



Article

Using Social Media & Sentiment Analysis to Make Investment Decisions

Ben Hasselgren , Christos Chrysoulas *, Nikolaos Pitropakis and William J. Buchanan

School of Computing, Engineering and the Build Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK

* Correspondence: c.chrysoulas@napier.ac.uk

Abstract: Making investment decisions by utilizing sentiment data from social media (SM) is starting to become a more tangible concept. There has been a broad investigation into this field of study over the last decade, and many of the findings have promising results. However, there is still an opportunity for continued research, firstly, in finding the most effective way to obtain relevant sentiment data from SM, then building a system to measure the sentiment, and finally visualizing it to help users make investment decisions. Furthermore, much of the existing work fails to factor SM metrics into the sentiment score effectively. This paper presents a novel prototype as a contribution to the field of study. In our work, a detailed overview of the topic is given in the form of a literature and technical review. Next, a prototype is designed and developed using the findings from the previous analysis. On top of that, a novel approach to factor SM metrics into the sentiment score is presented, with the goal of measuring the collective sentiment of the data effectively. To test the proposed approach, we only used popular stocks from the S&P500 to ensure large volumes of SM sentiment was available, adding further insight into findings, which we then discuss in our evaluation.

Keywords: social media; sentiment analysis; sentiment score; stock market; investment decisions



Citation: Hasselgren, B.; Chrysoulas, C.; Pitropakis, N.; Buchanan, W.J. Using Social Media & Sentiment Analysis to Make Investment Decisions. *Future Internet* **2023**, *15*, 5. <https://doi.org/10.3390/fi15010005>

Academic Editors: Paulo Quaresma and Filipe Portela

Received: 4 October 2022

Revised: 2 December 2022

Accepted: 20 December 2022

Published: 23 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

“Beating the market” has always captivated the minds of economists, investors, and researchers alike. There is endless discussion and investigation to try and better understand how to profit from it. One related area of interest considers using public sentiment from SM. This is exemplified by Mukhtar [1] who states, “leveraging social media to get the fastest news impacting stock prices isn’t a theory anymore, it is a reality”. Over 2.3 billion active users were recorded using Facebook in 2021 [2], and recent studies have proven that the sentiment data obtained from this vast pool of people can be utilized to make stock market decisions, and potentially even predictions [3]. This relationship between stock market performance and public sentiment can be seen clearly in certain instances. Most notably when popular individuals share their sentiment about certain stocks and directly impact the performance of that stock [4]. However, although there is continued research into this topic, much of the presented work includes mixed results [5], and there is still an opportunity for further investigation.

One of the main challenges faced in this field of work is being able to create a system which can extract and calculate the sentiment of vast quantities of SM data efficiently. Currently, the best way to achieve this is by using computational power and Machine Learning. Machine Learning is becoming ubiquitous in our lives and can be found in speech recognition technologies, image classification tools, autonomous driving, and many other areas. The specific area of Machine Learning that can be used to analyze the sentiment from SM efficiently is Sentiment Analysis (SA), a type of data intelligence used to identify the emotion behind text [6]. It can be implemented using Natural Language Processing (NLP), and to work effectively it needs to be configured specifically for its purpose, so

an understanding of the right datasets and types of algorithms to use is fundamental. Therefore, determining what type of SA algorithm to use for predicting SM sentiment is vital in our goal of helping users make investment decisions.

The first aim of this work is to provide insight into the field of study and outline existing research and technologies. Firstly, it will look to understand the relationship between public sentiment and stock market performance, then the sentiment from SM, and finally the topic of SA itself. This will also include a technical review aspect, where it will present existing approaches for creating SA models, extracting SM data, and evaluating results.

The second aim of this work is to show how it is possible to utilize public SM sentiment to help users make investment decisions. We will show this by designing and implementing a prototype that will look to track the stock market performance of selected S&P500 stocks, using the sentiment data that it will obtain from SM. Furthermore, a novel approach to factor SM metrics into the sentiment score will be presented, in which a post like weighting is factored into the SA algorithm sentiment score. The goal of this is to try and measure the collective sentiment of the data effectively.

The rest of the paper is structured as follows: In Section 2 an in-depth literature review is provided. Section 3 builds upon Section 2 and the proposed methodology and implementation of a prototype are presented. Section 4 discusses the results and the evaluation of the proposed approach and finally Section 5 concludes our work.

2. Literature Review

This section reviews the literature concerning the field of study. In Section 2.1, the relationship between public sentiment and stock market performance is explored. Section 2.2 then discusses the role of SM in obtaining public sentiment. Finally, Section 2.3 addresses topic of SA.

2.1. Public Sentiment and Stock Market Performance

Firstly, to understand how investment decisions can be calculated using SM sentiment, this paper will investigate the relationship between public sentiment and stock market performance. As previous psychological research has shown, human emotions do effect decision making [7,8]. Furthermore, this can be seen specifically in financial decision making [9] where psychologically enigmatic concepts such as the “gut feeling” have been better understood using neurobiology.

Previous research has proven the correlation between public mood and stock market performance. Interestingly, Atwater [7] demonstrates that the relationship between stock market and public sentiment can be weaker when you do not take inflation into account.

This is further explained in Figure 1, which compares a nominal measurement of the S&P 500 (depicted in blue) and the Bloomberg Consumer Comfort Index (depicted in red). In this illustration the two measurements start to differ around 2003 and are very different by 2012. However, further research by Atwater—as depicted in Figure 2—shows that if we deflate the value of the S&P 500 by the price of gold, we end up with a very similar result for each measurement. It’s important to note that by dividing the nominal market value by the price of gold you are negating the “extreme actions of monetary policymakers” [7] which gives us a more accurate representation of the actual market price. Once applying this rule, we can clearly see that public sentiment closely maps the performance of the stock market.

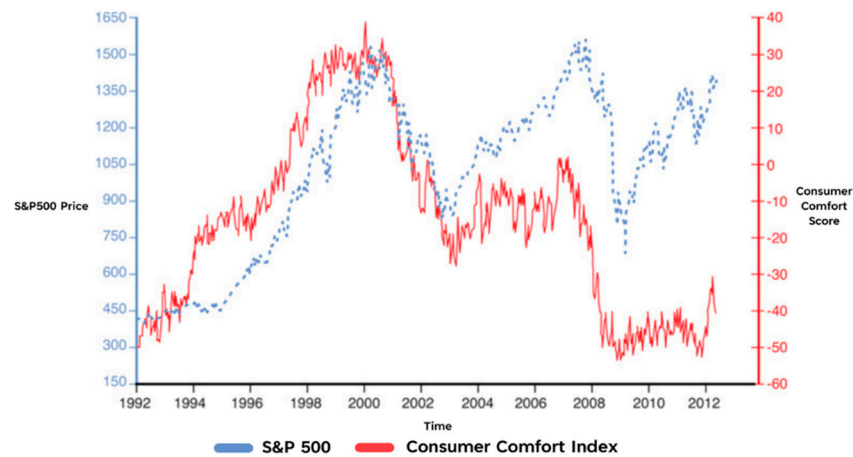


Figure 1. The Bloomberg Consumer Comfort Index versus the S&P 500 (in nominal terms), 1992–2012, [7].

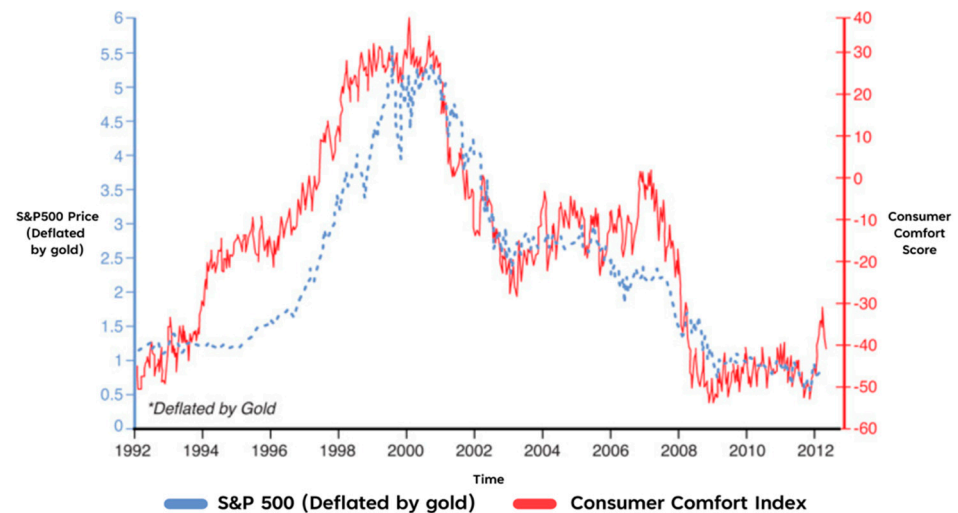


Figure 2. The Bloomberg Consumer Comfort Index versus the S&P 500 (deflated by gold), 1992–2012, [7].

As it is evident that there appears to be a strong relationship between the value of the stock market and public sentiment, it is not so far-fetched to suggest that public mood could potentially influence the performance of it. This is reinforced by Padhanarath et al. [10], who argues that investor confidence is one of the main motives behind an individual's decision to buy or sell a stock, and this confidence is determined from the views and emotions of other people. Ni et al., [11] agrees saying, “we find that the change of stock prices can be predicted by the public opinion”. Figure 2 also illustrates that the consumer comfort measurement predicted rises in the S&P 500 in certain years such as 1994 to 1997. However, it's important to understand that public sentiment is not the only driving factor behind market performance and other implications such as politics, natural disasters and economics also play a role [10]. In contrast, Ref. [6] quotes André Kostolany—a historic stock market investor—who stated, “Facts only account for 10% of the reactions on the stock market; everything else is psychology”. So, although research suggests that public sentiment may not be the only factor affecting stock market performance, there's certainly enough evidence to support a strong relationship between the two. Therefore, we can justify further investigation into the concept of using public sentiment to help make investment decisions.

2.2. Social Media

Now this paper has identified the strong relationship between public sentiment and stock market performance, an investigation into the ways of harvesting this sentiment for SA will be discussed. Additionally, this sentiment must be text based, related to financial data, and easy to access. Undoubtedly, one of the most effective ways to gather this data would be using SM. This is exemplified in [10], which explains that SM is a place where investors leave their thoughts and ideas, and it can be obtained by anyone who has access to the internet. However, there are many SM platforms to choose from, and it's vital that the one selected for our approach meets the criteria mentioned above.

The Twitter API is a common tool for extracting public sentiment and is used in almost every source referenced in this literature review. It enables development access to Twitter, a popular SM platform that allows users to share their opinions, most famously in a text-based format limited to 280 characters (as of October 2021), which is a format that ties perfectly into NLP [12]. Wołk [3], who investigated price predictions of cryptocurrencies using sentiment from Twitter and Google Trends, states, "Twitter is one of the most widely used social media platforms and reflects diverse perspectives from users worldwide" [3]. An older study by Bollen et al. [5] found that the value of the Dow Jones Industrial Average (DJIA) could be predicted more accurately using data from large-scale twitter feeds. However, the main weakness of this study is that it was conducted over a decade ago and there have been many political, technological, and economical changes since then: most relevantly, the huge increase of active SM users.

More recently, Shead in [13] found that certain stocks could skyrocket after positive SM content about that company were posted by influential individuals. In this article, the individual was Elon Musk and after tweeting, "I kinda love Etsy", the share price of the e-commerce company increased by 9%. CD Projekt shares also soared by more than 12% after Mr. Musk tweeted "With Cyberpunk, even the hotfixes literally have hotfixes, but . . . great game." These findings are like that of [6], who also acknowledges that influential twitter users such as CEO's, politicians and journalists can have a greater impact on individual stocks. Based on these findings, it appears that analyzing sentiment from more influential individuals could be an effective way to make investment decisions. This is well presented in Figure 3, which clearly illustrates the decrease of Bitcoin's value after a negative tweet about Bitcoin was posted by Mr. Musk. However, much of this work only focuses on a very limited group of highly influential individuals, and there is a lack of evidence to show that the same can be said for other less influential SM users. Additionally, the goal of this work is to analyze the general sentiment of the public and will not focus purely on highly influential individuals.

Mukhtar in [1], who analyzed the sentiment of tweets to make stock market predictions, found that posts with higher engagement is a good indicator on the sentiment agreeability of others. More specifically, a post with a higher number of likes, comments, and reactions, is a better post to analyze than a post with less, as it can give insight into the collective sentiment of multiple users. This is acknowledged in the work of Porcher [14], which factored Twitter likes, tweets and retweets—using a weighting factor formula—into a tweet sentiment system they designed, gaining insight into the collective social distancing beliefs of individuals throughout the COVID-19 pandemic. Although this study has acknowledged the work undertaken in [5] is dated in certain aspects, it also identifies relevant findings as well: especially in its discussion about extracting sentiment from SM. This was achieved by choosing tweets that contained phrases like "I feel" or "I'm feeling". Not only is this a useful way to extract posts that contain sentiment, but also a way to filter out spam. Chauhan et al. in [15] agrees with this, explaining how the data pre-processing stage of SA is crucial in preventing accuracy degradation in sentiment classifiers. Furthermore, Renault, in [16] found that the pre-processing technique used in their experiments impacted the patterns between the calculated sentiment and the stock returns. Their work also found that by not filtering out emojis and stop words from the SM data, there was improved precision results for the SA algorithms classifying the data.

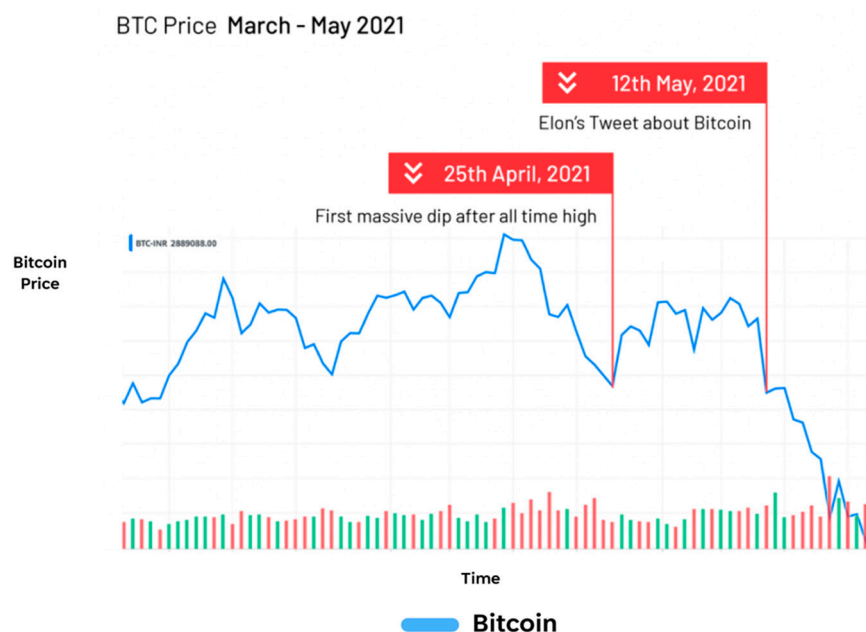


Figure 3. The Price of Bitcoin, March–May 2021 [17].

2.3. Sentiment Analysis

This review has shown that there is a strong relationship between public sentiment and stock market performance. It has also shown how SM—especially Twitter and its API—is an effective way to extract the latest emotional views of stock market data. It will now discuss how SA is an effective way to measure user sentiment from SM, exploring a variety of different approaches that will be considered when designing the prototype.

2.3.1. An Introduction to SA

SA is a part of “Data Intelligence Research” that focuses on data containing emotions [6]. It does this by analyzing the polarity of messages and stating if they are positive, negative, or neutral [18]. In the last decade, it has become ubiquitous in technologies and a useful tool for businesses to better understand the views of their customers. This is further discussed by Shukri et al. [19] which successfully used SA to analyze consumer data from the automotive industry and extract the polarity and emotional classes of different car models. The results identified Audi customers were the most satisfied, whereas as BMW customers were the least. From this, it’s clear that SA is an efficient tool in understanding the sentiment of large groups of people, which is why we will use SA to understand the sentiment of SM users in relation to the stock market.

Furthermore, to find out how SA can be used effectively in practice, this study will look at Supervised Natural Language Processing (SNLP). SNLP is the implementation of SA on a computer system using Machine Learning algorithms such as Support Vector Machines (SVM), Logistic Regression and Neural Networks (NN) [20]. Furthermore, the term supervised refers to the fact that these algorithms learn from a provided dataset. In this section, a range of these algorithms are investigated with the aim of identifying the best candidate for our prototype. Additionally, this paper will discuss other literature related to the topic to better understand how SA can be used to help make investment decisions.

2.3.2. Investigating SNLP Algorithms

The Logistic Regression Algorithm (LRA) stems from the field of statistics and is a popular choice of algorithm for solving binary classification problems, utilizing its sigmoid function to classify data into a value between 0 and 1. One of the most effective traits of the LRA is its simplicity in being implemented and trained [21]. It also provides a probabilistic output that can be useful in understanding how classified a value is. For example, it can

determine the positivity or negativity of sentiment data. The work of Krouska in [22], which evaluated a range of algorithms in determining the sentiment of tweets, found that the LRA performed well across all the datasets it was tested against. However, some disadvantages are its tendency to overfit data, it does not guarantee 100% convergence, and it cannot effectively handle outliers in provided data sets [6].

SVM is another popular algorithm used for regression problems and more notably classification problems. It works by plotting data in n -dimensional space, where n is the number of features or categories, and the value of each category has a corresponding coordinate in that space. Using the example of sentiment again, the included categories could be positive or negative. The data is then trained and classified by finding the optimized hyper-plane that differentiates the two classes, and its effectiveness in doing so makes it a common choice for classification problems. This can be seen in the work of Smailović et al. [4], as cited by Krouska et al. [22], which looked at making stock market predictions using financial sentiment and found that the SVM achieved the best results. However, the algorithm tends to be weaker in large datasets, and its lack of a probabilistic output for the classified data (in its simplest form), makes it less desirable than other algorithms such as the LRA.

Decision trees are another common classification algorithm used for SA. A decision tree gets its name for its tree like structure, in which nodes are connected by edges in a hierarchical structure. Using a dataset, an entropy formula, and an information gain formula, a decision tree can be built and trained to classify data effectively. One of the popular existing algorithms is known as ID3 [23]). The work in [22], which compared a range of SNLP algorithms found that the ID3 based algorithm performed well, achieving an F-Score of around 70% in testing. However, it did not perform as well as the SVM algorithm and the Logistic Regression algorithm which achieved higher F-Scores of around 80% in some of the tests.

A NN is another type of popular Machine Learning algorithm used for classifying data [24]. It's composed of layers containing multiple interconnected nodes. Each node has an associated weight and threshold, and if the output of that node is greater than the threshold, then it fires a signal to the next connected node in the network. With this structure, and the use of a training method (such as back propagation), this algorithm can be trained to classify data very accurately. One of the more popular NN algorithms used for SA is known as a Convolutional Neural Network (CNN). This is well presented in [25], which investigated a variety of NN's—such as doc2vec—to analyze the sentiment of experienced individual investors (from financial SM platform StockTwits) to predict stock market performance. It found that CNN outperformed the other networks with accuracy results of up to 86%. The paper explains how CNN's deep learning qualities overcome the shallow learning process of data mining approaches, which do not consider the order of words in a sentence, and as a result, lose insight when classifying the sentiment. However, NN's are sometimes inefficient and take a long time to train. Additionally, NN's are not always guaranteed to converge unlike the SVM.

2.3.3. Related Literature

Based on the above findings, this paper has found that the LRA, SVM, and NN algorithms all perform well for SA and will consider them when implementing the prototype. Finally, to complete this literature review, a higher-level investigation into other related research is undertaken to help find any information that could be of use to support our approach.

Recent research has shown that it could be possible to use SA to help make investment decisions by extracting and analyzing emotions from users. Wołk [3] agrees with this and suggests that SA is an effective tool to model capital markets and cryptocurrencies. This is further analyzed in the work of Agarwal et al. [26], which looked at stock trends from over the last 2–3 decades and found that SA can be used to help make investment decisions. Renault in [16], which used SA to compare the returns of selected stocks and SM sentiment,

found that certain algorithms were able to show a correlation between the two data sources. However, they also found that the sentiment was not able to provide insight into predicted stock returns. Overall, although the work in this field of study has found positive results, generally the conglomerate view is that more work needs to be undertaken to further prove this [10,11].

Earlier research [5] proposed that the performance of the DJIA could be predicted using sentiment gathered from Twitter to help make investment decisions. The sentiment of these tweets was predicted using tools such as Opinion Finder, a SA tool developed in a collaborative process by different universities. The findings showed accuracy results of up to 86.7% for predicting the daily ups and downs of the DJIA. However, the lack of test samples in this research devalues the findings.

The work of Zhang [27] proposed using three types of features extracted from social media/stocks data to make Chinese stock market predictions: Sentiment Features (SF), Stock Specific Features (SSF) and Stock Relatedness Features (SRF). The SFs were classified using naïve Bayes and proved to be a useful measurement to make predictions, and thus help make investment decisions. This is presented in Figure 4 which clearly shows that the accuracy results of the SF's + SSF's, is higher than that of just SSF and SSF + SRF. However, the accuracy value and the area under a curve value are both under 58%, which, devalues the findings.

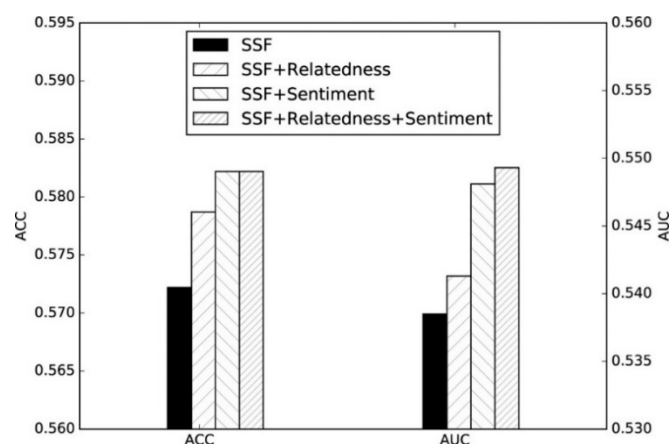


Figure 4. Prediction results trained on different combinations of features with SVM [25].

Other work undertaken by [28] addresses the issue of high-speed streaming data outputted by SM platforms and the challenges faced in analyzing the sentiment of it in real time. It goes on to explain how this vast SM data generally consists of short text, which can make it harder to identify the related topics, and the existing topic-based NLP models struggle processing this information quickly. Their work proposes a deep learning system that works at sentence level to extract topics using an online latent semantic indexing and a regularization constraint to efficiently extract the topics from SM content, and then uses a long, short term memory network and a topic level attention mechanism to perform SA. The system designed uses an online learning technique and works at sentence level to identify topics and process short text data in a serial fashion. This enables large datasets to be used and it makes the system scalable. The paper found that the system was able to process larger quantities of data much faster than the existing models and was able to achieve over 80% accuracy in certain topics. These findings are useful, as the prototype created in our work also needs to consider speed and scalability when mining data from SM.

Another recent paper [15], analyzed a variety of related sources that used SA techniques on SM data to make election predictions. One finding was that Twitters Streaming API was the by far the most used to extract data from SM, compared to less popular options such as the Twitter Search API. This paper also found that most of the analyzed sources used a combination of SA techniques and “volumetric” data to make election predictions.

This volumetric data involves analyzing the likes, retweets, and comments on posts to make predictions, as discussed in [1,14].

More can be found on the work of Catelli et al. [29–31], which compares a NN—specifically designed for SA (BERT)—against lexicon-based approaches, in identifying the sentiment in Italian e-commerce reviews, acknowledges the struggles of using SA in languages other than English and Chinese. More specifically, they discuss that common SA NN’s rely on large datasets to train the models effectively, which do not always exist for non-English/Chinese languages. Although our work will focus purely on textual data in English and circumvents the outlined issue, we acknowledge that we are limiting ourselves to the amount of sentiment we are obtaining from Twitter, and thus reducing the effectiveness of our prototype.

3. Methodology and Implementation

To test whether SM sentiment can be used to help make investment decisions, a prototype was developed. This section breaks down the methodology and implementation of this prototype. Section 3.1 discusses the functional and non-functional requirements. Section 3.2 looks at the prototypes design, discussing the chosen architecture and SA algorithm, in which the novel approach to factoring in SM metrics—to gauge collective sentiment—is discussed. Finally, Section 3.3 illustrates how the prototype was implemented.

3.1. Analysis and Requirements

As the main intention of our approach is to research the concept of making investment decisions using SM sentiment, it was important that the prototype was broken down into functional and non-functional requirements to ensure the most important features were developed to meet this intention. Firstly, a MOSCOW analysis was conducted to prioritize all the requirements and break them into Must Have, Should Have and Could Have features. This allowed us to clearly see what crucial functionality needed to be implemented to ensure a successful MVP was developed to test the proposed approach. For this prototype, only the Must Have and Should Have sections were used to create a list of functional and non-functional requirements:

Functional Requirements

- Login authentication and user accounts.
- An interface to allow users to add and delete stocks (assets) to a portfolio.
- A SA algorithm that can calculate the sentiment of tweets correlated to each asset (stock).
- Display the sentiment metrics on the user interface, from which the user can make investment decisions.
- A Twitter dataset to train and test the SA models.
- Twitter Integration to extract relevant and public tweets.
- Provide appropriate error and exception handling.
- Display a list of tweets on the user interface from the current day.

Non-Functional Requirements

- Interoperability.
- Scalability.

3.2. Design

3.2.1. Use Cases

Firstly, before designing the architecture of the application, it was important to determine the required functionality of it. To do this, user stories were created (See Figure 5), which were based on the functional requirements and described the narrative of user scenarios. In these specific stories, the user dashboard is depicted, in which a large portion of the prototype functionality takes place. From these user stories, the required functionality was identified to create a use case diagram (See Figure 6).

- As a user I want to be able to create an account, in which I can login and view my portfolio of assets. I also want to have the option to change my account security (password and email) and view my account details.
- As a user I want to be able to easily view, add and delete assets in my portfolio. I also want to be able to easily visualise the SM (social media) sentiment correlated to each asset over a specified time of my choice. I also want to be able to view a selection of the more popular SM posts on the current day. With all this information I want to be able to make an investment decision based off the sentiment.

Figure 5. User stories.

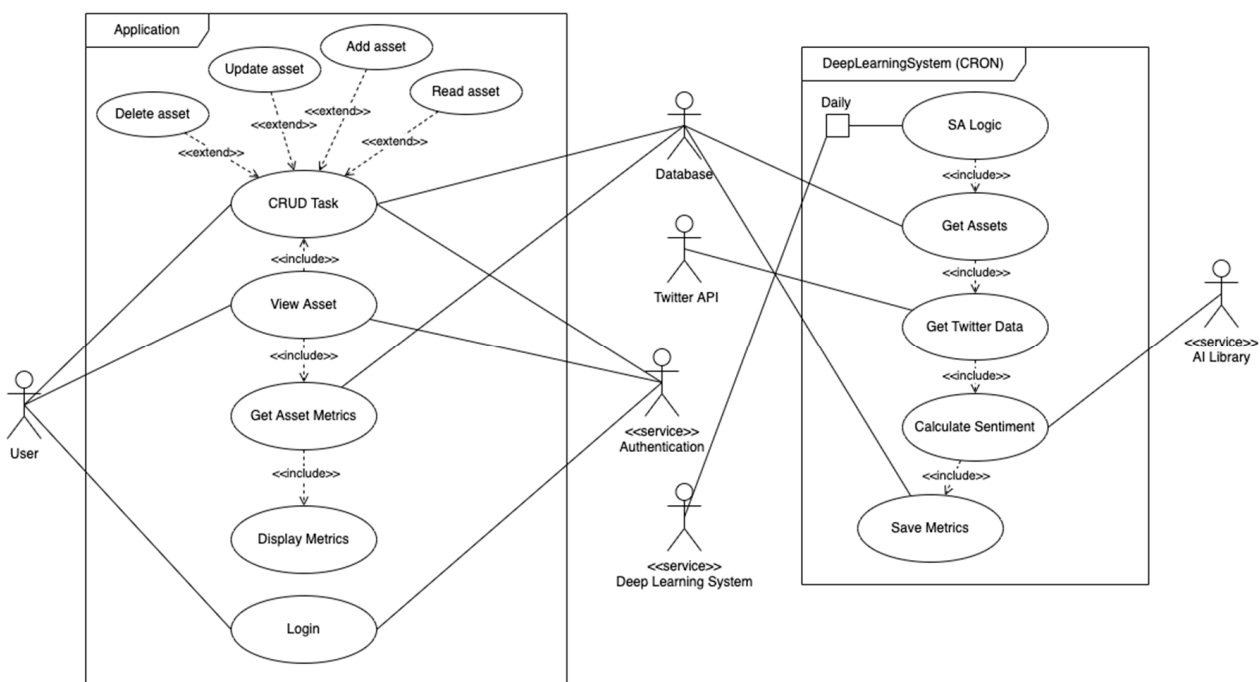


Figure 6. Use Case Diagram.

This Use Case Diagram is a graphical depiction of the entire system application, created to allow us to visualize it clearly. This diagram not only shows system functionality but also the different components. From this, we were able to identify two sub-systems that would be implemented to create the entire application.

1. **The Web Application, in which the user can:**

- Login or register.
- View their portfolio.
- Add, and delete assets.
- View the SA metrics of each asset to help users make investment decisions.

2. **The Deep Learning System:** which runs independently from the web application as a CRON job. This prevents any slowness on the web application as it's executed on a different server, which was crucial to ensure a good user experience. In this system:

- The assets in the database are obtained.
- Tweets correlated to those assets are extracted using the Twitter API.
- Those tweets are analyzed for sentiment using an AI Library.
- The SA metrics are saved to the database.

3.2.2. System Architecture

Once the rough structure and relationships of the system were identified (with the use case diagram), a high-level visualization of the system architecture was designed to understand how it could then be implemented (See Figure 7).

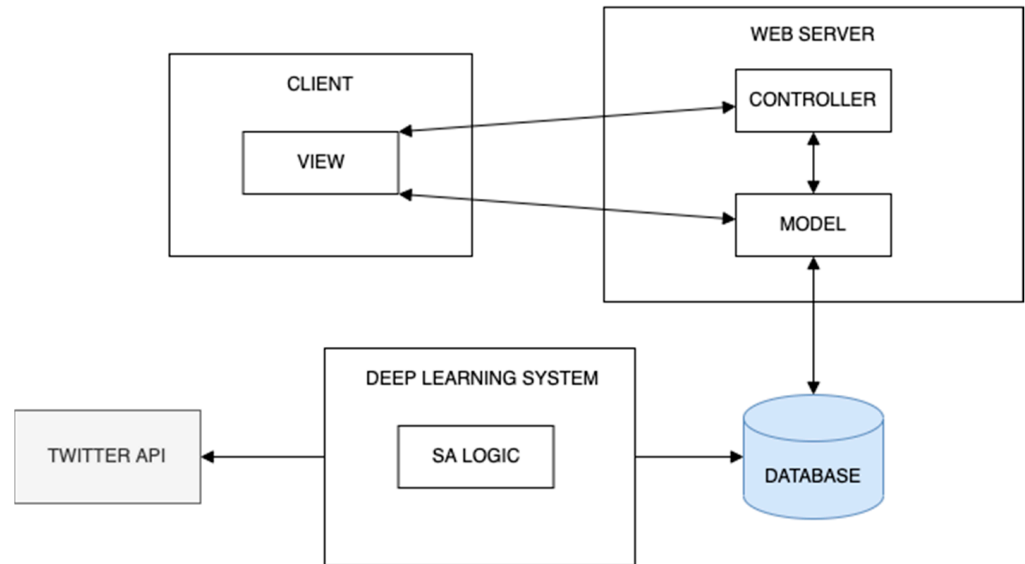


Figure 7. Web application diagram.

The chosen design was kept simple and practical. Keeping this principle in mind, a client-server, and Model View Controller (MVC) architecture was selected as the best candidate for the web application. This architecture is not only simple, but also loosely coupled (as the logic is separated from the data and UI) ensuring the system met its non-functional requirements: interoperability and scalability.

Figure 7 also depicts the Deep Learning System as separate from the web application component, as specified in the use case diagram (See Figure 6). This is a simple program executed as a daily CRON job and it also has READ/WRITE access to the web application database. It also contains the AI library to perform the SA operations on tweets and a connection to the Twitter API to extract those tweets, that correspond to each asset.

3.2.3. System Models

Once a high-level visualization of the system had been designed, we then created a class diagram to understand the web application models and the relationships they have with one another (See Figure 8).

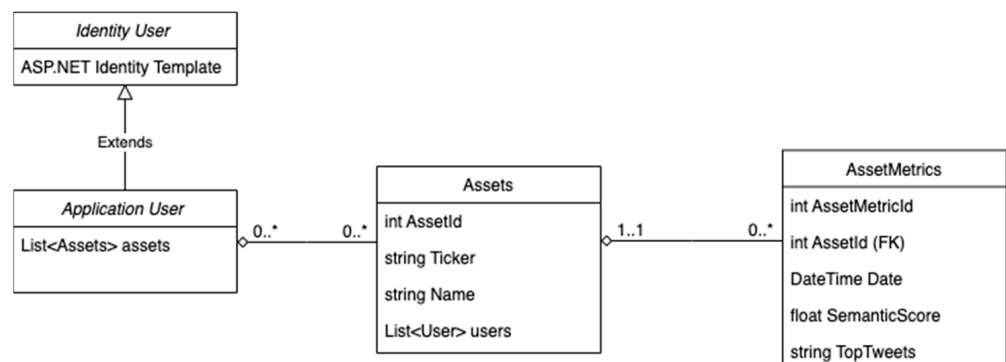


Figure 8. Class diagram.

As seen in this diagram, there is a reference to ASP.net in the Identity Table, which was the framework used to implement the system, and it will be discussed later in this section. Additionally, this diagram has a simple structure with only 3 models (excluding the ASP.net model):

Application User: A model which contains all the details of the user such as their email and password.

Assets: Has a many to many relationships with the Application User model. This structure allows users to add and remove existing assets without deleting the assets themselves. This was a design choice made while implementing the application for two reasons:

Preventing users from adding the same assets to the system, which as a result, makes it easier for the deep learning system to generate all the asset metrics (SA metrics).

Allowing us to have complete control over what assets can be added to the system was important because the whole approach was dependent on large volumes of relevant SM data. Utilizing this, we only added big companies (selected S&P500 Stocks) which would guarantee large quantities of relevant SM data, and overall provide more useful insight when testing our approach.

Asset Metrics: Has a many to one relationship with Asset and includes a date field, a sentiment score field and a range of tweets that have been “JSONified”.

The **sentiment score** is a number between 0 and 1 and is the average sentiment of all the tweets of a particular day for that specific asset.

If the score is less than, 0.5 it's negative; and if it's greater than 0.5, it's positive.

This way of quantifying the sentiment was a design choice as it allowed the sentiment to be easily plotted on a graph in the UI. That way, the trend of sentiment could be visualized and compared to the stock market performance of each Asset, to test the proposed approach effectively.

This model is mapped to an SQL table where the Deep Learning System saves the daily metric results to.

3.2.4. User Interface

For the UI, a wireframe diagram was created to visualize what the web application would look like. This diagram shows at a high level the rough design of the UI (See Figure 9).



Figure 9. User Interface (UI) wireframe diagram.

The overall design was kept simple to make the application more usable and readable for users. It was also to keep the design somewhat ambiguous so we could have flexibility when implementing the system, with the intention of speeding up the development process.

The design is based on the previous functional requirements listed in Section 3.1 The bottom left screen shows the admin dashboard, where the bulk of the functionality exists.

The graph section is where the asset metrics (sentiment score) are visualized, and the interface used by the user to understand the sentiment and make investment decisions. The side bar on the left, displays all the users' assets, where they can select each one by clicking on them. The delete and add functionality is attached to the corresponding delete button and plus icon. Finally, the top menu bar is used to indicate whether a user is logged in or logged out, and it will also provide links to the login and registration pages.

3.2.5. Deep Learning System

The SA model was designed based on the findings in Section 2.3.2. Using those findings, the Logistic Regression Algorithm (LRA) was selected as the ideal candidate to train and evaluate the model.

The LRA was chosen because it not only handles binary classification efficiently, utilizing its sigmoid function (See Figure 10), but it also can be used to output the probability of that binary value (positive or negative in this case) [6]. This probability gives a number between 0 and 1, where a positive tweet is greater than 0.5 and a negative tweet is less than 0.5. This value was important as it allowed a sentiment score to be calculated for each tweet, as required by the system (See Section 3.2.3).

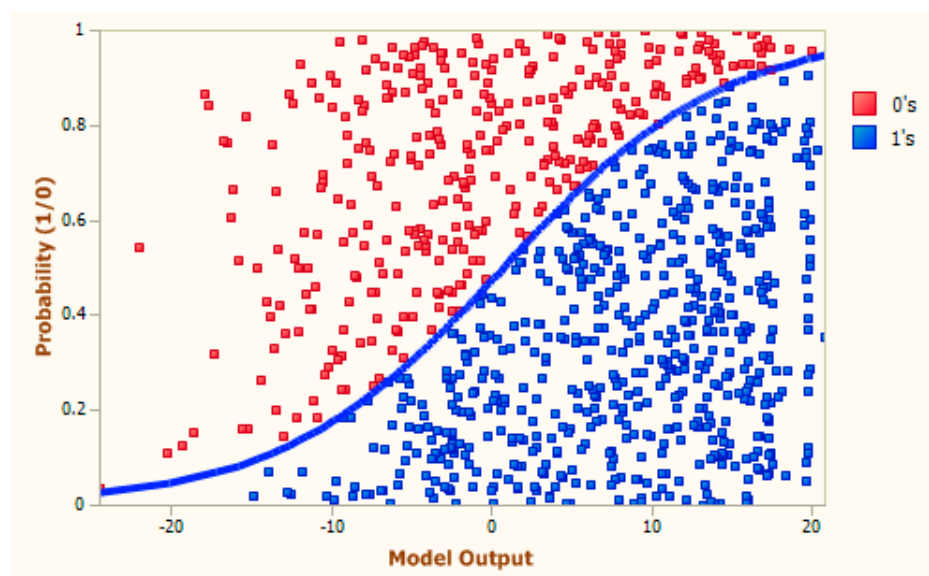


Figure 10. LRA Classification Diagram [21].

Factoring SM metrics into the sentiment score: Another finding in the literature review identified that using SM metrics is an effective way to help understand the collective sentiment of others. In our work, SM metrics are factored into the sentiment score, with the goal of trying to gauge the collective sentiment of a tweet. In this prototype, we chose to factor in the number of likes of a tweet because it shows how agreeable a tweet is with a large group of people. If a tweet has a high like count, then an assumption can be made that lots of other people feel the same way, thus identifying the collective sentiment for that tweet. To interpret this collective sentiment, a system was designed that increased the sentiment score of that tweet if it had a high like count. The aim of this feature was to make a tweet more positive or negative if lots of people agreed with it, thus more accurately mapping the collective sentiment of that tweet. To understand how this feature would work, a flowchart was created to break down the problem into simple steps (See Figure 11).

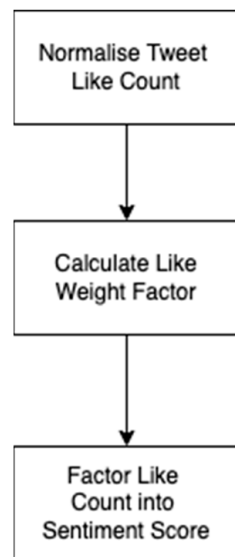


Figure 11. Deep Learning System flow chart.

As seen in the above flow chart, multiple steps are taken to factor in the like count. Firstly, the like count is normalized, which is calculated according to Equation (1):

$$NLC = \frac{ActualLC - MinLC}{MaxLC - MinLC} \quad (1)$$

where NLC is the normalized like count (a number between 0 and 1), $ActualLC$ is the original like count, and $MaxLC$ and $MinLC$ are the minimum and maximum number of likes in the dataset of tweets. Once the like count is normalized it is then factored into the probability score according to Equation (2):

$$Ps_1 = Ps + (LWF \times NLC), \quad (2)$$

where Ps_1 is the updated predicted sentiment score (factoring in the like count), Ps is the original predicted sentiment score (a number between -0.5 and 0.5), NLC is the normalized like count and LWF is a like weighting factor. LWF is used to limit how much sentiment can be added onto Ps to ensure the updated sentiment score stays within the maximum and minimum thresholds. It is calculated according to Equation (3):

$$LWF = MaxLWF - |Ps - MaxLWF| \quad (3)$$

where $MaxLWF$ sets the value limits of the LWF , 0.25 and -0.25 . Note that equations 2 and 3 only factor in positive sentiment scores (a number between 0 and 0.5), hence there is also opposite equivalents for each equation used to handle negative sentiment scores (a number between -0.5 and 0). It's also important to note that the sentiment score is originally in the range of 0 to 1. For these equations it is converted to lie within a range of -0.5 and 0.5 , and then converted back after.

The output of Equation (3) is illustrated in Figure 12. This diagram depicts the value of LWF , based on the value of Ps . From it, the following conclusions can be made:

The trend of the line shows that tweets with an already very positive or negative score, or tweets with a very low positive or negative score (aka neutral sentiment), will have a much lower LWF . Therefore, if these tweets have a high like count, less sentiment will be added to them (based on Equation (2)). Not only does this prevent already high sentiment scores from passing the maximum and minimum sentiment thresholds, but it also keeps tweets with neutral sentiment from becoming too positive or too negative, which suited the requirements of the system.

The trend of the line also shows that tweets with moderately positive or negative sentiment scores, will have much higher LWF. Therefore, if these tweets have a high like count, a greater amount of positivity or negativity will be added to them (based on Equation (2)), as required by the system.

It's also important to note that in the rare but possible case, where $P_s = 0.5$ or -0.5 , the LWF would then be 0, and in that case, nothing is added to the sentiment score, keeping P_s within the max and min thresholds, as expected.

Using the above design for this Deep Learning System, we were able to factor likes into the sentiment score, ensuring a high like count increased the positivity or negativity of a tweet.

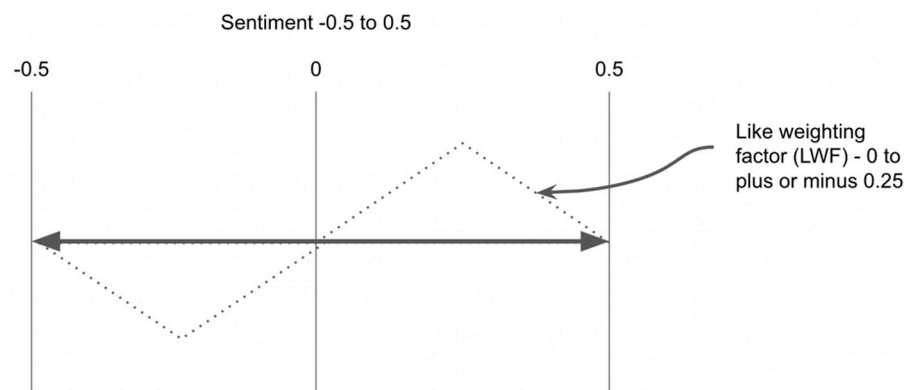


Figure 12. Like Weighting factor.

3.3. Implementation

In this section, a rough overview of the developed prototype is discussed. The implemented application can be broken down into two sections, the web application, and the deep learning system.

3.3.1. Web Application

The ASP.net framework was chosen to implement the web app (See Figure 13). This framework was selected for a variety of reasons. One being that it meant the web app could be deployed to Azure, an extensive cloud computing ecosystem created by Microsoft, that would then make it simple to set up a database server and other processes for the application.

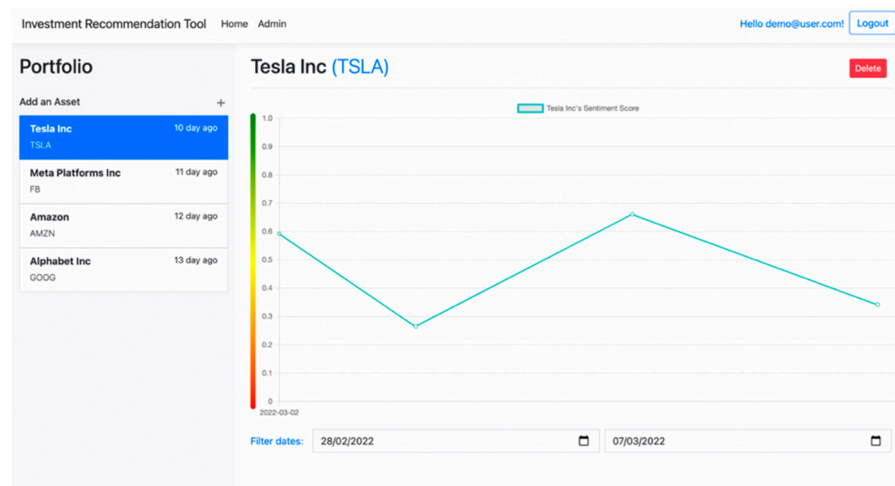


Figure 13. The Web Application Prototype.

Another beneficial feature of ASP.net is its Entity Framework (EF). EF is an ORM framework that allows users to build their models without having to worry about creating the equivalent SQL tables to store the model data.

This is captured in Figure 14, which depicts the SQL tables that were automatically created by the EF framework. In this diagram the three tables, Assets, AssetMetrics and AspNetUser (Application User), are the generated SQL tables for the corresponding models from the class diagram (see Figure 8).

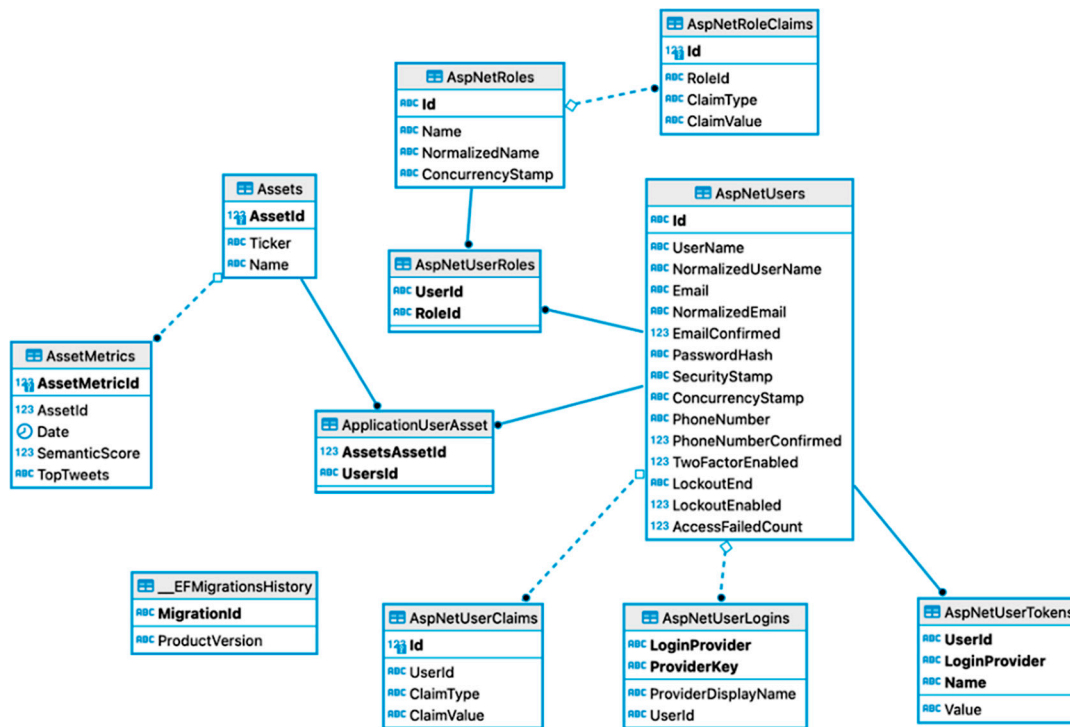


Figure 14. ER Diagram—Database Schema.

The model relationships can also be seen. Specifically, the many to many relationship between Assets and ApplicationUser (AspNetUsers), in which EF generated an additional table ApplicationUserAsset table to handle the relationship. The other tables are part of the Identity Template, a “user handler” template provided by ASP.net which handles all the user accounts, authentication, and sessions. This pre-written Identity Template was scaffolded into our work. This decision was made because although the user accounts were important for the application to work, it was not however an integral part in testing the functionality of our work. Choosing to use the Identity Template ensured a fully tested and robust user framework was implemented into the system, allowing us to focus on implementing the key features.

ASP.net comprehensive features also meant the entire middleware was handled automatically which allowed Controller actions and Views to be implemented with ease.

To handle the client-side functionality, ReactJS was integrated into the web application. ReactJS was used to handle the functionality of the users dashboard, and allowed us to dramatically increase the usability of the system by:

Data binding and state management: The HTML components could be bound to variables, that when updated would also update the bound DOM element automatically. This meant the page would not have to be refreshed every time a user clicked on a new Asset for example.

UI Components: ReactJS allowed the us to break the UI into modular components, improving the readability and maintainability of code, and drastically reducing the amount of code written.

Additionally, the web application was stylized using pre-written CSS and JS libraries. Bootstrap was implemented into the system to handle this, and it allowed us to quickly build attractive web pages without having to write lots of CSS code.

To display the Asset sentiment data on a graph, Chart.js was integrated into the application. Chart.js is a JS library that allows users to quickly create attractive and intuitive graphs. To display the sentiment data, a line chart was chosen. This decision was made as it enables users to see the trend in data, which was vital in allowing the user to make investment decisions. Using a trend line allows the sentiment to be compared to the stock market performance of that Asset, which is how our proposed approach tested.

To help improve the readability of the graph, a colored gradient line was displayed next to it, where green indicates very positive sentiment, yellow indicates neutral sentiment and red indicates very negative sentiment. We made this decision while implementing the system, as the sentiment score (ranging from 0 to 1) was not a very intuitive value by itself, and the more important things to show was the trend of data across the 2-dimensional plane of time and sentiment. This colored gradient line helped visualize this trend of sentiment.

Finally, date input fields were implemented below the graph, to allow the user to select the sentiment of an Asset over a selected period. This feature again makes the application more usable, as users can view the sentiment trends of their Assets over specified dates, and it also allows them to increase the polarity of data over smaller date ranges. It also prevents all the sentiment data from being loaded at once on the client side. Instead, AJAX calls are made to the server when date inputs change, and the server then responds with the corresponding data. Furthermore, as the dashboard is built with ReactJS, the graph automatically updates when new data is sent down from the server because of its reactive qualities, improving the overall user experience.

3.3.2. Deep Learning System

To implement the Deep Learning System, an Azure Functions project was created. Azure Functions is part of the .NET ecosystem, and it allows users to create programs that can be triggered by different events. In this system, the function is a CRON job, triggered at a selected time of day. One reason this framework was chosen to implement the system was because of the fact it's part of the .NET and Azure ecosystem. The Deep Learning System required READ/WRITE access to the Azure web application database so using an Azure Function made that implementation simpler, as the Microsoft.Data.SqlClient framework could be installed and utilized to connect to the database. This CRON job was vital, as it would gather the sentiment data of each Asset over a certain period, that would then be used to test our system.

Since the Deep Learning System was now implemented using .NET and C#, different aspects of the system had to factor this in:

Twitter API: Based on the findings in the literature review (See Section 2.3), the Twitter API was selected to extract the tweets related to each Asset on the web application. A Twitter developer account was created which gave limited access to the API, such as the number of pulled tweets being limited to 500,000 a month. However, these restrictions were not too much of an issue as the "Recent Search" endpoint allowed us to access the recent tweets related to the stock, as required by the design. However, the number of tweets pulled from this endpoint was restricted to around 200 on the active API plan. To integrate with the Twitter API, the .Net TweetinviAPI Library was utilized. It allowed us to connect to the API, set up queries, and get all the public metrics of each tweet using C#.

SA model: To implement the SA model, ML.net was used. ML.net is a Machine learning framework, created for the .NET suite. To handle the input data and prediction data, classes were implemented. The model was then trained and tested using the design from the previous section, utilizing ML.net's SdcaLogisticRegression method to train the model. The twitter dataset used to train and test the model was downloaded from Kaggle, and included around 5000 pre-processed tweets, each with a negative (0) or positive (1) score. To evaluate the model, metrics were recorded from the results of the test set. As

depicted in Figure 15, the Accuracy, Area Under the Curve, and the F1 Score were all used to quantify and evaluate the SA model. This process took time, and we adjusted the sizes of the training set to try and improve the results of the SA model. This process was important, as the model needed to be able to give somewhat accurate results when making predictions to test the proposed idea effectively.

```

===== Training Model with Training Data =====
===== End of training =====

===== Evaluating Model accuracy with Test data =====

Model quality metrics evaluation
-----
Accuracy: 78.39%
Auc: 85.78%
F1Score: 82.23%

===== End of model evaluation =====

```

Figure 15. SA Model evaluation.

Calculating the Sentiment Score: Once the Twitter API system and SA model had been implemented, the sentiment of tweets could then be calculated. Note, before sentiment predictions were made, the tweets were firstly pre-processed using regular expressions. This logic was based on the findings from the literature review, which highlighted the importance of pre-processing the SM data. Then, after the sentiment prediction was made, the like count was factored into the sentiment score using the methodology from Section 3.2.5. Finally, the average sentiment score of all the tweets was calculated, and then saved to the database alongside the current date.

3.4. Testing

To test the prototype, unit tests were created using xUnit.net, a test framework within the .NET ecosystem. These tests were implemented over the duration of the implementation stage to ensure that individual components of the system worked before being integrated with other components. For the web application, a mock database and application context was created for all the unit tests. This allowed us to test out the functionality of the controllers and ensure they were performing correctly, without having to depend on the production database or other production services. For the deep learning system, unit tests were also created to test out the different components such as the SA model, and the tweet pre-processor method.

4. Evaluation

Once the prototype was implemented, an approach to evaluate the system was required. Section 4.1 discusses the chosen approach and the strategy used to obtain and compare the sentiment data and corresponding stock data. In Section 4.2, the results are presented and then discussed, in which the trends of the two data sets are compared to evaluate the proposed method.

4.1. Realization

The realization of the evaluation was conducted over a period of two weeks—immediately after the completion of the prototype. This process was vital to ensure enough sentiment data could be obtained for each asset to identify trends in the data that could then be compared with the stock market performance of that particular asset. Note, that this period took place from the 8 March to the 22 March.

For testing purposes, S&P 500 stocks (assets) were selected to ensure there was a larger quantity of tweets to obtain sentiment. The selected stocks were Apple, Tesla, Facebook, Google, and Amazon.

Once the selected assets were chosen, the prototype was used to obtain the sentiment. Fortunately, this process was simplified because the deep learning system was implemented as a daily CRON job on Azure Functions, which meant the entire process was automated. Using this setup, the system pulled around 200 tweets down from the Twitter API every day for each asset, calculated the sentiment score, and then saved the results to the database along with a date time stamp. At the end of the two-week period, the web apps sentiment chart was displaying the trend of sentiment for each asset over the course of two weeks.

Another requirement to realize the evaluation was to obtain the stock market data for each asset over the same period. This was accomplished using Yahoo Finance.

However, it proved harder to compare the trend of the two datasets as no feature was designed in the prototype to visualize the two datasets in the same chart. Instead, images of the two charts were obtained and then compared with one another (See Figures 16–20). Although this means of comparison was still an effective solution, it was more time consuming, and it would have been more useful to have the stock market data in the same chart as the sentiment data.

4.2. Results and Discussion

Once all the sentiment data had been obtained over a two-week period, the results of each asset were compared with their corresponding stock market performance data. Figures 16–20 show the sentiment and stock price charts that were used to identify and analyze the trends. As the diagram shows, the sentiment and stock price of every asset is compared over a two-week period from the 8 March to the 22 March.

Overall, the results of the trends between the sentiment and stock performance are interesting and different patterns can be identified in each comparison.

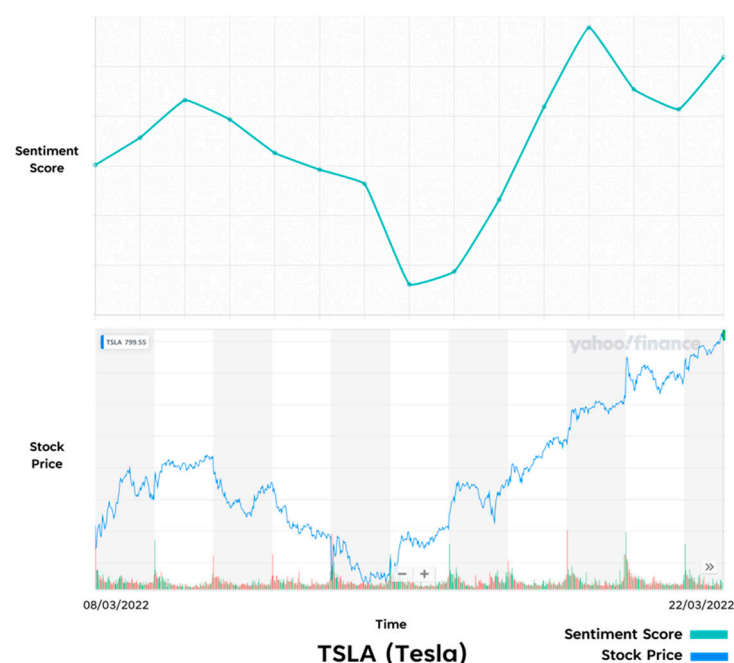


Figure 16. Comparison of Prototype results and Real-Life results—Tesla.

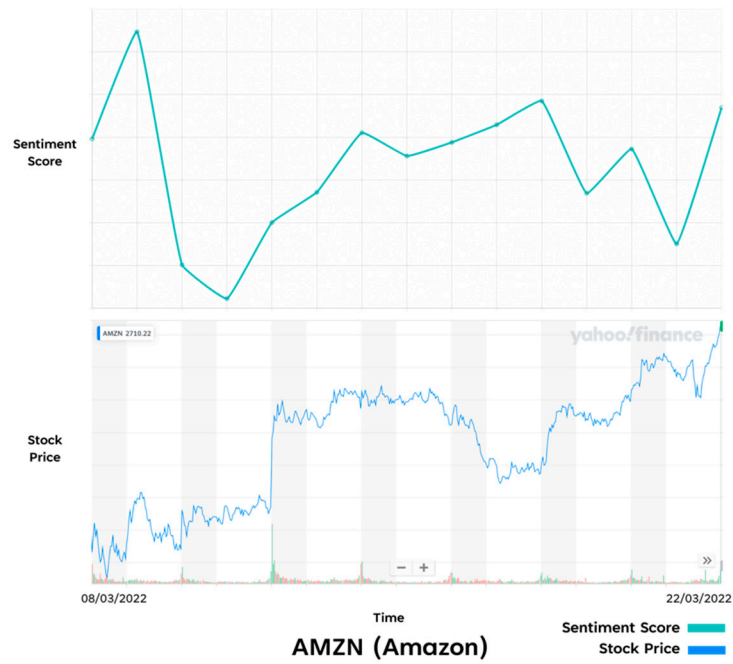


Figure 17. Comparison of Prototype results and Real-Life results—AMZN.

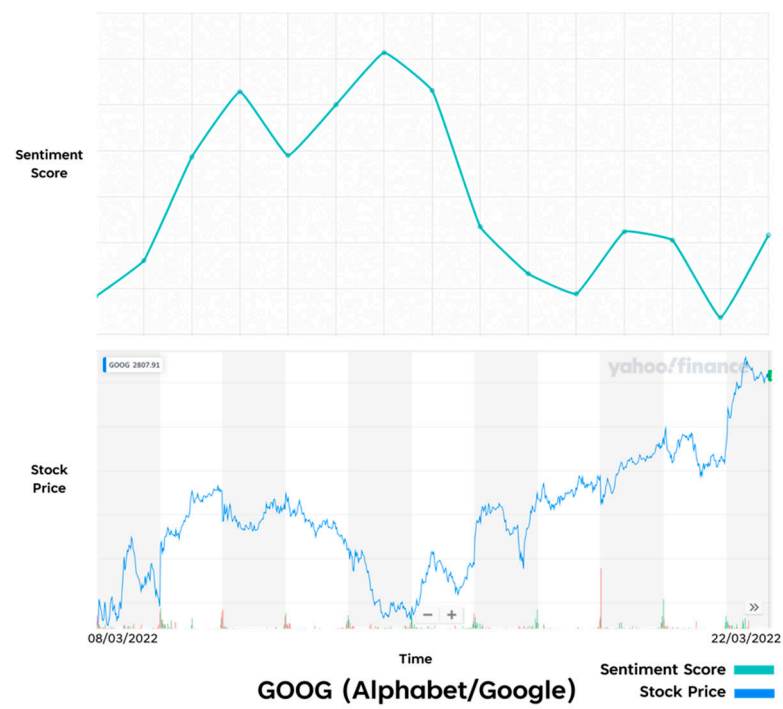


Figure 18. Comparison of Prototype results and Real-Life results—GOOG.

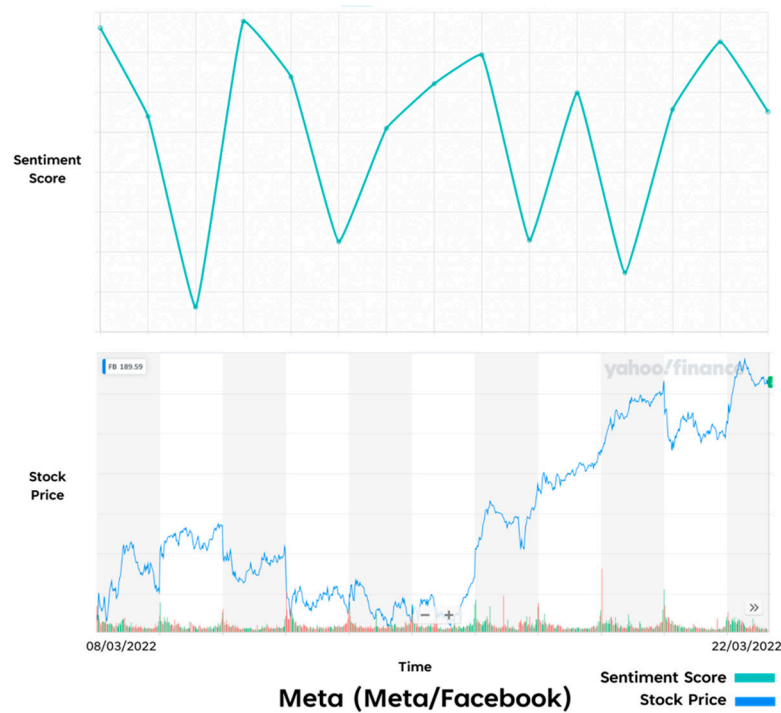


Figure 19. Comparison of Prototype results and Real-Life results—META.

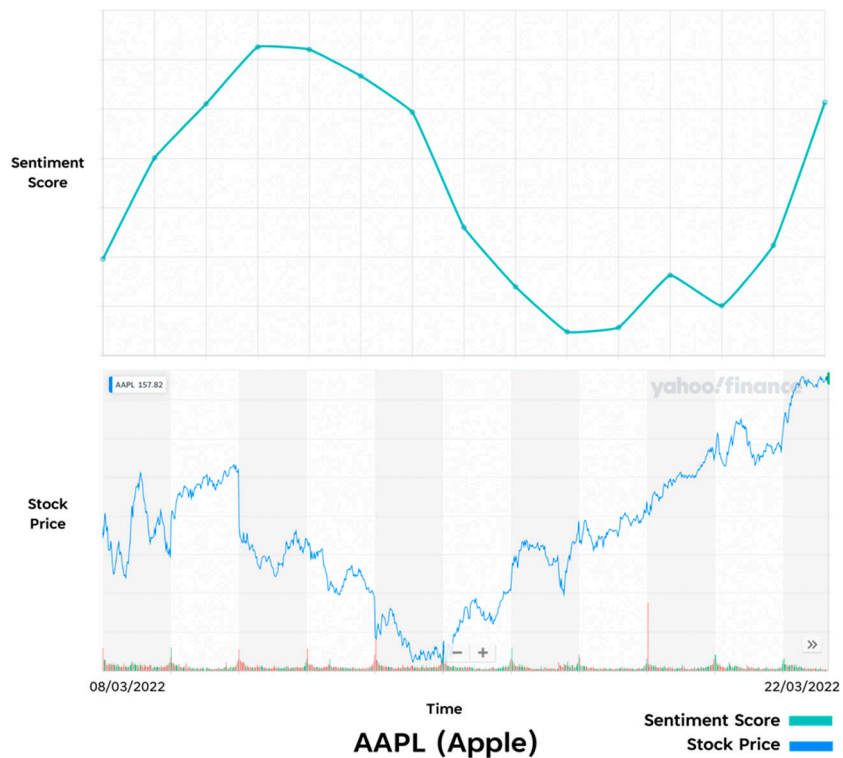


Figure 20. Comparison of Prototype results and Real-Life results—AAPL.

One of the most interesting results is the similarity of trends in the Tesla stock (See Figure 16). A very similar trend can be seen that shows both the sentiment and stock value having an increase and then decrease in the first week, and then a much steeper increase in the second week. Interestingly, Tesla’s stock price has always been a popular topic of discussion on SM, which could be why the sentiment score was so accurately mapped to the stock market performance.

The Apple (See Figure 20) and Amazon (See Figure 17) sentiment scores also appear to map the stock market performance at points. In Apple's case there is a smaller similarity in the overall trend, but it seems to be delayed compared to the stock performance. This could potentially be that the sentiment obtained was more reactionary to the stock performance. Amazon's sentiment score shows a noticeable similarity to that of the stock price around the end of the first week, where both trend lines increase, with the sentiment score increasing just before a sharp rise in the stock price. Again, this could be that the sentiment data obtained for these stocks provided better insight into the underlying assets performance, but further testing would be required to prove this.

One of the weakest results in trends was Facebook (See Figure 19). The sentiment score was volatile for the duration of the two weeks with repeating peaks and troughs throughout the experiment, while the stock market trend shows a steadier decrease in the first week and an increase in growth over the second week. The weakness of these results could be down to factors such as the SA model struggling to identify the correct sentiment for the tweets related to these assets. Additionally, it could be that the twitter sentiment related to these assets are less indicative of the stock market performance in general compared to a stock such as Tesla.

Google (See Figure 18) also has a weaker trend relationship, in which the stock market performance starts low and ends high. Meanwhile, the sentiment score starts low and ends low. This again could be down the same reasons outlined above.

5. Conclusions

The work presented in this paper proves how SM sentiment can be utilized to help users make investment decisions and creates a base to allow further research to be conducted. To achieve this outcome, we strategically designed and implemented a prototype based on the findings in the literature review. Furthermore, a novel approach was implemented by factoring in a Twitter like weighting into the LRA's sentiment score, with the goal of accurately mapping the collective sentiment of users from single tweets.

We found that the trend of SM sentiment had a similar pattern with the stock market performance for some of the selected assets. Based on these findings, a user would be able to utilize the sentiment to make investment decisions.

We have also shown how to build a system that can calculate SM sentiment related to financial assets, and how it can be visualised in a way to help users make investment decisions. As a result, the work of this paper can be used as a base to further explore the field of research.

However, we also found cases in the results that did not show similarities as clearly. Going forward, we recommend a continued thorough investigation into how SM sentiment can be utilized and visualized to make stock market predictions, which would provide more insight into making investment decisions. One recommendation would be to integrate stock market data into the assets sentiment chart, so users can make comparisons between the two trends more easily. Another recommendation for the prototype would be to improve the deep learning system. To do this, different datasets should be experimented with to train the SA model. Additionally, different algorithms such as NN's should be tried and tested to improve the performance of the sentiment predictions. Furthermore, our prototype only factors in tweets written in English, and as a result, is exposed to far less SM data for each asset, reducing its effectiveness in tracking the correlated public sentiment. We suggest that further work is undertaken to update the system to accept multiple languages. Finally, this paper only tested against popular S&P500 stocks. Further analysis could be undertaken on less popular stocks to see how usable the prototype could be against a wider range of companies.

Author Contributions: Conceptualization, B.H. and C.C.; methodology, B.H. and C.C.; software, B.H.; validation, B.H., C.C. and N.P.; writing—original draft preparation, B.H.; writing—review and editing, B.H., C.C., N.P. and W.J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data was created.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mukhtar, N. Can We Beat The Stock Market Using Twitter? | by Noah Mukhtar | Towards Data Science. Towards Data Science. 2020. Available online: <https://towardsdatascience.com/can-we-beat-the-stock-market-using-twitter-ef8465fd12e2> (accessed on 1 November 2022).
2. Facebook MAU Worldwide 2021; Statista: Hamburg, Germany, 2021. Available online: <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/> (accessed on 1 November 2022).
3. Wołk, K. Advanced social media sentiment analysis for short-term cryptocurrency price prediction. *Expert Syst.* **2020**, *37*, e12493. [[CrossRef](#)]
4. Smailović, J.; Grčar, M.; Lavrač, N.; Žnidaršič, M. Stream-based active learning for sentiment analysis in the financial domain. *Inf. Sci.* **2014**, *285*, 181–203. [[CrossRef](#)]
5. Bollen, J.; Mao, H.; Zeng, X. Twitter mood predicts the stock market. *J. Comput. Sci.* **2011**, *2*, 1–8. [[CrossRef](#)]
6. Nasdaq. *How Does Social Media Influence Financial Markets*; Nasdaq: New York, NY, USA, 2019.
7. Atwater, P. *Moods and Markets: A New Way to Invest in Good Times and in Bad*, 1st ed.; Pearson: London, UK, 2012.
8. Dolan, R.J. Emotion, Cognition, and Behavior. *Science* **2002**, *298*, 1191–1194. [[CrossRef](#)] [[PubMed](#)]
9. Bossaerts, P.; Bossaerts, P.; Bossaerts, P. How Neurobiology Elucidates the Role of Emotions in Financial Decision-Making. *Front. Psychol.* **2021**, *12*, 697375. [[CrossRef](#)] [[PubMed](#)]
10. Padhanarath, P.; Aunhathaweesup, Y.; Kiattisin, S. Sentiment analysis and relationship between social media and stock market: Pantip.com and SET. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2019; Volume 620, p. 12094. [[CrossRef](#)]
11. Ni, Y.; Su, Z.; Wang, W.; Ying, Y. A novel stock evaluation index based on public opinion analysis. *Procedia Comput. Sci.* **2019**, *147*, 581–587. [[CrossRef](#)]
12. Xing, F.Z.; Cambria, E.; Welsch, R.E. Natural language based financial forecasting: A survey. *Artif. Intell. Rev.* **2018**, *50*, 49–73. [[CrossRef](#)]
13. Shead, S. Elon Musk’s Tweets Are Moving Markets—And Some Investors Are Worried. CNBC. 2021. Available online: <https://www.cnbc.com/2021/01/29/elon-musks-tweets-are-moving-markets.html> (accessed on 1 November 2022).
14. Porcher, S.; Renault, T. Social distancing beliefs and human mobility: Evidence from Twitter. *PLoS ONE* **2021**, *16*, e0246949. [[CrossRef](#)] [[PubMed](#)]
15. Chauhan, P.; Sharma, N.; Sikka, G. The emergence of social media data and sentiment analysis in election prediction. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 2601–2627. [[CrossRef](#)]
16. Renault, T. Sentiment analysis and machine learning in finance: A comparison of methods and models on one million messages. *Digit. Financ.* **2020**, *2*, 1–13. [[CrossRef](#)]
17. Times, E. Extent of Elon Musk’s Influence on Cryptocurrency; Where Is IT Headed? The Economic Times. 2021. Available online: <https://economictimes.indiatimes.com/markets/cryptocurrency/extent-of-elon-musks-influence-on-cryptocurrency-where-is-it-headed/articleshow/83037268.cms> (accessed on 1 November 2022).
18. Shashank Gupta. Sentiment Analysis: Concept, Analysis and Applications | by Shashank Gupta | Towards Data Science. Towards Data Science. 2018. Available online: <https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17> (accessed on 1 November 2022).
19. Shukri, S.E.; Yaghi, R.I.; Aljarah, I.; Alsawalqah, H. Twitter sentiment analysis: A case study in the automotive industry. In Proceedings of the 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Amman, Jordan, 3–5 November 2015; pp. 1–5. [[CrossRef](#)]
20. Barba, P. *Machine Learning (ML) for Natural Language Processing (NLP)*; Lexalytics: Amherst, MA, USA, 2020; Available online: <https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing#supervised> (accessed on 1 November 2022).
21. Brownlee, J. Logistic Regression for Machine Learning. In *Machine Learning Mastery*; 2016. Available online: <https://machinelearningmastery.com/logistic-regression-for-machine-learning/> (accessed on 1 November 2022).
22. Krouska, A.; Troussas, C.; Virvou, M. Comparative evaluation of algorithms for sentiment analysis over social networking services Real-Time Face Mask Detector Using Convolutional Neural Networks Amidst COVID-19 Pandemic View project Comparative Evaluation of Algorithms for Sentiment Analysis over Social Networking Services. *J. Univers. Comput. Sci.* **2017**, *23*, 755–768. Available online: <http://www.internetlivestats.com/> (accessed on 1 November 2022).
23. Prasad, A. Decision Trees For Classification (ID3) | Machine Learning | by Ashwin Prasad | Analytics Vidhya | Medium. Medium. 2021. Available online: <https://medium.com/analytics-vidhya/decision-trees-for-classification-id3-machine-learning-6844f026bf1a> (accessed on 1 November 2022).
24. IBM Cloud Education. What are Neural Networks?—United Kingdom | IBM. IBM Cloud Education. 2020. Available online: <https://www.ibm.com/uk-en/cloud/learn/neural-networks> (accessed on 1 November 2022).

25. Sohangir, S.; Wang, D.; Pomeranets, A.; Khoshgoftaar, T.M. Big Data: Deep Learning for financial sentiment analysis. *J. Big Data* **2018**, *5*, 3. [[CrossRef](#)]
26. Agarwal, S.; Kumar, S.; Goel, U. Stock market response to information diffusion through internet sources: A literature review. *Int. J. Inf. Manag.* **2019**, *45*, 118–131. [[CrossRef](#)]
27. Zhang, X.; Shi, J.; Wang, D.; Fang, B. Exploiting investors social network for stock prediction in China's market. *J. Comput. Sci.* **2018**, *28*, 294–303. [[CrossRef](#)]
28. Pathak, A.R.; Pandey, M.; Rautaray, S. Topic-level sentiment analysis of social media data using deep learning. *Appl. Soft Comput.* **2021**, *108*, 107440. [[CrossRef](#)]
29. Catelli, R.; Pelosi, S.; Esposito, M. Lexicon-Based vs. Bert-Based Sentiment Analysis: A Comparative Study in Italian. *Electronics* **2022**, *11*, 374. [[CrossRef](#)]
30. Pota, M.; Ventura, M.; Catelli, R.; Esposito, M. An Effective BERT-Based Pipeline for Twitter Sentiment Analysis: A Case Study in Italian. *Sensors* **2021**, *21*, 133. [[CrossRef](#)] [[PubMed](#)]
31. Catelli, R.; Fujita, H.; De Pietro, G.; Esposito, M. Deceptive reviews and sentiment polarity: Effective link by exploiting BERT. *Expert Syst. Appl.* **2022**, *209*, 118290. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.