CIS600: Social Media & Data Mining

CIS 600: SMDM Project Report

"Reddit WallStreetBets Sentiment Analysis"

Project Members:

1.Sameer Balkawade SUID: 200774310
2.Shivani Neharkar SUID: 320953757
3.Ryan D'mello SUID: 689654361
4.Mitil Jawale SUID: 754778650
5.Atharvaa Rane SUID: 275976322

Abstract:

This project conducts a focused analysis of sentiment within the r/wallstreetbets community on Reddit. Utilizing advanced natural language processing techniques, we aim to decipher the sentiments expressed in posts and comments, with a specific focus on understanding their potential impact on stock prices and trading strategies.

The study encompasses the identification and tracking of influential stocks frequently discussed on the subreddit, providing insights into emerging trends and illustrating the community's influence on specific equities. Additionally, the project explores trading behaviors within r/wallstreetbets, shedding light on risk tolerance, investment horizons, and preferred strategies, catering to the informational needs of both retail and institutional investors.

Beyond sentiment and stock identification, the research evaluates the alignment between r/wallstreetbets discussions and real-world market movements. Analysis of stock price changes, trading volumes, and options activity associated with discussed stocks aims to uncover the community's potential influence on broader financial markets.

To enhance accessibility, the project employs data visualization tools to create interactive charts and graphs, facilitating a clear and concise presentation of project findings. The insights derived from this analysis contribute to a deeper understanding of the intricate relationship between online sentiments within r/wallstreetbets and the dynamics of the stock market.

Introduction:

In the digital realm of stock trading discussions, the r/wallstreetbets community on Reddit stands out for its influential role. This project, titled "Reddit WallStreetBets Sentiment Analysis," focuses on understanding the sentiments expressed within this online community and their potential impact on stock market dynamics.

At its core, the project employs advanced natural language processing (NLP) techniques to analyze sentiments in posts and comments on r/wallstreetbets. By

decoding the collective mood, we aim to uncover how these sentiments might influence stock prices and trading strategies.

Simultaneously, the study identifies and tracks influential stocks frequently discussed on the subreddit. This provides insights into emerging trends and illustrates the community's impact on specific equities, serving as a real-time resource for investors seeking to align with sentiments and trends within r/wallstreetbets.

Additionally, we explore trading behaviors within the community, quantifying risk tolerance, investment horizons, and preferred strategies. This offers a nuanced view of the diverse trading dynamics within r/wallstreetbets, relevant for both retail and institutional investors.

Our research also evaluates the alignment between r/wallstreetbets discussions and real-world market movements. By analyzing stock price changes, trading volumes, and options activity associated with discussed stocks, we aim to uncover the community's potential influence on broader financial markets.

To enhance accessibility, the project employs data visualization tools to create interactive charts and graphs. This ensures that stakeholders can readily engage with and derive actionable insights from the sentiment analysis and trends within r/wallstreetbets.

In essence, the "Reddit WallStreetBets Sentiment Analysis" project seeks to unravel the sentiments within this influential online community, providing insights that contribute to a deeper understanding of the evolving relationship between online discussions and market dynamics.

Method:

We followed the below methodology for our project -

1. Data gathering
2. Data pre-processing
3. Experimentation with VADER and LSTM
4. Evaluation and Analysis

1. Data gathering

We've used Kaggle dataset for posts of subreddit WallStreetBets[1]. Our second method requires a labeled dataset and for this purpose we've used the Kaggle dataset of Twitter Sentiment Analysis[2].

2. Data Pre-Processing

The steps of preprocessing differ for both our approaches. Vader is versatile in handling the format of data but for LSTM we have performed various data cleaning and preprocessing steps. This involves eliminating extraneous and irrelevant information, label encoding, word vector representations. To help improve the quality of text data, we've also performed lemmatization and stop word elimination.

3. Experimentation

I. VADER

Sentiment analysis is a powerful technique employed to extract subjective information from textual data, gauging the emotional tone behind words. This analysis is particularly relevant in the realm of social media, customer reviews, and online forums where understanding user sentiments is crucial. In our dataset, sourced from the WallStreetBets subreddit on Reddit, we aimed to unravel the sentiment expressed in both post titles and bodies.

To achieve this, we leveraged the VADER sentiment analysis tool, a pre-built lexicon and rule-based sentiment analysis tool designed for social media text. VADER excels at analyzing sentiments in a nuanced way, considering not only the polarity (positive, negative, or neutral) but also the intensity of the sentiment.

The first step involved integrating the Natural Language Toolkit (NLTK) library, a widely-used platform for building Python programs to work with human language data. Specifically, we installed the NLTK library and downloaded the VADER lexicon to equip our sentiment analysis model with the necessary linguistic information.

Next, we implemented the sentiment analysis on both the title and body of the posts in our dataset. For the titles, we utilized the VADER analyzer to compute sentiment scores for each title, encompassing positive, negative, neutral, and compound scores. The compound score represents the overall sentiment, while the positive, negative, and neutral scores offer a finer granularity.

By employing VADER, we have harnessed a sophisticated sentiment analysis tool that excels in handling the unique characteristics of social media language. The compound scores and categorized sentiments allow us to capture the nuanced emotions expressed in the posts, providing valuable insights into the sentiment landscape within the WallStreetBets subreddit.

This sentiment analysis not only serves as a descriptive tool for understanding the prevailing sentiments but can also pave the way for more advanced analyses. The results could be leveraged to identify trends, correlate sentiment with stock price movements, or even predict market sentiment shifts based on the collective mood within the community.

In summary, our utilization of VADER in the dataset involves integrating a robust sentiment analysis tool, applying it to both titles and bodies of posts, and extracting meaningful sentiment-related features. This empowers us to gain a comprehensive understanding of the sentiments circulating within the WallStreetBets subreddit, contributing to a more nuanced analysis of the social media discourse surrounding financial markets.

II. LSTM + GloVe Embeddings

We've used a pre-trained GloVe embedding layer from Kaggle to represent each phrase as a 100-dimensional vector. The block size we've kept is 32. So for the text with less than 32 characters, we've 0-padded and for the text with more than 32 characters, we've truncated it. The output of these blocks has been routed through several dense layers and finally a softmax activation function for 3-way classification.

4. Evaluation and Analysis

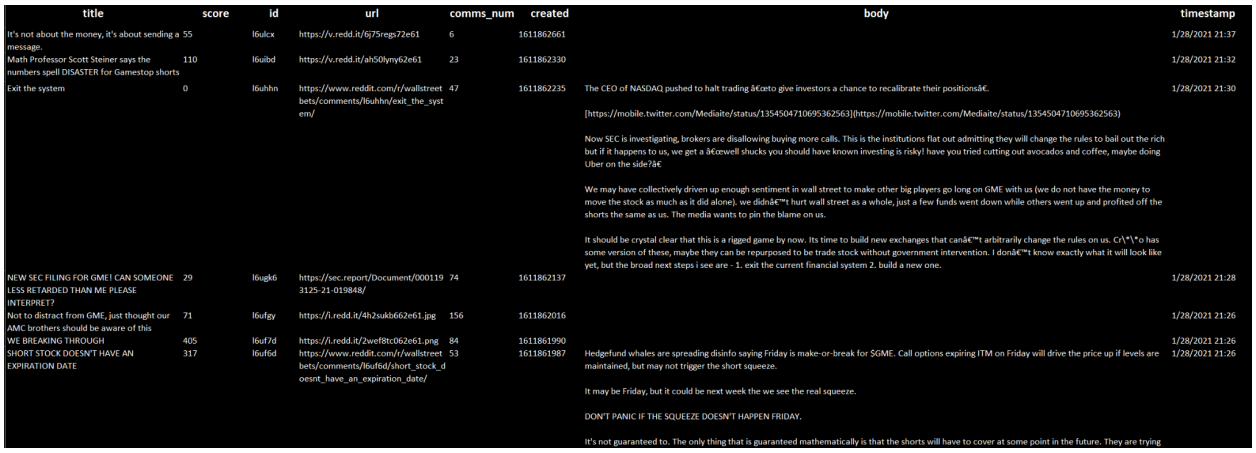
Finally we've compared the results we've got from both the approaches.

About the Dataset:

1. Dataset for VADER

This data is used to understand the trends initiated by the Reddit educated crowd. WallStreetBets (r/wallstreetbets, also known as WSB), is a subreddit where participants discuss stock and option trading,The dataset contains posts from Reddit starting from 29th January 2021. Be aware that data don't start before GME stock hype. Moreover, this dataset is regularly updated and the total number of rows increases.

Data Sources-
Reddit Posts and Comments: posts and comments from WallStreetBets
MetaData: Includes information like post titles, author, timestamp, upvote, downvotes and any attached links or images



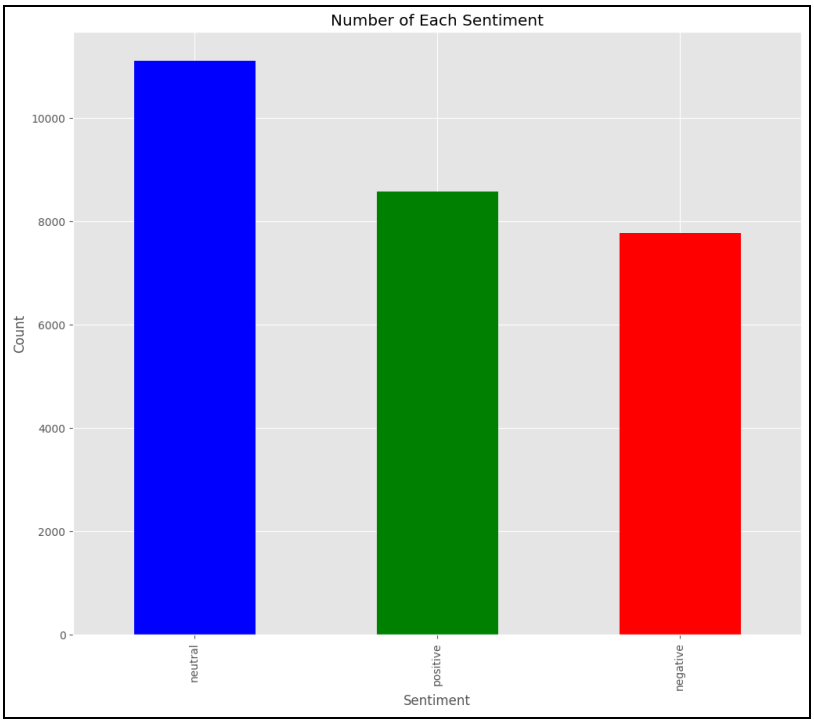
title	score	id	url	comms_num	created	body	timestamp
It's not about the money, it's about sending a message.	55	l6ulcx	https://v.redd.it/6j75regs72e61	6	1611862661		1/28/2021 21:37
Math Professor Scott Steiner says the numbers spell DISASTER for Gamestop shorts	110	l6uibd	https://v.redd.it/ah50lymy62e61	23	1611862330		1/28/2021 21:32
Exit the system	0	l6uhhn	https://www.reddit.com/r/wallstreetbets/comments/l6uhhn/exit_the_system/	47	1611862235	The CEO of NASDAQ pushed to halt trading â€œto give investors a chance to recalibrate their positionsâ€. https://mobile.twitter.com/Mediaite/status/1354504710695362563 Now SEC is investigating, brokers are disallowing buying more calls. This is the institutions flat out admitting they will change the rules to bail out the rich but if it happens to us, we get a â€œwell shucks you should have known investing is risky! have you tried cutting out avocados and coffee, maybe doing Uber on the side?â€ We may have collectively driven up enough sentiment in wall street to make other big players go long on GME with us (we do not have the money to move the stock as much as it did alone). we didnâ€™t hurt wall street as a whole, just a few funds went down while others went up and profited off the shorts the same as us. The media wants to pin the blame on us. It should be crystal clear that this is a rigged game by now. Its time to build new exchanges that canâ€™t arbitrarily change the rules on us. Cr**o has some version of these, maybe they can be repurposed to be trade stock without government intervention. I donâ€™t know exactly what it will look like yet, but the broad next steps i see are - 1. exit the current financial system 2. build a new one.	1/28/2021 21:30
NEW SEC FILING FOR GME! CAN SOMEONE LESS RETARDED THAN ME PLEASE INTERPRET?	29	l6ugk6	https://sec.report/Document/0001193125-21-019848/	74	1611862137		1/28/2021 21:28
Not to distract from GME, just thought our AMC brothers should be aware of this	71	l6ufgy	https://i.redd.it/4h2sukb662e61.jpg	156	1611862016		1/28/2021 21:26
WE BREAKING THROUGH	405	l6uf7d	https://i.redd.it/2wef8tc062e61.png	84	1611861990		1/28/2021 21:26
SHORT STOCK DOESN'T HAVE AN EXPIRATION DATE	317	l6uf6d	https://www.reddit.com/r/wallstreetbets/comments/l6uf6d/short_stock_doesnt_have_an_expiration_date/	53	1611861987	Hedgefund whales are spreading disinfo saying Friday is make-or-break for $GME. Call options expiring ITM on Friday will drive the price up if levels are maintained, but may not trigger the short squeeze. It may be Friday, but it could be next week the we see the real squeeze. DON'T PANIC IF THE SQUEEZE DOESN'T HAPPEN FRIDAY. It's not guaranteed to. The only thing that is guaranteed mathematically is that the shorts will have to cover at some point in the future. They are trying	1/28/2021 21:26

2. Dataset for LSTM

The dataset was obtained from Kaggle Twitter Sentiment Analysis dataset[2]. This dataset has separate training and testing dataset. The dataset has many features but currently we are only using the text column for our analysis. Nevertheless, below are the features -

- textID - The ID of the Tweet
- **text** - The actual tweet. We are only considering this column
- selected_text - Tweet Title
- **sentiment** - The target label
- Time of Tweet - Self explanatory
- Age of User
- Country
- Population-2020
- Land Area
- Density

We have a total of 27481 tweets from which 8582 are positive, 7781 are negative and 11118 are neutral. The class distribution is fairly similar and hence there won't be any bias in training of the model. We could reduce the neutral class slightly so it doesn't overfit for the neutral class.



Sentiment distribution in Twitter Data

Data Pre-Processing:

The original dataset of tweets had several issues. Firstly, the tweets included tags (denoted by "@") and hyperlinks, both of which were filtered out. Tags typically reference individuals, and hyperlinks lead to external websites. In addition, we converted all text to lowercase and removed punctuation, as varying capitalizations can lead to different word embeddings. We also discarded all stop words, such as 'the', 'of', 'for', and other commonly used words. For standardizing word forms, we employed lemmatization, which transforms a word to its base form; for example, 'geese' is converted to 'goose'. Furthermore, emojis present in tweets were also removed, as they are not suitable for our training model. The preprocessing of all tweets predominantly utilized the "re" and "nltk" libraries. Each step of this process is explained in detail with an example for clarification.

a) Conversion to Lowercase: The lower() function was utilized to transform the text into lowercase. This standardization simplifies the processing and analysis of the text.

b) Elimination of Punctuation: Punctuation, generally lacking significant meaning, was removed to streamline the text.

```
# pre-processing of text in 1 single function
def preprocess_text(text):
    # remove links from text - https
    non_http_list = [w for w in text.lower().split() if 'http' not in w]
```

c) Removal of Mentions: In handling tweet data, which often contains mentions (user references), these were excluded to prevent complications in the data analysis process.

```
# remove @ words from tweets
at_removed_list = [w for w in non_http_list if '@' not in w]
```

d) Elimination of URLs: The dataset's raw tweets included various hyperlinks redirecting to external sources. These were filtered out as they were deemed irrelevant to the data.

```
# pre-processing of text in 1 single function
def preprocess_text(text):
    # remove links from text - https
    non_http_list = [w for w in text.lower().split() if 'http' not in w]
```

e) Lemmatization, a prevalent method in Natural Language Processing (NLP), is employed to transform different inflected forms of a word to their base or root form. This technique aids in normalizing the text, ensuring that various forms of a word are consolidated into a unified representation.

```
# lemmatization
lemmatized = " ".join([lemmatizer.lemmatize(w) for w in tokens])

return lemmatized
```

Apart from the above mentioned steps, we also removed stop-words which were very common like ‘the’, ‘in’, ‘of’, etc.

Word representations using global vectors:

After preprocessing the tweets, we adopt a method known as Global Vectors for Word Representation (GloVe) to map the words into a vector space. This technique translates

each word into a vector of length 100. Consequently, the output of this network is an array with dimensions N x 100, where N represents the total word count in the network. To align with the LSTM's predefined state count, we reshape this array from N x 100 to 32 x 100, padding the output as necessary. In instances where a word from our training dataset is transformed and not found in the existing vocabulary, it is assigned a unique random vector of 100 dimensions. This transformation ensures that words with similar meanings are positioned closer to each other in the resulting vector space.

Approach 1: VADER

In our sentiment analysis methodology tailored for Reddit posts, we opted for the VADER (Valence Aware Dictionary and sEntiment Reasoner) approach, specifically designed for social media text. As Reddit posts typically comprise both titles and bodies, this method accommodates the inherent nuances present in both components. After standard preprocessing steps, such as converting text to lowercase, removing stopwords, and addressing special characters, we applied VADER to evaluate the sentiment of each post. Leveraging a pre-built lexicon with sentiment scores for words common in social media, VADER captures the polarity (positive, negative, or neutral) and intensity of sentiments.

During the analysis, each word in both the title and body contributed to an overall sentiment score. VADER considered contextual nuances, including negations and intensifiers, enhancing its comprehension of sentiment expression within the distinctive language used in Reddit discussions. The output was a compound sentiment score for each Reddit post, reflecting the aggregate sentiment of both title and body. The categorization of posts into positive, negative, or neutral sentiments relied on a predefined threshold, which could be adjusted for specific sensitivity requirements. This VADER-based approach provided an efficient solution for sentiment analysis, offering rapid insights into the sentiment of Reddit posts within the preprocessing pipeline.

Unlike tweets, Reddit posts often contain more extended and diverse content, requiring a nuanced approach to sentiment analysis. The VADER method's adaptability to both titles and bodies allowed us to capture sentiment nuances present in different parts of the post. By integrating VADER into our preprocessing pipeline, we achieved a comprehensive understanding of the overall sentiment in Reddit discussions, making it a valuable tool for sentiment analysis in this dynamic and content-rich social media

platform.

```
# Apply VADER to the title column
title_scores = df['title'].apply(lambda x: sid.polarity_scores(str(x)))
df['title_positive'] = title_scores.apply(lambda x: x['pos'])
df['title_negative'] = title_scores.apply(lambda x: x['neg'])
df['title_neutral'] = title_scores.apply(lambda x: x['neu'])
df['title_compound'] = title_scores.apply(lambda x: x['compound'])
df['title_sentiment'] = df['title_compound'].apply(lambda x: 'positive' if x >= 0 else 'negative')

# Apply VADER to the body column with error handling
def get_body_sentiment(x):
    try:
        return sid.polarity_scores(str(x))
    except:
        return {'pos': 0.0, 'neg': 0.0, 'neu': 1.0, 'compound': 0.0}

body_scores = df['body'].apply(get_body_sentiment)
df['body_positive'] = body_scores.apply(lambda x: x['pos'])
df['body_negative'] = body_scores.apply(lambda x: x['neg'])
df['body_neutral'] = body_scores.apply(lambda x: x['neu'])
df['body_compound'] = body_scores.apply(lambda x: x['compound'])
df['body_sentiment'] = df['body_compound'].apply(lambda x: 'positive' if x >= 0 else 'negative')
```

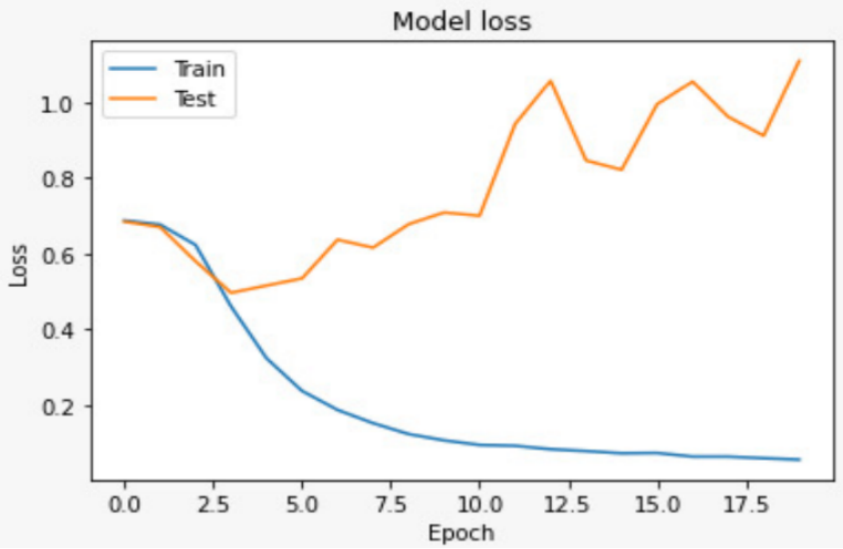
Approach 2: LSTM + GloVe

We sequentially process each word within the given phrase by utilizing an LSTM network. In order to represent each sentence as a 100-dimensional vector, we employ both a trainable embedding layer and a pre-trained GloVe embedding layer from Twitter. It's worth noting that tweets are limited to a maximum of 32 characters. For tweets shorter than 32 characters, we pad them with zero vectors before feeding them into the LSTM layer. Conversely, for tweets exceeding 32 characters, we truncate them from the end. The resulting output from these LSTM cells undergoes a series of complex layers, ultimately culminating in the determination of the sentiment of the tweet.

During the data preprocessing phase, we followed standard procedures, including converting text to lowercase and removing stopwords. We also removed emojis, hyperlinks, and tags from our Twitter dataset since they don't contribute to understanding tweet content. Instead of stemming, we opted for lemmatization to simplify words into their basic forms. This choice was made because stemming sometimes results in unrealistic word forms, such as 'coming' being truncated to 'com,' which doesn't exist in the pre-trained GloVe dictionary and negatively impacts performance. The next step was selecting a word embedding method to represent words in a vector space. Two commonly used approaches are GloVe and Word2Vec, both aiming to capture semantic relationships effectively. We chose the GloVe embedding method because it includes an embedding model pre-trained on a vast Twitter dataset containing 2 billion tweets, making it more suitable for our task and potentially offering better representations for Twitter-specific phrases. Furthermore, it

delivers performance comparable to Word2Vec and is relatively straightforward to grasp.

We conducted experiments using basic vectorization techniques because they are both straightforward and effective for addressing a wide range of issues. It is evident that we must generate vectors thoughtfully and understand the long-term relationships between words to improve performance on various tweet types. As a result, we tested an LSTM-based model with a trainable embedding layer. However, after approximately three epochs, the train/test loss curve depicted in the figure below revealed a significant issue. While the training loss continued to decrease, the test loss saw a sharp increase, indicating clear signs of overfitting. One potential explanation for this phenomenon could be the lack of sufficient data to adequately train both the embedding layer and the LSTM model.



Overfitted Curve

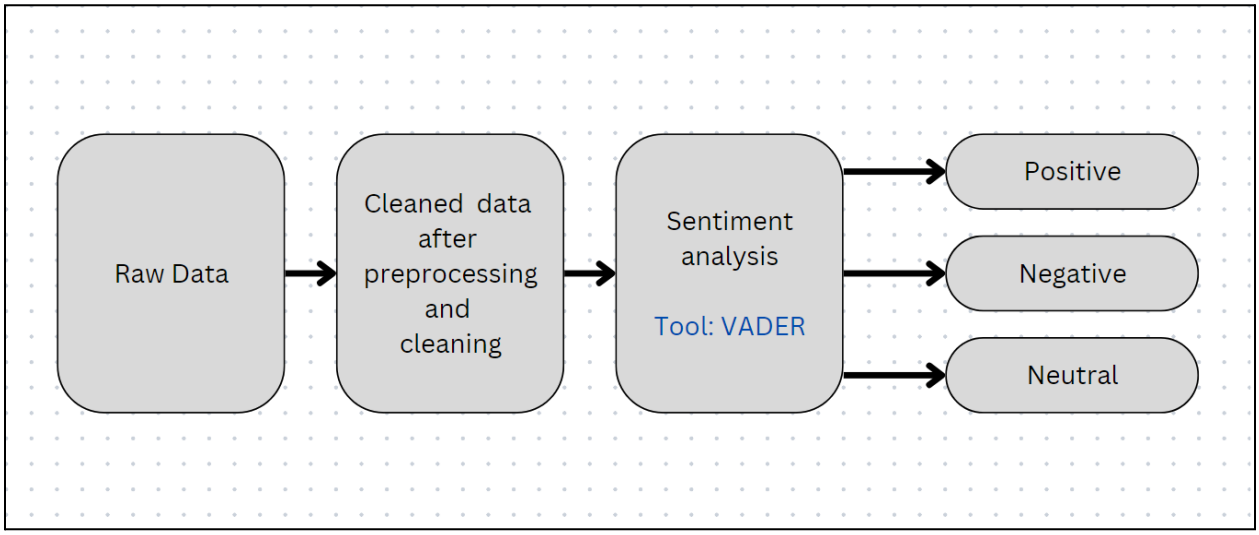
To mitigate the need for training a large number of weights, as discussed in our primary approach, we substitute the trainable embedding layer with pre-trained GloVe embeddings. This adjustment aims to tackle the overfitting problem.

Pretrained Embeddings:

The training and validation data were divided into 80% for training and 20% for validation. Text preprocessing and enhancement techniques were applied. We employed two pre-trained GloVe embeddings, one trained on Wikipedia text and the other on Twitter tweets. We fine-tuned various hyperparameters for each fully

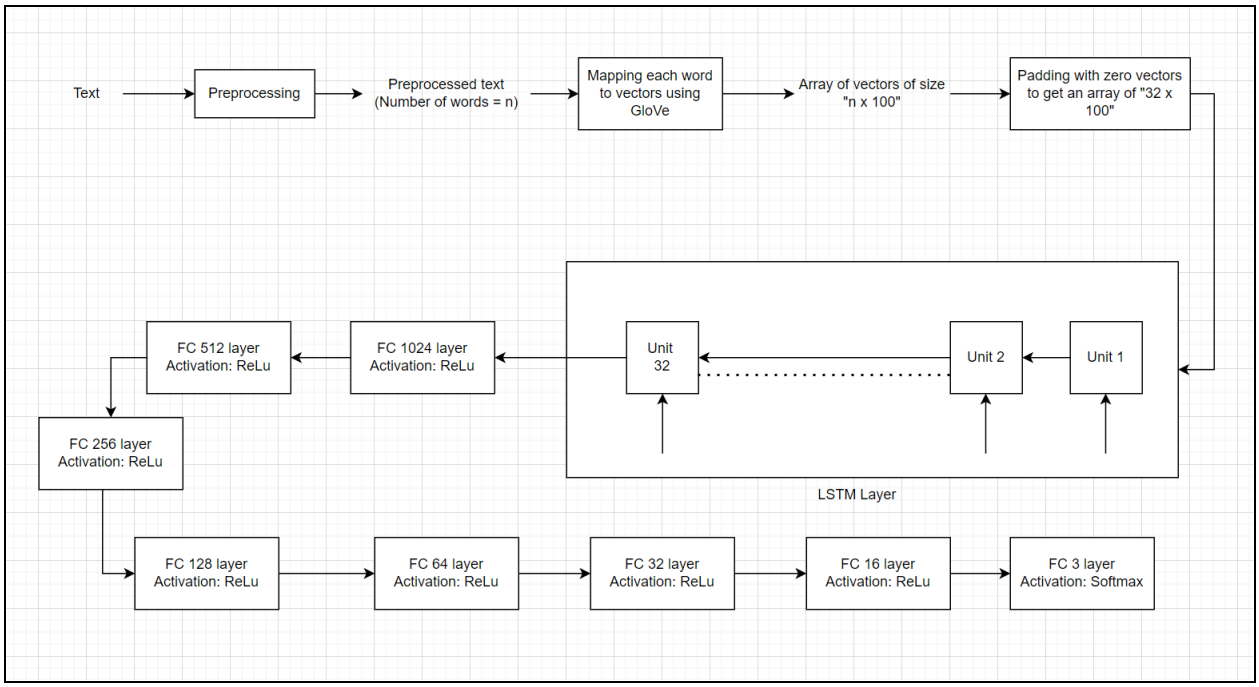
connected (FC) layer, including batch size, learning rate, and dropout rate. To evaluate this approach, we measured the mean F-1 score on the test set and the validation accuracy on the validation set using Kaggle.

Architecture:



VADER Architecture

The VADER architecture employs a lexicon-based approach for sentiment analysis, leveraging a pre-built dictionary with sentiment scores assigned to words. It considers both the polarity and intensity of sentiments in each word, allowing for nuanced sentiment analysis. VADER utilizes a rules-based system to handle complexities such as negations and intensifiers, enhancing its contextual understanding. The output is a compound sentiment score derived by aggregating individual word scores, providing an overall sentiment assessment for the given text. This architecture is particularly effective in capturing sentiment nuances within short-form social media text. Its adaptability to diverse language contexts and ability to discern sentiment strength make it a powerful tool for quick and context-aware sentiment analysis tasks.



GloVe + LSTM Model Architecture

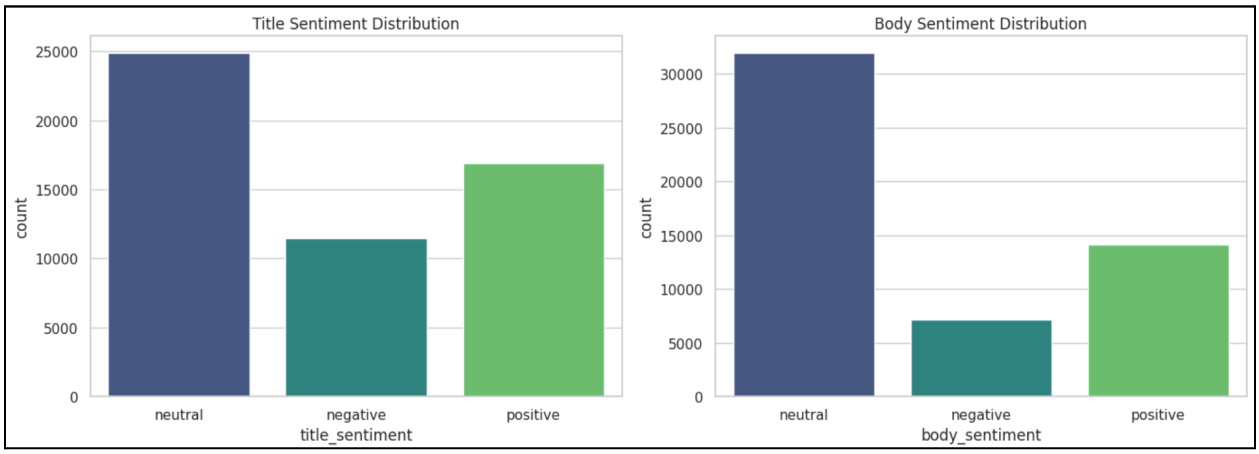
The depicted diagram illustrates the workflow of our ultimate architecture. To begin, we preprocess the data by performing several actions: converting the text to lowercase, eliminating emojis, hyperlinks, stopwords, punctuation, and tags. We undertake these actions because they do not provide any meaningful contribution to tweet analysis. Additionally, we apply lemmatization to transform words into their base forms.

Moving forward, when it came to employing word embedding techniques for converting words into vectors, we opted for the GloVe embedding method. Our preference for GloVe over Word2Vec stemmed from the fact that GloVe had undergone specific pre-training on a Twitter dataset, making it better suited for our specific use case—generating improved representations that are particularly relevant to Twitter. Furthermore, GloVe proved to be more straightforward to comprehend and exhibited performance on par with Word2Vec.

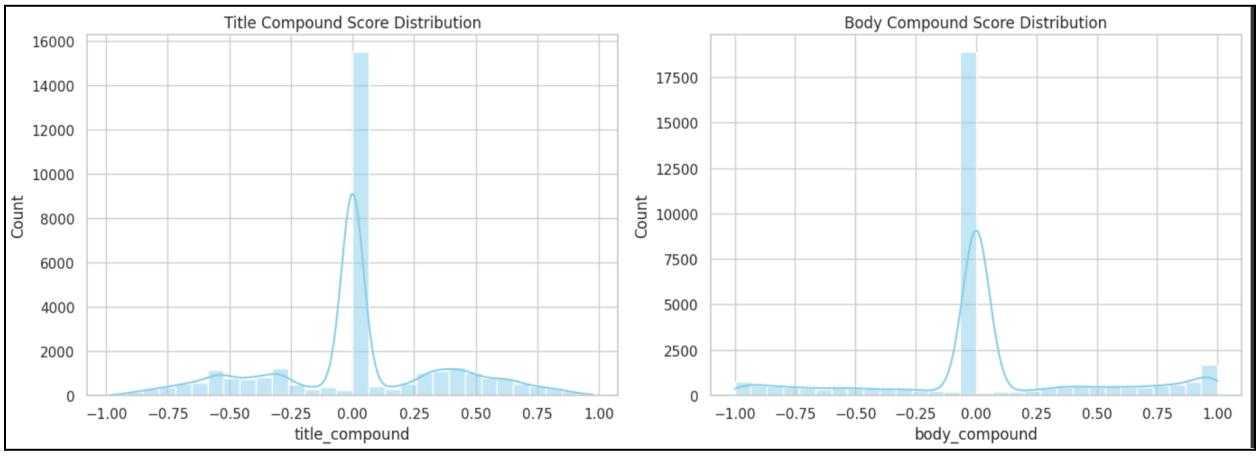
Results:

VADER -

1. Sentiment Distribution:
   The count plots depict a prevalence of positive sentiment in both titles and bodies. The majority of entries exhibit positive sentiment, contributing to an overall positive trend in the dataset.

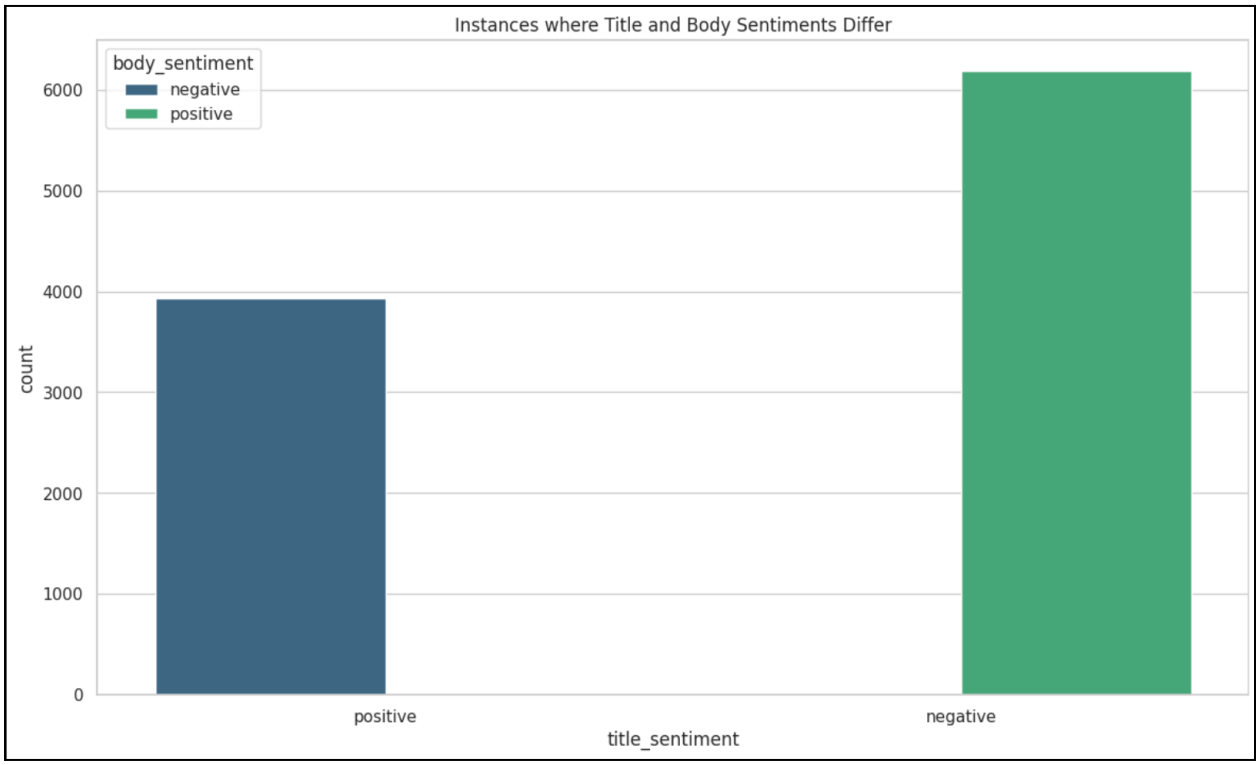


2. Sentiment Discrepancies:
   The scatter plot reveals instances where the sentiment in titles and bodies differs. Points deviating from the diagonal line highlight discrepancies, signaling variations in sentiment expressions between titles and bodies. Further investigation into these deviations can provide insights into nuanced sentiment patterns within the dataset.

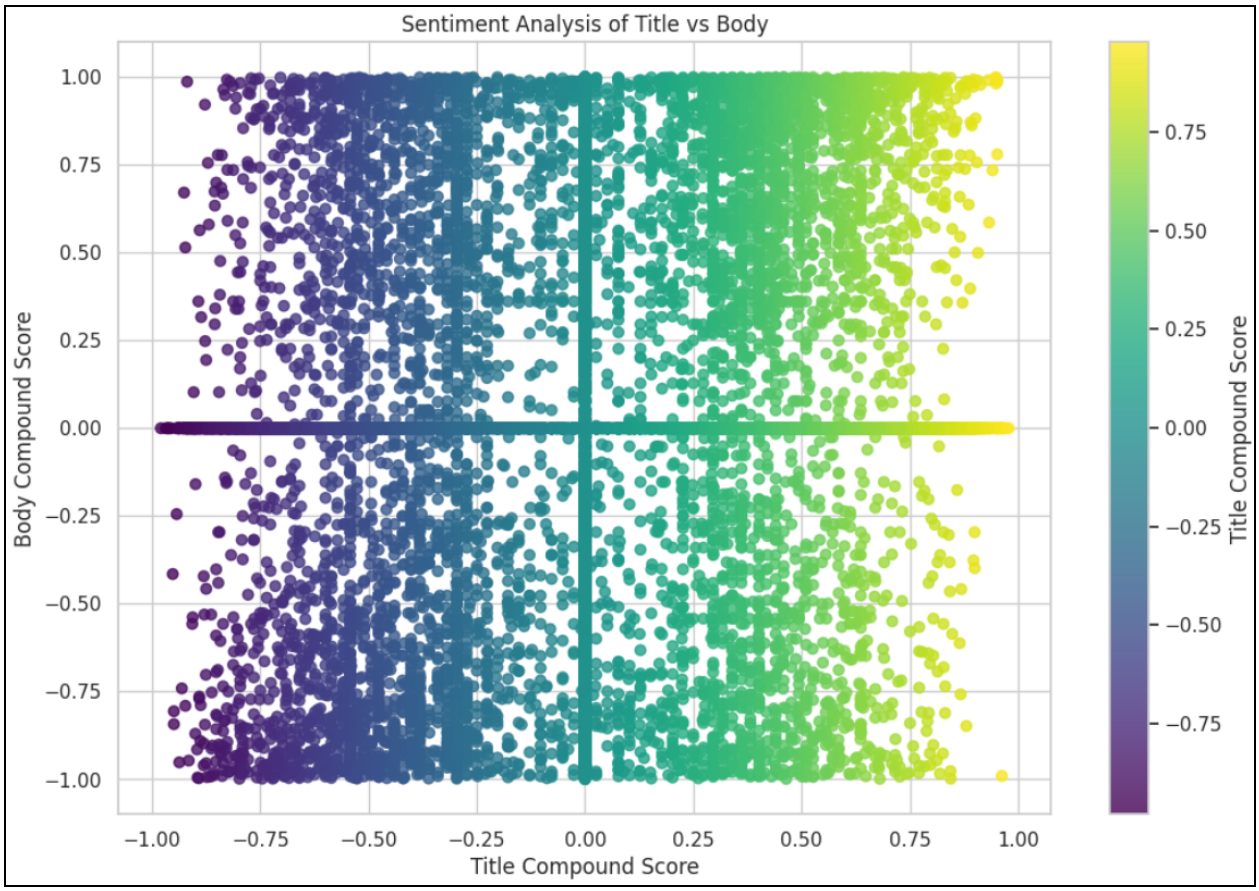


3. Mismatched Sentiments Analysis:
   By systematically identifying instances where title sentiment contradicts body sentiment, our analysis reveals nuanced discrepancies. The combined bar plot

visually demonstrates entries where titles, initially marked as positive, exhibit negative sentiment in the body, and vice versa. These mismatches offer valuable insights into potential complexities in sentiment expression within the dataset.

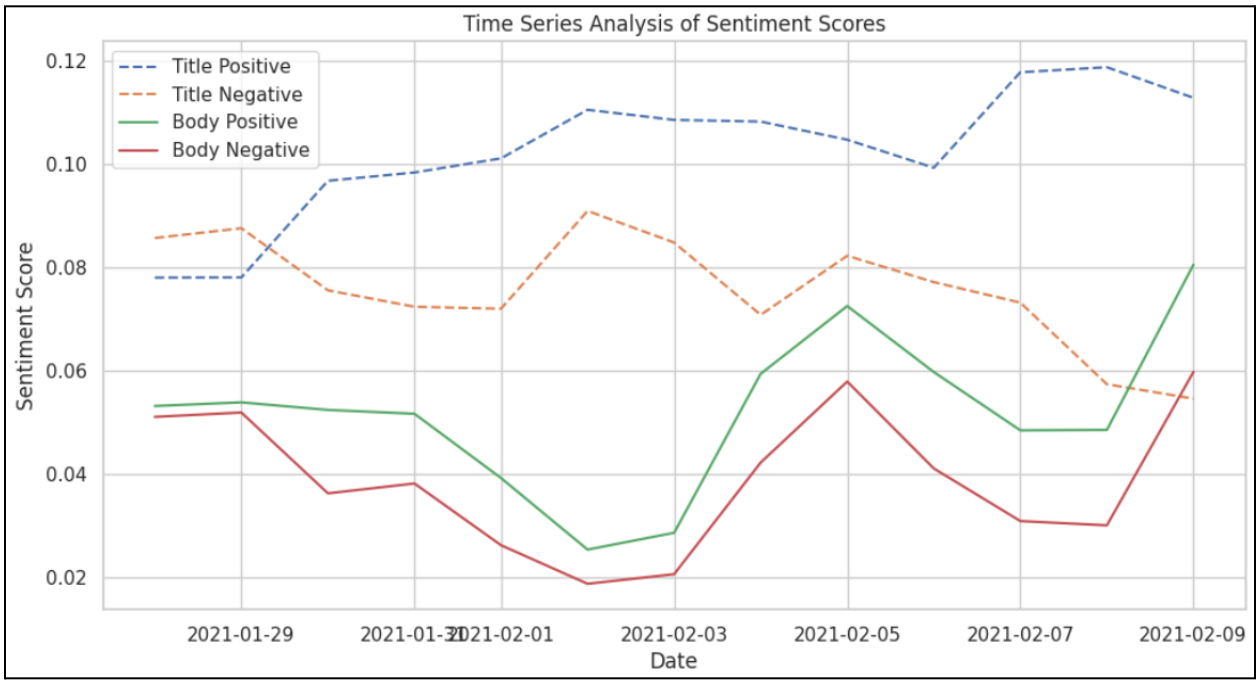


4. Sentiment Discrepancy Visualization:
   Utilizing a scatter plot, the technical analysis visually maps the compound sentiment scores of titles and bodies. The x-axis represents the title's compound sentiment score, while the y-axis signifies the body's compound sentiment score. The color of each point corresponds to the title's compound score, indicated by a color bar. Deviations from the diagonal line highlight instances where title and body sentiments diverge, offering a detailed view of sentiment discrepancies within the dataset.

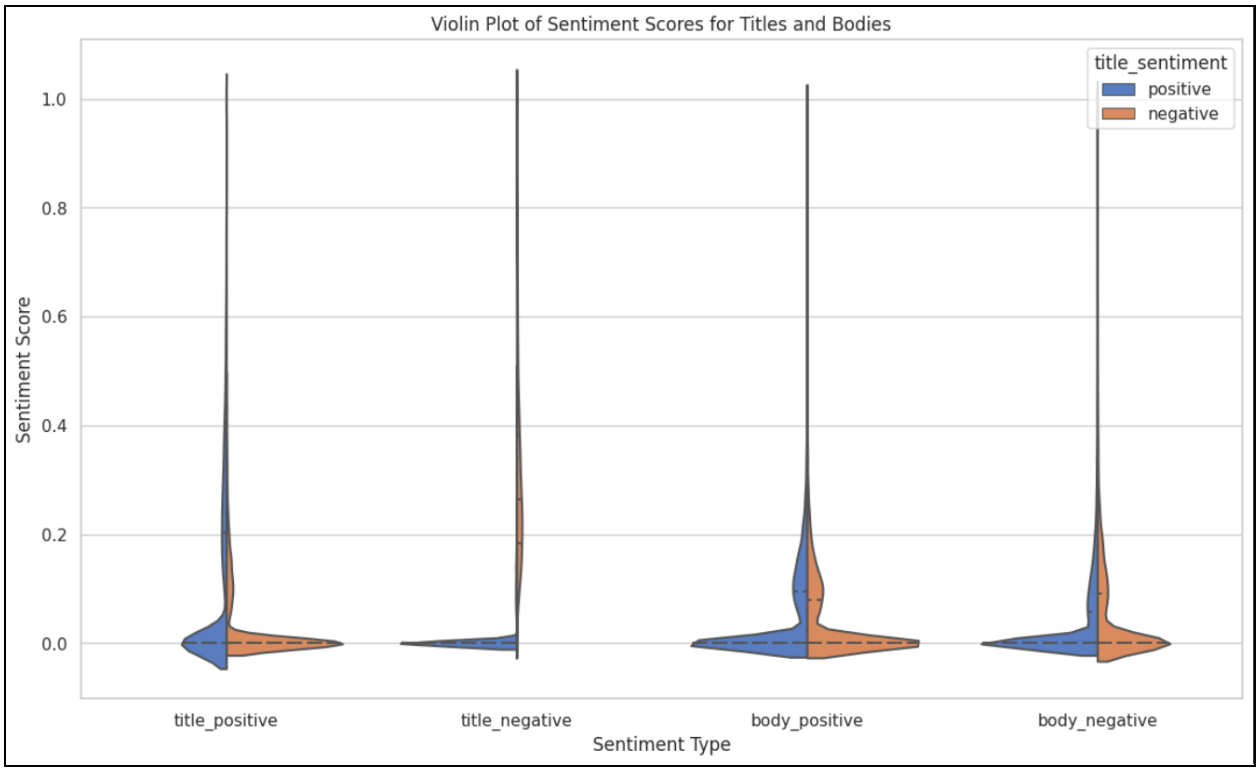


5. Time Series Analysis of Sentiment Scores:
   The code employs time series techniques on the 'timestamp' column, assuming it contains datetime information. The data is resampled to daily frequency, and the mean sentiment scores for positive and negative sentiments in both titles and bodies are calculated. The resulting time series plot visualizes the temporal trends of sentiment scores, showcasing daily fluctuations and patterns. The dashed lines represent sentiment scores for titles (positive and negative), while solid lines represent scores for bodies, providing a comprehensive view of

sentiment dynamics over time.



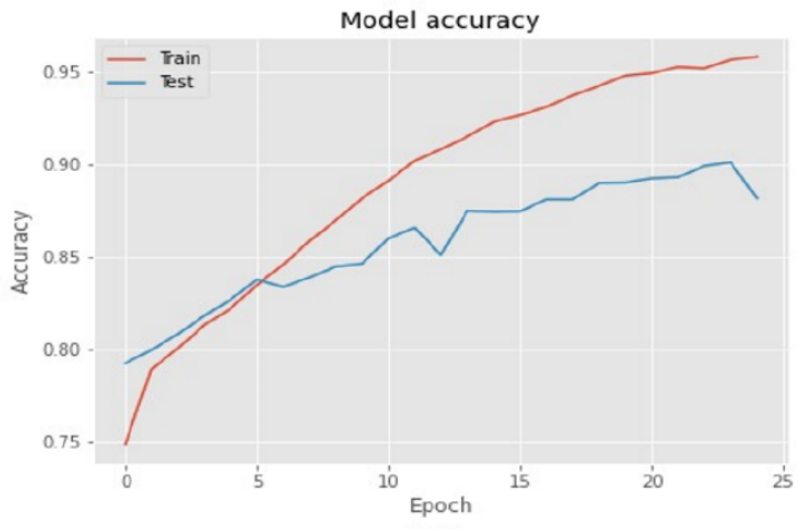
6. Violin Plot of Sentiment Scores:
   The code utilizes seaborn to create a violin plot for visualizing sentiment scores in titles and bodies. The data is reshaped for better plotting, and the violin plot showcases the distribution of sentiment scores, distinguishing between positive and negative sentiments. The plot provides a concise view of sentiment variations within each sentiment category, offering insights into the distributional characteristics of sentiment scores for titles and bodies.



LSTM

The below parameters gave us the best results on the test data -

- Batch size = 64
- Number of Epochs = 25
- Dropout Rate = 0.2
- Learning Rate = 0.01
- Optimizer = Adam
- Loss Function = Categorical Cross Entropy



Model accuracy

	precision	recall	f1-score	support
0	0.90	0.90	0.90	2614
1	0.87	0.87	0.87	1954
accuracy			0.89	4568
macro avg	0.89	0.89	0.89	4568
weighted avg	0.89	0.89	0.89	4568

Model classification report

Comparison between both the methods -

Architecture	# Positive posts	# Negative Posts	# Neutral Posts
Vader	14120	7149	31918
Glove + LSTM	14894	6789	31504

Some of the differences of sentiment classification that we found out with respect to both our methods -

Post Title	Vader	GloVe + LSTM
Feeling like exercising, dunno may delete this later	Positive	Negative
begging is cringe	Neutral	Negative
The Wallstreetbets War Museum	Negative	Neutral
No fear! This is going to be the new winner! $XSPA	Negative	Positive

Conclusion:

- Our sentiment analysis of the WallStreetBets subreddit provides valuable insights into the emotions and sentiment expressed within this influential online community.
- We found that WallStreetBets exhibits a dynamic emotional tone, closely tied to real-world financial events and market performance.
- Emotions within the subreddit range from exuberance to frustration, reflecting the diverse sentiments of retail investors.
- The data collected from WallStreetBets may be subject to biases, as it represents only a subset of user-generated content.
- Some members might use sarcasm or humor in their posts, which can be hard for sentiment analysis tools to understand correctly.

References:

[1]. https://www.kaggle.com/datasets/gpreda/reddit-wallstreetsbets-posts/data.
[2]. https://www.kaggle.com/datasets/rtatman/glove-global-vectors-for-word-representation/data.
[3]. https://www.kaggle.com/datasets/abhi8923shriv/sentiment-analysis-dataset.