Course code and name:	F20PA
Type of assessment:	**Individual**
Coursework Title:	Dissertation
Student Name:	Harshal Shinoy Thachapully
Student ID Number:	H00335729

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

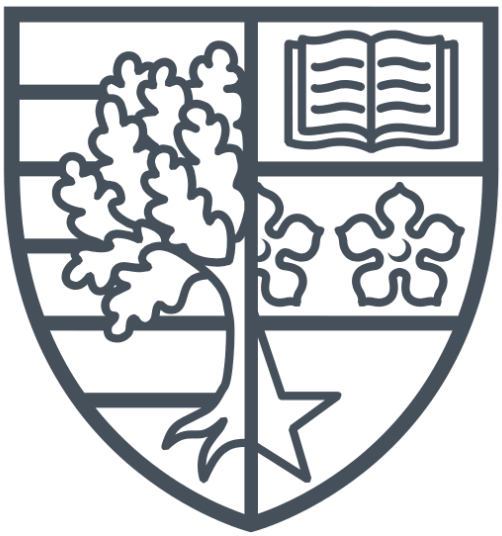
- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Harshal Thachapully*

Date: *20/04/2022*

Predicting Stock Movement Using Sentiment Analysis on Tweets

Harshal Thachapully
Heriot Watt
BSc (Hons) Computer Science
Final Year Dissertation
Supervised by: Prof. Hadj Batatia



Author Note

I, Harshal Thachapully, confirm that the following work that is submitted is of my own and expressed with my own words. There will be proper acknowledgement of authors that are used in any form (e.g., ideas, equations, figures, text, tables, programs, etc.). A bibliography including all the references used will be provided as well.

Signed: Harshal Thachapully

Date: 20/04/2022

Abstract

Stocks trends are generally considered extremely difficult to predict historically, inspiring many to investigate in different manners to resolve the issue. We contribute to this phenomenon by proposing the investigation of the prediction of stock trends by considering public sentiment from Twitter. We treat the trend prediction problem as a machine learning classification problem through the prediction of the direction of the trend of the next day. By taking into account of several different features, such as the past price history of the stock and its technical indicators, and engineering some features, such as the sentiment obtained by using Vader Sentiment, we try to classify if the trend is moving up or down. We consider different normalization methods and utilize 5 different machine learning models from Sci-Kit Learn, Logistic Regression, Random Forest, Gradient Boosting Classifier, Gaussian Naïve Bayes, and SVM. The best performing models were Logistic Regression and Random Forest, with Random Forest generally obtaining the highest scores and Logistic Regression being the best once considering the standard scaler.

Acknowledgements

Firstly, all praises and thanks are to Krishna the Supreme Personality of Godhead for providing me the ability, strength, and mental fortitude to complete this project. Jai Shri Krishna,

Firstly, I would like to thank my supervisor, Hadj Batatia, for helping me with the research, development, and guiding me on writing my thesis. Your knowledge has led me to stay on track and provided great feedback, for which has been immensely valuable and for which I am tremendously grateful for. I would also like to thank Idress Ibrahim for providing extensive and detailed instructions to guide me to building a better dissertation all together

I would also like to thank my family for not just their support during the project, but throughout the time I have passed in this course. I would like to express my utmost gratitude to my family, my mother, Geeta, my father, Shinoy, and my brother, Harihar, especially for helping me throughout this journey and providing me the mental and physical support without which I just could not do.

Contents

1. Introduction

Trying to understand the stock market has been a recurring issue since its inception and many theories surrounding the topic which try to provide an understanding to the issue. One such theory suggested a Random Walk Hypothesis where stock market prices do not have any inherent patterns and that past information about the stock cannot be exploited to predict future prices [1]. On the other hand, The Efficient Market Hypothesis (EMH) [2] suggested that prices of a particular stock are an immediate reflection of its past, thus using analysis of recent performance of the stock, ideally it would be possible to predict the future price.

With the conception of the interest arose several social media platforms where people can share their personal opinions and life on the internet, of which Twitter is the most popular and influential opinion sharing platform. With 140+ active users that share 400+ million tweets every day, which include trending matters on a range of topics and fields, thus also acts as a valuable source to obtain relevant information. With tools such as the Twitter API and sentiment analysis to be able to analyze, Bollen [3] was the first to implement and use Twitter as a mood indicator as an investor sentiment index and managed to predict the stock price with an accuracy of 87%.

In this dissertation we will create 2 models that will incorporate sentiment analysis upon several tweets and machine learning to predict the movement of the stock price as well as forecasting the stock price. By being able to monitor the sentiment of tweets, we can extrapolate a relationship between the sentiment shown by the public on Twitter and its relationship to the direction of how the stock prices move.

1.1 Hypothesis and Goals

The hypothesis for this project is that the use of tweets, which contain opinions about different companies held by the public on the hyper popular platform, Twitter, will influence how stock movement occurs and thus would be predictable.

1.2 Objectives

- Investigate methods in predicting the stock market movement
- Build a clear understanding about sentiment analysis and sentiment analysis methods

- Learn and consider methods used for prediction which include sentiment analysis and stock predictions through the literature review
- Build a predictive model to predict the direction of the stock based on historical data from the stock and a comparative model which uses sentiment as an additional feature
- Vary different parameters and features in the model to achieve better results
- Compare and contrast the effect of sentiment analysis as well as other manipulated parameters and features with the prediction using methods not using sentiment analysis
- Explore different limitations of the developed model and provide suggestions for possible improvements and implementations over the current model.

1.3 Project & Research Questions

- What sort of relationship exists between tweets and stock prices?
- Is it possible to construct a model which predicts the movement of the market using tweets?
- How can we improve the model?
- What limitations does the current model have?
- Ethical and legal issues that are present when making the model?

1.4 Report Structure

The structure of the report after the following introduction will include Background information about stock markets, different machine learning and deep learning techniques as well as understanding Natural Language Processing. Followed by the Background, we will conduct a literature review where we will survey different papers that are in relation with stock market predictions using machine learning as well as different papers that have researched into Sentiment Analysis and used it in predicting the movement of the stock market as well as forecasting stock prices. We will then overlook a Method which would generally outline what the experiment will consist of and then finally we will review the Implementation and discuss the Results procured through experimentation. We will end the project by concluding about the project and how the results correlated with the hypothesis.

2. Background

2.1 Stock Market Analysis

2.1.1 Introduction

Stock trading has a long tradition and can be traced back to around the 1500'S in Antwerp. The stock market is a place where public markets engage in selling, buying, and providing stocks on the stock exchange. Each stock is a fractional representation of ownership within the business that is providing the stock; this stock is bought and sold by investors.

Stocks are crucial for two reasons: Firstly, they can provide capital, through equity crowdfunding, for a company to aid the expansion of the business, thus avoiding the potential to be in debt. Secondly, investors can profit from buying stocks, and this is realized in two different manners, one being where they are paid a regular dividend and the other being where they sell stocks at higher value than initially bought. Due to the nature of the stock market, which contains non-stationary and chaotic data, it provides a challenge to predict whether the prices will rise or fall. Thus, several theories and methods were born in the pursuance of understanding how the stock market functions.

2.1.2 Conventional Stock Prediction

1. Efficient Market Hypothesis

The EMH theory [29] states that stock prices are an instant reflection of new information that enters the market, thus any trading strategy (e.g., fundamental, and technical analysis) lack the potential for bringing any excess profits as the market is already efficient, therefore the idea of beating the market is impossible. The stocks that are traded are traded on a fair market value, but to make any form of profit, the investor should invest in a passive portfolio.

2. Fundamental Analysis

When one uses Fundamental Analysis to predict the future of the stock price, they consider metrics that relate with the business's (a company that is listed publicly) financial statements. This provides an in-depth analysis on the current financial health of the business. By a thorough analysis of a business's financial statements, the analyzer deducts a score, from which if the score is less than 1, it

is known that business is not financially stable. Such factors are important to consider for understanding the capabilities of the business and if in the short term as well it can live up to the obligations it has set out for e.g., paying dividends.

3. Technical Analysis

When considering a technical analysis approach, one tries to extrapolate a possible future outcome through investigating the past and current performance of the business in question. Charts are a key tool in aiding the understanding of the company's performance. This method however is only possible when supply and demand are in direct correlation with the stock's prices as external factors such as stock splits, mergers, etc. will not lead to a successful prediction.

2.1.3 Computing approaches to Stock Prediction

With the rise of computers, and progressively over the years more powerful computers, the potential to compute higher amounts of data also incrementally increased. Thus, the ability to make sense of a large set of seemingly chaotic data in the stock market could be possible through machine learning. Machine learning, such as using Support Vector Machines and Artificial Neural Networks, recognize patterns in data that can seem oblivious and possibly forecast financial markets for us.

2.1.4 Conclusion

To conclude this section, we have explored the understanding of a stock market and its relevance in the world today, and several hypothesis and analysis techniques, some being conventional and some being modern and computational, used in stock market analysis and prediction today.

2.2 Machine Learning

2.2.1 Conventional Machine Learning

Support Vector Machines

It is an algorithm capable of doing regression as well as classification (linear and nonlinear) problems. When acting as a classifier, it places a hyperplane [5] which is a line of separation between data, or also known as the decision boundary, which allows us to distinguish data from each other and putting them in their respective groups. The hyperplane is placed according to the furthest away from the

nearest points which are also known as support vectors. They have applications in sentiment analysis [6][7], image detection [8] and many other fields.

Naïve Bayes

The Bayes Classifier is a statistical classifier which takes into consideration different known factors and calculates accordingly providing the probability of the outcome at question. An assumption made of the features that will be processed is that they are all independent, thus the existence of one feature will not adversely or positively affect other features. It can be used on problems such as text classification [9], sentiment analysis [10] as well as spam detection [11].

Random Forest Trees

It is an ensemble method that is used to solve regression as well as classification problems. The way it functions for classification, is it creates multiple decision trees simultaneously during training time and chooses the output democratically based upon the number of decision trees that provide the same output. In the case for regression problems, it continues to create multiple decision trees from which an average is taken as the prediction [12]. It is used widely, especially for image detection [13].

2.2.2 Deep Learning

Multilayer Perceptron Neural Networks (MLPNN)

Due to being straightforward, it is known as the vanilla of neural networks. It consists of three principal layers which are: 1. Input Layer, 2. Hidden Layer, 3. Output Layer. Whilst training, MLPNN utilizes a supervised learning technique known as backpropagation [14]. Every node that is known as a neuron has utilized a nonlinear activation function which is used for creating and recognizing distractions within data which can be separated in a linear fashion.

Recurrent Neural Networks (RNNs)

RNNs are used for sequential or time series related problems. They use ‘memory’ to understand information obtained earlier to improve its understanding of the current input and output. It works well for shorter sequences as it can consider the immediate past and how it affects the immediate present, but it lacks the ability to reconcile such information with longer sequences as it will lose its

‘memory’ thus it is dubbed as having ‘short term memory’. RNNs are usually used for handwriting recognition [15] and speech recognition.

Long Short-Term Memory (LSTMs)

Built upon RNNs, it solves similar problems. It was developed to combat an issue that RNN’s faced with longer sequences, consequently it was an improved model which could process longer sequences of information without forgetting, therefore making better sense of current input and output data than an RNN. Uses include time series forecasting, NLP, speech recognition, and anywhere where temporal information is the essence of the information.

2.2.3 Conclusion

In conclusion we have investigated two types of machine learning that is popular today: Conventional Machine Learning and Deep Learning. Within both, we have discussed several algorithms that are popularly used and understood how they are used and how each of them function.

2.3 NLP & Sentiment Analysis

2.3.1 Introduction

During our research, we encountered several papers which used NLP as a form of analysis for textual data and sentiment analysis to forecast stocks, thus in this section we aim to understand the different parts of NLP and provide a short introduction to what NLP is.

NLP is a subfield within Artificial Intelligence, providing the computer the ability to derive meaning from human language in spoken and written form. Human languages contain several features such as idioms, sayings, metaphors, sarcasm, tones, the surrounding context affecting the way certain words are used, grammar, grammatical irregularities and much more. These features that make up a language take several years for humans to comprehend right from scratch.

A lot of sentiment analysis research has been conducted on social media platforms, more specifically Twitter. Twitter is a platform where people share 'Tweets' which are small bitesize pieces of text that contain ideas and opinions shared by the author. Tweets can be responded to and shared several times

by similarly opinionated people. Thus, certain ideas assert higher levels of influence than other tweets, causing some to prevail over others, consequently more popular tweets have a stronger influence on the public. Such influences are crucial to stock forecasters who aim to predict stock values according to public perception of the company the stock is provided by. Therefore, with potential of a strong correlation between the stock prices and the potential of influence from Twitter on these stock prices, we have decided to explore the relationship using sentiment analysis.

2.3.2 Different Applications of NLP

- **Autocorrect and Autocomplete**
    - Serves to correct grammatical mistakes that are made as we type or text. It also predicts and provides text after analyzing the surrounding context of the sentence.
- **Targeted Advertising**
    - By analyzing the keywords used by a user, advertisements are presented that are associated with those keywords, also known as keyword matching. It makes it easier for businesses to reach potential customers.
- **Voice Assistants and Chatbots**
    - Through Speech Recognition, the computer converts sound data, captured through a microphone, into a text or a format comprehendible to the machine. After making sense of the input, it can respond accordingly according to preprogrammed rules and patterns.
- **Email Filtering**
    - By analyzing the text and classifying it according to the keywords it provides, it places it in groups which correspond to the content of the email. This implemented in Gmail where Gmail segregates the content into Primary, Social and Promotional groups.

- **Sentiment Analysis**
    - By analyzing certain words that provide a certain context. Words are provided a value based on the intensity and type of sentiment they project. When all the words come together, the program can deduct what the potential sentiment of the text may be.

2.3.3 Limitations of NLP

- **Ambiguity**
    - Deciphering the meaning of the text is a challenging task through a thorough analysis of the words used. To counteract this problem, the surrounding context is analyzed as well as to clear the meaning of the text.
- **Multiple meanings**
    - Depending on the context of the text, the user could say something which means different things in different contexts. This is often prominent when slang words are considered, e.g. if someone says: 'You have a good drip', ‘drip’ here symbolizes an outfit but in the context of a liquid, it means that the liquid is literally dripping.
- **Multiple Intentions**
    - It is harder to identify the tone being used in text when compared to a voice note, this issue is also prominent in conversations between two humans where sometimes the tone of the text is ambiguous. The AI requires to be able to distinguish intentions in text format as texts can often have an array of different intentions.
- **Objectivity and Subjectivity**
    - Being able to distinguish between an event that has an opinion behind it in comparison to an event which is held in regard as being factual.
- **Sarcasm and Irony**
    - Being able to distinguish between sarcasm

2.3.4 NLP Pipeline

When processing text, we must go through a series of data preparation steps to feed the data to a Machine Learning model or a Statistical Model, thus we will explore what steps go into pre-processing the data before the data is used to predict.

Text Processing

Before we convert the text into numerical values, it imports to process the text to get rid of unnecessary parts and identify certain parts to make the feature extraction step easier to handle.

- **Cleaning** - removing irrelevant
- **Normalization** - all words made lowercase and removal of punctuation and extra spaces
- **Tokenization** - data is split into tokens, each token represents a word
- **Stop words removal** - most common words, such as connectives like ‘and’, ‘a’, ‘am’, and ‘the’ are removed
- **Parts of speech tagging** - parts of speech are identified for the rest of the words
- **Named Entity Recognition** - Recognizing names in the data
- **Stemming and Lemmatization** - transforming words into their dictionary definition
    - **Stemming** - word is reduced to its root form
    - **Lemmatization** - reducing words to a normalized form and map the words variants to a common root.

Feature Extraction

Computers don’t contain a standard representation of words, thus many ways of representing textual information have developed to encompass relationships between words. For different texts, depending on their size and complexity, different models have been developed to comprehend and accordingly

provide a 'meaning' to the text so as not to mislead the NLP algorithm. Each word is defined as a feature and the way we decide what values each feature will hold is completely dependent on us.

Bag of words

Each piece of text is converted into a vector of numbers where a count is kept of each word (each word has equal importance) and the number of times it occurs, eventually this set of numbers becomes a corpus, from which you can determine the context for vectors that are calculated.

Term Frequency — Inverse Document Frequency

Doesn’t treat each word as equally important as it counts how often each word shows up by searching it in several documents giving us the document frequency of the word. The document frequency is used to divide the number of times a word shows up, thus giving us a metric to give value to a certain word in comparison to other words.

One hot encoding

Each sentence is represented as a binary value of 1 or 0. 1 indicated that the word exists within the document, and vice versa for 0. Neural networks often utilize one hot encoding but since it represents every word within the sentence, it can lead to too much complexity when larger text is being dealt with as it can include too many input neurons.

Word Embeddings

Essentially converts real values vectors that are largely sparse and places them in lower dimensional space which helps keep intact the semantic relationships between words. Thus, words that have a closer semantic meaning can be found closer together.

Modeling

Once you prepared all your data and made them numerical (which is now machine learning and statistically friendlier), you can plug the data into a statistical or machine learning model and eventually use those models to see how well each model reacts with the training data and to what level can it provide an accurate understanding of test data.

2.3.5 NLP Approaches

Rule Based

It is system of rules constructed upon linguistic structures which are commonly and consistently found within the grammar that's present in the language [16]. Placing rules provides computers to deduce meaning by understanding basic grammar rules and differences, for example the different between nouns and verbs, or words that having specific endings such as -ing, -tion, -ness. Although this is easily adaptable and has a higher rate of accuracy, it requires proficient linguists and developers to develop such a system.

Conventional Machine Learning

With statistical methods and algorithms, it analyses data and develops its own understanding of the data, thus 'Learning' and accordingly make associations based on past data that it has been provided to make sense of the current data. Though it has the potential to be scaled easily and applied generally. Issues generally faced through this method as it does not actually develop a proper understanding of the language, it can suffer the butterfly effect, where a small change in the text can completely change the prediction of the meaning of the text.

Deep Learning

Using neural networks and deep learning methods, it mimics our brain function. Essentially through a large corpus of data, it infers rules and its own understandings of the data by building system from a

large pool of text and learns from those training samples. Using neural networks, we can avoid having to do feature engineering as the network will identify key attributes, moreover the larger the size of data, the more accurate it will be.

2.3.6 Conclusion

Within this chapter we have discussed about NLP, current limitations that plague NLP as well as considered different approaches to how one can use various models for analyzing as well as predicting data. Moreover, we have discussed the entire NLP Pipeline in detail, outlining several methods and the different steps in preprocessing the data before it is placed into a model to help us predict our data.

2.4 Literature Review

2.4.1 Machine Learning in Stock Market Predictions and Forecasting

The use of algorithmic trading has lately boomed due to the potential computational power presents to calculate large amounts of data. In paper [17], they took advantage of machine learning methods and investigated into predicting the stock market. They investigated predicting the Trend by using single feature and multiple features with classification and regression algorithms. They found that SVM's performed worse by getting a 21.6 rmse value in comparison to the Generalised Linear Model (28.7 rmse) and Linear Regression (24.8 rmse). But when considering multiple features, especially the best correlated features, SVM (74.4% accuracy) did better than a regression algorithm called MART (70.4% accuracy), but when all features, well correlated and badly correlated, were considered SVM performed considerably worser than MART by 10%. This would suggest that SVM's are sensitive to the number of features that are used when used for prediction.

Paper [18] had reached the conclusions that results using Linear Regression Model have a better accuracy rate after PCA (Principal Component Analysis) is applied to pick the most relevant features. SVM showed a high accuracy data that was of a non-linear classification nature, on the other hand Linear Regression performs better for linear data due to the high confidence value during prediction. When considering binary classification, MLP and Random Forest provided the highest accuracy rate and lowest error in the process of making predictions.

In paper [19], neural networks are used to forecast stock prices. An MLP (Multilayer Perceptron) and an Elman Recurrent Neural Network are used and are trained using a back propagation algorithm. They took into consideration, the lowest, highest, and average value in desired number of past days for training so they could forecast the price and the results of the neural network were compared to the results obtained from Linear Regression. What was found that the neural networks that were employed performed worse than the Linear Regression model, but the error for MLP was lower than that of the Elman Recurrent NN and Linear Regression suggesting a higher level of reliability over the result that MLP obtained.

In paper [27], we observe the use of SVM's to predict stock market movement and its comparison to the Random Walk Model, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Elman Backpropagation Neural Networks (EBNN). SVM had the highest predictive accuracy compared to the other models that were employed. One reason for this occurring is due to the architectural design of the algorithm which aids in minimizing structural risk; thus, it is less vulnerable to overfitting.

Conclusion and Critical Analysis

There were a variety of methods employed in determining of the movement of the stock market. [17] highlighted the sensitivity of SVM's when employed in scenarios where there were too many unwanted variables, thus degrading its performance. [18] displayed PCA enhancing the performance of linear regression, but a criticism of this would be PCA cannot be applied in all situations as some situations would have a small sample of variables which are being investigated, thus Linear Regression overall may not be an optimal model to employ. On the other hand, I found it informative that Random Forest Trees could predict with high accuracy and low rate of error. An overall criticism I would have like to also see be explored would be the effect that scalers have on the data and how that effected the predictive power of the models employed.

2.4.2 Sentiment Analysis

In [20] they predict the sentiments of tweets. They tokenize all the words in given tweet and place it an array, replace all the punctuation within the tweet as they are unwanted characters, creates a JDBC-

ODBC connection for storing all the tweets as well as removing unwanted characters and words that display inconsistencies. They match words in the array from a dictionary that they provide and according to that they provide a sentiment score. The sentiment score is calculated by:

$$\text{Sentiment Score} = \frac{N(Positive\ Terms) - N(Negative\ Terms)}{N(Positive\ Terms) + N(Negative\ Terms)}$$

And to calculate the error:

$$\text{Error} = \text{Actual Score} - \text{Calculated Score}$$

From the results that were collected, the tweets were analysed with the model were analysed with an 80.6% accuracy with a time complexity of O(m*n).

In [21], VADER was a simple rule-based model that was developed after considering building on top of several previously well-established lexicon dictionaries. It was developed and specialised towards calculating the sentiment of microblog and similar content, thus the lexicon base includes lexicons such as commonly used internet slang as well as emoticons which are uniquely associated with microblog like content, making it well attuned to produce sentiment values in scenarios including social media. There were 5 major grammatical and syntactic rules that were applied on the model that were found through the study of several hundreds of tweets which provide the characteristic that a tweet may possess, thus in combination with the lexicon dictionary along with the 5 rules, it was tested upon 4 different scenarios, out of which in three it performed the best just behind humans, and in the case of classifying sentiment from tweets, it outperformed humans.

[31] looked to classify sentiments of words and sentences. For to classify the sentiment of the word (a binary classification) they included synonyms and antonyms, thus providing more data for the model train upon. In the case of the Sentence Sentiment Classifier, the sentiment is dependent by the person or organization and the claim made by them. Thus, sentiment can be found more reliably in accordance with the holder of the opinion, otherwise it is harder to classify the sentiment. The found that from their English word list, they managed to achieve a result of 75.66% for Human 1 and the Machine and 77.88% as a lenient predictor alongside with a high recall rate suggesting the model, they developed is very reliable.

Conclusion and Critical Analysis

The main take away from here was the effectiveness of VADER in [21] and the great generalisability it possesses for calculating sentiment over a variety of different situations, especially when considering social media. Although [31] had a good model it had some limitation that were needed to be taken into consideration, one being that they used a unigram model which would not be sufficient by itself as there are certain common words that do not carry any form of sentiment by itself (e.g. 'and') but instead the sentiment is determined by the context it is present in. When considering its method for the sentence sentiment classification, an issue found was that models cannot infer sentiment from facts in each sentence as the lack of using verbs, adjectives, and nouns in a sentence wouldn't provide enough information in order for the system to categorise the sentiment of the holder. One more limitation includes that detecting who the holder of the given opinion is difficult to identify, therefore the system holds the potential to choose the wrong holder when the opinion has a possibility of belonging to several others.

2.4.2 Predicting Stock Market Movement Using Sentiment Analysis

In [22], the trend was predicted, and the way the trend was determined was by creating two different trend features. The first being the today trend which was calculated by considering the open and close price of the same day and based on the value it was classified as an upward or downward trend. The other trend that was calculated was considering the closing price of tomorrow and taking it away from the closing price of today, given the result, it was classified either as an upward or downward trend. A Relative Strength Index, Simple Moving Average, and Stochastic Oscillator indicator were also considered. Sentiment analysis was done using the Vader Sentiment Analysis tool which it would apply on the news headlines it had collected. They also performed a feature correlation analysis through which they recognized that Close, Open, High, Low, Adj Close and SMA were the highest correlating features. From the results procured, it found that the best predictive was Random Forest with an accuracy score of 65% on the training set and 64% on the testing set, as it outperformed other models that were also used in metrics such as Accuracy, Precision, Recall, and F-Score.

In [23] we see a different approach to sentiment analysis, where instead of using VADER, they instead made their own sentiment classifier using NLTK (an NLP python library), we can see this the paper [26] as well. They classified tweets into positive, negative, neutral from the 250,000 tweets they had procured. They validated it by taking a sample from the large tweet database that they had, had it labeled by humans and then ran a Random Forest Classifier upon it and compared how the classifier performed using N-gram and Word2vec textual representations. For stock to sentiment correlation, they let sentiment aggregate for 3 days and the increase and decrease in stock price was denoted by a 1 or 0. Using LibSVM, when the model was trained on 90% of the data, the accuracy was 71.82%, thus showing a strong correlation between stock and sentiment.

In [24] and [25], different NLP models are explored and are applied before being placed into a placed into a predictive model namely being: LDA-based, JST-based, and Aspect-based. These models were then compared to the performance of models that considered sentiment classified data from both a sentiment predictive model and human based as well as considering solely the price itself. From all the models that were employed, the best model was the Aspect-based model which got an average best result of 54.41%.

In the case of [26], they performed sentiment analysis in a similar manner to [23] but instead the collected data was not based on Twitter but another social media Twitter like platform called Stock Twits. This is a platform specialized in talking about stocks in general. They used sentiment analysis which was performed using TF-IDF and logistic regression and applied the sentiment on a time series model which considered an aggregate of sentiment from the past 5 days. They were able to demonstrate a slight increase in prediction from without using sentiment thus showing a slight correlation between sentiment and stock price trends.

[28] creates their own sentiment dictionary much like [23], [24], [25], and [26] and classifies tweets in a similar manner to [23]. Although the general method of the paper is similar, it focuses more on individuals such as influencers and speculators more than just taking the public sentiment. They found an average accuracy rate of 38.5%. Using Random Forest, they reached 41% and with K Nearest Neighbours they managed to reach 42%. They maintained the Gaussian Process and Quadratic Discriminant as default values to compare against, which was 36% accuracy. When using RBF

kernels SVMs, they managed to achieve 58% accuracy. They also considered threshold optimization as they model as they found out increasing the threshold increased pseudo accuracy as neutral prediction increased simultaneously as well.

Conclusion and Critical Analysis

An overall criticism as I had provided earlier would be that the effect that scalers have on the data should be explored more because some scalers would perform better than the others. [22] had some interesting methods on calculating what trends it would be considering and how it affected the predictive power of the model as well as how well Random Forest Trees performed which corresponded to the findings found in [18]. [22] also had used VADER instead of a custom-made sentiment classifier model, which in contrast [23], [24], [25], and [26] had not done. [23] provided an insight into how different train test sizes could affect the model performance. [25] and [26] had explored interesting NLP models which was used for carrying out sentiment analysis and then the results being placed in a predictive model, showing that the Aspect-based model performing the best. I believe that [26] showed an interesting insight into using StockTwits instead of Twitter which allowed to see if there was an interesting correlation between StockTwits and the stock prices and that of Twitter. Although it is a specialised platform which is meant to be all about stocks, I believe the sheer reachability and influence that Twitter carries is unparalleled, especially when compared to platforms such as StockTwits. Moreover, users on Twitter would talk about a range of things which would involve the business itself, whilst StockTwits would be more focused on the stock itself, thus Twitter would be a better candidate as a representation of public sentiment to take into consideration when investigating a correlation between stock trends and public sentiment.

3. Methodology and Requirements

3.1 Proposal

This dissertation proposes predict stock market movements by using sentiment procured through sentiment analysis on Tweets as an additional feature to our classification models.

3.2 Methods Adapted

The method used by [22] will be considered where some features showed a higher degree of importance and correlation and compare the results before and after using sentiment analysis. Moreover, considering the accuracy of VADER on microblog content from [21], thus it would be beneficial to consider using its sentiment analyser model for deriving sentiment. We are going to classify tweets as [23] did and consider the predictive models used in [17] and [18] for binary classification. [22] also used a Minmax Scaler, so I would like to experiment with seeing the effect with different scalers.

3.3 Proposed Model

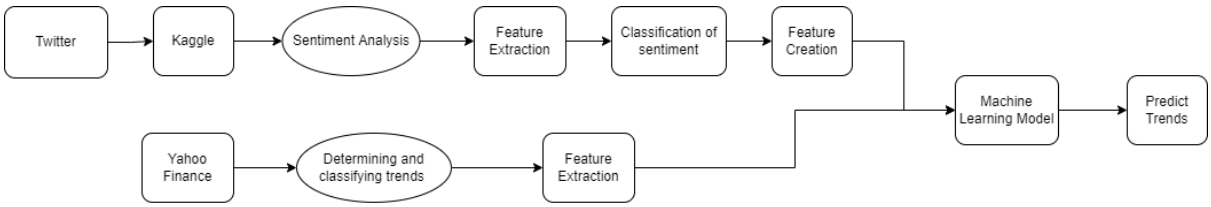


Figure 1: Implementation Overview

Stock data from each respective company will be pulled from Yahoo Finance and be processed to find the percentage change between accordingly to classify the trends which will become the eventual target variable. The most important features will also be extracted from the extracted datasets.

Tweets will be pulled from a publicly available dataset on Kaggle which was scraped using an open-source Tweet Scraping library. The extracted data will be processed through sentiment analysis and will be averaged for each day. Numerical features present on the dataset will be used to gauge their correlation with the Adjusted Close price to view as well as will be used to enrich the data.

All the data will be processed through different normalization methods by using different scalers before being passed into the machine learning models to predict trends.

3.4 Experiment Protocols

We have designed a protocol that we will be following involves testing different parts of the predictive model. Since there are several components and features, we will break down the testing for all of them.

- **Sentiment Analysis**: For sentiment analysis, we will investigate if the predictive model performs better by adding sentiment.
- **Sentiment Distribution Threshold**: Considering different thresholds for the determining which tweets will be classified as positive and negative.
- **Normalisation Methods**: Comparing different scalers, mainly Standard Scaler and the Minmax Scaler.
- **Classification Models**: Comparing several different classifiers: Decision Trees, Random Forest, SVM, Gaussian Naïve Bayes, and Gradient Boosting Classifier.

To gauge how well our model performed, we will be using 3 main metrics:

- ROC-AUC
- Accuracy
- F-1 Score

3.5 Functional and Non-Functional Requirements

Functional Requirements are crucial to outline the construction of the project and encapsulates the goals of this project. The goals are set up are present to present the process through what milestones were required to eventually reach the final point of the project. Below are the Functional Requirements for the project:

Key for requirement tables	
	Completed
	Provided an alternate solution
	Did not complete

Functional Requirements

Functional Requirements			
FR No.	Functional Requirement	MoSCoW	Current Update
1	Create a model for predicting trends for a particular stock	Must have	
2	Procure a tweets dataset for each of the individual stocks being investigated.	Must Have	
3	Make sure that all the datasets are cleaned and contain data related to the project.	Must Have	
4	Provide an overview of the fluctuations between tweets and stock	Could Have	Did not create a graph to present overview, showed graph of sentiment over time and graph of stock price over time.
5	Insights on accuracy, F1-score, and ROC-AUC score on trend classification for stock movement	Must Have	

Non-functional Requirements

Non-Functional Requirements			
FR No.	Functional Requirement	MoSCoW	Update
1	All code is publicly available, and all the technologies used are open sources and are available publicly as well	Must have	
2	Security of personal information is kept safe and replaced with unique ID's	Must Have	
3	Quality measures to ensure reliability of the model that is being created and tested	Must Have	
4	Use GitHub as a form of version control	Could Have	Stored multiple copies on local computer due to the size of files being far to large
5	Optimize code for the project	Must Have	
6	Documentation of the project throughout project lifeline including testing and training phases	Must Have	

4. Technical Implementation

4.1 System Requirement and Tools

To make sure everything will work as expected, the following software is needed:

- Windows 10 64-bit
- Jupyter Notebook Version
- Python 3.8.8

The python libraries needed are:

- NumPy 1.19.5 - it a scientific computing library in Python which aids in providing multidimensional arrays and fast mathematical operations which would not be solely available on python itself
- Keras 2.6.0 - Keras is an open-source library used for the development and evaluation of deep learning models with just a few lines of code.
- Sci-kit learn 0.24.2 – It is a widely used machine learning library used for classification, predictive analytics and several other machine learning related tasks.
- Pandas 1.3.2 – An open-source data manipulation library which is built upon other libraries such NumPy.
- vaderSentiment 3.3.2 – A sentiment analysis library which lexicon and rule based and is specialised in extracting sentiments accurately from social media.
- matplotlib 3.4.3 – It is a data widely used data visualisation tool for plotting graphs and is built upon NumPy.
- seaborn 0.11.2 – It is a widely used data visualisation library based on matplotlib which provides attractive and high-level interface to display informative statistical graphs

4.2 Data Collection

Data was collected and studied upon five different popular stocks:

	Ticker Symbol	Description
Apple	APPL	Technological Company
Tesla	TSLA	Automotive Company
Amazon	AMZN	Online Store Company
Google	Goog	Technological Company
Google	Googl	Technological Company

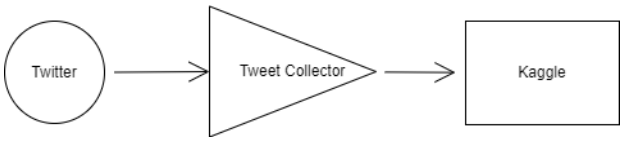
4.2.1 Stock Data

Stock data was collected from Yahoo Finance. Data was selected for each of the individual ticker symbols between the 2015 and 2020. Each was collected and downloaded as a CSV file which could be later accessed as a Pandas data frame when loaded on. The data returned only includes working days, consequently none of the weekends were present in the data.

Format of the data

Feature	Type	Description
Date	Integer	The unique id of each tweet
Open	String	The ticker symbol associated with each tweet
High	String	The author of the tweet
Low	Integer	The date it was posted at
Close	String	The content of the tweet itself
Volume	Integer	The number of comments for that tweet
Adj Close	Integer	The number of retweets

4.2.2 Twitter Data



As it was not possible to collect tweets using the Twitter using the Twitter API due to Twitter not having verified my developer account, instead a Kaggle dataset was used, found at : https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015-to-2020/discussion?sort=votes. The Twitter data stored inside this was retrieved from Twitter using the Tweet Scraping library at: https://github.com/omer-metin/TweetCollector. The same dataset is used by [28] in their investigation of Speculator and Influencer Evaluation in the stock market using social media.

Format of the data:

Feature	Type	Description
tweet_id	Integer	The unique id of each tweet
ticker_symbol	String	The ticker symbol associated with each tweet
writer	String	The author of the tweet
post_date	Integer	The date it was posted at
body	String	The content of the tweet itself
comment_num	Integer	The number of comments for that tweet
retweet_num	Integer	The number of retweets
like_num	Integer	The number of likes

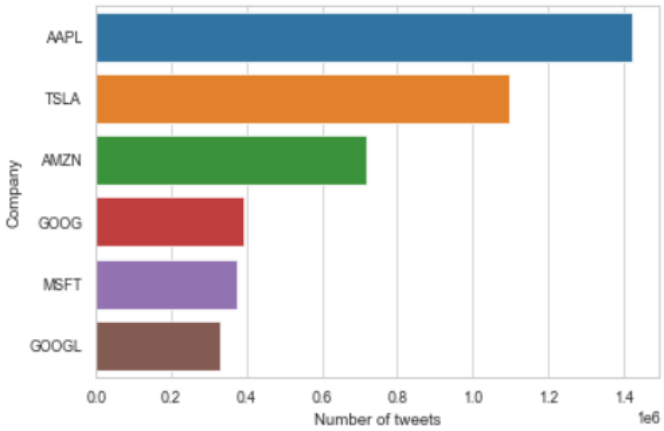


Fig 1. Quantity of tweets per ticker symbol

The collected data did not have an even distribution of the tweets, thus certain companies that were being investigated had a higher number of tweets than other within the investigation.

4.3 Implementation

This dissertation will be building upon previous papers mentioned in the Literature Review, more specifically in 2.4, as well as performing experiments that have highlighting in Chapter 3 under Experiment Protocols.

4.3.1 Data Preparation

1. Determining the stock market direction

To determine the direction of the movement of the stock price, the percentage change difference was calculated between the current day's adjusted close value and the previous day's adjusted close value. The main reason the Adjusted Close price of yesterday to today was considered was because we wanted to cover all Twitter activity between the entire day, if we had chosen solely Open to Close/Adjusted Close, we would have not taken into account the effect that Twitter Activity would have especially during the night. This method is like that of paper [28] but instead the percentage difference was calculated instead.

$$PC\ (Percentage\ Change) = \frac{AdjClose_{Today} - AdjClose_{Yesterday}}{AdjClose_{Yesterday}} * 100$$

The difference, which was either positive or negative allowed to classify the direction of the stock market in accordance with the change. If the change was positive, it was classified it as 'Positive' to

signify that the stock value direction had moving up, if it was negative, then it was classified as 'Negative' to account for the stock value moving down. This project differs from [28] as we classified the percentage change to describe different trends into which the change was classified into, a method similarly adopted by [22]. Once the trends were classified, they were added as a feature into the dataset which would then later be separated as it would be the target value that we would be trying to predict.

$$Trend\ Classification = \begin{cases} PC > 0, & Positive \\ PC < 0, & Negative \end{cases}$$

2. Normalisation/Scaling of the data

To observe how large of a difference is there, this project employs 2 different types of scalers in the data preparation to normalise the data. Initially, the Minmax scaler was solely considered as the sentiment values that would be derived would fall within the range of -1 to 1, but upon further reading an issue rose highlighting the fault of considering a Minmax scaler to scale stock data as it would place an artificial ceiling value that limits the maximum possible price that a particular stock could be worth at any given time based on the data that we have provided it, which is not possible. To counteract and contrast the effect that this scaler would have on the prediction power of my models, this project sough to use the Minmax scaler and compare its results to a Standard Scaler to observe the change in the overall performance of the models that were employed.

3. Sentiment Analysis

Library

To calculate the sentiment of each tweet, the library called vaderSentiment was utilised (found at: https://github.com/cjhutto/vaderSentiment). Below is an extension of the literature review done earlier in chapter 2 upon VADER.

Introduction to VADER

VADER [21] is a simple rule-based model specially attuned towards calculating the sentiment polarity of microblogs, such as the ones found in social media like Twitter, but also accurately predict the sentiment of general text just as well. VADER utilizes a combination of qualitative and

quantitative methodologies to construct an empirically validated gold standard sentiment lexicon. The sentiment lexicon is then used in combination with five generalised rules which encapsulate the syntactic and grammatical conventions used generally in expressing/emphasising the sentiment intensity. VADER has been observed to produce results as accurate as human interpreters of sentiment of a particular text or better in scenarios involving classification of a text as positive, neutral, or negative.

Constructing a lexicon

VADER constructed its gold standard lexicon list by considering previously well-established lexicon lists such as LIWC, ANEW, and GI, and then added on lexical features which were unique to microblogs like emoticons and emojis which also indicate sentiment, acronyms such 'LOL' and 'WTF', and popular internet slang like 'nah' or 'meh'. Next, by using a wisdom of the crowd approach, they assessed the general capability for different acquired sentiment expressions and produced a valid point estimate for intensity of the sentiment of every candidate they acquired, free from any form of surrounding context. They asked the crowd to rate each of the individual candidates they had acquired from a range of -4 (extremely negative) to +4 (extremely positive) with 0 being neutral. Through selecting lexicons with a non-zero rating and standard deviation which was lesser than 2.5 left them with 7500 lexical features to be used which comprised of the VADER gold standard lexicon list which accessible at their GitHub repository (vaderSentiment/vader_lexicon.txt at master · cjhutto/vaderSentiment · GitHub).

General Rules Used in Assessing the Intensity of the Sentiment

400 positive and 400 negative tweets were extracted from a larger 10,000 tweet set and were analysed by two people who provided a sentiment score between -4 (extremely negative) to +4 (extremely positive) and then accordingly, through a qualitative process, the characteristics that affected the way the sentiment valence was perceived was considered which led to generalisable grammatical and syntactic rules. These rules allow to cover more than what would be encapsulated using a bag of words model. There were 5 main rules being: 1. Punctuation, mainly the exclamation mark, exaggerates the intensity of the sentiment. 2. Capitalisation of words increases the intensity of the

sentiment 3. Degree modifiers can increase or decrease the sentiment intensity (e.g., ‘good’ vs ‘extremely good’) 4. Contrastive conjunctive, e.g., ‘but’ can suggest a flip in sentiment, thus giving a mixed sentiment intensity 5. Examining the trigram that comes before a sentiment-laden lexical feature, it allows the catch where negation would flip the polarity of the text (e.g., ‘The food here isn’t really great’).

Testing the sentiment analyser

Results were compared on four different samples, each containing a significant number of microblogs or different pieces of text. The samples were Tweets, Movie Reviews, Technical Product Reviews, and Opinion New Articles. In general, the VADER sentiment was the best classifier behind humans except for classifying tweets where it performed better at classifying them than humans.

Sentiment Analysis Method

By taking into consideration of the results produced by the VADER sentiment library, we used the library to provide a sentiment intensity value for the Tweets that were to be analyzed, this sentiment analysis method was also adopted by [22] for analyzing the sentiment of their new articles, although they preprocessed the text data by removing regular expressions, making all the text lower case, using keywords to search for new related to the stocks they investigated and used tokenisation and removed stop words. On one hand, it should see an improvement more significant them due to VADER being more specialised for microblog like content such as the content present on Twitter rather than news headlines, however on the other hand it would be interesting to see if the preprocessing they took into consideration would have a significant impact or not as this was not done within the implementation.

A function was created where it would derive a sentiment score and created a new feature in the dataset to house the sentiment values that corresponded to each of the tweets. When using vaderSentiment, there are four different values that returned which are: ‘pos’, ‘compound’, ’neu’, and ‘neg’. The compound value was taken into consideration rather than the other three as its score was calculated by considering the other three features. In other words, it’s a normalised, weighted composite score. The score ranged from +1 to -1.

4.3.2 Features

Feature Engineering

Since the extracted tweets dataset contained tweets every day from 2015 to 2020, the data when concatenated with the stock data would not join properly, thus feature manipulation and engineering had to take place.

Engagement

The popularity of a certain tweet was considered, it was calculated by checking how engaged each tweet was with the public present on Twitter. We amalgamated the number of likes, retweets, and comments which in totality displayed the popularity of a certain tweet, the average sentiment (average public sentiment) and total sentiment (all the sentiment added together) was calculated as well as the average and total engagement for each specific day. Once calculated, the date was then processed and converted the datetime64[ns] format and was accordingly concatenated with the stock data with the date being the index.

$$Engagement = Likes + Retweets + Comments$$

Once the total engagement for each individual tweet was collected, the all the engagements for all tweets for a given day was added up to provide a total engagement feature per day. The average engagement was also calculated by finding the what the average amount of engagement took place for a given day and was also added in as a feature.

$$Average\ Engagement = \frac{Total\ Engagemetn\ for\ the\ day}{Instances\ of\ engagement\ in\ day}$$

Sentiment

After performing sentiment analysis, the average sentiment and total sentiment was also calculated for a given day much like the average and total engagement and were added as features. The average sentiment was also classified into positive and negative at different threshold values as the sentiment value distribution in terms of positive and negative varied on the threshold for what tweets were considered positive and what tweets were considered negative. This is like how paper [28] classified their tweets but instead of using three different categories which were positive, neutral, and negative,

it was decided to investigate using only two being positive and negative. This was due to extremely heavy biases taking place and consequently resulting in overfitting when the data was placed in the models due to a false sense of accuracy from low variability in the data, thus, to avoid it binary classification was preferred over multiclass classification.

$$Threshold\ Classification = \begin{cases} x > n, & Positive \\ x < n, & Negative \end{cases}$$

Before being placed into the machine learning algorithm, one hot encoding was done on the classified sentiment to ensure the variables were separated and converted into numerical features. One experiment that was carried out in a similar manner was changing the thresholds, this was to compare between the implementation of [28] and the implementation of this dissertation as one of the experiments that was to be conducted in Chapter 3.4 (Experiment Protocols) to see how different classification thresholds for the sentiment would affect the results after being placed in the pipeline.

Features Correlation and Feature sets

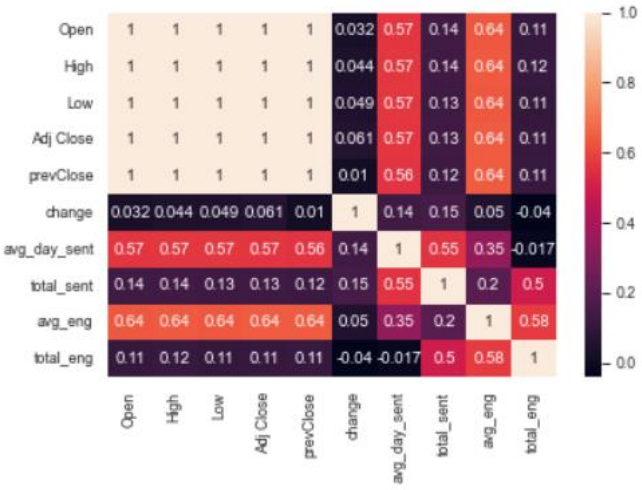


Fig 2. Feature Correlation Heatmap

There were two main feature sets used, one being the basic feature set with just ‘Adjusted Close’ and the other being the complex feature set that included Adjusted Close as well as Open, High, and Low. The main feature is ‘Adjusted Close’, was chosen instead ‘Close’ due to the ease evaluation on the stock performance by allowing investors to understand how much profit would have been made on a given asset which is because it encompasses factors such as dividends, stock splits and new stock offerings which just ‘Close’ would not consider. The features in the complex feature set were decided

upon based on the level of correlation they held to the stock price which was determined by observing a correlation heatmap. This separation took place to contrast the predicting power solely using the Adjusted Close values with the feature set that included the High, Low and Open when used in models with and without sentiment later in the implementation.

4.3.3 Model Implementation

Models

The same five models used earlier in predicting the direction solely using stock features were also considered to see what effect Twitter data and sentiment had on predicting the direction of the market. The model was cross validated in both scenarios, with and without the sentiment to find the most accurate value by finding the average of the accuracies predicted.

The models used were:

- Gradient Booster Classifier (GB)
- Gaussian Naïve Bayes (GNB)
- Logistic Regression (LR)
- Random Forest Classifier (RF)
- Support Vector Machines (SVM)

Model Pipeline

The Scikit-Learn machine learning library will be used which supports the use of several different classification models and perform data preprocessing. It will be possible to split the data in testing and training sections as well as provide evaluation metrics. The model pipeline will contain the following steps:

- Data Normalization
- One Hot Encoding
- Separation into Train and Test Datasets
- Model Evaluation

Training and Validation Data Setup

As it would be preferred to avoid data leakage as we do not want our models to develop a false sense of accuracy about future trends, we will be using scikit-learn to split our problem into train and test datasets. We will be using the default split which would be 25% test and 75% train.

Model Evaluation

To evaluate the models, three factors were considered: Accuracy, F1-Score, and the AUC-ROC curve. This would allow to compare the accuracy, but not just the accuracy but how well the models perform in general as well as how reliable is the accuracy.

Accuracy

Accuracy would allow to see how many True Positives the model has placed, thus giving a general idea for how well the model had performed. Although it does give a general idea on how well the model had classified, it wouldn't be a good measurement if the data contains some form of imbalance.

F1-Score

The F1 score can prove to be a better accuracy metric, considering the data is imbalanced in terms of sentiment classification, as it would allow me to evaluate the reliability of my accuracy score as well as inform me how well the model has distinguished the classification of the target data.

AUC-ROC Score

Since this is a binary classification problem, this project uses the AUC-ROC to determine how well the model is distinguishing between the classification of the predicted data at different threshold values, the higher the value, the stronger the distinguishing power of the model is.

Issues Faced During Implementation

One of the major issues faced during the implementation was that my models were overfitting over the testing data, this was due to selecting data which exceeded a certain threshold level, which in theory should have let me use data which was supposedly highly influential, however due to a lack of variation in the data, the data was being overfitted, thus all tweets were considered instead

5. Evaluation

5.1 Experiments

The goal of this dissertation is to investigate if by if tweets influence the direction towards which the stock price moves, and the hypothesis was that the tweets indeed do influence the stock movement. Therefore, in this section we will be exploring the results procured through implementation of the method (Chapter 3) in the Technical Implementation section of this dissertation (Chapter 4).

We primarily conducted 4 different kinds of experiments (outlined in Chapter 3.4) to observe if there were a relationship which would improve the model's predictive ability. The four methods that we will be exploring are:

- Sentiment Analysis: this is for answering the main the question of this research project, which was the investigate if tweets affect stock market trends.
- Different thresholds for sentiment distribution: we will be considering how the sentiment distribution of the positive and negative ratios of tweets could affect the predictive ability of the model. We would also be considering the effect if we solely used the average day sentiment value instead of classifying it.
- Using different normalization methods: Different normalisation methods will also be considered as this was a criticism of some of the methodologies used which were covered in the Literature Review section of this dissertation.
- Using feature engineered features in conjunction with sentiment analysis: Certain features were engineered as well as previously discussed in the features section of the technical implementation to see if other features when coupled with sentiment improve the predictive power of our deployed models.

5.2 Results

5.2.1 Adding sentiment Analysis

Differences in accuracy with and without sentiment

As we proceeded with this experiment, we used this experiment as a baseline to compare to other experiments as this experiment would investigate the main question of this dissertation being do tweets affect stock trends. We used the basic and complex feature sets that were described under Features in the Technical Implementation. We also kept the separated the tweets into positive and negative at a threshold of one, thus all tweets which carried a sentiment valence score of above 0.1 were considered positive and the rest were considered negative. See Appendix B for the sentiment distribution at threshold 0.1.

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	55%	49%	52%	51%	55%	55%	55%	55%	50%
Tesla	54%	57%	50%	54%	52%	57%	47%	57%	53%	57%
Amazon	55%	58%	50%	65%	55%	58%	55%	58%	54%	56%
Goog	56%	56%	51%	72%	56%	56%	56%	55%	56%	60%
Googl	55%	55%	52%	72%	55%	55%	55%	55%	55%	56%

Table 1. Accuracy Without vs With Sentiment Basic Threshold (0.1)

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	54%	65%	62%	55%	55%	49%	55%	56%	52%
Tesla	55%	57%	75%	75%	53%	57%	48%	57%	56%	58%
Amazon	55%	58%	69%	67%	53%	58%	53%	58%	51%	56%
Goog	56%	56%	73%	72%	56%	56%	56%	56%	57%	60%
Googl	55%	55%	70%	71%	56%	55%	55%	55%	56%	56%

Table 2. Accuracy Without vs With Sentiment Complex Threshold (0.1)

We have displayed the results over a 10 k-cross validation model and displayed the results in a box chart which represents the ROC-AUC value as well as the accuracy and the F1 weighted score. (See graphical results comparison in Appendix B).

5.2.2 Experimenting with Thresholds

In this experiment we decided to change threshold value and use no threshold value, just the raw average day sentiment valence score achieved after sentiment performing sentiment analysis. We chose the threshold specifically due to bring a nearly equally distribution of positive to negative tweets for the apple tweets as we wanted to see if the balance of the ratio of Negative to Positive tweets affected the performance of the model. We also wanted to investigate if the classification of tweets itself could be affecting the performance of the model, therefore we used the raw average sentiment value to gauge to observe the effects.

Using Threshold 0.14

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	56%	49%	53%	51%	52%	55%	56%	55%	51%
Tesla	54%	51%	50%	54%	52%	48%	47%	48%	53%	51%
Amazon	55%	56%	50%	62%	55%	58%	55%	60%	54%	60%
Goog	56%	56%	51%	74%	56%	54%	56%	54%	56%	60%
Googl	55%	50%	52%	69%	55%	51%	55%	49%	55%	52%

Table 3. Accuracy Without vs With Sentiment Basic Threshold (0.14)

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	56%	65%	61%	55%	53%	49%	57%	56%	52%
Tesla	55%	52%	75%	76%	53%	51%	48%	48%	56%	51%
Amazon	55%	58%	69%	62%	53%	58%	53%	60%	51%	60%
Goog	56%	56%	73%	73%	56%	54%	56%	54%	57%	60%
Googl	55%	50%	70%	73%	56%	51%	55%	49%	56%	52%

Table 4. Accuracy Without vs With Sentiment Complex Threshold (0.14)

We have displayed the results over a 10 k-cross validation model and displayed the results in a box chart which represents the ROC-AUC value as well as the accuracy and the F1 weighted score. (See graphical results comparison in Appendix E). Using No Threshold and just Raw Values

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	51%	49%	47%	51%	51%	55%	55%	55%	53%
Tesla	54%	55%	50%	56%	52%	59%	47%	57%	53%	57%
Amazon	55%	62%	50%	62%	55%	58%	55%	59%	54%	61%
Goog	56%	56%	51%	68%	56%	55%	56%	57%	56%	54%
Googl	55%	58%	52%	67%	55%	58%	55%	54%	55%	53%

Table 5. Accuracy Without vs With Sentiment Basic No Threshold

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	52%	65%	54%	55%	53%	49%	55%	56%	53%
Tesla	55%	62%	75%	72%	53%	68%	48%	58%	56%	55%
Amazon	55%	62%	69%	60%	53%	58%	53%	59%	51%	61%
Goog	56%	56%	73%	70%	56%	55%	56%	57%	57%	54%
Googl	55%	58%	70%	67%	56%	58%	55%	54%	56%	53%

Table 6. Accuracy Without vs With Sentiment Complex No Threshold

We have displayed the results over a 10 k-cross validation model and displayed the results in a box chart which represents the ROC-AUC value as well as the accuracy and the F1 weighted score. (See graphical results comparison in Appendix F).

5.2.3 Considering different types of scalers

Earlier in the experiment, a Minmax scaler had been utilised for normalizing the values of the data that were being placed within the classification models. The reason was stated the reason as to why the Minmax scaler was used originally but the criticisms of the using it were also brought forward, therefore an observation was wanted to examine if there was any favourable or adverse effect using a Standard scaler in the place of a Minmax scaler

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	55%	49%	51%	51%	55%	55%	55%	55%	50%
Tesla	54%	57%	50%	54%	52%	57%	47%	57%	53%	57%
Amazon	55%	63%	50%	65%	55%	57%	55%	58%	54%	56%
Goog	56%	80%	51%	72%	56%	51%	56%	55%	56%	60%
Googl	55%	67%	52%	71%	55%	55%	55%	55%	55%	56%

Table 7. Accuracy Without vs With Sentiment Basic Using Standard Scaler Threshold (0.1)

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	62%	62%	68%	59%	55%	55%	49%	55%	56%	52%
Tesla	75%	78%	74%	76%	54%	57%	48%	57%	56%	58%
Amazon	60%	63%	70%	68%	53%	57%	53%	58%	51%	56%
Goog	74%	80%	70%	69%	56%	51%	56%	55%	57%	60%
Googl	75%	67%	68%	71%	56%	55%	55%	55%	56%	56%

Table 8. Accuracy Without vs With Sentiment Complex Using Standard Scaler Threshold (0.1)

We have displayed the results over a 10 k-cross validation model and displayed the results in a box chart which represents the ROC-AUC value as well as the accuracy and the F1 weighted score. (See graphical results comparison in Appendix G).

5.2.4 Using average engagement of the day along with sentiment

Average engagement was also considered as one of the potential affecting factors upon the stock trend as there was an increase in Tesla stock price (see Appendix D) which correlated with the increase in the average amount of engagement, therefore we wanted to explore the possibility of there being a relationship between stock trends and sentiment coupled with average engagement.

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	55%	49%	56%	51%	55%	55%	54%	55%	54%
Tesla	54%	57%	50%	60%	52%	57%	47%	57%	53%	58%
Amazon	55%	59%	50%	59%	55%	57%	55%	58%	54%	56%
Goog	56%	56%	51%	65%	56%	54%	56%	52%	56%	54%
Googl	55%	55%	52%	64%	55%	55%	55%	55%	55%	53%

Table 9. Accuracy Without vs With Sentiment Analysis & Average Engagement Basic Threshold (0.1)

	Logistic Regression		Random Forest		SVM		Naïve Bayes		Gradient Booster	
	Without	With	Without	With	Without	With	Without	With	Without	With
Apple	55%	53%	65%	60%	55%	55%	49%	54%	56%	54%
Tesla	55%	57%	75%	74%	53%	57%	48%	57%	56%	58%
Amazon	55%	59%	69%	57%	53%	57%	53%	58%	51%	56%
Goog	56%	56%	73%	65%	56%	54%	56%	52%	57%	54%
Googl	55%	55%	70%	66%	56%	55%	55%	55%	56%	53%

Table 10. Accuracy Without vs With Sentiment Analysis & Average Engagement Complex Threshold (0.1)

We have displayed the results over a 10 k-cross validation model and displayed the results in a box chart which represents the ROC-AUC value as well as the accuracy and the F1 weighted score. (See graphical results comparison in Appendix H).

5.3 Discussion

5.3.1 Adding sentiment

When comparing the basic set before and after adding the sentiment, we can see that there is not a significant increase in most cases. The best performing algorithms we the Random Forest and Gradient Boosting Classifier. The Random Forest Classifier rose the accuracy especially for AMZN, GOOG, and GOOGL by around 15-20% whilst APPL and TSLA saw minor increases of around 2-4%. This could suggest the number of tweets (Fig 1) which was considerably lower for AMZN, GOOG, and GOOGL could influence the predictive power. Furthermore, TSLA saw a 3-10% improvement across all the models also suggesting the ratio at which the positive and negative tweets are distributed in are important factors to be considered as well. AMZN saw an increase across all models as well, but the increases were not as big as the TSLA predictions except for the Random Forest Classifier. Looking at Appendix B, we can see on average for all stock and on most models that the ROC-AUC value increased significantly and the range of values held by F1 also increased indicating that although there might have not been an increase on the accuracy of a lot of models performances by a significant amount, the performance of the model increased thus showing that the accuracy results we have derived are quite reliable compared to just using Adjusted Close as the sole feature to predict the trend. When adding features such as Open, High, and Low, the predictive power of the model when using Random Forest especially increase, but in most cases the sentiment acted more as noise, or we saw slight improvements and, in some cases, it was slightly as a degrading factor across the board. Although such changes did take place, the ROC-AUC values, and the F1-score, just like the basic set saw significant improvements suggesting that adding sentiment potentially has a strong relationship as it improves the models performs thus enhancing the reliability of the accuracy of the model.

The improvements show, especially after adding sentiment shows that there is a relationship between sentiment and stocks, consequently affirming the hypothesis that had been stated earlier.

5.3.2 Adjusting Thresholds

Threshold at 0.14

The aim of this experiment was to determine if the threshold values at which the positive and negative tweets were separated affected the predictive power of the model that we were using. As assumed, the APPL saw an increase in the accuracy, however the accuracy just increased by an insignificant amount in general, on the other hand the predictive power for TSLA had significantly been worsened. One reason for this could be that 0.1 could the average sentiment across the spectrum of sentiment across all days and this was not taken into consideration for APPL thus showing a contrast in the results. Other stocks also suffered a similar fate of seeing significant decreases in their predictive power, both when compared to just using Adjusted Close as well as using performing the same experiment with a threshold at 0.1. When looking Appendix E, we can observe through a comparison with the model using a threshold of 0.14 when compared to a model using a threshold of 0.1, through 10 k-cross validation, APPL has seen the most improvement in all three metrics which include the ROC-AUC, F1-Score, and the accuracy which was true for both the basic and complex set. There were no significant improvements on the other ticker values that were investigated in both the complex set and the basic set when compared with the model at a threshold of 0.1, there were instances where the ROC-AUC value and the F1-score also had decreased slightly suggesting that changing the threshold does influence the predictive power of the model.

No Threshold

When considering solely the average day sentiment value, the results show overall an increase as there would be no bias. Most models show an increase in the accuracy, especially using Random Forest. Tesla and Amazon just as in Table 1 have seen an increase on all the models within the basic set but they were not as significant as when compared to the Table 1 (Basic Threshold 0.1) and in most cases performed worse than the model at threshold 0.1. When considering the complex set, some models showed a significant improvement for certain stocks, and some showed a performance far

worse. The model that showed the most improvement was Logistic Regression in comparison with Table 1, however adding more features only acted as noise except for the Tesla stock in the case when we compared the results of the complex set with no threshold against the basic set with no threshold. Random Forest Trees on the other hand displayed degradation in its results in comparison to results using the basic and complex sets from the model with a 0.1 threshold (Table 1 and 2) as well as the model with 0.14 (Table 3 and 4) as well as the Table 5, insinuating that Random Forest Trees works better with categorized data in comparison to all data being numerical. Using SVM's, Tesla saw a significant increase in both Table 5 and 6, however other stocks using SVM did not benefit in the same manner. When looking at Appendix F, a general trend is observed where the F1 score has decreased, on the other hand, the ROC-AUC score increased. This suggest that the classifier is not good at classifying the trends based without the classification, although due to high ROC-AUC it is possible to find a suitable optimal threshold for classifying the data.

5.3.3 Experimenting with scalers

One of the main criticisms I provide in my literature review is the lack of exploration using different types of scalers. Therefore, I decided to explore a Standard scaler in comparison the Minmax scaler I have used in all the other previous experiments. In this experiment we have only changed the scaler in respect to the model, thus the threshold is at 0.1.

The results for the basic dataset showed nearly no difference at all for most models in comparison to Table 1 which used a Minmax but when using Logistic Regression, there were some notable changes. AMZN, GOOG, and GOOGL so massive increases to their accuracy, especially GOOG at 80%. Another notable factor which was noticed was that when using a standard scaler, the original values without any sentiment added onto them also increased for Logistic Regression and consequently so did the accuracy rates, however for the rest of the models there were not any notable differences except for a minor different the occasional 1% decrease or increase. When observing Appendix G, using a standard scaler improves the ROC-AUC as well as the F1-score signified by the small interquartile block shifts that move towards a higher value. When using a standard scaler, the ROC-

AUC score of SVM and Logistic Regression increases thus suggesting that those two models specifically, especially Logistic Regression, are sensitive to the type of scaler we can use. The F1-score however remained or saw a slight improvement.

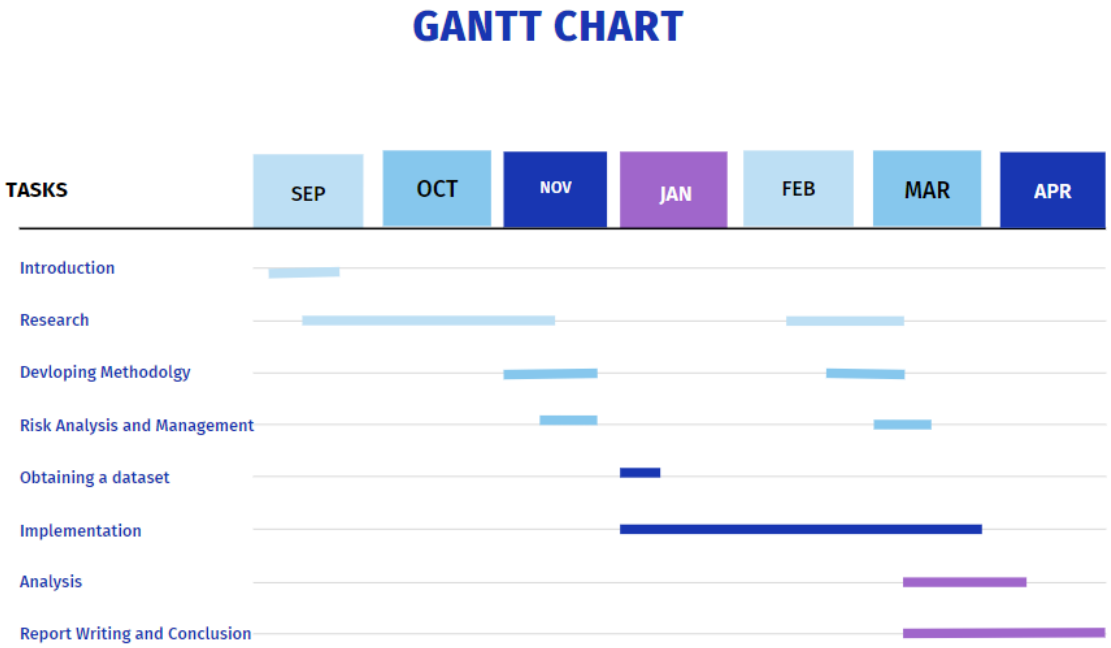
5.3.4 Using average engagement of the day along with sentiment

The essence of this experiment was to try to discover if there were any other factors from Twitter which could improve the performances of our models, thus we looked towards the average engagement of the day and coupled that with the sentiment class. When observing the results, the results, although displaying an improvement in areas compared to the feature sets without the sentiment analysis and average engagement, overall, it performed far worse than the results displayed in Table 1 and Table 2. A general observation made was that there was a reduction in the predictive power in comparison to just using the sentiment in comparison to using sentiment with the average day all across the board, moreover, when analyzing Table 10 the results had performed far worse for Random Forest Trees where in most cases there were significant drops in accuracy, other models didn't achieve any results that was significant in any manner, just minor increases or decreases in accuracy by 2% or were the same as Table 9 suggesting that the extra features in the complex feature set just acted as noise. When looking at Appendix H, in most cases there is a slight or no improvement in the ROC-AUC score, but the F1-score decreases sees a general decrease however in some cases as, especially with the complex feature, there is an increase in the F1-score sometimes but a lot of the times there is a degradation in the F1-score and the ROC-AUC score. This insinuates that the average engagement per day acts as noise or a deprecating factor when coupled with the sentiment and would not be suitable to consider for predicting stock trends.

6. Project Management

6.1 Planning & Timeline

I have described my timeline in the form of a Gantt Chart. M1 to M4 represents the 4 months where the research and design phase takes place. In this phase I investigate and understand the information related to my topic as well as designing my methodology and whatever comes and goes along with it. In M5 to M6 I am implementing and recording all my results as well as concluding the findings that I have come across. Once I have noted everything down and gathered all my findings in the research dissertation, I will submit the project.



A weekly breakdown is available in Appendix I. Research (including literature review), methodology and risk analysis and management were re-written during the implementation section as there were some mistakes earlier, thus corrections were made to the literature review and other highlighted sections and those parts were documented in the weekly plan.

6.2 Risk Analysis & Management

Every risk holds potential risks against it and this project is no different. Although risks for this project do exist, there are work arounds to resolve the problem or mitigate it to have minimum to no effect on the project.

Risk	Likelihood	Impact	Plan to deal with the issue
Lack of time during testing and implementation	Medium	High	Concentrate on completing high priority tasks and leave low priority tasks aside in the case of less time being available.
Insufficient Computing Resources	Low	High	Use an alternative resource such as the computers at university or cloud computing such as Google Collab
Sparseness within the data containing the tweets	Low	High	Look for an alternative source of data or another dataset on Kaggle with lower levels of no levels of sparsity in the data or scrape the data using the Twitter API.
Corruption in files	Low	Low	Create GitHub repository, if corruption in dataset, find different dataset or make your own.
Tweet sentiment not having any effect	Medium	Medium	Gain best possible accuracy through using alternative features or engineered features.

6.3 Public, Legal, Social and Ethical Issues

6.3.1 Professional Issues

I will obtain all my data from Kaggle datasets, one of them being Tweets that were obtained through scraping through the Twitter Platform, and the other being a collection of Stock information upon the corporations: Apple, Microsoft, Amazon, and Google. All the information is accessible online as well as the publishers of the data will be referenced accordingly. All the technologies that are going to be used are publicly available as well.

6.3.2 Legal Issues

Since Twitter is a public platform, it falls under the GDPR (General Data Protection Regulation) standards where it is stated that any form of information that holds the potential to be identified with

an individual has a requirement of transparency and consent and they should have the right to be able to retract their consent at any time they want to [24]. As it is a public platform, Twitter asks for user consent on access as well as any form of usage on the data that belongs to them and exists on the Twitter platform, consequently the users also hold the power to retract their consent and have it removed from the platform. any form of identification pertaining to individual that are present in the data have been replaced with unique identification. Moreover, the dataset is obtained through Kaggle, thus how the data was obtained is referenced specifically over there. All tools used to develop this project are open-source, publicly available and accessible tools for programming.

6.3.3 Ethical and Social Issues

The use of human subjects is non-existent within the premises of this project as well as data that holds any relational value has been replaced with unique identifiers. Sentiment analysis on opinions provided by individuals on the platform will be used to aid predicting the stock market value due the possibility of their tweets having a proportionate effect in the stock market value which is the hypothesis being investigated within this project.

7. Conclusions

7.1 Conclusion

Within this study, we aimed to predict stock trends using sentiment analysis of tweets by analyzing tweets between 2015 to 2020 on APPL, TSLA, AMZN, GOOG, and GOOGL. We attempted to build a model that would predict the next day trend by considering solely just the Adjusted Close as well as features that were highly correlated with Adjusted Close including Open, High, and Close. We managed to observe an overarching result where we could determine that there was relationship between tweets and the stock market trend as there was a general increase in the predictive power of the models employed, especially considering the initial experiment. The best performing models were Logistic Regression and Random Forest Classifiers when employed over a range of different experiments that we had conducted. Random Forest Classifiers in most cases was the best performing classifier when solely adjusted close was concerned along with the sentiment analysis, however when extra highly correlated features (complex set) were added in as well, the results varied where the performance the Random Forest Classifier diminished (evident in Table 6), when sentiment was considered when contrasted to application of Random Forest on the complex set without sentiment. Other models processed the extra features as noise in most cases and did not any significant improvements when compared to the initial experiment as well. The model that saw the most significant improvement was Logistic Regression when the type of scaler considered was different, thus showing that the method considered when scaling the values is important as Logistic Regression (see Table 7) is sensitive to it when sentiment is added on. In most cases, the increase and decrease when compared to with and without sentiment had no significant increase or decrease. Thresholds at which the sentiment was also distributed also had effects on the performance of the models as there were minor improvements to the models which saw an equalisation of their negative to positive (e.g., APPL in Table 3) sentiment distribution whereas others saw significant diminishes when these changes were considered (e.g., TSLA in Table 3). In general, all my models generally saw a higher an accuracy rate when compared to the results of paper [28] and compared to paper [22] saw similar

increases in accuracy. When we added in extra features that were highly correlated (complex set), it had the highest adverse effect on SVM's insinuating that SVM's are sensitive to the addition of more features, this affirms what was observed in paper [17].

7.2 Limitation & Future Work

Some of the limitations within the current model pipeline of this research project was the number of tweets that was present in the tweet database that I had extracted from Kaggle. In the future I would like to venture into compiling tweets on my own by either using the Twitter API or Selenium to scrape tweets for analysis later. This would allow for a larger pool of sentiment and thus more equal level of sentiment distribution to occur, thus allowing for far better results when used along with my models. Another possible limitation would be the sentiment analysis method I employed where I only used VADER as opposed to paper [22] where they did some preprocessing on their data to make it easier for the VADER sentiment analyser to process, therefore in the future I would like to explore other NLP techniques for pre-processing my data before applying the VADER sentiment analyser on it. As we have seen there being a relationship between sentiment and the stock movement, in the future I would also like to explore if I could predict the stock price itself using regression algorithms and neural networks. I would also like to take into consideration other sources such as new headlines, other hyper-popular social media platforms such Instagram, Discord, and Facebook and see if an amalgamation of all these different sources could improve the performance of my models.

Bibliography

[1] - Malkiel, Burton G. (1973). *A Random Walk Down Wall Street* (6th ed.). W.W. Norton & Company, Inc. ISBN 978-0-393-06245-8.

[2] - Sewell, M., 2011. History of the efficient market hypothesis. *Rn*, *11*(04), p.04.

[3] - J. Bollen, H. Mao and X. Zeng, "Twitter mood predicts the stock market", *J. Comput. Science*, vol. 2, pp. 1-8, 2011

[4] - A. W. Lo and A. C. MacKinlay, “Stock market prices do not follow random walks: Evidence from a simple specification test,” The review of financial studies, vol. 1, no. 1, pp. 41–66, 1988.

[5] - Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. Cancer genomics & proteomics, 15(1), 41–51. https://doi.org/10.21873/cgp.20063

[6] - Ahmad, M., Aftab, S. and Ali, I., 2017. Sentiment analysis of tweets using svm. Int. J. Comput. Appl, 177(5), pp.25-29.

[7] Huq, M.R., Ali, A. and Rahman, A., 2017. Sentiment analysis on Twitter data using KNN and SVM. International Journal of Advanced Computer Science and Applications, 8(6), pp.19 25.

[8] - Y. Lin et al., "Large-scale image classification: Fast feature extraction and SVM training," CVPR 2011, 2011, pp. 1689-1696, doi: 10.1109/CVPR.2011.5995477.

[9] - Dey Sarkar, S., Goswami, S., Agarwal, A. and Aktar, J., 2014. A novel feature selection technique for text classification using Naive Bayes. International scholarly research notices, 2014.

[10] - C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno and J. Caro, "Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning," IISA 2013, 2013, pp. 1-6, doi: 10.1109/IISA.2013.6623713.

[11] - Schneider, K.M., 2003, April. A comparison of event models for naive bayes anti-spam e mail filtering. In 10th Conference of the European Chapter of the Association for Computational Linguistics.

[12] - Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original (PDF) on 17 April 2016. Retrieved 5 June 2016.

[13] - Belgiu, M. and Drăguţ, L., 2016. Random forest in remote sensing: A review of applications and future directions. ISPRS journal of photogrammetry and remote sensing, 114, pp.24-31.

[14] - Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961

[15] - Graves, Alex; Liwicki, Marcus; Fernandez, Santiago; Bertolami, Roman; Bunke, Horst; Schmidhuber, Jürgen (2009). "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 31 (5): 855–868. doi:10.1109/tpami.2008.137.

[16] - Santaholma, M.E., 2007. Grammar sharing techniques for rule-based multilingual NLP systems. In Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA).

[17] - Shen, S., Jiang, H. and Zhang, T., 2012. Stock market forecasting using machine learning algorithms. Department of Electrical Engineering, Stanford University, Stanford, CA, pp.1-5.

[18] – M. Misra, A. P. Yadav and H. Kaur, "Stock Market Prediction using Machine Learning Algorithms: A Classification Study," *2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE)*, 2018, pp. 2475-2478, doi: 10.1109/ICRIEECE44171.2018.9009178.

[19] - M. P. Naeini, H. Taremian, and H. B. Hashemi, “Stock market value prediction using neural networks,” in 2010 international conference on computer information systems and industrial management applications (CISIM). IEEE, 2010, pp. 132– 136.

[20] - R. K. Bakshi, N. Kaur, R. Kaur and G. Kaur, "Opinion mining and sentiment analysis,"2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 452-455.

[21] - Hutto, C.J & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eight International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[22] - Kabbani, T. and Usta, F.E., 2022. Predicting The Stock Trend Using News Sentiment Analysis and Technical Indicators in Spark. *arXiv preprint arXiv:2201.12283*.

[23] - Pagolu, V.S., Reddy, K.N., Panda, G. and Majhi, B., 2016, October. Sentiment analysis of Twitter data for predicting stock market movements. In 2016 international conference on signal processing, communication, power and embedded system (SCOPES) (pp. 1345-1350)

[24] - Thien Hai Nguyen, Kiyoaki Shirai, Julien Velcin, Sentiment analysis on social media for stock movement prediction, Expert Systems with Applications, Volume 42, Issue 24, 2015, Pages 9603 9611, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2015.07.052.

[25] - Rajput, V. and Bobde, S., 2016. Stock market forecasting techniques: literature survey. *International Journal of Computer Science and Mobile Computing*, *5*(6), pp.500-506.

[26] - Gupta, R., & Chen, M. (2020). Sentiment Analysis for Stock Price Prediction. *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 213-218.

[27] - Huang, W., Nakamori, Y., & Wang, S. (2005). Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res., 32,* 2513-2522.

[28] - M. Doğan, Ö. Metin, E. Tek, S. Yumuşak and K. Öztoprak, "Speculator and Influencer Evaluation in Stock Market by Using Social Media," *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 4559-4566, doi: 10.1109/BigData50022.2020.9378170.

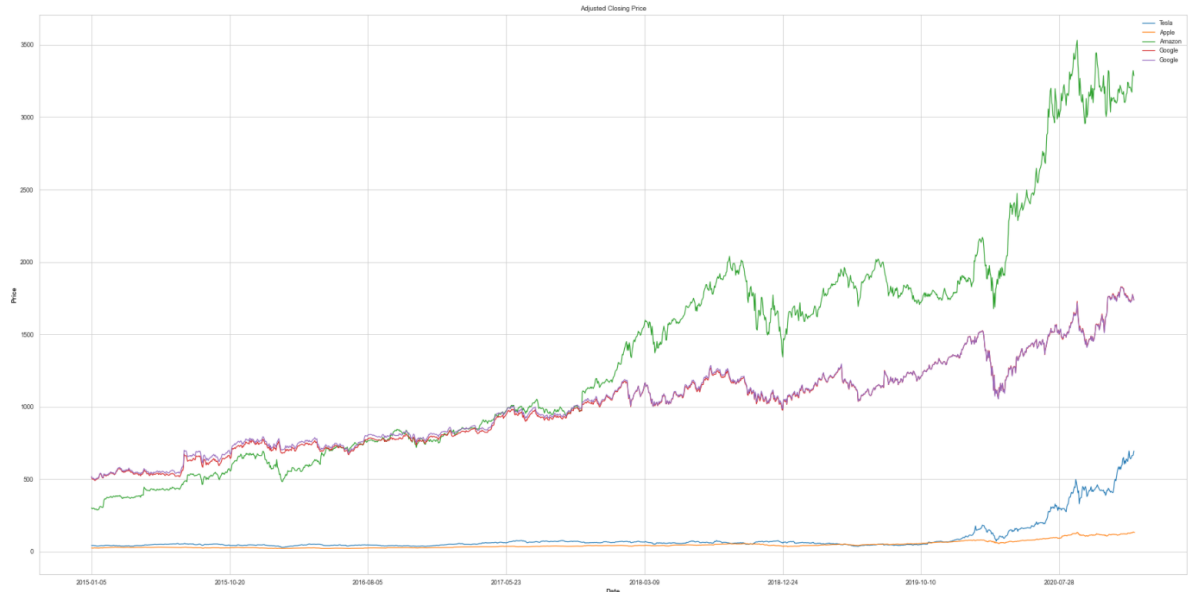
[29] - Malkiel, B. G., & Fama, E. F. (1970). Efficient Capital Markets: A Review Of Theory And Empirical Work*. The Journal of Finance, 25(2), 383–417. doi: 10.1111/j.1540-6261.1970.tb00518.x

[30] - Horne, J. C. V., & Parker, G. G. (1967). The Random-Walk Theory: An Empirical Test. Financial Analysts Journal, 23(6), 87–92. doi: 10.2469/faj.v23.n6.87

[31] - Kim, S.M. and Hovy, E., 2004. Determining the sentiment of opinions. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 1367 1373).

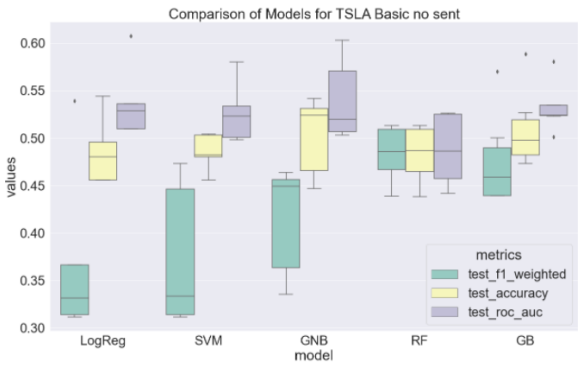
Appendix

Appendix A – Stock prices for all the companies in question

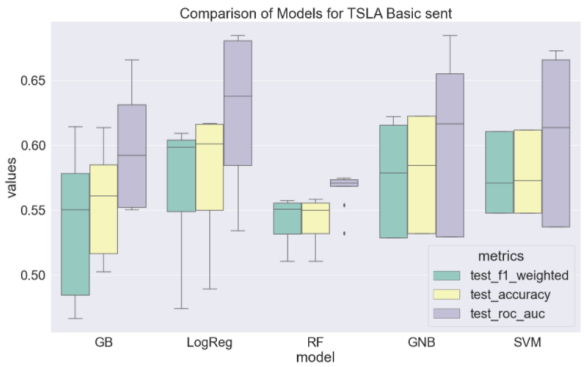


Appendix B – Results for adding sentiment analysis

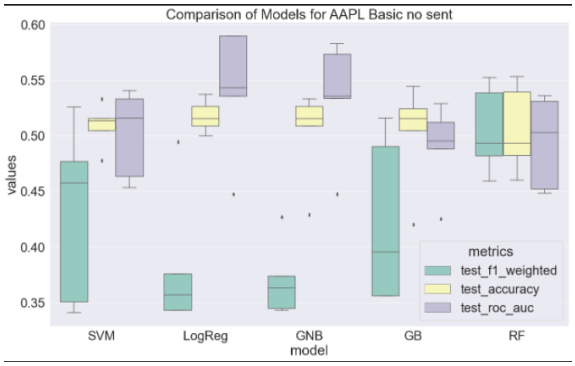
Basic:



Tesla Basic Threshold (0.1) No Sent

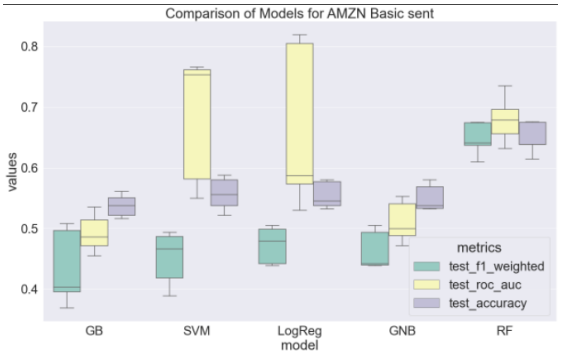


Tesla Basic Threshold (0.1) with Sent

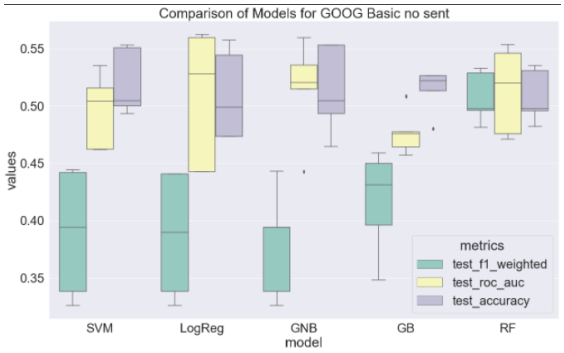


Apple basic Threshold (0.1) No Sent

Apple Basic Threshold (0.1) with Sent



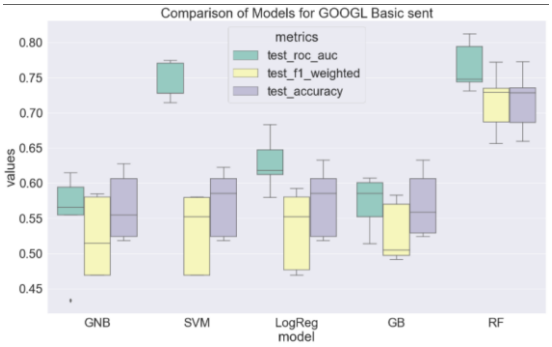
Amazon Basic Threshold (0.1) No Sent



Amazon Basic Threshold (0.1) with Sent

Goog Basic Threshold (0.1) No Sent

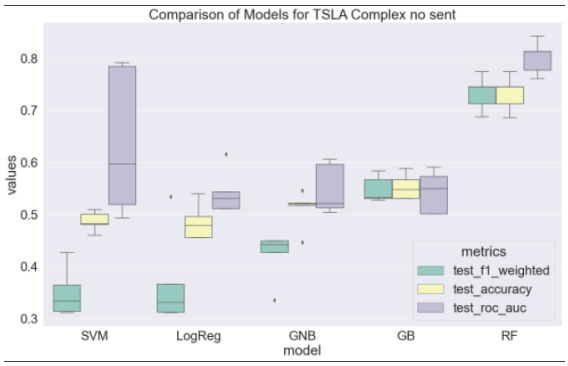
Goog Basic Threshold (0.1) with Sent



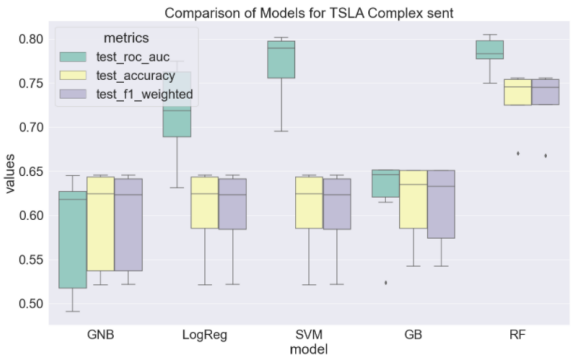
Googl Basic Threshold (0.1) No Sent

Googl Basic Threshold (0.1) with Sent

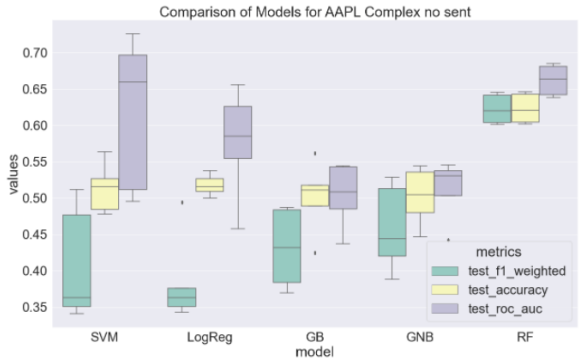
Complex:



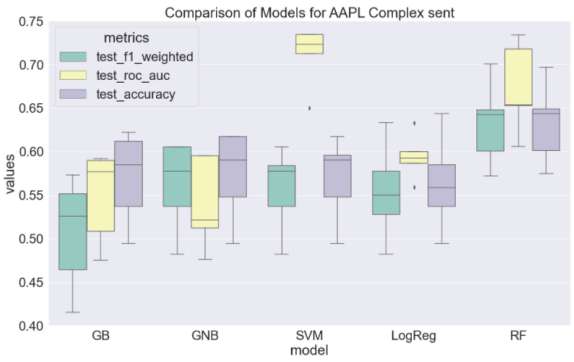
Tesla Complex Threshold (0.1) No Sent



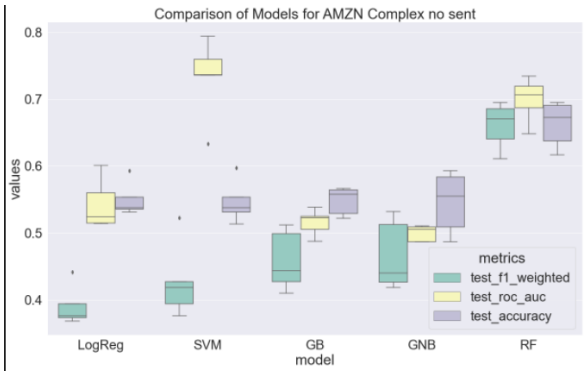
Tesla Complex Threshold (0.1) with Sent



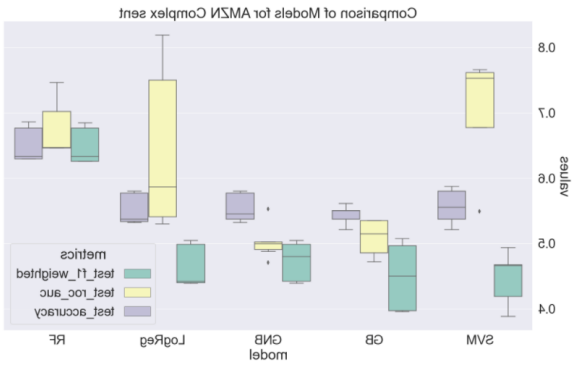
Apple Complex Threshold (0.1) No Sent



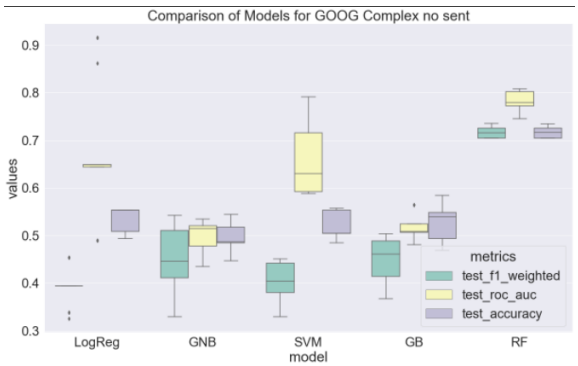
Apple Complex Threshold (0.1) with Sent



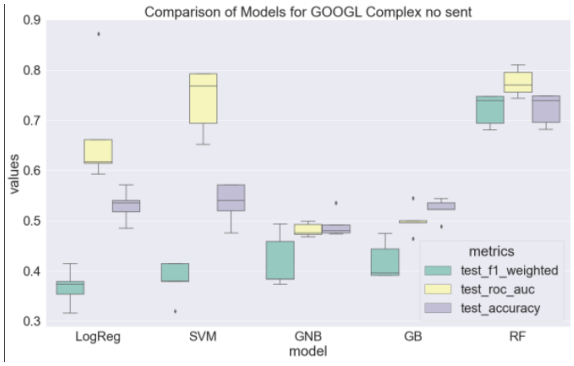
Amazon Complex Threshold (0.1) No Sent



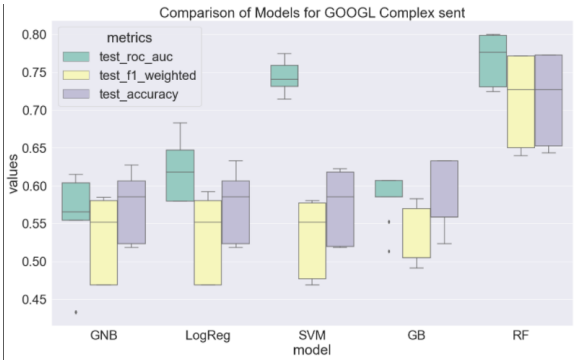
Amazon Complex Threshold (0.1) with Sent



Goog Complex Threshold (0.1) No Sent



Goog Complex Threshold (0.1) with Sent

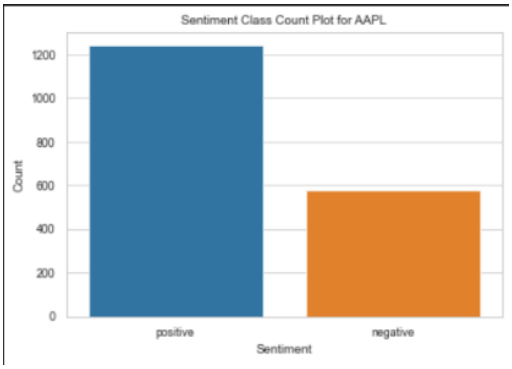


Googl Complex Threshold (0.1) No Sent

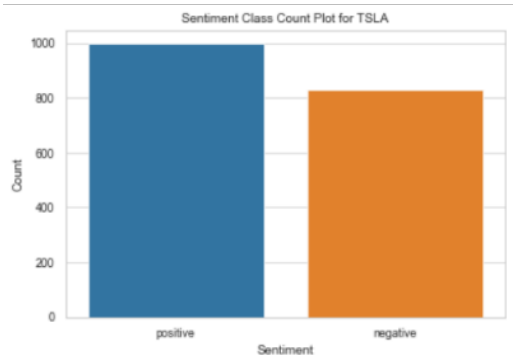
Googl Complex Threshold (0.1) with Sent

Appendix C – Sentiment distributions

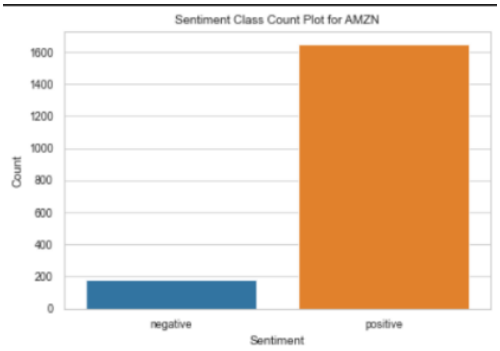
Threshold 0.1



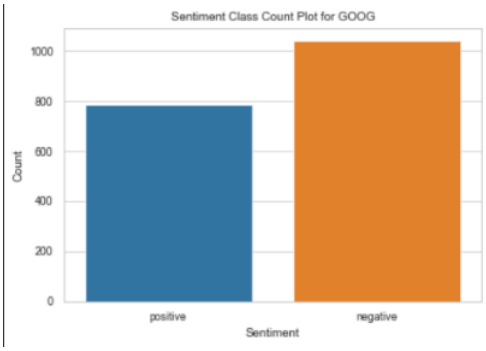
Sentiment Distribution for AAPL at threshold 0.1



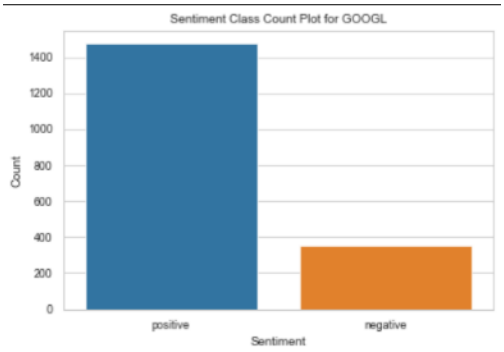
Sentiment Distribution for TSLA at threshold 0.1



Sentiment Distribution for AMZN at threshold 0.1

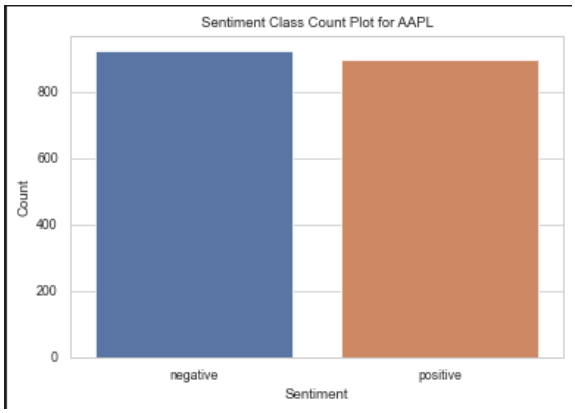


Sentiment Distribution for GOOG at threshold 0.1

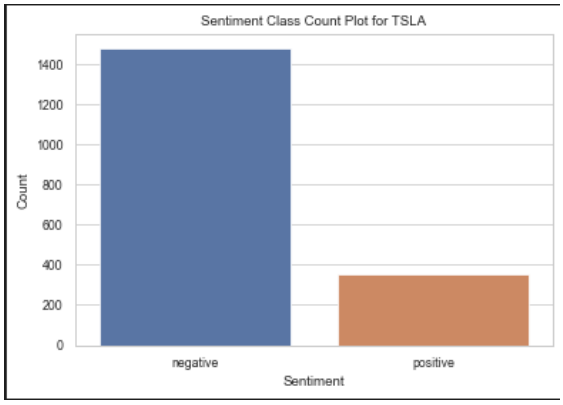


Sentiment Distribution for GOOGL at threshold 0.1

Threshold 0.14

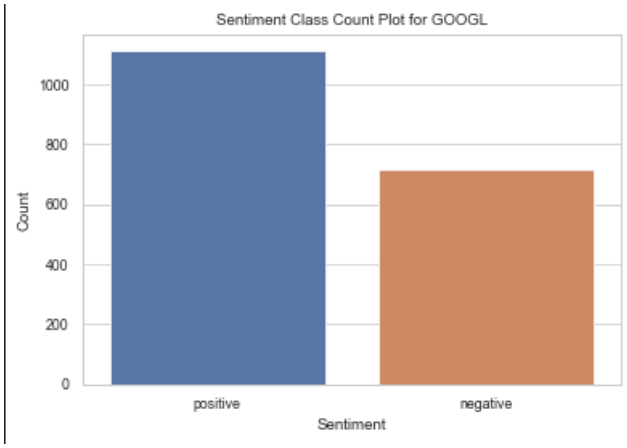


Sentiment Distribution for AAPL at threshold 0.14



Sentiment Distribution for TSLA at threshold 0.14

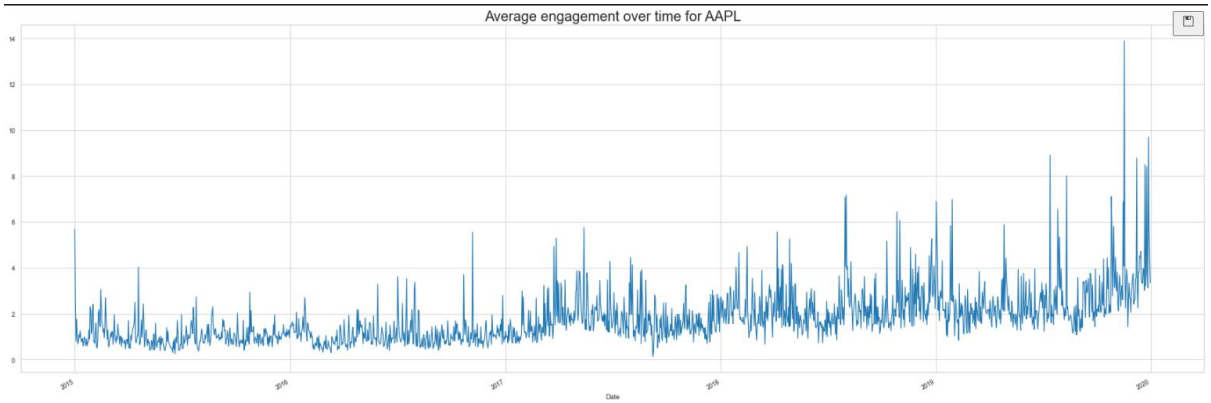
Sentiment Distribution for AMZN at threshold 0.14

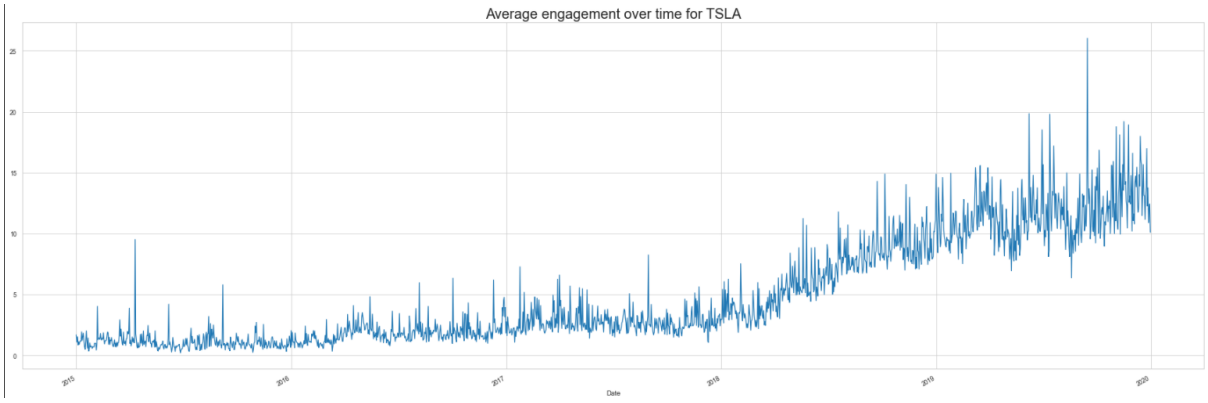


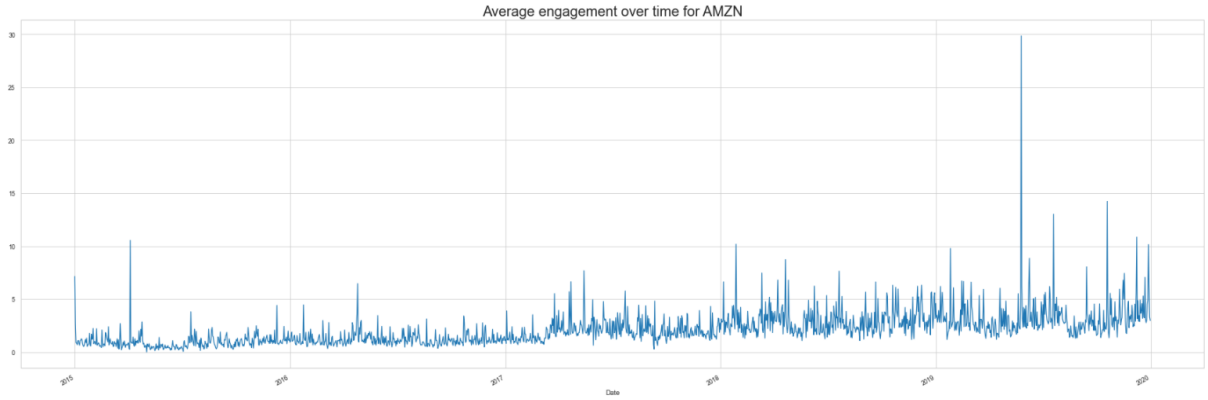
Sentiment Distribution for GOOGL at threshold 0.14

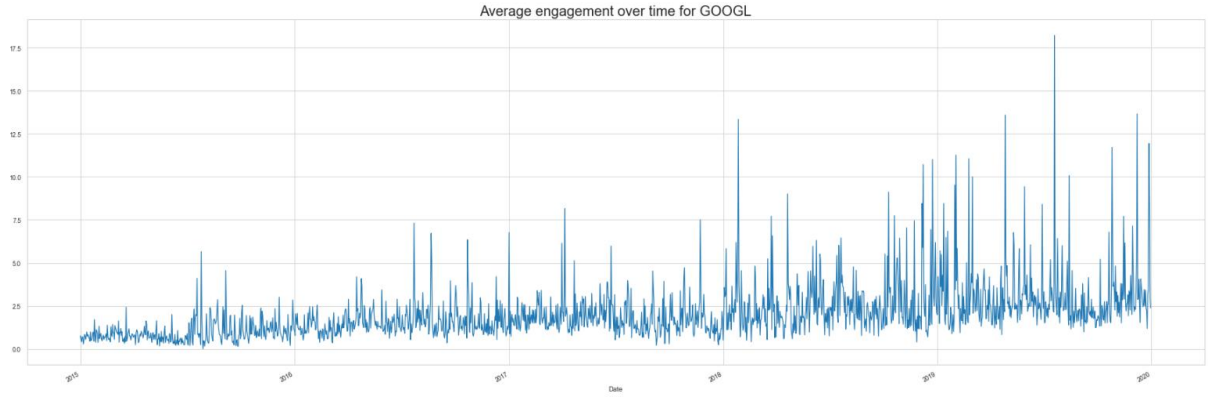
Sentiment Distribution for GOOG at threshold 0.14

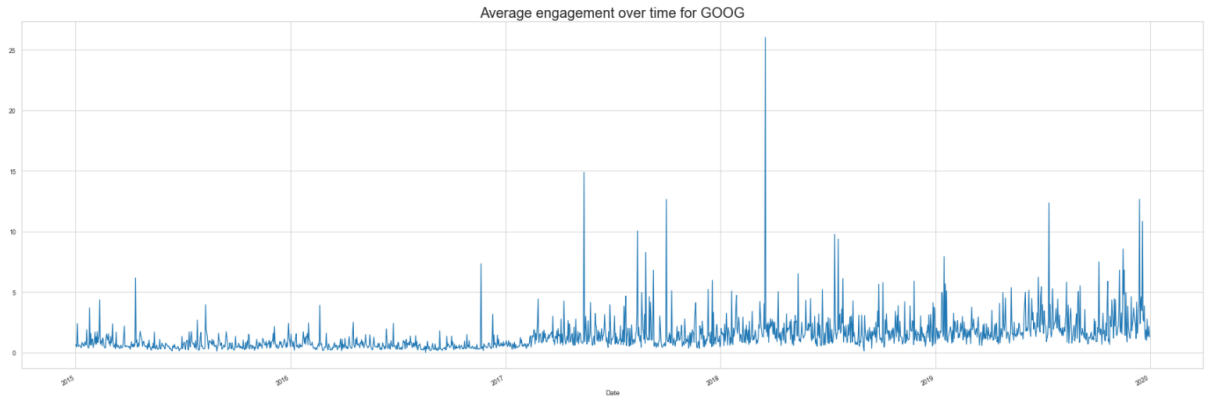
Appendix D – Average Engagement




Average engagement over time for TSLA
Date

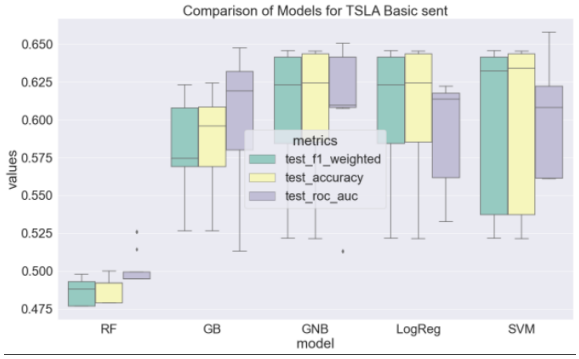

Average engagement over time for AMZN
Date


Average engagement over time for GOOGL
Date

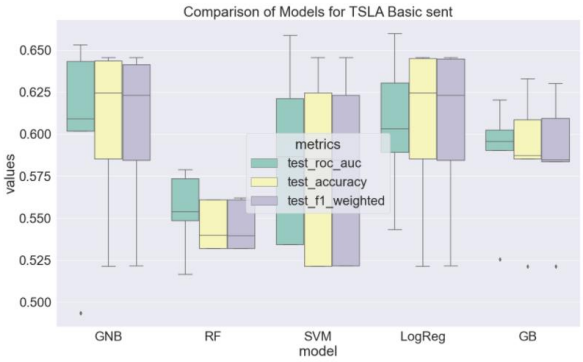

Average engagement over time for GOOG
Date

Appendix E – Comparing threshold 0.1 results to threshold 0.14 results

Basic:

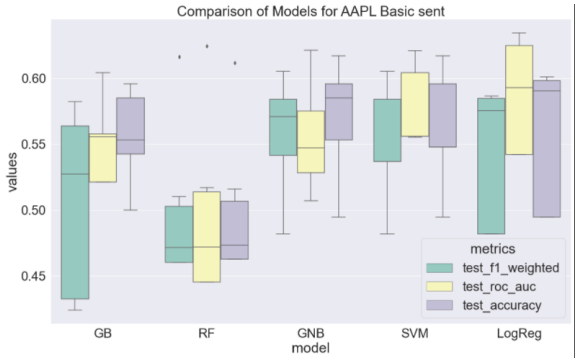


Tesla Basic Threshold (0.1) with Sent

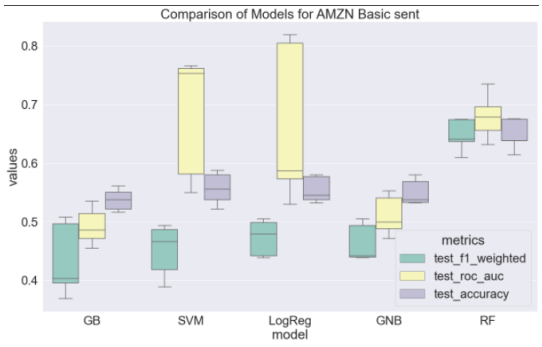


Tesla Basic Threshold (0.14) with Sent

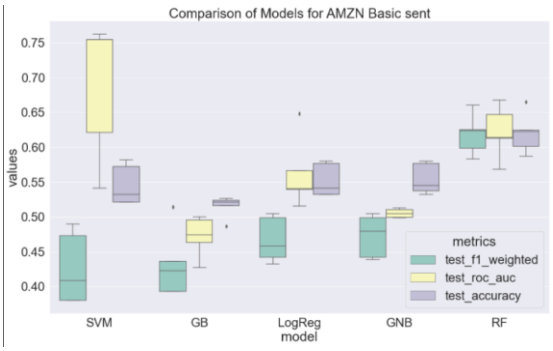
Apple Basic Threshold (0.1) with Sent



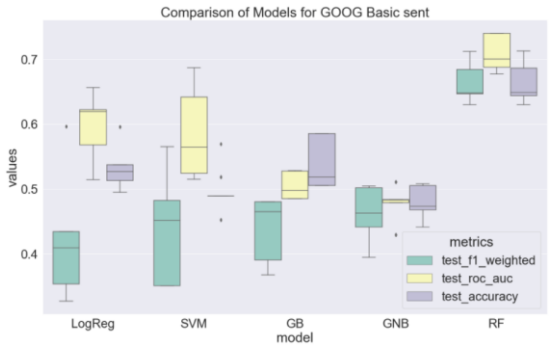
Apple Basic Threshold (0.14) with Sent



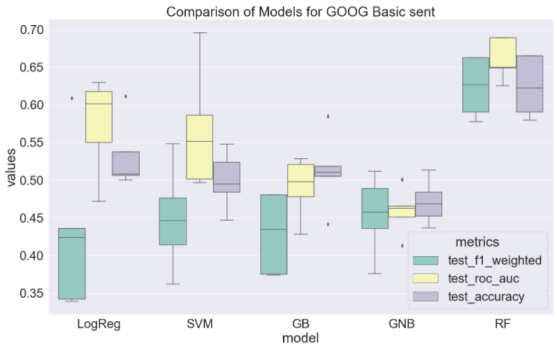
Amazon Basic Threshold (0.1) with Sent



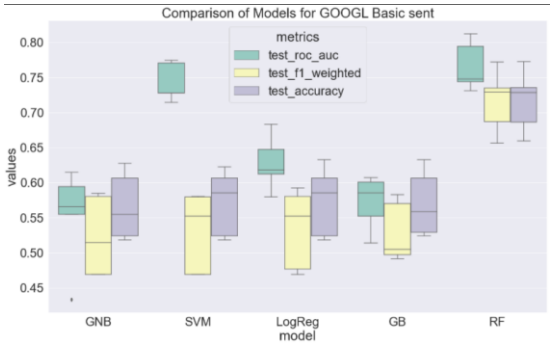
Amazon Basic Threshold (0.14) with Sent



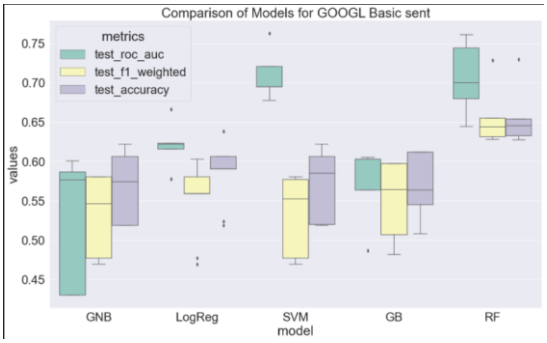
Goog Basic Threshold (0.1) with Sent



Goog Basic Threshold (0.14) with Sent

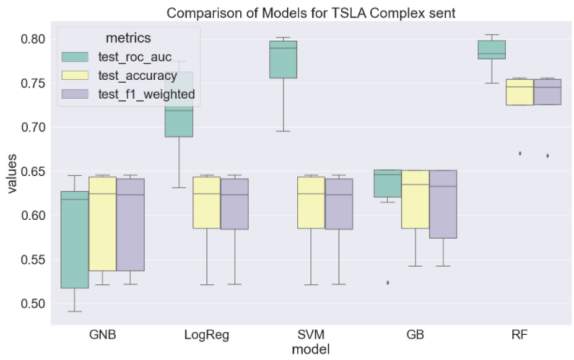


Googl Basic Threshold (0.1) with Sent

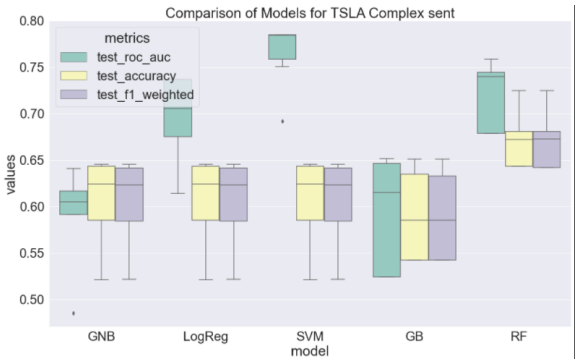


Googl Basic Threshold (0.14) with Sent

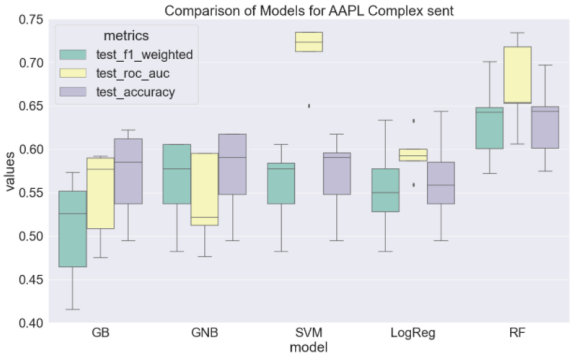
Complex:



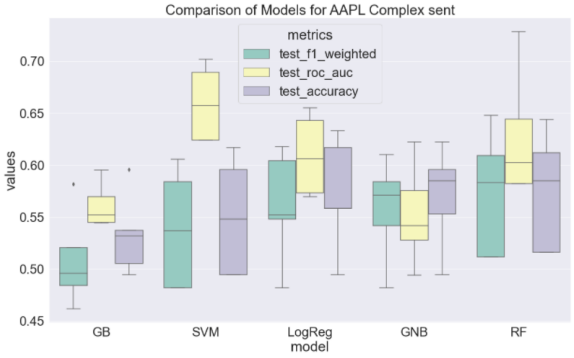
Tesla Complex Threshold (0.1) with Sent



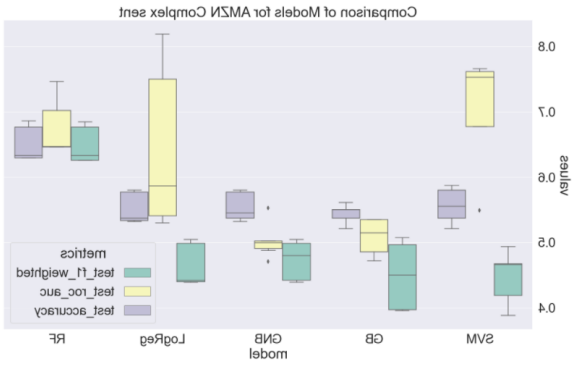
Tesla Complex Threshold (0.14) with Sent



Apple Complex Threshold (0.1) with Sent



Apple Complex Threshold (0.14) with Sent

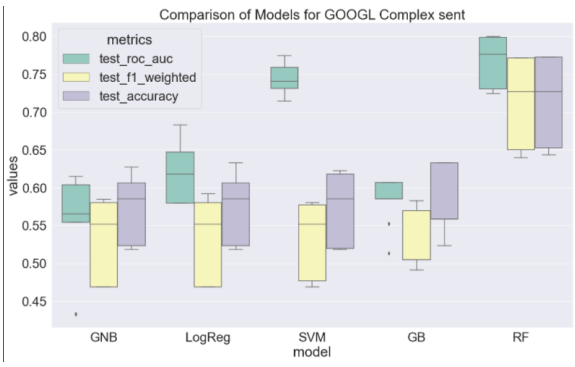


Amazon Complex Threshold (0.1) with Sent

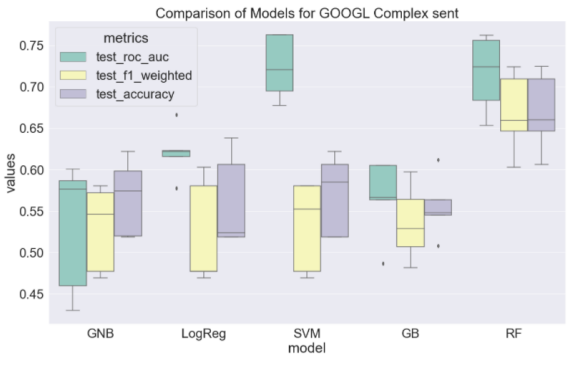
Amazon Complex Threshold (0.14) with Sent

Goog Complex Threshold (0.1) with Sent

Goog Complex Threshold (0.14) with Sent



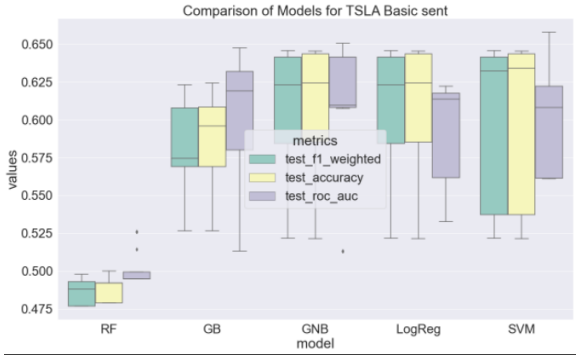
Googl Complex Threshold (0.1) with Sent



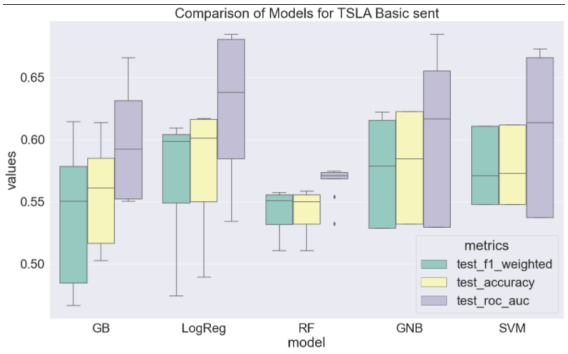
Googl Complex Threshold (0.14) with Sent

Appendix F – Comparing threshold 0.1 results to No threshold results

Basic:

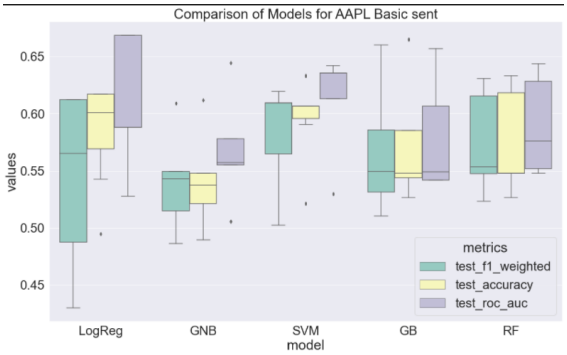


Tesla Basic Threshold (0.1) with Sent

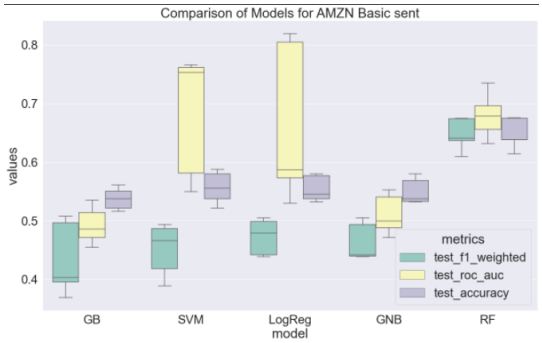


Tesla Basic No Threshold with Sent

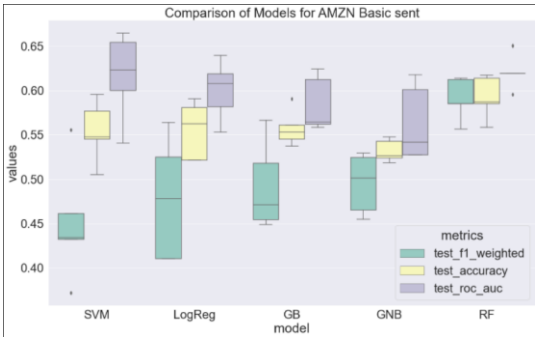
Apple Basic Threshold (0.1) with Sent



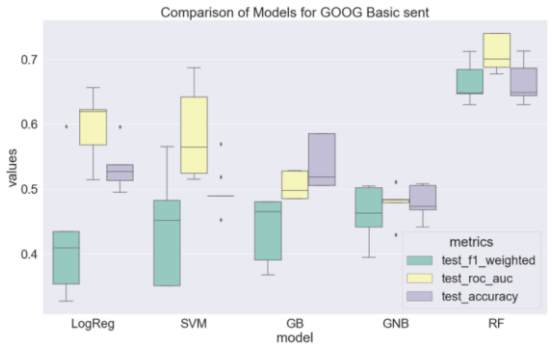
Apple Basic No Threshold with Sent



Amazon Basic Threshold (0.1) with Sent

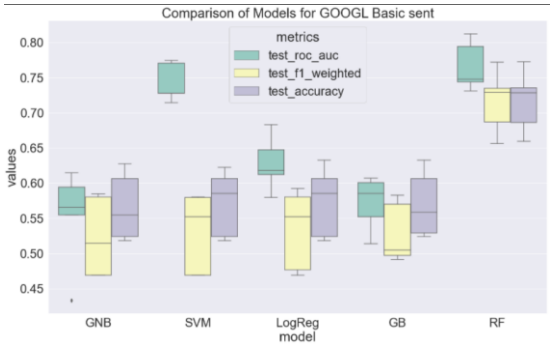


Amazon Basic No Threshold with Sent

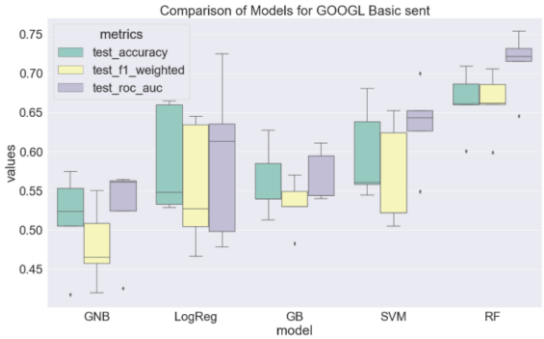


Goog Basic Threshold (0.1) with Sent

Goog Basic No Threshold with Sent

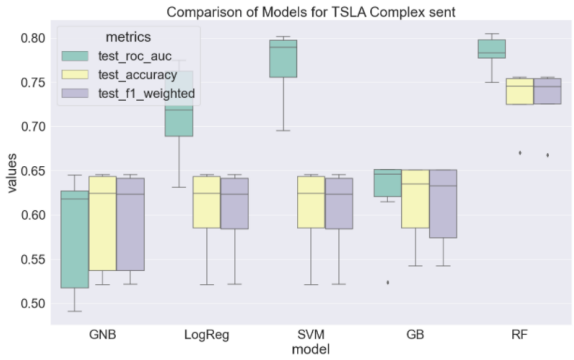


Googl Basic Threshold (0.1) with Sent

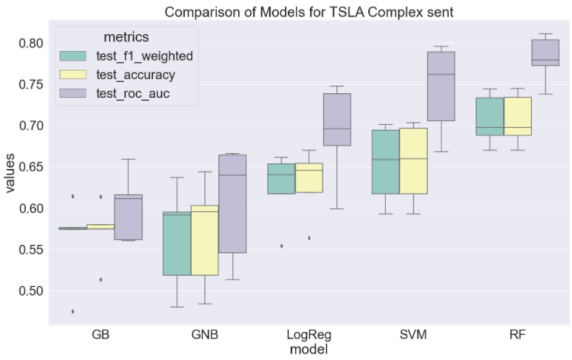


Googl Basic No Threshold with Sent

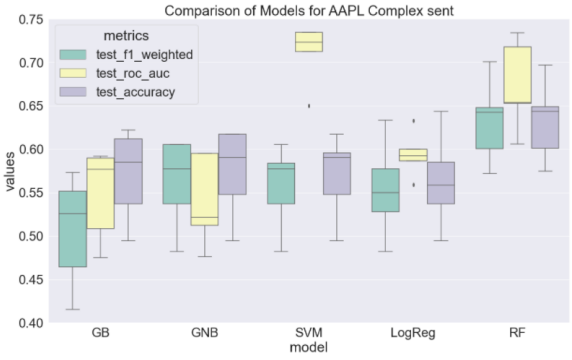
Complex:



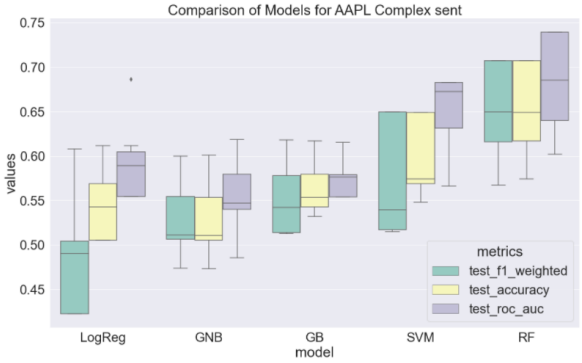
Tesla Complex Threshold (0.1) with Sent



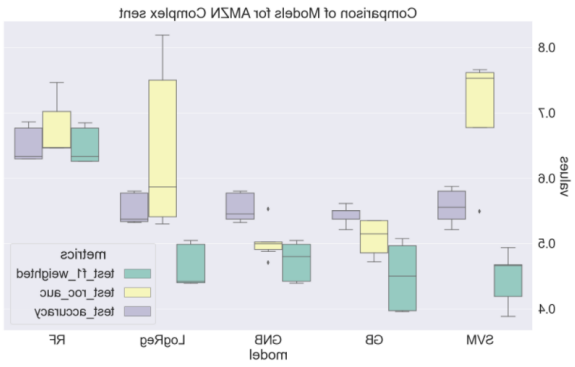
Tesla Complex No Threshold with Sent



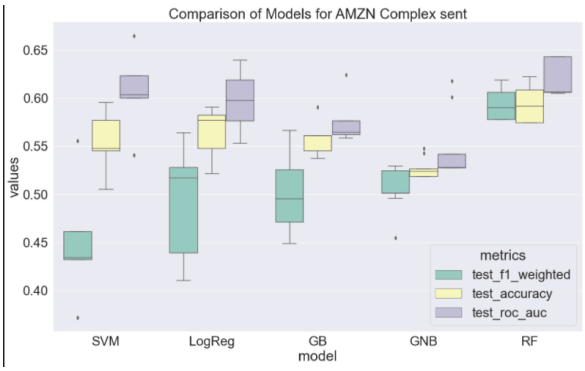
Apple Complex Threshold (0.1) with Sent



Apple Complex No Threshold with Sent

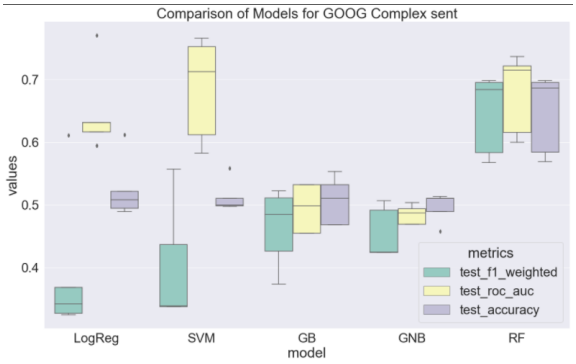


Amazon Complex Threshold (0.1) with Sent

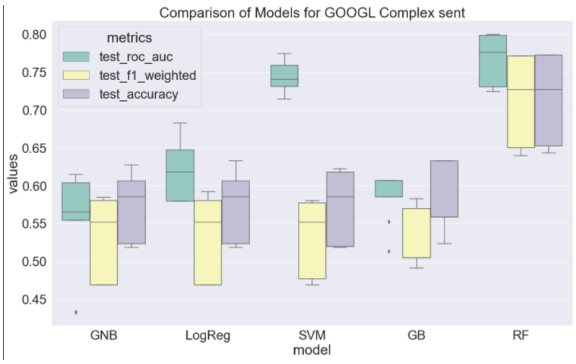


Amazon Complex No Threshold with Sent

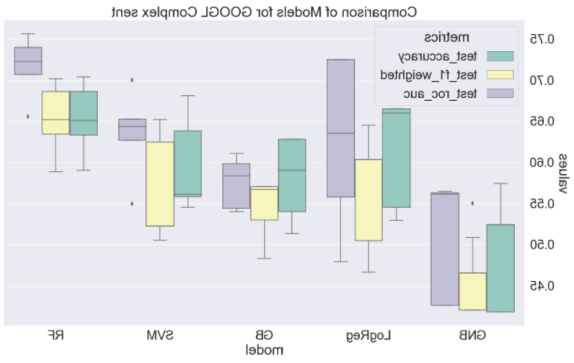
Goog Complex Threshold (0.1) with Sent



Goog Complex No Threshold with Sent



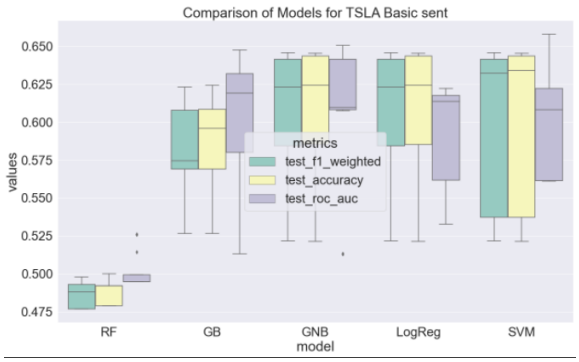
Googl Complex Threshold (0.1) with Sent



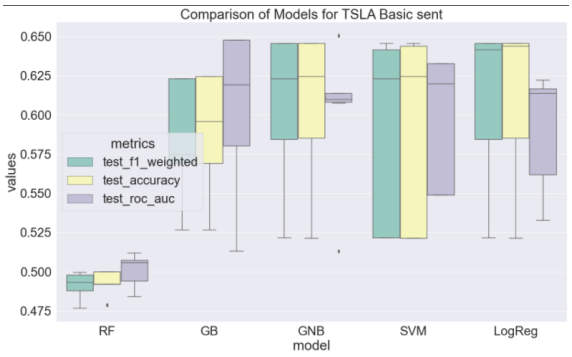
Googl Complex No Threshold with Sent

Appendix G – Comparing Threshold 0.1 Minmax Scaler results to Threshold 0.1 Standard Scaler results

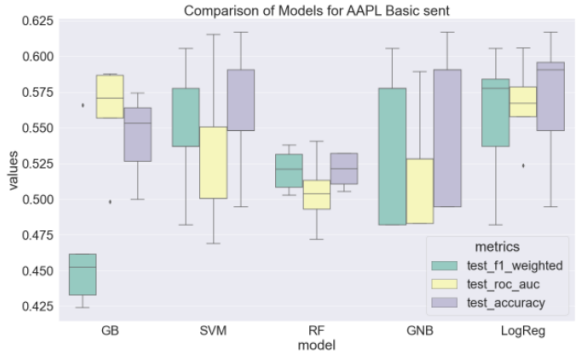
Basic:



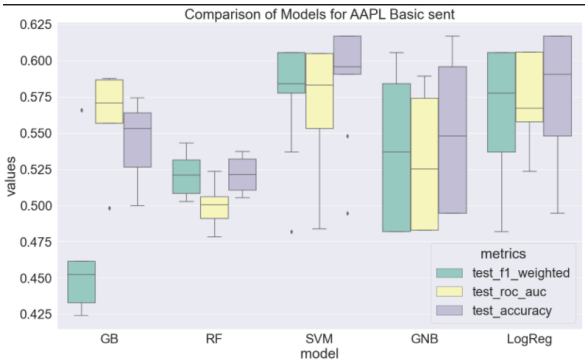
Tesla Basic Threshold (0.1) with Sent MinMax



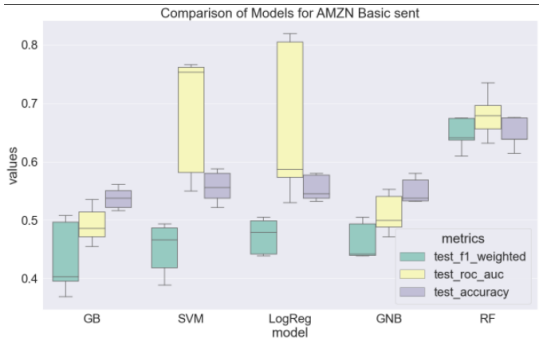
Tesla Basic Threshold (0.1) with Sent Standard



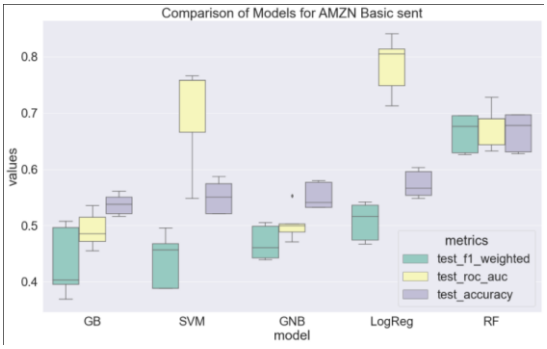
Apple Basic Threshold (0.1) with Sent MinMax



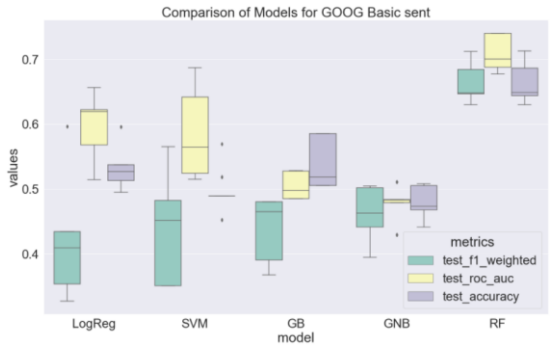
Apple Basic Threshold (0.1) with Sent Standard



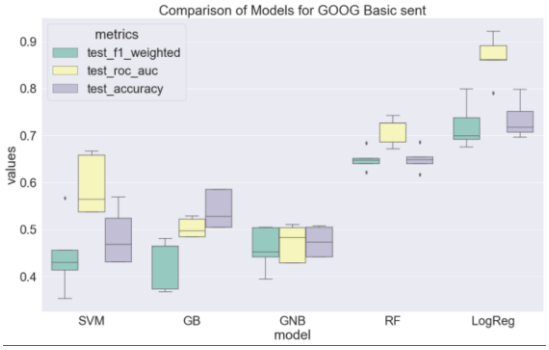
Amazon Basic Threshold (0.1) with Sent MinMax



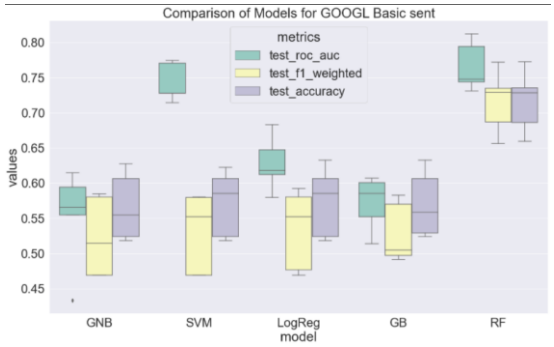
Amazon Basic Threshold (0.1) with Sent Standard



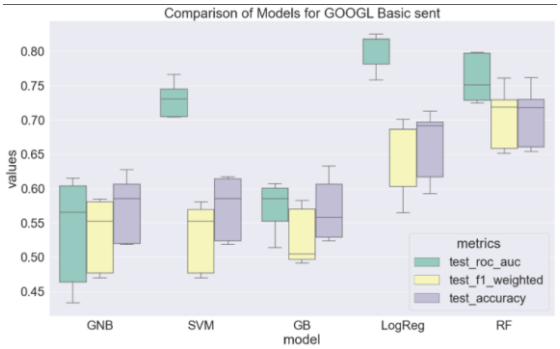
Goog Basic Threshold (0.1) with Sent MinMax



Goog Basic Threshold (0.1) with Sent Standard

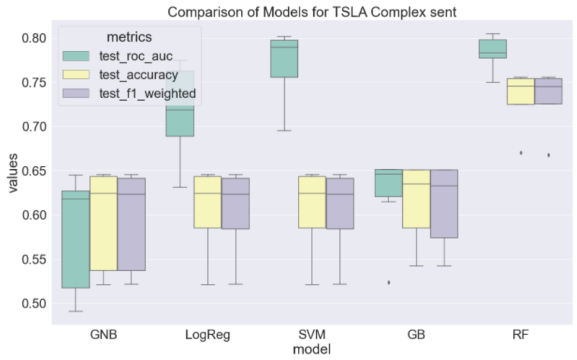


Googl Basic Threshold (0.1) with Sent MinMax

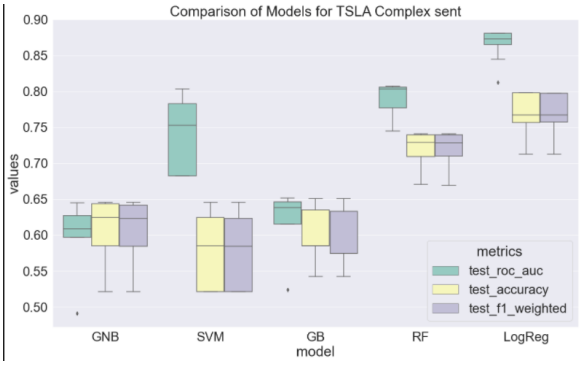


Googl Basic Threshold (0.1) with Sent Standard

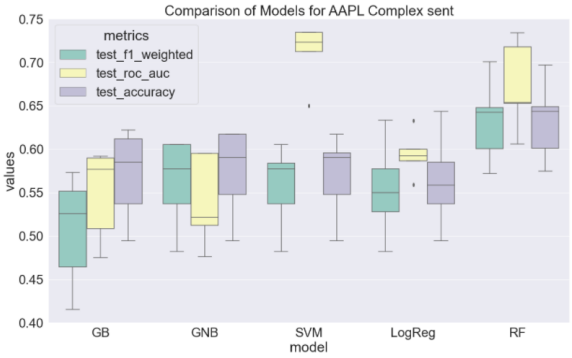
Complex:



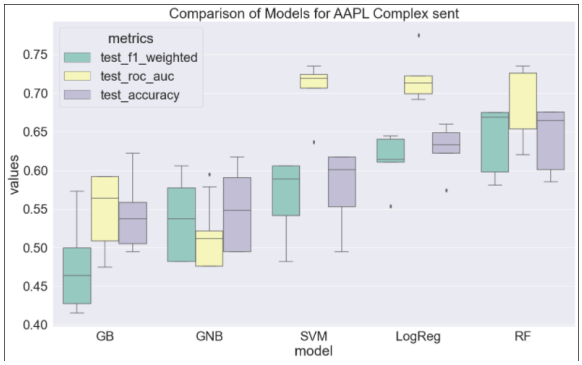
Tesla Complex Threshold (0.1) with Sent Minmax



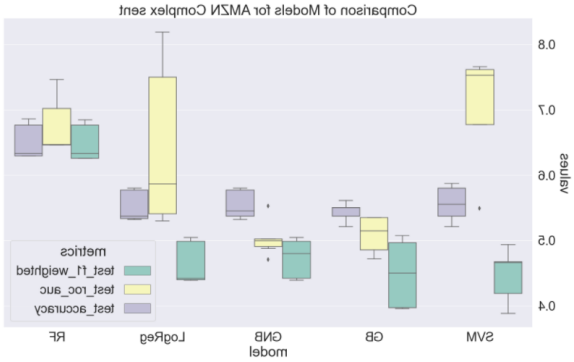
Tesla Complex Threshold (0.1) with Sent Standard



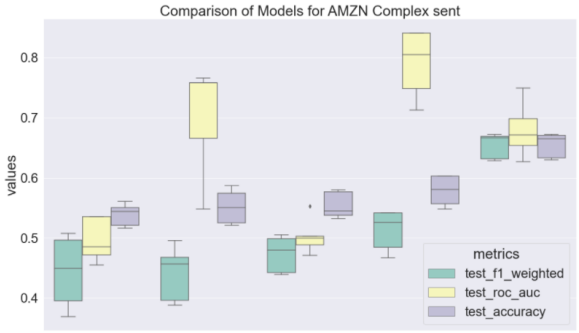
Apple Complex Threshold (0.1) with Sent Minmax



Apple Complex Threshold (0.1) with Sent Standard

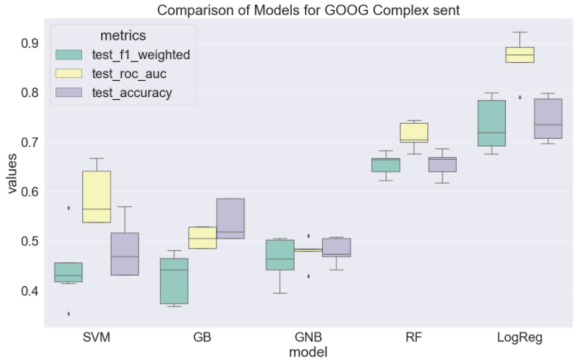


Amazon Complex Threshold (0.1) with Sent Minmax

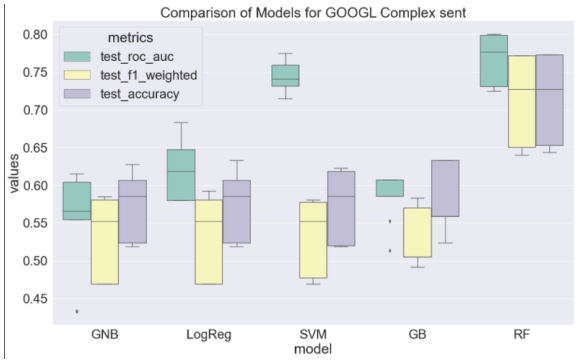


Amazon Complex Threshold (0.1) with Sent Standard

Goog Complex Threshold (0.1) with Sent Minmax



Goog Complex Threshold (0.1) with Sent Standard

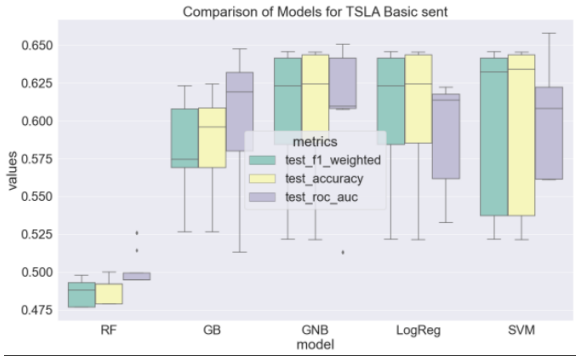


Googl Complex Threshold (0.1) with Sent Minmax

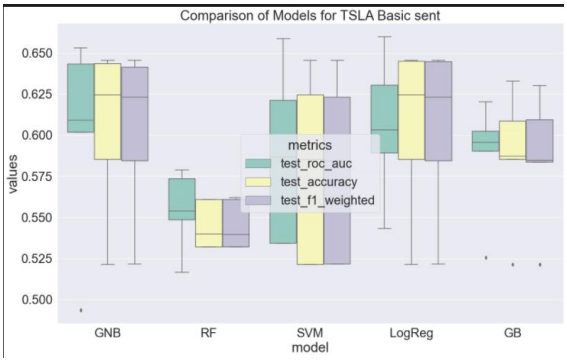
Googl Complex Threshold (0.1) with Sent Standard

Appendix H – Comparing threshold 0.1 sentiment class only results to threshold 0.1 sentiment class with average engagement per day results

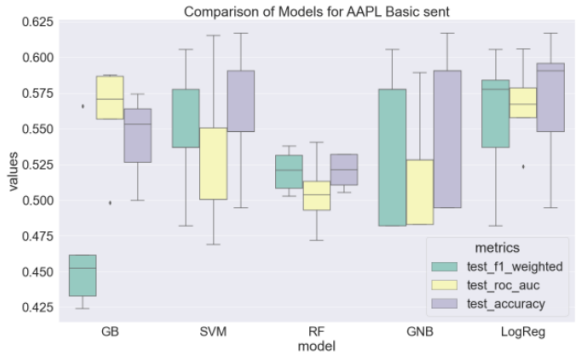
Basic:



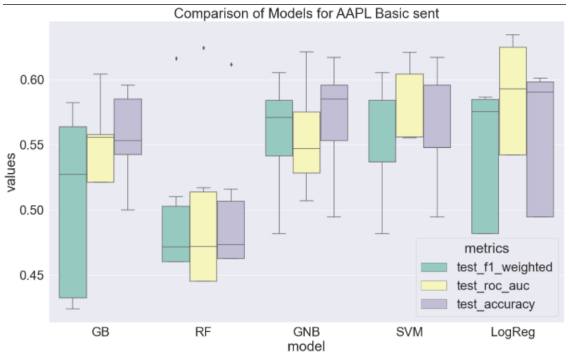
Tesla Basic Threshold (0.1) with Sent & No Average Engagement



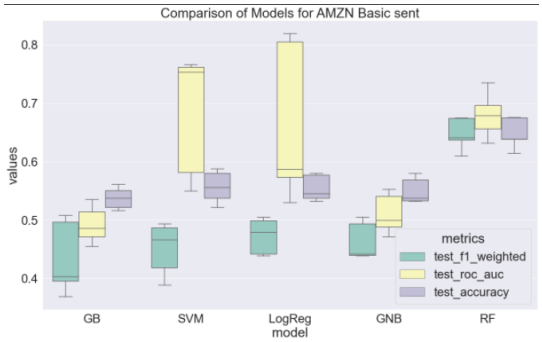
Tesla Basic Threshold (0.1) with Sent & Average Engagement



Apple Basic Threshold (0.1) with Sent & No Average Engagement



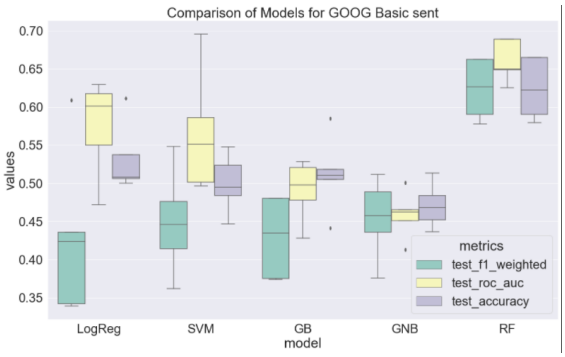
Apple Basic Threshold (0.1) with Sent & Average Engagement



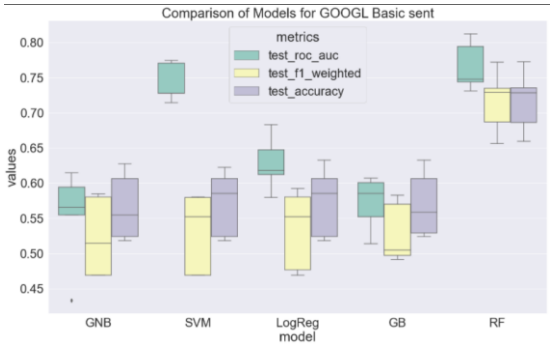
Amazon Basic Threshold (0.1) with Sent & No Average Engagement

Amazon Basic Threshold (0.1) with Sent & Average Engagement

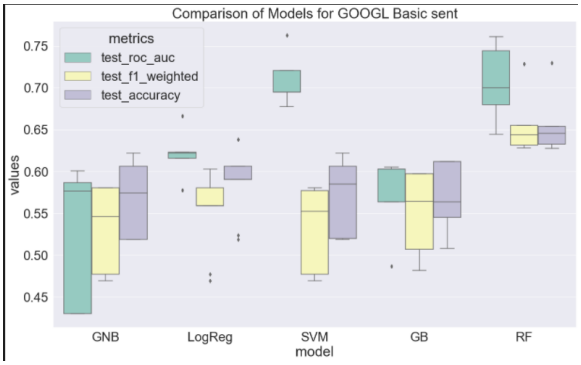
Goog Basic Threshold (0.1) with Sent & No Average Engagement



Goog Basic Threshold (0.1) with Sent & Average Engagement

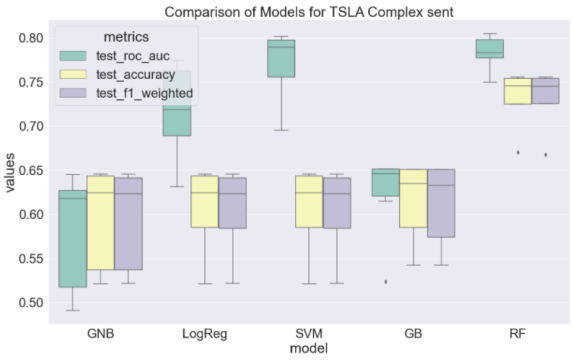


Googl Basic Threshold (0.1) with Sent & No Average Engagement

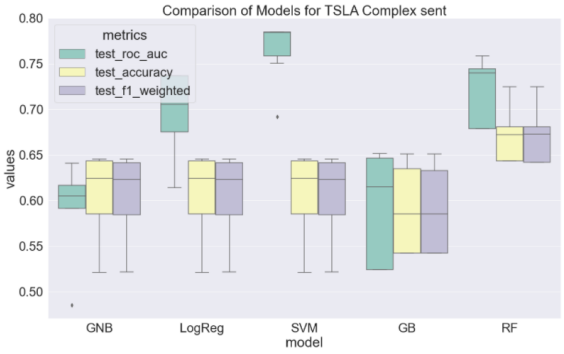


Googl Basic Threshold (0.1) with Sent & Average Engagement

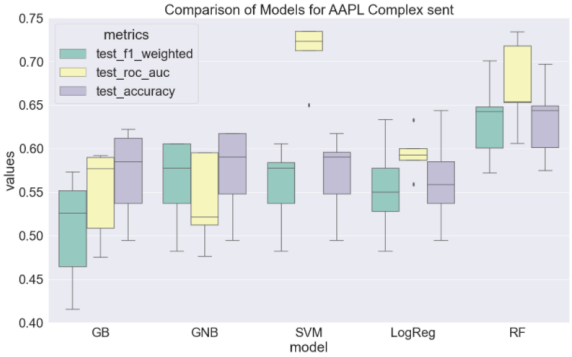
Complex:



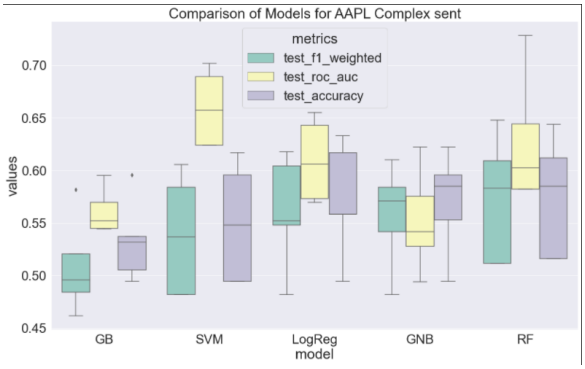
Tesla Complex Threshold (0.1) with Sent & No Average Engagement



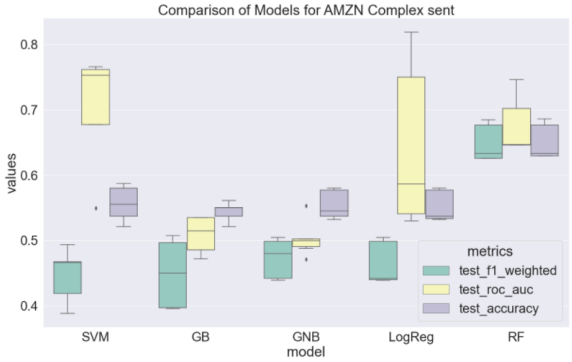
Tesla Complex Threshold (0.1) with Sent & Average Engagement



Apple Complex Threshold (0.1) with Sent & No Average Engagement



Apple Complex Threshold (0.1) with Sent & Average Engagement

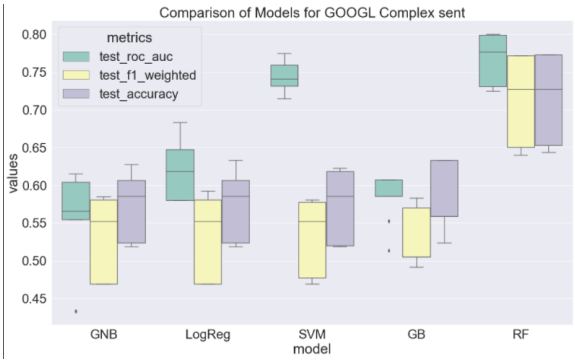


Amazon Complex Threshold (0.1) with Sent & No Average Engagement

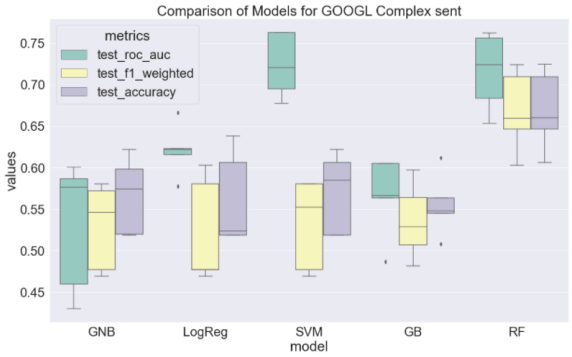
Amazon Complex Threshold (0.1) with Sent & Average Engagement

Goog Complex Threshold (0.1) with Sent & No Average Engagement

Goog Complex Threshold (0.1) with Sent & No Average Engagement



Googl Complex Threshold (0.1) with Sent & No Average Engagement



Googl Complex Threshold (0.1) with Sent & No Average Engagement

Appendix I – Weekly Break Down

Semester 1

Week 1

- Project Approval

Week 2

- Introduction, Hypothesis, Motivations and Objectives for this Project
- Start reading papers

Week 3 – 6

- Continue reading papers and start writing literature review

Week 7 – 10

- Continue reading papers and start writing literature review
- As there is an idea on general principles, design methodology overview on draw.io for methodology
- Research different evaluation metrics
- Design methodology based on methodology overview
- Outline the evaluation metrics

Week 11 – 12

- Add in risk analysis and PLESE issues
- Proof-read and submit D1

Semester 2

Week 1

- Obtain a dataset from Kaggle and Yahoo Finance
- Start on implementation

Week 2-5

- Re-write Introduction and Objectives
- Improve literature review and methodologies
- Add in functional and non-functional requirements
- Continue with implementation
- Continue reading papers and start writing literature review

Week 7 – 10

- Finish implementation and start analysis on results achieved
- Start writing technical implementation

- Start writing evaluation from the results achieved
- Re-write Risk Analysis

Week 11 – 12

- Finish evaluation
- Finish conclusion
- Fine tune report by adjusting certain parts
- Proof-read and submit D2 and code