## Abstract

## Introduction

Modern society has been able to access information, communicate ideas, and become part of a community due to the advent of the internet. Online discussion boards are playing a critical role by providing a platform where people can do so. Those discussion boards are also used by a variety of people to talk about the stock market and discuss trading strategies. Recently, the Reddit forum WallStreetBets has become one of the most well-known and influential investing online-forums.

Even though the Reddit subforum was created in 2012 already, it received the majority of its media exposure in 2021 as a result of a short-squeeze of the GameStop (GME) stock, which drove the stock price up hundreds of percentage points. However, it was not the rapid price appreciation that amazed market participants. Instead, it was the unprecedented decentralized and coordinated buying of Gamestop shares by members of the WallStreetBets community that attracted attention (Anand & Pathak, 2021)

Organizing the mass-coordinated buying of stock, however, requires that enough participants share the same sentiment. According to several studies, social media sentiment has a particularly strong impact on uninformed traders (Danbolt, Siganos, & Vagenas-Nanos, 2015). Write a bit more about retail investors.

Interestingly, finance scholars did not consider Reddit as a platform capable of having such a significant impact on the financial markets. As a result, the site has been neglected in their research (Long, Lucey, & Yarovaya, 2021).

Hence, this thesis will try to answer the following Research Question:

*How can sentiment analysis best be performed on the WallStreetBets Reddit-forum?*

*Can sentiment analysis of the WallStreetBets Reddit-forum be used to predict daily changes in the stock price of Gamestop?*

To answer this research question several fields in the domains of machine learning and finance need to be explored. To begin, it must be determined how the discussions about the Gamestop stock on WallStreetBets should be handled to serve as suggestive input features for sentiment analysis. One of the challenges, is the heavy use of peculiar terminology and domain-specific phrases on the WallStreetBets forum, as well as many novel words (Anand & Pathak, WallStreetBets Against Wall Street: The Role of Reddit in the GameStop Short Squeeze, 2021). According to recent research, sentiment lexicons and corpora with a focus on a certain domain produce superior sentiment analysis results compared to a general-purpose sentiment lexicon or corpora (Park, Lee, & Moon, 2015). Furthermore, the text data needs to be cleaned and pre-processed in order to be accurately processed by a machine learning algorithm (Jemai, Hayouni, & Baccar, 2021). As a result, the following sub-research question was formed:

*How can the domain-specific language of the Reddit forum WallStreetBets best be incorporated into sentiment analysis?*

Subsequently, the machine learning models can be trained to perform sentiment analysis. However, each machine learning algorithm has its own idiosyncrasies and assumptions, and no single classifier works optimally in all possible scenarios. Hence, it is a good idea to evaluate the results and performance of different machine learning algorithms. As a result, the best model with a given set of hyperparameters can be selected to solve a particular problem (Raschka & Mirjalili, 2019). This thesis will explore traditional machine learning methods such as Naïve Bayes (NB) and Support Vector Machines (SVMs), as well as deep learning methods like Long Short Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT). Due to the high dimensionality of textual data, deep learning methods have shown to outperform traditional machine learning techniques in recent research. That can be explained by the ability of deep learning methods to automatically learn the most important features, whereas traditional methods may suffer from the curse of dimensionality (Fu, Yang, Li, Fang, & Wang, 2018).

As was mentioned earlier, however, no classifier works best on all scenarios which is why the next research question needs to be answered:

*Which sentiment analysis approach performs best on predefined key performance indicators?*

The impact of sentiment on stock prices has gained attention by researchers in recent years. For example, it is shown that social media sentiment can have a direct effect of how market participants perceive a company, which can lead to changes in the stock price of companies. This is especially true for smaller firms with low analyst coverage (Feng & Johansson, 2019). Other researchers show that sentiment obtained from Twitter can be used to predict returns of a broader stock market index (Gu & Kurov, 2020). In other research the emotions of discussions on WallStreetBets are studied by performing sentiment analysis. The research suggests that only some emotions demonstrate a significant impact on one-minute returns of the Gamestop shares (Long, Lucey, & Yarovaya, 2021).

There are several models that have shown strong results with regards to forecasting time-series. The most prevalent in the financial industry is Auto Regressive Integrated Moving Average (ARIMA) which captures temporal structures in time-series data. However, it is not designed to include other features, such as sentiment. This is why this thesis will also compare other models such as LSTMS and XGBoost, which have also demonstrated strong predictive capabilities with regards to time-series data. As a result, the final sub-research question will explore the final part of the main research question.

*Which machine learning algorithm delivers the best predictive performance for changes in daily stock prices of Gamestop based on the sentiment analysis performed earlier?*

## Literature

Talk about literature. Relevant, use a lot, processed with new insights provided. Clearly connected to RQ.

Gauging sentiment of online forums to predict movements in stock prices has been a research subject for many years now. Das & Chen (2007) did a study on the Yahoo! message board, which was amongst the first ones on the internet for investors to exchange ideas. In their paper, they show that the relationship of stock price to sentiment is significant and that market activity is related to activity of the message boards. Other researchers, such as (Lyócsa, Baumöhl, & Vŷrost, 2021) also showed that as the discussion volume on WallStreetBets increased, the volatility of certain stocks got amplified. (Umar, Gubareva, Yousaf, & Ali, 2021) also found that sentiment of investors on WallStreetBets affected the returns of the Gamestop stock. However, they also show that other features such as the put-call ratio and the short-sale volume had a strong impact on the stock price.

Long, Lucey, & Yarovaya (2021) tried to uncover the impact of specific emotions such as *“Angry, Fear, Happy, Sad and Surprise”* from the comments on WallStreetBets discussions on intraday changes of the stock price of the affected stock. While they conclude that the tone as well as the number of comments have an impact on the stock price, they show that the number of comments is not directly related to sentiment. Additionally, they argue it is the number of comments that is posted within an hour that has the biggest effect on one minute changes in the stock price. Furthermore, the paper shows that the emotions *Sad, Anger* and *Surprise* have a significant impact on the gamestop 1-minute stock price. The *Happy* sentiment does not show a significant impact on 1-minute price changes, however, a causality test showed a link between the *Happy* sentiment and intraday returns of the GME stock. In addition, the paper shows, that sentiment only impacts intraday returns if a thread has more than 2000 comments. Hence, the authors confirm that Reddit sentiment has an impact on the stock market. They also argue that any asset that is targeted by a large crowd from wallstreetbets can become a subject of excessive volatility, without being driven by any fundamental reasons.

However, since the WallStreetBets ‘meme-stock movement’ is a relatively recent phenomenon, there is very little research on the impact of WallStreetBets on individual stocks, especially with regards to sentiment analysis. Additionally, of all the published research none account for the domain-specific language used on the forum. Because of the frequent usage of terminology that is specific to WallStreetBets, this can lead to incorrect conclusions.

Of course, this also applies to research in other fields, which usually also use a general-purpose sentiment lexicon, because of the cost associated with building a domain-specific one. However, it has been demonstrated that using a domain-specific knowledge base results in more accurate sentiment analysis (Park, Lee, Moon – 1-s2.0-S0….).

It is argued that there is no general-purpose sentiment lexicon that can be optimally applied on all domains. In different domains, some terms can have completely different meanings. A good example is the word “unpredictable”, which would have negative sentiment for electronics but can be a positive label for movies. It has been demonstrated that by adapting sentiment lexicons to a certain domain performance for sentiment classification can be enhanced (Lu, Castellanos, Dayal, & Zhai, 2011). This adapted lexicon can then be searched to find and score the sentiment of a specific word (Asghar, 2014).

While lexicon-based methods have found widespread adoption, mainly due to their simplicity, more advanced machine learning methods have also shown strong performance (tfidf/Wang2020\_Article\_...pdf). For this reason other research deviates from the aforementioned lexicon-based approaches. Instead, they examine how deep learning methods can be used to automatically detect and identify domain-specific words from sentences. By doing so it is assumed that the algorithm can not only detect whether domain-specific words are used (sentence-level detection), but also to identify the exact position of the term in the sentence (token-level identification). Hence, it is possible to detect new meanings of words in an already existing corpus. In addition, this approach also allows to classify novel words, that do not yet exist in a dictionary. This can be achieved by having models that formulate domain-specific word detection as a sequence-labelling task. Furthermore, novel domain-specific words can be learned by understanding the contextual structure of a sentence (Pei, Sun, & Xu, 2019). Those out-of-vocabulary tokens can be learned in the hidden layers of LSTMs (Hochreiter & Schmidhuber, 1997). To further optimize performance, models can be improved, by applying a character-based convolutional neural network to encode the spelling of words (Pei, Sun, & Xu, 2019). Even though the literature suggests many innovative ways to enhance model performance by a few percentage points, the biggest benefits seem to come from high quality input data in the form of domain-specific labeled data.

Creating a domain-specific annotated corpus to train machine learning models, however, is not without its own challenges. For example, working with multiple human annotators can lead to discrepancies in the annotation results (Kim, Ohta, & Tsujii, 2008). Additionally, it is hard to estimate the total annotation cost and can depend on whether the annotator is capable of understanding the language for the task at hand (Arora, Nyberg, & Rosé, 2009). Additionally, labelling an entire dataset incurs extremely high costs, which can be avoided. With the support of an Active Learner, a complete domain-specific corpus with its respective labels can be created using only partial annotations (Park, Lee, Moon).

One of the key concepts of Active Learners is that if a machine learning algorithm is allowed to choose the data from which it learns, it will achieve higher accuracy with less training data. If a considerable amount of the data is unlabeled, this is especially desirable. As a result, the total cost of annotation can be reduced drastically. Research shows that the total number of manual annotations can be reduced by 80% when using an Active Learner instead of randomly selecting data to label (Active Learning and Cost of annotation – Jason Baldridge and Miles Osborne).

If data is manually annotated at random, the annotator will invest a lot of time into labeling irrelevant instances. This may incur costs which could be avoided with an Active Learner. It is argued that Passive Learning, or randomly selecting instances to be labeled by an annotator, is especially costly if the class distribution of the data is imbalanced or if there are many very similar documents. For example, if a specific feature set appears on only 1% of instances, the annotator would have to label 1000 documents to cover the feature set on 10 relevant documents. When it comes to document similarity, large clusters of very similar documents might be identifiable. Because features may be barely distinctable, the annotator might spend a lot of effort labeling uninformative instances when selecting them random. An Active Learner, on the other hand, suggests which instances the annotator should label. Those instances can be determined on various quantitative metrics (active-learning-approaches-for-labelling-text, Miller, Linder, Mebane).

Write about sentiment classifiers.

## Methodology

Clear explanation of methods with connections drawn to other methods, appropriate robustness checks of assumptions, consistent, transparent, and correct.

### Data

While Reddit does offer an official API, the API is most useful for streaming data. There are some strict limitations on loading historical data. As a result, the official API is not the best choice for this thesis. However, pushshift.io provides a solution for the strict limits. Pushshift is maintained by the /r/datasets mod team. The FAQ on the pushshift subreddit states, that pushshift data is best used to:

* Analyze large quantities of Reddit data
* Grab data for a specific date range in the past
* Search for comments
* Aggregate data

Pushshift copies data from Reddit at the time it is posted. Since Pushshift uses the document-based database Elastic, it is extremely fast to query data. However, currently Pushshift does not regularly update certain metadata, such as scores, edits to a submission’s text or comments. Hence, there might be some minor inconsistencies of what is shown on Reddit and what is in the database. The scores, for example can easily be accessed via the official reddit API and, if needed, joined with the data obtained from Pushshift. Based on the data verification I performed, the number of comments only deviates by a marginally small amount. It is hypothesized that the small difference can be explained by forum moderators deleting spam. Those spam comments are assumed to not have a big impact the thesis anyways, which is why the small difference in the number of comments do not need to be addressed.

To access the Pushshift API, I used an API wrapper called PMAW. Since requests are I/O-bound, PMAW is multithreaded. Hence requests can be run asynchronously which allows the data to be loaded much faster.

When making the API request, the most important parameters are the following:

* subreddit: Name of the subreddit
* q: The search term based on which the subreddit is queried
* before: The starting date of the query
* after: The end date of the query

For this thesis all Gamestop (GME) related posts between January 1st, 2020 and October 26th, 2021 were requested from the subreddit WallStreetBets. The query returns 89 columns. Most of which, however, can be dropped since they either aren’t useful or contain no data. The most important columns are the number of comments, the title of the post and the content of the post. Emoticons are also included in the content text. In total 179,544 posts were obtained.

Out of the 179,544 posts, 10% or 17,955 were manually labeled as bearish, neutral or bullish.

### Data Preprocessing

The research by Jemai, Hazouni, and Baccar (2021) presents a system for structuring a sentiment analysis project. The data collection phase is the first step, where textual data is obtained from a source. The data is then cleaned in the second step, the data pre-processing phase. To do so, several actions need to be performed. Data tokenization is one of the actions. This is a common technique in which a large body of text is broken down into multiple sentences, each of which is then broken down into a list of words. Stop words such as *is, the, a* and other common words are also removed during the pre-processing phase. In addition, special characters such as @ and urls should also be removed. It is also suggested that the text is converted to lowercase. As the final step, the research proposes lemmatization. By doing so, the structure of a word is analyzed and converted to its normalized form.

Since it is shown that having data with emoticons leads to more accurate results than data without emoticons, emoticons are not removed from the corpus (Parveen & Pandey, 2016).

Eventually, term frequency-inverse document frequency (tf-idf), was applied on the text corpus. Using this representation allows the extraction of the most descriptive terms in a document and easy computations. However, it fails to capture semantics and word embeddings. For computational reasons, only words that occur at least five times were included in the tf-idf representation.

### The Case for a Supervised Method over an Unsupervised Method

Since the data obtained from Reddit is unlabeled, it cannot be fed into a supervised machine learning algorithms. However, many promising sentiment analysis methods rely on labeled data (Sazzed & Jayarathna, 2021). One approach to label data is using unsupervised machine learning models. Unsupervised models are commonly applied in Natural Language Processing and text classification (Unsupervised.pdf, Jung Lee). However, unsupervised models are a better choice for uncovering hidden patterns in a dataset, especially without any a priori knowledge of the structure of the data. As a result, unsupervised models excel at summarizing or exploration a large text corpus.

For the case at hand, a t-Distributed Stochastic Neighbor embedding (t-sne) algorithm was applied on the data to extract similarity features and project them onto a lower dimension (Devassy & George, 2020). As can be seen in the visualization below, admittedly at a low dimension, the majority of the data do not belong to any particular cluster.

A picture containing chart

Description automatically generated

*Red: Negative Sentiment; Blue: Positive Sentiment; Yellow: Neutral Sentiment*

*Based on manually labeled seed data that was fed into the AL*

Even though there are some approaches to clustering high dimensional data, it generally is difficult to do so. One of the explanations for that is the increased sparsity and the difficulty to distinguish between the distances of specific instances (Tomašev, Radovanović, Mladenić, & Ivanović, 2014).

If there are labeled instances, supervised learning methods are more applicable than an unsupervised method. One of the major disadvantages of supervised models, however, is the cost associated with manually labelling the data (active-learning-approaches-for-labelling-text, Miller, Linder, Mebane). This thesis proposes a methodology of creating a labeled dataset for the fraction of the total annotation cost. As a result, a domain-specific labeled text corpus is created, which can be used to compare the performance of different supervised machine learning algorithms.

The proposed methodology is an Active Learner. With its support, a complete domain-specific corpus can be labeled while only relying on partial annotations (Park, Lee, Moon).

### Active Learner Workflow

The illustrated workflow provides an overview of how an Active Learner works. To begin with, cleaned and pre-processed data needs to be available that can be used by the Active Learner. Furthermore, the Active Learner can also be trained with some initial training data, which is also referred to as the seed. By using clustering algorithms, the *seed* data can be selected methodologically, which allows the Active Learner to achieve higher accuracy faster when compared to randomly picking the initial seed data (Kang, Ryu, & Kwon, 2004). All the unlabeled instances will become the *pool* data, which need to be labeled. The seed data is fed into the Active Learner and trains an estimator, which needs to be defined when creating the Active Learner.

In addition, a query strategy needs to be defined, based on which the Active Learner queries new instances from the aforementioned pool. A query strategy evaluates the informativeness of unlabeled samples. Common strategies are *uncertainty sampling, query-by-committee, expected model change, expected error reduction and variance reduction*.

While each strategy has its own intricacies, all essentially try to find instances that are hard for the model to classify and hence might benefit from annotation.

After the query function selected instances from the pool, an oracle needs to label those. An oracle is normally at least one human with knowledge on how to annotate the data at hand (Settles, 2009). However, human annotators are oftentimes inconsistent and the result may vary from person to person (Ali & Gayar, 2019). Once the new instances are labeled, those instances need to be removed from the pool, since they are now part of the labeled data. The Active Learner then needs to be taught the new instances, which he can use to adjust the model. After each iteration, the results can be evaluated. A common performance measure for Active Learners is accuracy.

If a predefined stopping criterion is not yet met, the query strategy selects more instances from the pool and repeats the process. If the stopping criterion is met, the process ends (Lu, Henchion, & Namee, 2019).

Diagram

Description automatically generated

Visualized Workflow of an Active Learner. Created with lucid.app

### Active Learner Implementation

To implement an Active Learner the modAL package was used. modAL was designed with modularity, flexibility and extensibility as high priorities (Danka & Horvath, 2018). The estimator defined in the Active Learner object is a Support Vector Machine (SVM). A SVM was chosen because of its strong generalization performance (Alves, Baptista, Firmino, de Oliveira, & de Paiva, 2014). Additionally, SVMs can be used to solve both regression and classification problems. For the case at hand, the algorithm needs solve a classification problem, by optimally separating the data between bearish, neutral and bullish instances. Classification is done by finding a hyper-plane with the biggest margin, meaning it looks for the greatest distance to the nearest sample points (Jemai, Hayouni, & Baccar, 2021). SVMs use spatial transformations, commonly known as kernel functions, to fit the hyperplane. Kernels can be linear, RBF or others. The radial basis function (rbf) kernel is best used for non-linear problems and is a general-purpose kernel that is often used in pattern recognition problems. The linear kernel, on the other hand, is typically used when there are only two classes present. A good example for that might be positive and negative sentiment (Alves, Baptista, Firmino, de Oliveira, & de Paiva, 2014).

Uncertainty sampling was chosen as the query strategy because it has been demonstrated to be a strong baseline strategy. This query strategy assumes, that instances that are far from the decision boundary are adequately explainable and instances close to the decision boundary are uncertain. Naturally, this complements the SVM estimator very well. As a result, the Active Learner queries the samples about which it is most uncertain about (Osborne & Baldridge, 2004).

## Sentiment Analysis Models

The next section explores the machine learning models that will be used to perform sentiment analysis on the domain-specific corpus created by the Active Learner. The classification of the best performing model will then be used as an input to predict changes in stock prices. Because a SVM was used in the Active Learner to label the ground truth data, it will not be applied in the sentiment classification task. Otherwise, the results might lack robustness and be biased.

Before training the models, 20% of the data were set aside as the test set. To account for class imbalances, stratification was applied.

Naïve Bayes (NB): NB is a probabilistic supervised machine learning model. By working probabilistically, the classifier assigns the probability of belonging to a given class based on certain features (Jemai, Hayouni, & Baccar, 2021). Because of the high dimensionality of text data, which can be handled very well by NB, this algorithm has established itself as one of the standards for sentiment analysis. This thesis will use Multinomial Naïve Bayes to classify the sentiment of the text. This is due to the model’s ability to handle larger vocabulary sizes. In addition, the algorithm is simple to implement, suitable for real-time applications, and highly scalable. However, the algorithm’s prediction accuracy is frequently lower than that of other sentiment analysis techniques (Song, Kim, Lee, Kim, & Youn, 2017). Due to the easy implementation and fast training of the algorithm, Naïve Bayes will serve as the baseline classifier.

Long Short Term Memory (LSTM): LSTMs are becoming increasingly popular for sentiment classification. LSTMs are built on a recurrent neural network architecture (RNN). In an RNN the neurons are connected to themselves through time. As a result, the input from a time instance ti will also be used as an input for the next time instance ti+1.That leads to the problem of vanishing gradients. LSTMS are designed to overcome that problem.

The LSTM architecture does so via its four constituents: A memory cell which can remember a lot of information from previous states, an input gate which controls the inputs into the neurons, an output gate with an activation function and lastly a forget gate which resets the neuron (Priyantina & Sarno, 2019).

BERT: BERT is a relatively new machine learning algorithm developed by Google in 2018 and mainly designed for natural language processing. BERT is pretrained on the English Wikipedia and BooksCorpus. Because of the pretraining users won’t need as much computing power to achieve good results, even if the dataset is relatively small (Devlin, Chang, Lee, & Toutanova, 2019). The BERT github page even states that “Most NLP researchers will never need to pre-train their own model from scratch” (google-research, 2020).

### Hyperparameter Tuning

In order to find the best performing models, some hyperparameter tuning steps were taken. For the implementation of the Naïve model, five-fold grid search cross-validation was used to find the best parameters in a pre-defined parameter grid.

For the deep learning models, LSTM and BERT, a loop was created that iterates over a set of parameters. Within each iteration, the model is fit on the training data while setting aside 20% of the data for validation.

### Data, Code and Ethics Statements

To query the data from pushshift, the explanation of pmaw API wrapper from Github was used: <https://github.com/mattpodolak/pmaw>

The data was manually verified, by comparing specific, randomly-sampled, instances with the actual posts on reddit.

To label the initial train set, I created a graphical user interface using tkinter: <https://docs.python.org/3/library/tkinter.html>

To create the t-sne visualization, I relied on the documentation provided by Yellowbrick: <https://www.scikit-yb.org/en/latest/api/text/tsne.html>

To implement the Active Learner, I used modAL: <https://github.com/modAL-python/modAL>

All scikit-learn packages and classes, such as train\_test\_split, TfidfVecotricer, LabelEncoder, GridSearchCV, Pipeline, SVM and NB were implemented based by utilizing material provided during the Machine Learning course at Tilburg University, taught by Dr. Güven Ç and Dr. Önal, I.

The LSTM was implemented by using material provided during the Deep Learning course at Tilburg University, taught by Dr. Vanmassenhove E. and Dr. Saygili G.

To implement BERT the following tutorial was used: <https://skimai.com/fine-tuning-bert-for-sentiment-analysis/>

The code for this thesis is shared in the following github repository: <https://github.com/StefanWinterToo/Master-Thesis>

Note: Currently the code mainly consists of notebooks. However, before the final submission, I will properly structure the code.

All graphics used in this thesis were created by myself.

Due to time constraints, this first-submission does not use the full set of all possible hyperparameters. Furthermore, for the Active Learner the VADER sentiment lexicon was used as an oracle.

To the best of my knowledge, the literature used was referenced appropriately.

Results

Clear, transparent, and original presentation of results. Detailed visualizations, insightful multi-level evaluation of model performance.

## Discussion

The “to the moon” WallStreetBets movement had a tremendous impact on the lives of individuals, both to the positive and negative. Besides that, however, many investment funds have also been negatively impacted by the recent short-squeezes. While it might seem noble to root for individuals who try to force large funds out of their positions at big losses, it is easy to forget that many of those funds manage money for charitable endowments, pensions and others. Furthermore, such disruptions to the financial markets can harm its stability, thus causing spillover effects which can also negatively impact the lives of many people (Lyócsa, Baumöhl, & Vyrost, 2021). By being able to accurately measure and monitor the sentiment on WallStreetBets, market participants and regulators are able to preemptively take measures.

However, since the wallstreetbets subreddit has become very popular just recently, there is little academic research about the impact of the community on financial markets so far. Even though there is some research about sentiment analysis on wallstreetbets, that research does not use state of the art algorithms to perform sentiment analysis. This thesis not only tries to shine some light on those new and influential market participants, but also tries to put forward some methods that work best to perform sentiment analysis on the forum.

Not only did this thesis compare the performance of different models, but also proposed a highly efficient and reliable way to create a domain-specific annotated corpus, which can be used as the input to aforementioned models. To my knowledge, this thesis is the first research that creates a domain-specific corpus for the WallStreetBets forum.

Researchers, such as Talamás (2021), specifically propose future work on “inclusion of features derived from alternative manipulation of the data like sentiment analysis could lead to new insights“. I strongly believe that the methods proposed in my thesis can lead to better sentiment classifiers, which can then be used in other scientific or industrial applications.