sentiment analysis models, while the *date* column allows one to perform an exploratory analysis of sentiments about F1 events.

3.2 Data Preprocessing

Datasets were meticulously preprocessed before the learning phase. The whole process included:

- converting all text to lowercase, facilitating uniformity and reducing variability in the data;
- replacing abbreviations in English using a specific dictionary with more than 350 translations, ensuring clarity and understanding of the text;

[1] https://www.kaggle.com/datasets/kazanova/sentiment140?select=training.1600000.processed.noemoticon.csv

[2] https://www.kaggle.com/datasets/kaushiksuresh147/formula-1-trending-tweets?select=F1_tweets.csv

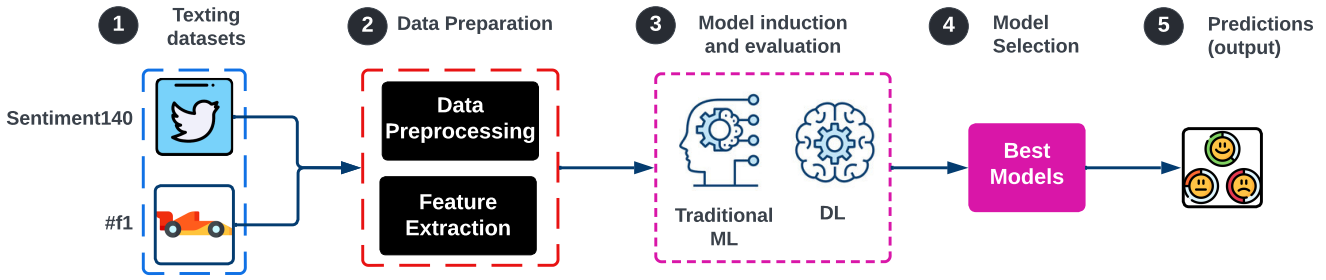


Figure 1: Methodology for Twitter Sentiment Analysis about Formula 1.

- translating *emojis* (modern versions of *emoticons*), a common way of representing emotions on social media, into their corresponding textual representations[3], which allowed the analysis of these elements as part of the textual content;
- extracting URLs present in *tweets*, replacing them with detailed information about the domain, path, and parameters. This was crucial to remove irrelevant information for sentiment analysis and focus on the textual content;
- similarly, mentions and *hashtags* were extracted and replaced with clear textual representations, employing regular expressions to identify and replace these elements with keywords such as *"mention"* and *"hashtag"*, followed by their respective usernames and topics;
- removing punctuation and repeated spaces to clean the text and avoid interference with the analysis. All the punctuation characters were removed, except of periods and commas between numbers.

After these cleaning steps, the texts were subjected to tokenization and lemmatization. Tokenization splits the text into individual words, while lemmatization converts words to their base forms, or *lemmas*. Words considered as *"stop words"* were removed to reduce noise and focus on meaningful words for analysis [19].

3.3 Feature Extraction

While DL models can naturally work with sequential data such as texts, since it has the ability to learn features from data, traditional ML algorithms typically require additional steps, like feature extraction, to effectively handle such data. Thus, we performed a feature extraction step exploring some different techniques. The first was the Term Frequency-Inverse Document Frequency (TF-IDF) technique, which extracts n-grams of the texts (unigrams, bigrams, trigrams). Alternatively, a different feature set was obtained by Word2Vec's "Google News *negative* 300" pre-trained model, a widely recognized technique for generating vector representations of words that capture semantic and syntactic relationships. Lastly, an important feature was the *score* of text polarity, obtained through `TextBlob`. It is a library that provides simple but compelling sentiment analysis and can return the polarity and subjectivity of a sentence. The polarity lies between [-1,1], where -1 defines a negative and 1 defines a positive sentiment. Negative words reverse the polarity. The library has semantic labels that help with fine-grained analysis, for example, emoticons, exclamation marks, emojis, etc.

All these extracted and processed features were combined into a single sparse matrix, which allows an efficient and compact representation of data, given the high dimensionality and sparse nature of textual data. The resulting matrices and relevant information were saved in serialized files with *pickle* Python module. These files were later used in the model training step, facilitating efficient access to the data and characteristics necessary for the model learning process.

3.4 ML and DL Algorithms

Several algorithms were carefully selected and trained in the model induction and evaluation step. The Logistic Regression (LR) algorithm was chosen to test the hypothesis that feelings expressed in texts have linear characteristics. Other algorithms established in the literature and included in experiments were: Naïve Gaussian Bayes (NB), Support Vector Machines (SVMs), Random Forest (RF), and Multilayer Perceptron (MLP).

NB was selected because it is a well-known baseline for textual classification tasks, especially in high-dimensional data sets. SVM was chosen due to its robustness and effectiveness in finding the best class separation margin. RF, in turn, is an *ensemble* method that uses multiple Decision Trees (DTs) to increase precision and avoid overfitting, being accurate for a wide range of problems. Finally, MLP is a simple architecture of neural networks selected for its ability to capture complex relationships in data.

Alternatively, Deep Learning (DL) models were also explored in the experiments, specifically Convolution Neural (CNNs) and Long Short-Term Memory (LSTM) networks. The DL architectures are shown in Table 3. They were carefully designed to capture spatial patterns in textual data. Both architectures (CNN and LSTM) start with an input layer, which receives the sequence of *tokens* from the text. Then, a *embedding* layer is used to transform these *tokens* into dense vectors, a richer and more informative representation. For CNN, there is subsequently a 1D convolutional layer applied to extract sequential patterns from the text. This layer is followed by a *max pooling* layer, which plays a crucial role in reducing the dimensionality of the data, and a flattening layer, which prepares the data for the classification phase. The model integrates dense layers that process Word2Vec representations and sentiment polarity. This

[3]For example, *emoji* :) is converted for the text *smiley face*.

Table 3: CNN and LSTM architectures for textual data analysis.

Model	Layer	Parameters	Activation	Description
CNN	Input			Receives sequence of tokens from text
	Embedding	128		Transforms tokens into dense vectors
	1D Convolutional	128 x 5	ReLU	Extracts sequential patterns from text
	Max Pooling	5		Reduces dimensionality of data
	Flattening			Prepares data for classification
	Dense	64	ReLU	Processes Word2Vec representations and sentiment polarity
	Dense	64	ReLU	Processes Word2Vec representations and sentiment polarity
	Dense	1	Sigmoid	Combines features for final classification
LSTM	Input			Receives sequence of tokens from text
	Embedding	128		Transforms tokens into dense vectors
	LSTM/Dropout	128, 0.2		Captures long-term dependencies includes dropout for regularization
	Dense	64	ReLU	Processes Word2Vec representations and sentiment polarity
	Dense	1	Sigmoid	Uses sigmoid activation for binary classification

information is combined with features extracted from the text to make the final classification decision.

The LSTM topology takes advantage of the model's ability to capture long-term dependencies within textual sequences. Similarly to the CNN architecture, the LSTM model starts with an input layer that receives *tokens*, followed by a *embedding* layer that converts them into dense vectors. The distinguishing feature of LSTM lies in its namesake layer, which includes *dropout* for regularization, an effective technique for preventing overfitting in neural networks. This layer is essential for capturing and preserving information throughout the text sequence, allowing the model to understand and process long-range dependencies fundamental in sentiment analysis. Subsequent dense layers process Word2Vec representations and sentiment polarity, culminating in an output layer that uses the sigmoid activation function, ideal for binary classification tasks.

Therefore, CNN and LSTM models were designed with a sequential and hierarchical approach, considering the peculiarities of text processing and the nature of sentiment analysis data. These architectures allow the efficient extraction of relevant features and patterns from textual data, which is fundamental for accurately classifying sentiments expressed in texts.

3.5 Evaluation and Reproduction of Experiments

All the traditional ML algorithms used in experiments were set with their corresponding default hyperparameter values provided by the scikit-learn library. DL models were configured to run for 100 epochs and with an *early stop* criteria configured for halting executing if the validation loss does not improve in 10 successive epochs.

The `Sentiment140` dataset was split using a holdout resampling strategy, with 60% of the data used for training and 40% for testing. For the DL models, a third of the training examples were separated as a validation set, keeping 40% for training, 20% for validation, and 40% for testing. Given the large database volume, a smaller percentage of training data was considered sufficient to achieve acceptable performance of the models without significantly compromising the quality of the results. A *walltime* of 300 hours was defined for each model to ensure adequate training time management.

DL models were trained with `Keras`, optimizing the binary cross-entropy loss, using the Adam optimizer. Accuracy, Precision, *Recall*, F1-Score, and AUC were used for performance metrics. These metrics were chosen because they provide a comprehensive view of the models' performance in different aspects, such as the correct prediction capacity, the balance between Precision and Sensitivity, and the ability to discriminate between classes.

The experiment environment was structured using Docker and Docker Compose and executed in a dedicated server: an HPE ProLiant DL360 Gen10 equipped with 24 Intel Xeon Silver 4214 CPUs at 2.20GHz. Two virtual machines (VMs) were configured on the server to execute the algorithms, each with eight processing cores, 64 GB of RAM, 96 GB of storage, and the Ubuntu Server 22.04 LTS operating system. It is important to highlight that GPUs were not used in experiments since they are unavailable on our university server.

Focusing on the reproducibility of the experiments, automated tests using `pytest`[4]. The testing approach made it possible to validate the functionality and behavior of the different code components, ensuring that all data manipulation and analysis processes were executed correctly without the need to stop and restart execution on the server. All the code developed in this study is publicly available in Github [5].

4 RESULTS

This section presents the experimental results. The following subsections will give one more details regarding some TF-IDF configuration definitions, overall ML and DL results, and projecting best-trained models in a new F1 unseen dataset for sentimental analysis.

4.1 Defining the best TF-IDF configuration

Firstly, a systematic experiment was conducted using the TF-IDF technique to establish the most effective n-gram configuration.

[4] https://docs.pytest.org/en/7.4.x/
[5] https://github.com/matheus-mileski/F1TT-Sentiment-Analysis

Table 4: Complete experimental setup.

Category	Tool	Objective
Container	Docker	Ensure consistency in project execution across different machines and operating systems.
Server	HPE ProLiant DL360 Gen10 2.20 GHZ, 24 CPUs, Intel Xeon Silver 4214	Running a large volume of jobs and data
Virtual Machines	8 cores, 64 GB RAM, 98 GB Storage, Ubuntu Server 22.04 LTS	Enable parallel execution of scripts
Libraries	Re, Emoji, Polars, Numpy, NLTK e URLLib TfidfVectorizer TextBlob JobLib, Numpy, Gensim, sciPy SeaBorn e MatplotLib Scikit-Learn e Keras PyTest	Data cleaning and preprocessing Transforming texts into TF-IDF matrices Sentiment polarity analysis in texts Feature extraction from texts Data visualization Model induction and evaluation Unit tests
Pre-trained models	Wor2Vec GoogleNews	Semantic word embedding
ML algorithms	LR, NB, SVM, RF, MLP	Classification using handcrafted features
DL algorithms	CNN, LSTM	Classification and feature extraction
Resampling	Holdout	Splitting data into training, validation and testing sets (40%, 20% and 40% respectively)
Loss	Binary cross-entropy	Train DL models
Optimizer	Adam	Reduce DL training loss
Evaluation Measure	Accuracy, Precision, Recall, F1-Score AUC	Assess induced models

This experiment involved the analysis of six distinct datasets generated from the Sentiment 140 dataset, each representing a specific configuration of n-grams. There were tested all possible permutations considering unigrams, bigrams, and trigrams: their values in isolation, a combination of 1-2 grams, 2-3 grams, and, finally, a configuration with all of them.

Each of these alternative datasets was subjected to an evaluation using four different algorithms: NB, LR, SVM, and RF. All models were implemented with the default hyperparameters provided by the scikit-learn library to ensure consistency and comparability of results. This methodological procedure was chosen to determine which n-gram configuration would most efficiently capture the nuances and variations present in textual data for sentiment analysis.

The experiment results revealed that the combination of unigrams, bigrams, and trigrams (1-to-3 grams) consistently excelled, outperforming the lower complexity n-gram configurations and demonstrating an improvement in key performance metrics. The superiority of this configuration can be attributed to its ability to capture a wider range of textual information. Unigrams provide a solid foundation by capturing the most frequent and relevant words. At the same time, bigrams and trigrams add a contextual layer by identifying common combinations of words and phrases that are frequently used together, reflecting specific nuances and language patterns. This proves particularly useful in tasks such as sentiment analysis, where understanding context and identifying specific expressions are key.

4.2 Overall results predicting tweet emotions

Once the best TF-IDF configuration was defined, we performed experiments with both ML and DL models. Overall training results revealed notable performance differences among algorithms. LR and LSTM models excelled, demonstrating balanced and reliable sentiment classification performance with an Area Under the Curve (AUC) of 0.86. LR stood out for its efficiency and rapid training time of 6.86 minutes, achieving 78.21% accuracy. Its effectiveness in handling binary variables and providing robust results proved viable for sentiment classification. LSTM, with a significantly longer training time of more than 7 hours and 13 minutes, showed comparable performance to LR with slightly lower accuracy but more balanced precision and recall. Further executions of these models could provide a deeper statistical performance analysis.

Other models, such as RF, MLP, and CNN, underperformed compared to LR and LSTM. RF's reliance on multiple decision trees might not suit the complexities of text data, while MLP struggles with sequential data. Effective in image processing tasks, CNN may not optimally capture text structure and semantics due to its architectural design.

Comparing neural network models, both CNN and LSTM were halted by early stopping after 12 training epochs, indicating rapid learning stagnation. For CNN, a decrease in validation accuracy and a significant increase in validation loss suggested overfitting,

limiting its generalization to new data. LSTM showed a less pronounced variation in validation accuracy and a moderate increase in validation loss, indicating a tendency toward overfitting but to a lesser extent.

In the comparison between LR and LSTM, both exhibited notable performance with accuracies around 78%. Confusion matrices for both models, Figure 2a and Figure 2b, indicated LR's slightly higher effectiveness in predicting true positives. At the same time, LSTM demonstrated a more balanced distribution between true positives and negatives. This subtle difference highlights LSTM's broader classification capacity, reflecting its proficiency in capturing long-range relations in text crucial for understanding expressed sentiments.

Table 5 presents a direct comparison between the sentiment predictions made by LR and LSTM models for tweets. Both models consistently classify positive sentiments in tweets with clear enthusiasm about F1 races and qualifications. However, it also reveals notable differences in their predictions. For instance, LR and LSTM diverge in interpreting tweets with more complex emotional content, highlighting LSTM's deeper contextual sensitivity, possibly detecting nuances and subtleties that LR misses.

Figure 3 shows a comparison between the predictions of both models, highlighting instances where LR and LSTM disagreed, underscoring the differences in their learning mechanisms and sensitivity to language nuances in F1 tweets. The x-axis shows all the individual tweets, while the y-axis lists all the predictive algorithms. Each cell paints the corresponding prediction obtained by an algorithm in that tweet. Black cells indicate texts predicted as negative, while gray cells indicate positive tweets.

In 142,045 instances, LR classified sentiments as positive, whereas LSTM interpreted them as negative, and conversely, in 141,007 instances, LSTM detected a positive sentiment where LR predicted a negative. These discrepancies highlight the distinct approaches of the two models in interpreting and classifying sentiments. On the other hand, both models concurred in classifying 115,754 instances as negative and 233,582 instances as positive.

Both models consistently classified positive sentiments in clearly enthusiastic statements about F1 races. However, they diverged in some predictions, suggesting LSTM's deeper contextual sensitivity, possibly detecting subtleties and ironies not captured by LR. Both models correctly classified negative sentiments in tweets criticizing race decisions or direction, demonstrating a reasonable interpretation of words and phrases contextualizing negative sentiments.

Despite these precise predictions, there were discrepancies in tweets with more subtle emotional complexity, indicating LSTM's better absorption of textual context. The comparison suggests that while LR may be faster in training, LSTM offers a more nuanced understanding of natural language.

4.3 Testing best model in F1 dataset

After model assessment and validation, the Long Short-Term Memory (LSTM) model was applied to the F1 tweets dataset to predict fan sentiment. This exploratory analysis validated the LSTM model in a real-world scenario and offered invaluable insights into the emotional dynamics prevalent in the context of F1 racing.

This application covered the period from July 24, 2021, to August 20, 2022, providing a comprehensive view of fan engagement and emotional response throughout different stages of the F1 season. The data obtained from this sentiment analysis enabled a detailed exploratory study, focusing on each race to decipher the emotional impact of the events on Formula 1 fans. The histogram in Figure 4 shows an increase in engagement during the season finales, reflecting the intensity of these races, contrasted with a quieter start to the subsequent season despite the highly emotional ending of the 2021 season.

Two of the Grand Prix, Abu Dhabi and Netherlands, stood out, showcasing distinct patterns of emotional engagement and public perception. The sentiment histograms of these Grand Prix are presented in Figure 5.

The Abu Dhabi Grand Prix on December 12, 2021, marked by a contentious battle for the title between Max Verstappen and Lewis Hamilton, resulted in Verstappen's controversial win. This race generated the highest volume of tweets in the dataset, with 54,877 positive and 38,857 negative tweets. The polarized nature of these tweets reflects the dramatic race conclusion and the emotionally charged debates it sparked among fans. The surge in tweet volume during this event underscores the impact of pivotal moments in sports on social media engagement.

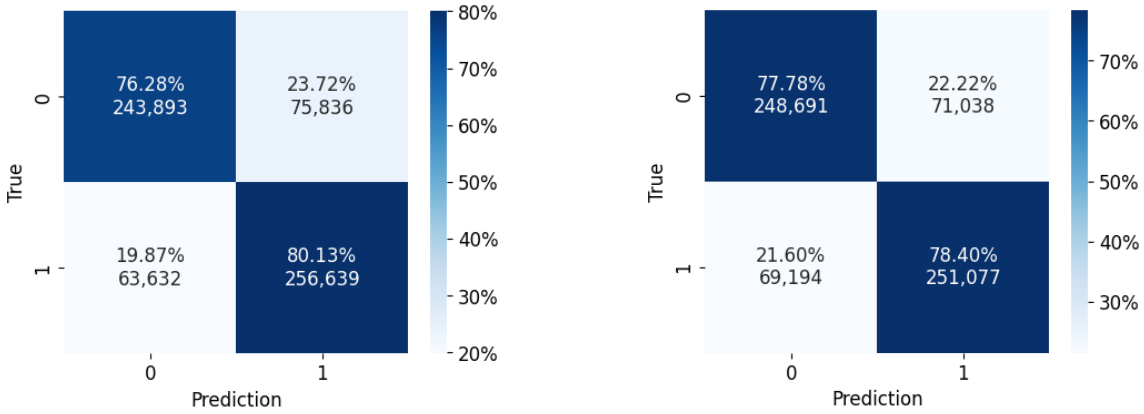
In contrast, the Netherlands Grand Prix on September 5, 2021, presented a unique case in the dataset. It was the only race where negative tweets (9,741) outnumbered positive ones (6,827). The tweet distribution showed a peak in negative expressions towards the race's conclusion. Given the context and events associated with this race, this anomaly merits careful consideration.

The return of F1 to the Zandvoort Circuit and Verstappen's victory may elicit positive reactions. However, the intense rivalry between Verstappen and Hamilton during that season likely contributed to the polarization. The heightened negative sentiment could also stem from logistical challenges and environmental concerns surrounding the event. Thus, the Netherlands Grand Prix highlighted how sports rivalries and external factors could significantly influence social media sentiment, diverging from the expected positive response to a local driver's victory.

5 CONCLUSIONS

This research successfully employed advanced Natural Language Processing (NLP) techniques to investigate the emotional dynamics of the Formula 1 (F1) fan community on Twitter, mainly focusing on the 2021 and 2022 seasons. A robust exploratory analysis was conducted by applying a Long Short-Term Memory (LSTM) model to public datasets. This analysis delved into each F1 race, extracting and examining sentiments and engagement patterns among users during significant events of the season.

Despite encountering challenges such as changes in Twitter API policies and the inherent limitations of public datasets, the study achieved its primary objective of identifying engagement patterns and user sentiments, utilizing sophisticated NLP techniques for sentiment analysis on Twitter. Other study's limitations, including the time-intensive nature of experiments and the technical complexities of NLP, were acknowledged but did not detract from the integrity and relevance of the outcomes.



(a) Logistic regression **(b) LSTM**

Figure 2: Confusion matrices of the best induced models

Table 5: Sentiment prediction of LR and LSTM

Tweet	LR	LSTM
Qualifying is going to be exciting! #F1 #USGP	Positive	Positive
How can you not like F1!? Brilliant racing. #BrazilGP #F1 #SkyF1	Positive	Positive
To sum up, going unnecessarily slow under safety car (no penalty), catching up to the car in front during a VSC (no penalty), driving into the back of another car (no penalty)…Max gives position back and still receives a 5 second penalty.… Seems a bit odd. #SaudiArabiaGP #f1	Negative	Negative
Every single F1 race the directing is appalling. How did they not have Ocon/Bottas on-screen? #F1	Negative	Negative
What a time to be a #F1 fan!	Positive	Negative
Never have I seen where only ONE car took a race start!! One for the record books? @SkySportsF1 @tedkravitz #F1 @f1 #HungarianGP https://t.co/tiObTmDkj5	Positive	Negative
Can we race in Saudi Arabia every week? This is brilliant. #f1	Negative	Positive
What an epic season and a bizarre ending. Wooooow. Max!!!! Yes!!!!! #f1	Negative	Positive

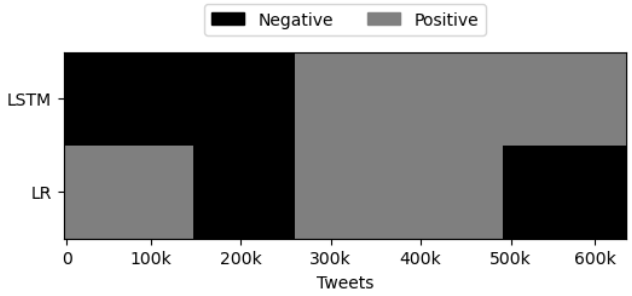


Figure 3: Comparison between predictions of LR and LSTM.

The insights gained from this study validated the LSTM model in a real-world scenario and offered a deeper understanding of emotional dynamics in the context of F1 racing. The findings contribute significantly to comprehending F1 fan behavior on social media, laying a solid foundation for developing more effective marketing and communication strategies in sports.

Future research in sentiment analysis, particularly within Formula 1 social media contexts, presents promising directions, including repeated experiments for robust statistical analysis and applying Named Entity Recognition to link sentiments with specific F1 entities. Advanced techniques like PCA and LDA could be explored for deeper data insights. The development of models to identify a range of emotions and the utilization of sophisticated language models like BERT and GPT offer potential for significant advancements. These efforts aim to enhance understanding and practical applications in sentiment analysis, addressing the complex emotional dynamics in the high-engagement arena of Formula 1.

In conclusion, this research provides valuable insights into the sentiment dynamics within the F1 fan community, reflecting the

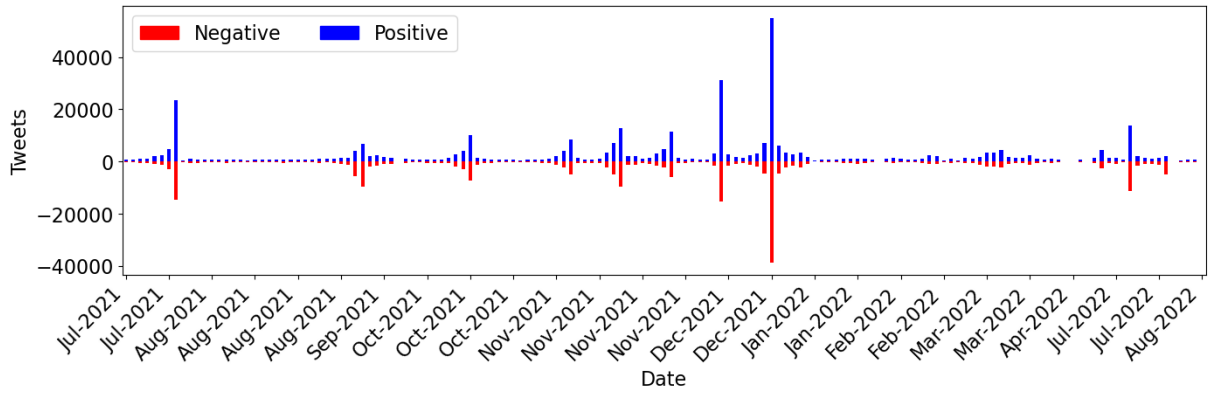
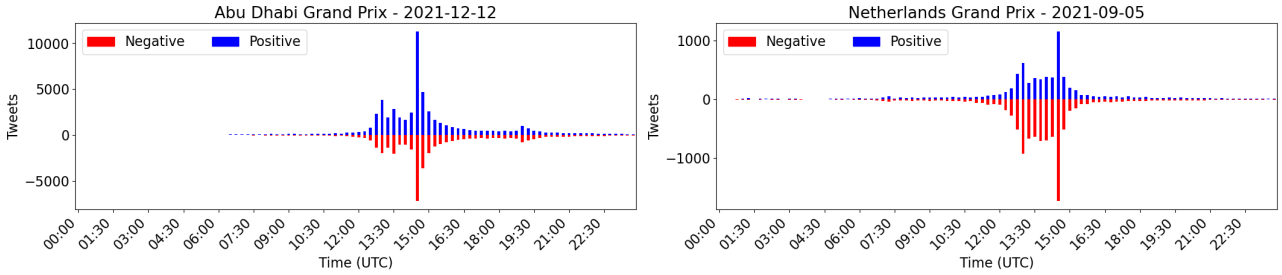


Figure 4: Sentiments histogram during the time-span analyzed.



(a) Abu Dhabi Grand Prix.

(b) Netherlands Grand Prix.

Figure 5: Sentiments histogram by Grand Prix.

challenges and potential of using advanced data science techniques in social media sentiment analysis, particularly in emotionally charged and dynamic contexts like motorsports events.

REFERENCES

[1] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. doi: 10.1126/science.aaa8415.

[2] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. 2021. URL https://d2l.ai/.

[3] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13:55–75, 08 2018. doi: 10.1109/MCI.2018.2840738.

[4] Bilal Saberi and Saidah Saad. Sentiment analysis or opinion mining: A review. *Int. J. Adv. Sci. Eng. Inf. Technol*, 7(5):1660–1666, 2017.

[5] Erik Cambria, Björn Schuller, Yunqing Xia, and Catherine Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013. doi: 10.1109/MIS.2013.30.

[6] Dastan Hussen Maulud, Subhi R. M. Zeebaree, Karwan Jacksi, Mohammed A. Mohammed Sadeeq, and Karzan Hussein Sharif. State of art for semantic analysis of natural language processing. *Qubahan Academic Journal*, 1(2):21–28, Mar. 2021. doi: 10.48161/qaj.v1n2a44.

[7] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[8] Alexander Pak, Patrick Paroubek, et al. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[9] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of Twitter data. In Meenakshi Nagarajan and Michael Gamon, editors, *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 30–38, Portland, Oregon, June 2011. Association for Computational Linguistics.

[10] Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. Topic sentiment analysis in twitter: A graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 1031–1040, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450307178. doi: 10.1145/2063576.2063726.

[11] Hassan Saif, Miriam Fernández, Yulan He, and Harith Alani. Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. In *1st Interantional Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013)*, 2013.

[12] Fajri Koto and Mirna Adriani. A comparative study on twitter sentiment analysis: Which features are good? In Chris Biemann, Siegfried Handschuh, André Freitas, Farid Meziane, and Elisabeth Métais, editors, *Natural Language Processing and Information Systems*, pages 453–457, Cham, 2015. Springer International Publishing.

[13] Pedro M Sosa. Twitter sentiment analysis using combined lstm-cnn models. *Eprint Arxiv*, 2017:1–9, 2017.

[14] Dionysis Goularas and Sani Kamis. Evaluation of deep learning techniques in sentiment analysis from twitter data. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pages 12–17, 2019. doi: 10.1109/Deep-ML.2019.00011.

[15] T Swathi, N Kasiviswanath, and A Ananda Rao. An optimal deep learning-based lstm for stock price prediction using twitter sentiment analysis. *Applied Intelligence*, 52(12):13675–13688, 2022.

[16] S Saranya and G Usha. A machine learning-based technique with intelligent-wordnet lemmatize for twitter sentiment analysis. *Intelligent Automation & Soft Computing*, 36(1), 2023.

[17] Serpil Aslan, Soner Kızıloluk, and Eser Sert. Tsa-cnn-aoa: Twitter sentiment analysis using cnn optimized via arithmetic optimization algorithm. *Neural Computing and Applications*, pages 1–18, 2023.

[18] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[19] Ralph Grishman. Information extraction. *IEEE Intelligent Systems*, 30(5):8–15, 2015.