University of Essex

Centre for Computational Finance and Economic Agents

CF981 DISSERTATION

YOUR PROJECT TITLE HERE

CHRISTOPHER JASON SAGAYARAJ

Supervisor: **Dr. Alexandros Voudouris**

August 26, 2023
Colchester

Abstract

This dissertation explores the impact of the internet community, particularly Reddit's r/wallstreetbets, on the stock market. Utilizing sentiment analysis and data processing techniques, this study aims to understand how the popularity and discussion of certain companies within the meme community influence stock prices. The data collected from Reddit is analysed with sentiment analysis models such as Finbert, Vader, Roberta, and GPT-3.5, a large language model. These scores are then analysed with neural network models to observe if the sentiment analysis could provide us with any indication of how the returns of the stock are affected from one day to one week.

Acknowledgements

I would like to thank my family for giving me this opportunity, especially my mother for always cheering me on.

Contents

List of Figures

CHAPTER 1

Introduction

The advent of social media has catalyzed a shift in the financial world, particularly in the context of predictive analytics and price forecasting. The ability to gauge public sentiment and translate it into actionable insights, where opinions and mass sentiments can sway market trends in unprecedented ways, has become an interesting field of study in financial research.

The rise of internet communities such as r/wallstreetbets has given many retail investors a place to discuss and comment on the events occurring in the stock market. Costola et al. (2023) [8] studies the phenomenon of meme stocks or "mementum". The authors defined these stocks based on 3 conditions. 1. There should be considerable coordinated social media signals about a stock, and these tweets are compared between the price and the trading volume. 2. These discussions from the social media sites need to be in sync with the change in stock price and volume. 3. The changes in tweet volume and the price and stock volume need to be persistent over a period of time. Using this definition, the authors studied how social data affects the market. They argue that it can be a form of market manipulation when certain conditions are met. The paper studies the discourse originating from this community on certain stocks such as GME, AMC and KOSS. All these stocks witnessed a rise in tweet volume, volume traded and subsequently, the price also saw a sharp increase. This thesis aims to delve into the heart of this new paradigm by analyzing the impact of online communities on stock market behaviour.

We have collected headlines from r/wallstreetbets and applied state-of-the-art sentiment analysis models to derive numerical sentiment scores. This rich data set represents the collective psychology of online retail investors, providing an empirical foundation for our analysis.

Leveraging advanced neural networks like LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network), and a combined CNN-LSTM model, this research predicts the bidirectional return of stocks influenced by online communities. Our methodology integrates cutting-edge machine learning techniques with traditional financial analysis, offering a comprehensive understanding of how social media sentiment translates into market action.

In the subsequent chapters, we will present our methodology, data collection, sentiment analysis, predictive modelling, and a thorough discussion of the results. By exploring the nexus between online communities like r/wallstreetbets and the stock market, this thesis strives to shed light on a new frontier in financial research - one where the voices of individual investors resonate as powerfully as those of institutional giants.

CHAPTER 2

Literature Review

The literature review will delve into related works that have sought to understand the relationship between social media discourse and financial markets to make an algorithmic decision on whether the returns can be predicted. This paper also tries to study the difference in performance between the standard text-classification models such as FinBERT, Vader, and Distil-Roberta and also tests the performance of GPT-3.5 on sentiment analysis.

The main inspiration for this paper is highlighted in the next section and then the subsequent literature will focus on the models and methods used to build this paper.

2.1 Sentiment Analysis for Financial Modelling

A recent study by Zou and Herreans (2022) [30] proposes a multimodal model for Bitcoin's extreme price prediction using the tweets gathered. The paper mentions how Bitcoin's price is susceptible to market sentiments, citing examples of price changes in response to tweets by Elon Musk. The data consists of two primary components: Twitter text analysis and Technical Analysis (TA). The dataset called PreBit contains over 9 million tweets with the keyword "Bitcoin". This study utilizes normalized data and FinBERT embeddings of 768-dimensional vectors to capture the full context of tweets for a meaningful representation of Twitter data. This paper implemented two different CNN architectures; a parallel (2.6 million trainable parameters) and a sequential model (7.6 million). The authors chose FinBERT [21] for text classification, as the model is pre-trained on six tasks of over 61GB of text from various financial websites. Zou and Herraans [30] claim that FinBERT outperformed BERT by 10 to 20% in the Financial PhraseBank dataset.

Coppens (2022) [7] highlights similar market behaviour, especially in Bitcoin alternate currencies known as altcoins. The author studies the various subreddits dedicated to these coins including r/Bitcoin, r/CardanoTrading, and r/SafeMoonBuySellAdvice just to name a few. Coppens studies the state of the sentiment analysis and also how cryptocurrency prices fluctuate based on the sentiments from the subreddits. The author chooses three lexicon models to derive the sentiment score of the posts from these subreddits and concludes that VADER [15] outperformed the other tested models. Coppens points out the fact that VADER was trained to capture the sentiment values on social media text thus leading her to choose this model. The correlation of the sentiment values and the crypto prices revealed that strong-performing cryptos had a higher positive-to-negative ratio in sentiment scores, while the lower average in positive score to negative score in weaker-performing cryptos was observed.

Carter (2022) [3] studies the subreddit r/wallstreetbets with a dataset found on Kaggle containing discussions dating back to 2012. This Reddit information combined with the stock data was split into different datasets for each stock. The analysis of this data results in GME having the most number of mentions, this matches with the work from Costola et al. (2021) [8] which points out the correlation between GME price movement and a high volume of tweets during the same period. Carter (2022) [3] opted to employ Flair over VADER for the sentiment analysis of the headlines. Carter argued that Flair provided better results as it can pick up on sentence context but at the cost of longer classifying times. However, the author does state that the output of VADER and Textblob are easier to work with as they provide float values. The author also compared the efficiency of price prediction with and without the sentiment analysis gathered from the dataset. For this, a Long Short Term Memory (LSTM) model was used with 128 units at each layer resulting in 461,441 trainable parameters. It was observed that some stocks such as AMD performed poorly when WSB data was solely used to predict the price of the stock, but stocks like SNAP are more responsive to the hype, discussions, or sentiments expressed within this specific online community.

Although the literature in Financial sentiment analysis is vast, these papers provided a good base on which this thesis can be built. Next, we look at the sentiment models used in this paper.

2.2 Sentiment Analysis Models

In 2019, Delvin et al. [10] from Google AI presented a new model called BERT (Bidirectional Encoder Representations from Transformers). As the name suggests the authors try to pre-train deep bidirectional representations allowing for richer context-aware embeddings. BERT was released in different sizes and was designed to be fine-tuned with just one additional layer, paving ways to create state-of-the-art models for any domain. This paper set a precedent for subsequent work in deep learning models. The next two models are built on top of the work done in this paper.

2.2.1 FinBERT

Liu et al. [21] were the first to create a domain-specific BERT by pre-training and fine-tuning it on financial data, this was called FinBERT. This was pre-trained on six self-supervised tasks and fine-tuned on three financial text-mining tasks. The results were quite evident that FinBERT had a clear advantage over Vanilla BERT and BERT-task, even though BERT-task was pre-trained on a financial classification dataset. The creation of this model allowed many more papers to easily utilise FinBERT for financial sentiment analysis. One such paper by Jiang and Zeng (2023) [18] explores the real-world application of FinBERT in a stock news dataset focusing on extracting sentiments that could be influential in predicting stock trends. The authors used LSTM on the stock price data for financial time series prediction and observed that by integrating FinBERT sentiment scores into the dataset they were able to outperform pure LSTM. The authors hypothesize that financial sentiment from news and social media should have a strong correlation with market movements based on the Efficient Market Hypothesis. Halder (2022) [12] also compared FinBERT-LSTM combination with Multilayer Perceptron (MLP), a simple feed-forward neural network, and the results are conclusive that adding FinBERT sentiment score improves the price prediction in a model as it provides more market information than just the price data of stock.

2.2.2 RoBERTa

Researchers from the University of Washington along with the Facebook AI team proposed RoBERTa [20] stating that BERT was "significantly undertrained". BERT is improved by implementing four modifications: (1) training on a larger dataset and for longer periods, (2) eliminating next sentence prediction, (3) training on increased sequence length, and (4) dynamically altering the masking pattern. These changes allowed the researcher to achieve state-of-the-art results in varying evaluation metrics. A study by Alissa and Alzoubi (2022) [1] deployed a RoBERTa fine-tuned on the financial PhraseBank from Kaggle. They were able to confirm higher scores than BERTWEET-base in accuracy, precision and recall. They concluded that fine-tuning RoBERTa with a financial dataset can be effective in classifying financial texts.

2.2.3 VADER

VADER or Valence Aware Dictionary for sEntiment Reasoning was developed by Hutto and Gilbert (2014) [15] to better capture the sentiments of social media texts. The authors believed that the new form of content such as micro-blogging as seen in Twitter cannot provide enough context for sentiment analysis tools to effectively capture the sentiment score of the text. The model was pre-trained with human annotators which helped the author create a "gold standard" of lexicons for social media content. The authors identified generalised heuristics in social media, impacting the sentiment intensity: (1) Punctuation, (2) Capitalization, (3) Degree modifiers, (4) Contrastive conjunction e.g. "but", and (5) Negated sentences. The paper also states that VADER is domain agnostic, meaning the model does not need extensive training but can perform on a broad range of domains. The parameters used by VADER are also publicly available allowing for a better understanding of the model. The model is also very quick and does not sacrifice accuracy in the process. Pano and Kashef (2020) [23] implemented VADER to study the sentiments from Twitter relating to Bitcoin during Covid-19. They observed that VADER scores have a short-term correlation with Bitcoin prices. The authors try different pre-processing methods for VADER totalling up to 13 variations. Since VADER can capture punctuation and capitalisation, cleaning tweets of their tweet syntax and splitting the sentence showed higher correlations to Bitcoin prices.

2.2.4 GPT 3

In 2015 a few of the industry's big names including Sam Altman, Elon Musk, and Peter Thiel, got together to create OpenAI, with the aim to be the first to create Artificial General Intelligence (AGI) (Dale, 2021) [9]. OpenAI announced GPT (Generative Pre-trained Transformer) in 2018, which was followed by GPT-2 in February 2019. GPT-2 had 1.5 billion parameters and showcased improvements in various NLP tasks. Concerns about potential misuse initially led OpenAI to withhold the full model, releasing only smaller versions. And finally, in 2020, OpenAI released its third generation of GPT. Brown et al. (2020) [2] presented a language model with 175 billion parameters, 10x more than the previous iteration. The model was trained with Common Crawl Dataset consisting of nearly a trillion words. After filtering the team had 570GB of compressed plain text to train on. The authors also added some high-quality reference corpora to increase the diversity in the training data.

Since ChatGPT's release, there has been a lot of hype and fear around this model as it was able to produce very coherent answers based on the questions. Kheiri and Karima (2023) [19] created SentimentGPT, trying to capture advanced sentiment analysis using GPT-3. The authors published promising results across different linguistic nuances in Emojis, Slang, Hashtags, Negation and Sarcasm, Mixed sentiments, Cultural context, and Modern Abbreviations. This fine-tuned model was able to outperform RoBERTa in these nuances. Wang et al. (2021) [27] also studied how GPT-3 can help in labelling data at a lower cost and achieve comparable performance with human-labelled data. The data also suggested that utilising GPT-3 labels in your model can achieve better performance than raw GPT. Hu et al. (2023) [14] compare the GPT-3 and FinBERT on financial statements, and although GPT-3 showcased high competence in the task, it was easily outperformed by FinBERT in financial sentiment analysis. The paper also observes that FinBERT still outperforms GPT-3 in determining the market reaction. Zhang et al. (2023) [28] created Instruct-FinGPT, which trains the LLMs on a subset of labelled financial datasets to achieve higher accuracy over FinBERT on financial sentiment analysis. This shows that instructing GPT-3 on a specific task improves its performance.

2.3 Financial Price Predictions

Next, we look at the literature for the price prediction models.

2.3.1 LSTM

As seen in the previously mentioned work, LSTM is a very popular method to predict the price of the stock [3, 12, 18]. LSTM was introduced by Hochreiter and Schmidhuber (1997) [13] to overcome the vanishing gradient problem that plagued traditional RNNs. It can learn to bridge minimal time lags of more than 1,000 discrete time steps. With subsequent improvements over the years, it was widely adopted as a standard tool for sequence modelling in various domains. Various papers study different approaches to price prediction with LSTM [26, 16, 4]. LSTM has evolved to become a key tool in ML models for its ability to capture long-term dependencies makes it particularly apt for price prediction in financial markets. The incorporation of additional data sources further enriches the LSTM's effectiveness in this domain.

2.3.2 CNN

CNN is also a popular model to predict the price of stocks. Although it was associated with image recognition, the model has been adapted for price prediction as well. CNN was first proposed for time series analysis by Zhao et al. (2017) [29]. Recent studies that compare the different models for price predictions have made assertions that CNN architecture is the best at capturing change in trends Selvin et al. (2017) [26]. Using 1D convolution filters on time series data can capture localized patterns and incorporate additional features. Research across various asset classes employing CNN has shown comparable results.

CHAPTER 3

Methodology

3.1 Data Collection

The community r/wallstreetbets was created in January 2012 and has over 14.2 million members. It is ranked as the 44th largest community on Reddit. With so much engagement and discussion in the community, there is a vast pool of information related to the market at any given period.

A program was created to collect the headlines from Reddit using the Python Reddit API Wrapper (PRAW).[1] PRAW allows users to collect information from Reddit with ease and is built to follow Reddit's API rules. [2] The program creates an authorised Reddit instance allowing it to extract a larger number of headlines. It also utilised a custom session so as to not deal with any exception in HTTPS proxy. Once the PRAW object is initiated with the client_id, client_secret, user_agent, username, and password, it creates an authorised Reddit instance. This object allows the program to access any subreddit and its six different sort categories through their respective methods. The "hot" and "new" categories were selected and 100 headlines were extracted on the days executed. Both methods return a ListingGenerator, which needs to be iterated through one by one. These headlines were stored in a text file along with the date it was collected. Both of the datasets possess different characteristics, the hot section contains the more upvoted and more engaged topics, suggesting that the headlines have gained good

[1] https://praw.readthedocs.io/en/stable/index.html

[2] https://github.com/reddit-archive/reddit/wiki/API

traction. The new section has the potential to provide further information which could influence the market in the short term. The decision to not collect the data retroactively was due to the constraints of PRAW which does not allow you to capture information from the past. Only the headlines were collected and no data about the author was stored in order to maintain the anonymity of the users.

In July 2023, Reddit changed their terms and services [3] especially with regards to API access by implementing a premium tier for access to complete information from Reddit. Although metadata about the discussions would drastically improve the effectiveness in extracting trend movements, the changes in T&S and the waiting list for the developer platform [4] in place, caused the API server to be down for a few weeks thus restricting the quality and the amount of data collected. A total of 3714 unique sentences were captured between both datasets.

The stock information is collected from Yahoo! Finance through the 'yfinance' API. The Open, High, Low, Close, and Volume (OHLCV) data can be accessed using the download method. This method takes in the stock ticker and the time period as the input variable. The price data is collected for the past year as it is easier to match it with the dates on which the headlines were captured. The pct_change() function was used to calculate the returns of the closing price for 1 to 5 days.

3.2 Unsupervised Sentiment Classification

This research paper aims to compare the sentiment scores extracted from FinBERT, Distill-RoBERTa, VADER and GPT 3.5 without any extra fine-tuning to the model. The models chosen reflect the unique nature of the data source as it contains financial information but has an informal tone.

The sentiment models finetuned for finance namely FinBERT and Distill-RoBERTa used in this paper utilise the Hugging Face library to streamline the process of collecting the sentiment scores from the headline. Hugging Face is a huge repository of Transformers library accompanied by varying datasets and Tokenizers. Another major component of this library is the Pipeline provided which abstracts three components

[3] https://www.reddit.com/r/reddit/comments/12qwagm/an_update_regarding_reddits_api/

[4] https://tinyurl.com/3mfczpzk

namely Tokenizer, Model, and Post-processor[17]. This pipeline allows for faster text classification using fine-tuned models, allowing the user to pass their text without any training. Both the models iterate through the text and output a score of 0 to 1 and the label: positive, neutral, negative.

FinBERT is hosted by ProsusAI [5]. This is the most popular FinBERT model hosted, with over 1.3 million downloads last month. The distilled version of RoBERTa is hosted by Manuel Romero in Hugging Face[25]. This model was finetuned on a financial PhraseBank [6]. The PhraseBank was created by Malo et al. (2013) [22] and is the industry standard for fine-tuning models in financial literature. 16 human annotators were used to curate this dataset collected from financial news texts and company press releases.

The VADER model is part of the NLTK package and importing the SentimentIntensityAnalyzer allows the user to get the polarity score for any sentence passed. It maps the lexical features from the text to the dictionary of scores and assigns a value to each word in the sentence. The sentiment score is derived by summing up all the intensity scores provided for each word. Unlike the previous models, VADER outputs four different categories in its results, a score for each sentiment positive, neutral, negative and a compound score which is the overall sentiment score for a given sentence from -1 to 1

Zhang et al. 2023[28] trains the GPT-3 model to perform sentiment analysis by creating instruction prompts. Although GPT-3 is not a sentiment analyzer at its foundation, with the correct prompt, it may generate a label and a score for a given text. GPT-3.5 turbo was chosen from the available models offered by OpenAI, as it was the most cost-effective model. At 4K context, the model charged $0.0015/1K tokens for the inputs and $0.002/1K tokens for the output. A system message and a user message are provided as inputs to the model. The system message instructs the model to perform sentiment analysis on the user message with the following prompt: "Analyze the given text and classify it into: negative, or positive. Also, provide a sentiment score within the range of -1 to 1. Score values must be calculated with high precision with up to three decimal places. Your response format should be: sentiment, score e.g., ('negative, -0.145')." The results are then segregated and stored as two outputs for each sentence.

[5] https://huggingface.co/ProsusAI/finbert

[6] https://huggingface.co/datasets/financial_phrasebank

3.3 Parameters

Next, we look at the neural networks' parameters and how both LSTM and CNN are set up to predict the returns of the given stock. Certain pre-processing and input layers are shared by both models. The models use a multi-input architecture that encompasses both textual and numerical elements, and as a result, a Keras functional API was employed since it provides for greater flexibility in taking various inputs and outputs. Users can also connect layers to any other layers using the functional API[11].

The text is processed through a Tokenizer converting the string into a sequence of integers, where each integer is the index of a dictionary of the most frequent words[6]. The maximum number of words to be captured is set to 10,000 words. The text data is then passed through an Embedding layer which allows the ML model to capture the semantic meaning and context of the words[5]. This layer reads the token-encoded vocabulary and finds the embedding vector for each word-index. During training, the model learn these vectors.

The stock ticker and labels are one-hot encoded to transform them into a format that ML algorithms can utilise to improve the model's performance. One-hot encoding is used over label encoding to avoid the model misinterpreting the encoded values as having an ordinal connection, which is not the case for stock tickers and labels. The models are equipped with layers to prevent overfitting of the data and help make more generalised predictions. These layers include (1) BatchNormalization: normalizes the dataset to make the mean 0 and unit variance, (2) Dense layer with l2 regularisation: penalizes the large weights to filter out the noise, and (3) Dropout layer: randomly sets some input nodes to 0 so as to not specialise on a single node. The dataset is split into 80-20 for training and testing and a further 20% of the training data is used for validation.

3.3.1 LSTM

The text input is first embedded in a 128-dimensional space, and this is supplied into a 128-unit LSTM layer. The second LSTM layer, which has 32 units, is utilised for market data and processes aspects such as opening, closing, high, low, and volume. These layers capture the temporal trends in textual and market data, which may be necessary

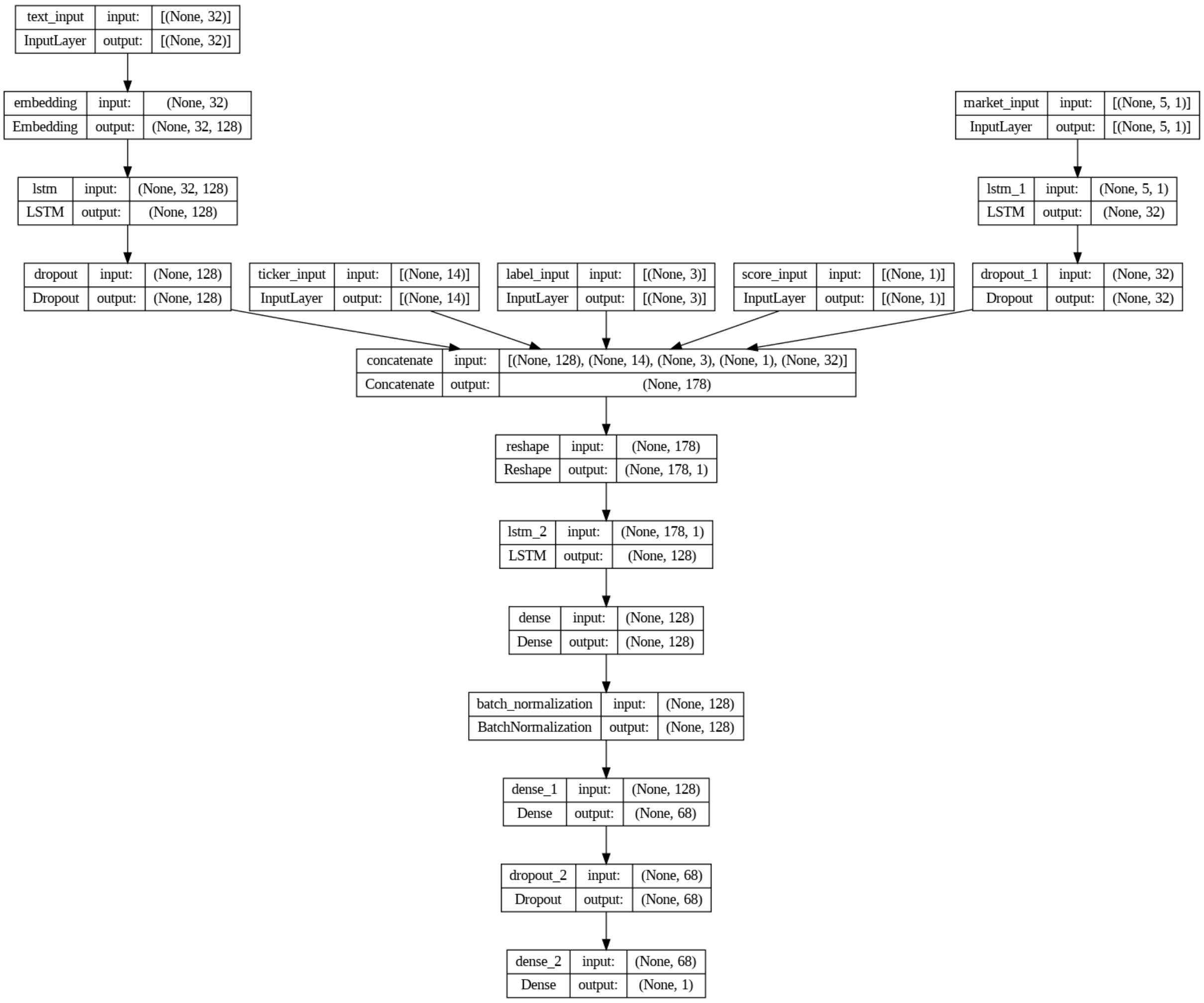

text_input | input: | [(None, 32)]
InputLayer | output: | [(None, 32)]
embedding | input: | (None, 32)
Embedding | output: | (None, 32, 128)
lstm | input: | (None, 32, 128)
LSTM | output: | (None, 128)
dropout | input: | (None, 128)
Dropout | output: | (None, 128)
ticker_input | input: | [(None, 14)]
InputLayer | output: | [(None, 14)]
label_input | input: | [(None, 3)]
InputLayer | output: | [(None, 3)]
score_input | input: | [(None, 1)]
InputLayer | output: | [(None, 1)]
market_input | input: | [(None, 5, 1)]
InputLayer | output: | [(None, 5, 1)]
lstm_1 | input: | (None, 5, 1)
LSTM | output: | (None, 32)
dropout_1 | input: | (None, 32)
Dropout | output: | (None, 32)
concatenate | input: | [(None, 128), (None, 14), (None, 3), (None, 1), (None, 32)]
Concatenate | output: | (None, 178)
reshape | input: | (None, 178)
Reshape | output: | (None, 178, 1)
lstm_2 | input: | (None, 178, 1)
LSTM | output: | (None, 128)
dense | input: | (None, 128)
Dense | output: | (None, 128)
batch_normalization | input: | (None, 128)
BatchNormalization | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 68)
dropout_2 | input: | (None, 68)
Dropout | output: | (None, 68)
dense_2 | input: | (None, 68)
Dense | output: | (None, 1)

Figure 3.1: LSTM model Architecture

for forecasting financial returns. The combined feature set is reshaped and processed through a third LSTM layer after the outputs from the text LSTM layer are incorporated with other features such as tickers, labels, scores, and the market LSTM layer. This layer has a distinctive function of capturing the connections between the various sets of inputs—text, market data, tickers, labels, and scores. The model has 1,508,361 trainable parameters.

The final layer uses a sigmoid activation function suitable for binary classification tasks. The model is compiled with Adam optimizer and Binary Cross-Entropy loss function. The model is evaluated on a test dataset to ascertain its predictive accuracy.

3.3.2 CNN

Conv1D layers are used in this model to handle both text and market data. Conv1D layers are designed to function on one-dimensional sequences, making them well-suited for dealing with sequential data. The text Conv1D layer is made up of 128 filters of size 3 and uses a ReLU (Rectified Linear Unit) activation function. The layer captures local dependencies by applying filters on a window of three embedded tokens at a time. It is followed by a max-pooling layer, which decreases the dimensionality of the previous convolutional layer's output. An additional 1D convolutional layer is utilised for market data, although it runs with 32 filters of size 2. The pooling layer downsamples by picking the maximum value from a 2-dimensional window. The model has 1,361,641 trainable parameters.

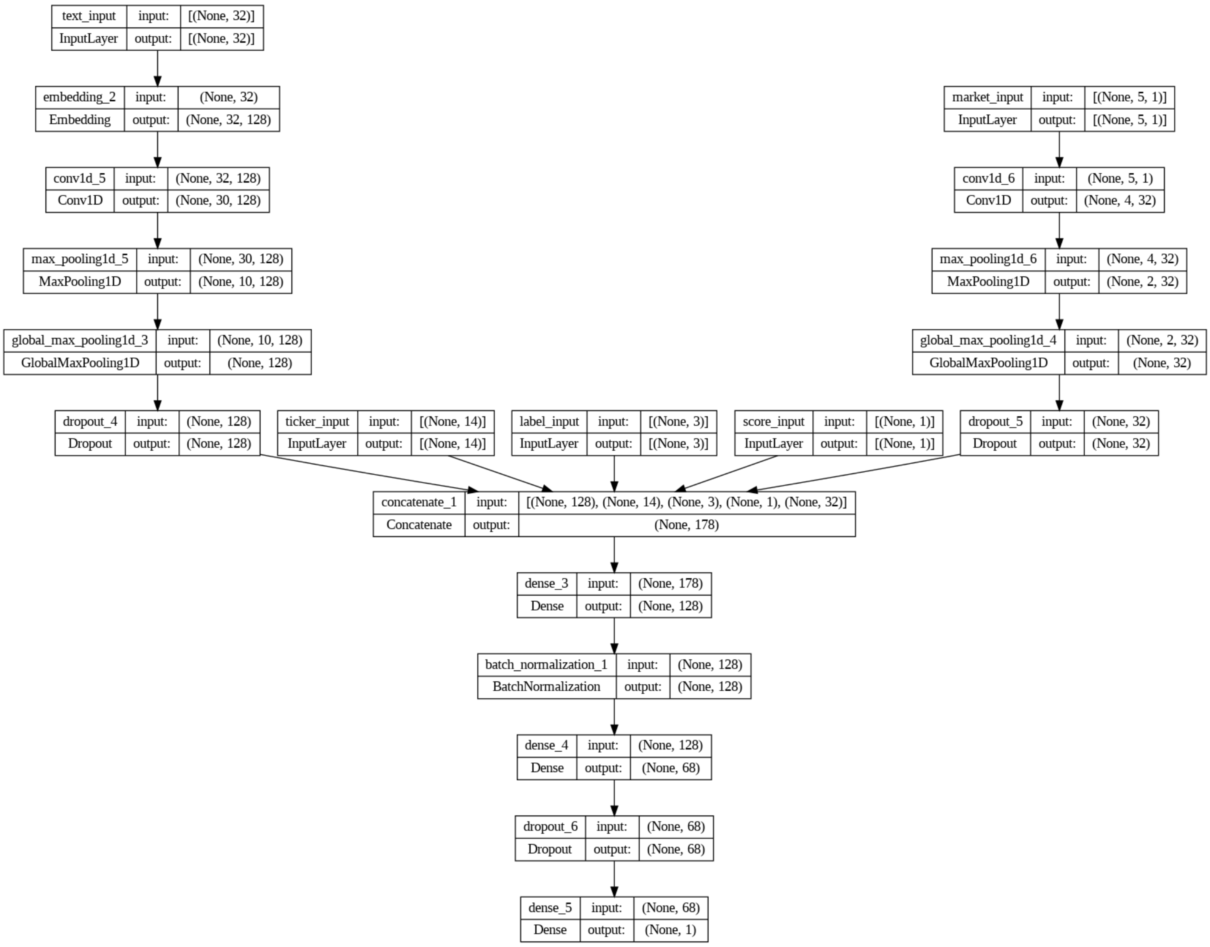

text_input | InputLayer | input: [(None, 32)] | output: [(None, 32)]
embedding_2 | Embedding | input: (None, 32) | output: (None, 32, 128)
conv1d_5 | Conv1D | input: (None, 32, 128) | output: (None, 30, 128)
max_pooling1d_5 | MaxPooling1D | input: (None, 30, 128) | output: (None, 10, 128)
global_max_pooling1d_3 | GlobalMaxPooling1D | input: (None, 10, 128) | output: (None, 128)
dropout_4 | Dropout | input: (None, 128) | output: (None, 128)
market_input | InputLayer | input: [(None, 5, 1)] | output: [(None, 5, 1)]
conv1d_6 | Conv1D | input: (None, 5, 1) | output: (None, 4, 32)
max_pooling1d_6 | MaxPooling1D | input: (None, 4, 32) | output: (None, 2, 32)
global_max_pooling1d_4 | GlobalMaxPooling1D | input: (None, 2, 32) | output: (None, 32)
dropout_5 | Dropout | input: (None, 32) | output: (None, 32)
ticker_input | InputLayer | input: [(None, 14)] | output: [(None, 14)]
label_input | InputLayer | input: [(None, 3)] | output: [(None, 3)]
score_input | InputLayer | input: [(None, 1)] | output: [(None, 1)]
concatenate_1 | Concatenate | input: [(None, 128), (None, 14), (None, 3), (None, 1), (None, 32)] | output: (None, 178)
dense_3 | Dense | input: (None, 178) | output: (None, 128)
batch_normalization_1 | BatchNormalization | input: (None, 128) | output: (None, 128)
dense_4 | Dense | input: (None, 128) | output: (None, 68)
dropout_6 | Dropout | input: (None, 68) | output: (None, 68)
dense_5 | Dense | input: (None, 68) | output: (None, 1)

Figure 3.2: CNN model Architecture

3.4 Architecture/Structure

The model is this project can be divided into four main sections starting from (1) the data collection from r/wallstreetbets using PRAW API for Python. The sentences are searched for any mention of a specific company and if none is found then the market index such as S&P 500 can be used. (2) The price data for these companies are downloaded from Yahoo! Finance and the returns are calculated. (3) The sentences are then passed to a sentiment analysis model to extract a numerical value. (4) The text and the scores combined with the stock data are fed as inputs for LSTM and CNN to predict the bi-directional move of the stock return. The models predict the returns for 5 days from when the text was captured, this is done to observe if the text has impacted the market throughout the week.

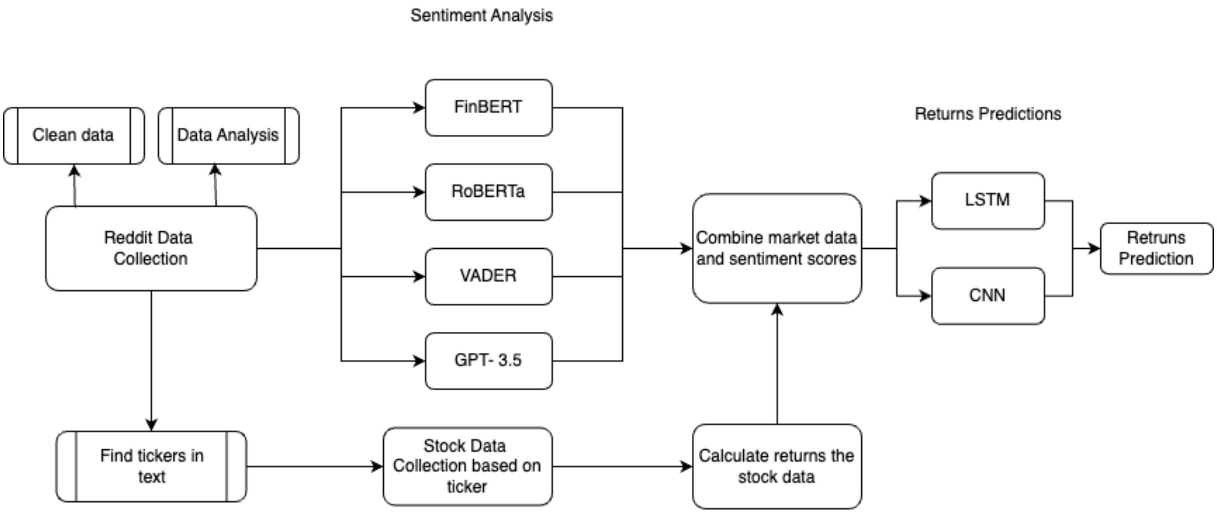


Figure 3.3: Architecture of the thesis model.

In the next section, we will look at data collected and how the sentiment scores vary across the models selected.

CHAPTER 4

Data

4.1 Data Pre-processing

The data gathered from Reddit is checked for duplicate lines and discarded. This information is then taken from the text file and saved in a pandas dataframe with two columns labelled 'date' and 'text'. The text was then checked with the NASDAQ stock screener[1] using regex to detect any mentions of firms on the NASDAQ stock market. For the purpose of this study, only tickers of companies with a market capitalisation of over 300 million were considered. A new column of these tickers was created to add a preceding "$" since it is common practice in the community e.g. "$AAPL". If the sentences do not have any company's ticker, then S&P is recorded instead to observe the market as a whole. These tickers were aggregated and those with 10 or more mentions (12 companies) were selected and the respective prices were downloaded. The remaining tickers were then changed to S&P which totalled 2492 sentences. The return on the closing price of these twelve companies and the S&P 500 are then stored in their respective CSV files.

A copy of the dataframe is created to convert the sentences to lowercase, this is done to improve the efficiency of the sentiment analysis. The lowercase sentences is only used for FinBERT and RoBERTa, since VADER has been trained on capitalized words [15] and GPT-3 can also derive sentiment scores based on the raw text. These

[1] https://www.nasdaq.com/market-activity/stocks/screener

Ticker	Mentions	Ticker	Mentions
NVDA	57	AMC	14
TSLA	57	AAPL	14
UBS	37	AMD	13
CVNA	36	RTX	12
PYPL	21	ES	11
DISH	15	MSFT	11
RIVN	14	HOOD	11

Table 4.1: List of companies with 10 or more mentions

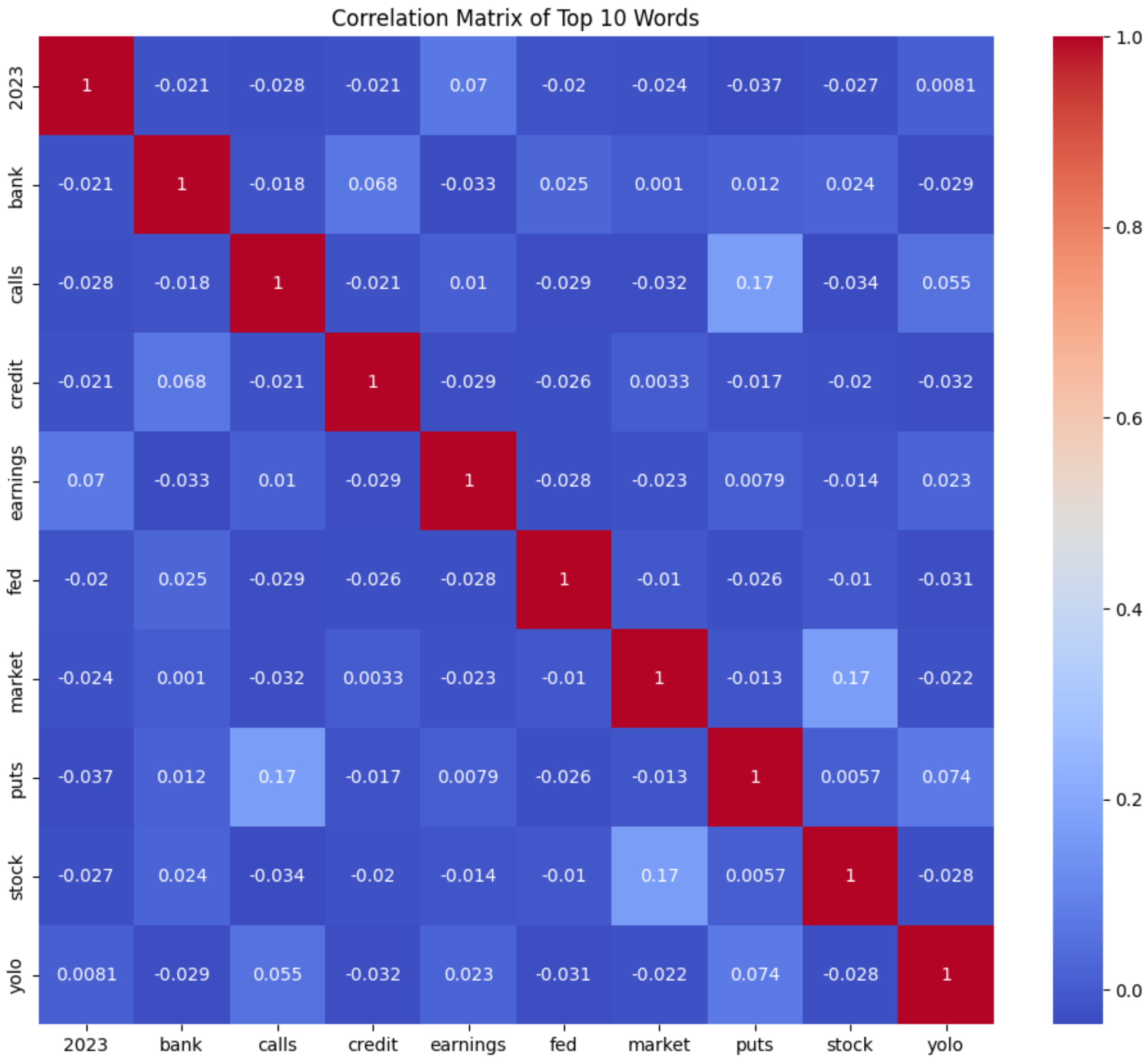
sentences are then passed to all four models, and with the exception of VADER the sentiment scores were returned with two components: the label and the score. The scores captured are then saved in a new dataframe with the following columns 'date', 'text', 'ticker', 'label', and 'score'. For VADER, the results output four components: 'neg', 'neu', 'pos', and 'compound' which are added to the dataframe containing the text and tickers. All these scores are saved in a CSV file for further work. A point to consider is that GPT-3 required the most time to execute since the rate limits on the API were in place, resulting in ServerUnavailable errors. The use of a try/catch and rate limiting resulted in an execution time of 75 minutes. For some sentences, the model could not derive a sentiment score and thus was labelled as an error, and these sentences were removed for further analysis. This is a crucial consideration when working with huge volumes of data with GPT-3. VADER was the fastest to obtain the sentiment scores.

The last step of the pre-processing is to match the price data and the returns calculated for each of the sentences. A dictionary of all the stock price data is created with the ticker as the key value. For each row in the text dataframe, the ticker and the date are separated and matched to the dictionary values. The OHLCV data is then combined with the sentiment scores from each model with all the calculated returns for up to 5 days. If the sentences were collected on a weekend then the price data of the following Monday is matched, since the market can only react to the the news on the next working day.

4.2 Data Analysis - Text

In this section, we explore the data collected and the statistics of the sentiment scores gathered from the models.

Using CountVectorizer [2] from scikit-learn's feature extraction we create a Correlation matrix of the top 10 words after removing the English stop words. This gives us an insight into the kind of language used in the community.



	2023	bank	calls	credit	earnings	fed	market	puts	stock	yolo
2023	1	-0.021	-0.028	-0.021	0.07	-0.02	-0.024	-0.037	-0.027	0.0081
bank	-0.021	1	-0.018	0.068	-0.033	0.025	0.001	0.012	0.024	-0.029
calls	-0.028	-0.018	1	-0.021	0.01	-0.029	-0.032	0.17	-0.034	0.055
credit	-0.021	0.068	-0.021	1	-0.029	-0.026	0.0033	-0.017	-0.02	-0.032
earnings	0.07	-0.033	0.01	-0.029	1	-0.028	-0.023	0.0079	-0.014	0.023
fed	-0.02	0.025	-0.029	-0.026	-0.028	1	-0.01	-0.026	-0.01	-0.031
market	-0.024	0.001	-0.032	0.0033	-0.023	-0.01	1	-0.013	0.17	-0.022
puts	-0.037	0.012	0.17	-0.017	0.0079	-0.026	-0.013	1	0.0057	0.074
stock	-0.027	0.024	-0.034	-0.02	-0.014	-0.01	0.17	0.0057	1	-0.028
yolo	0.0081	-0.029	0.055	-0.032	0.023	-0.031	-0.022	0.074	-0.028	1

Figure 4.1: Correlation Matrix of the top 10 words.

As we can see the words are not highly correlated, but the words aggregated correlate

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

to the real-world events during the collection range from March to August 2023. Four banks in the US have collapsed in the same period. Credit Suisse was one of 30 banks designated as globally systemically important but was bought by UBS since a complete collapse of Credit Suisse would have damaged the world's financial system. [24]. This is also reflected in the number of mentions of UBS tickers. We can see the timeline and the asset values of the four banks in the US at the time of collapse. This data is maintained by the Federal Deposit Insurance Corporation (FDIC) [3]

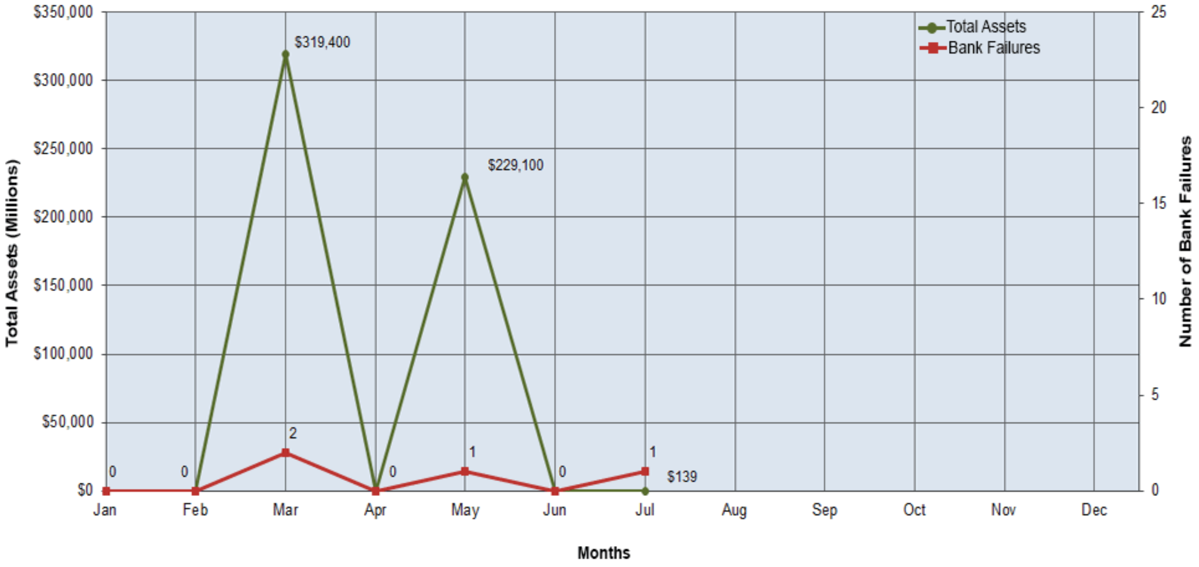


Figure 4.2: Timeline of bank failures in the US

Another macroeconomic event that occurred during this period was the sharp rise of interest rates by the Federal Reserve as it is traced by Trading Economics[4]. This is indicative in the correlation matrix as "fed" was the sixth most used word in the dataset. A manual look at the data also includes a lot of headlines with "JPOW" in them but was not captured by the CountVectorizer since it is an acronym by the community for Jerome H. Powell, the chair of the Federal Reserve [5]

[3] https://www.fdic.gov/bank/historical/bank/bfb2023.html

[4] https://tradingeconomics.com/united-states/interest-rate

[5] https://www.federalreserve.gov/aboutthefed/bios/board/powell.htm

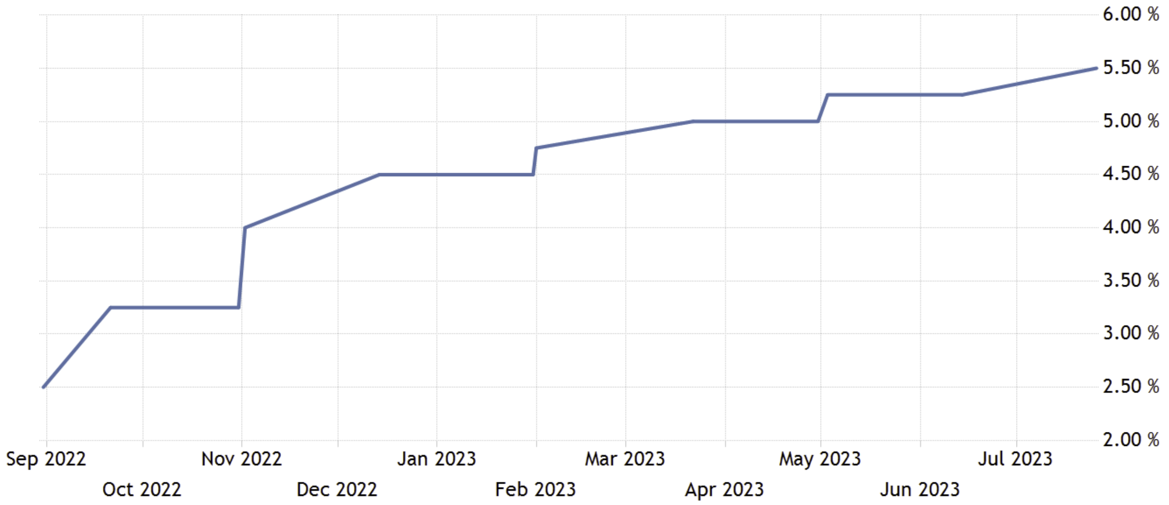


Figure 4.3: Interest Rate hike from Sept 2022 - August 2023

4.3 Data Analysis - Sentiment Analysis

Next, we will look at the sentiment scores collected from the different unsupervised sentiment models. This will allow us to understand how the different models score the same sentences. Both the 'hot' and the 'new' datasets were combined for this analysis as we are comparing the differences in models and not the dataset. Since FinBERT, RoBERTa, and GPT-3 have labels as an output we can make further analysis on the label, compared to VADER with only the numerical value of the scores.

First, we look at the overall sentiment distribution for the three models with the label.

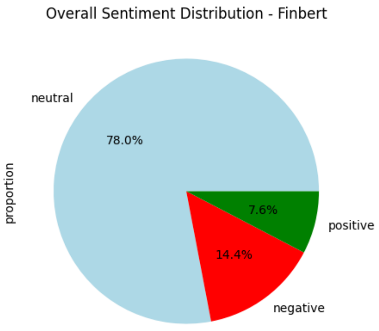


Figure 4.4: FinBERT labels

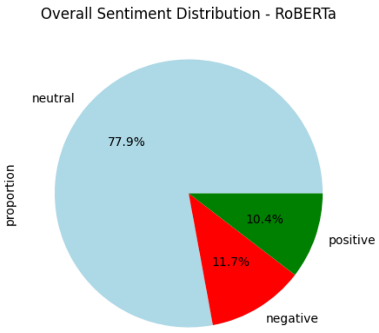


Figure 4.5: RoBERTa labels

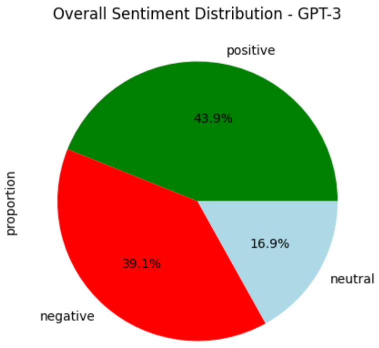


Figure 4.6: GPT-3 labels

Here we can see that FinBERT and RoBERTa follow a similar level of label distribution between positive, negative, and neutral, while GPT-3 has a much larger distribution of positive, and negative scores. When we look at the distribution of the scores, we see an entirely different picture.

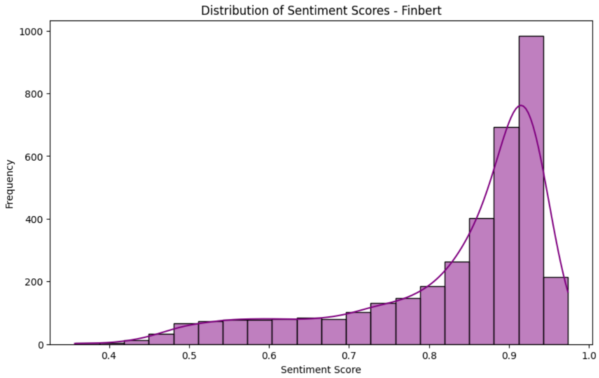


Figure 4.7: FinBERT score distribution

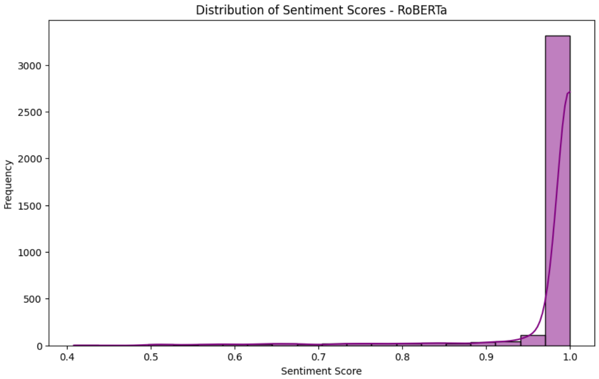


Figure 4.8: RoBERTa score distribution

Here we can observe that the scores are distributed differently for both the BERT-based models. FinBERT scores are mostly centred around 0.9, meaning a high value for the sentiments was captured. RoBERTa has most, if not all the scores between 0.9 and 1, meaning that all the scores of any sentiment have a high value. Both these models also do not store a negative number for the negative labelled sentiments, and thus all the scores lie between 0 and 1.

We can also see the distribution of the VADER and GPT-3 scores below.

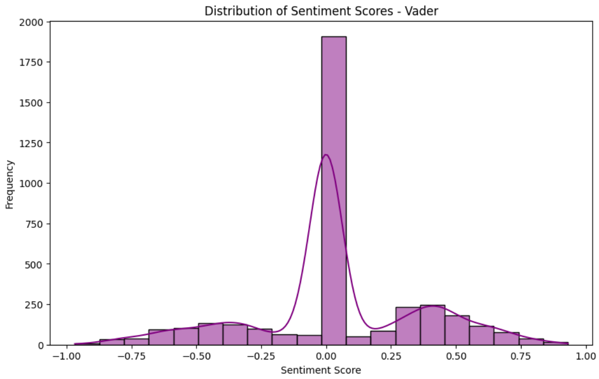


Figure 4.9: VADER score distribution

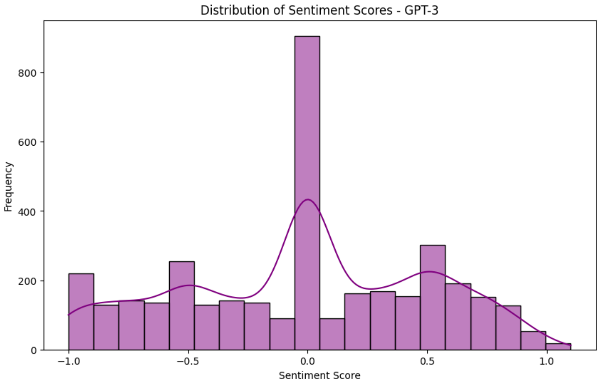


Figure 4.10: GPT-3 score distribution

GPT-3 was instructed to give negative values in case the label was negative, and thus the scores are distributed between -1 and 1, similar to the scores received from VADER. We can see an even distribution of scores except in the middle where there is a huge spike indicating a large number of sentiments with scores near 0. This means that even though GPT-3 had a large number of positive and negative scores compared to the other two models, the values of these scores were low. Comparing VADER with GPT-3 we can see that the average VADER score is even lower than of GPT-3.

Finally, we can look at how the 3 models assigned labels for the top 5 mentioned stocks. Blue is neutral, red is negative, and green is a positive label.

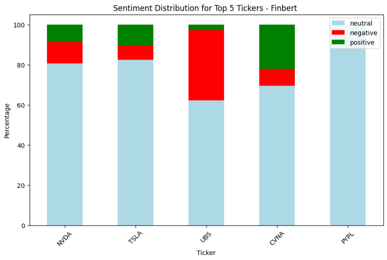


Figure 4.11: FinBERT Top 5

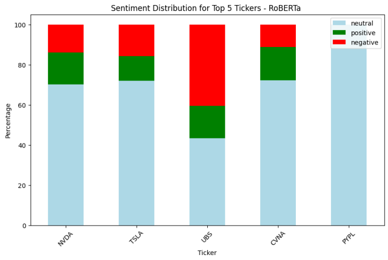


Figure 4.12: RoBERTa Top 5

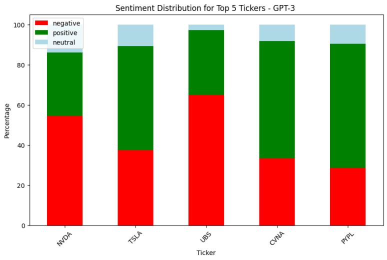


Figure 4.13: GPT-3 Top 5

We can observe some similarities between the BERT-based models, with UBS having a large number of negative sentiments. Although UBS acquired Credit Suisse, a lot of discussions revolving around Credit Suisse were attributed to UBS since it was the company acquiring it. Both the models did not have anything positive or negative for PYPL as we can see it is completely light blue. GPT-3 on the other hand, has a different scoring system and has large numbers of positive and negative scores. We can still see that UBS has the most number of negative scores in GPT-3, this is consistent with all the other models.

In the following chapter, we will explore the results of these sentiment scores paired with the financial data on returns prediction.

CHAPTER 5

Results

Bibliography

[1] Kefah Alissa and Omar Alzoubi. Financial sentiment analysis based on transformers and majority voting. In *2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA)*, page 1–4, Dec 2022.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. (arXiv:2005.14165), Jul 2020. arXiv:2005.14165 [cs].

[3] Caymen Carter. Stock price prediction using sentiment analysis and lstm. May 2022. Accepted: 2022-06-17T19:23:11Z.

[4] Wenjuan Cheng and Siyi Chen. Sentiment analysis of financial texts based on attention mechanism of finbert and bilstm. In *2021 International Conference on Computer Engineering and Application (ICCEA)*, page 73–78, Jun 2021.

[5] Saturn Cloud. Understanding embedding layers in keras: A comprehensive guide, Jul 2023.

[6] Saturn Cloud. Understanding the keras tokenizer method: A comprehensive guide for data scientists, Jul 2023.

[7] Emily Coppens. The correlation between reddit sentiment and the strongest- and weakest performing cryptocurrencies of 2021, 2021.

[8] Michele Costola, Matteo Iacopini, and Carlo R. M. A. Santagiustina. On the "mementum" of meme stocks. *Economics Letters*, 207:110021, Oct 2021.

[9] Robert Dale. Gpt-3: What's it good for? *Natural Language Engineering*, 27(1):113–118, Jan 2021.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. (arXiv:1810.04805), May 2019.

[11] Rahmat Faisal. Keras model sequential api vs functional api, Nov 2020.

[12] Shayan Halder. Finbert-lstm: Deep learning based stock price prediction using news sentiment analysis. (arXiv:2211.07392), Nov 2022. arXiv:2211.07392 [cs, q-fin].

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, Dec 1997.

[14] Nan Hu, Peng Liang, and Xu Yang. Whetting all your appetites for financial tasks with one meal from gpt? a comparison of gpt, finbert, and dictionaries in evaluating sentiment analysis. (4426455), Jul 2023.

[15] C. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(11):216–225, May 2014.

[16] Abid Inamdar, Aarti Bhagtani, Suraj Bhatt, and Pooja M. Shetty. Predicting cryptocurrency value using sentiment analysis. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, page 932–934, May 2019.

[17] Shashank Mohan Jain. *Hugging Face*, page 51–67. Apress, Berkeley, CA, 2022.

[18] Tingsong Jiang and Andy Zeng. Financial sentiment analysis using finbert with application in predicting stock movement. (arXiv:2306.02136), Jun 2023. arXiv:2306.02136 [q-fin].

[19] Kiana Kheiri and Hamid Karimi. Sentimentgpt: Exploiting gpt for advanced sentiment analysis and its departure from current machine learning. (arXiv:2307.10234), Jul 2023. arXiv:2307.10234 [cs].

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. (arXiv:1907.11692), Jul 2019. arXiv:1907.11692 [cs].

[21] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. Finbert: A pre-trained financial language representation model for financial text mining. volume 5, page 4513–4519, Jul 2020.

[22] Pekka Malo, Ankur Sinha, Pyry Takala, Pekka Korhonen, and Jyrki Wallenius. Good debt or bad debt: Detecting semantic orientations in economic texts. (arXiv:1307.5336), Jul 2013. arXiv:1307.5336 [cs, q-fin].

[23] Toni Pano and Rasha Kashef. A complete vader-based sentiment analysis of bitcoin (btc) tweets during the era of covid-19. *Big Data and Cognitive Computing*, 4(44):33, Dec 2020.

[24] NATHAN REIFF. What happened at credit suisse, and why did it collapse?

[25] Manuel Romero, Aug 2023.

[26] Sreelekshmy Selvin, R Vinayakumar, E. A Gopalakrishnan, Vijay Krishna Menon, and K. P. Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, page 1643–1647, Sep 2017.

[27] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? gpt-3 can help. (arXiv:2108.13487), Aug 2021. arXiv:2108.13487 [cs].

[28] Boyu Zhang, Hongyang Yang, and Xiao-Yang Liu. Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models. (arXiv:2306.12659), Jun 2023. arXiv:2306.12659 [cs, q-fin].

[29] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, Feb 2017.

[30] Yanzhao Zou and Dorien Herremans. A multimodal model with twitter finbert embeddings for extreme price movement prediction of bitcoin, 2022.

APPENDIX A

A Long Proof

The code takes a inspiration from Kritanjali Jain's code on Kaggle[1] and built upon the data pre-processing techniques shown there.

```
print("Hello World")
```

[1] https://www.kaggle.com/code/kritanjalijain/twitter-sentiment-analysis-lstm

Appendix B

Another Appendix

Text goes here