The Chinese University of Hong Kong
Department of Computer Science and Engineering
Final Year Project Report (Semester 1)

Stock Trend Prediction with News Data using Deep Learning

LYU2004

Supervisor:
Professor Michael Rung Tsong LYU

Member:
Li Kam Po 1155108393
Lau Tsz Yui 1155108869

Abstraction

Stock price prediction is a challenging topic as there are many factors that are contributing to the changes in demand and supply. However, stock price prediction is the action that people have been doing for a long time. In order to find a perfect timing for transaction, build a portfolio to reduce the risk. And the goal is to make more money. There are some techniques that are frequently used to analyze stock price, including technical analysis, fundamental analysis, and quantitative analysis, etc. With machine learning algorithms are getting more and more powerful as well as the support from the hardware in recent years. People have started to leverage artificial intelligent to study the stock market.

Investors combine every information they have to make a transaction decision. Besides analyzing historical stock price, the other important information source is the daily news. Therefore, we think news is another contributing consideration of their decision making. In this paper, we will use machine learning to analyze the historical data of the stock price. The data we use are the open, close, high, low, adjust close, and volume of standard and poor's 500 index, Nasdaq composite index and Apple Inc, to predict the close price of the next day of Apple Inc. Besides numerical analysis, we also perform sentiment analysis on news data that is related to Apple. We crawl data from New York Time and analysis the sentiment of the headlines and the leading paragraphs in each article.

Acknowledgements

We are grateful and take this opportunity to sincerely thank our final year project supervisor, Professor Michael Rung Tsong LYU, for his generous support, practical suggestion, and encouragement. His words from weekly progress report really inspire us on this project.

We would like to extend our sincere appreciation to Edward YAU, who is this final year project assistant, for his constant support. He always open to questions and provides invaluable guidance. This project would not have been successful without his generous support.

We would also want to thank the members of other teams. We are getting a lot inspiration and ideas from their remarkable work.

Table of Content

Table of Figure

1. Introduction

Stock trading is a way of making money by buying low and selling high. Most of the investment in stocks is to earn money or to protect against inflation. Investors need to analyze whether a company can make money in the future and then decide when to buy. There are many aspects to analyze whether a company can make money. The most intuitive way is to observe the company's business scope, for examples technology, medical care, real estate, etc. Some businesses will make more money than others, such as technology. A company's financial report is another way to reflect the company's ability to make money. A company with a good financial record is intended to distribute more dividends and attract more people to buy the company's stock, which will increase the demand for the stock.

There are many ways to indirectly reflect the company's ability to make money. We can tell if other people believe that the company will make money in the future by analyzing the stock data. There are many methods of analyzing stock data, investor can directly apply them to analyze past stock data, to analyze the future upside and trend of stocks, and then deciding whether to buy stocks, so that to achieve buy low and sell high. At the same time, any good news or negative news about the company will affect whether investors believe the company can make money, and these news can change the stock price in the short term. Paying attention to whether the company's internal personnel hold company stocks can be one of the considerations. Because whether company personnel hold stocks reflects whether they trust the company. Insiders will know the company's development better than others. Of course, what is referred to here is not insider trading, but whether as an employee believes in the company's ability to make money in general.

The above roughly introduced the method of analyzing stocks. The rise and fall of stock prices are the changes in demand and supply surrounding whether investors trust the company's ability to make money. This makes stock trading different from gambling, which is a completely unpredictable random event and its expected profit is negative. Stock investment depends not on luck but on analysis to make money.

As more and more people invest to the field of artificial intelligent, this technology has become more common and close to people. It has a wide range of application and is good at data analysis, which attract companies to it refine their business. Data is not rare as it was in the past and become more valuable. Organizations start to collect their business data and even some of them are willing to public their collected data. The reason why artificial intelligent can develop so fast is also because data acquisition of data has become easy.

The topic of this final year project is prediction stock trend with historical data and news data using machine learning and deep learning. To be more specific, it will focus on predicting the close price of the next day. Including classify is the stock will increase or decrease in the next day and estimate the close price of the next day. This chapter will give an introduction and a brief overview of this final year project.

1.1 Background

1.1.1 Deep Learning

Neural network was started by understanding how the neurons inside the brain can work together to solve problem. It found that the based on the "all-or-none" characteristic of the neural operation and the relations among the neurons, they can perform complex logical operation (McCulloch, W.S., Pitts, W., 1943). However, although people start the discussion on neural network since 1940 era, it experienced up and down and not widely used in the following decades. The computational power of neural network is depending on the complexity of the weights of the network. [1]. If you want the neural network to show powerful computational power, you need to prepare a computer with powerful computing capabilities. Until now, in the recent decades, the chip industry has achieved breakthrough development, power of GPU and TPU catch up the requirement of applying neural network. And more and more researches are conducted on machine learning.

Today, machine learning is moving to a higher level and is developing to deeper and wider, which is called deep learning. The computational model consists of multiple layers that learn data representations with multiple levels of abstraction, so that we can use it to find structure in a complex dataset. It helps us to achieve breakthrough developments in many aspects, such as speech recognition, image recognition, etc.. [2]. Using deep learning can save us a lot of time in feature engineering. Feature engineering is the basic work in machine learning, which improves the accuracy of the model by extracting useful features from raw data. Compared with other machine learning algorithms, the main advantage of deep learning is its ability to automatically perform feature engineering. It can scan data, search for and combine relevant features, so that deep learning is more likely to find that people may miss or more It is a complex combination method, while saving the time of manual combination. Another benefit of deep learning is that there is no need to label data. Efficient training needs to label the data, but under different data sets, labeling the data may be a long and expensive task. For example, the data set is very large, or professional knowledge is needed to complete the classification work, such as cancer cell image classification, weather image classification, etc. [3].

Improving the expressive ability of deep neural networks will increase the depth of the network exponentially [4]. Many studies have been conducted on the depth of neural networks, however, the number of neurons in each layer affects the expressive ability of the entire neural network. Moreover, the expressive power of a wide neural network cannot be realized by a narrow and deep neural network [5]. Deep neural network requires a very long training time, depending on the depth of the visual symbol network, it may take days or weeks to complete [6]. When doing deep learning, it would be better to start with a finer-scale neural network to reduce the time required for training. Then gradually increase the width and depth of the network, while avoiding overfitting.

1.1.2 Relations between News and Stock Market

When investors decide whether to buy a certain stock, they will use a variety of information, including stock historical data, company financial statements and magazines, news and so on. In the face of different company businesses, the information that investors value will be different. For example, when considering whether to buy stocks in pharmaceutical companies, investors may not pay much attention to the company's financial statements, because pharmaceutical companies are in the early stage of developing new drugs, and the company's income mainly negative values, the company will not have any signs of making money until the company successfully develops a new drug. In this way, investors will focus on the company's announced R&D progress or other phased results, as well as the current obstacles encountered. These news are likely to be reported by the news. Investors then decide whether to buy stocks based on this information. Therefore, we believe that news is different from the company's financial statements. Both the good news and the negative news brought out by the news will be noticed by investors, and the company's business will not reduce the attention to news.

There is indeed a positive correlation between news and stock trading volume or financial market trends [7]. The study pointed out that the trading volume of stocks mentioned in the financial news increased significantly on the day before and on the day the financial news was released. The change in trading volume means that the balance between supply and demand is broken, and the price of stocks is bound to change. Therefore, by knowing whether the news is positive news or negative news, you can roughly know the stock's short-term trend. At the same time, it means that adding news and new information to the machine learning model helps us to study the trend of stocks. Therefore, what we need to do is quantifying news as a sentiment value before feed to the model.

1.1.3 Candlestick Chart

The candlestick chart is one of the important tools for investors to analyze stock market trends. Each candlestick represents four pieces of information, the opening price, the closing price, the highest price in the day, and the lowest price in the day. These four pieces of information make each candlestick have a different pattern, and the different patterns not only show the offensive of buyers and sellers in the market, but also hint at the future trend of the stock market.

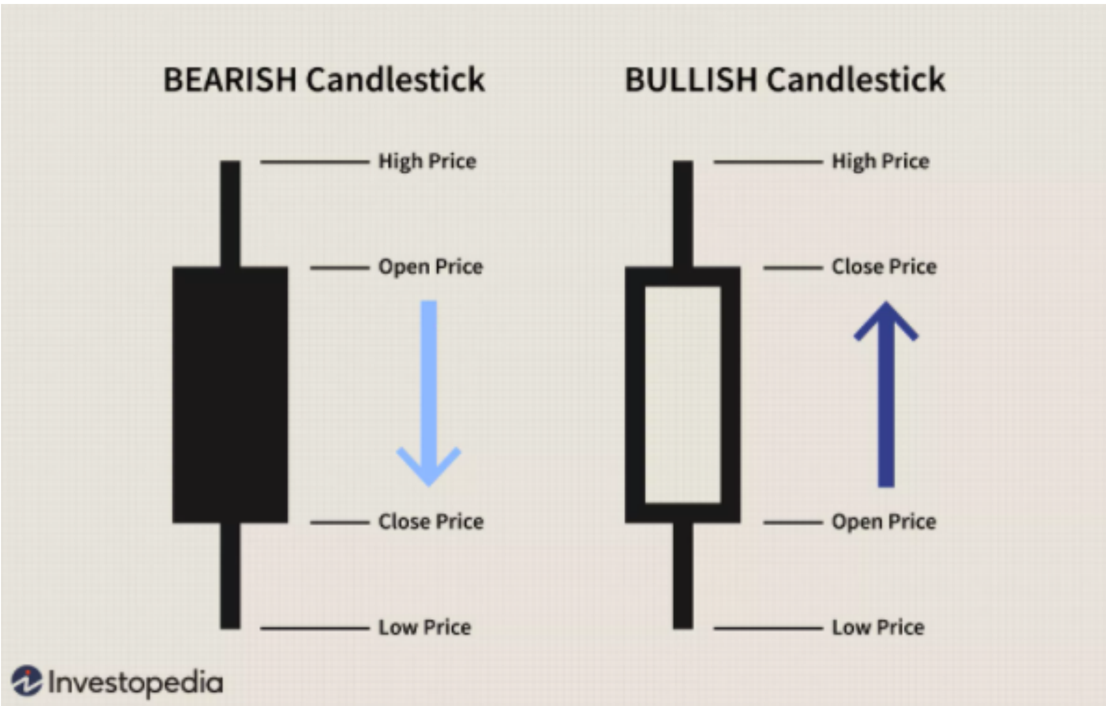


Figure 1: Candlesticks https://www.investopedia.com/articles/active-trading/062315/using-bullish-candlestick-patterns-buy-stocks.asp

There are two type of candlestick, bearish candlestick meaning that opening price is high than closing price, while a bullish candlestick meaning that closing price is high than opening price. A candlestick char may use different type of candlestick to represent it is a bearish or a bullish candlestick, for example hollow candlesticks represent bullish, solid candlesticks represent bearish. Sometime, some charts use different colors to represent whether it is increase or decrease. To study the chart, there are few terminologies, real body is the thick bar with one end showing the opening price and one end showing the closing price; upper shadow is the upper thin stick with the upper end pointing to the highest price of that date; lower shadow is the lower thin stick with the lower end pointing to the lowest price of that date.

There are many forms of candlesticks, such as hammer, spinning top, doji, dragonfly joji. A candlestick with almost no upper shadow, a short body and a long lower shadow. Hammer indicates that initially supply is increasing, buy then more buyer joined the market and make the price increased. It is a hint of an increase trend

Figure 2: A candlestick in hammer form

Shooting start is s hammer-like formation occurs at the end of an uptrend, which indicates the beginning of a decline and the weakening power of buyers. A small body (rising or falling) and a candle with a long top shadow.

Figure 3: A shooting star candlestick

Bullish engulfing candle is a pattern of two candlesticks, a bearish candlestick following a bullish candlestick. There are some essential criteria, the body of the bullish candlestick must largest than and completely cover the body of the bearish candlestick, also the bearish candlestick is not a doji. This sign shows that the trend will continue be positive.

Figure 4: Bullish engulfing candle

Bearish engulfing candle is similar to bullish engulfing candle, but it will be a bullish candlestick follows a bearish candlestick. The body of the bearish candlestick must largest than and completely cover the body of the bullish candlestick. This pattern indicates that the trend is negative trend.

Doji happens when the price fluctuates at the same level (or a certain degree from the opening to the closing), a doji is formed. What happens after the doji and the price level at which this candlestick pattern appears makes it meaningful. Generally, the doji may occur near the resistance level. After this, when you see the bearish market after the Doji, it may indicate that there are more sellers in the market. However, if you see the bullish candlestick after the doji, you can infer that the market trend is upward.

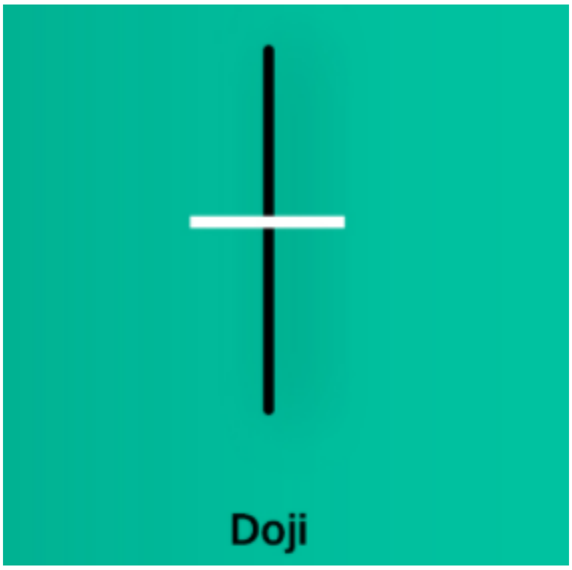


Figure 5: Doji

1.2 Motivation

One of the trading modes of stocks is to buy today and sell tomorrow, BTST trading. The profit method of BTSK trading is very clear, which is to find a stock that will appreciate tomorrow and obtain profit through frequent trading. The problem with this trading method is his trading costs. In addition to rising stocks, the difference earned must cover the overall trading costs in order to be profitable. On the other hand, the risk prediction of this method is very clear, that is, the rate of rise and fall of a stock in one to two days. The risk of applying BTST trading on large-cap stocks is lower, in usual, it is just a few percentage change on one dat. On the contrary, the risk of applying BTST trading on small-cap stocks is higher, the stock prices of them can change with dozens of percentages.

Many work have been done on stock prediction with historical stock data. We think news is also an import factor to stock to stock volatility. And there are relatively fewer researches done on predicting stock with news, which are mainly conducted within the recent decade. We hope to use mechanical learning to analyze the short-term rise and fall patterns of stocks and combine news information to help investors who mainly use BTST trading mode choosing the timing to buy the stock and increasing the their success rate of making money.

1.3 Objective

In this project, we will analysis whether the close price of the next trading day increase or not. The trading cost is ignored in this project. Our objective is to build a model with two components, including numerical analysis for stock historical record and sentiment analysis for news information.

We will conduct experiment with k-nearest neighbors algorithm (KNN), gate recurrent unit (GRU), long short-term memory (LSTM), Prophet for the numerical analysis. For sentiment analysis, besides LSTM, we will also use Valence aware dictionary and sentiment reasoner (VADER) and Bidirectional encoder representations from transformers (BERT). The last experiment is to combine the two model to get a better prediction.

1.4 Report Overview

This report describes the analysis of stock trend of Apple Inc. (AAPL) with stock and news data.

In chapter 2, we describe the studies have been done on stock analysis with sentiment information of news and social medica.

In chapter 3, we describe the methodology used to crawl out dataset, as well as the preprocess jobs done on the dataset. This chapter also introduces the model we will used in the coming experiment.

In chapter 4, we describe the experiments done with different models on numerical analysis, sentiment analysis, and the models that combine the two types of analysis. For each experiment, we will discuss the result and out finding.

In chapter 5 & 6, we conclude all of our experiments and the future development in the next semester.

2. Related Work

There are many stock prediction researches have been done in the past. For example predicting stock market with Prophet upon ARIMA [8], it provides steps of how to use Prophet to forecast the market. Or using artificial neural network and random forest to perform the prediction [9], they showed that using artificial neural network can gives a better prediction than using random forest.

There are also some researches that performed textual analysis on news or social media to assist the stock prediction. For example, there is a research using multiple machine learning technique , including single later and multi-layer perceptron, and support vector machine etc., to predict the performance of closing of Karachi Stock Exchange(KSE) [10]. It used oil rate, gold and silver rates, interest rate, foreign exchange rate as well as news and media as the input. Statistic techniques simple moving rate and Autoregressive Integrated Moving Average are also used as input. Their conclusion is that MLP performed the best among all models and the oil rate is most relevant feature to the performance of closing price of KSE. There is a project that predict the influence of news to the stock trend, by studying the headline of news as well as the historical stock prices, this project achieved 78% accuracy in predicting the influence of new to the stock price [11]. Besides news, social medical also contains many useful information for stock prediction. Two researchers find out that information on twitter also affect the Dow Jones Industrial Average value. They analysis the sentiment of twitter posts to obtain the public mood. They result in decent profit over 40 day [12].

3. Methodology

3.1 Data Crawling

Stock Data

The way to obtain historical stock data is using a python package which is called Pandars-Datareader. The API from this package provides multiple data sources, and we use Yahoo Finance. Our main analysis object is Apple, so we have obtained Apple's data from this data source. The daily high, low, opening, closing, adjusted closing price and trading volume. At the same time, data from the Standard & Poor's 500 Index and the Nasdaq Composite Index in the same period are also taken, in order to understand the economic trend in the market.

News Data

There are many news datasets available in the internet. We downloaded a dataset called "Sentiment Analysis for Financial News" and a data set called "All the news" from Kaggle. The first dataset contains two columns, a column of "news headline", which is nicely labeled with the sentiment classification with another column "sentiment". That makes the dataset good for training sentiment analysis model. There are three type of sentiment, positive, neutral and negative. The second data contains article titles, publishing data, content of the article as well as other information. We also find a dataset specifically about "Apple" later. However, the dataset mentioned are the dataset crawler by other people in the past, they will not be updated every single day. Now we have the past news for training purpose, we do need a news source that can provides us the daily news about apple. It is important because daily news is one of the input features to perform daily prediction.

In order to obtain latest news every day. We wrote a crawler to get the news for us. We targeted a stock website called MarketWatch, the reason why we choose this website is that they can provide news which are related to Apple. The crawler we wrote will execute once per day to collect the latest update in the recent news section.

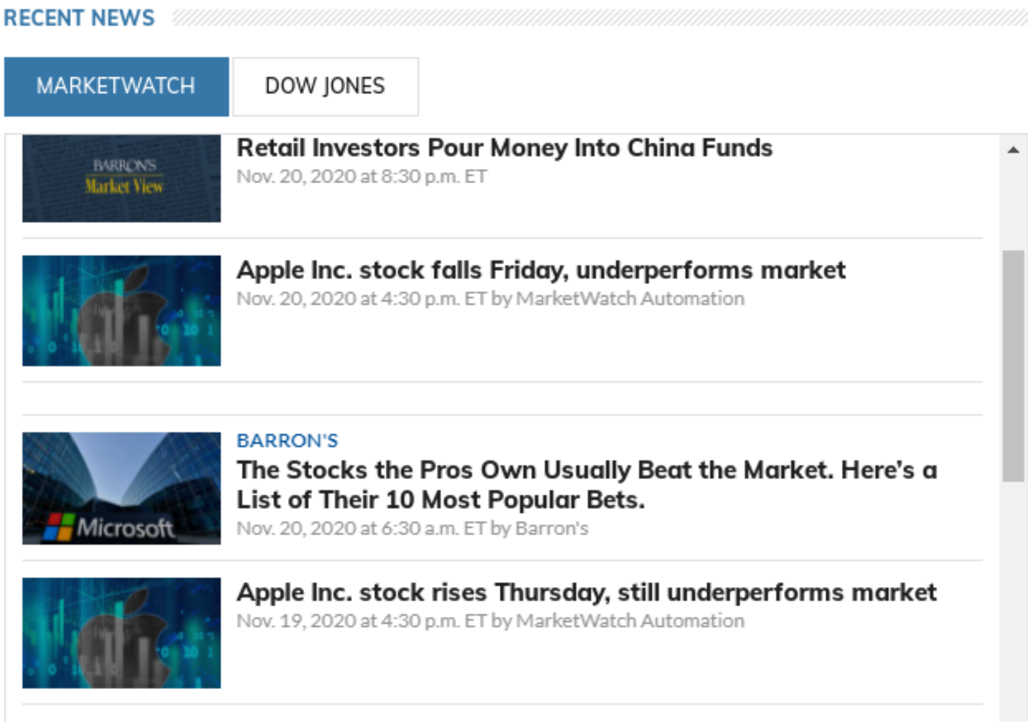


Figure 6: Sample of recent news in MarketWatch (1)

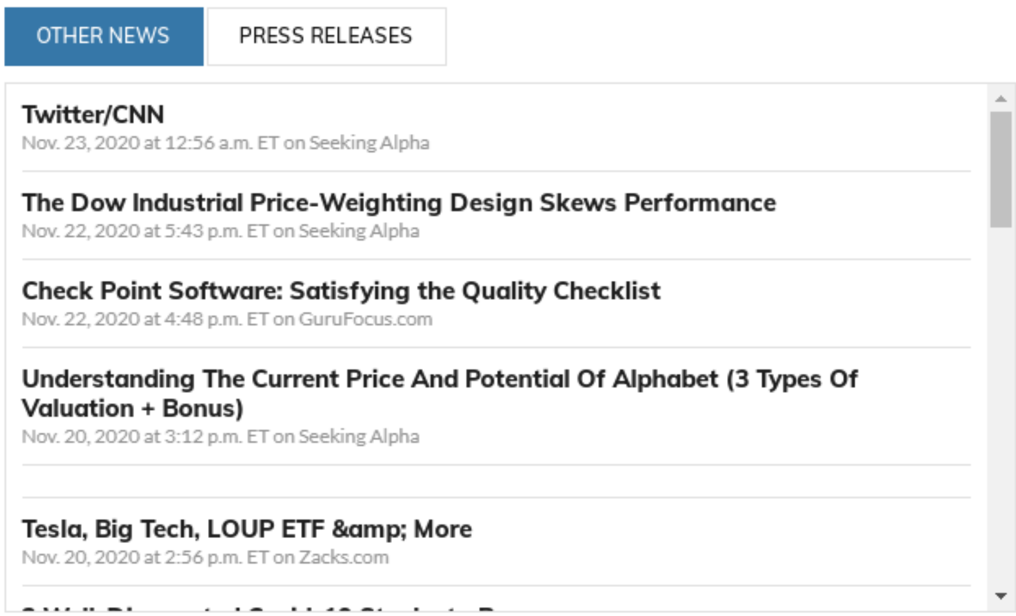


Figure 7: Sample of recent news in MarketWatch (2)

The last dataset we have was obtained from New York Time. They privide a API that allow everyone to download their news resource. However, it may have security and server stress considerations, we have limit on requesting resources. We cannot issue more than 10 request per minute and 4000 requests per day. Therefore, we need to write a program to autometically crawler articles with New York Time API. At the end, we get all the aritcles from January 1, 2010 to October 30, 2020.

FAQ

11. Is there an API call limit?
Yes, there are two rate limits per API: 4,000 requests per day and 10 requests per minute. You should sleep 6 seconds between calls to avoid hitting the per minute rate limit. If you need a higher rate limit, please contact us at code@nytimes.com.

Figure 8: API request limit of New York Time

Details of our dataset

- Apple (AAPL), S&P 500 Index(^GSPC), Nasdaq Composite Index(^IXIC)
  - Period: From January 1, 2010 to October 30, 2020
  - Features: high, low, open, close, adjusted close, volume (For each stock or index)
  - Size: 2729

- Sentiment Analysis for Financial News
  - Period: not provided
  - Features: News headline, sentiment classification (Ratio: Positive, Neutral, Negative is 59%, 28%, 12%)
  - Size: 4837

- All the news
  - Period: November 22, 2011 to June 21, 2017
  - Features: Title, Publication, Author, date, URL, content
  - Size: 143000

- News and Blog articles that mention “Apple”
  - Period: January 31, 2017 to April 1, 2017
  - Features: Title, date, URL

- Size: 106089

- News crawler form MarketWatch
  - Period: Starting from November 1, 2020
  - Features: Title, date, URL
  - Size: 379 (until November 23, 2020)

- News crawler from New York Time
  - Period: From January 1, 2010 to October 30, 2020
  - Feature: date, headline, abstract, leading paragraph, URL
  - Size: 29084

3.2 Preprocessing

Labeling

For stock data, we add a new column labeling whether AAPL will increase or decrease on the trading day. The way we define whether the stock will increase or decrease is the following:

Increase: If close of the next trading date > close of today

Decrease: If close of the next trading date <= close of today

Normalizing

After labeling for the dataset, we then normalize the dataset in order to reduce the training cost [13].

$$X_{normalized} = \frac{X - mean(X)}{man(X) - \min(X)}, \quad \text{Eq. 1}$$

where X is vector of value of same feature, the mean, max, and min function is to find out the mean value of vector X and the largest and smallest value among X respectively. Note that man(X) – min(X) might be replaced by std(X) which is the standard deviation of X, in the coming experiment.

Cross dataset concatenation

As we used stock data of Apple, Standard & Poor's 500 index and

Digitization

For news datasets, if there are any non-numerical values in the dataset like the word "positive", "neutral", and "negative" in "Sentiment Analysis for Financial News" dataset. We need to replace them by 1, 0, -1 respectively.

Information Extraction

The data we crawled is in html format, therefore, there will be some labels like <html>, <div> etc., or in json format, which means there will be some keys and values pairs in there. We need to extract the information from these structures.

Removing Punctuation

In order to make the training set clearer, less noise in the data set, we remove punctuation from the headline, for example semicolon, double quotes etc.

3.3 Literature Review

3.3.1 Sentiment Analysis

In short, sentiment analysis is the process of determining whether a piece of writing is positive, negative or neutral [14]. A sentiment analysis system (also know as opinion mining or emotion AI) uses natural language processing (NLP) and machine learning to identifies, extracts and classify subjective information from source materials [15]. It usually assign weighted sentiment scores to a word, sentence, paragraph or the whole document [16].

There are 3 main types of sentiment analysis:

1. Rule based: system performs sentiment analysis based on a set of manually crafted rules [16].
2. Automatic: system learns the data by itself using machine learning techniques [16].
3. Hybrid: system combines rule-based and automatic approaches [16].

Rule-based approach is set of rules that manually crafted by people, and using it to calculate the sentiment score of the target. These rules may include various techniques developed in computational linguistics, such as stemming, tokenization, part-of-speech tagging, parsing and lexicons [16]. For example, given a sentence to the system. It first identifies the pre-defined polarized words (e.g. negative words such as bad, worst, etc. Positive words such as good, best, etc.). Then it counts the number of occurrences of positive and negative words. The side that has greater number determines the polarity of the sentence, or neutral if they are even.

Automatic approach taking the advantage of machine learning, it does not rely on manually crafted rules, but on varies machines learning techniques/algorithms to achieve sentiment analysis. A machine learning algorithm usually involves in two parts of process, training and prediction. In the training process, a model is generate by associating an input(text) to the corresponding output(tag) based on the training samples. In the prediction process, unseen input is extract by feature extractor to extract the features. These features are then fed into the trained model to generate prediction tags, in this case they are positive, negative and neutral. There are lot of machine learning algorithms that suitable for sentiment analysis, like Long-Short Term Memory, BERT, etc.

Hybrid approach combine the desirable elements of rule-based and automatic techniques into one system. One huge benefit of these systems is that results are often more accurate [16].

3.4 Model Description

3.3.1 Long-Short Term Memory (LSTM)

It takes a long time to learn how to store information in extended time intervals through recurrent backpropagation, which is mainly caused by insufficient attenuation and false backflow, as known as the vanishing gradient problem. The proposal of LSTM is to solve this problem. LSTM is an improved version of recurrent neural network (RNN). By truncating the gradient, LSTM learn from data with very long time lags.

The structure of RNN is a chain. Each layer of RNN units not only takes input from the previous layer, but within the RNN units of the same layer, each unit uses the output of the previous unit as its input. Within each RNN unit, there is a state which recording a value, like a memory cell in the unit. The value inside will be serve as the output of the unit for the next unit of the same later or for the next layer of neural network [17].

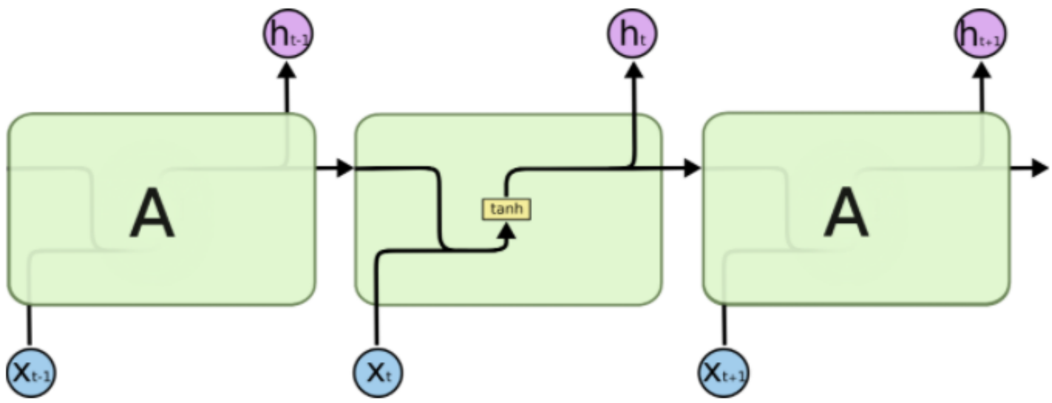


Figure 9: RNN layer structure

LSTM as the improved version of RNN, its structure is also in chain shape. But it is more complicated inside each unit. There are 3 extra gates in each unit, forget gate, input gate, and output gate. The forget gate decides whether delete the memorized value in the cell state or not. Input date decides whether the new coming input should be added to the cell state or not. Finally, the output gate decides whether this LSTM unit should output the value of the cell state [17].

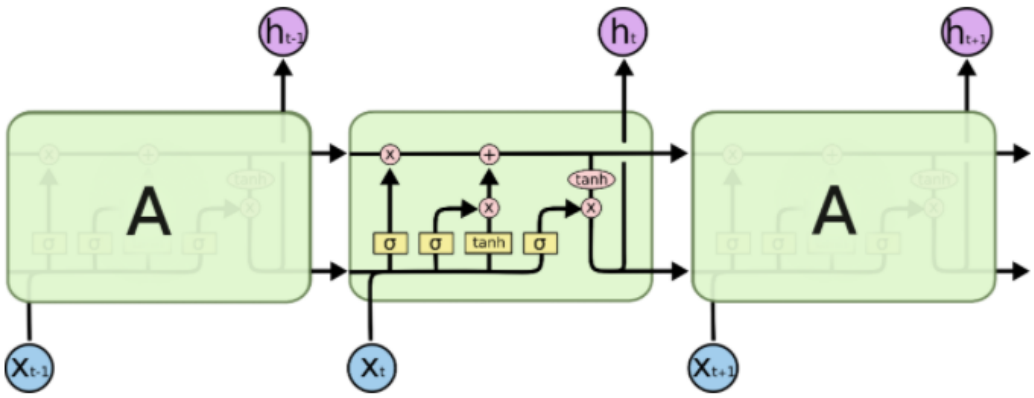


Figure 10: LSTM layer structure

However, LSTM is not a completely perfect model, nothing is. There are many variant of LSTM was proposal later to solve the drawback of LSTM, for example the training time required.

3.3.2 Gated Recurrent Unit (GRU)

The forget gate of LSTM is very important, because when the LSTM is processing related continuous inputs, the memory state value in the LSTM unit will always grow, and grow indefinitely, without an upper limit, and eventually cause the neural network to collapse. The solution to this problem is to add the forget gate, so that the LSTM order can automatically release the internal memory state value at an appropriate time. LSTM with forget gate can elegantly solve the problem of infinite growth of memory state value [18].

But LSTM still has an important problem, it requires a very long training time. Later in 2014, Cho et al. proposed Gated Recurrent Units (GRU). It is like LSTM, but it does not have the forget the gate, so it reduces a lot of parameters which making GRU faster to train when compared to LSTM. Besides, the performance of parameter updates and generalization of GRU can outperform LSTM units. [19]

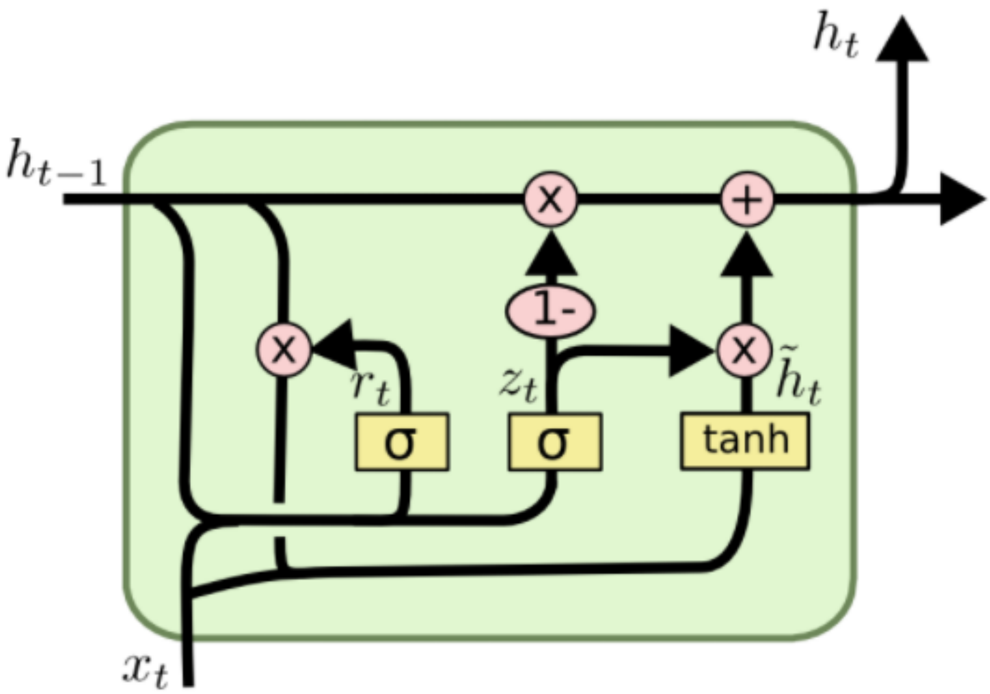
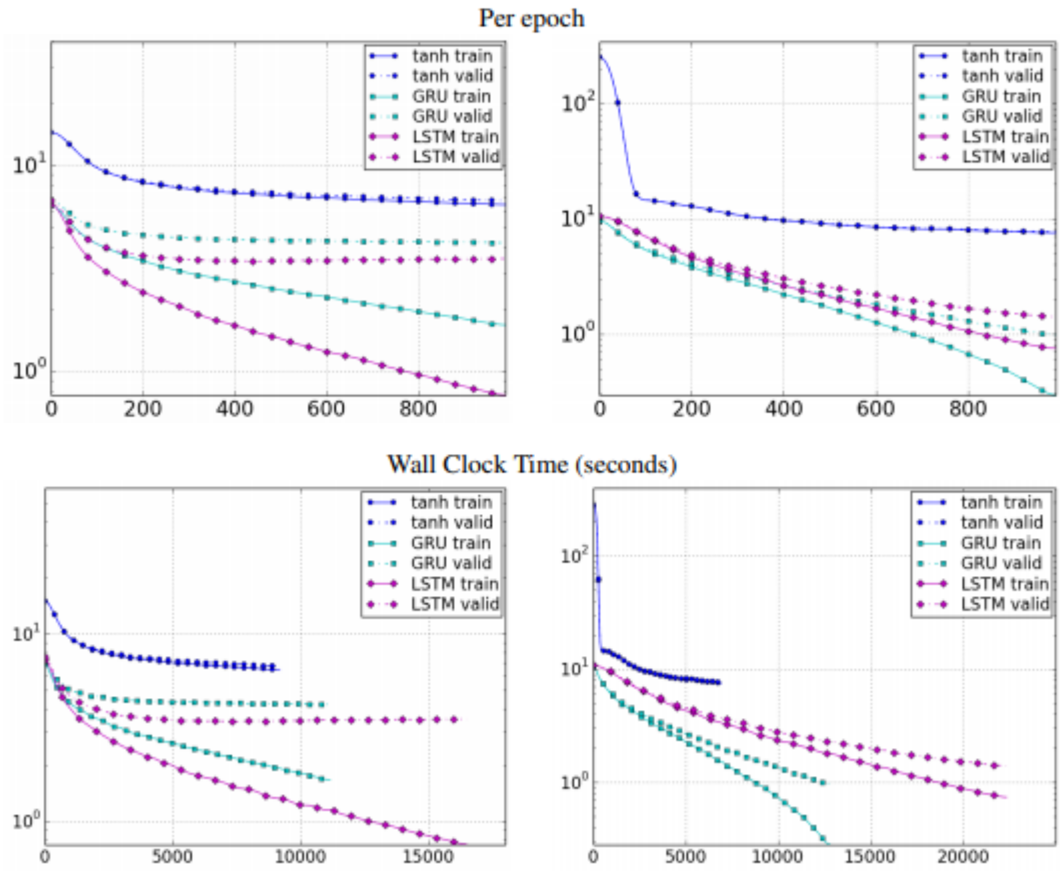


Figure 11: Structure of a GRU unit

The difference between GRU and LSTM is that GRU lacks the forget gate and input gate in LSTM, and instead is replaced by an update gate. The reset and input functions are completed through a single gate. Which makes GRU faster to finish the training process. [17]

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 12: Equations of gates



(a) Ubisoft Dataset A (b) Ubisoft Dataset B

Figure 3: Learning curves for training and validation sets of different types of units with respect to (top) the number of iterations and (bottom) the wall clock time. x-axis is the number of epochs and y-axis corresponds to the negative-log likelihood of the model shown in log-scale.

Figure 13: Experiment result of comparing RNN, GRU, and LSTM

It can be found that both GRU and LSTM perform better than traditional RNN (tanh, blue line graph) in terms of final results or convergence speed. However, although LSTM and GRU have their own advantages and disadvantages in different data and tasks, there is no big difference. In practice, whether to use LSTM or GRU depends on the situation. [19]

3.3.3 K-Nearest Neighbors (KNN)

K-nearest neighbors algorithm can be used in classification problems or regression problems.

For classification, KNN algorithm is to classify any unclassified sample point into the main classification around it. The specific method is to observe the nearest K classified sample points of the sample point, called K nearest neighbors, and then classify the unclassified sample points into the dominant category among those K nearest neighbors [20]. There are many variants of the classification process, such as classifying the unclassified sample points with the distance of its K nearest neighbors weighted, or simply classifying it into the class that occurs the most in the K nearest neighbors.

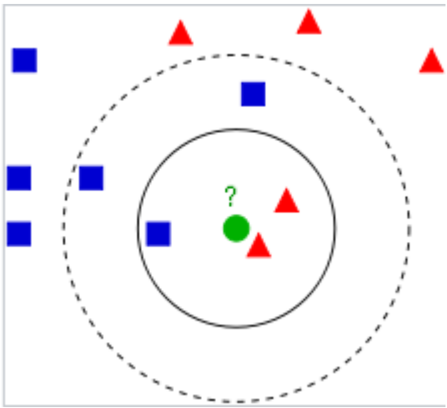


Figure 14: Geometrical meaning of KNN

KNN regression is similar to classification, but it gives a prediction of a continuous value instead of a class. For a simple point, again, the algorithm first finds out its k nearest neighbors. Then KNN algorithm can simply find the mean value of the k neighbor which will be the prediction of the sample point. In addition, we can weight those neighbors with respect to their distance to the sample point, for example, the closer neighbors will be scaled more compared to those farther neighbors, then the prediction of the sample point will be the mean of those weighted values.

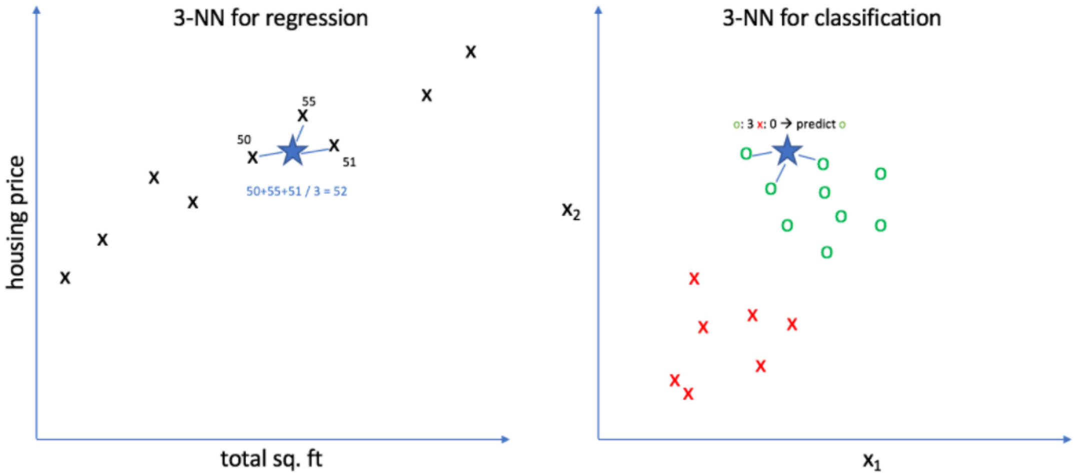


Figure 15: An example of KNN regression (https://www.jeremyjordan.me/k-nearest-neighbors/)

3.3.4 Prophet

Facebook Prophet is a new time series forecasting model proposed in 2017 released by Core Data Science Team of Facebook.

Prophet perform time-series data prediction with an addition model. It has good performance on the time-series data which have strong periodical effect and several seasons of the historical data. Data missing and any shifts in trend will not affect the performance of the model. There are three main parts in the Prophet model, trend, seasonality, and holidays [21].

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \qquad \text{Eq. 2}$$

For the trend function, g(t), it models the non-periodic trend of the time-series data set. s(t) represent the periodic change in the time-series data set, for example the seasonal changes of air-conditioner company, s(t) can also represent some weekly or daily changes, with in terms of day of week or hour. h(t) represent the holiday effect, as data of the holiday may look different with other day, which makes the data not following the non-periodical change function or periodical change function. Therefore, this model allows users to specify those special dates in addition to the built-in statutory holiday [21].

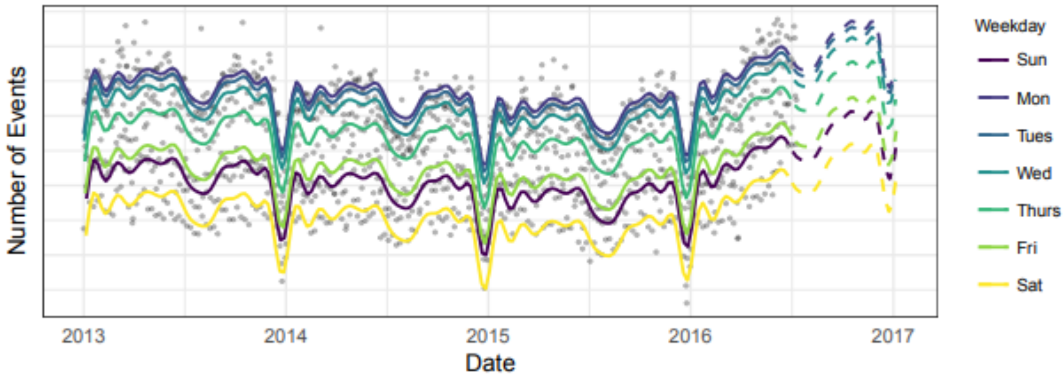


Figure 16: Example of periodic change

3.3.5 Clustering

Clustering a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labelled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples [22]. In short, clustering is a task that grouping data points in such a way that the data points in same group should have similar properties or features, while data points in different groups should have no similarity between [23].

There are many clustering algorithms, each of them has its own design purpose and advantage. We will take a look on two popular clustering algorithms, K-Means Clustering and Agglomerative Hierarchical Clustering.

K-Means Clustering is the simplest clustering algorithm that it works by partition n data points into k clusters where each data points belongs to the cluster having the nearest mean serving as a prototype of the cluster. It is fast and easy to implement, but notice that choosing an optimal k value is also challenging.

Agglomerative Hierarchical Clustering, or Hierarchical Clustering, is a method to analysis hierarchical data to build a hierarchical cluster. An Agglomerative Clustering means it is a bottom-up algorithm. It builds the tree by recursively merges the similar data points and similar clusters until all of the clusters/data points becoming one group of root cluster. Agglomerative Hierarchical Clustering is good at finding small clusters, in which terms in some small datasets, Agglomerative Hierarchical Clustering might perform better than K-Means Clustering.

3.3.6 VADER Sentiment

VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains [24]. VADER is able to determine the polarity of sentiment of a given text when the data being analysis is not labelled. Moreover, VADAR not only tells about the polarity (positive/negative) of the text, but also identifies the intensity (strength) of the expressed emotion. For example, words like ‘love’, ‘enjoy’, ‘happy’, ‘like’ all convey a positive sentiment. Also VADER is intelligent enough to understand the basic context of these words, such as “did not love” as a negative statement. It also understands the emphasis of capitalization and punctuation, such as “ENJOY” [25]. Below is an example of how lexicon is structured, with each word having a valence rating:

Word	Sentiment rating
Tragedy	-3.4
Rejoiced	2
Insane	-1.7
Disaster	-3.1
Great	3.1

Figure 17 lexicon structure [26]

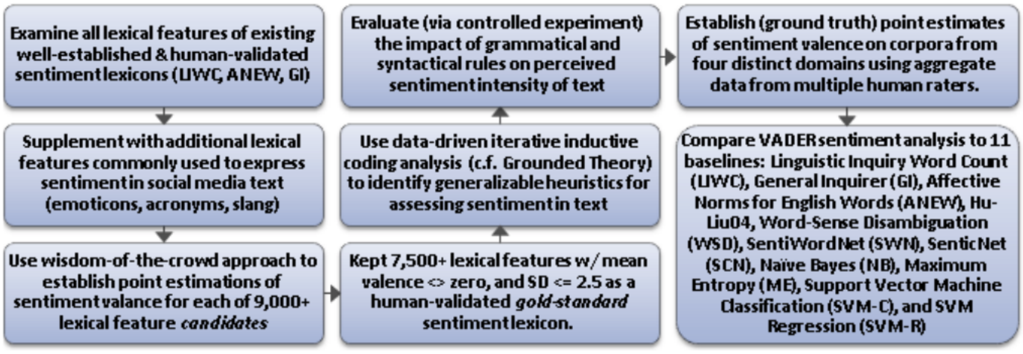


Figure 18 methods and process approach overview [27]

VADAR is the easiest sentiment analysis model current available since it does not require a number of preprocessing to work. In some supervised methods of NLP, preprocessing work must have done in order to move to learning stage. These preprocesses such as tokenization, stemming, lemmatization are no need for VADAR. Furthermore, VADAR is smart enough to understand the valence of non-conventional text. For example, emojis/emoticons like "😊", ":)" generally refers to positive sentiment, acronyms like "FML", "WTF" generally refers to negative sentiment. Taking this further, slangs like "Nah", "meh", words that are capitalized like "sad" vs "SAD", excessive punctuation like "?" vs "?????", they are all distinguishable and can measure different sentiment scores. With the ability to automatically remove stop words, no preprocess needed and tons of smart detection on the text, VADAR is the most newbie-friendly sentiment analysis model that does not sacrifice speed and accuracy.

3.3.7 TextBlob

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more [28]. By using its sentiment analysis API, it will give two results on the input data, polarity and subjectivity. Polarity is a floating point number that ranges from -1 to 1 where -1 means negative sentiment and 1 means positive sentiment. Subjectivity is also a floating point number that ranges from 0 to 1 where 0 means objective and 1 means subjective.

Similar to VADAR Sentiment, TextBlob is also a “Plug-and-Play” sentiment analysis library that does not require any tweaking. Therefore it is very easy to use and suitable for beginners who are leaning natural language processing for the first time.

3.3.8 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers was released by Google in 2018. It was well known it has a good performance and breaks the record in solving NLP problems.

Before BERT was released, there are already exist some pre-trained model, for example Embeddings from Language Models (ELMO) and OpenAI GPT. However, there is a problem with the these pre-training. When pre-training, only the one-way order of the text is considered. Whether it is from left to right or from right to left, it is still not a good solution. Want to learn this vocabulary at the same time Contextual information problem. Although ELMO is although a bidirectional model, it is separated in two executions, first time it reads form left to right, and then it reads from right to left. At the end, it put two results together to do the final prediction [29].

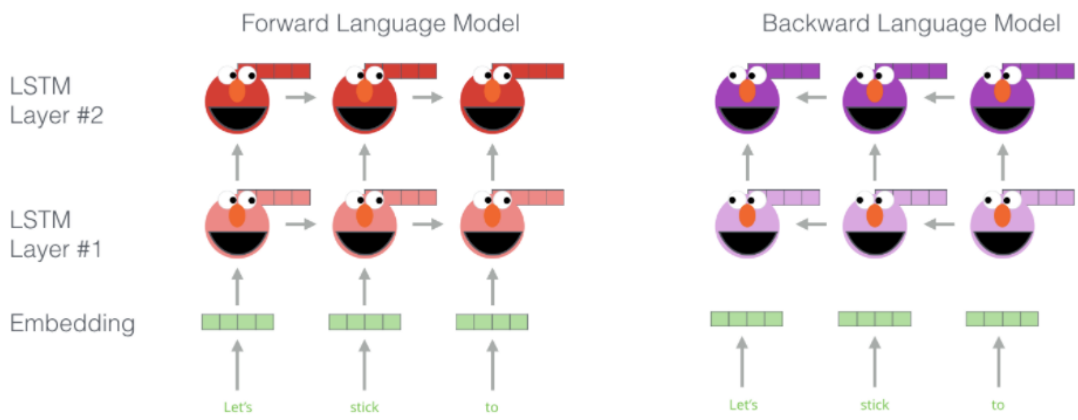


Figure 19: Graph explanation of "Bi-direction" in ELMO http://jalammar.github.io/illustrated-bert/

The difference from the language model proposed in recent years is that BERT no longer only focuses on the information before or after a word, but all layers of the entire model pay attention to the context information of its entire context. The experimental results prove that using the pre-trained BERT model, just wrap a layer of output layer behind and fine-tune it for training, you can get very good results, and even the accuracy of several tasks has exceeded that of humans [30].

From the figure below, we can see the different in these three models. It clearly shows the different between BERT and ELMO, although they can both process the sentence in two directions.

The way ELMO do is dividing them into two components, the left component read the sentence from left to right, and the right component read the sentence from right to left, then it combines two component and output its answer. Even it analyzes the sentence from two direction, each unit of the network has no idea of the unprocessed words. In the other words, in the left component of the model, it read some of the words from the left of the sentence, the information of the rest words of the sentence are not used to analysis, similar for the right component, until the model combine these two components.

The way BERT do is that taking every single word in the sentence to each unit. Each unit has a clear vision of the whole sentence and then analyze the word with the whole sentence.

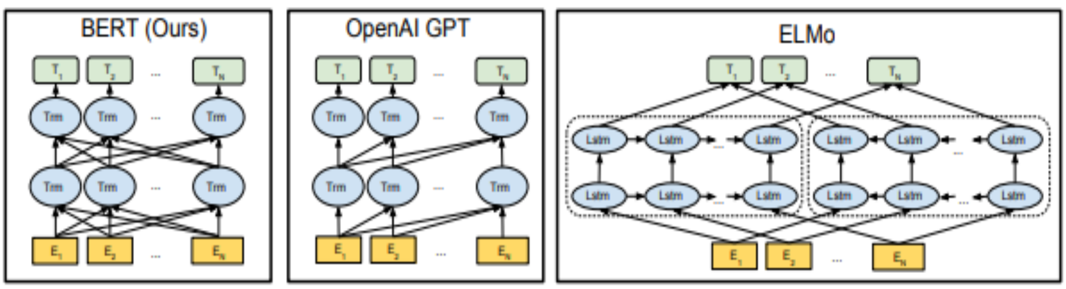


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

Figure 20: Model comparison

3.3.9 Artificial Neural Network (ANN)

Artificial neural network is a model of biological neural network, which is inspired by the biological neural networks that constitute animal brains. In ANN, perceptron or units are group as layer. There are many connections between the units and each connection is associated with a weight. If the connections between the units do not form a cycle, then it is a feedforward neural network. Feedforward neural network is the first and simplest ANN, each layer's output will server as the input of the next layer, so that the information only flow to the output direction. There can be an activation function on the output layer for the final prediction. In network training, the performance of the neural network is measured by a loss function and then update the weights of unit connections through a mechanism called back propagation. [31]

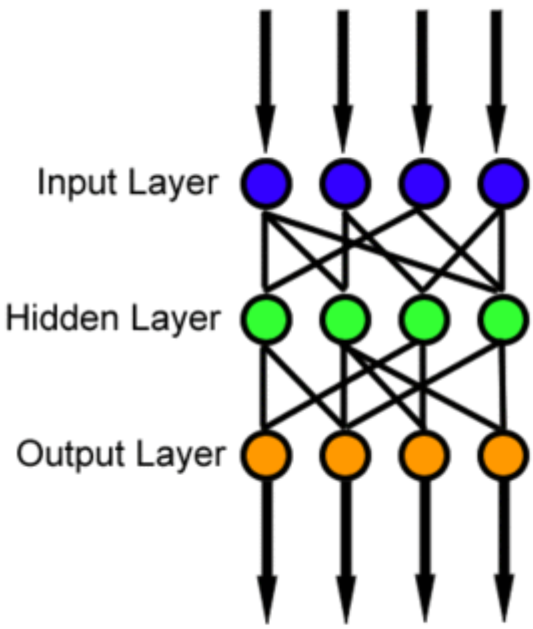


Figure 21: An example of ANN

4. Experiments

4.1 Experiment on Data Visualization

In the early days, I learned that investors generally pay attention to the candlesticks of stocks. As introduced in the background section, the candlestick pattern will imply the future trend of the stock market. In other words, the candlestick pattern is the different combination of lengths of upper shadow line, lower shadow line and candle body. Therefore, we guess these three lengths maybe able to help us to see some pattern. What we did is plotting a three-dimensional graph with upper shadow length, lower shadow length and the length of body as the three axes. Then we further mark those point as green or read, indicating that the stock will rise on the next trading day with the combination of three axes.

Visualization result:

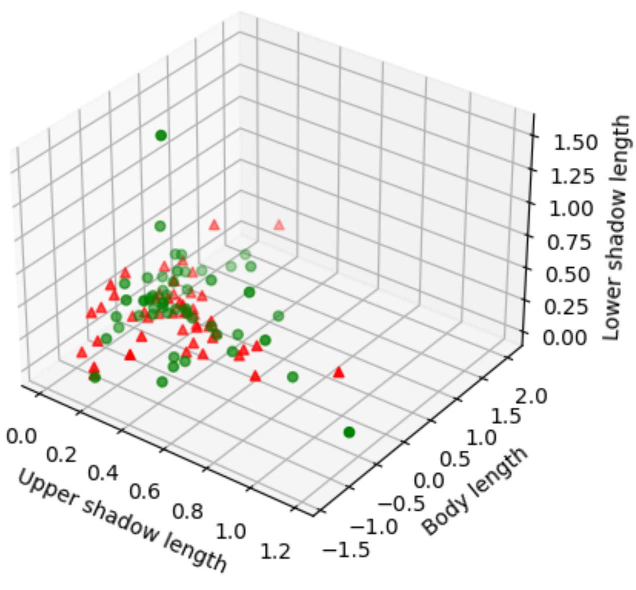


Figure 22: 3D graph of 3 lengths of candlestick

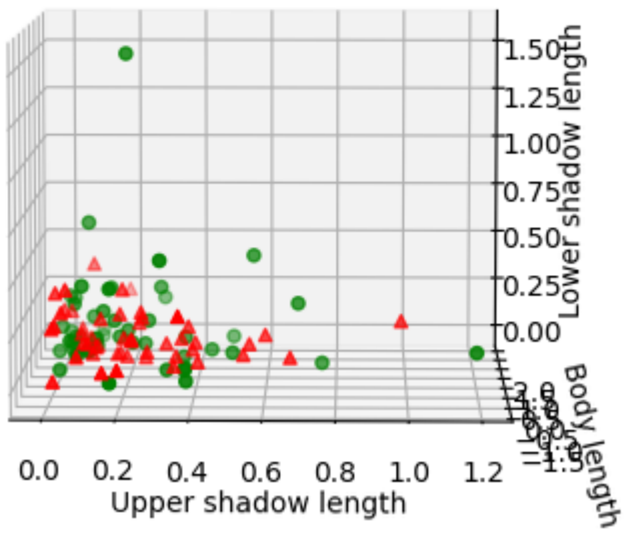


Figure 23: Focusing on upper and lower shadow length

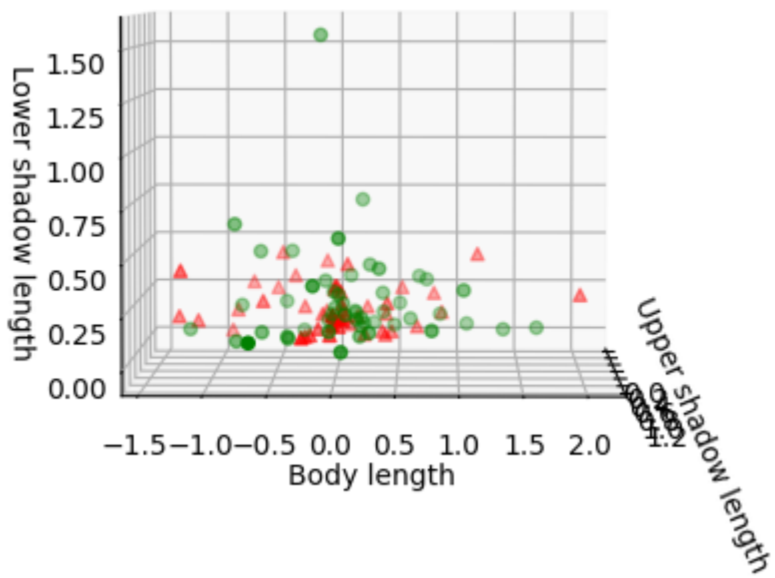


Figure 24: Focusing on lower shadow and body length

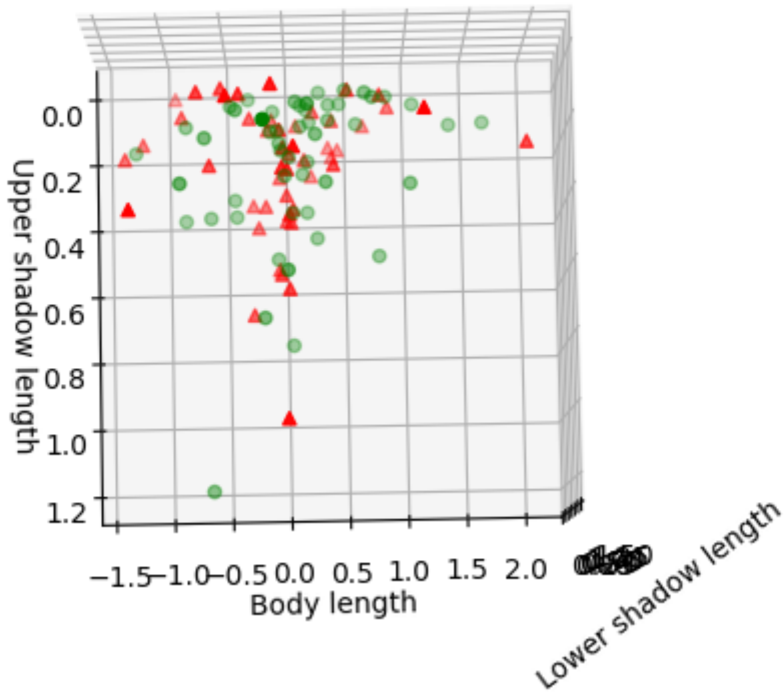


Figure 25: Focusing on upper shadow and body length

The above 3D graph doesn't show us any sign of grouping. We cannot see a cluster of green dots or a cluster of red triangles. They are all concentrated together. Then we test if a two-dimensional graph can help us. We defined a term, net shadow length, which is derived from the upper shadowing length subtracting the lower shadow length. The result can be a negative value which means that the lower shadow length is longer than the upper shadow length.

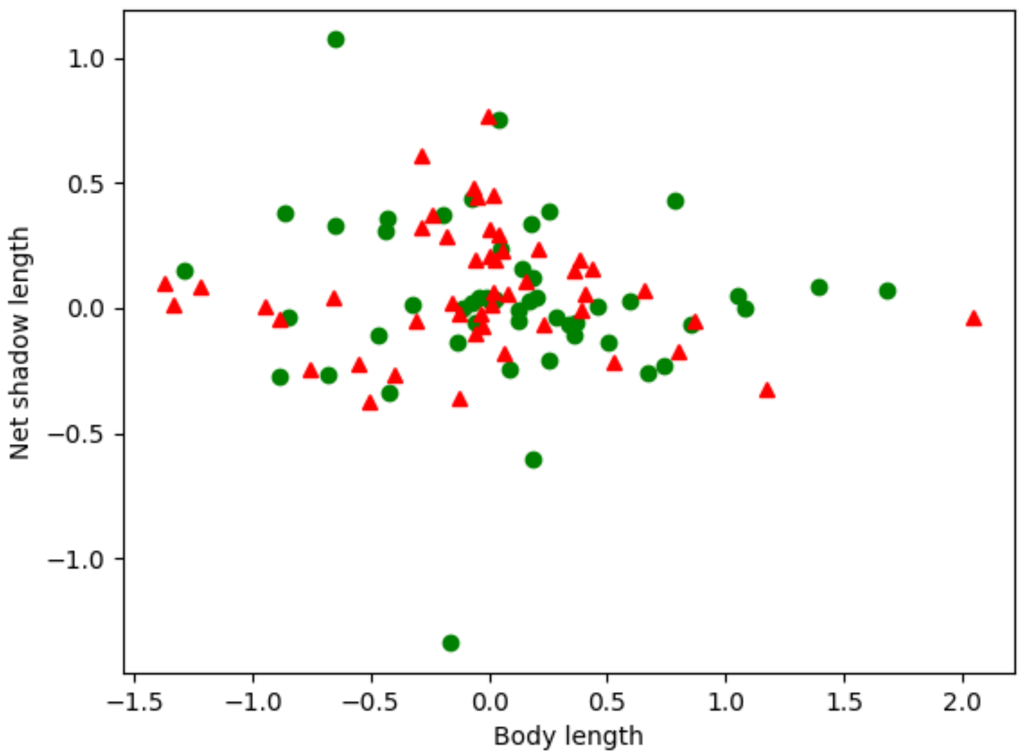


Figure 26: 2D graph, Net shadow lenth vs body length

The result is also showing us there is not two clear group that contains only green dots or red triangles. They all concentrated to together.

Then, our final test was plotting three-dimensional graph with open, close, and the final axis was picked from upper, lower, or net shadow length. However, these three graphs also did not show any sign of grouping.

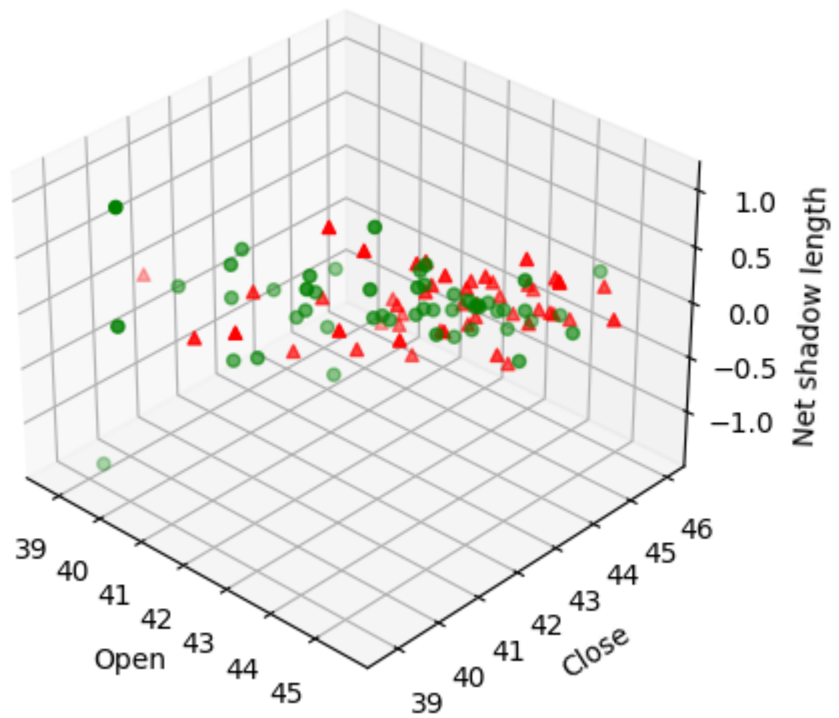


Figure 27: 3D graph of open, close, and net shadow length

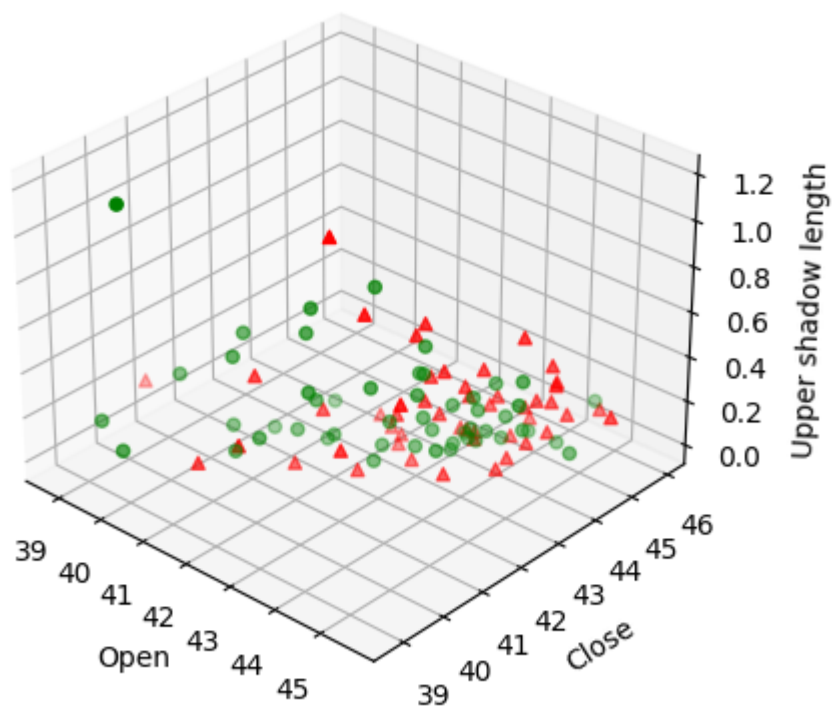


Figure 28: 3D graph of open, close, and upper shadow length

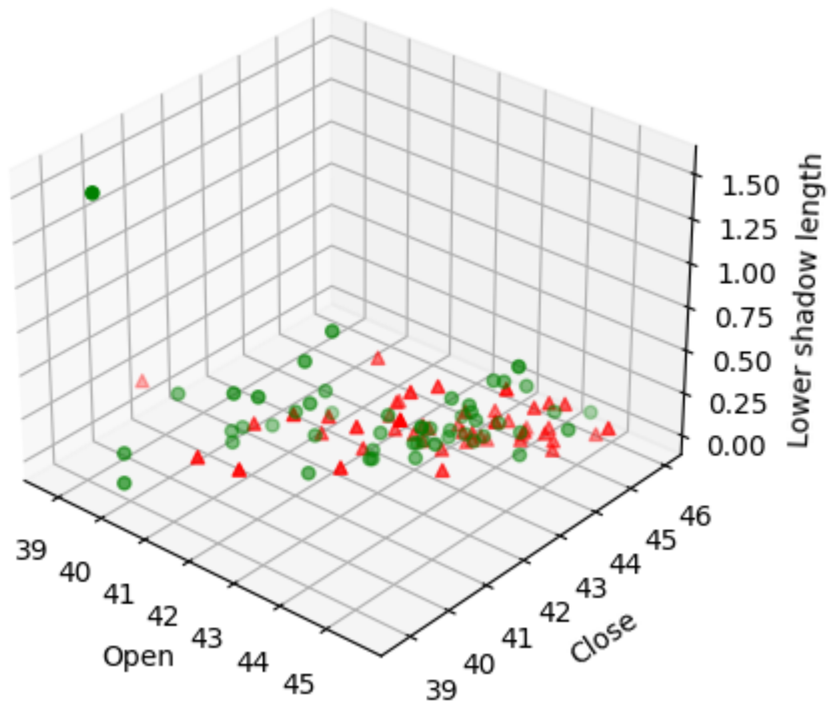


Figure 29: 3D graph of open, close, and lower shadow length

4.2 Experiments on Numerical Analysis

4.2.1 LSTM

Our a first model built is LSTM, using 22 days of data to predict the closing price of the closing price of the 23-th trading day. Input features used include Year, Month, Day, High, Low, Open, Close, Adjusted close, and Volume. The size of the sliding window is 22 days. For the model architecture, it has 1 input layer with 22x9 = 198 units, 2 LSTM layers each has 128 units, 1 dense layer with 32 unit and using relu as its activation function, and 1 output layer with 1 unit and the activation is linear. We use mean-square error (MSE) or root mean squared error (RMSE) to evaluate the model.

	Year	Month	Day	High	Low	Open	Close	Volume
0	-0.5495628	0.4960977	0.50927398	-0.303476	-0.3032738	-0.3025696	-0.3034496	0.02872037
1	-0.4495628	-0.5039023	-0.390726	-0.3028583	-0.3022774	-0.302407	-0.3016752	0.10571026
2	-0.4495628	-0.5039023	-0.3573927	-0.3022728	-0.3018011	-0.3017731	-0.3014751	0.16464366
3	-0.4495628	-0.5039023	-0.3240594	-0.3024662	-0.3031698	-0.3018923	-0.3033197	0.1375429
4	-0.4495628	-0.5039023	-0.290726	-0.3042011	-0.3041005	-0.3033174	-0.3035307	0.09666752
5	-0.4495628	-0.5039023	-0.2573927	-0.3042011	-0.304095	-0.3041031	-0.3027734	0.08058493
6	-0.4495628	-0.5039023	-0.1573927	-0.303664	-0.304429	-0.3027484	-0.303785	0.08854919
7	-0.4495628	-0.5039023	-0.1240594	-0.3053989	-0.3055403	-0.3047046	-0.3050778	0.16058755
8	-0.4495628	-0.5039023	-0.090726	-0.3047758	-0.3068104	-0.3054198	-0.3034928	0.16681587
9	-0.4495628	-0.5039023	-0.0573927	-0.3050282	-0.3041169	-0.304206	-0.3041528	0.07256728
10	-0.4495628	-0.5039023	-0.0240594	-0.3044159	-0.3058414	-0.3037617	-0.3060461	0.16037399
11	-0.4495628	-0.5039023	0.10927398	-0.3024877	-0.3050914	-0.3051706	-0.3011181	0.23443353
12	-0.4495628	-0.5039023	0.14260731	-0.3022943	-0.3038541	-0.3016051	-0.3029086	0.17022673
13	-0.4495628	-0.5039023	0.17594065	-0.3034975	-0.3051078	-0.3031386	-0.3048885	0.16804842
14	-0.4495628	-0.5039023	0.20927398	-0.3066181	-0.3106099	-0.3060105	-0.3104711	0.31711175
15	-0.4495628	-0.5039023	0.30927398	-0.308122	-0.308951	-0.3083242	-0.3075933	0.41731714

Figure 30: Sample of input (normalized)

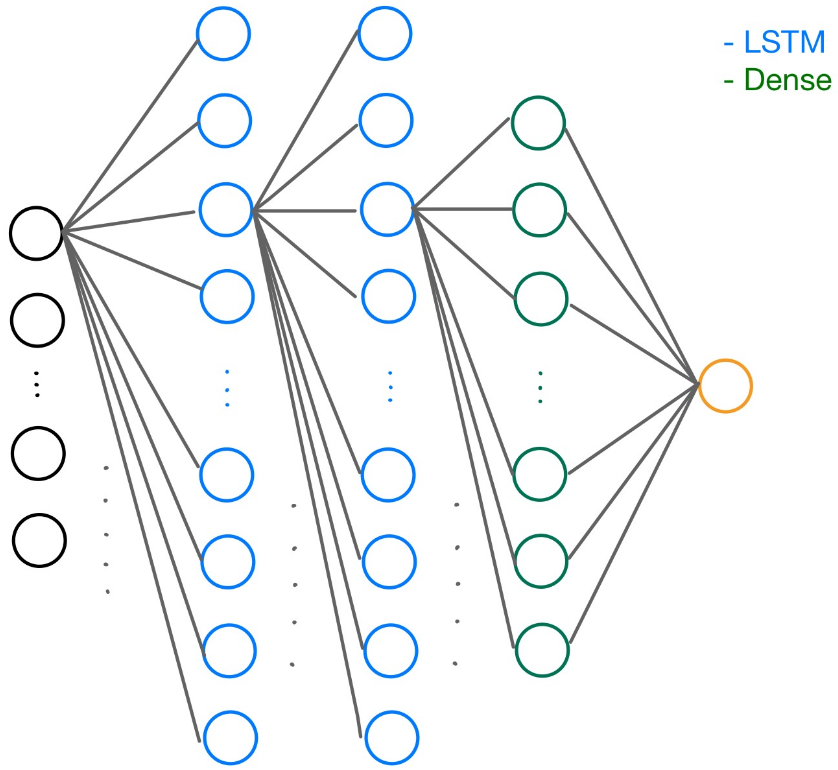


Figure 31: Architecture of the network

The size of the training set is 2000 day, about 8 years. The size of the testing set is 300, about 1 year.

Result of this experiment is 0.0013385 MSE on the training set, 0.00224169 MSE on the testing set. Then we de-normalize the result and find out that the MSE of the testing set is 9.771 while the RMSE is 3.126.

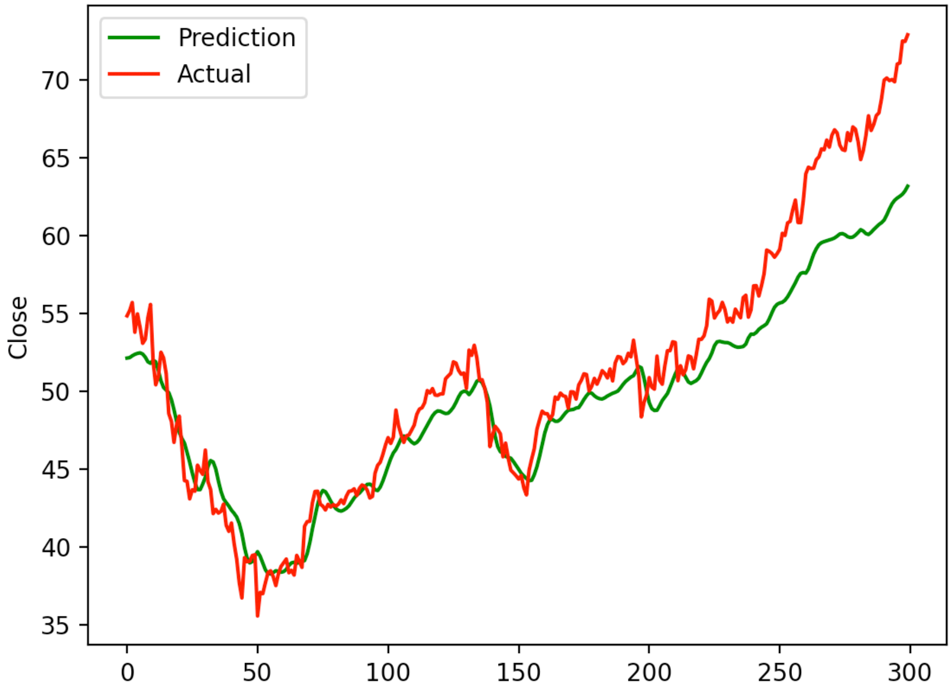


Figure 32: The prediction of the test set

It is wired that the prediction is getting far away from the ground truth. Then, after performing fine turning on the model. We obtained the best result, 1.088 MSE. Our final lstm model has 1 lstm layer less than the initial model and the window size is set to 10.

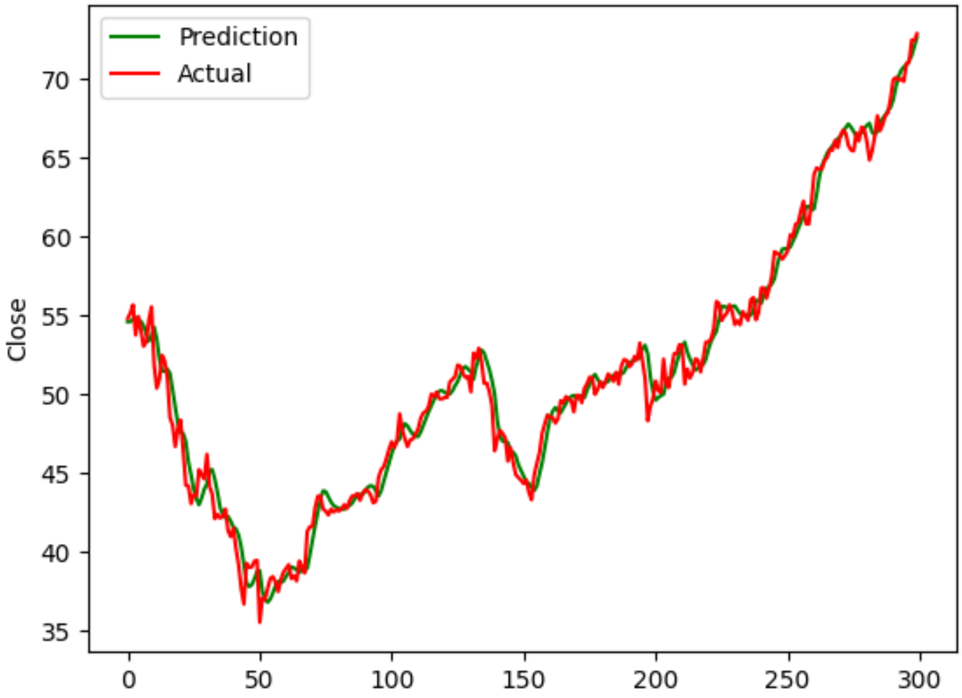


Figure 33: Prediction of the final model

However, in the observation throughout the experiment, the prediction seems to be the delayed version of the ground truth. As the shape of the prediction graph is similar to the shape ground truth but with a little shift to the right, which means that the model gives a prediction close to the current day to predict the close of the next training date. Therefore, we think this model doesn't give us a good prediction, in a decreasing trend, the model will give us a result that is too optimistic, while in an increasing trend, the model will give us a pessimistic result. Besides, the model will never successfully predict any turning point, it can only give the result with the previous trend.

4.2.2 GRU

The next model is gated recurrent units (GRU). The setting is similar to the LSTM model. The architecture of the model is one input layer, one GRU layer, and one dense layer. Sliding window size is 10. The size of the training set is 2000 day, about 8 years. The size of the testing set is 300, about 1 year.

The result of the GRU is slightly worser than the LSTM model. The result of this experiment is 0.00005175 MSE on the training set, 0.0002694 MSE on the testing set. The de-normalize MSE result of the testing set is 1.17354401.

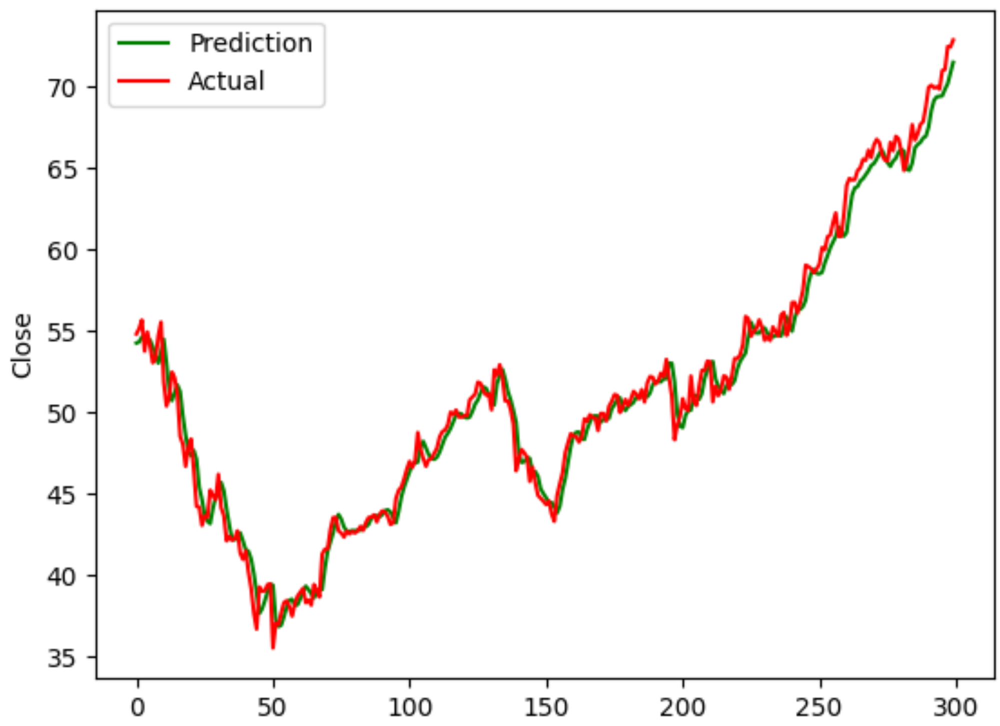


Figure 34: GRU model prediction

This model also suffer from the shifting problem is more obvious, which makes this model also is not a good prediction mode. It is still also giving a prediction based on the trend of the ground truth and there is not turning point is successfully predicted.

4.2.3 KNN Classification

First, we use KNN algorithm to build a classification model. The input features are High, Low, Open, Close, Adjusted Close, and Volume. The output of the model is a binary prediction, 1 means the stock will on the next trading day, while 0 means it will fall.

As this is a supervised learning algorithm, we need to label our training input and testing input. There is an extra column in the input set, will-rise. We will mark “will-rice” of date t to be 1 if the closing price of date t+1 is high than the closing price of date t.

	Date	High	Low	Open	Close	Volume	Adj Close	Will-rise
0	2009/12/31	7.619643	7.52	7.611786	7.526072	3.52E+08	6.503574	1
1	2010/1/4	7.660714	7.585	7.6225	7.643214	4.94E+08	6.604801	1
2	2010/1/5	7.699643	7.616071	7.664286	7.656428	6.02E+08	6.616219	0
3	2010/1/6	7.686786	7.526786	7.656428	7.534643	5.52E+08	6.51098	0
4	2010/1/7	7.571429	7.466072	7.5625	7.520714	4.77E+08	6.498945	1
5	2010/1/8	7.571429	7.466429	7.510714	7.570714	4.48E+08	6.54215	0
6	2010/1/11	7.607143	7.444643	7.6	7.503929	4.62E+08	6.484439	0
7	2010/1/12	7.491786	7.372143	7.471071	7.418571	5.94E+08	6.410679	1
8	2010/1/13	7.533214	7.289286	7.423929	7.523214	6.06E+08	6.501104	0

Figure 35: Sample of inputs of KNN classification model

The size of training set is 2256, size of testing set is 251, which is the number of trading date of 2019. In the experiment, we find that the k equal to n can give us the most accurate result.

```
K = 3
Accuracy: 0.529880
confusion matrix:
      |  Positive| Negative
 True|         49|       58
False|         60|       84
```

Figure 36: KNN classification, K = 3

```
K = 4
Accuracy: 0.498008
confusion matrix:
      |  Positive| Negative
 True|         69|       38
False|         88|       56
```

Figure 37: KNN classification, K = 4

```
K = 5
Accuracy: 0.565737
confusion matrix:
     |  Positive| Negative
 True|        53|       54
False|        55|       89
```

Figure 38: KNN classification, K = 5

```
K = 6
Accuracy: 0.517928
confusion matrix:
     |  Positive| Negative
 True|        68|       39
False|        82|       62
```

Figure 39: KNN classification, K = 6

```
K = 7
Accuracy: 0.529880
confusion matrix:
     |  Positive| Negative
 True|        45|       62
False|        56|       88
```

Figure 40: KNN classification, K = 7

```
K = 8
Accuracy: 0.470120
confusion matrix:
     |  Positive| Negative
 True|        52|       55
False|        78|       66
```

Figure 41: KNN classification, K = 8

K	3	4	5	6	7	8
Accuracy	0.530	0.498	0.566	0.518	0.530	0.470

Figure 42: Accuracy of different K values

K	3	4	5	6	7	8
Balanced Accuracy	0.521	0.518	0.557	0.534	0.516	0.473

Figure 43: Balanced accuracy of different K values

For either accuracy or balanced accuracy, K of value 5 have the best performance.

4.2.4 KNN Regression

We use KNN regression to predict the closing price of the next trading date. Again, the size of training set is 2256, size of testing set is 251, which is the number of trading date of 2019. The input features set is different from the input features set used in KNN classification. By removing the "will-rise" column and adding an extra column "Next close" which is the closing price of the next trading date.

	High	Low	Open	Close	Volume	Adj Close	Next Close
0	7.619643	7.520000	7.611786	7.526072	352410800.0	6.503574	7.643214
1	7.660714	7.585000	7.622500	7.643214	493729600.0	6.604801	7.656428
2	7.699643	7.616071	7.664286	7.656428	601904800.0	6.616219	7.534643
3	7.686786	7.526786	7.656428	7.534643	552160000.0	6.510980	7.520714
4	7.571429	7.466072	7.562500	7.520714	477131200.0	6.498945	7.570714
...	...	...	...	...	...	...	...
2510	70.662498	69.639999	70.557503	69.860001	275978000.0	69.381348	71.000000
2511	71.062500	70.092499	70.132500	71.000000	98572000.0	70.513535	71.067497
2512	71.222504	70.730003	71.172501	71.067497	48478800.0	70.580566	72.477501
2513	72.495003	71.175003	71.205002	72.477501	93121200.0	71.980911	72.449997
2514	73.492500	72.029999	72.779999	72.449997	146266000.0	71.953598	72.879997

Figure 44: Sample of training set

Initially we set K to be 10 to perform the prediction. On the first experiment, we observed a prediction curve become flatten when the testing day larger than 190. Seems that it cannot break a certain limitation.

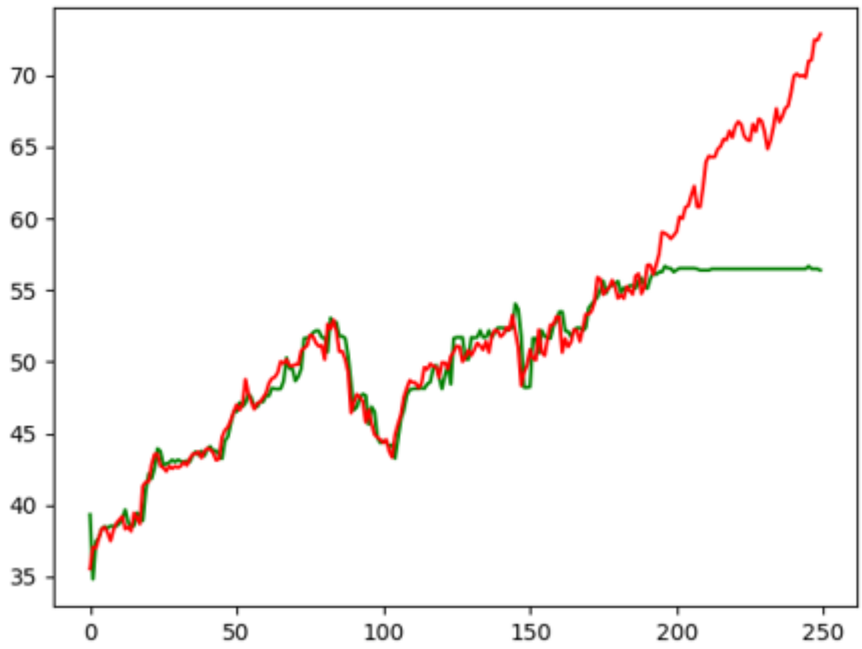


Figure 45: First prediction of KNN regression model

We found an explanation from the following graph, which is the stock record of AAPL from 2010 to 2020. We can see that there is an overall increasing. The green box indicates the training sample. In the training set, the highest close price is 56.9, which means that the prediction of the model can never be higher then 56.9.

Figure 46: The stock price record of AAPL from 2010 to 2020

Our solution is that we check the close price of the testing set, if the close price is close the highest closing price of the training set, then we shift the whole row with a certain value and then add it number back to predicted value.

```
limit <- maxClose(TrainSet)

for each day in TestSet
    shiftValue <- 0

    if day.close >= limit*0.8
        shiftValue <- day.close - limit*0.8
        day.open -= shiftValue
        day.close -= shiftValue
        day.high -= shiftValue
        day.low -= shiftValue
        day.adjClose -= shiftValue
        day.volumn -= shiftValue

    prediction = knnrPrediction(day)
    if day.close >= limit*0.8
        prediction += shiftValue

```

Figure 47: Pseudo code of shifting input feature

After modifying the code, now the model can handle the input set that is larger than the highest value in the training set.

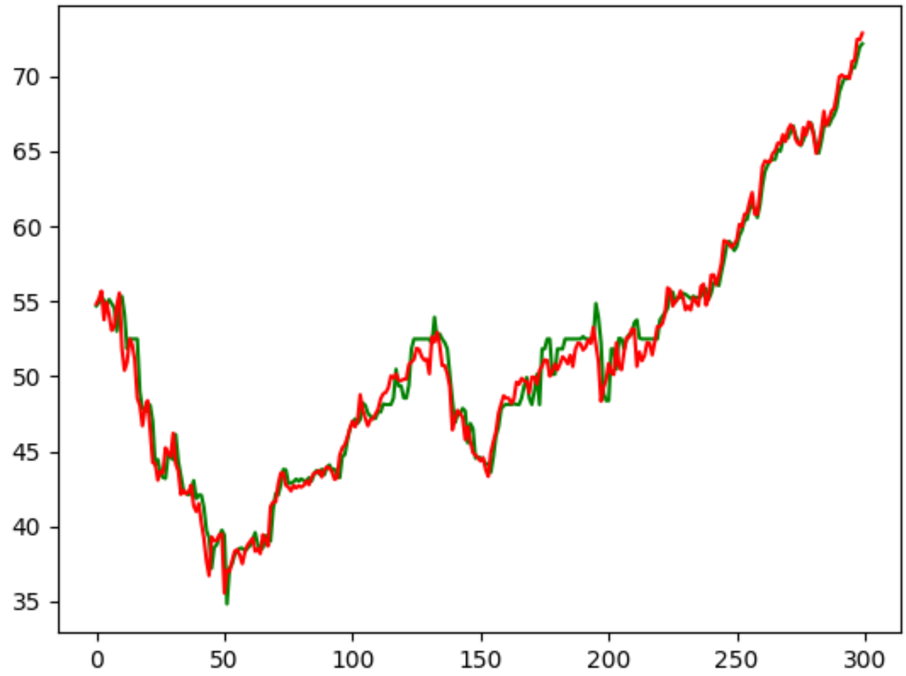


Figure 48: After shifting the input

After testing K with different value, we found out that K = 13 will give the optimal prediction.

K	7	10	12	13	14	15
MSE	1.191	1.180	1.163	1.162	1.186	1.169
RMSE	1.091	1.086	1.078	1.078	1.089	1.081

Figure 49: Experiment result

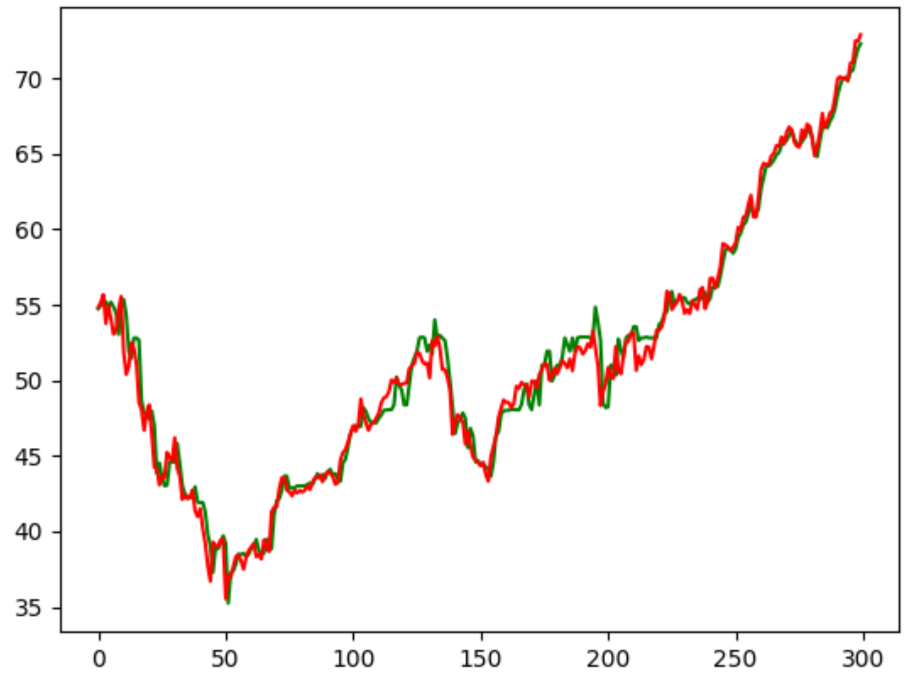


Figure 50: Prediction graph with k = 13

As a model to predict the closing price of the next trading date, KNN regression perform better than LSTM and GRU model. KNN regression is still cannot prediction the turning point and show some degree of delay, but KNN regression only show a delayed graph when there is an increasing or decreasing trend. The following two graphs show the delay when there is obvious trend. Green line is the prediction and the red line is the ground truth.

Figure 51: KNN regression show delay in negative trend

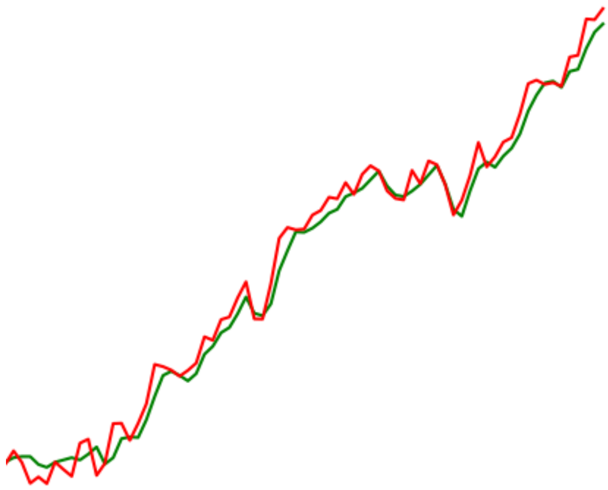


Figure 52: KNN regression show delay in positive trend

If the stock move steadily KNN regression can give us something different, we can see it gives us underestimation or overestimation. Circled in the following graph.

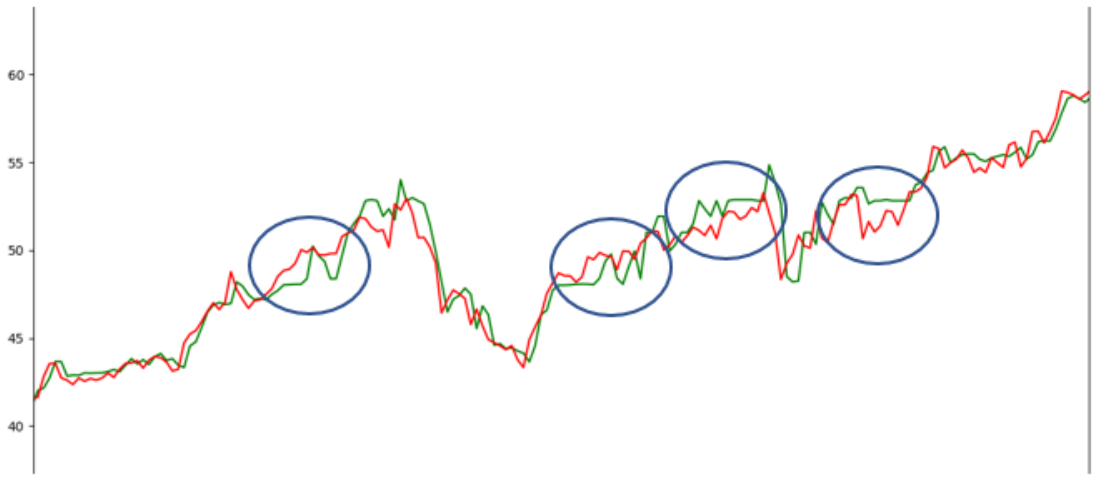


Figure 53: Performance of KNN when stock move steadily

We also experimented KNN regression used as a classifier. After getting the prediction of KNN regression model. In order to make a classification on whether the stock will rise on the next trading day, we need to compare the actual closing price of day t and the predicted closing price of day t + 1.

After getting the list of classification, we examine the accuracy by the following criteria:

$$Correct\ if\ \left(\widehat{Close}_{t+1} > Close_t\ and\ \ Close_{t+1} > Close_t\right)$$
$$or(\widehat{Close}_{t+1} \leq Close_t\ and\ \ Close_{t+1} \leq Close_t)$$

$$Wrong\ if\ \left(\widehat{Close}_{t+1} > Close_t\ and\ \ Close_{t+1} \leq Close_t\right)$$
$$or(\widehat{Close}_{t+1} \leq Close_t\ and\ Close_{t+1} > Close_t)$$

We will say the classification is correct if and only if both predict closing price of day t + 1 and actual closing price of day t + 1 are higher then the closing price of day t.

K	7	10	11	12	13	15
Accuracy	0.553	0.573	0.583	0.580	0.567	0.477

Figure 54: KNN regression as a classifier

4.2.5 Prophet

Because our analysis object is Apple, and it happens to have some special events at a few fixed time periods each year, such as spring and autumn conferences, and the Apple Worldwide Developers Conference event. We think these days will have an impact on Apple's stock price. Therefore, we used Facebook's Prophet module for analysis and prediction. There is a formula for holidays in the Prophet module. We treat Apple’s activities as a holiday and enter it into the module. The module analyzes how these days will affect Apple’s stock price.

We used 2010-2019 Apple stock data as the training data set to predict the stock trend in 2020. There are three main components in a Prophet model, $\mathrm{y(t)} = \mathrm{g(t)} + \mathrm{s(t)} + \mathrm{h(t)} + \epsilon_t$, these functions represent the non-periodic, periodic, and holiday changes. The following graphs shows trend learned by Prophet from the training set. In term of week, the third sub-graph shows that AAPL rise in Tuesday in general. In term of month, the fourth sub-graph shows that AAPL overall rise in September to November.

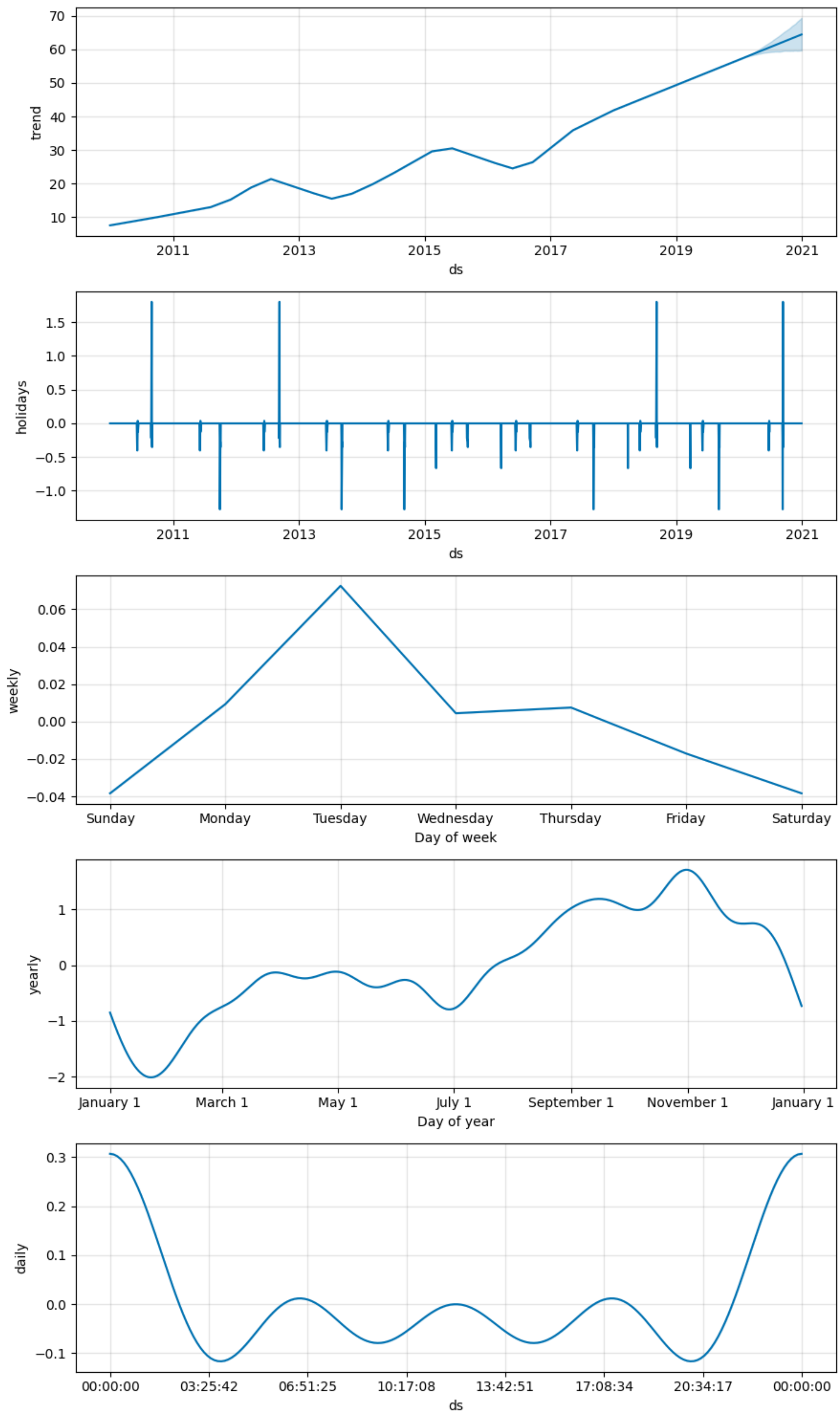


Figure 55: Graphes of three components of Prophet

From the graph bellow shows the effect of each apple event, all three apple event, spring (the Apple Spring Event column) and autumn conferences (the Apple Special Event), and the Apple Worldwide Developers Conference event (WWDC) have negative effect on the stock price in general.

ds	Apple Spring Event	WWDC	Apple Special Event
2015-03-09	-0.665117	0.0	0.0
2016-03-21	-0.665117	0.0	0.0
2018-03-27	-0.665117	0.0	0.0
2019-03-25	-0.665117	0.0	0.0

ds	Apple Spring Event	WWDC	Apple Special Event
2019-06-03	0.0	-0.402441	0.0
2019-06-04	0.0	-0.243307	0.0
2019-06-05	0.0	-0.059114	0.0
2019-06-06	0.0	0.037821	0.0
2019-06-07	0.0	-0.125540	0.0
2020-06-22	0.0	-0.402441	0.0
2020-06-23	0.0	-0.243307	0.0
2020-06-24	0.0	-0.059114	0.0
2020-06-25	0.0	0.037821	0.0
2020-06-26	0.0	-0.125540	0.0

ds	Apple Spring Event	WWDC	Apple Special Event
2018-09-12	0.0	0.0	-0.352346
2019-09-05	0.0	0.0	-0.217935
2019-09-06	0.0	0.0	-1.275926
2019-09-09	0.0	0.0	-0.273382
2019-09-10	0.0	0.0	-0.352346
2020-09-10	0.0	0.0	-0.217935
2020-09-11	0.0	0.0	-1.275926
2020-09-13	0.0	0.0	1.803734
2020-09-14	0.0	0.0	-0.273382
2020-09-15	0.0	0.0	-0.352346

Figure 56: Effect of the events

Then we compare the prediction of stock trend in 2020 made by Prophen to the ground truth of 2020.

Figure 57: Prediction made by prophet

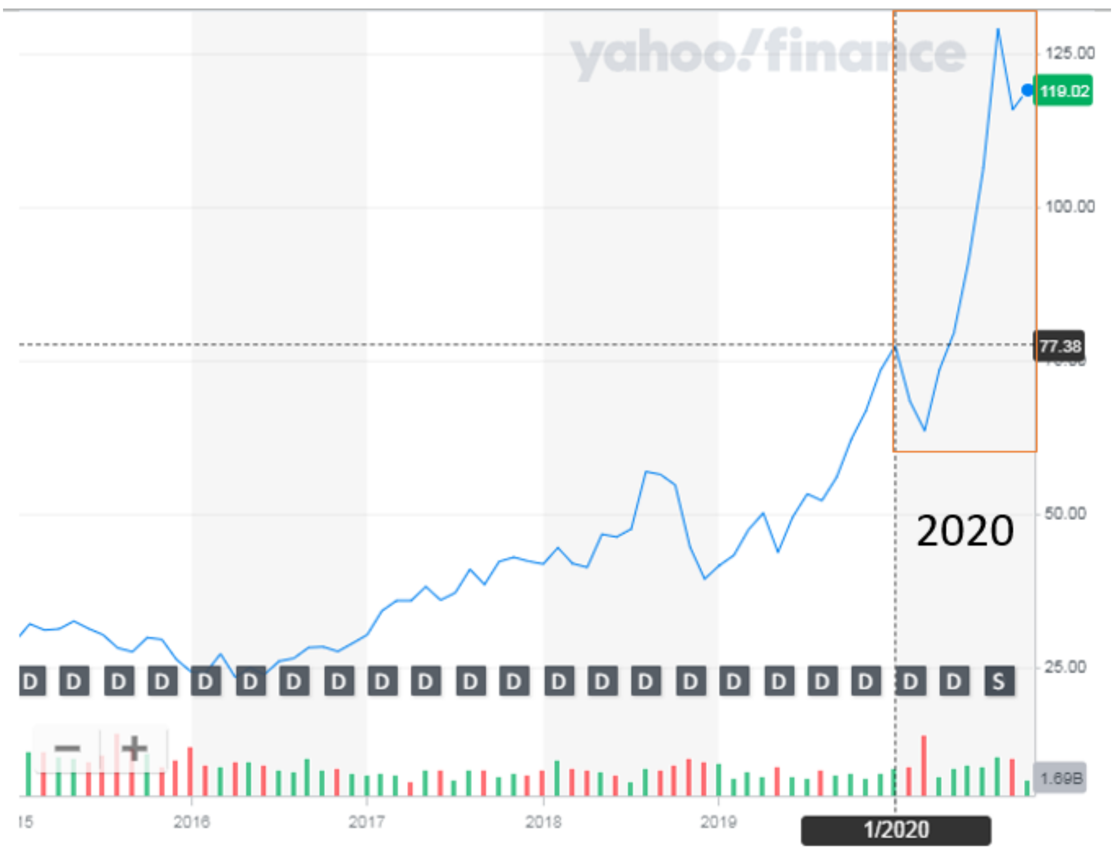


Figure 58: Actual movement of AAPL

From the above Figure 55, we can see that Prophet gives us a general prediction of the stock trend. When comparing the Figure 56, although the trend of prediction is positive which match the read trend, the increasing rate is far less than the ture rate.

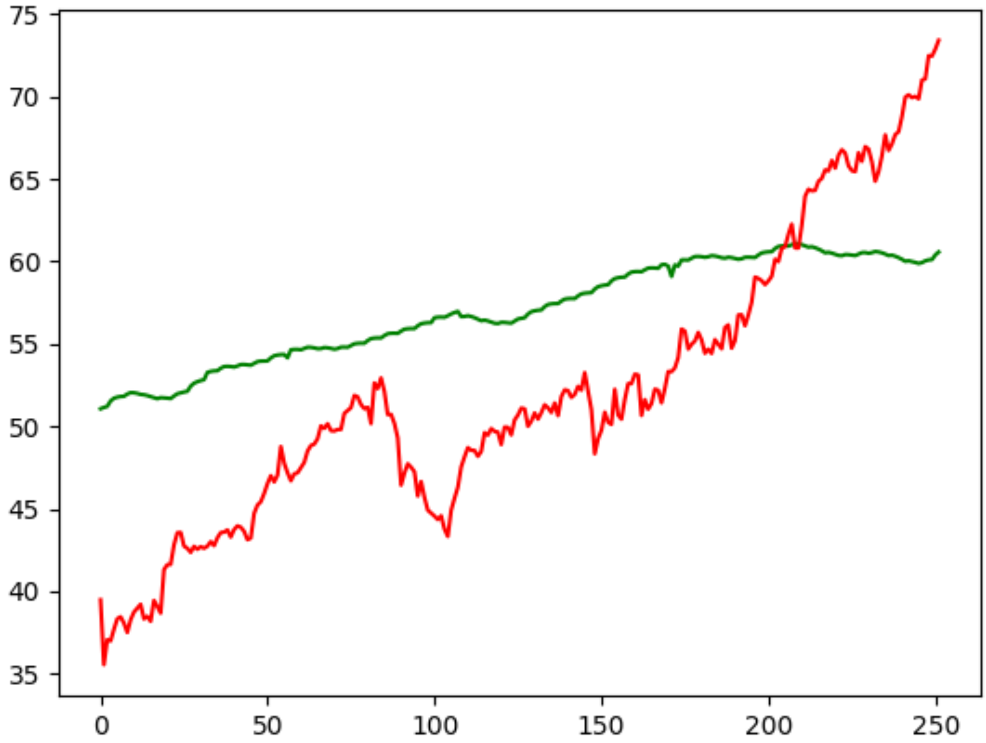


Figure 59: Prophet predicting 2019

We once again use Prophet to predict the stock trend in 2019. From the figure above, we can see that Prophet's forecast is not sensortive to local ups and downs. The MSE of the prediction is about 150, RMSE is around 12. In a nutshell, Prophet is good at give a general trend of the data set, but is not a good model to predict rapidly changing data, like predicting the closing price of stock.

4.3 Experiments on Textural Analysis

4.3.1 TextBlob + Clustering

We first use TextBlob as our starting point to try out what is sentiment analysis. We use “Sentiment Analysis for Financial News” dataset from Kaggle to test and validate the TextBlob model.

```
      Sentiment                                          Headlines  tb_hl_polarity
0             0  According to Gran , the company has no plans t...               0
1             0  Technopolis plans to develop in stages an area...               1
2            -1  The international electronic industry company ...               0
3             1  With the new production plant the company woul...              -1
4             1  According to the company 's updated strategy f...               0
...         ...                                                ...             ...
4841         -1  LONDON MarketWatch -- Share prices ended lower...              -1
4842          0  Rinkuskiai 's beer sales fell by 6.5 per cent ...               0
4843         -1  Operating profit fell to EUR 35.4 mn from EUR ...               0
4844         -1  Net sales of the Paper segment decreased to EU...               1
4845         -1  Sales in Finland decreased by 10.5 % in Januar...              -1

[4846 rows x 3 columns]

accurancy: 49.113
```

Figure 60: polarity score by TextBlob

As we can see from the figure above, “Sentiment” is the pre-defined polarity value for a headline, “tb_hl_polarity” is the polarity value generate from TextBlob. Comparing the two value, it shows that TextBlob has 49% of accuracy.

We then perform TextBlob on another dataset, New York Time. Noticed that New York Time general news are different from financial news, there are specific terms for financial area, so the result may vary. We first pass the dataset into TextBlob to generate polarity scores.

```
       Unnamed: 0        date  ...                                                url tb_hl_polarity
0               0  2010-01-01  ...  https://www.nytimes.com/2010/01/01/world/europ...         0.1375
1               1  2010-01-01  ...  https://schott.blogs.nytimes.com/2010/01/01/wo...         0.0000
2               2  2010-01-01  ...  https://fifthdown.blogs.nytimes.com/2010/01/01...         0.0000
3               3  2010-01-01  ...  https://bits.blogs.nytimes.com/2010/01/01/five...         0.0000
4               4  2010-01-02  ...  https://www.nytimes.com/2010/01/03/weekinrevie...         0.0000
...           ...         ...  ...                                                ...            ...
29080       29080  2020-10-30  ...  https://www.nytimes.com/interactive/2020/10/30...         0.0000
29081       29081  2020-10-31  ...  https://www.nytimes.com/2020/10/30/opinion/tec...         0.0000
29082       29082  2020-10-31  ...  https://www.nytimes.com/2020/10/31/business/cu...         0.0000
29083       29083  2020-10-31  ...  https://www.nytimes.com/2020/10/31/movies/sean...         0.0000
29084       29084  2020-10-31  ...  https://www.nytimes.com/2020/10/31/us/coronavi...         0.6000

[29044 rows x 5 columns]
```

Figure 61: New York Time dataset with TextBlob

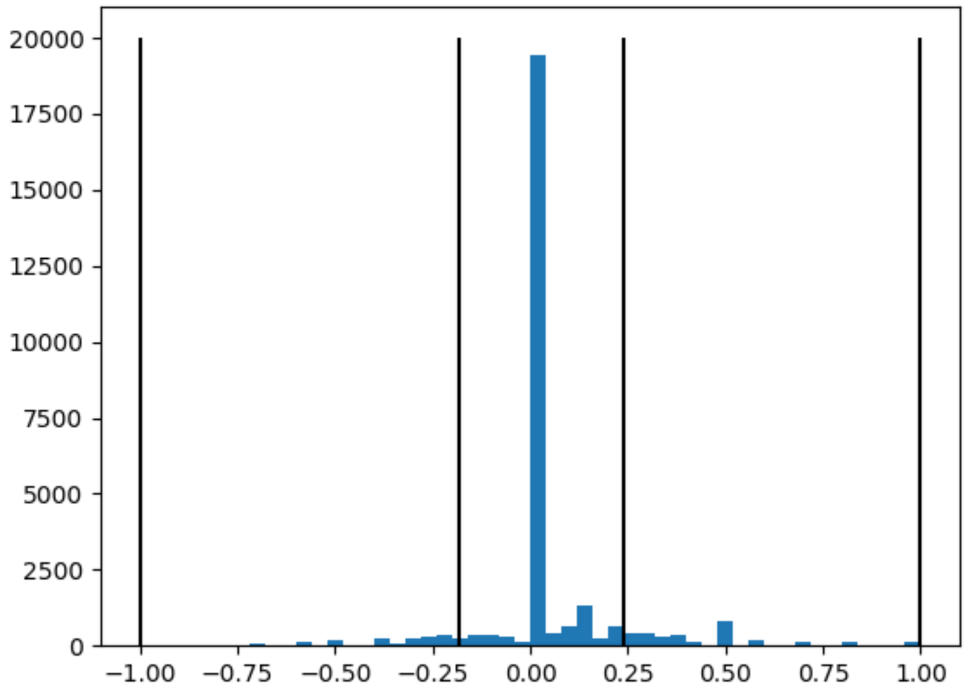


Figure 62: sentiment clustering from TextBlob

We apply the polarity scores generate from TextBlob to perform Jenks natural breaks optimization. It is a data clustering method designed to determine the best arrangement of values into different classes, since our dataset is only one-dimensional [32]. A clustering helps us to determine the actual category of a data, whether it is really a negative sentiment or it is actually false negative, since we cannot blindly rely on its value.

```
[-1.0, -0.18181818181818182, 0.2380952380952381, 1.0]
```

Figure 63: Jenks break result for TextBlob

We find that values between [-1, -0.18] represent negative sentiment data, values between [-0.18, 0.23] are neutral, values between [0.23, 1] are positive sentiment.

4.3.2 VADAR Sentiment + Clustering

We continue to try out more sentiment analysis techniques, this time we picked

VADAR Sentiment. As previous experiment, we will also use "Sentiment Analysis for Financial News" dataset from Kaggle to quickly test and validate our VADAR Sentiment model.

```
      Sentiment                                          Headlines  vadar_polarity
0             0  According to Gran , the company has no plans t...              -1
1             0  Technopolis plans to develop in stages an area...              -1
2            -1  The international electronic industry company ...               0
3             1  With the new production plant the company woul...               1
4             1  According to the company 's updated strategy f...               1
...         ...                                                ...             ...
4841         -1  LONDON MarketWatch -- Share prices ended lower...              -1
4842          0  Rinkuskiai 's beer sales fell by 6.5 per cent ...               0
4843         -1  Operating profit fell to EUR 35.4 mn from EUR ...               1
4844         -1  Net sales of the Paper segment decreased to EU...               1
4845         -1  Sales in Finland decreased by 10.5 % in Januar...               0

[4846 rows x 3 columns]

accurancy: 54.354
```

Figure 64: polarity score by VADAR

As we can see from the figure above, "Sentiment" is the pre-defined polarity value for a headline, "polarity" is the polarity value generate from VADAR Sentiment. Comparing the two value, it shows that VADAR Sentiment has 54% of accuracy.

We then perform VADAR Sentiment on another dataset, New York Time. Noticed that New York Time general news are different from financial news, there are specific terms for financial area, so the result may vary. We first pass the dataset into VADAR Sentiment to generate polarity scores.

```
       Unnamed: 0        date  ...                                                url vadar_compounds
0               0  2010-01-01  ...  https://www.nytimes.com/2010/01/01/world/europ...           0.701
1               1  2010-01-01  ...  https://schott.blogs.nytimes.com/2010/01/01/wo...           0.000
2               2  2010-01-01  ...  https://fifthdown.blogs.nytimes.com/2010/01/01...           0.000
3               3  2010-01-01  ...  https://bits.blogs.nytimes.com/2010/01/01/five...           0.000
4               4  2010-01-02  ...  https://www.nytimes.com/2010/01/03/weekinrevie...           0.000
...           ...         ...  ...                                                ...             ...
29080       29080  2020-10-30  ...  https://www.nytimes.com/interactive/2020/10/30...           0.000
29081       29081  2020-10-31  ...  https://www.nytimes.com/2020/10/30/opinion/tec...           0.000
29082       29082  2020-10-31  ...  https://www.nytimes.com/2020/10/31/business/cu...           0.000
29083       29083  2020-10-31  ...  https://www.nytimes.com/2020/10/31/movies/sean...           0.000
29084       29084  2020-10-31  ...  https://www.nytimes.com/2020/10/31/us/coronavi...          -0.296

[29044 rows x 5 columns]
```

Figure 65: New York Time dataset with VADAR

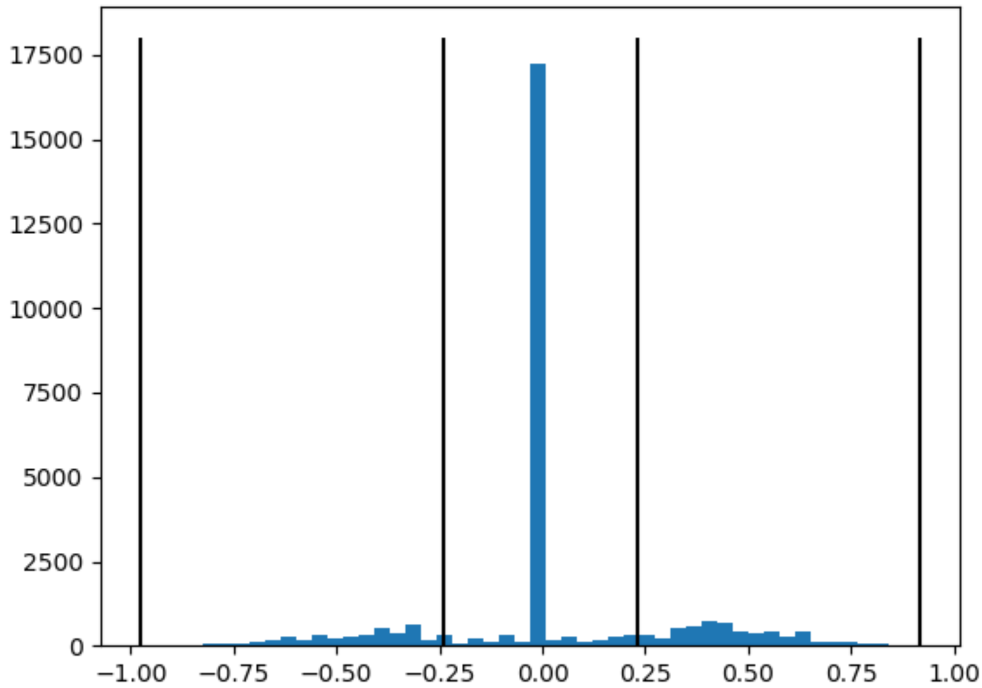


Figure 66: sentiment clustering from VADAR

We apply the polarity scores generate from TextBlob along with Jenks natural breaks optimization to get a more accurate result.

```
[-0.9776, -0.2382, 0.2321, 0.9186]
```

Figure 67: Jenks break result for VADAR

As we see from the figure, values between [-0.9776, -2.2382] are negative sentiment, values between [-0.2382, 0.2321] are neutral sentiment, values between [0.2321, 0.9186] are positive sentiment.

4.3.3 ANN

We once again using “Sentiment Analysis for Financial News” dataset from Kaggle to test and validate our ANN model.

First we built our test model that inspired from online resources, using Cross-Entropy as loss function, multiple hidden layer with ReLU as activation function and Softmax as output layer's activation function, while choosing Adam as optimizer.

Here are the model summary from various test model:

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 128)               855040
_________________________________________________________________
dense_7 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_8 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_9 (Dense)              (None, 16)                1040
_________________________________________________________________
dense_10 (Dense)             (None, 8)                 136
_________________________________________________________________
dense_11 (Dense)             (None, 3)                 27
=================================================================
Total params: 881,011
Trainable params: 881,011
Non-trainable params: 0
```

Figure 68: ANN test model 1

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_12 (Dense)             (None, 64)                427520
_________________________________________________________________
dense_13 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_14 (Dense)             (None, 16)                1040
_________________________________________________________________
dense_15 (Dense)             (None, 8)                 136
_________________________________________________________________
dense_16 (Dense)             (None, 3)                 27
=================================================================
Total params: 432,883
Trainable params: 432,883
Non-trainable params: 0
```

Figure 69: ANN test model 2

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_17 (Dense)             (None, 256)               1710080
_________________________________________________________________
dense_18 (Dense)             (None, 256)               65792
_________________________________________________________________
dense_19 (Dense)             (None, 128)               32896
_________________________________________________________________
dense_20 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_21 (Dense)             (None, 16)                1040
_________________________________________________________________
dense_22 (Dense)             (None, 8)                 136
_________________________________________________________________
dense_23 (Dense)             (None, 3)                 27
=================================================================
Total params: 1,818,227
Trainable params: 1,818,227
Non-trainable params: 0
```

Figure 70: ANN test model 3

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_24 (Dense)             (None, 128)               855040
_________________________________________________________________
dense_25 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_26 (Dense)             (None, 32)                2080
_________________________________________________________________
dense_27 (Dense)             (None, 16)                528
_________________________________________________________________
dense_28 (Dense)             (None, 8)                 136
_________________________________________________________________
dense_29 (Dense)             (None, 3)                 27
=================================================================
Total params: 866,067
Trainable params: 866,067
Non-trainable params: 0
```

Figure 71: ANN test model 4

And here are their train model loss, train model accuracy, test model loss and test model accuracy respectively:

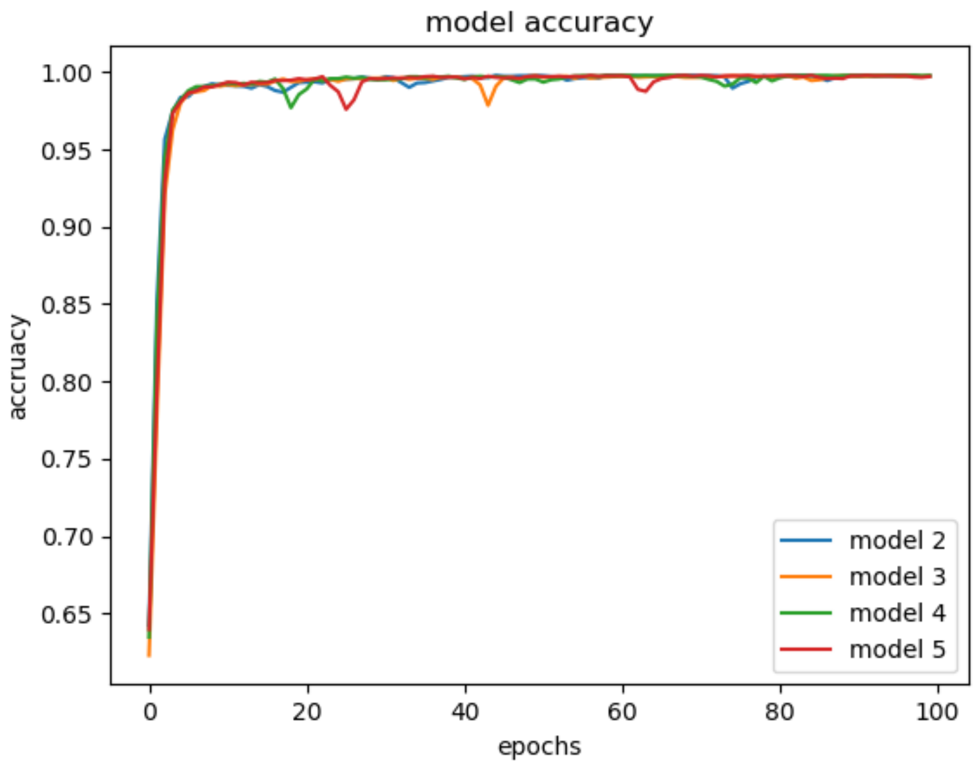


Figure 72: ANN test model accuracy

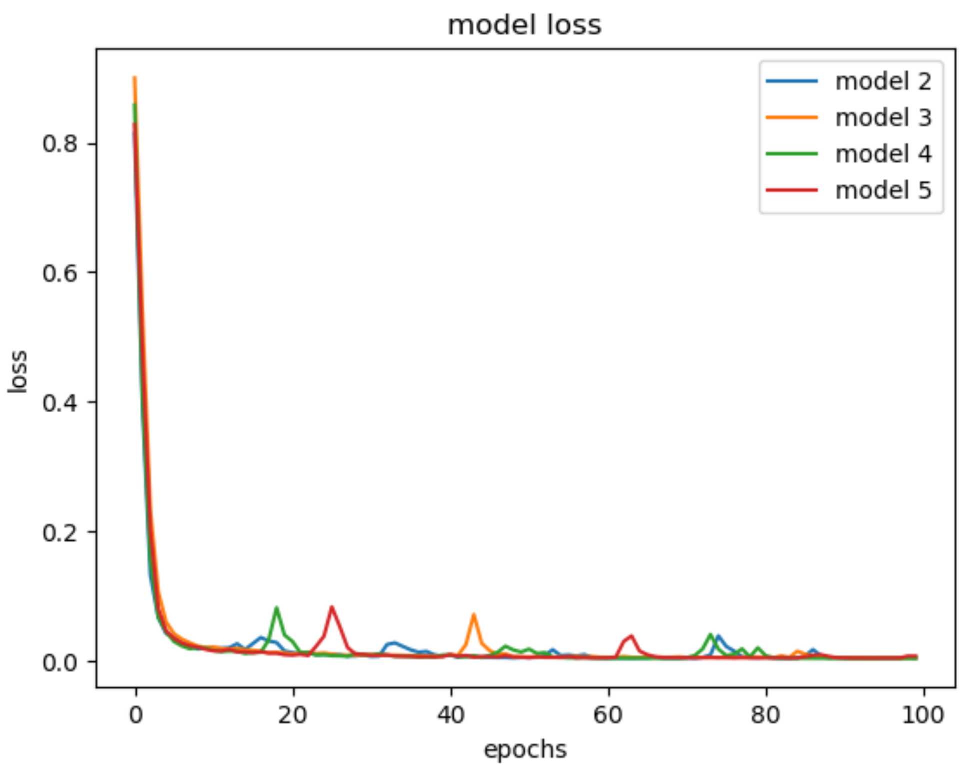


Figure 73: ANN test model loss

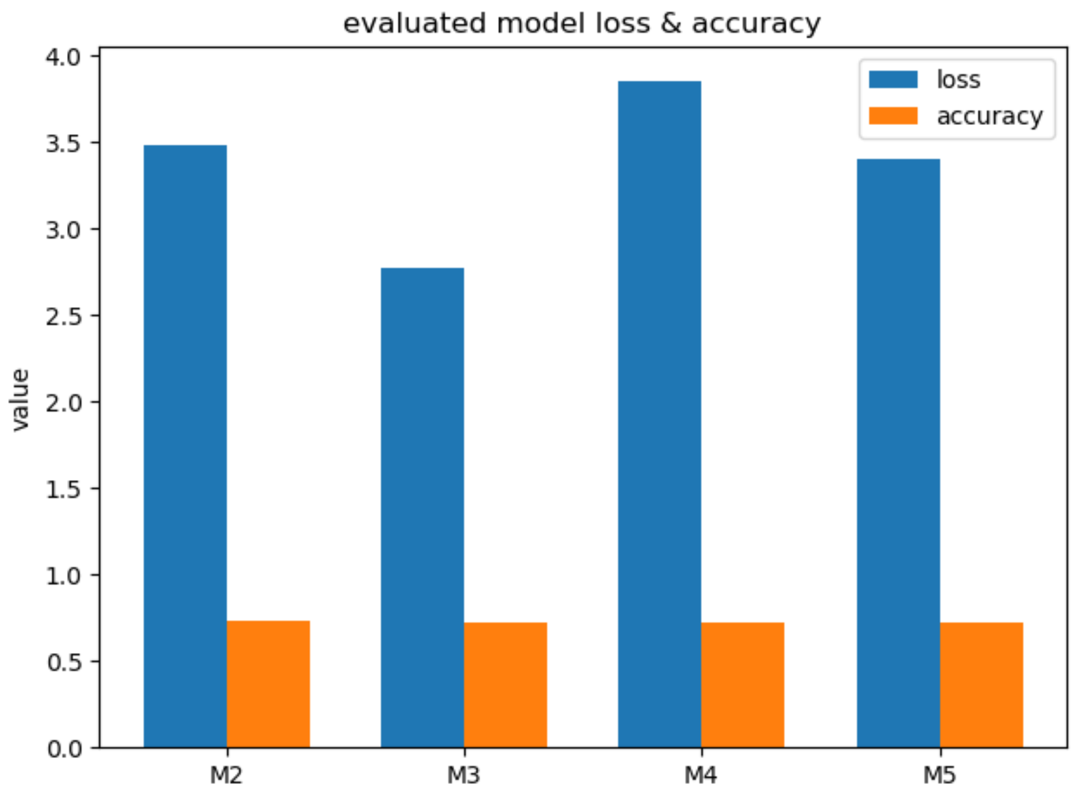


Figure 74: ANN test model validation loss & accuracy

We can see from the figure, there is no much difference between the number of layers and number of neurons in the neural network, and there is no much change when epoch reaches 20. So we concluded the following model:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 128)               855040
_________________________________________________________________
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_3 (Dense)              (None, 16)                1040
_________________________________________________________________
dense_4 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_5 (Dense)              (None, 3)                 27
=================================================================
Total params: 881,011
Trainable params: 881,011
Non-trainable params: 0
```

Figure 75: ANN final model

This model has a total of 6 layer and 20 epochs. It's accuracy is 0.7278 and Cross-entropy loss is 2.2990.

4.3.4 BERT

We use the "Sentiment Analysis for Financial News" dataset to train the BERT model. This label is already labeled so it is good to use as training dataset. We take 80% of the dataset as the training set, the rest is the test set.

1	neutral	According to Gran , the company has no plans to move all production to Russia , although that is where the company is growing .
2	neutral	Technopolis plans to develop in stages an area of no less than 100,000 square meters in order to host companies working in compu
3	negative	The international electronic industry company Elcoteq has laid off tens of employees from its Tallinn facility ; contrary to earlier lay
4	positive	With the new production plant the company would increase its capacity to meet the expected increase in demand and would improv
5	positive	According to the company 's updated strategy for the years 2009-2012 , Basware targets a long-term net sales growth in the range o
6	positive	FINANCING OF ASPOCOMP 'S GROWTH Aspocomp is aggressively pursuing its growth strategy by increasingly focusing on tec
7	positive	For the last quarter of 2010 , Componenta 's net sales doubled to EUR131m from EUR76m for the same period a year earlier , while
8	positive	In the third quarter of 2010 , net sales increased by 5.2 % to EUR 205.5 mn , and operating profit by 34.9 % to EUR 23.5 mn .

Figure 76: Input sample

The accuracy of the analysing test set is 81.6%. The Matthews correlation coefficient (MCC) is 0.62585. Balanced accuracy is 80.3%.

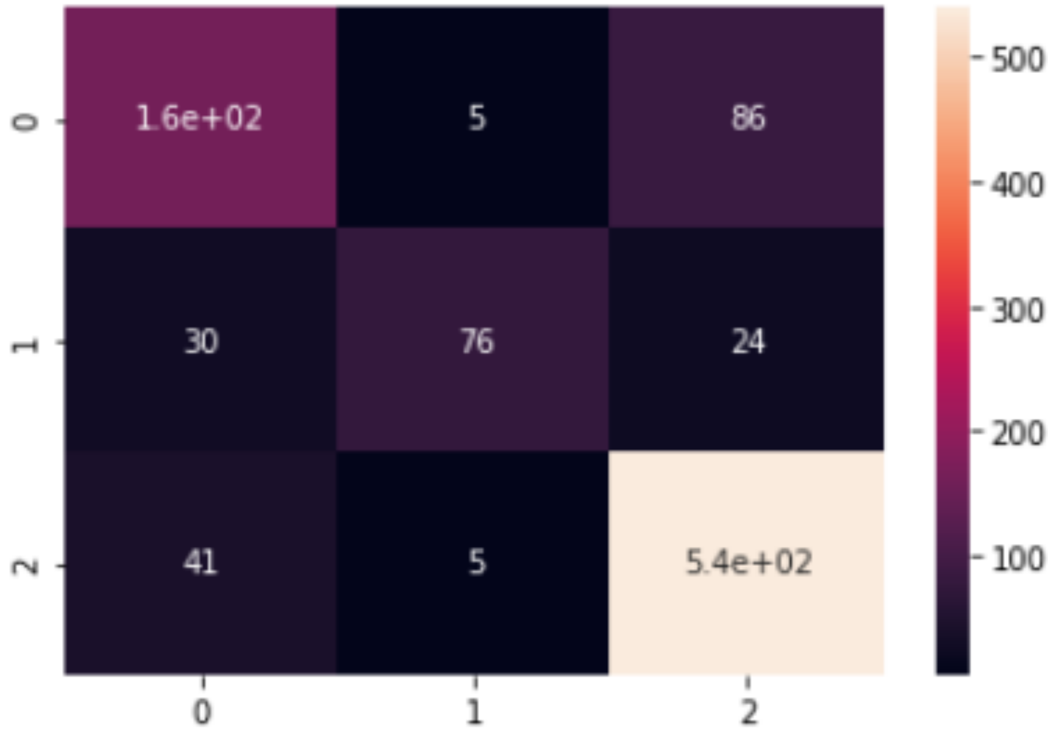


Figure 77: Heatmap of confusion matrix

4.4 Experiments on Merged Model

4.4.1 KNN + VADAR Sentiment

We combine KNN model for numerical analysis and VADAR for textural analysis to test the indication of raise or drop for apple stock in one day range, in other words, using today's sentiment to predict whether or not the stock price will increase or decrease tomorrow. To achieve this, we use news data crawl from New York Time to build polarity score of each news report using VADAR Sentiment, and group/cluster them using Jenks Algorithm. If there are multiple news per day, we will calculate the average polarity score of that day.

	date	vadar_compounds	vadar_compounds_updated
0	2010-01-01	0.7010	1
1	2010-01-01	0.0000	0
2	2010-01-01	0.0000	0
3	2010-01-01	0.0000	0
4	2010-01-02	0.0000	0
5	2010-01-02	0.0000	0
6	2010-01-02	0.0000	0
7	2010-01-03	0.0000	0
8	2010-01-03	-0.5859	-1
9	2010-01-04	0.0000	0

Figure 78: VADAR clustered, before taking mean

date	vadar_compounds	vadar_compounds_updated
2010-01-01	0.175250	0.250000
2010-01-02	0.000000	0.000000
2010-01-03	-0.292950	-0.500000
2010-01-04	0.061471	0.117647
2010-01-05	0.082738	0.125000
2010-01-06	0.062394	0.166667
2010-01-07	-0.057414	-0.142857
2010-01-08	0.087255	0.181818
2010-01-09	0.159225	0.250000
2010-01-10	0.000000	0.000000

Figure 79: VADAR after taking mean

	date	close	will-rise	actual-rise
0	2009-12-31	7.526072	1.0	1
1	2010-01-04	7.643214	1.0	1
2	2010-01-05	7.656428	0.0	1
3	2010-01-06	7.534643	0.0	0
4	2010-01-07	7.520714	1.0	0
5	2010-01-08	7.570714	0.0	1
6	2010-01-11	7.503929	0.0	0
7	2010-01-12	7.418571	1.0	0
8	2010-01-13	7.523214	0.0	1
9	2010-01-14	7.479643	0.0	0

Figure 80: stock price data

	date	close	will-rise	actual-rise	vadar_compounds	vadar_compounds_updated
0	2010-01-04	7.643214	1.0	1	0.061471	0.117647
1	2010-01-05	7.656428	0.0	1	0.082738	0.125000
2	2010-01-06	7.534643	0.0	0	0.062394	0.166667
3	2010-01-07	7.520714	1.0	0	-0.057414	-0.142857
4	2010-01-08	7.570714	0.0	1	0.087255	0.181818
5	2010-01-11	7.503929	0.0	0	0.021744	0.111111
6	2010-01-12	7.418571	1.0	0	0.318200	1.000000
7	2010-01-13	7.523214	0.0	1	-0.087322	-0.111111
8	2010-01-14	7.479643	0.0	0	0.109686	0.285714
9	2010-01-15	7.354643	1.0	0	0.189640	0.400000

Figure 81: stock price & VADAR sentiment data

We observed that there are improvements when deciding the stock price will increase or not on the coming day using sentiment score. While prefer using sentiment score over KNN classification to predict tomorrows trend, it has higher accuracy of 51.39% over 50.67%.

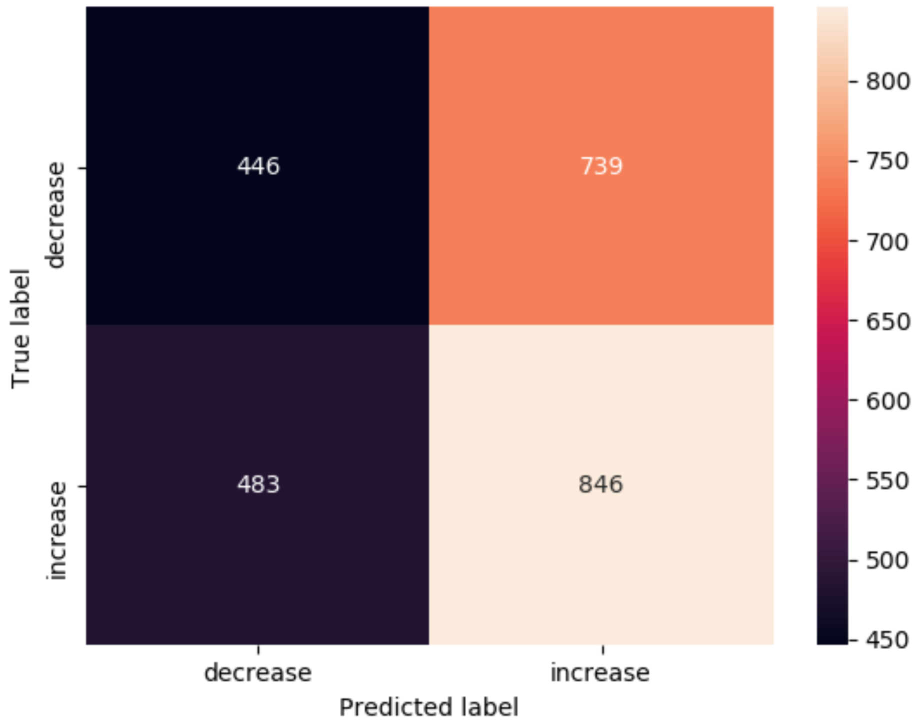


Figure 82: confusion matrix w/ sentiment

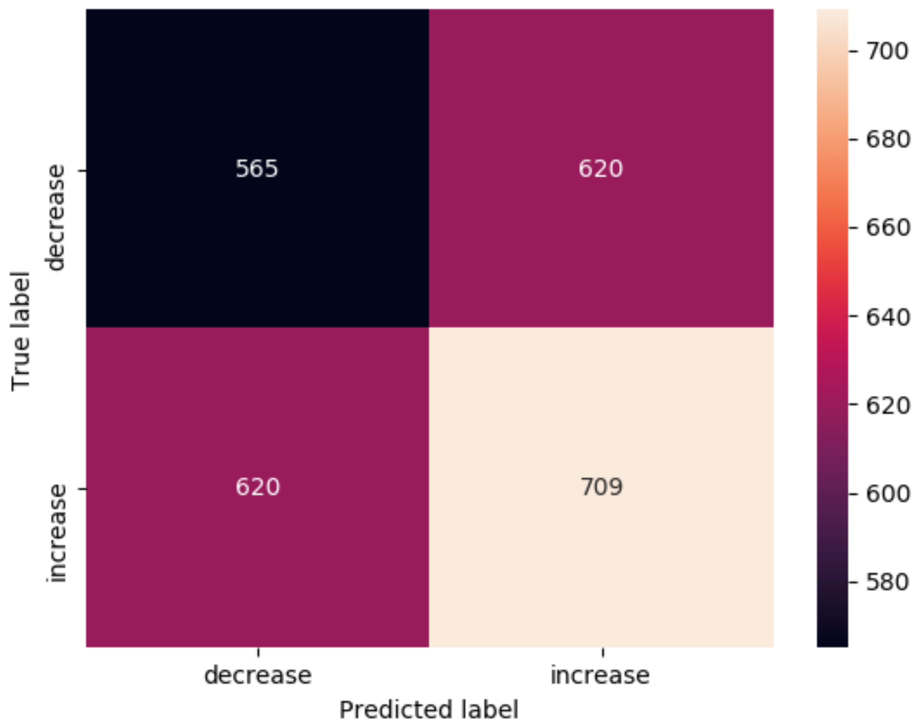


Figure 83: confusion matrix w/o sentiment

	w/ sentiment	w/o sentiment	changes
accuracy	51.39%	50.68%	0.72%
balanced accuracy	50.65%	50.51%	0.13%

Figure 84: KNN + VADAR overall performance

Although the result seems have no impact on stock price prediction, it is still meaningful to investors since it can play as a reference for them to have a general view on the market's emotion and trend.

4.4.2 LSTM + BERT

After the completion of BERT model training, we use to analysis the dataset we clawler from New York Time. There are many infrom for each news or article, including headline, abstract, leading paragraph. We mainly focus on analyzing the abstract with BERT, and we use BERT model to prepare a sentiment classification for each news or articles.

We cannot directly use the previously mentioned LSTM module and sentiment analysis module to integrate, because the experimental results of that LSTM module only give a delayed version of the real trend. It is meaningless to directly use the predicted closing price of day t + 1 and compare it with the closing price of the day t to classify up or down. If we use this classification, it will always predict the stock will rise on the next trading day when the local trend is going downward, and it will always predict the stock will fall in an upward trend. From this we can know that the balanced accuracy of the prediction will not be good as it will give to many false negative and false positive predictions.

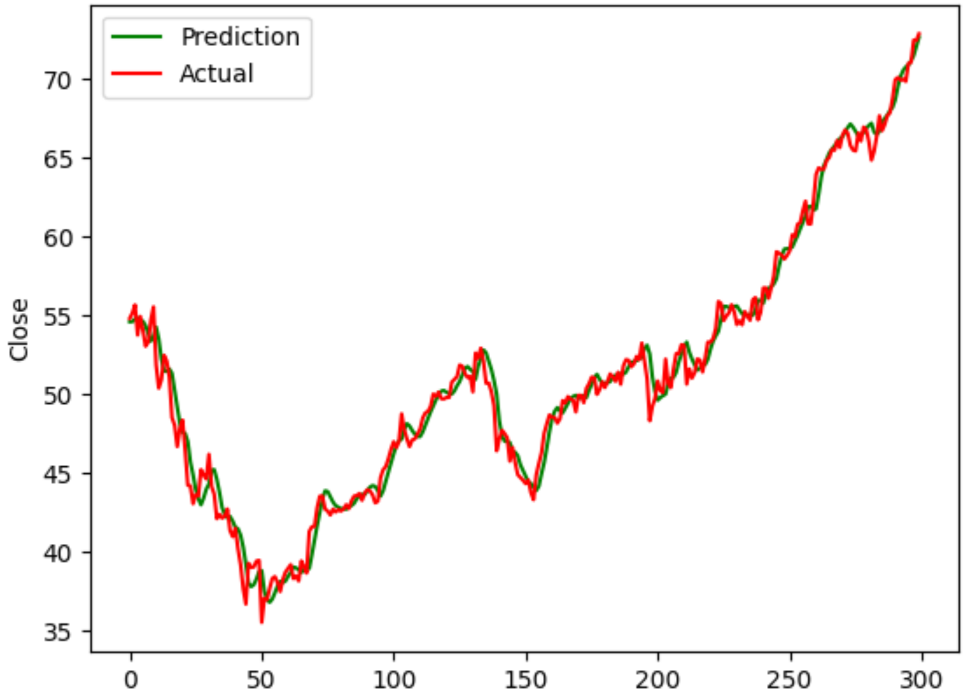


Figure 85: Prediction of the previous LSTM model

Therefore, before integrating both model, we need to modify the lstm model to directly prediction whether the stock will increase or not instead of predicting the closing price of next trading day. And also adding the stock data of Standard & Poor 500 index and Nasdaq Composite Index to the model.

As the sentiment analysis output from BERT is based on each news or articles, but the input unit of the LSTM is day. There is more than one news or article in a day, and then we need to find the average sentiment value for each date before apply to the LSTM model.

	AAPL_high	AAPL_low	AAPL_open	AAPL_close	AAPL_volume	AAPL_adj close	^IXIC_high	^IXIC_low	^IXIC_open	^IXIC_close
0	7.660714	7.585000	7.622500	7.643214	493729600.0	6.539882	2311.149902	2294.409912	2294.409912	2308.419922
1	7.699643	7.616071	7.664286	7.656428	601904800.0	6.551187	2313.729980	2295.620117	2307.270020	2308.709961
2	7.686786	7.526786	7.656428	7.534643	552160000.0	6.446983	2314.070068	2295.679932	2307.709961	2301.090088
3	7.571429	7.466072	7.562500	7.520714	477131200.0	6.435065	2301.300049	2285.219971	2298.090088	2300.050049
4	7.571429	7.466429	7.510714	7.570714	447610800.0	6.477847	2317.600098	2290.610107	2292.239990	2317.169922
...	...	...	...	...	...	...	...	...	...	...
2721	116.550003	112.879997	114.010002	115.050003	111850700.0	114.851852	11545.629883	11221.059570	11440.639648	11358.940430
2722	117.279999	114.540001	115.489998	116.599998	92276800.0	116.399178	11465.059570	11361.860352	11409.339844	11431.349609
2723	115.430000	111.099998	115.050003	111.199997	143937800.0	111.008476	11249.950195	10999.070312	11230.900391	11004.870117
2724	116.930000	112.199997	112.370003	115.320000	146129200.0	115.121384	11287.629883	11030.190430	11064.469727	11185.589844
2725	111.989998	107.720001	111.059998	108.860001	190272600.0	108.672516	11129.809570	10822.570312	11103.469727	10911.589844

	^IXIC_volume	^IXIC_adj close	^GSPC_high	^GSPC_low	^GSPC_open	^GSPC_close	^GSPC_volume	^GSPC_adj close	sentiment
0	1931380000	2308.419922	1133.869995	1116.560059	1116.560059	1132.989990	3991400000	1132.989990	0.614706
1	2367860000	2308.709961	1136.630005	1129.660034	1132.660034	1136.520020	2491020000	1136.520020	0.827375
2	2253340000	2301.090088	1139.189941	1133.949951	1135.709961	1137.140015	4972660000	1137.140015	0.623944
3	2270050000	2300.050049	1142.459961	1131.319946	1136.270020	1141.689941	5270680000	1141.689941	-0.574143
4	2145390000	2317.169922	1145.390015	1136.219971	1140.520020	1144.979980	4389590000	1144.979980	0.872545
...	...	...	...	...	...	...	...	...	...
2721	3186950000	11358.940430	3441.419922	3364.860107	3441.419922	3400.969971	3988080000	3400.969971	-0.189500
2722	3079530000	11431.349609	3409.510010	3388.709961	3403.149902	3390.679932	3946990000	3390.679932	0.341500
2723	3912580000	11004.870117	3342.479980	3268.889893	3342.479980	3271.030029	5129860000	3271.030029	0.282444
2724	3222460000	11185.589844	3341.050049	3259.820068	3277.169922	3310.110107	4903070000	3310.110107	-0.272000
2725	3662840000	10911.589844	3304.929932	3233.939941	3293.590088	3269.959961	4840450000	3269.959961	0.468818

Figure 86: Input sample

The architecture of the integrated model is starting with a input layer, because the sliding window size is 10 and 19 feature each each day, so there are 190 unit in the input layer. The model followed by two LSTM layers with 256 and 64 units respectively. And then there is a dense layer with 32 unit. Finally, the output layer has one unit to give the classificaiton.

We have observed in the training set or test set that stock appreciation (the closing price on day t + 1 is higher than the closing price on day t) occurs more often than devaluation. The stock price inceased in 51.37% days of training set and 58.92% days of testing set. It observation mattch the overall trend of AAPL. The accuracy of the model is 0.5208, and the balanced accuracy is 0.5213. We think this model although give us a balanced accurcy high then 50%, always predict the stock will increase on the next trading day is still better than our model in the case of AAPL.

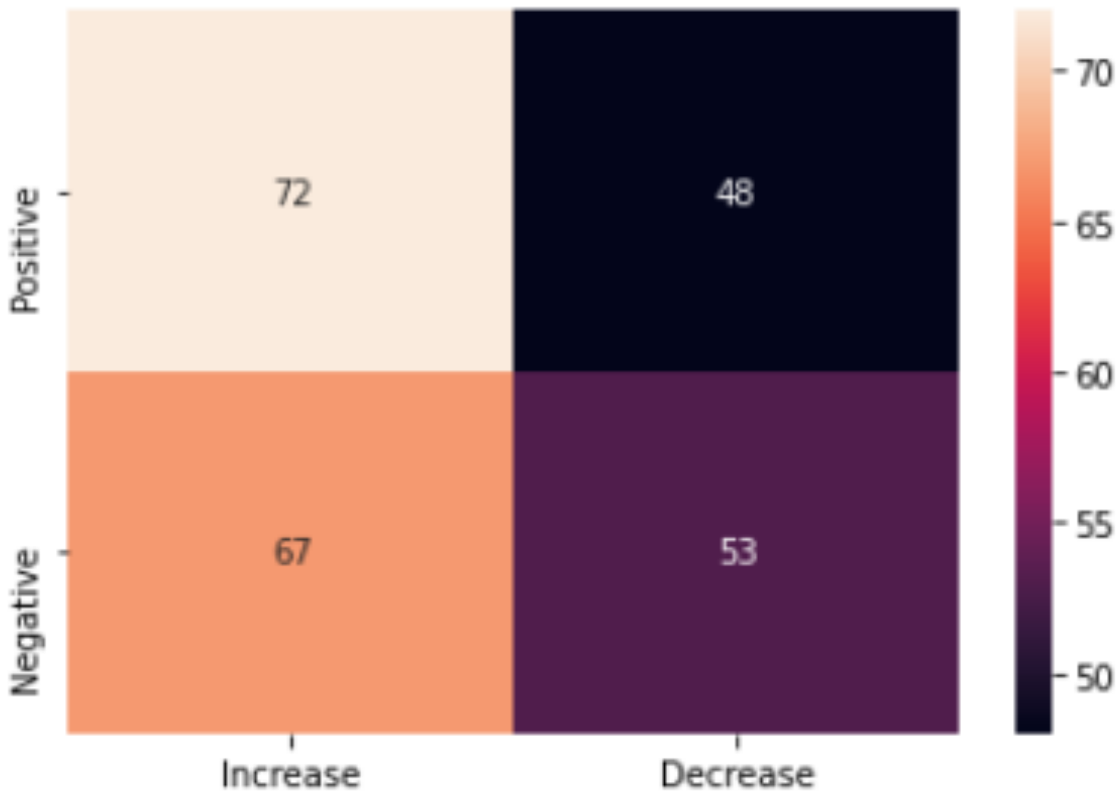


Figure 87: Heatmap of confusion matrix

However, adding news and articles which are related to Apple to our model indeed gives us an imporvement in predicting tomorrow trend. We use stock data and news sentiment to train our model and then use two testing sets to test the model. These two testing sets almost identical, the only difference is that we set all sentiment values to be 0 in one of them.

From this experiment, we observe that the testing set with sentiment values set to be 0 have a lower accuracy in prediction the closeing preice of the next trading day. The accuracy is 0.5083 and the balanced accuracy is 0.5092.

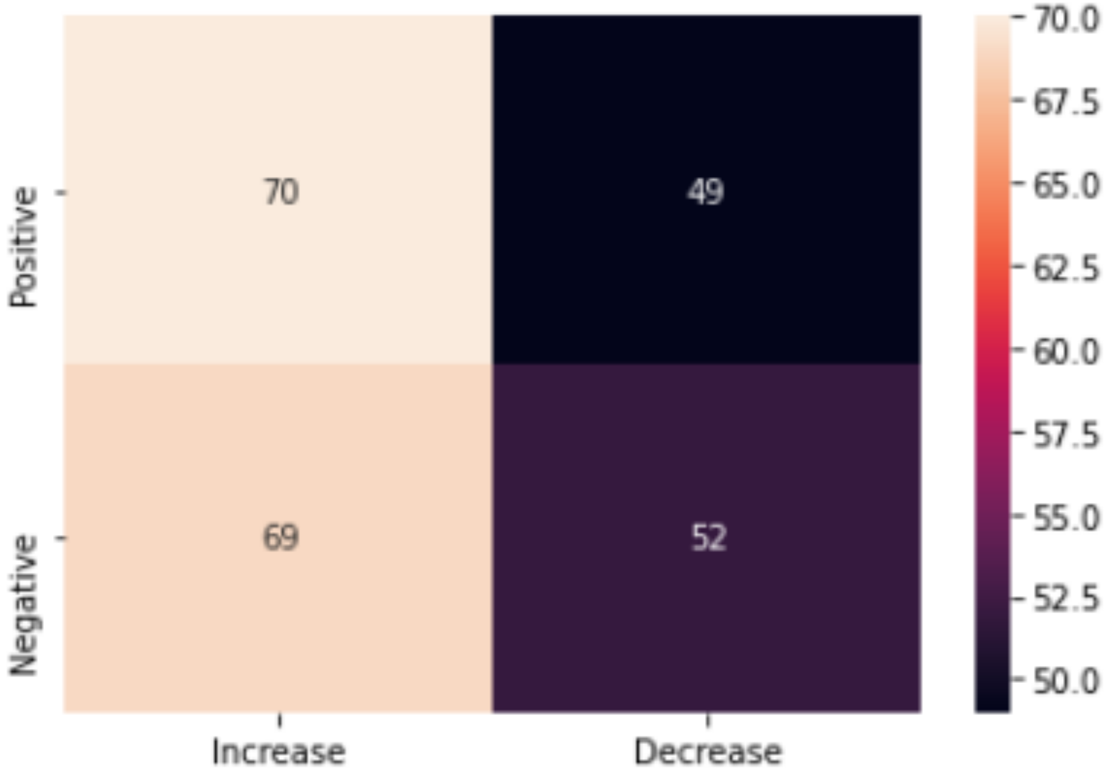


Figure 88: Heatmap of confusion matrix (sentiment set to 0)

Adding sentiment value to each day improved the model by 1.25% and 1.22% in accuracy and balanced accuracy respectively.

	No sentiment	With sentiment	Changes
Accuracy	50.83%	52.08%	+1.25%
Balanced accuracy	50.92%	52.14%	+1.22%

Figure 89: Result comparison

Here we do a case study on the how the adding sentiment values to the model imporve the perfomce. The example below show that the classification without sentiment is “Fall”, however, an the correct classification should be “Rise”. In this case, adding the sentiment value make the model give a corrent prediction.

	High	Low	Open	Close	Volume	Adj Close	Sentiment	Ground truth	Prediction
Without sentiment	44.80	44.17	44.49	44.58	8.49E+07	43.97	N/A	Fall	Rise
With sentiment							-0.613		Fall
Without sentiment	44.48	42.57	43.9	43.325	1.62E+08	42.74	N/A	Rise	Fall
With sentiment							0.359		Rise

Figure 90: Case studies

5. Conclusion

In this project, we used several machine learning models, such as Gated Recurrent Units (GRU), Long-Short Term Memory (LSTM), K-Nearest Neighbor Regression (KNNR), K-Nearest Neighbor Classification (KNNC), and Prophet for numerical analysis. For textural analysis, we use TextBlob, VADAR Sentiment, Artificial Neural Network (ANN) and Bidirectional Encoder Representations from Transformers (BERT). Each of those techniques have its own pros and cons, it depends on the use case and environment. There is no perfect solution for either stock price prediction nor sentiment analysis. All we can do is use the right method at the right place, on the right time.

In the aspect of numerical analysis, all the prediction of these models show different degree of delay. Whenever there is an upwards or downwards trend, the delay in the prediction is more obvious. In the other words, the model uses the past closing as the prediction of the next closing price. Therefore, These numerical forecasts cannot be used to predict whether the closing price of tomorrow will rise or fall.

First off, sentiment data for general news and finicial news are different. These two model cannot directly apply on each other. For example, if we train a LSTM model using finicial news data, and apply prediction on genreal news. The polarity score may become strange or may even affect the predition accuracy. Unfortunately, there is only financial news dataset that is pre-labelled with polarity score, we are forced to use this dataset to train our model.

Moreover, the polarity scores generated by the sentiment analysis modules are not always correctly representing the “sentiment” of the text. For instance, there is still no sentiment analysis techniques that is able to distinct sarcasm in a given text. Therefore, such a false result will be used as the input of the final module, so the input of the module will have a certain degree of error, and this error will further affect the accuracy of prediction. However, adding the sentiment value of the news to the input feature does improve the prediction in some scenario, such as providing some sort of reference for investors to have a feeling on market’s emotion and its trend.

6. Furture development

1. To better understand the sign of upward trend, we can get more data from other stock, not limiting to Apple only. To identify the patterns that may imply an upward or downward trend.

2. News can only help us to guest the reaction of the investor but not directly understand their emotion. We can study what have been discussed on the social media to directly observe their emotion and use this information to futher imporve our prediction model.

3. Formulate investment strategies, train the modules to learn to find out the market entry and delisting opportunities, evaluating the model by the profit it can make.

4. Based on the developed model to build an investment assistance tool. Users will be able to adjust some parameter, such as adjusting the influence of the daily news to the prediction given by the model.

Reference

[1] J.L. Balcazar, R. Gavalda, H.T. Siegelmann, "Computational power of neural networks: a characterization in terms of Kolmogorov complexity," in *IEEE Transactions on Information Theory (Volume: 43, Issue: 4)*, 1977.

[2] Yann LeCun, Yoshua Bengio, Geoffrey Hinton , "Deep learning," in *Nature volume 521*, 2015.

[3] V. Shchutskaya, "Deep Learning: Strengths and Challenges," 27 7 2018. [Online]. Available: https://indatalabs.com/blog/deep-learning-strengths-challenges.

[4] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, Jascha Sohl-Dickstein, "ON THE EXPRESSIVE POWER OF DEEP NEURAL NETWORKS," in *ICLR 2017*, 2017 .

[5] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, Liwei Wang, "The Expressive Power of Neural Networks: A View from the Width," in *arXiv*, 2017.

[6] Daniel Justus, John Brennan, Stephen Bonner, Andrew Stephen McGough, "Predicting the Computational Cost of Deep," 2018.

[7] Merve Alanyali, Helen Susannah Moat & Tobias Preis, "Quantifying the Relationship Between Financial News and the Stock Market," in *Scientific Reports 3*, 2013.

[8] CH.RAGA MADHURI, MUKESH CHINTA, V V N V PHANI KUMAR, "Stock Market Prediction for Time-series Forecasting," in *7th IEEE International Conference on Smart Structures and Systems*, 2020.

[9] Mehar Vijha, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," in *Procedia*

Computer Science 167, 2020.

[10] Mehak Usmani, Syed Hasan Adil, Kamran Raza, Syed Saad Azhar Ali, "Stock market prediction using machine learning techniques," in *3rd International Conference on Computer and Information Sciences* , Kuala Lumpur, Malaysia, 2016.

[11] A. Tipirisetty, "Stock Price Prediction using Deep Learning," in *Master's Projects. 636*, 2018.

[12] Anshul Mittal, Arpit Goel, "Stock Prediction Using Twitter Sentiment Analysis," in *Stanford University Working Paper*, 2012.

[13] Nazri Mohd Nawi, Walid Hasen Atomi, M. Z. Rehman , "The Effect of Data Pre-Processing on Optimized Training of Artificial Neural Networks," in *Procedia Technology 11*, 2013.

[14] Lexalytics Inc, "Sentiment Analysis Explained," 2020. [Online]. Available: https://www.lexalytics.com/technology/sentiment-analysis.

[15] Wikipedia, "Sentiment analysis," Wikipedia, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Sentiment_analysis. [Accessed 2020].

[16] MonkeyLearn Inc, "Sentiment Analysis: A Definitive Guide," 2020. [Online]. Available: https://monkeylearn.com/sentiment-analysis/.

[17] C. Olah, "Understanding LSTM Networks," 27 08 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[18] F.A. Gers, J. Schmidhuber, F. Cummins, "Learning to forget: continual prediction with LSTM," in *Ninth International Conference on Artificial Neural Networks*

ICANN 99, 1999.

[19] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," in *arXiv*, 2014.

[20] T. Cover, P. Hart, "Nearest neighbor pattern classification," in *IEEE Transactions on Information Theory, vol. 13*, 1967.

[21] Sean J. Taylor, Benjamin Letham, "Forecasting at scale," 2017.

[22] GeekforGeeks, "Clustering in Machine Learning," 23 February 2020. [Online]. Available: https://www.geeksforgeeks.org/clustering-in-machine-learning/. [Accessed 2020].

[23] Wikipedia, "Cluster analysis," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Cluster_analysis.

[24] C. Hutto, "VADER-Sentiment-Analysis," 2020. [Online]. Available: https://github.com/cjhutto/vaderSentiment. [Accessed 2020].

[25] A. Beri, "SENTIMENTAL ANALYSIS USING VADER," [Online]. Available: https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664. [Accessed 2020].

[26] R. Malde, "A Short Introduction to VADER," 8 June 2020. [Online]. Available: https://towardsdatascience.com/an-short-introduction-to-vader-3f3860208d53. [Accessed 2020].

[27] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Atlanta, CA, 2014.

[28] S. Loria, "TextBlob: Simplified Text Processing," 2020. [Online]. Available: https://github.com/sloria/TextBlob. [Accessed 2020].

[29] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, "Deep contextualized word representations," in *arXiv:1802.05365*, 2018.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *arXiv:1810.04805*, 2018.

[31] P. Werbos, "Backpropagation through time: what it does and how to do it," in *Proceedings of the IEEE Vol:78*, 1990 .

[32] Wikipedia, "Jenks natural breaks optimization," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Jenks_natural_breaks_optimization. [Accessed 2020].

[33] Warren S. McCulloch, Walter Pitts, "A logical calculus of the ideas immanent in nervous activity," in *Bulletin of Mathematical Biophysics 5*, 1943.

[34] A. Tipiriset, "Stock Price Prediction using Deep Learning," 2018.

[35] G. Seif, "The 5 Clustering Algorithms Data Scientists Need to Know," 6 February 2018. [Online]. Available: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68. [Accessed 2020].

[36] P. Pandey, "Simplifying Sentiment Analysis using VADER in Python (on Social Media Text)," 23 Septemper 2018. [Online]. Available: https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f. [Accessed 2020].

[37] K. Lee and R. Timmons, "Predicting the Stock Market with News Articles," 2007.