

The Chinese University of Hong Kong  
Department of Computer Science and Engineering  
Final Year Project Report (Semester 2)

# 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 many factors contribute to the changes in demand and supply. However, stock price prediction is the action that people have been doing for a long time. To find a perfect timing for the transaction, build a portfolio to reduce the risk. And the goal is to make more money. Some techniques are frequently used to analyze stock price, including technical analysis, fundamental analysis, and quantitative analysis, etc. Machine learning algorithms are getting more and more powerful and support from the hardware in recent years. People have started to leverage artificial intelligence 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 and social media content. Therefore, we think the news is another contributing consideration of their decision making, we also consider the public emotion in social media. In this project, 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. In textual analysis, we will perform sentiment analysis on news data related to Apple and on the comments and posts posted in social media, like YouTube, Twitter, and Reddit. We crawl data from APIs provided by these platforms and analyse the headlines' sentiment 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 the weekly progress report inspire us on this project.

We would like to extend our sincere appreciation to Edward YAU, this final year project assistant, for his constant support. He always opens 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 of inspiration and ideas from their remarkable work.

# Table of Content

Abstraction.....	2
Acknowledgements.....	3
Table of Content.....	4
Table of Figure.....	7
1. Introduction .....	13
1.1 Background .....	14
1.1.1 Deep Learning .....	14
1.1.2 Relations between News and Stock Market .....	16
1.1.3 Relations between Twitter and Stock Market.....	17
1.1.4 Candlestick Chart .....	17
1.1.5 Double bottom pattern .....	21
1.2 Motivation.....	22
1.3 Objective .....	23
1.4 Report Overview .....	23
2. Related work .....	25
3. Methodology.....	27
3.1 Data Crawling .....	27
3.2 Preprocessing.....	32
3.2.1 Stock Data Preprocessing.....	32
3.2.2 Text Data Preprocessing.....	33
3.3 Literature Review .....	35
3.3.1 Sentiment Analysis.....	35
3.4 Model Description.....	36
3.3.1 Long-Short Term Memory (LSTM).....	36
3.3.2 Gated Recurrent Unit (GRU) .....	38
3.3.3 K-Nearest Neighbors (KNN).....	40
3.3.4 Prophet .....	41
3.3.5 Clustering .....	42
3.3.6 VADER Sentiment.....	43

3.3.7 TextBlob.....	44
3.3.8 Bidirectional Encoder Representations from Transformers (BERT) ....	45
3.3.9 Artificial Neural Network (ANN).....	47
3.3.10 Fourier Transform .....	48
4. Experiments .....	50
4.1 Experiment on Data Visualization .....	50
4.2 Experiments on Numerical Analysis.....	56
4.2.1 LSTM.....	56
4.2.2 GRU .....	59
4.2.3 KNN Classification .....	60
4.2.4 KNN Regression.....	63
4.2.5 Prophet .....	69
4.3 Experiments on Textural Analysis .....	73
4.3.1 TextBlob + Clustering.....	73
4.3.2 VADAR Sentiment + Clustering.....	76
4.3.3 ANN .....	77
4.3.4 BERT .....	82
4.4 Experiments on Merged Model .....	84
4.4.1 KNN + VADAR Sentiment.....	84
4.4.2 LSTM + BERT.....	87
4.5 Pattern Recognition .....	92
4.5.1 Hard-coded Pattern Recognizer .....	92
4.5.2 Dataset .....	99
4.5.3 LSTM Model Experiment Result.....	100
4.6 YouTube comment and Stock price.....	102
4.6.1 Data Acquisition .....	102
4.6.2 Experiment Result .....	103
4.7 Twitter tweets and Stock price .....	107
4.7.1 Data acquisition .....	107
4.7.2 Experiment result (IA) .....	107
4.7.3 Experiment result (twint) .....	109
4.8 Reddit comment and Stock price.....	115
4.8.1 Data acquisition .....	115

4.8.2 Experiment result.....	115
5. Tool development .....	122
5.1 Tool Overview.....	122
5.2 Stock Pattern recognition.....	122
5.3 Stock Prediction .....	124
5.4 YouTube Sentiment Analysis .....	124
5.5 Twitter Sentiment Analysis .....	126
5.6 Reddit Sentiment Analysis .....	129
5.6 Backtest.....	131
6. Conclusion .....	136
References.....	138

# Table of Figure

Figure 1: Candlesticks <a href="https://www.investopedia.com/articles/active-trading/062315/using-bullish-candlestick-patterns-buy-stocks.asp">https://www.investopedia.com/articles/active-trading/062315/using-bullish-candlestick-patterns-buy-stocks.asp</a> ...	18
Figure 2: A candlestick in hammer form .....	19
Figure 3: A shooting star candlestick .....	19
Figure 4: Bullish engulfing candle .....	20
Figure 5: Doji .....	20
Figure 6: Double bottom .....	21
Figure 7: Sample of recent news in MarketWatch (1) .....	28
Figure 8: Sample of recent news in MarketWatch (2) .....	28
Figure 9: API request limit of New York Time .....	29
Figure 10: RNN layer structure .....	37
Figure 11: LSTM layer structure .....	37
Figure 12: Structure of a GRU unit .....	38
Figure 13: Equations of gates .....	39
Figure 14: Experiment result of comparing RNN, GRU, and LSTM .....	39
Figure 15: Geometrical meaning of KNN .....	40
Figure 16: An example of KNN regression ( <a href="https://www.jeremyjordan.me/k-nearest-neighbors/">https://www.jeremyjordan.me/k-nearest-neighbors/</a> ) .....	41
Figure 17: Example of periodic change .....	42
Figure 18 lexicon structure [33] .....	44
Figure 19 methods and process approach overview [34] .....	44
Figure 20: Graph explanation of "Bi-direction" in ELMO <a href="http://jalammar.github.io/illustrated-bert/">http://jalammar.github.io/illustrated-bert/</a> .....	46
Figure 21: Model comparison .....	47
Figure 22: An example of ANN .....	48
Figure 23 illustration of fourier transform .....	49
Figure 24: 3D graph of 3 lengths of candlestick .....	50
Figure 25: Focusing on upper and lower shadow length .....	51
Figure 26: Focusing on lower shadow and body length .....	51
Figure 27: Focusing on upper shadow and body length .....	52

Figure 28: 2D graph, Net shadow length vs body length .....	53
Figure 29: 3D graph of open, close, and net shadow length .....	54
Figure 30: 3D graph of open, close, and upper shadow length .....	54
Figure 31: 3D graph of open, close, and lower shadow length .....	55
Figure 32: Sample of input (normalized) .....	56
Figure 33: Architecture of the network .....	57
Figure 34: The prediction of the test set.....	58
Figure 35: Prediction of the final model .....	58
Figure 36: GRU model prediction.....	60
Figure 37: Sample of inputs of KNN classification model .....	61
Figure 38: KNN classification, $K = 3$ .....	61
Figure 39: KNN classification, $K = 4$ .....	61
Figure 40: KNN classification, $K = 5$ .....	62
Figure 41: KNN classification, $K = 6$ .....	62
Figure 42: KNN classification, $K = 7$ .....	62
Figure 43: KNN classification, $K = 8$ .....	62
Figure 44: Accuracy of different K values .....	63
Figure 45: Balanced accuracy of different K values .....	63
Figure 46: Sample of a training set .....	64
Figure 47: First prediction of KNN regression model .....	64
Figure 48: The stock price record of AAPL from 2010 to 2020 .....	65
Figure 49: Pseudo code of shifting input feature.....	65
Figure 50: After shifting the input.....	66
Figure 51: Experiment result.....	66
Figure 52: Prediction graph with $k = 13$ .....	67
Figure 53: KNN regression show delay in negative trend .....	67
Figure 54: KNN regression show delay in positive trend .....	68
Figure 55: Performance of KNN when stock move steadily.....	68
Figure 56: KNN regression as a classifier .....	69
Figure 57: Graphs of three components of Prophet.....	70
Figure 58: Effect of the events .....	71
Figure 59: Prediction made by prophet .....	72
Figure 60: Actual movement of AAPL .....	72



Figure 61: Prophet predicting 2019 .....	73
Figure 62: polarity score by TextBlob .....	74
Figure 63: New York Time dataset with TextBlob .....	74
Figure 64: sentiment clustering from TextBlob.....	75
Figure 65: Jenks break result for TextBlob .....	75
Figure 66: polarity score by VADAR .....	76
Figure 67: New York Time dataset with VADAR.....	76
Figure 68: sentiment clustering from VADAR .....	77
Figure 69: Jenks break result for VADAR.....	77
Figure 70: ANN test model 1.....	78
Figure 71: ANN test model 2.....	78
Figure 72: ANN test model 3.....	79
Figure 73: ANN test model 4.....	79
Figure 74: ANN test model accuracy.....	80
Figure 75: ANN test model loss.....	80
Figure 76: ANN test model validation loss & accuracy .....	81
Figure 77: ANN final model .....	82
Figure 78: Input sample .....	82
Figure 79: Heatmap of confusion matrix .....	83
Figure 80: VADAR clustered, before taking mean .....	84
Figure 81: VADAR after taking mean.....	84
Figure 82: stock price data .....	85
Figure 83: stock price & VADAR sentiment data.....	85
Figure 84: confusion matrix w/ sentiment.....	86
Figure 85: confusion matrix w/o sentiment.....	86
Figure 86: KNN + VADAR overall performance .....	87
Figure 87: Prediction of the previous LSTM model.....	88
Figure 88: Input sample .....	89
Figure 89: Heatmap of a confusion matrix .....	90
Figure 90: Heatmap of the confusion matrix (sentiment set to 0) .....	91
Figure 91: Result comparison.....	91
Figure 92: Case studies .....	92
Figure 93: Example of locating local minima .....	93

Figure 94: Lines connecting .....	94
Figure 95: Example of fitting parabola .....	95
Figure 96: Remaining lines after fitting parabola.....	95
Figure 97: Lines remaining .....	96
Figure 98: Example of line which is not possible to be a base of a double bottom pattern .....	96
Figure 99: Qualified baseline .....	97
Figure 100: A double bottom pattern recognized by our program .....	98
Figure 101: Overview of pattern recognized from 2010 to 2019 in Apple stock history .....	98
Figure 102: Before scaling. Timespan is 64 days.....	99
Figure 103 After scaling. Timespan is 235 days .....	100
Figure 104: Average validation loss.....	101
Figure 105: Average accuracy in predicting validation set.....	101
Figure 106: Example of not all training was successful.....	102
Figure 107: Daily average sentiment vs stock price .....	104
Figure 108: Covariance of close and polarity .....	104
Figure 109: Change in sentiment vs change in stock price .....	105
Figure 110: Covariance of change in close and change in polarity.....	105
Figure 111: Yesterday sentiment vs stock price .....	106
Figure 112: Covariance of yesterday sentiment and stock price .....	106
Figure 113 [twitter] today polarity vs today stock price .....	108
Figure 114 [twitter] covariance & correlation of today's polarity vs today's stock price .....	108
Figure 115 [twitter] yester polarity vs today stock price .....	109
Figure 116 [twitter] covariance & correlation of yesterday polarity vs today stock price .....	109
Figure 117 [twitter] today polarity vs today stock price .....	111
Figure 118 [twitter] covariance & correlation of today polarity vs today stock price .....	111
Figure 119 [twitter] yesterday polarity vs today stock price .....	112
Figure 120 [twitter] covariance & correlation of yesterday polarity vs today stock price .....	112

Figure 121 [twitter] delta yesterday polarity vs delta stock price .....	113
Figure 122 [twitter] covariance & correlation of delta yesterday polarity vs delta stock price .....	113
Figure 123 [fourier polarity 10, price 5] Twitter sentiment vs stock price	114
Figure 124 [fourier polarity 10, price 5] Twitter covariance and correlation .....	114
Figure 125 [reddit] today polarity vs today stock price .....	116
Figure 126 [no fourier] Reddit covariance and correlation .....	116
Figure 127 [reddit] yesterday polarity vs today stock price.....	117
Figure 128 [reddit] covariance & correlation of yesterday polarity vs today stock price .....	117
Figure 129 [reddit] delta today polarity vs delta today stock price .....	118
Figure 130 [reddit] covariance & correlation of delta today polarity vs delta today stock price .....	118
Figure 131 [fourier 5] Reddit sentiment vs stock price.....	119
Figure 132 [fourier 10] Reddit sentiment vs stock price.....	119
Figure 133 [fourier 15] Reddit sentiment vs stock price.....	120
Figure 134 [fourier 20] Reddit sentiment vs stock price.....	120
Figure 135 [fourier 5] Reddit covariance and correlation.....	121
Figure 136 [fourier 10] Reddit covariance and correlation.....	121
Figure 137 [fourier 15] Reddit covariance and correlation.....	121
Figure 138 [fourier 20] Reddit covariance and correlation.....	122
Figure 139: Pattern recognition interface .....	123
Figure 140: Pattern recognition result .....	124
Figure 141: Prediction result.....	124
Figure 142: YouTube sentiment analysis interface .....	125
Figure 143: Result of YouTube sentiment page .....	126
Figure 144 UI of Twitter sentiment analysis .....	127
Figure 145 result of Twitter sentiment analysis.....	128
Figure 146 Example of multiple graph representation .....	129
Figure 147 UI of Reddit sentiment analysis .....	130
Figure 148 result of Reddit sentiment analysis.....	130
Figure 149 Example of multiple graph representation .....	131

Figure 150: Using stock prediction model in backtesting .....	132
Figure 151: Using Youtube sentiment in backtesting .....	133
Figure 152: All positives .....	134
Figure 153: At least three positives.....	134
Figure 154: At least two positives .....	135
Figure 155: At least one positive.....	135

# 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 that indirectly reflecting 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 this 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 an employee believes in the company's ability to make money in general.

Public emotion is another index that reflecting the opinions toward a company. By analyzing the massive data generated on YouTube, Twitter, and Reddit every day, we can obtain whether people are having a good feeling for a company. People having a good feeling for a company is beneficial to the stock growth, vice versa.

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 in the field of artificial intelligence, this technology has become more common and close to people. It has a wide range of application and is good at data analysis, which attracts 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 intelligence can develop so fast is also that data acquisition of data has become easy.

The topic of this final year project is prediction stock trends with historical data and news data using machine learning and deep learning. To be more specific, we are building a tool to help people to analyze a stock, AAPL. Including a model to classify whether the stock is going to increase or decrease in the next day and estimate. A pattern recognizer to identify stock patterns. As well as analyzing the public emotion on YouTube, Twitter, and Reddit. This chapter will give an introduction and a brief overview of this final year project.

## 1.1 Background

### 1.1.1 Deep Learning

The neural network was started by understanding how the neurons inside the brain can work together to solve problems. It found that based on the “all-or-none” characteristic of the neural operation and the relations among the neurons, they can perform the complex logical operation (McCulloch, W.S., Pitts, W., 1943). However, although people start the discussion on the neural network since the 1940 era, it experienced up and down and not widely used in the following decades. The

computational power of the 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 with 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 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, labelling 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. This news is 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 the 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 feeding to the model.



### 1.1.3 Relations between Twitter and Stock Market

If we say news affects people's feeling about a company, data in social media shows the exact feeling of the public toward a company. It is hard to tell the influence of news. There is no clue for us to estimate how many decisions were affected by a new. News that isn't be seen by the public will have no impact on the stock market even though it has strong sentiment. We need another tool that can directly show us the feeling of the public toward a company.

There are many technologies that help us to analyze the public emotion of social media. Such as the APIs provided by the social media itself. Opinion detector, which is an emotion corpus-based method, analyzing the emotion of sentience. Opinion Finder is a publicly available software package detecting emotional polarity [8]. Another research found that Although the relationship between the public mood and investment behaviour in short term is not yet determined as statistically significant, there is evidence showing the causation between the public emotion and the daily closing price. It also shows the possibility of using Twitter data to forecast the stock price, there is evidence showing time lag exists public emotion and the daily close price [9].

### 1.1.4 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.

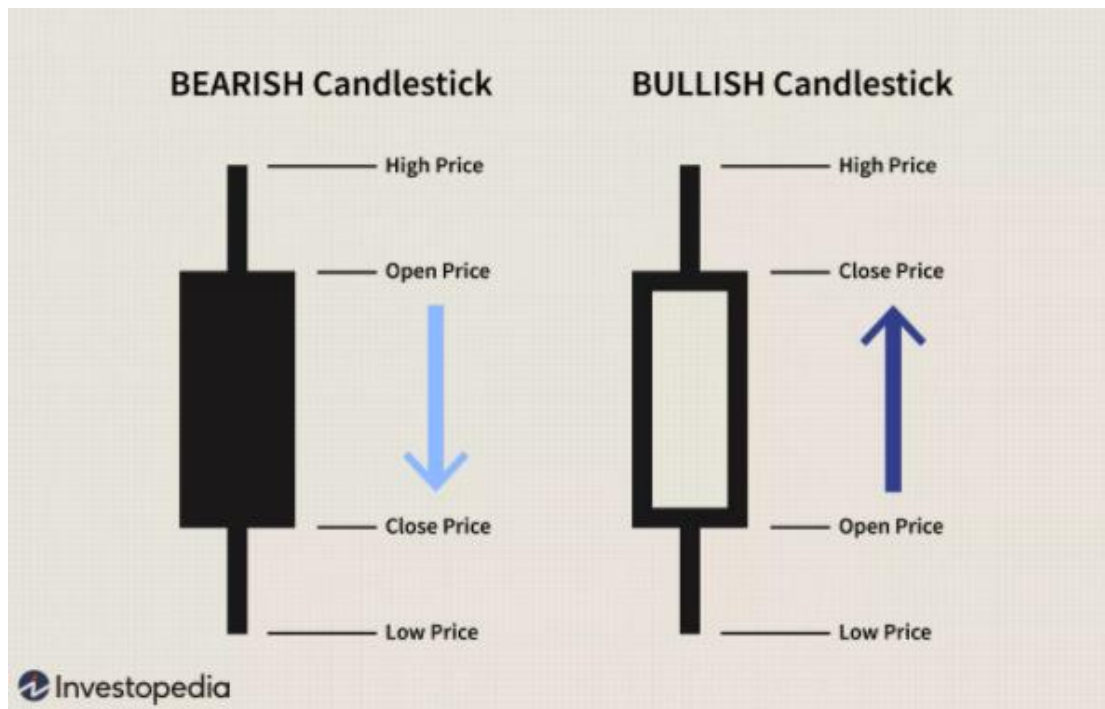


Figure 1: Candlesticks <https://www.investopedia.com/articles/active-trading/062315/using-bullish-candlestick-patterns-buy-stocks.asp>

There are two types of candlesticks, bearish candlestick meaning that the opening price is high than the closing price, while a bullish candlestick meaning that the closing price is high than the opening price. A candlestick chart may use a different type of candlestick to represent it is a bearish or a bullish candlestick, for example hollow candlesticks represent bullish, solid candlesticks represent bearish. Sometimes, some charts use different colours to represent whether it is increasing or decreasing. To study the chart, there are few terminologies, the real body is the thick bar with one end showing the opening price and one end showing the closing price; the upper shadow is the upper thin stick with the upper end pointing to the highest price of that date; the 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 doji. A candlestick with almost no upper shadow, a short body and a long lower shadow. Hammer indicates that initially supply is increasing, but then more buyers joined the market and made the price increase. It is a hint of an increasing trend.

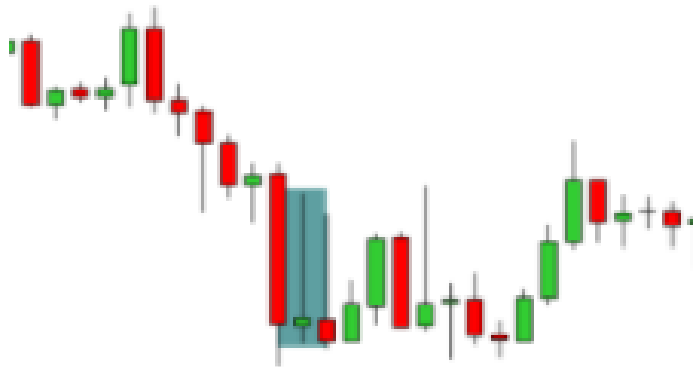


Figure 2: A candlestick in hammer form

Shooting star is a hammer-like formation that 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 be larger 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 to be positive.



Figure 4: Bullish engulfing candle

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

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.

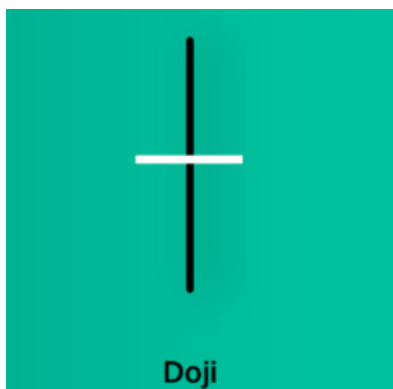


Figure 5: Doji

### 1.1.5 Double bottom pattern

In technical analysis, analysts can study the stock action patterns shown in the candlestick graph to decide whether to buy or not to buy the stock. A double bottom pattern describes a reverse trend, from decreasing to increasing. It is identified by a drop, a rebound, and another drop to the previous local minimum, ended by an increase. The double bottom pattern looks like a “W” letter. A good double bottom pattern will start with a drop of 10-20% percent, and the second bottom should fall into a range of 3-4% of the first bottom. The timespan between the two bottoms should be at least 3 months. The longer the duration, the higher the probability that the chart pattern will be successful [10].

A statistic on stock action pattern of 10 years of stock data and over 200,000 patterns which come from sixteen kinds of patterns [11]. The accuracy of the double bottom pattern is 75.55%. Which is the third one among sixteen patterns. The first and the second highest patterns are the bearish rectangle pattern and triple bottom pattern. The pattern is considered completed when it breaks through the highest price between the two bottoms. It then considered as successful when it covers a distance same as the distance between the double bottoms and the highest point between the double bottoms.

#### Double Bottom



Figure 6: Double bottom

## 1.2 Motivation

Many works have been done on stock prediction with historical stock data. We think the news is also an important factor to stock to stock volatility. And there are relatively fewer researches done on predicting stock with news, which is mainly conducted within the recent decade. We hope to use machine 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 their success rate of making money.

As there is research found that there is evidence showing that public emotion on Twitter can be related to stock price [9]. We think that data from other social media can also help us to analyze the stock trend. For example, YouTube and Reddit. We want to gather more information for the stock analysis in order to reduce the risk in stock investment.

## 1.3 Objective

In this project, we will analyse 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 experiments with the 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.

By gathering data from multiple social media to induce the public emotion and related the sentiment data to the stock price. Showing how likely the stock will rise if we found positive emotion from the social media. Finally, we will build a tool to let users obtain public emotion in social media in a given period.

## 1.4 Report Overview

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

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

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

In chapter 4, we describe the experiments done with different models on numerical analysis, news and social media sentiment analysis, pattern recognition. For each experiment, we will discuss the result and our finding.

In chapter 5, we will describe the investment tool we built in this project and the features that the tool has.

Chapter 6 will conclude our project.



## 2. Related work

There are many stock prediction kinds of research that have been done in the past. For example, predicting the stock market with Prophet upon ARIMA [12], provides steps of how to use Prophet to forecast the market. Or using an artificial neural network and random forest to perform the prediction [13], they showed that using an artificial neural network can give a better prediction than using a random forest.

Some researchers performed textual analysis on the news or social media to assist the stock prediction. For example, there is 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) [14]. 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. They conclude that MLP performed the best among all models and the oil rate is the most relevant feature to the performance of the closing price of KSE. There is a project that predicts the influence of news on 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 [15].

Besides news, social media also contains much useful information for stock prediction. This is research studied the relationship between microblogging and stock data. It shows that there is an improvement in synthesizing public emotion analysis in the process of predicting the stock market. This research extract how the stock market will behave and respond to external factors [16]. Two researchers find out that information on Twitter also affects the Dow Jones Industrial Average value. They analyse the sentiment of Twitter posts to obtain the public mood. They result in decent profit over 40 days [17]. The research found that Although the relationship between the public mood and investment behaviour in short term is not yet determined as statistically significant, there is evidence showing the causation between the public emotion and the daily closing price. It also shows the possibility of using Twitter data

to forecast the stock price, as there is evidence showing time lag exists between public emotion and the daily close price [9].

## 3. Methodology

### 3.1 Data Crawling

#### Stock Data

The way to obtain historical stock data is using a python package which is called Pandas-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 on 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 labelled with the sentiment classification with another column "sentiment". That makes the dataset good for training the sentiment analysis model. There are three types of sentiment, positive, neutral and negative. The second data contains the 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 is 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 provide us with 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 the 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 it can provide news which is related to Apple. The crawler we wrote will

execute once per day to collect the latest update in the recent news section.

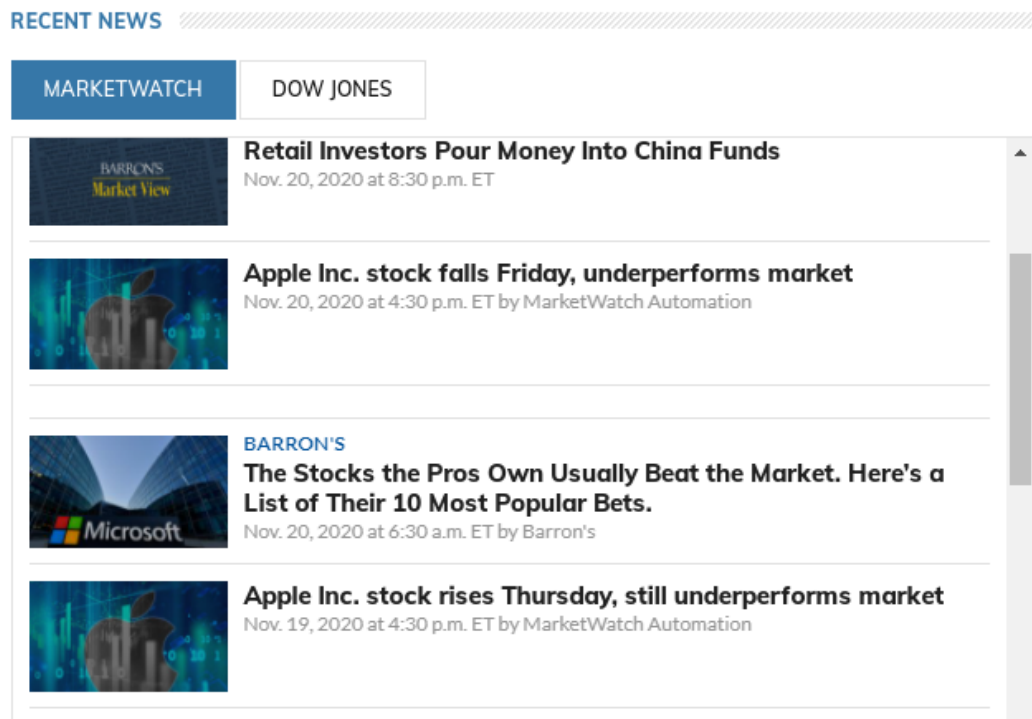


Figure 7: Sample of recent news in MarketWatch (1)

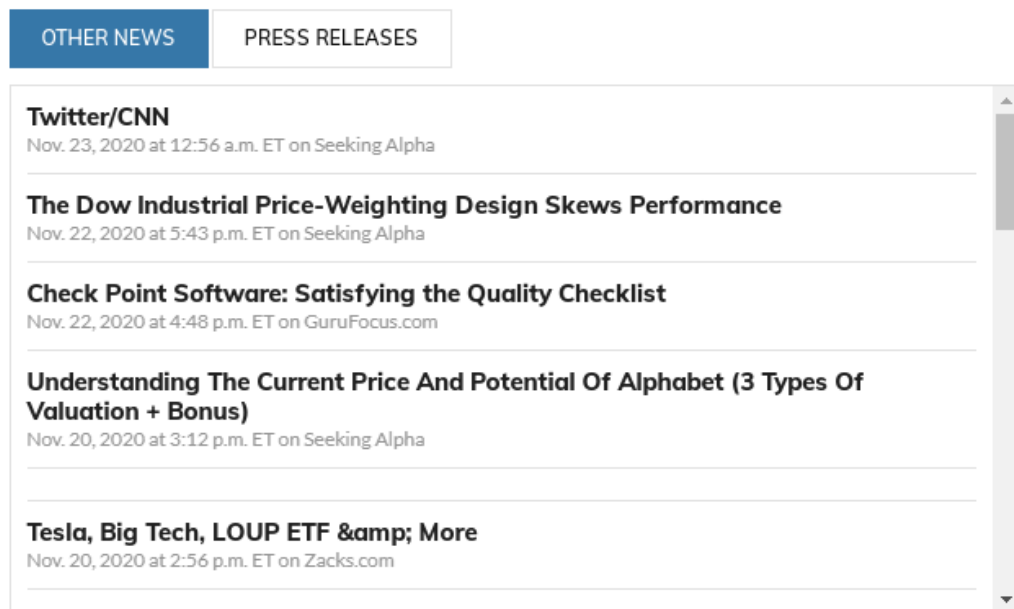


Figure 8: Sample of recent news in MarketWatch (2)

The last dataset we have was obtained from New York Time. They provide an API that allows everyone to download their news resource. However, it may have security and server stress considerations, we have a limit on requesting resources. We cannot issue more than 10 requests per minute and 4000 requests per day. Therefore, we need to write a program to automatically crawl articles with New York Time API. In the end, we get all the articles 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](mailto:code@nytimes.com).

Figure 9: API request limit of New York Time

## Social Media Data

Many large-scale social media platforms provide API for accessing a part of their data. As long as the data originally public accessible, these APIs provide a more convenient channel to download their data. In this project, we are gathering data from YouTube, Twitter, and Reddit.

The YouTube Data API allows us to get search responses, we are searching by the keyword “Apple”. We can filter the result by categories, we need the videos which are coming from the “Science and Technology” category so that the noise in the data has been significantly reduced for us before any data cleansing. After that, we are interested in the comments about the videos as we need to perform sentiment analysis. We use the API to get the comments in all videos in the previous search result. There are some limitations in using this API. Once we exceed the daily limit, we cannot use the API on that day, the quota will refresh on the next day. The second limitation of the API is that we cannot filter the comments by date, it always returns the comments in the descending order of date, which makes it difficult to get the comment within a certain period.

Collecting Twitter data is somewhat an obstacle. Although Twitter provided an API that returns public tweets encoded in JSON format, there is a limitation on quotas that do not guarantee the completeness of the retrieved data. In other terms, we may not get all the related tweets that we need due to the quota limit. Therefore, we need to find another way to get the required data. There are several ways to get tweets from Twitter, one is through its official API, the second is using a third-party library such as twint and tweepy, and lastly using third-party group's pre-compiled data source such as Internet Archive Team Project [18] [19].

Internet Archive Team is a loose collective of rogue archivists, programmers, writers and loudmouths dedicated to saving the digital heritage. Its primary focus is the copying and preservation of content housed by at-risk online services [18]. Twitter is one of its archive projects. Since the official Twitter API has a straight limitation on the API quota, it is not enough for our project. Therefore we look at the Internet Archive Team Project, it has a well-structured database for tweets they collect and the data looks promising.

Although Archive Team collected good amounts of Twitter data, it lacks instantaneity and portability. The data that Internet Archive Team collect is discontinued, without the ability to the specific time interval, and too large in size. It is necessary to obtain a fast and convenient way to collect the data we want within a controllable time interval. Therefore our second approach to obtain Twitter data is through a third-party library. We use twint for its capability of bypassing the limitation of the Twitter API quota. Moreover, twint is capable to obtain real-time data due to the nature of scrapping. By configuring twint, we can get a specific dataset we want to test in any time interval.

Regarding Reddit data collection, Reddit itself also provided an official API for people to collect data. However similar to Twitter, it is foreseeable that the API will certainly have some limitations. In this project, we will use pushshift.io to target the specific submission (thread) we want, and then use PRAW to extract all comments in the corresponding submission. PRAW is a Reddit API wrapper for its organized return data and easy to use characteristics [19].

## 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
  - Features: date, headline, abstract, leading paragraph, URL
  - Size: 29084

- YouTube comments:
  - Period: From January 1, 2010 to June 30, 2019
  - Features: Comment
  - Size: 225011
  
- Twitter tweets (Internet Archive Team Project)
  - Period: From January 2019 to June 2020
  - Features: created\_at, final-text, final-hashtag, final-lang, user\_screen\_name, user\_location, followers\_count, verified
  - Size: ~128GB
  
- Twitter tweets (twint)
  - Period: customizable, From January 2019 to June 2019
  - Feature: same as Internet Archive Team Project
  - Size: 233217
  
- Reddit comment
  - Period: customizable, From September 1, 2018 to September 1, 2019
  - Feature: date, author, body, url
  - Size: 123857

## 3.2 Preprocessing

### 3.2.1 Stock Data Preprocessing

#### Labelling

For stock data, we add a new column labelling 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 labelling for the dataset, we then normalize the dataset in order to reduce the training cost [20].

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

where  $X$  is a vector of the 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  $\text{man}(X) - \text{min}(X)$  might be replaced by  $\text{std}(X)$  which is the standard deviation of  $X$ , in the coming experiment.

## Stock Pattern Recognizer

As we cannot found a large dataset either labelled or not labelled for stock pattern recognition, we need to prepare our own dataset by writing a rule-based pattern recognizer with the criteria of the double bottom pattern.

### 3.2.2 Text Data Preprocessing

#### Digitization

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

## Removing Reply tags

In YouTube comments, users starting with a username tag to reply to other people's comment. The pattern of the tags is "@username", as there might be spaces in the name, we cannot simply remove the first word if it is started with the "@" sign. We need to collect all the usernames in the same comment thread to construct a set. And then remove the first few words in the comments if appear in the set.

## Removing URLs

URLs are common in the comments. They affect the sentiment of that sentence, we need to remove them in advance in order to perform sentiment analysis. The pattern of URLs is simple, they start with "http" and no space in the URL string. We can remove them simply by a regular expression "http\S+".

## Removing Emojis

Emojis often appear in the comment. As we are only interested in analyzing the English comment. We take advantage of this, we convert the comments to ASCII format, simply ignore those characters if they cannot be converted to ASCII. After the conversion, all the emojis will be removed.

## 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 [21]. A sentiment analysis system (also known as opinion mining or emotion AI) uses natural language processing (NLP) and machine learning to identifies, extracts and classify subjective information from source materials [22]. It usually assigns weighted sentiment scores to a word, sentence, paragraph or whole document [23].

There are 3 main types of sentiment analysis:

1. Rule-based: system performs sentiment analysis based on a set of manually crafted rules [23].
2. Automatic: system learns the data by itself using machine learning techniques [23].
3. Hybrid: system combines rule-based and automatic approaches [23].

A rule-based approach is a set of rules 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 [23]. 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 a greater number determines the polarity of the sentence, or neutral if they are even.

The automatic approach taking the advantage of machine learning, it does not rely on manually crafted rules, but on various machines learning techniques/algorithms to achieve sentiment analysis. A machine learning algorithm usually involves two parts of the process, training and prediction. In the training process, a model is generated by associating an input(text) to the corresponding output(tag) based on the training

samples. In the prediction process, unseen input is extracted by the 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 a lot of machine learning algorithms that suitable for sentiment analysis, like Long-Short Term Memory, BERT, etc.

The hybrid approach combines 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 [23].

## 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 learns 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 serve as the output of the unit for the next unit of the same later or for the next layer of the neural network [24].

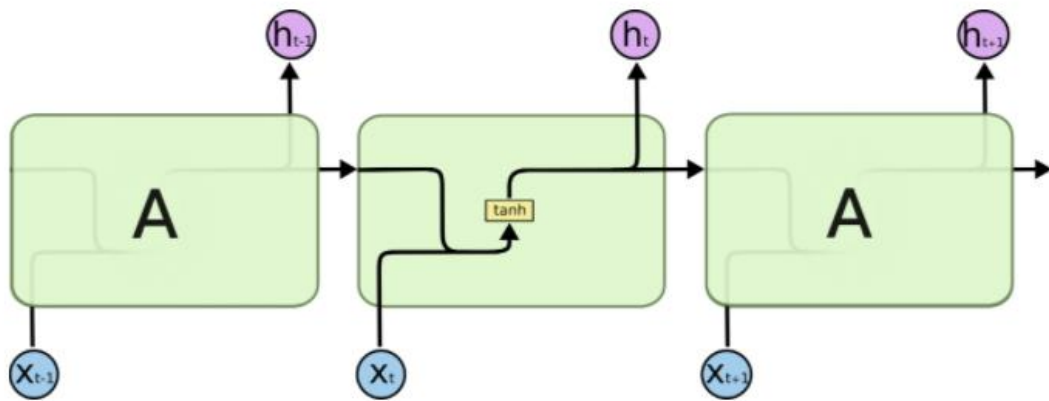


Figure 10: RNN layer structure

LSTM is 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 gate 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 [24].

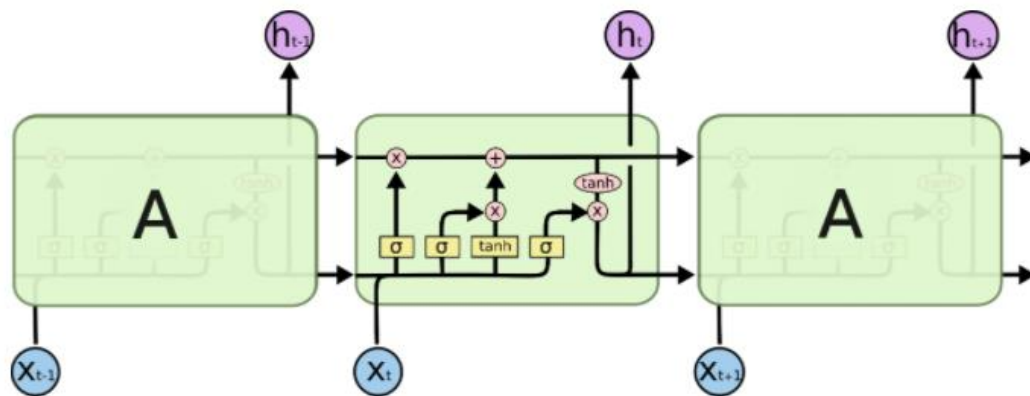


Figure 11: LSTM layer structure

However, LSTM is not a completely perfect model, nothing is. There are many variants of LSTM was proposed 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 the infinite growth of memory state value [25].

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. [26]

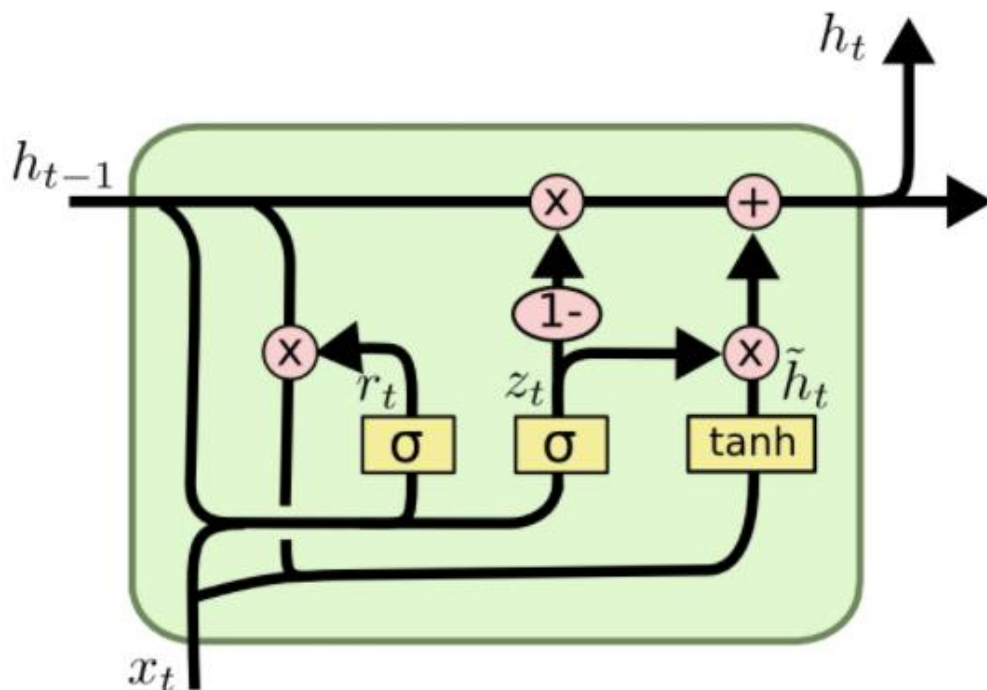


Figure 12: Structure of a GRU unit

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. [24]

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

Figure 13: Equations of gates

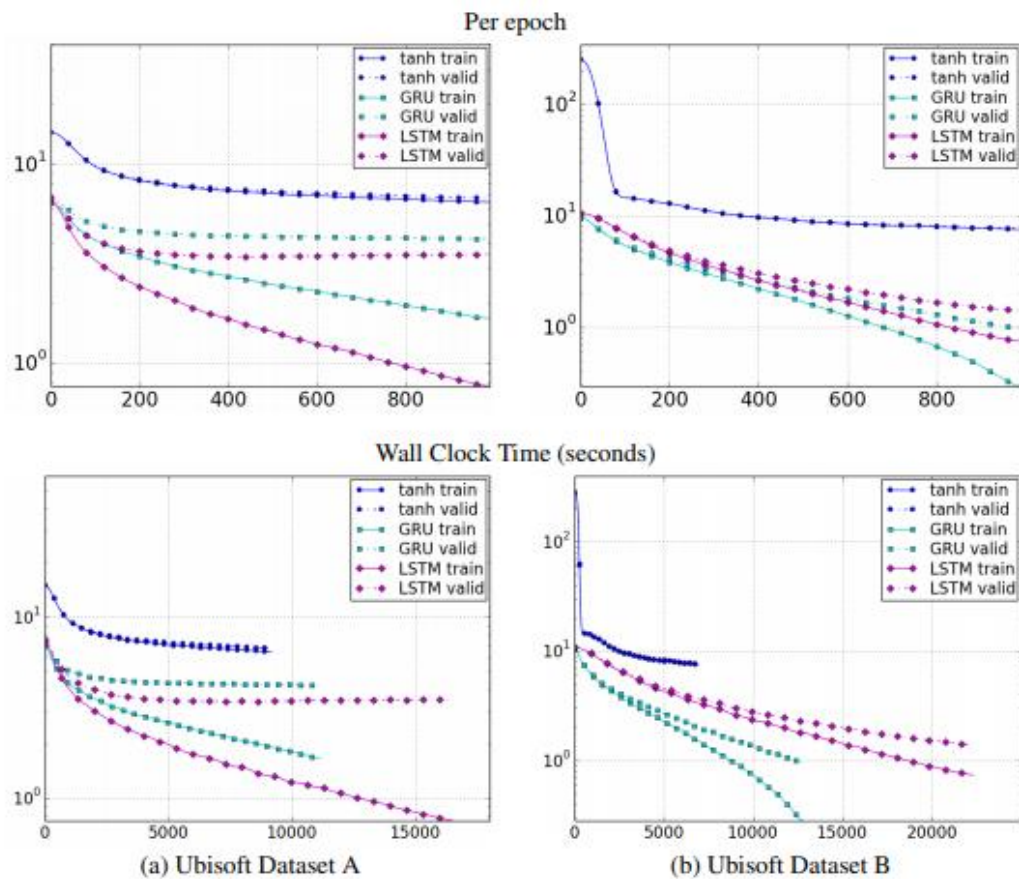


Figure 3: Learning curves for training and validation sets of different types of units with respect to (top) the number of iterations and (bottom) the wall clock time. x-axis is the number of epochs and y-axis corresponds to the negative-log likelihood of the model shown in log-scale.

Figure 14: 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. [26]

### 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 [27]. There are many variants of the classification process, such as classifying the unclassified sample points with the distance of its K nearest neighbors weights or simply classifying it into the class that occurs the most in the K nearest neighbors.

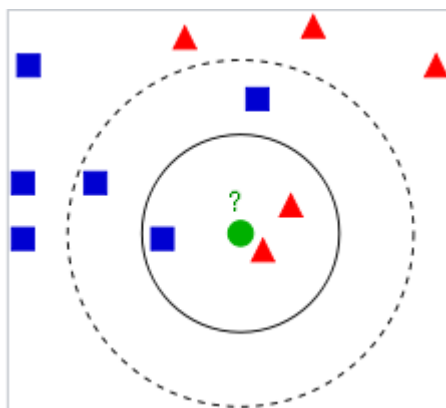


Figure 15: Geometrical meaning of KNN

KNN regression is similar to classification, but it gives a prediction of a continuous value



instead of a class. For a sample 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 weigh those neighbors concerning their distance to the sample point, for example, the closest 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.

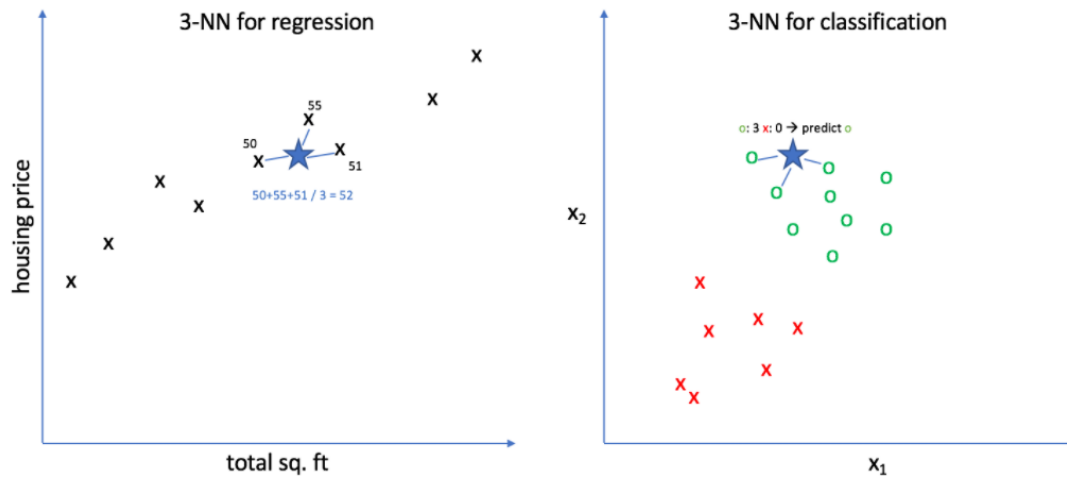


Figure 16: An example of KNN regression (<https://www.jeremyjordan.me/k-nearest-neighbors/>)

### 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 performs time-series data prediction with an addition model. It has good performance on the time-series data which have a 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 [28].

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

For the trend function,  $g(t)$ , it models the non-periodic trend of the time-series data set.  $s(t)$  represents 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.  $h(t)$  represents the holiday effect, as data of the holiday may look different from another 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 [28].

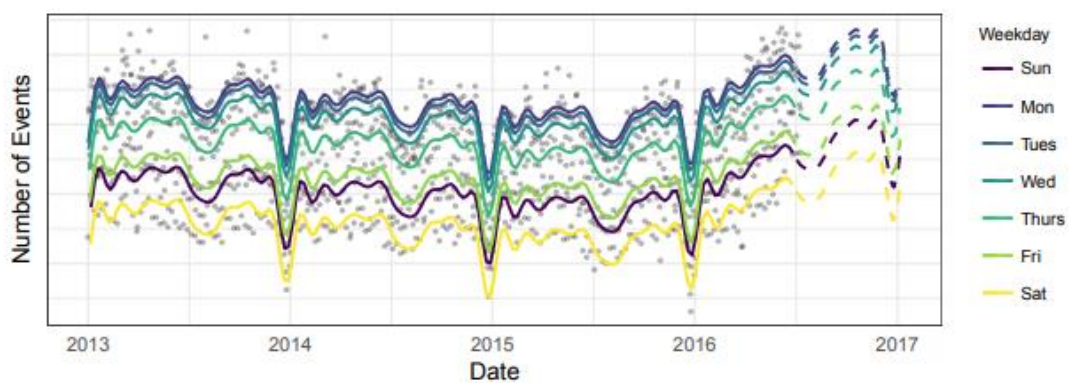


Figure 17: Example of periodic change

### 3.3.5 Clustering

Clustering is 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 [29]. In short, clustering is a task that groups data points in such a way that the data points in the same group should have similar properties or features, while data points in different groups should have no similarity between [30].

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

K-Means Clustering is the simplest clustering algorithm that works by partitioning data points into  $k$  clusters where each data point 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 analyze hierarchical data to build a hierarchical cluster. An Agglomerative Clustering means it is a bottom-up algorithm. It builds the tree by recursively merging the similar data points and similar clusters until all of the clusters/data points become 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 [31]. VADER is able to determine the polarity of the sentiment of a given text when the data being analysed is not labelled. Moreover, VADER 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" [32]. 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 18 lexicon structure [33]

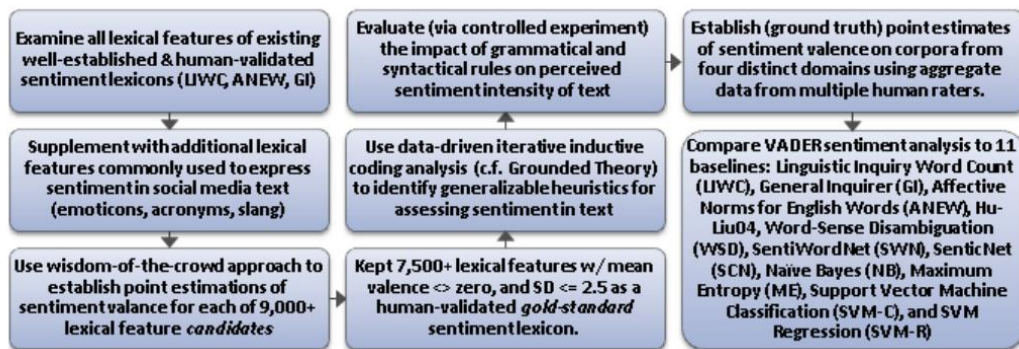


Figure 19 methods and process approach overview [34]

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 the learning stage. These preprocess such as tokenization, stemming, lemmatization is 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 “?????”, 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 [35]. 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 learning 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 this 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 into two executions, first time it reads from left to right, and then it reads from right to left. In the end, it put two results together to do the final prediction [36].

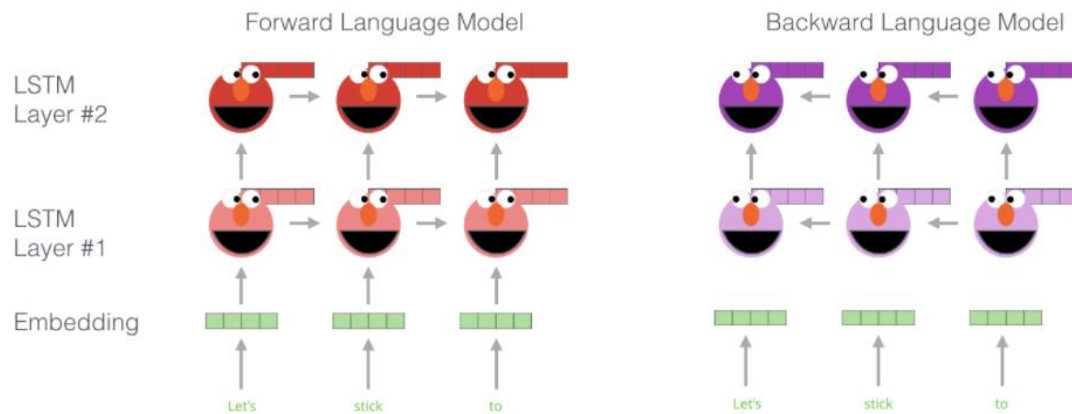


Figure 20: Graph explanation of "Bi-direction" in ELMO <http://jalammar.github.io/illustrated-bert/>

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 [37].

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

The way ELMO do this is by 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 outputs its answer. Even it analyzes the sentence from two directions, 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.

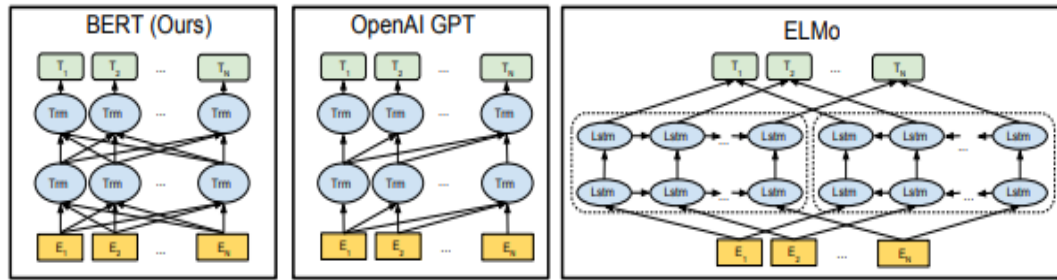


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT 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 21: Model comparison

### 3.3.9 Artificial Neural Network (ANN)

The artificial neural network is a model of the biological neural network, which is inspired by the biological neural networks that constitute animal brains. In ANN, perceptron or units are grouped as a 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 serve as the input of the next layer so that the information only flows 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. [38]

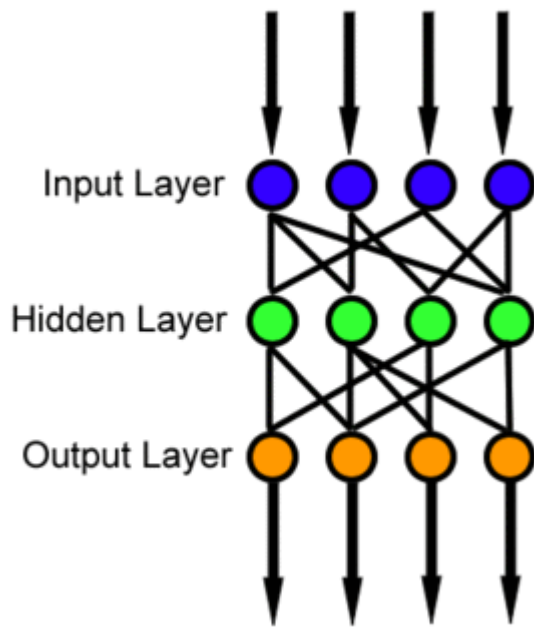


Figure 22: An example of ANN

### 3.3.10 Fourier Transform

In mathematical terms, The Fourier Transform is a technique that decomposes signal depending on space or time into functions depending on spatial or temporal frequency [39]. In short, it is a mathematical technique that transforms a function of time,  $f(t)$ , to a function of frequency,  $f(w)$ .

The equation of fourier transform and inverse fourier transform are below.

Equation 1 fourier transform

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx,$$

Equation 2 inverse fourier transform

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i x \xi} d\xi,$$

Below is a short illustration of fourier transform.



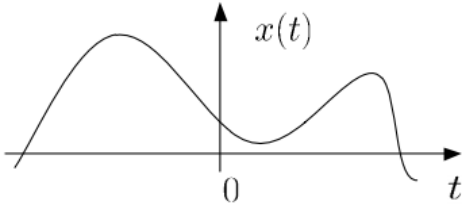
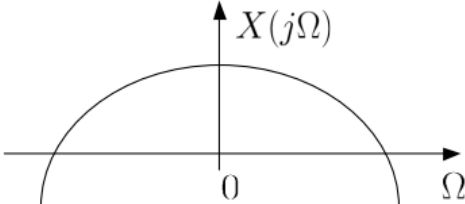
time domain	frequency domain
 $X(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt \Rightarrow$ $\Leftarrow x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t} d\Omega$	
continuous and aperiodic	continuous and aperiodic

Figure 23 illustration of fourier transform

Fourier transform is widely used not only in signal processing, like radio, acoustic, etc. It is also frequently used in image analysis, for example edge detection, image filtering, image reconstruction, and image compression [40].

## 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 may be 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 the body as the three axes. Then we further mark those point as green or red, indicating that the stock will rise on the next trading day with the combination of three axes.

Visualization result:

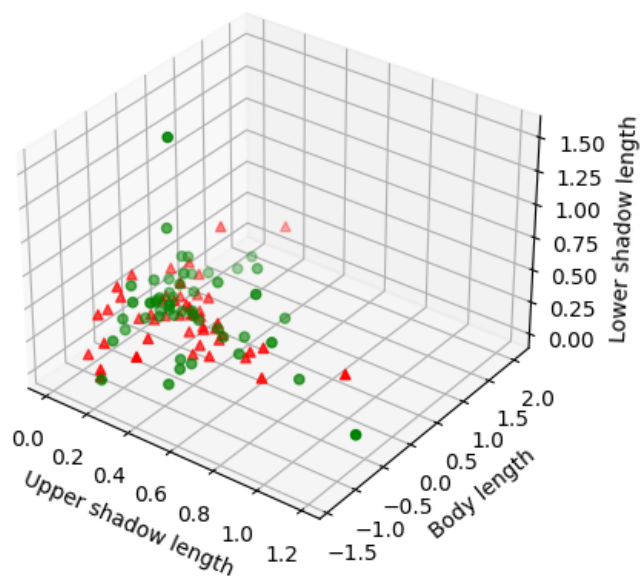


Figure 24: 3D graph of 3 lengths of candlestick

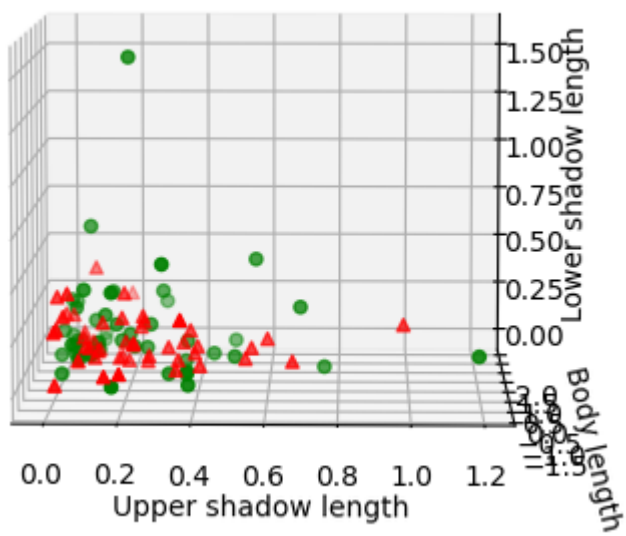


Figure 25: Focusing on upper and lower shadow length

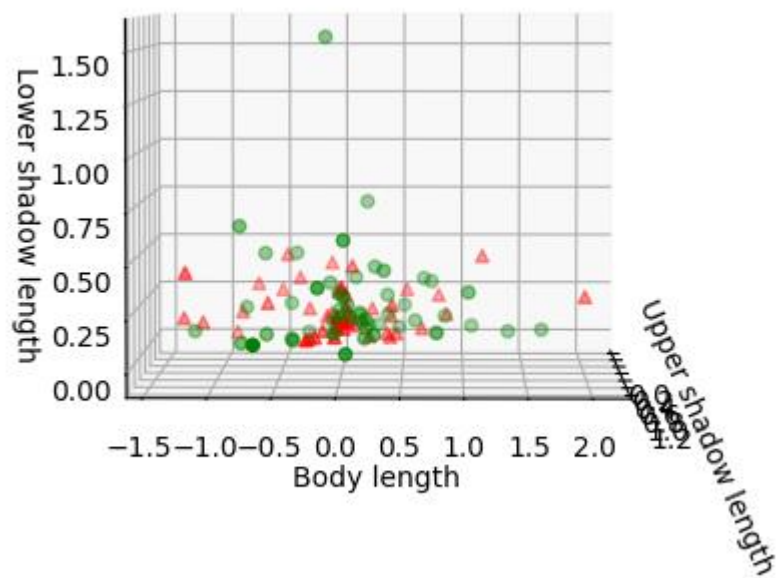


Figure 26: Focusing on lower shadow and body length

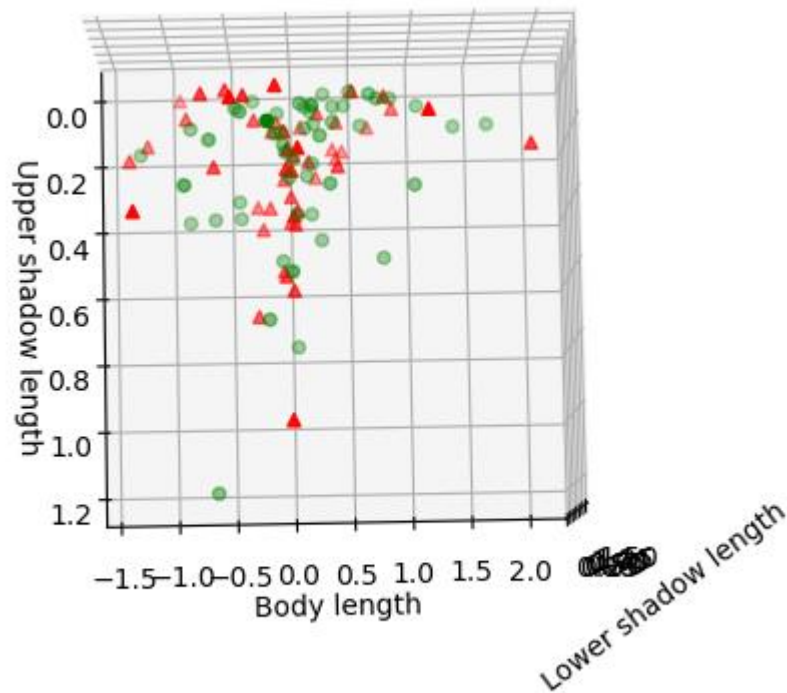


Figure 27: Focusing on upper shadow and body length

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.

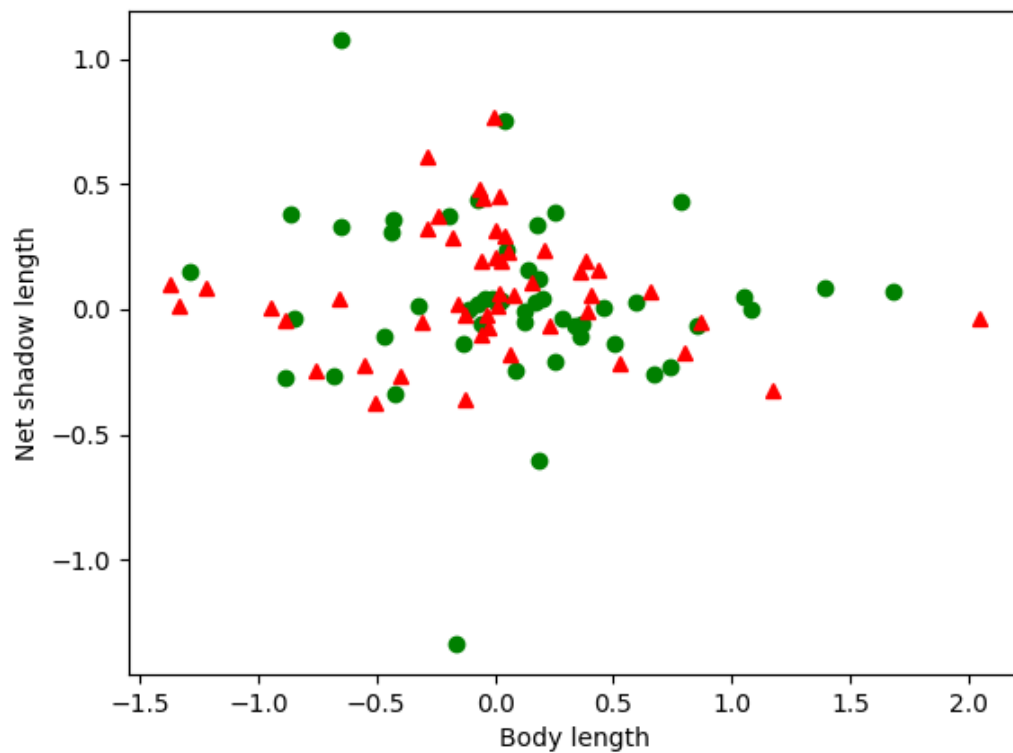


Figure 28: 2D graph, Net shadow length vs body length

The result is also showing us there are not two clear groups that contain only green dots or red triangles. They all concentrated together.

Then, our final test was plotting a 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.

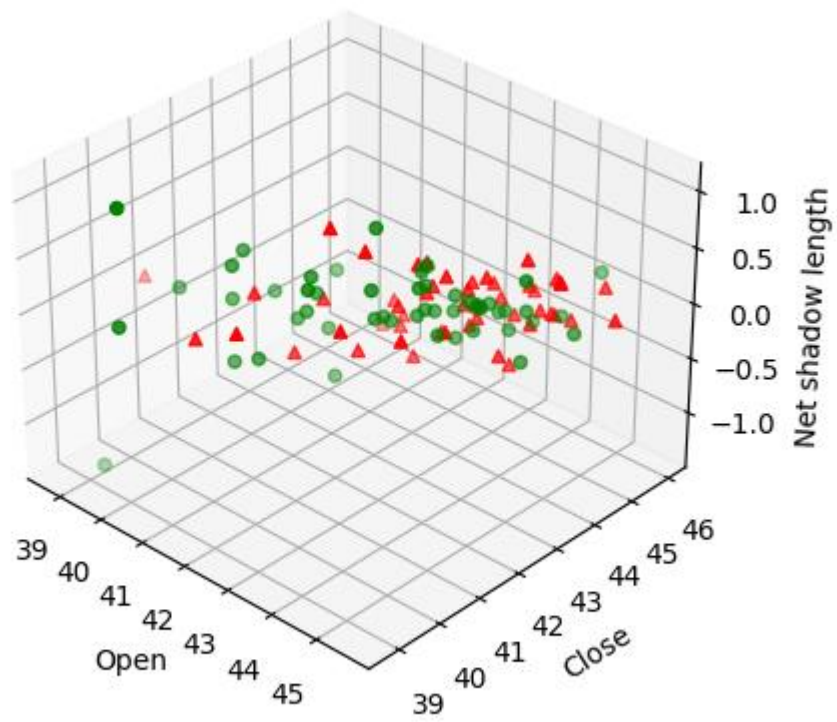


Figure 29: 3D graph of open, close, and net shadow length

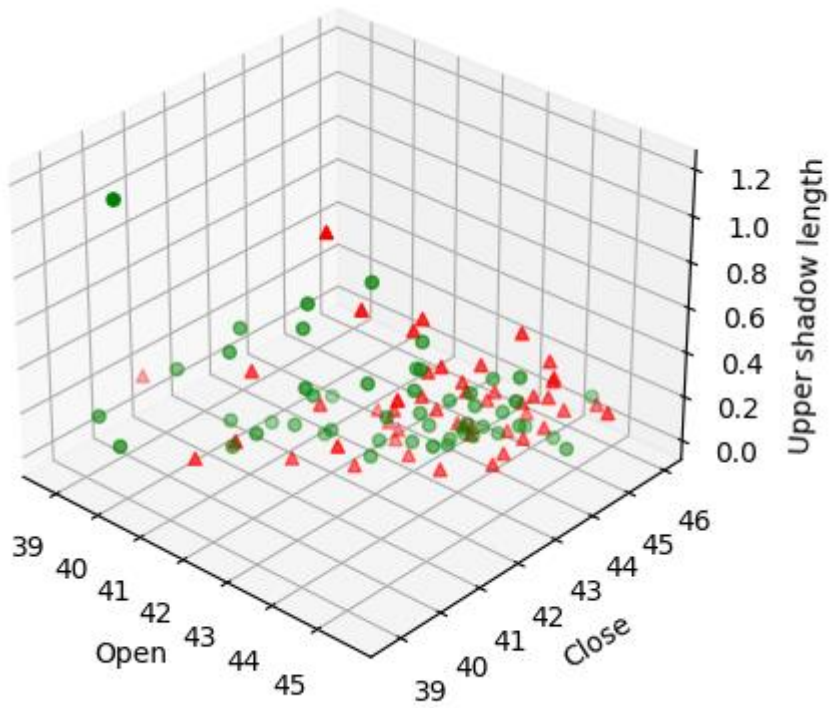


Figure 30: 3D graph of open, close, and upper shadow length

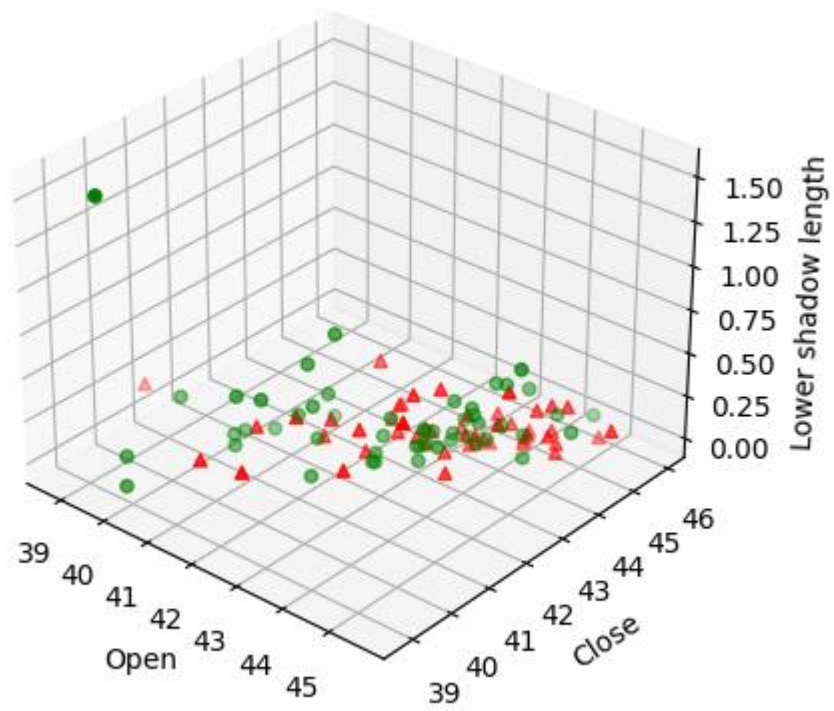


Figure 31: 3D graph of open, close, and lower shadow length

## 4.2 Experiments on Numerical Analysis

### 4.2.1 LSTM

Our first model built is LSTM, using 22 days of data to predict the closing price of the 23-the 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  $22 \times 9 = 198$  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 32: Sample of input (normalized)



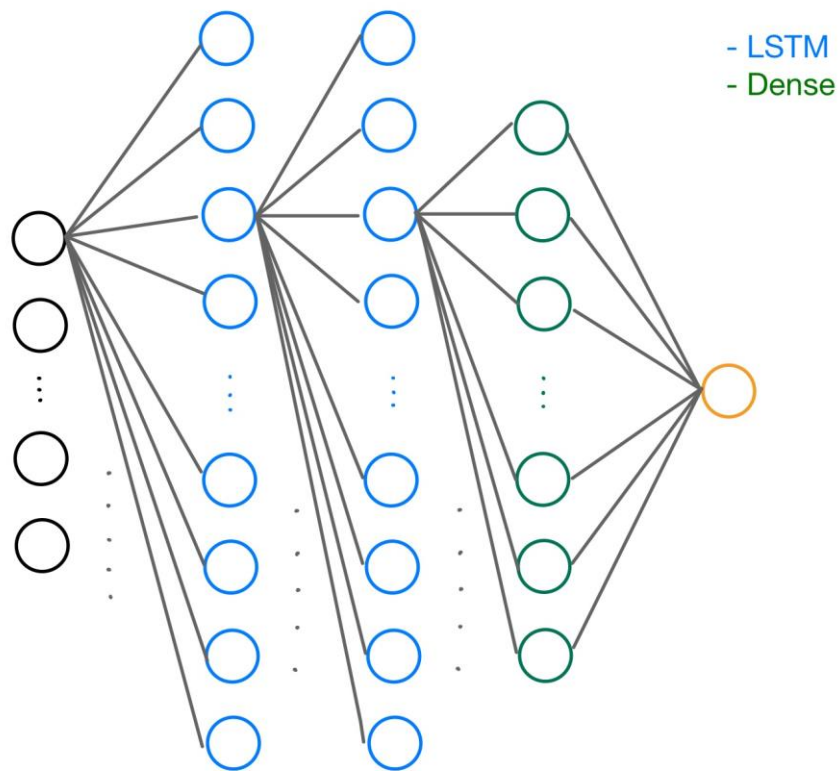


Figure 33: Architecture of the network

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

The 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.

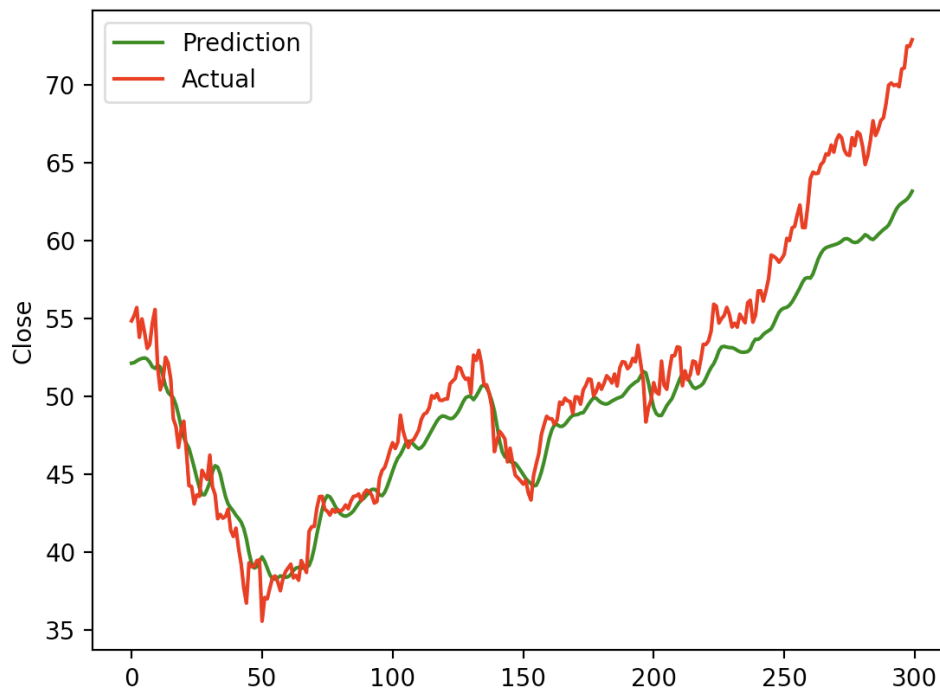


Figure 34: The prediction of the test set

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.

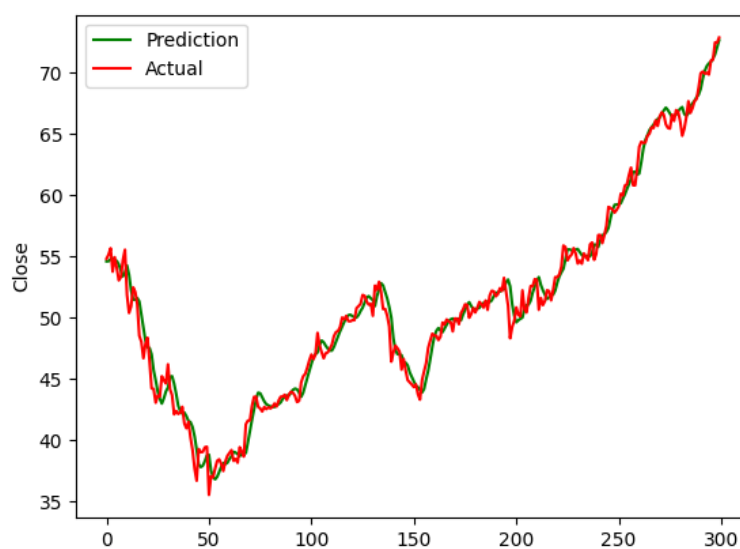


Figure 35: Prediction of the final model

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. The 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 worse 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-normalised MSE result of the testing set is 1.17354401.

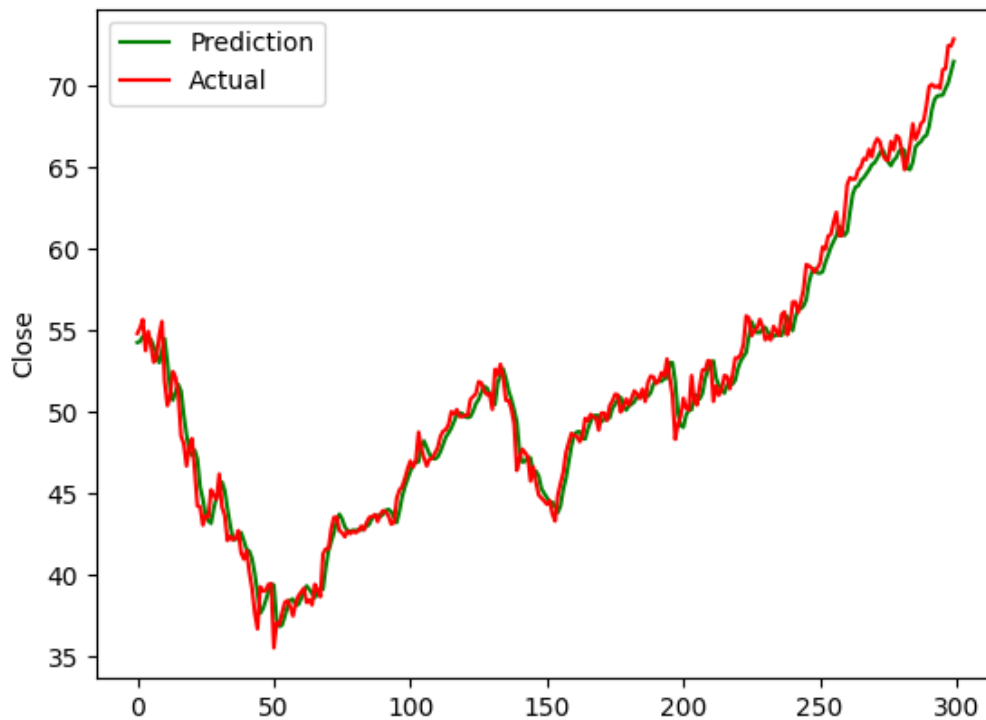


Figure 36: GRU model prediction

This model also suffers from the shifting problem that 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-rise” 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 37: 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 38: KNN classification, K = 3

```

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

```

Figure 39: KNN classification, K = 4

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

Figure 40: KNN classification, K = 5

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

Figure 41: KNN classification, K = 6

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

Figure 42: KNN classification, K = 7

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

Figure 43: 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 44: 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 45: Balanced accuracy of different K values

For either accuracy or balanced accuracy, K of value 5 has 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 the training set is 2256, size of the 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
<b>0</b>	7.619643	7.520000	7.611786	7.526072	352410800.0	6.503574	7.643214
<b>1</b>	7.660714	7.585000	7.622500	7.643214	493729600.0	6.604801	7.656428
<b>2</b>	7.699643	7.616071	7.664286	7.656428	601904800.0	6.616219	7.534643
<b>3</b>	7.686786	7.526786	7.656428	7.534643	552160000.0	6.510980	7.520714
<b>4</b>	7.571429	7.466072	7.562500	7.520714	477131200.0	6.498945	7.570714
...	...	...	...	...	...	...	...
<b>2510</b>	70.662498	69.639999	70.557503	69.860001	275978000.0	69.381348	71.000000
<b>2511</b>	71.062500	70.092499	70.132500	71.000000	98572000.0	70.513535	71.067497
<b>2512</b>	71.222504	70.730003	71.172501	71.067497	48478800.0	70.580566	72.477501
<b>2513</b>	72.495003	71.175003	71.205002	72.477501	93121200.0	71.980911	72.449997
<b>2514</b>	73.492500	72.029999	72.779999	72.449997	146266000.0	71.953598	72.879997

Figure 46: Sample of a training set

Initially, we set K to be 10 to perform the prediction. In 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.

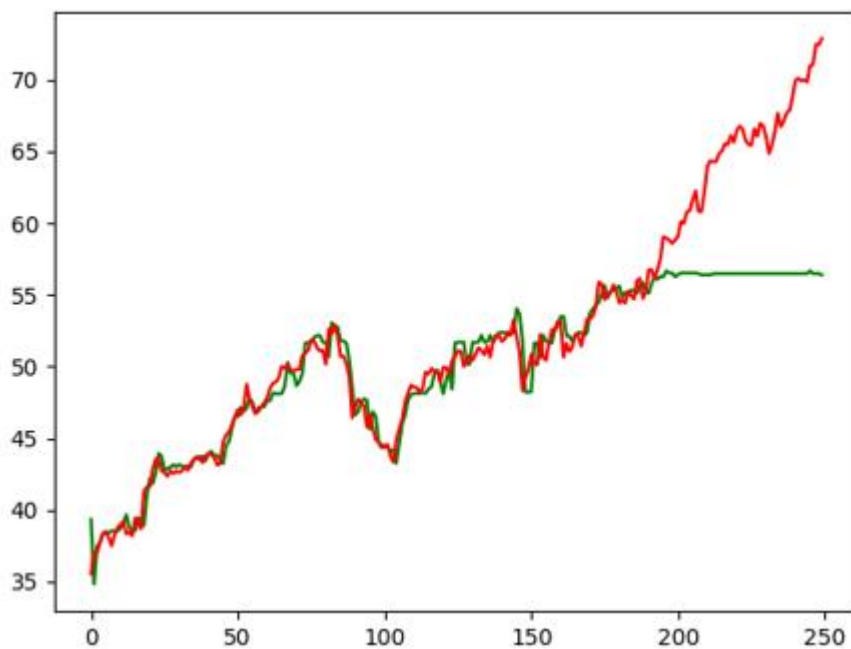


Figure 47: First prediction of KNN regression model



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 increase. 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 than 56.9.



Figure 48: 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.

```

1  limit <- maxClose(TrainSet)
2
3  for each day in TestSet
4      shiftValue <- 0
5
6      if day.close >= limit*0.8
7          shiftValue <- day.close - limit*0.8
8          day.open -= shiftValue
9          day.close -= shiftValue
10         day.high -= shiftValue
11         day.low -= shiftValue
12         day.adjClose -= shiftValue
13         day.volumn -= shiftValue
14
15     prediction = knnrPrediction(day)
16     if day.close >= limit*0.8
17         prediction += shiftValue
18

```

Figure 49: 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.

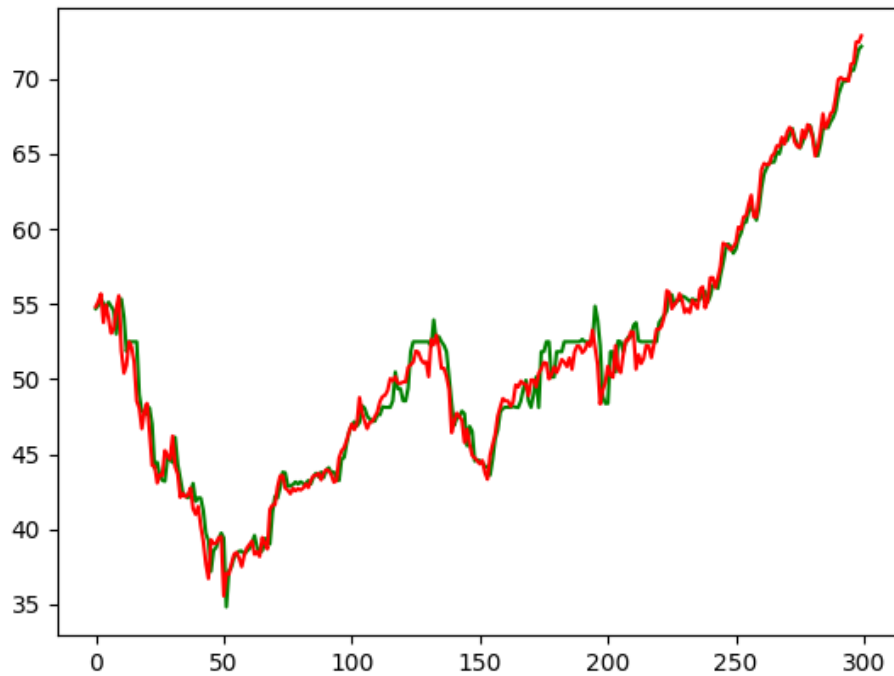


Figure 50: After shifting the input

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 51: Experiment result

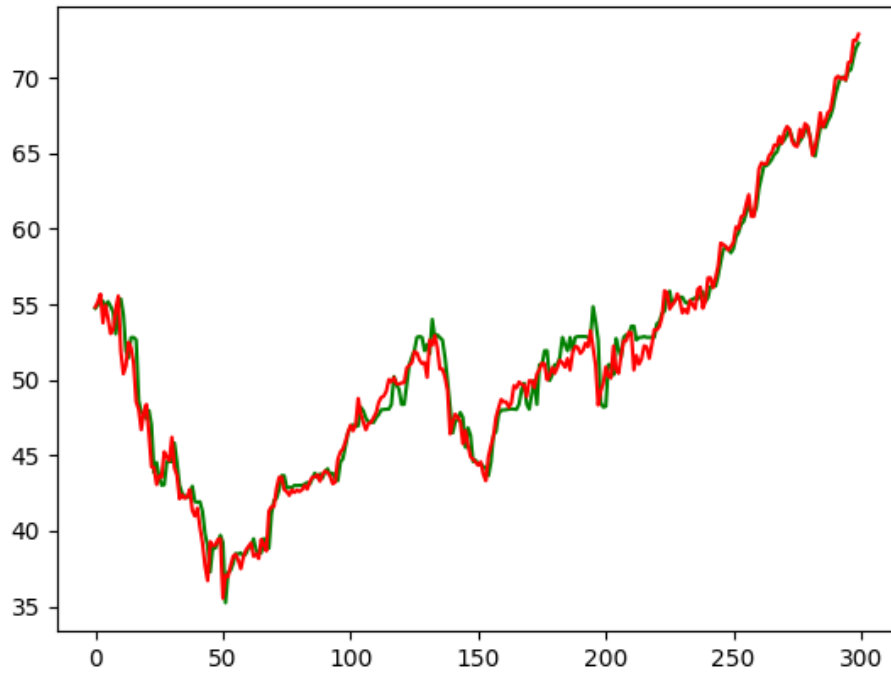


Figure 52: Prediction graph with  $k = 13$

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



Figure 53: KNN regression show delay in negative trend

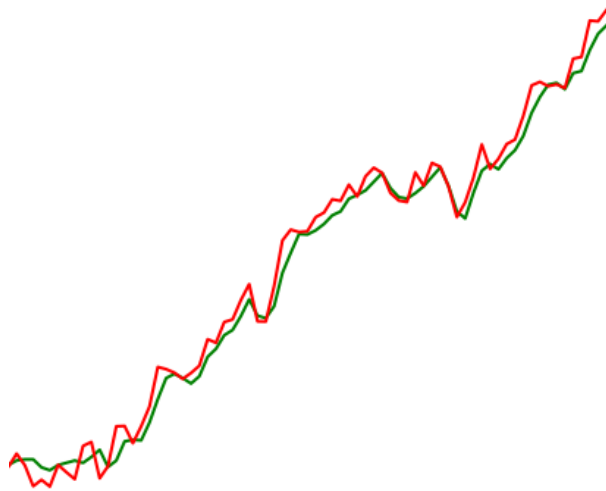


Figure 54: KNN regression show delay in positive trend

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.

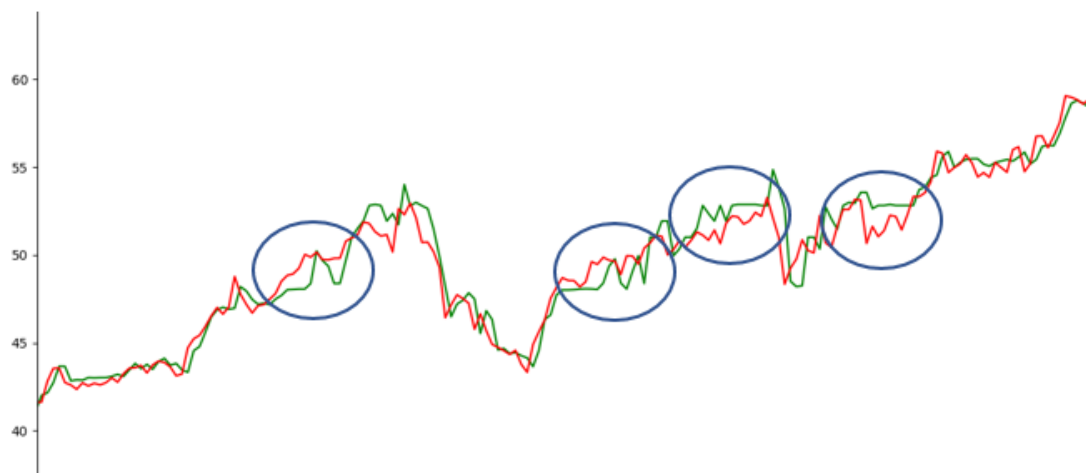


Figure 55: Performance of KNN when stock move steadily

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  $(\widehat{Close}_{t+1} > Close_t \text{ and } Close_{t+1} > Close_t)$   
or  $(\widehat{Close}_{t+1} \leq Close_t \text{ and } Close_{t+1} \leq Close_t)$

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

We will say the classification is correct if and only if both predict the closing price of day  $t + 1$  and the actual closing price of day  $t + 1$  are higher than 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 56: 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 them 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,  $y(t) = g(t) + s(t) + h(t) + \epsilon_t$ , these functions represent the non-periodic, periodic, and holiday changes. The following graphs show the trend learned by Prophet from the training set. In term of week, the third sub-graph shows that AAPL rise on Tuesday in general. In term of the month, the fourth sub-graph shows that AAPL

overall rise in September to November.

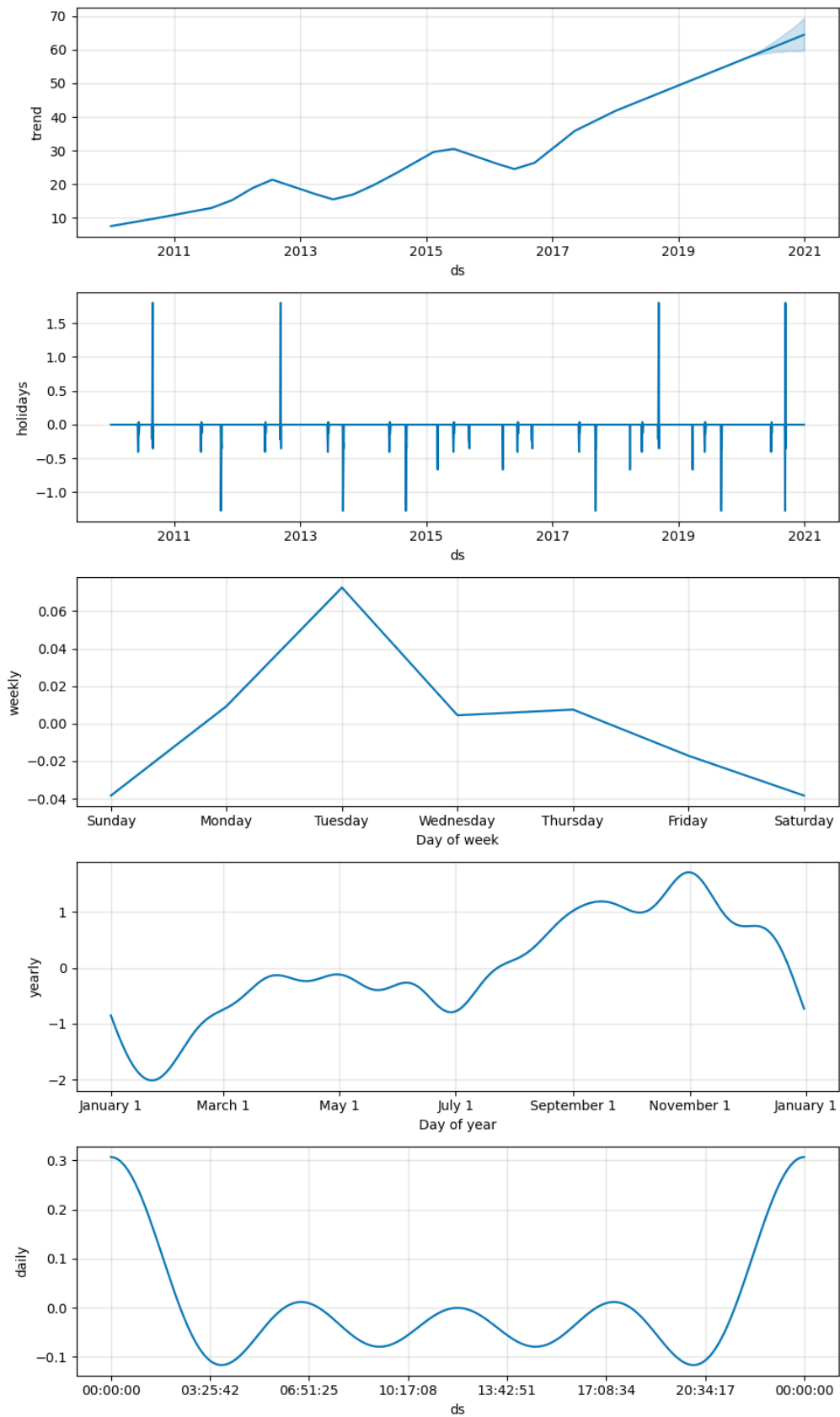


Figure 57: Graphs of three components of Prophet

The graph below 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 a 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 58: Effect of the events

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



Figure 59: Prediction made by prophet

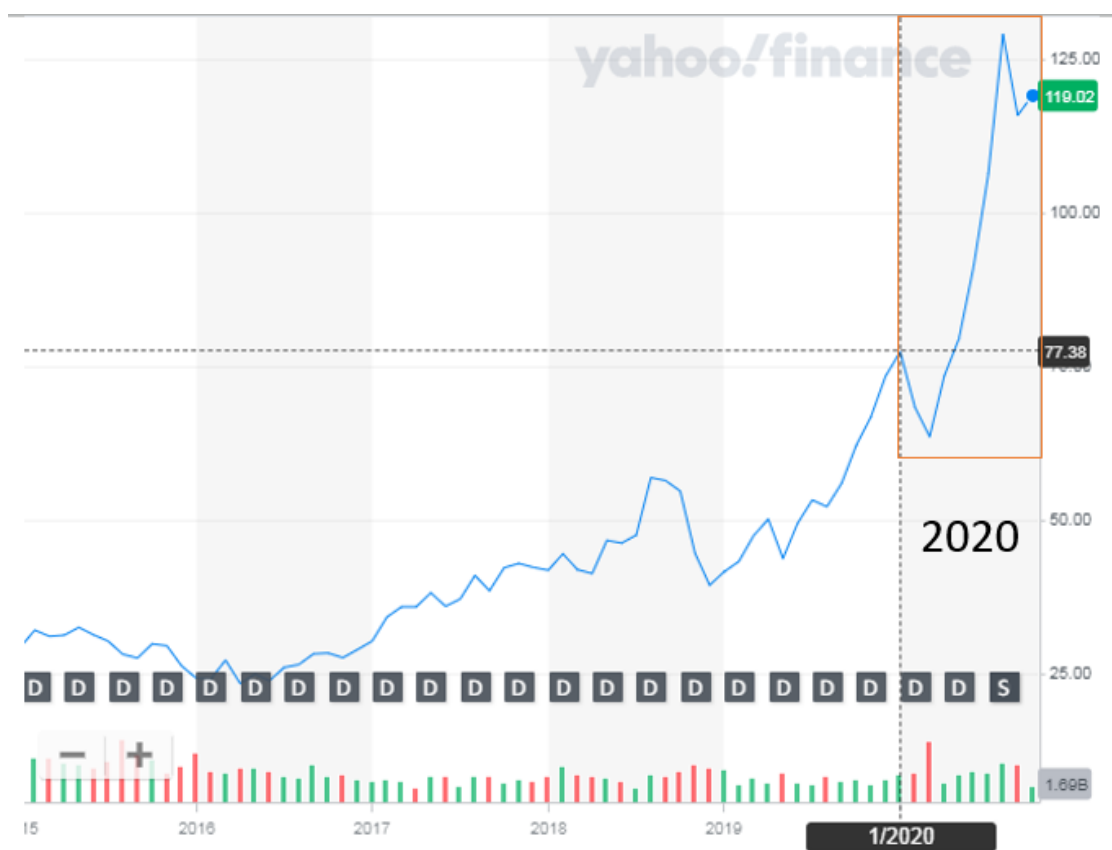


Figure 60: Actual movement of AAPL



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

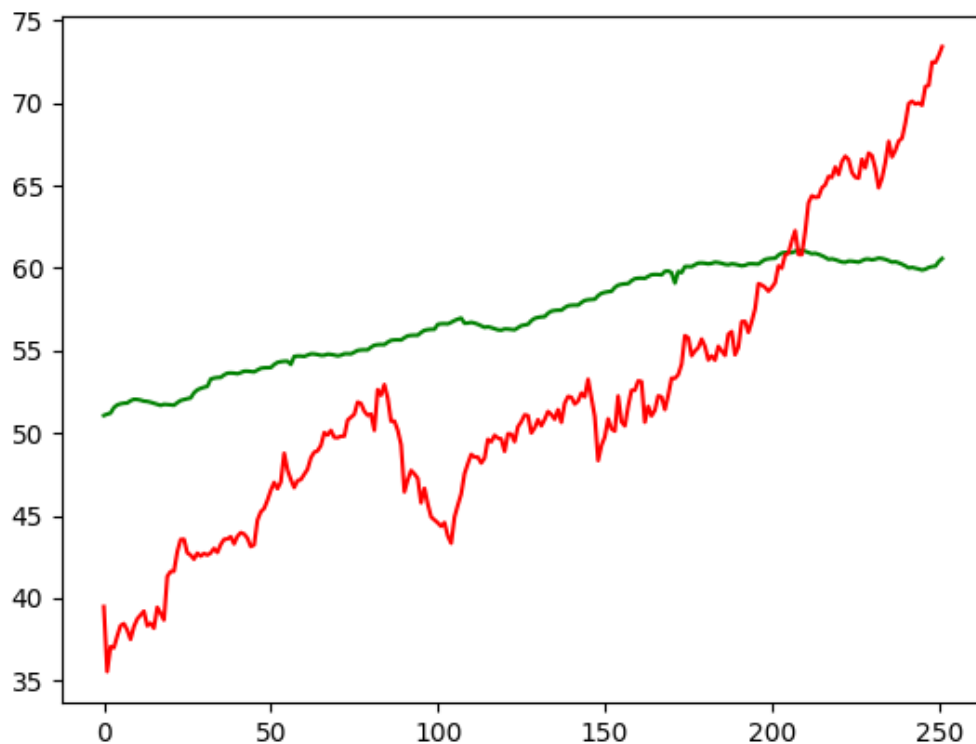


Figure 61: Prophet predicting 2019

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 sensitive to local ups and downs. The MSE of the prediction is about 150, RMSE is around 12. In a nutshell, Prophet is good at giving a general trend of the data set but is not a good model to predict rapidly changing data, like predicting the closing price of the 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 the “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]

accuracy: 49.113

Figure 62: 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 is different from financial news, there are specific terms for the 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 63: New York Time dataset with TextBlob

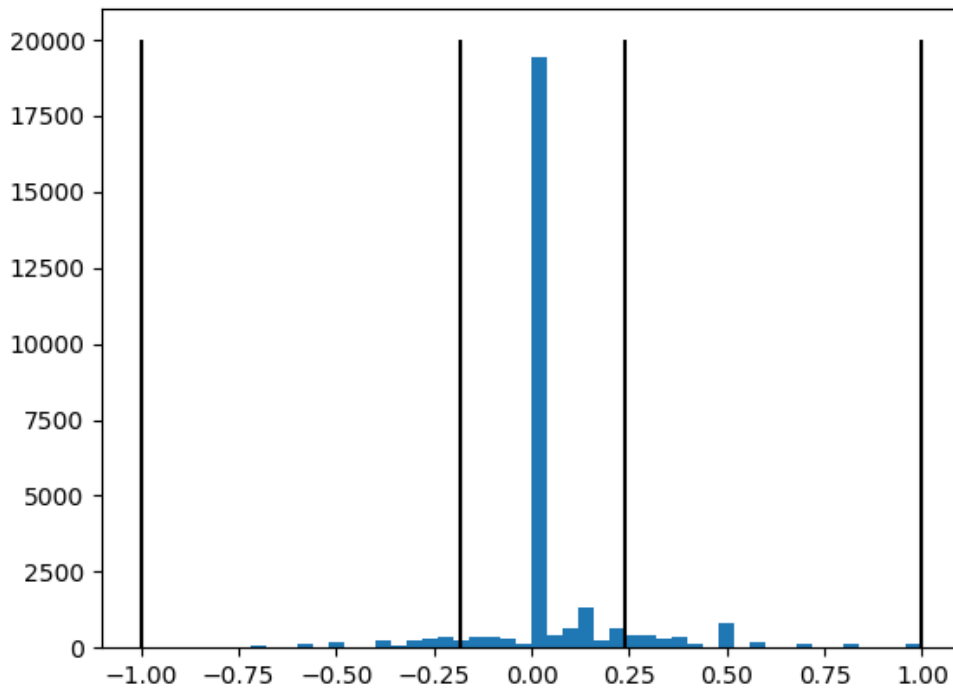


Figure 64: sentiment clustering from TextBlob

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 [41]. A clustering helps us to determine the actual category of 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 65: 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 a previous experiment, we will also use the “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]

accuracy: 54.354

Figure 66: 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 is different from financial news, there are specific terms for the 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 67: New York Time dataset with VADAR

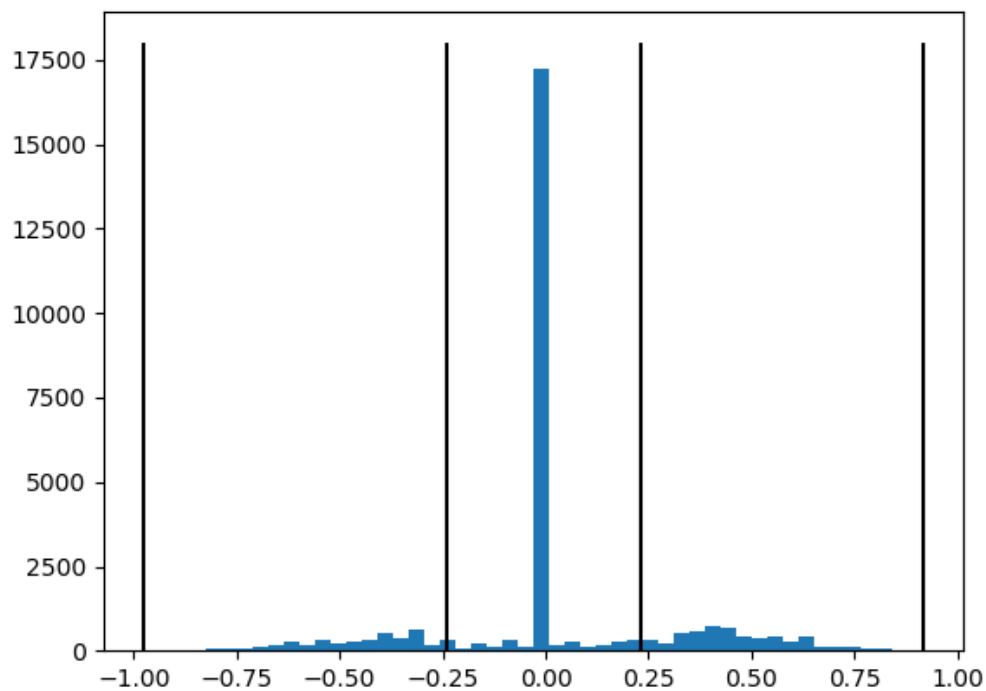


Figure 68: sentiment clustering from VADAR

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 69: Jenks break result for VADAR

As we see from the figure, values between `[-0.9776, -0.2382]` 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 the “Sentiment Analysis for Financial News” dataset from Kaggle to test and validate our ANN model.

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

Here is 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 70: 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 71: 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 72: 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 73: ANN test model 4

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

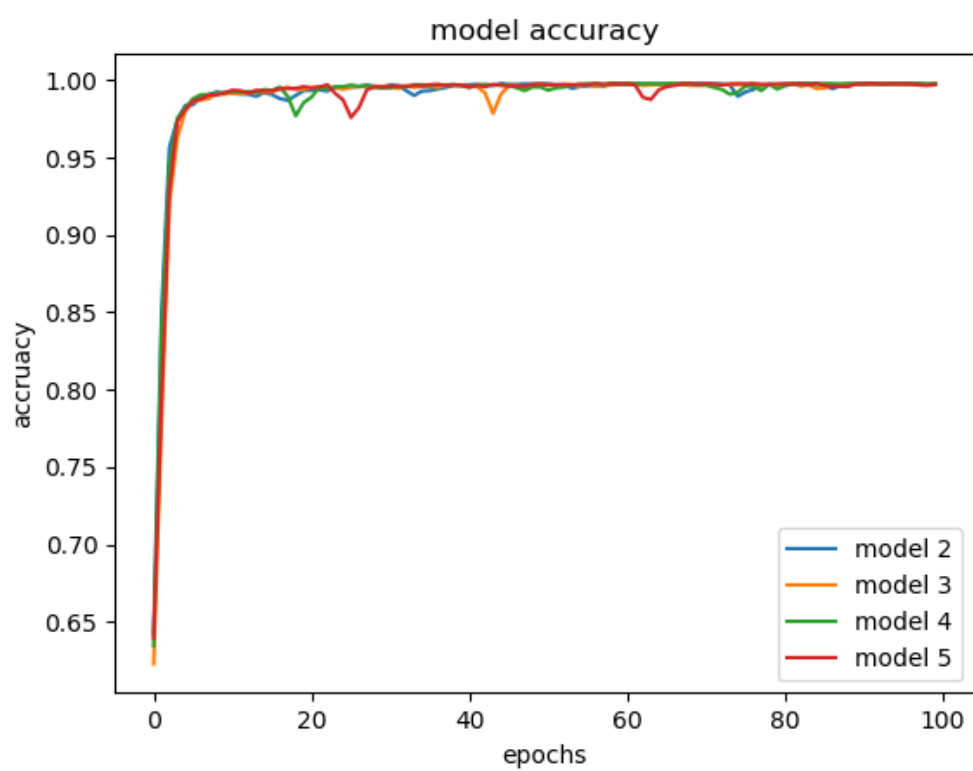


Figure 74: ANN test model accuracy

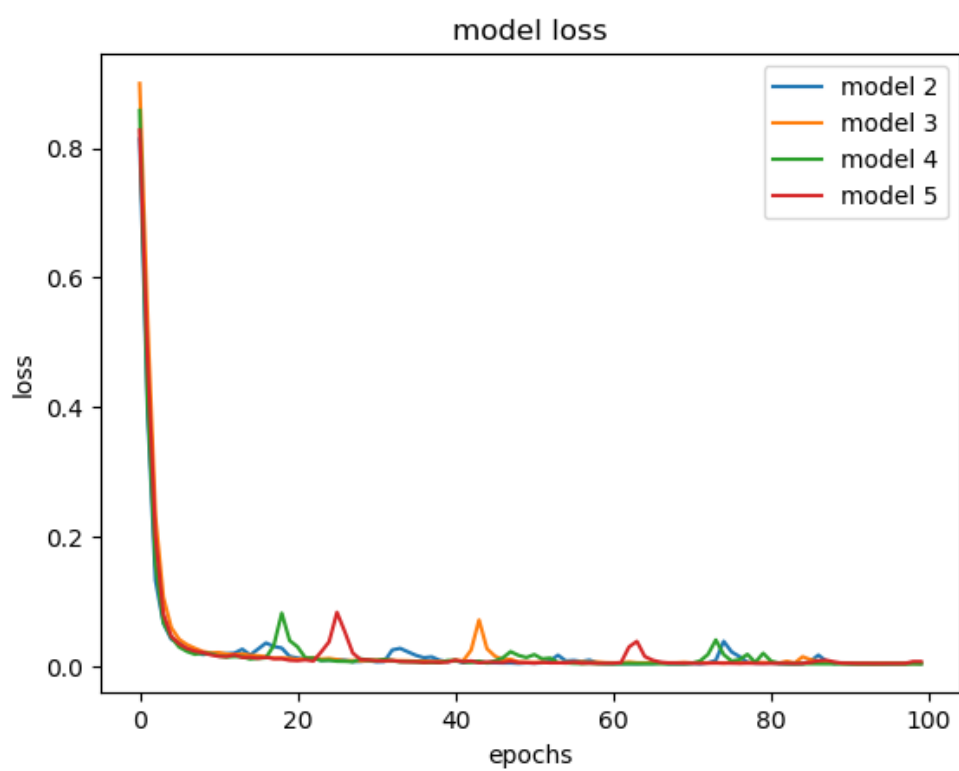


Figure 75: ANN test model loss



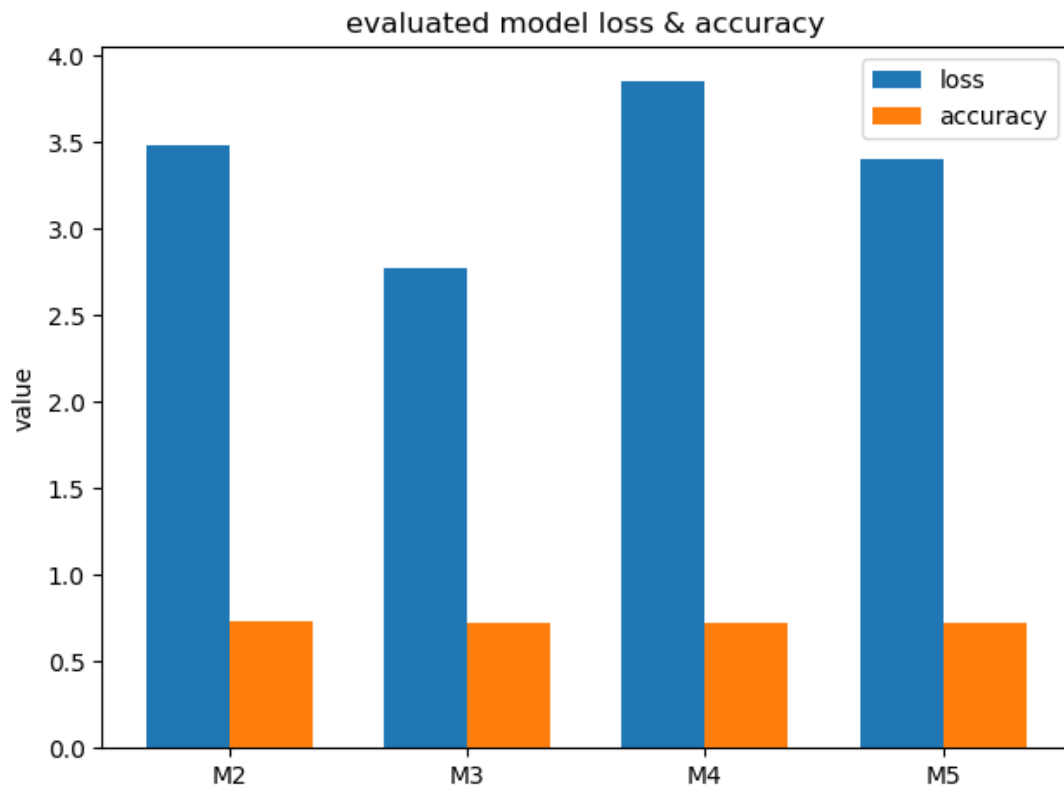


Figure 76: ANN test model validation loss & accuracy

We can see from the figure, there is no much difference between the number of layers and the number of neurons in the neural network, and there is no much change when the 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 77: ANN final model

This model has a total of 6 layer and 20 epochs. Its 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 labelled so it is good to use as a 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 78: 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%.

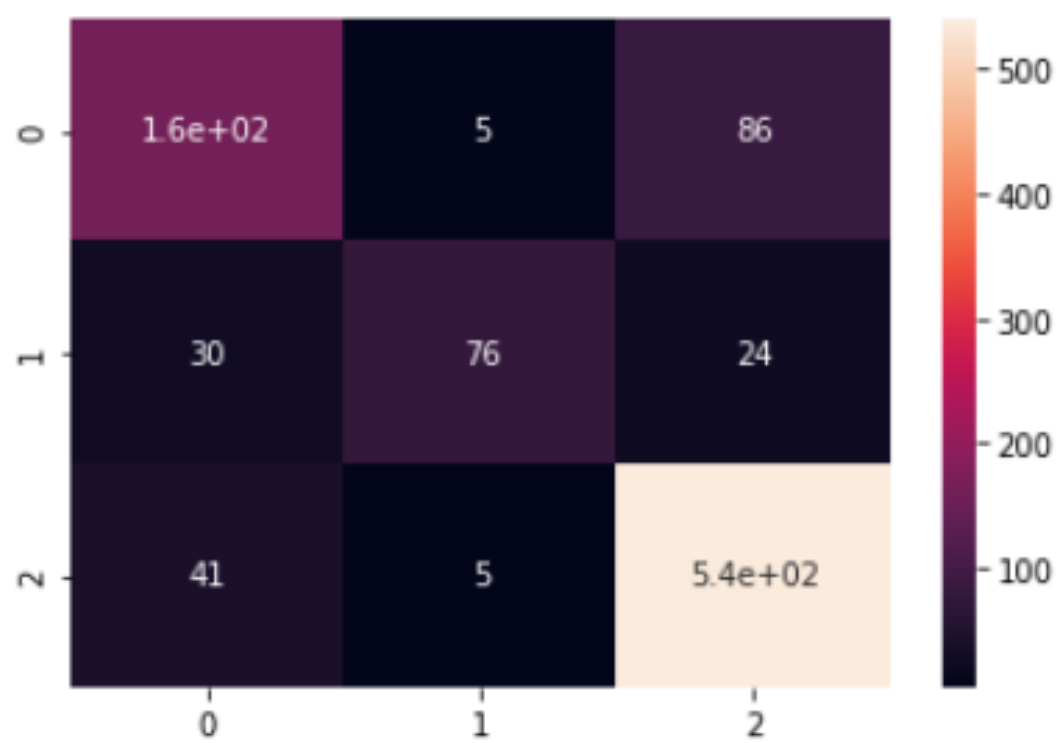


Figure 79: Heatmap of confusion matrix

## 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 80: 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 81: 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 82: 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 83: stock price & VADAR sentiment data

We observed that there are improvements when deciding the stock price will increase or not on a 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%.

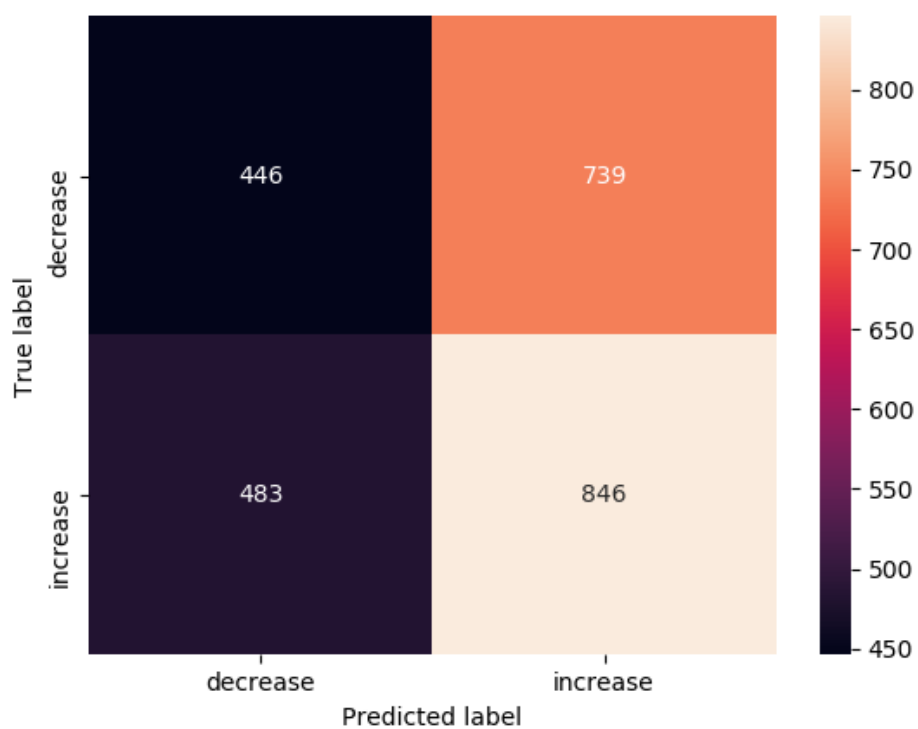


Figure 84: confusion matrix w/ sentiment

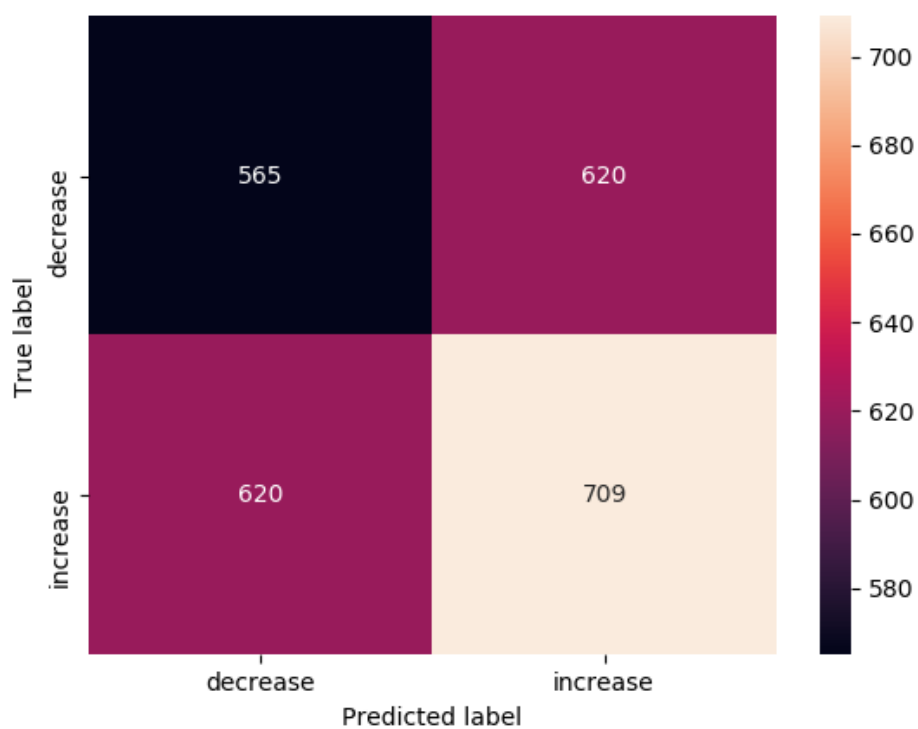


Figure 85: confusion matrix w/o sentiment

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

Figure 86: KNN + VADAR overall performance

Although the result seems to 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 of the market's emotion and trend.

## 4.4.2 LSTM + BERT

After the completion of BERT model training, we use to analyse the dataset we crawler from New York Time. There are much information for each news or article, including headline, abstract, leading paragraph. We mainly focus on analyzing the abstract with BERT, and we use the 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 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 too many false negative and false positive predictions.

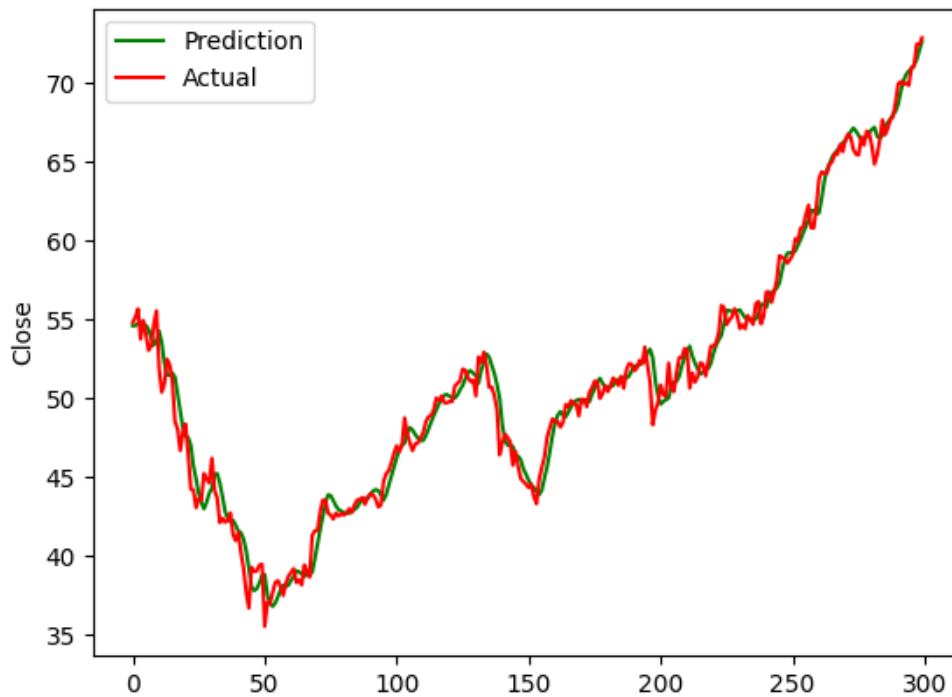


Figure 87: Prediction of the previous LSTM model

Therefore, before integrating both models, we need to modify the LSTM model to directly predict whether the stock will increase or not instead of predicting the closing price of the next trading day. And also adding the stock data of the 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 in days. 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	^DXIC_high	^DXIC_low	^DXIC_open	^DXIC_close	^DXIC_volume	^DXIC_adj_close	^GSPC_high	^GSPC_low	^GSPC_open	^GSPC_close	^GSPC_volume	^GSPC_adj_close	sentiment
0	7.660714	7.585000	7.622500	7.643214	493729600.0	6.539882	2311.149902	2294.409912	2294.409912	2308.419922	1931380000	2308.419922	1133.869995	1116.560059	1116.560059	1132.989990	3991400000	1132.989990	0.614706
1	7.699643	7.616071	7.664286	7.656428	601904800.0	6.551187	2313.729980	2295.620117	2307.270020	2308.709961	2367860000	2308.709961	1136.630005	1129.660034	1132.660034	1136.520020	2491020000	1136.520020	0.827375
2	7.686786	7.526786	7.656428	7.534643	552160000.0	6.446983	2314.070068	2295.679932	2307.709961	2301.090088	2253340000	2301.090088	1139.189941	1133.949951	1135.709961	1137.140015	4972660000	1137.140015	0.623944
3	7.571429	7.466072	7.562500	7.520714	477131200.0	6.435065	2301.300049	2285.219971	2298.090088	2300.050049	2270050000	2300.050049	1142.459961	1131.319946	1136.270020	1141.689941	5270680000	1141.689941	-0.574143
4	7.571429	7.466429	7.510714	7.570714	447610800.0	6.477847	2317.600098	2290.610107	2292.239990	2317.169922	2145390000	2317.169922	1145.390015	1136.219971	1140.520020	1144.979980	4389590000	1144.979980	0.872545
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2721	116.550003	112.879997	114.010002	115.050003	111850700.0	114.851852	11545.629883	11221.059570	11440.639648	11358.940430	3186950000	11358.940430	3441.419922	3364.860107	3441.419922	3400.969971	3988080000	3400.969971	-0.189500
2722	117.279999	114.540001	115.489998	116.599998	92276800.0	116.389178	11465.059570	11361.860352	11409.339644	11431.349609	3079530000	11431.349609	3409.510010	3388.709961	3403.149902	3390.679932	3946990000	3390.679932	0.341500
2723	115.430000	111.099998	115.050003	111.199997	143937800.0	111.008476	11249.950195	10999.070312	11230.900391	11004.870117	3912580000	11004.870117	3342.479980	3268.889893	3342.479980	3271.030029	5129860000	3271.030029	0.282444
2724	116.930000	112.199997	112.370003	115.320000	146129200.0	115.121384	11287.629883	11030.190430	11064.469727	11185.589844	3222460000	11185.589844	3341.050049	3259.820068	3277.169922	3310.110107	4903070000	3310.110107	-0.272000
2725	111.989998	107.720001	111.059998	108.860001	190272600.0	108.672516	11129.809570	10822.570312	11103.469727	10911.589844	3662840000	10911.589844	3304.929932	3233.939941	3293.590088	3269.959961	4840450000	3269.959961	0.468818

Figure 88: Input sample

The architecture of the integrated model is starting with an input layer, because the sliding window size is 10 and 19 feature each day, so there are 190 units 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 classification.

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 increased in 51.37% days of training set and 58.92% days of the testing set. Its observation matches 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 accuracy high than 50%, always predict the stock will increase on the next trading day is still better than our model in the case of AAPL.

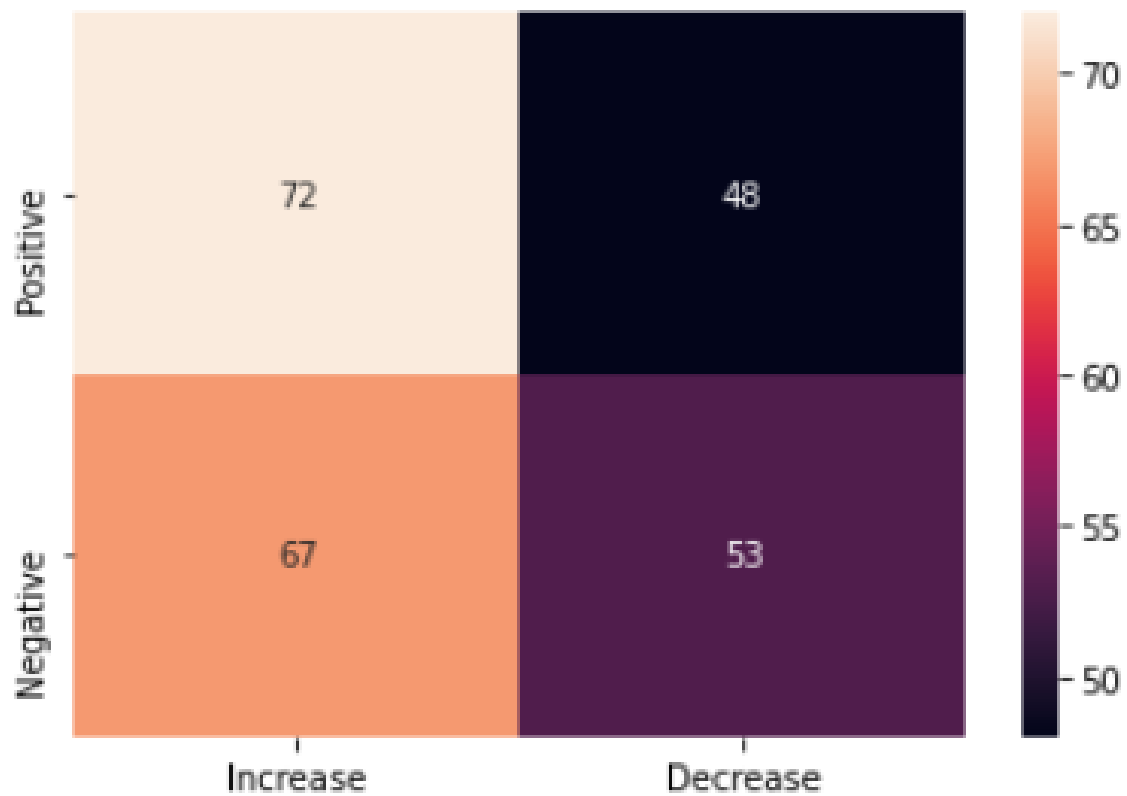


Figure 89: Heatmap of a confusion matrix

However, adding news and articles which are related to Apple to our model indeed gives us an improvement 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 lower accuracy in predicting the closing price of the next trading day. The accuracy is 0.5083 and the balanced accuracy is 0.5092.

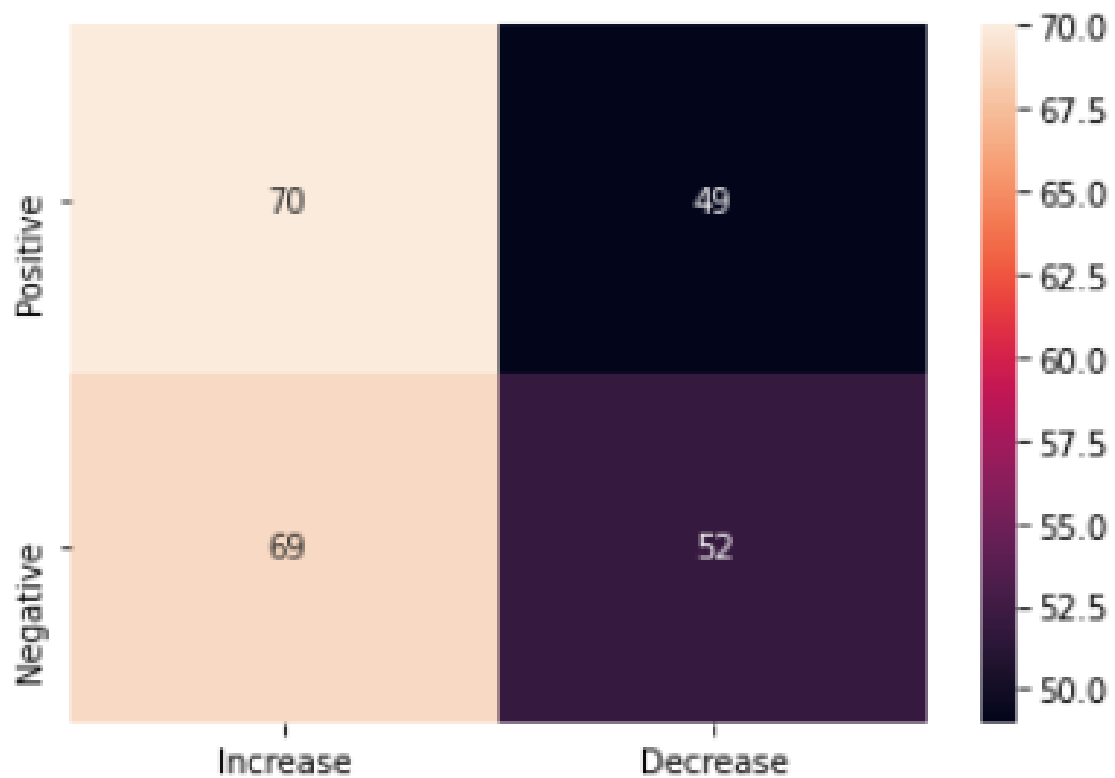


Figure 90: Heatmap of the confusion matrix (sentiment set to 0)

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 91: Result comparison

Here we do a case study on how adding sentiment values to the model improves performance. The example below shows that the classification without sentiment is “Fall”, however, the correct classification should be “Rise”. In this case, adding the sentiment value make the model give a correct 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 92: Case studies

## 4.5 Pattern Recognition

### 4.5.1 Hard-coded Pattern Recognizer

As we cannot find a dataset for pattern recognition, we need to prepare our own dataset to train the model. We get the stock data of the component of the S&P 500 index with a python package “Pandas-datareader”, from 2010 to 2019. The target pattern is “Double Bottom Pattern”. We follow the traits mentioned in [10]. Based on them, we write a program to search for the double bottom patterns among the 500 stock. The first step is to locate all the local minima in the data. For demonstration, there is a double bottom in the figure, we will gradually show how we identify the whole pattern. The red dots in figure 92 are the local minima,

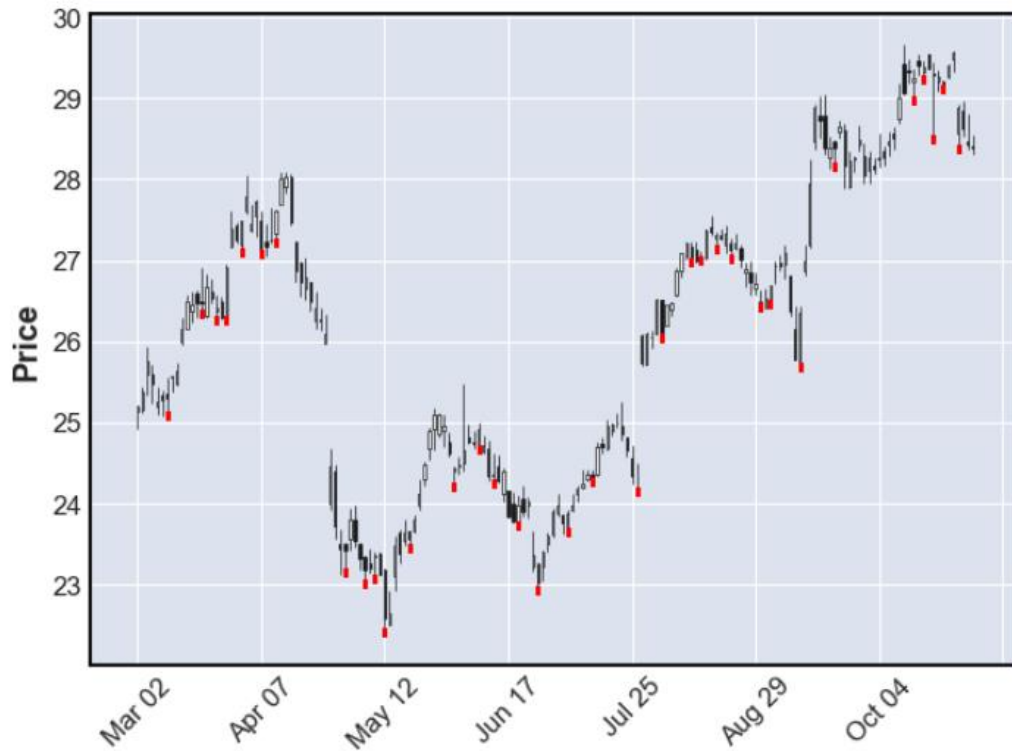


Figure 93: Example of locating local minima

The difference between the two bottom points should be within 3-5% is the second criteria of the double bottom pattern. Therefore, I connect the two local minima with a line if their difference is within 3-5%. In this step, we will generate many lines, shown in figure 93, and we will remove some of them in the following step until only two local minima which compose the double bottom pattern.

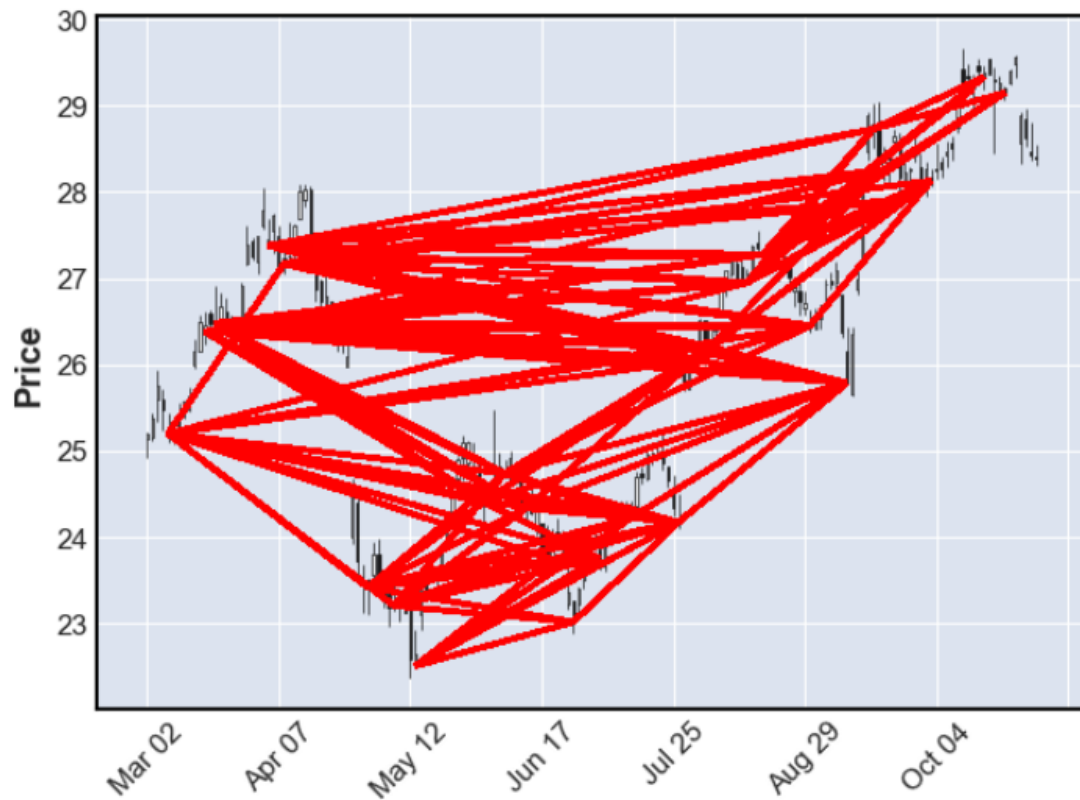


Figure 94: Lines connecting

There is a small rebound between the two bottoms. We check the existence of the rebound by two things. The rebound should be at least 5%, then we will check whether we can fit a quadratic equation between the two bottoms. Figure 94 shows an example of fitting a parabola between the two bottoms which are connected by the red line

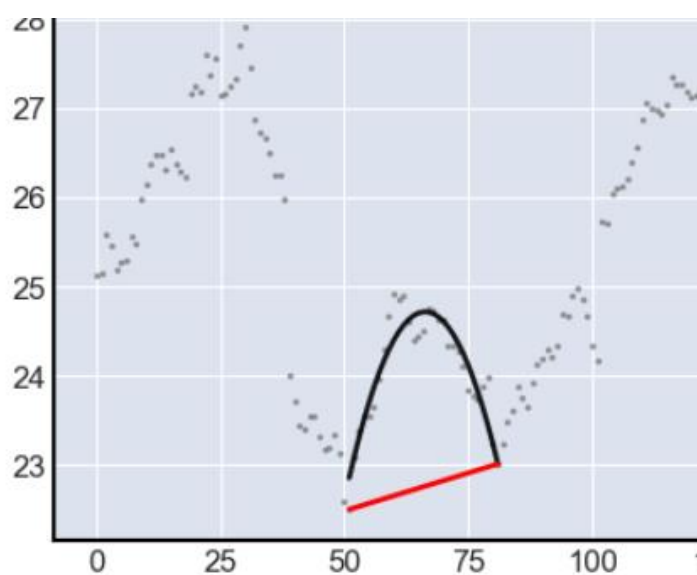


Figure 95: Example of fitting parabola

Then we further remove some red lines in figure 93 if we cannot fit a parabola between the two ends of the red lines. Figure 95 is the result of removing those lines.

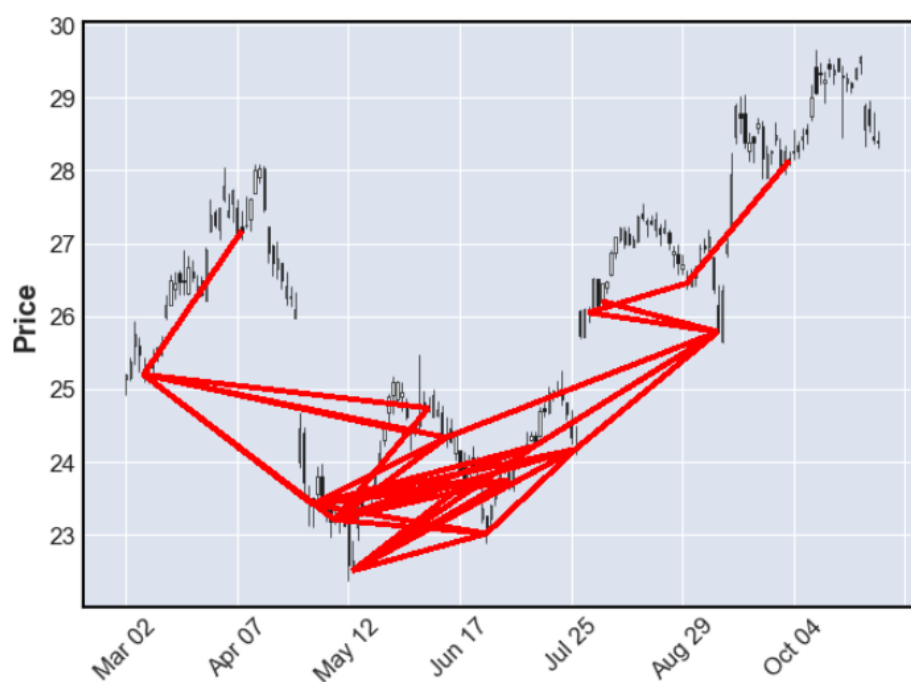


Figure 96: Remaining lines after fitting parabola

The red lines supposed to be connecting the two bottom points of a pattern, and thus they shouldn't cut through any candlesticks. We can remove those lines if they do. Figure 96 shows the remaining lines.

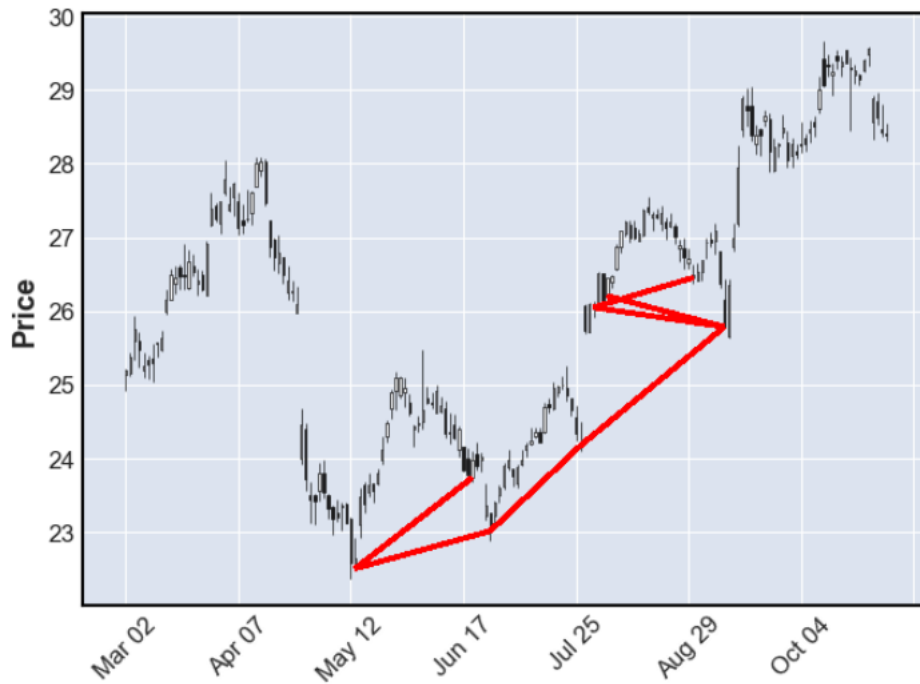


Figure 97: Lines remaining

The red line is indeed the bottom line of a double bottom pattern, then there should not be any candlesticks below a red line before or after the red line. For example, the figure 97 shows the red line is not the base of a double bottom pattern as there are some candlesticks under the red line. After removing that kind of lines, we now left those lines which are most likely to be the base of the pattern shown in figure 98.



Figure 98: Example of line which is not possible to be a base of a double bottom pattern



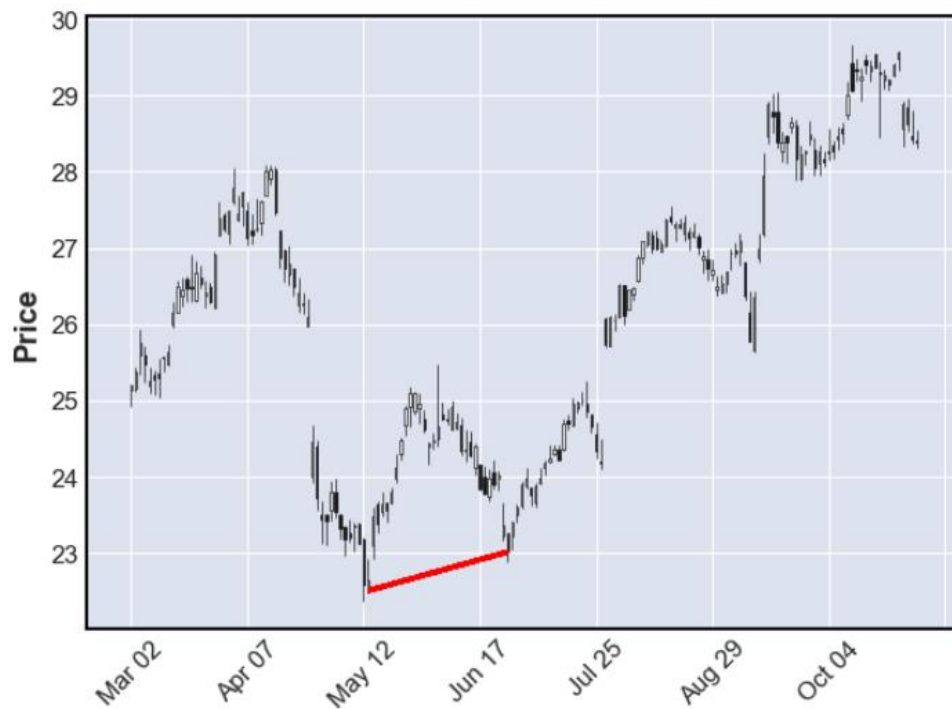


Figure 99: Qualified baseline

Now we have the baselines, in the following, we will identify the first significant drop and the last significant rise of the pattern. A first drop should be around 10-20% while the last rise should over the overall loss to reach the starting price of the whole pattern. We would say a pattern is identified if we can find these three lines, the first significant drop, the baseline and the last significant rise, as shown in figure 99 with the red, blue and green lines respectively.

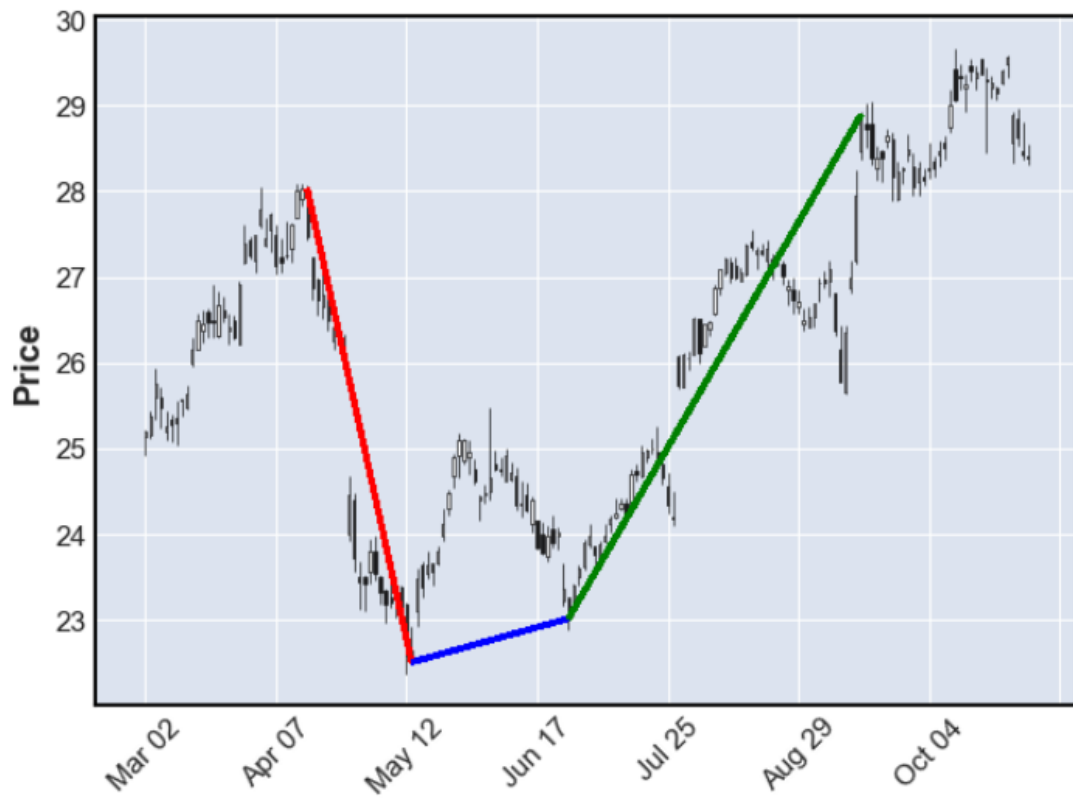


Figure 100: A double bottom pattern recognized by our program



Figure 101: Overview of pattern recognized from 2010 to 2019 in Apple stock history

## 4.5.2 Dataset

We target the component of Standard & Poor's 500 indexes, applying our hard-coded pattern recognizer to them and find out the double bottom pattern for later machine learning model training use. However, the result is not satisfying. The quality of the patterns recognized variant a lot. After we filter them one by one manually. Among the five hundred company, our program can only find 234 double bottom patterns which are useful from 182 companies. The time span of the patterns varies from 55 days to 235 days. From the rest of the 318 companies, we randomly picked 234 periods of stock record from 55 days to 235 days as the negative samples.

As LSTM is our main model to learn the dataset, we cannot fit the data with different sequence length. And thus, we need to scale the data to a unified sequence length. For all sample with a time span smaller than 235 days, they will be scaled by repeating themselves to 235 days. More specifically, repeating  $n$  times where  $n$  is the quotient of 235 divided by the time span of that pattern, and then randomly pick  $k$  days before scaling with no replacement, where  $k$  is 235 modulo the time span of that pattern. The figures below show the result of this scaling process.

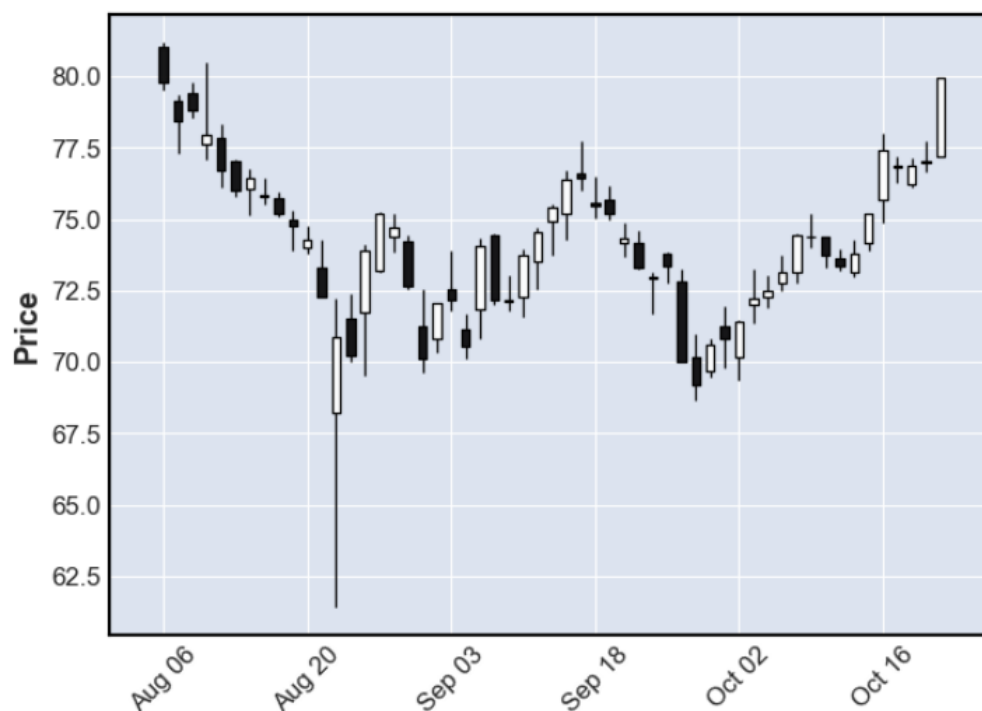


Figure 102: Before scaling. Timespan is 64 days

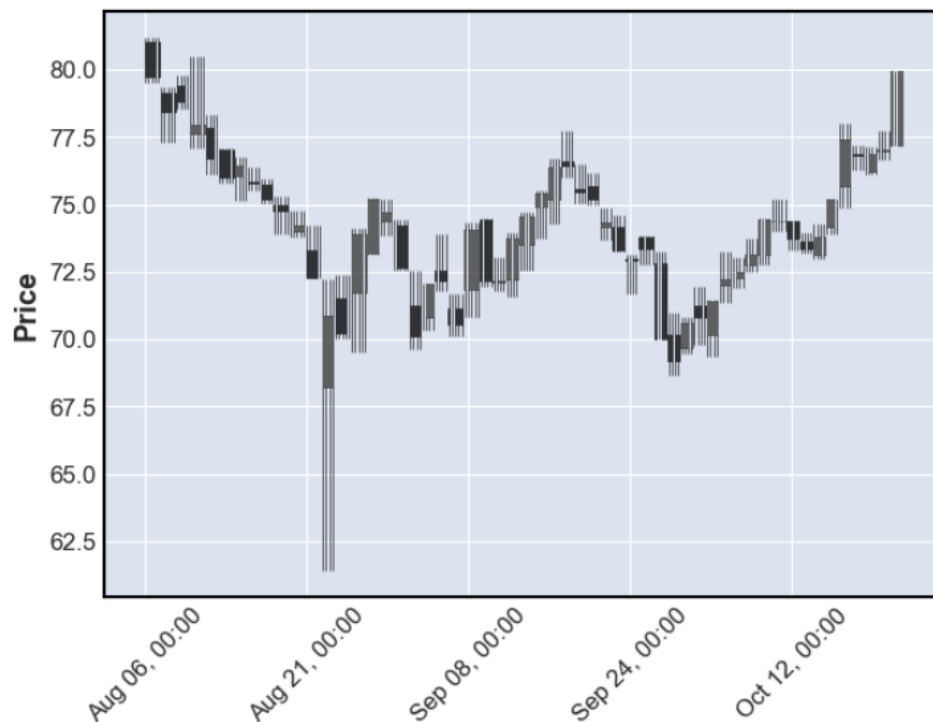


Figure 103 After scaling. Timespan is 235 days

### 4.5.3 LSTM Model Experiment Result

We use the cross-validation method to test different hyper-parameters. For each set of hyper-parameters, we will train the model 10 times for finding the mean and variance of loss.

Figure 103 shows the combination of later and the average validation loss. And figure 104 shows the average accuracy of the predicting validation set. In each column and row has a list of integers. For example [16,8] means this model has 2 layers, the first layer has 16 units, the second layer has 8 units. Each model has at least one LSTM layer and at least one dense layer. Each floating-point number in the data represent the average validation loss in the 10 separated training.

The result shows us that using one LSTM layer of 4 units and 3 dense layers with 16, 8, 1 unit respectively can give us the best result among all the other combination of layers. The validation loss is 0.960 and the accuracy is 0.565.

LSTM	[4]	[8]	[16]	[32]	[8, 4]	[16, 4]	[16, 8]	[32, 8]	[32, 16]	[16, 8, 4]
Dense										
[1]	0.726831	0.696726	0.699982	0.707643	0.766637	0.702040	0.698780	0.711815	0.723233	0.704372
[4, 1]	0.707077	0.693055	0.694595	0.703030	0.697822	0.806689	0.730751	0.751265	0.696239	0.693188
[8, 1]	0.694618	0.700404	0.701528	0.704191	0.765635	0.694593	0.693295	0.694566	0.707367	0.693120
[16, 1]	0.695024	0.709587	0.700311	0.703101	0.694556	0.693155	0.704758	0.694818	0.704564	0.694842
[16, 8, 1]	0.690570	0.698109	0.699208	0.708586	0.719594	0.695222	0.711063	0.690496	0.701868	0.693134

Figure 104: Average validation loss

LSTM	[4]	[8]	[16]	[32]	[8, 4]	[16, 4]	[16, 8]	[32, 8]	[32, 16]	[16, 8, 4]
Dense										
[1]	0.536232	0.543478	0.538043	0.538043	0.543478	0.532609	0.539130	/	0.539855	0.521739
[4, 1]	0.543478	0.586957	0.532609	0.534783	0.532609	/	0.528986	/	0.538043	0.543478
[8, 1]	0.543478	0.543478	0.530435	0.573913	0.532609	0.543478	0.521739	0.532609	0.554348	0.521739
[16, 1]	0.556522	0.543478	0.543478	0.539855	0.536232	0.543478	0.543478	0.536232	0.543478	0.543478
[16, 8, 1]	0.565217	0.526087	0.532609	0.536232	0.521739	0.543478	0.536232	0.565217	0.543478	/

Figure 105: Average accuracy in predicting validation set

However, not every single training was successful, the loss was never reduced in the iteration. For example, the figure 105, in this case, we have 3 training ended with training loss and validation loss of around 7.712. So, this training was not counted in the mean. 0.711815 in column [32,8] and row [1] was computed with the other 7 numbers. Similar to handling the mean accuracies, we excluded those 0.5s in the calculation as the model is predicting purely 1 or 0 for any input. Therefore, some entries in figure 105 have no number, as all the prediction result is all 0.5.

```
{'lstm': [32, 8],
'dense': [1],
'train_loss': [0.6783584356307983,
0.6860380172729492,
0.6598420143127441,
7.712474822998047,
7.712474346160889,
0.6632499694824219,
0.6781659126281738,
0.6660026907920837,
7.712474346160889,
0.6869624257087708],
'val_loss': [0.6989296078681946,
0.6965937614440918,
0.7346721887588501,
7.7124738693237305,
7.7124738693237305,
0.7233051657676697,
0.7057515978813171,
0.7307489514350891,
7.7124738693237305,
0.6927011609077454],
'train_accu': [0.5614973306655884,
0.5,
0.6203208565711975,
0.5,
0.5,
0.6096256971359253,
0.5614973306655884,
0.6203208565711975,
0.5,
0.5614973306655884],
'val_accu': [0.5,
0.5,
0.43478259444236755,
0.5,
0.5,
0.47826087474823,
0.5,
0.41304346919059753,
0.5,
0.47826087474823]}
```

Figure 106: Example of not all training was successful

## 4.6 YouTube comment and Stock price

### 4.6.1 Data Acquisition

In order to perform sentiment analysis on the video comments about the Apple company. We use YouTube Data API to get a list of videos first, and then we further request the comments for each video in the list. There are some limitations in using this API. Once we exceed the daily limit, we cannot use the API on that day, the quota

will refresh on the next day. The second limitation of the API is that we cannot filter the comments by date, it always returns the comments in the descending order of date, which makes it difficult to get the comment within a certain period a long time ago. Therefore, we pick the period from Jan 1 of 2021 to Mar 31 of 2021 for our analysis. And there are 84805 comments had in this period.

## 4.6.2 Experiment Result

We want to study the relationship between YouTube sentiment and the stock price. In this section, we will have three experiments. They will gradually show the relationship between the sentiment value and the stock price. The table below is the summary of the experiment.

Table 1 youtube experiment

	Graph	Covariance	Correlation
1(base case)	Sentiment vs stock price	0.00796	0.125
2	Change in sentiment vs change in stock price	0.0439	0.541
3	Yesterday sentiment vs change in stock price	0.0508	0.661

The first experiment is to check the covariance of sentiment and stock with and performing any modification to the data. We plot a graph of daily average sentiment vs the close price of AAPL. This will be our base case for later comparison. The figure below shows their trend. The red line is the average sentiment polarity. The blue line is the close price of AAPL.

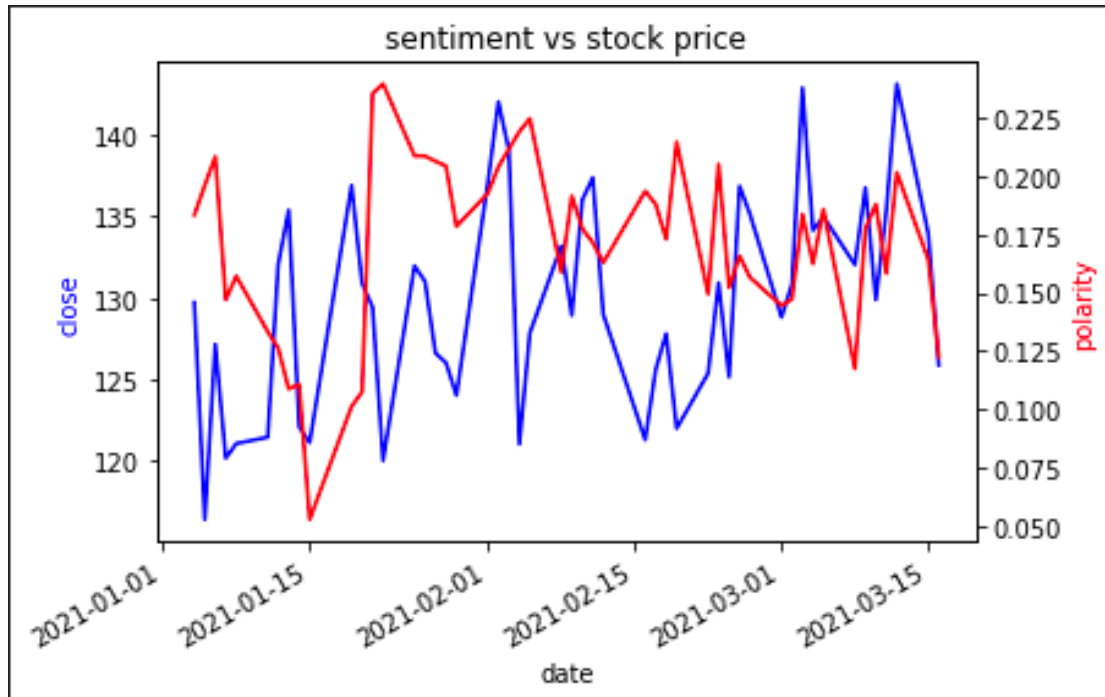


Figure 107: Daily average sentiment vs stock price

We can see that these two lines have similar movement because we can see the spikes are anastomosed sometimes. Their covariance of close and polarity is 0.00796, the correlation is 0.125. It shows that they have some positive relation.

	close	polarity
close	42.579533	0.007960
polarity	0.007960	0.001496

Figure 108: Covariance of close and polarity

The second experiment aim at studying the change in sentiment value and the change in stock price. Through this experiment, we can know if a positive change in sentiment value in YouTube comments can lead to a positive change in stock price. Figure 108 shows the graphical representation of this experiment.



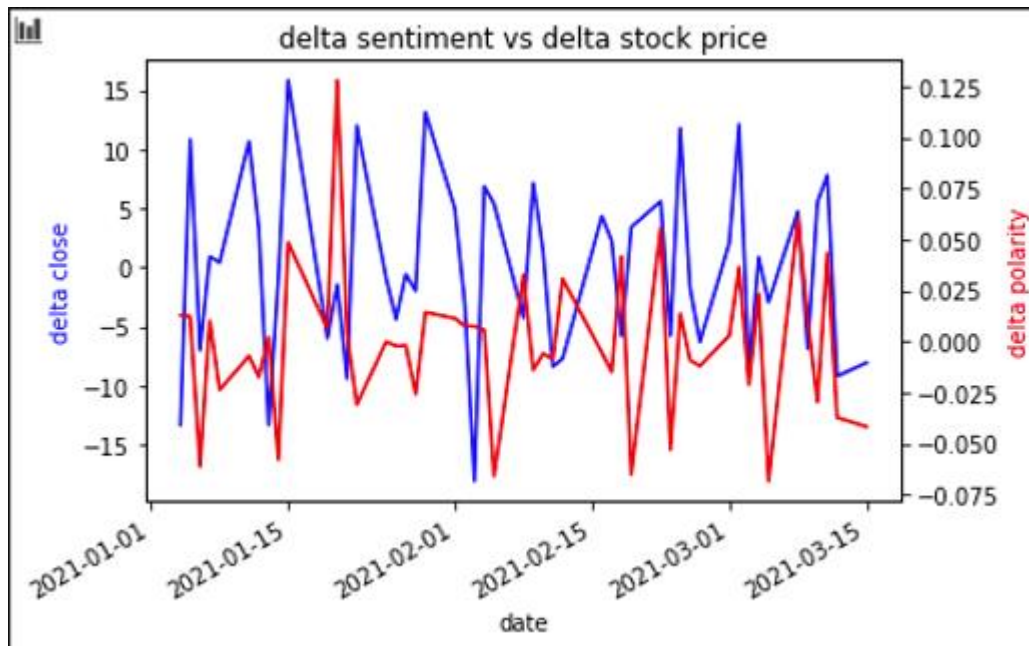


Figure 109: Change in sentiment vs change in stock price

We can see that the spikes of the two line have a better anastomosis compared to the base case. Therefore, we can expect they will have better covariance as shown in figure 109, the covariance is about 0.0439, the correlation is significantly increased to 0.541. That means the change in sentiment and change in stock price is quite related.

	delta_close	delta_polarity
delta_close	59.668095	0.043852
delta_polarity	0.043852	0.001357

Figure 110: Covariance of change in close and change in polarity

The last experiment is aimed at studying the time lag of the YouTube comment sentiment and the stock price. Through this experiment, we can know if we can use the sentiment of yesterday to predict the stock price of today. Figure 110 shows the sentiment of yesterday, the line, to the stock price of today.

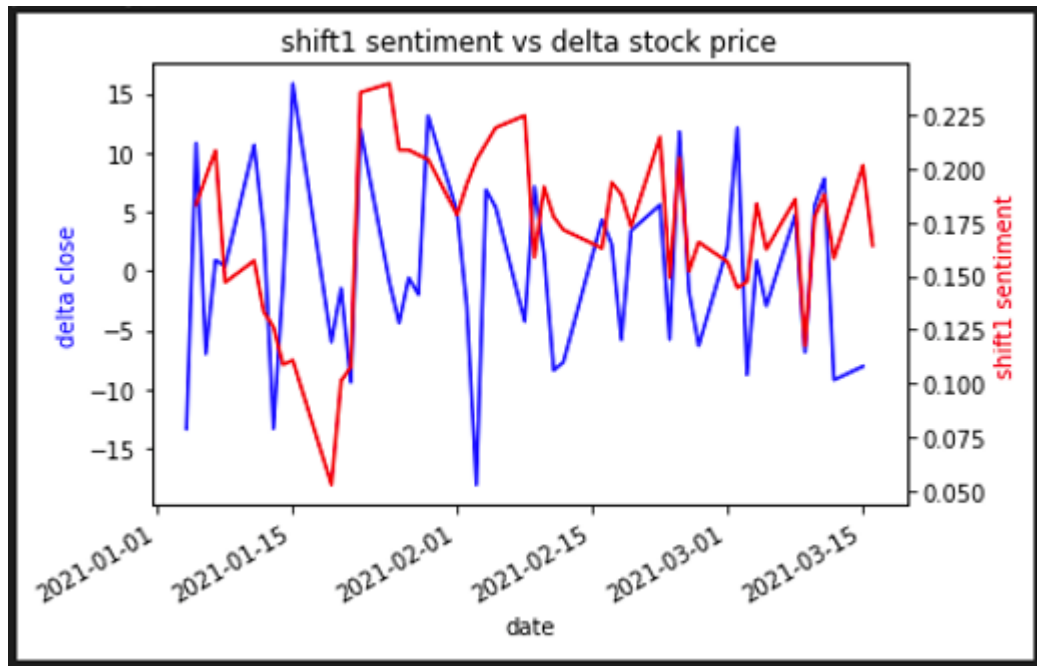


Figure 111: Yesterday sentiment vs stock price

This experiment gives us even a better result. The covariance of yesterday sentiment and stock price is 0.0508, the correlation of them is 0.661. This result shows us there is a relationship between them. If we see a high polarity today, it is likely that the stock price of tomorrow will rise.

	delta_close	polarity_shifted1
delta_close	59.668095	0.050821
polarity_shifted1	0.050821	0.001476

Figure 112: Covariance of yesterday sentiment and stock price

## 4.7 Twitter tweets and Stock price

### 4.7.1 Data acquisition

The first dataset we obtained is from Internet Archive. We will use their “twitter stream” collection from 2019-01 to 2019-06 in this experiment.

The second dataset we obtained is from twint. Since twint is a python library package, it collects tweets in real-time while we run our program. For simplicity, we use both approaches to collect the tweets from 2019-01 to 2019-06 to perform the experiment.

### 4.7.2 Experiment result (IA)

The first result is the dataset from Internet Archive. We will use the collection set from Internet Archive to collect Twitter tweets for us to perform the experiment. We want to study the relationship between Twitter sentiment and the stock price. We will use twint to collect Twitter tweets for us to perform the experiment. We want to study the relationship between Twitter sentiment and the stock price.

This table summarize the result.

Table 2 twitter experiment (Internet Archive)

	title	covariance	correlation
1	Today's polarity vs today's stock price	0.023761	0.139869
2	Yesterday's polarity vs today's stock price	0.032677	0.19541
3	Delta polarity vs delta stock price	-0.014967	-0.128372
4	Delta yesterday polarity vs delta stock price	0.007984	0.068237

We will start from the base case, which is using today's polarity and today's stock price

to analysis their relationships.

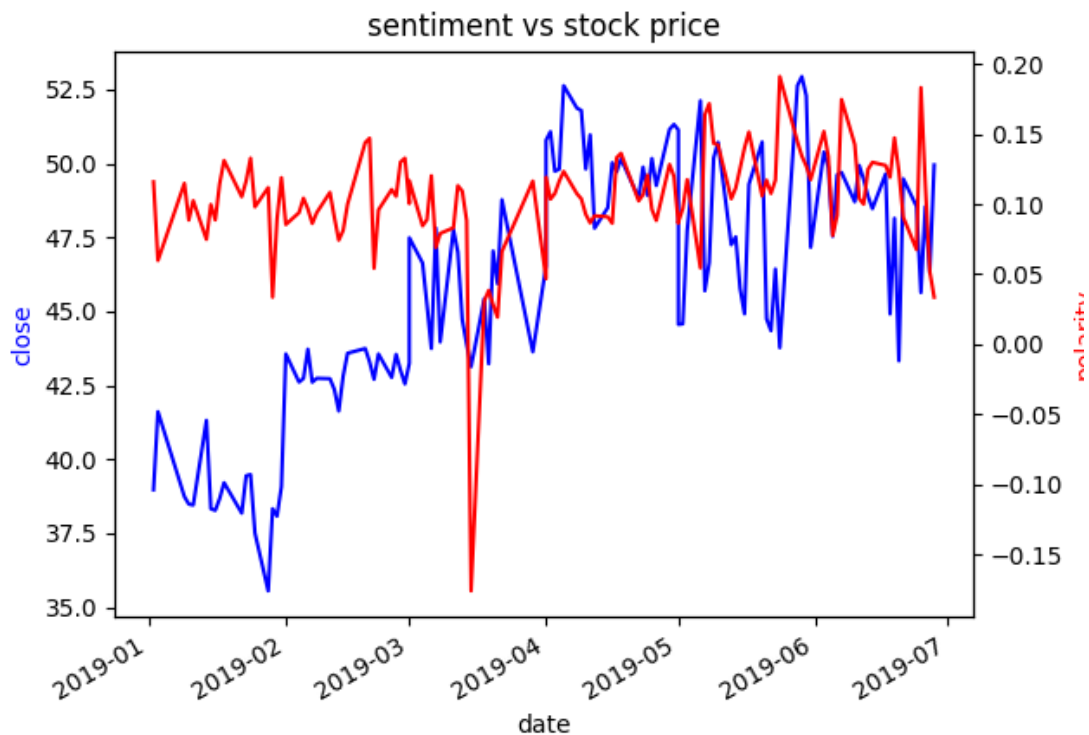


Figure 113 [twitter] today polarity vs today stock price

```
no fourier covariance:
      close  polarity
close  17.547609  0.023761
polarity  0.023761  0.001645
no fourier correlation:
      close  polarity
close  1.000000  0.139869
polarity  0.139869  1.000000
```

Figure 114 [twitter] covariance & correlation of today's polarity vs today's stock price

We can see from the graph, there are some relationship between today's stock price and today's polarity. The covariance is 0.023761 and correlation is 0.139869, which also indicates there are some relationship between today's stock price and today's polarity.

Next we perform another comparison model on yesterday's polarity and today's stock price.

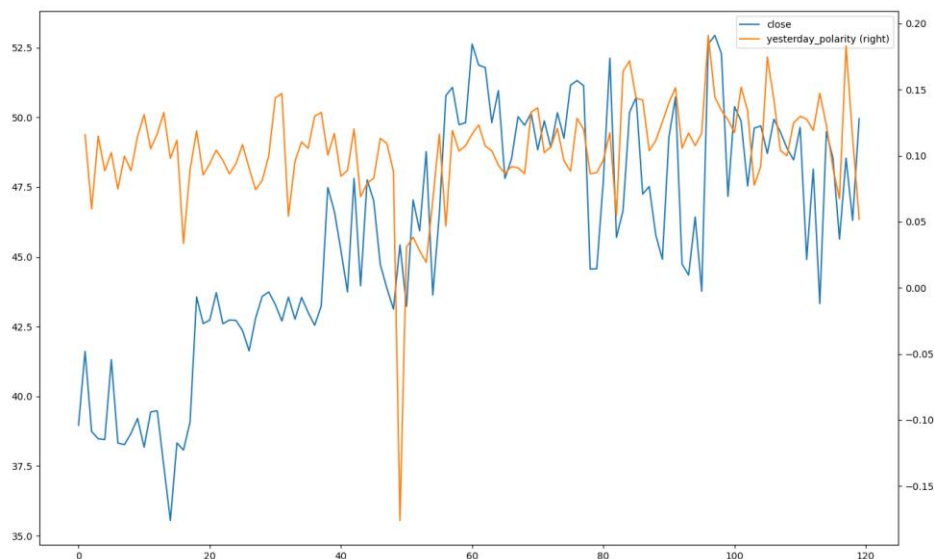


Figure 115 [twitter] yester polarity vs today stock price

```
close yesterday_polarity covariance:
               close yesterday_polarity
close          17.547609          0.032677
yesterday_polarity  0.032677          0.001617
close yesterday_polarity correlation:
               close yesterday_polarity
close          1.000000          0.19541
yesterday_polarity  0.19541          1.00000
```

Figure 116 [twitter] covariance & correlation of yesterday polarity vs today stock price

We can see from the correlation value, it increased from 0.139 to 0.195. Although it is a slight increasement, it still indicates that using yesterday's polarity to predict today's stock price will have a better result.

### 4.7.3 Experiment result (twint)

The second result is the dataset from twint. We will use twint to collect Twitter tweets for us to perform the experiment. We want to study the relationship between Twitter sentiment and the stock price. We conducted several experiments with different values. The table below is the summary of the experiment.

Table 3 twitter experiment (twint)

	Fourier polarity	Fourier price	covariance	correlation
1	N/A (today)	N/A (today)	-0.015243	-0.059368
2	N/A (yesterday)	N/A (today)	0.015503	0.060237
3	N/A ( $\Delta$ today)	N/A ( $\Delta$ today)	-0.023813	-0.09649
4	N/A ( $\Delta$ yesterday)	N/A ( $\Delta$ today)	0.050172	0.201895
5	5	5	0.004014	0.228205
6	5	10	0.004586	0.081181
7	5	15	0.004586	0.048984
8	5	20	0.004586	0.041089
9	10	5	0.004586	0.240461
10	10	10	0.000304	0.004969
11	10	15	0.000696	0.006855
12	10	20	0.000696	0.00575
13	15	5	0.004586	0.231462
14	15	10	0.000696	0.010936
15	15	15	-0.002506	-0.023763
16	15	20	-0.003914	-0.031133
17	20	5	0.004586	0.226844
18	20	10	0.000696	0.010718
19	20	15	-0.003914	-0.036375
20	20	20	-0.008449	-0.06586

We perform calculation of covariance and correlation value to see the relationship of each comparison. After that, we performed fourier transform to make the graph into something that is easier to analysis. In this particular case, we will perform inverse fourier transform to convert the real life data (time domain) to a easy to analysis sinusoids (frequency domain). In addition, we have a parameter to zero-fill at the middle to get a different analysis point-of-view of the graph.

We will start from the base case, which is today's polarity vs today's stock price. Below is the corresponding graph.

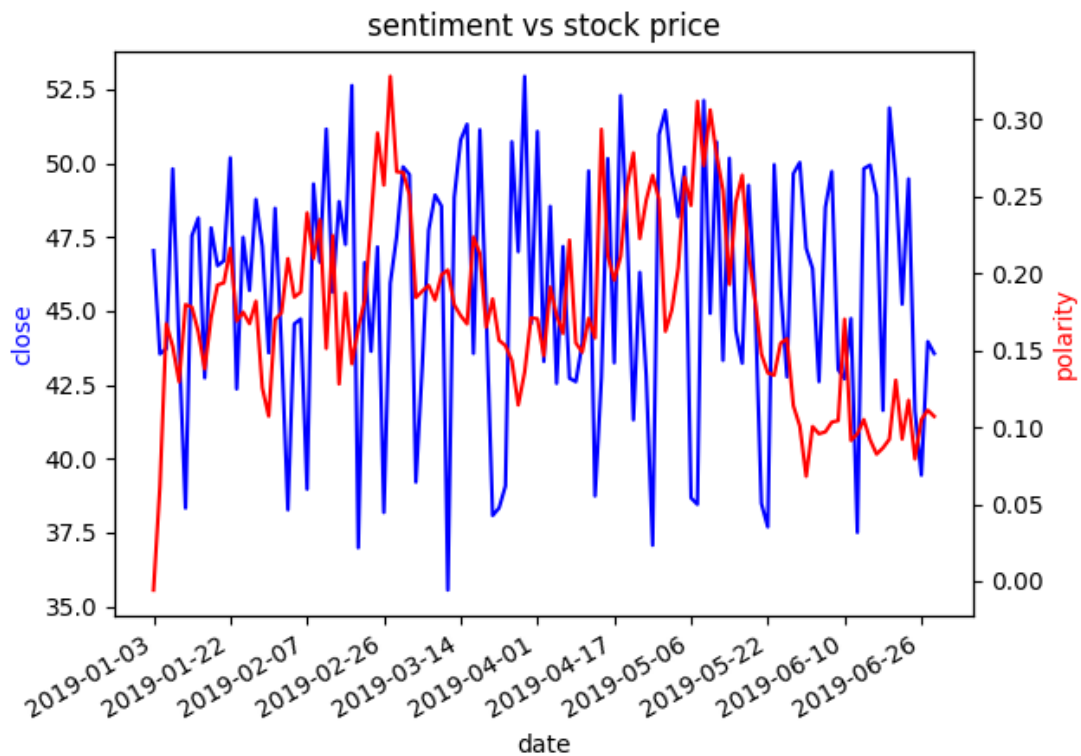


Figure 117 [twitter] today polarity vs today stock price

```
[covariance] no fourier      close  polarity
close      18.053754 -0.015243
polarity   -0.015243  0.003651
[correlation] no fourier    close  polarity
close      1.000000 -0.059368
polarity   -0.059368  1.000000
```

Figure 118 [twitter] covariance & correlation of today polarity vs today stock price

From the graph, we can barely see the relationship between stock price and tweets sentiment. It looks like there is nothing to do with Twitter's emotion and Apple stock price. The low covariance and correlation value also indicates the observation, ignoring the direction, with only 0.0152 covariance and 0.0593 correlation.

Then we perform next comparison on yesterday's polarity and today's stock price.

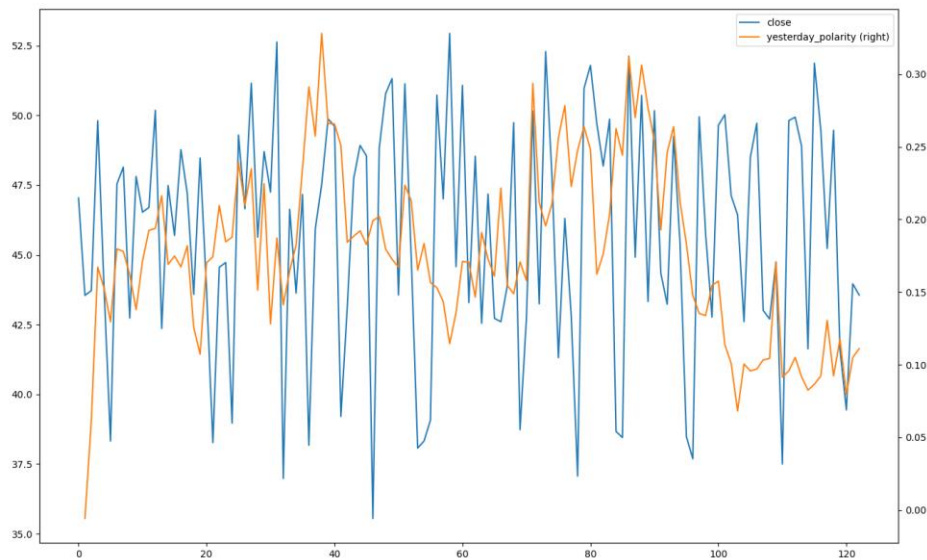


Figure 119 [twitter] yesterday polarity vs today stock price

```
close yesterday_polarity covariance:
                                close yesterday_polarity
close          18.053754         0.015503
yesterday_polarity 0.015503         0.003642
close yesterday_polarity correlation:
                                close yesterday_polarity
close          1.000000         0.060237
yesterday_polarity 0.060237         1.000000
```

Figure 120 [twitter] covariance & correlation of yesterday polarity vs today stock price

We can see from this comparison model, correlation value increased from 0.059 to 0.060. However this small improvement is negligible.

Then we perform another comparison on change in yesterday's polarity and change in stock price. We would like to see if the increase/decrease in yesterday's polarity value would lead to rise/fall in today's stock price or not.



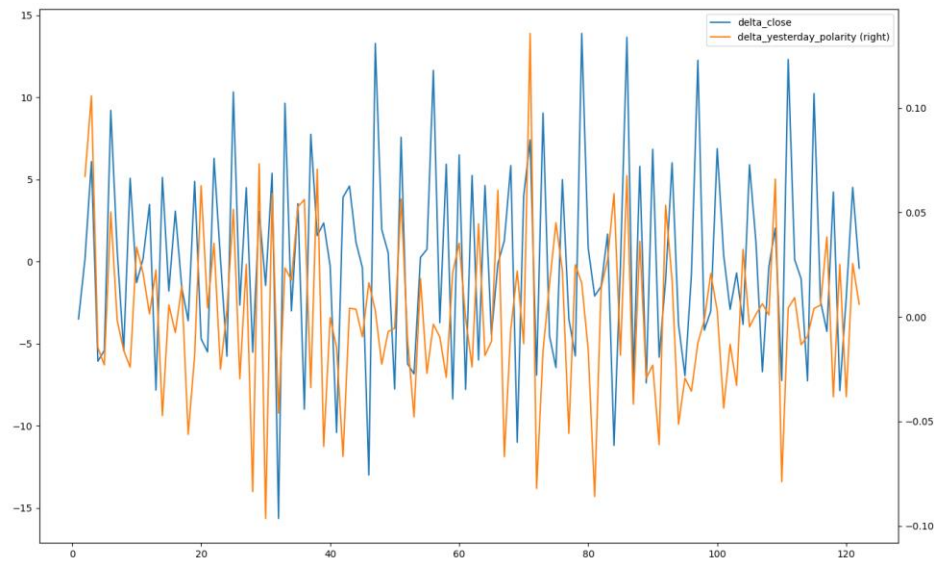


Figure 121 [twitter] delta yesterday polarity vs delta stock price

```

delta_close delta_yesterday_polarity covariance:
                                delta_close delta_yesterday_polarity
delta_close          37.705962          0.050172
delta_yesterday_polarity  0.050172          0.001629
delta_close delta_yesterday_polarity correlation:
                                delta_close delta_yesterday_polarity
delta_close          1.000000          0.201895
delta_yesterday_polarity  0.201895          1.000000

```

Figure 122 [twitter] covariance & correlation of delta yesterday polarity vs delta stock price

We can see that this comparison model gives us the best correlation result among the previous two result. The correlation value now becomes 0.201, which have more than 200% increase in correlation. This means that it is best to use change in yesterday's polarity to predict the change of today's stock price.

Although the above base experiments did not give us a shining result, we used the Fourier transform techniques from Reddit sentiment analysis to see if it helps predict stock price in long term. From Table 3, we pick test case No. 6 to examine because it has the highest covariance and correlation value. Below is the corresponding graph.

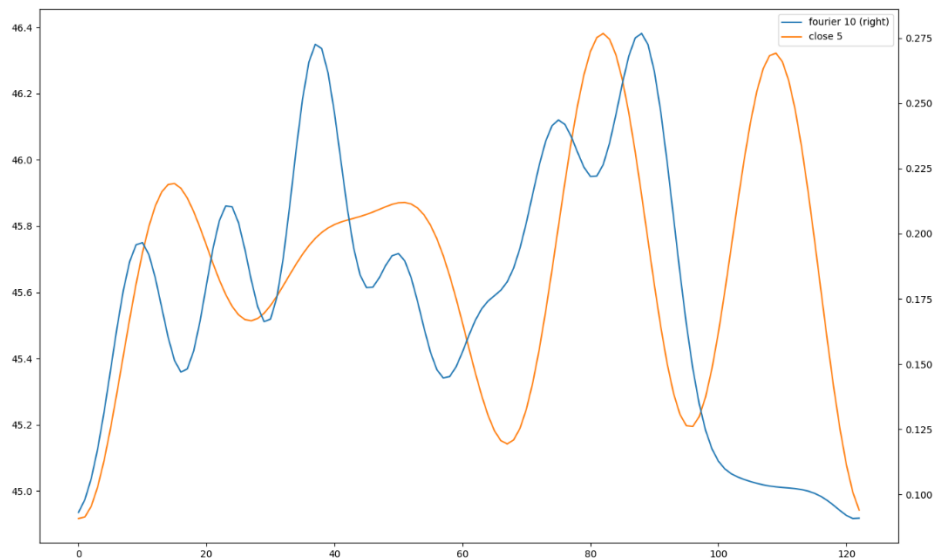


Figure 123 [fourier polarity 10, price 5] Twitter sentiment vs stock price

We can see from the graph, between 0-20 y-axis, the polarity (blue line) accurately predicts the rise of stock price (orange line). It also does a good job between the 50-100 y-axis, by correctly foresee the fall and rise of the stock price.

```
[covariance] fourier 10 stock 5      close 5  fourier 10
close 5      0.131916    0.004586
fourier 10    0.004586    0.002757
[correlation] fourier 10 stock 5      close 5  fourier 10
close 5      1.000000    0.240461
fourier 10    0.240461    1.000000
```

Figure 124 [fourier polarity 10, price 5] Twitter covariance and correlation

We can see from its covariance and correlation value, perform Fourier transformation indeed improves the prediction of the stock in the long term. This indicates that the trend of the Twitter sentiment value is likely to be the trend of rising/fall of Apple's stock price.

## 4.8 Reddit comment and Stock price

### 4.8.1 Data acquisition

The data we obtained is using a python library called PRAW. PRAW is a Reddit API wrapper that simple yet powerful to collect required data [19]. In this experiment, we get the data from 2018-09-01 to 2019-09-01 in the subreddit called r/wallstreetbet. We will be collecting comments from thread such as “Daily Discussion Thread”, “What Are Your Moves Tomorrow”, and individual post that related to AAPL.

### 4.8.2 Experiment result

We conducted the experiment in several ways in order to study the relationships between Reddit sentiment scores and stock prices. The following table sum up the result of the experiment.

Table 4 reddit experiment

	Fourier polarity	Fourier price	covariance	correlation
1	N/A (today)	N/A (today)	5.413715	0.068209
2	N/A (yesterday)	N/A (today)	5.726788	0.072307
3	N/A ( $\Delta$ today)	N/A ( $\Delta$ today)	3.453137	0.19473
4	N/A ( $\Delta$ yesterday)	N/A ( $\Delta$ today)	-0.148415	-0.008348
5	5	N/A	0.791039	0.030308
6	10	N/A	2.285104	0.065746
7	15	N/A	2.628963	0.069038
8	20	N/A	3.745961	0.087944

We perform calculation of covariance and correlation value to see the relationship of each comparison. After that, we performed fourier transform to make the graph into something that is easier to analysis. In this particular case, we will perform inverse fourier transform to convert the real life data (time domain) to a easy to analysis sinusoids (frequency domain). In addition, we have a parameter to zero-fill at the middle to get a different analysis point-of-view of the graph.

We first look at the base case, which is today's polarity vs today's stock price. Below is the corresponding graph.

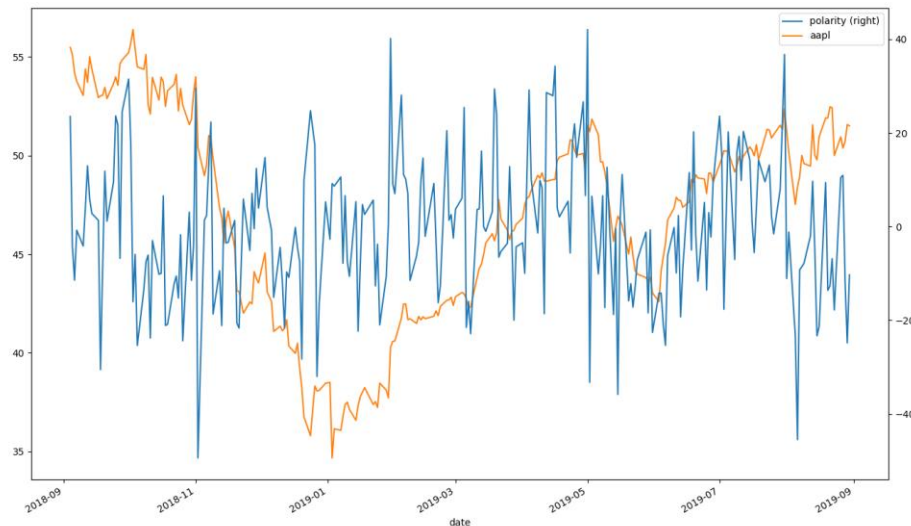


Figure 125 [reddit] today polarity vs today stock price

```
no fourier covariance:
      aapl  polarity
aapl    26.927363  5.413715
polarity  5.413715 233.943454
no fourier correlation:
      aapl  polarity
aapl    1.000000  0.068209
polarity 0.068209  1.000000
```

Figure 126 [no fourier] Reddit covariance and correlation

The graph didn't show much information as it is hard to identify a trend. However, from the covariance and correlation, it has 5.413715 in covariance and 0.0628 in correlation. The relationship of the polarity and stock price are not obvious as the values are too low.

Then we perform comparison on yesterday's polarity and today's stock price, to see if it have a better performance.

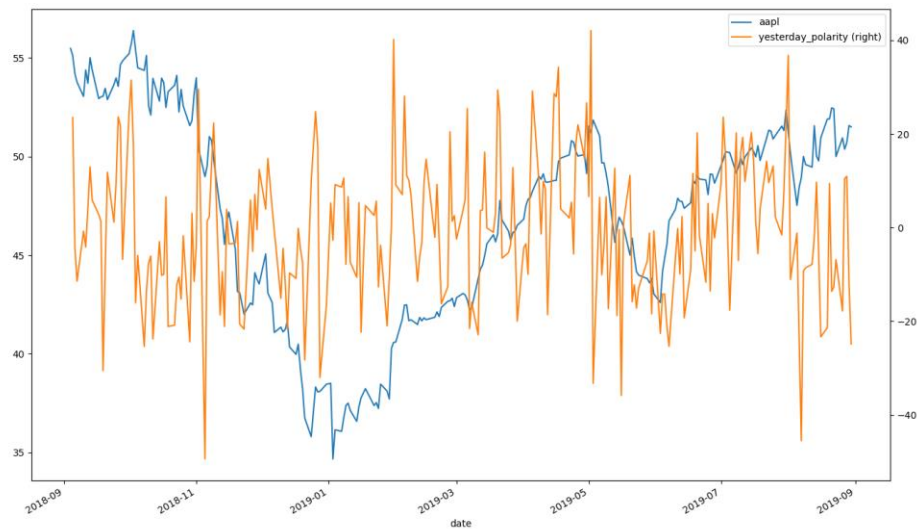


Figure 127 [reddit] yesterday polarity vs today stock price

```
aapl yesterday_polarity covariance:
aapl yesterday_polarity
aapl      26.927363      5.726788
yesterday_polarity  5.726788      234.514172
aapl yesterday_polarity correlation:
aapl yesterday_polarity
aapl      1.000000      0.072307
yesterday_polarity  0.072307      1.000000
```

Figure 128 [reddit] covariance & correlation of yesterday polarity vs today stock price

This comparison model wants to change the factor of today's polarity to yesterday's polarity to see if it improves the result. However, looking from the correlation value, it only slightly increase from 0.0682 to 0.0723. Since the magnitude of the correlation value is low, we did not see a obvious relationship between yesterday's polarity and today's stock price.

Then we performed another comparison model on change in today's polarity and change in today's stock price to see if increase/decrease in today's polarity would lead to rise/fall in today's stock price or not.

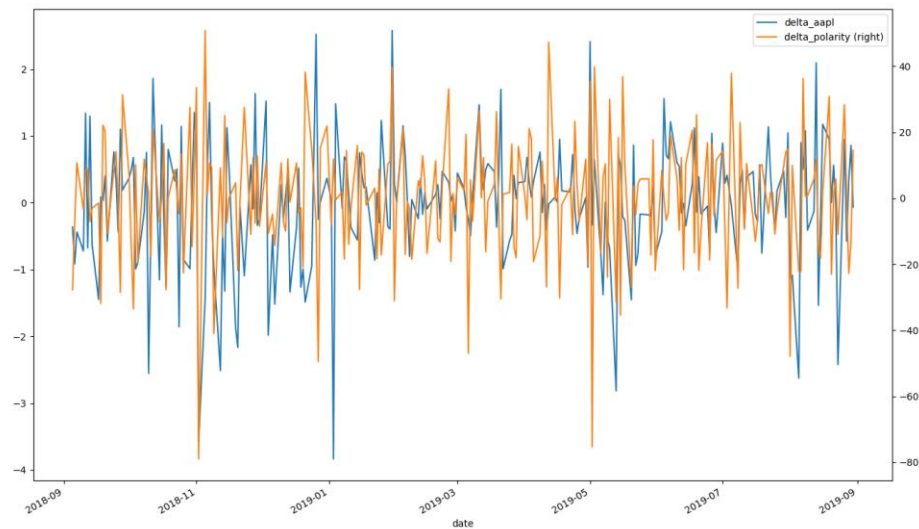


Figure 129 [reddit] delta today polarity vs delta today stock price

```

delta_aapl delta_polarity covariance:
              delta_aapl delta_polarity
delta_aapl      0.873541      3.453137
delta_polarity   3.453137     359.978426
delta_aapl delta_polarity correlation:
              delta_aapl delta_polarity
delta_aapl      1.00000      0.19473
delta_polarity   0.19473      1.00000

```

Figure 130 [reddit] covariance & correlation of delta today polarity vs delta today stock price

We can see from the correlation value, it increased from 0.0682 to 0.194, which is over 180% increase in value. This is also the highest correlation value among the previous two comparison model. This means it is best to use change in today's polarity to predict the change in today's stock price.

Next, we are going to examine all the Fourier transformation graph. Below is the 4 Fourier Transformation graph.

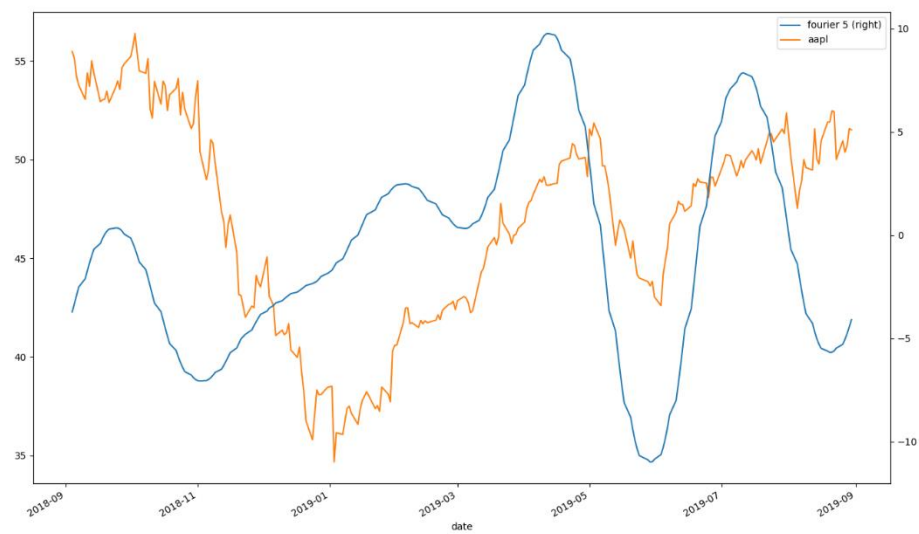


Figure 131 [fourier 5] Reddit sentiment vs stock price

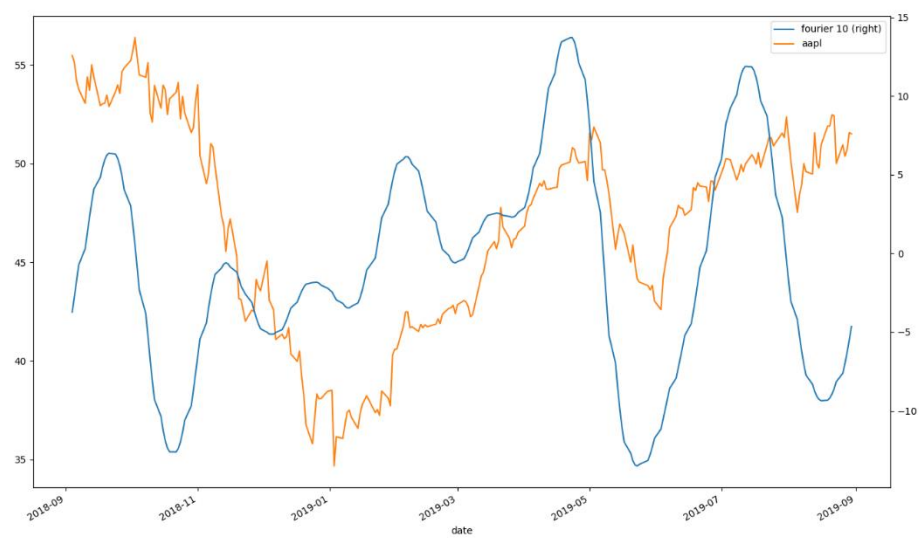


Figure 132 [fourier 10] Reddit sentiment vs stock price

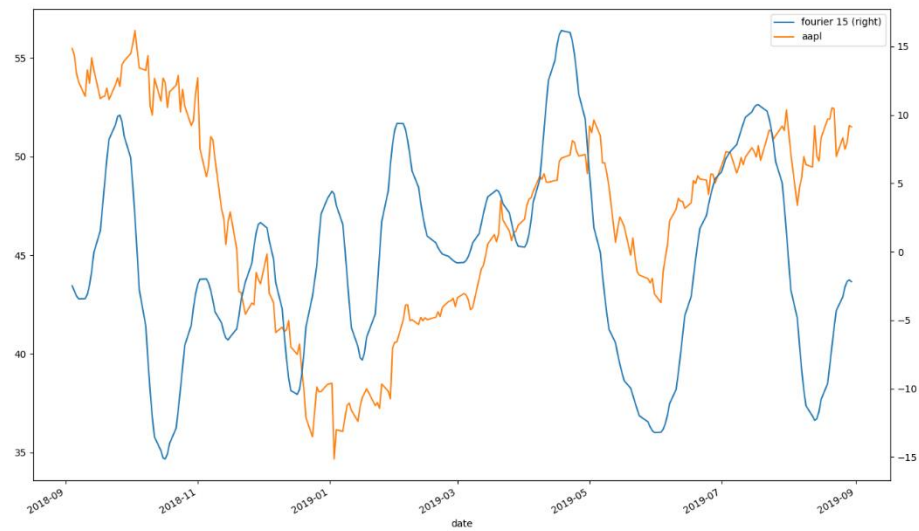


Figure 133 [fourier 15] Reddit sentiment vs stock price

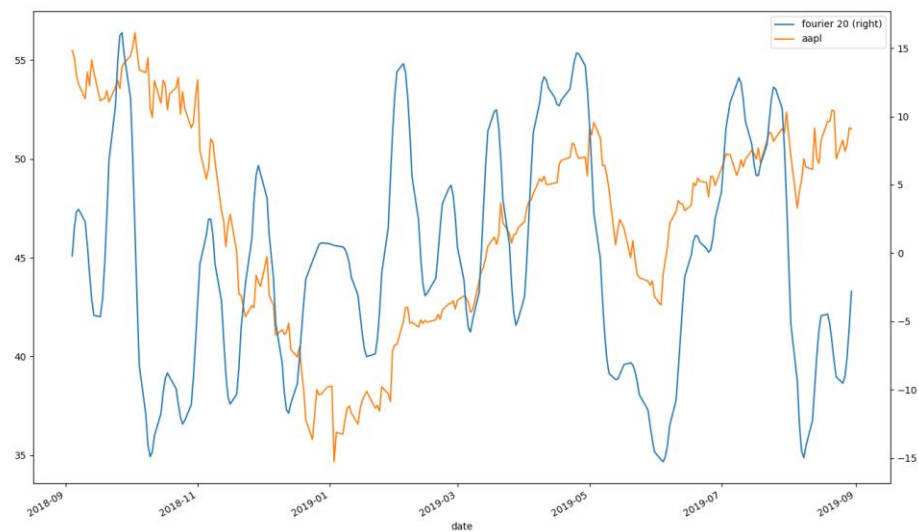


Figure 134 [fourier 20] Reddit sentiment vs stock price

According to the above Fourier graph, the graph with Fourier 5 transformation introduced a brief trend between Reddit sentiment and Apple's stock price. The graph with Fourier 10 and Fourier 15 transformation indicated a more precise relation of



Reddit sentiment and Apple's stock price. For example, using Fourier 10 and Fourier 15, we can see a line having relatively similar movement between 2019-01 to 2019-05. If an investor wants a brief indication, using Fourier 10 and Fourier 15 transformation graph is enough to give them the required information. The result of today's Reddit sentiment value is likely to be the rise/fall of tomorrow's Apple stock price. Besides, the Fourier 20 graph gives the most detailed and precise indication of the relation between Reddit sentiment and Apple's stock price, so as the covariance and correlation value.

```
fourier 5 covariance:
      aapl  fourier 5
aapl    26.927363  0.791039
fourier 5  0.791039  25.298112
fourier 5 correlation:
      aapl  fourier 5
aapl    1.000000  0.030308
fourier 5  0.030308  1.000000
```

Figure 135 [fourier 5] Reddit covariance and correlation

```
fourier 10 covariance:
      aapl  fourier 10
aapl    26.927363  2.285104
fourier 10  2.285104  44.862678
fourier 10 correlation:
      aapl  fourier 10
aapl    1.000000  0.065746
fourier 10  0.065746  1.000000
```

Figure 136 [fourier 10] Reddit covariance and correlation

```
fourier 15 covariance:
      aapl  fourier 15
aapl    26.927363  2.628963
fourier 15  2.628963  53.851098
fourier 15 correlation:
      aapl  fourier 15
aapl    1.000000  0.069038
fourier 15  0.069038  1.000000
```

Figure 137 [fourier 15] Reddit covariance and correlation

```
fourier 20 covariance:
      aapl  fourier 20
aapl      26.927363    3.745961
fourier 20  3.745961    67.378216
fourier 20 correlation:
      aapl  fourier 20
aapl      1.000000    0.087944
fourier 20  0.087944    1.000000
```

Figure 138 [fourier 20] Reddit covariance and correlation

## 5. Tool development

### 5.1 Tool Overview

This tool is composed of all the features we have done on this project, including the stock prediction model we did in semester one, pattern recognition, and sentiment analysis for different social media, YouTube, Twitter, and Reddit. And the one last thing is a small backtest, test how much can we earn with the above features.

### 5.2 Stock Pattern recognition

The first feature is pattern recognition. User can input a start date and an end date. It will try to fetch the data in local data storage. We prepare the Apple stock from 2011 to 2019. If the inputted period is not in this range, the program will get the data in real-time. After it has fetched the data, the data will be inputted to the LSTM pattern recognition model and get a result.

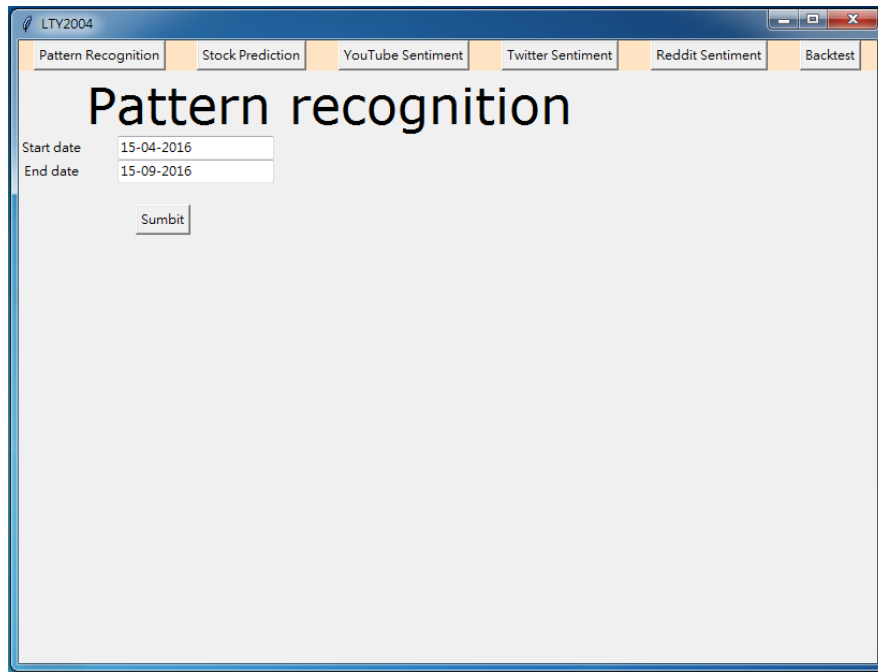


Figure 139: Pattern recognition interface

After we click submit, the model will give its prediction. In this case, the result is positive with a possibility of 0.655. The three lines in the graph are plotted by the hard-coded pattern recognizer to show where is the location of the double bottom pattern.

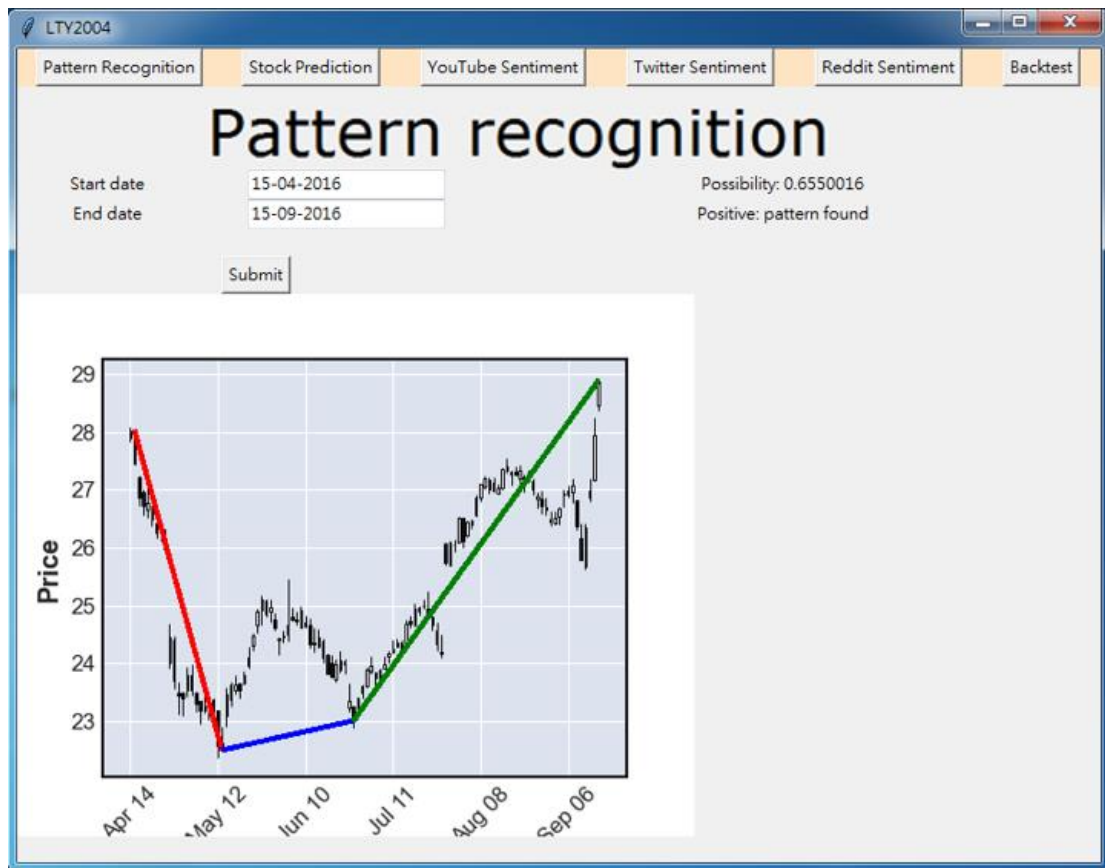
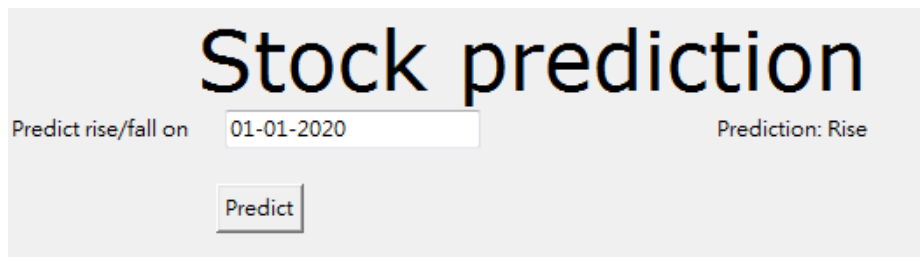


Figure 140: Pattern recognition result

## 5.3 Stock Prediction

The second feature is the stock prediction model. User can input data to the model, the model will return a prediction of stock movement on that date. We prepared the data from 2010 to March of 2021. If the input is not in this range, the program will automatically download the data in real-time, which may take a while to finish. For example, we want to predict 2020-01-01. The model predicts it will rise on that date.

A web interface for stock prediction. At the top, the title "Stock prediction" is displayed in a large, bold, black font. Below the title, on the left, is the text "Predict rise/fall on" followed by a text input field containing "01-01-2020". To the right of the input field, the text "Prediction: Rise" is displayed. Below the input field, there is a button labeled "Predict".

Stock prediction

Predict rise/fall on  Prediction: Rise

Figure 141: Prediction result

## 5.4 YouTube Sentiment Analysis

The third feature is YouTube sentiment analysis. We prepared the YouTube comments for the video posted from 2019 January to 2019 June. For another period, the program will need to download the comments data in real-time.

LTV2004

Pattern Recognition Stock Prediction YouTube Sentiment Twitter Sentiment Reddit Sentiment Backtest

# YouTube sentiment analysis

Start date 01-01-2019  
end date 30-06-2019

Submit

Figure 142: YouTube sentiment analysis interface

The page will show four graphs, listed as below:

- Close vs polarity
- Change in close vs change in sentiment
- Change in close vs yesterday sentiment
- Change in close vs change in yesterday sentiment

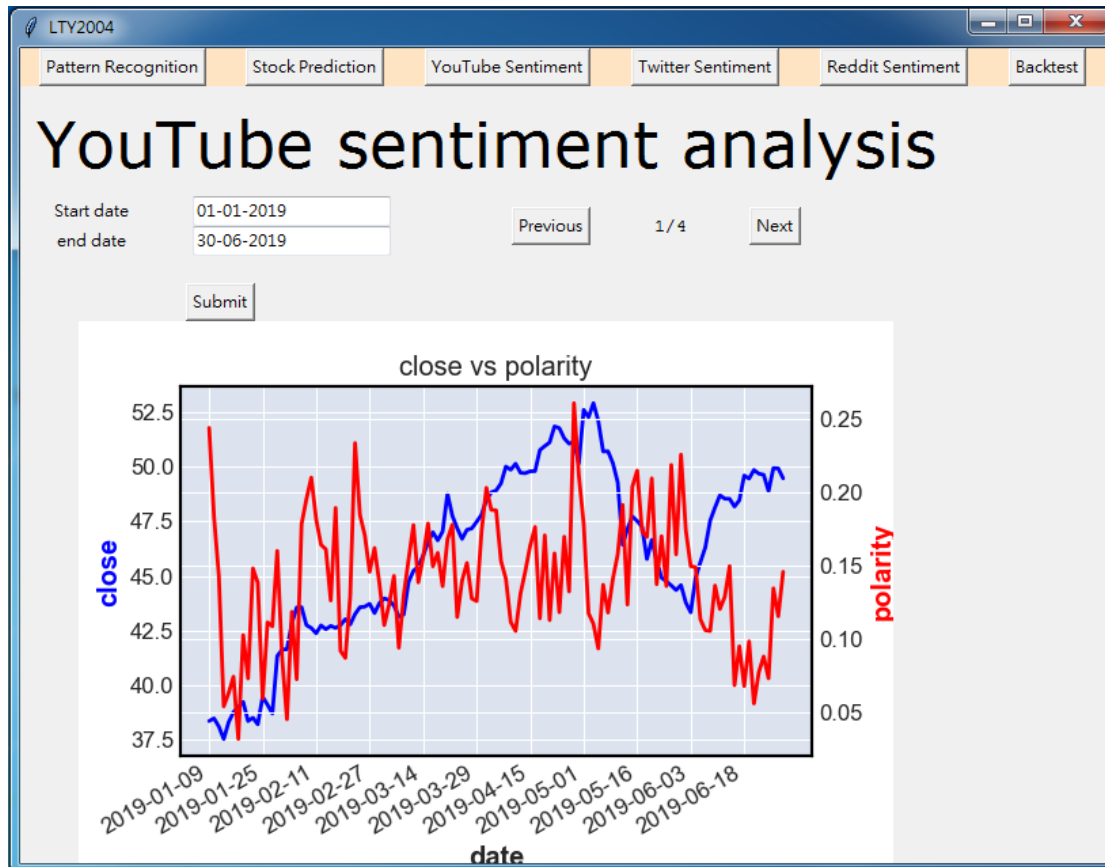


Figure 143: Result of YouTube sentiment page

## 5.5 Twitter Sentiment Analysis

The fourth feature is Twitter Sentiment Analysis. We implement the data crawling process using twint for its instantaneity and up-to-date information, and then analyse the sentiment using VARDAR sentiment analyzer for simplicity. User can input their preferred time interval to get the corresponding data from that date range.

LTY2004

Pattern Recognition Stock Prediction YouTube Sentiment **Twitter Sentiment** Reddit Sentiment Backtest

# Twitter sentiment

Start date 2019-01-01  
End date 2019-06-30

Submit  
[DEBUG]Plot

Figure 144 UI of Twitter sentiment analysis

After clicking the “submit” button, the program will crawl the Twitter tweets in real-time and plot the corresponding graph. Below is the graph features from 2019-01 to 2019-06.

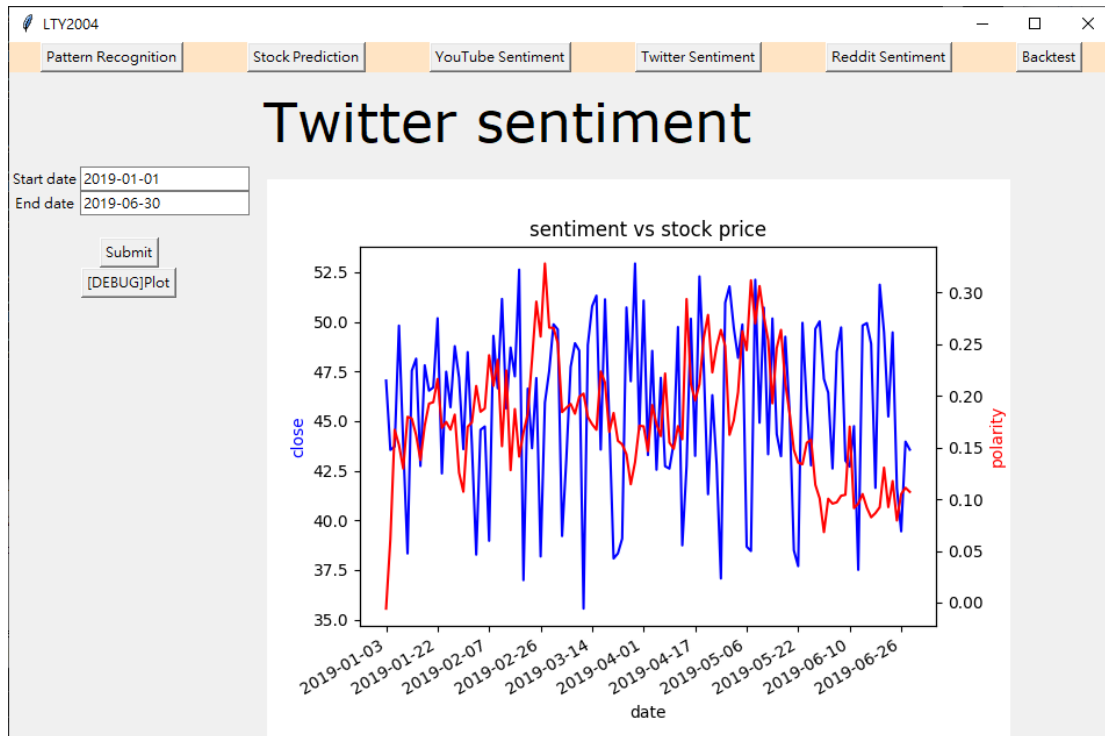


Figure 145 result of Twitter sentiment analysis

This page consists of multiple graph representation. Including a graph of the base case, the graph of Fourier 5 transformation for polarity and Fourier 10 transformation for stock price, etc. Complete the list of graphs please refer to table 2.



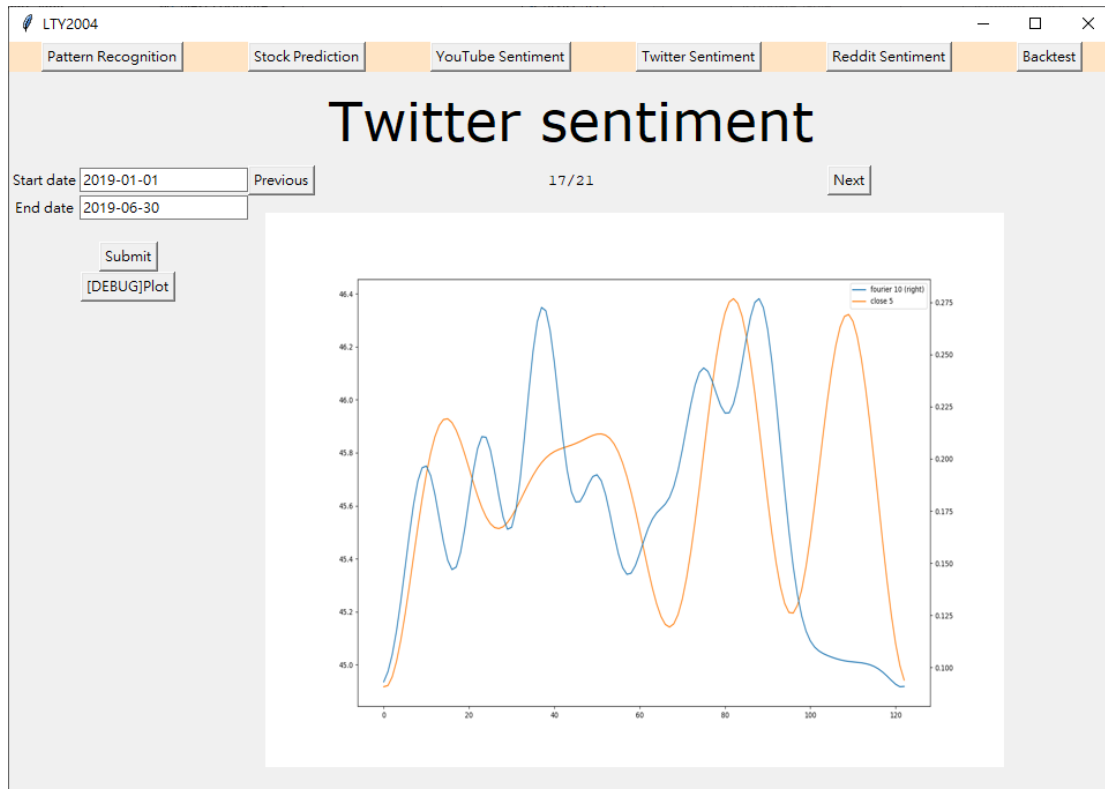


Figure 146 Example of multiple graph representation

## 5.6 Reddit Sentiment Analysis

The fifth feature is Reddit Sentiment Analysis. We implement the data crawling process by using PRAW for its simple to use and fast characteristic, and then analyse the sentiment using VARDAR sentiment analyzer for simplicity. User can input their preferred time interval to get the corresponding data from that date range.

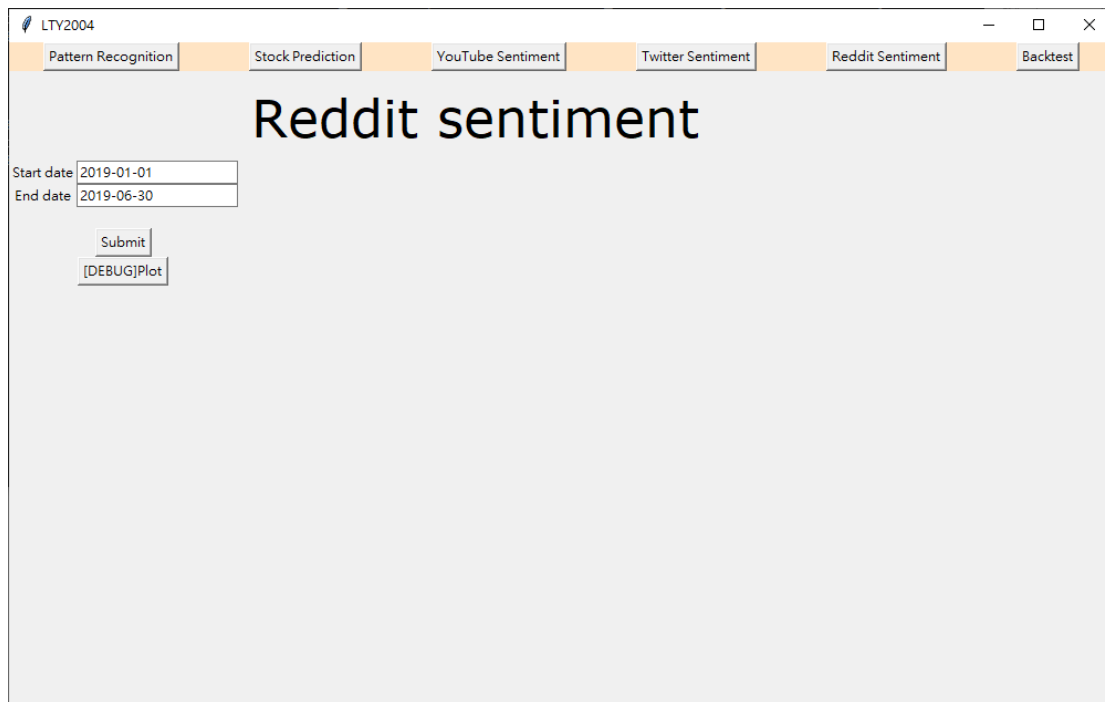


Figure 147 UI of Reddit sentiment analysis

After clicking the “submit” button, the program will crawl the Reddit comments in real-time and plot the corresponding graph. Below is the graph features from 2018-09 to 2019-09.

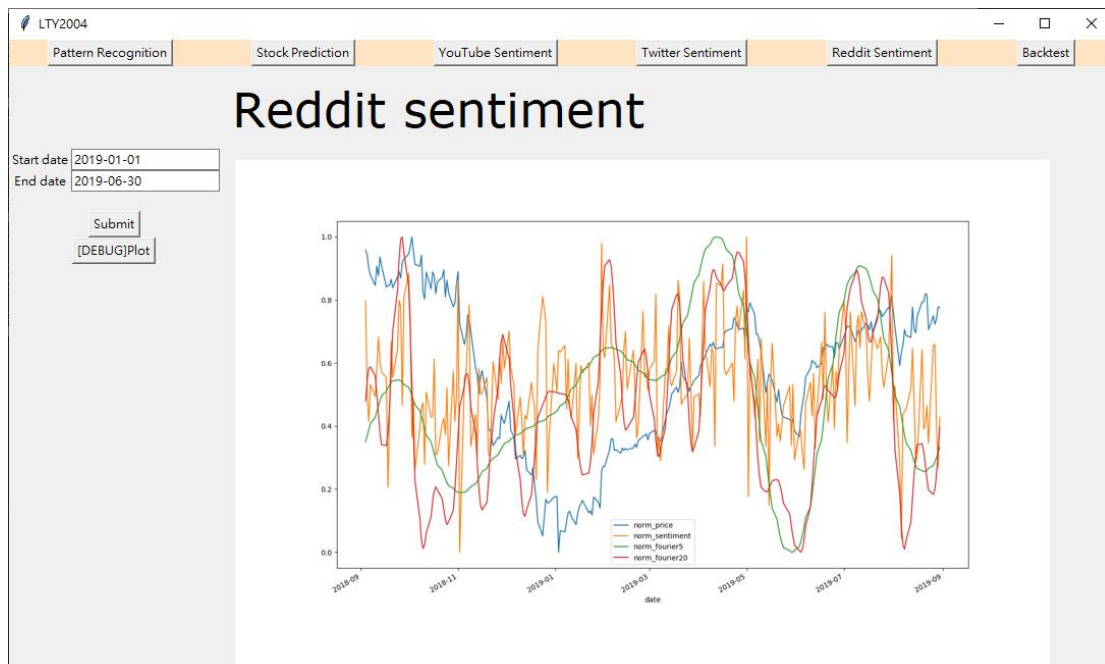


Figure 148 result of Reddit sentiment analysis

This page consists of multiple graph representation. Including a graph of the base case, the graph of Fourier 5 transformation for polarity, etc. Complete the list of graphs please refer to table 4.

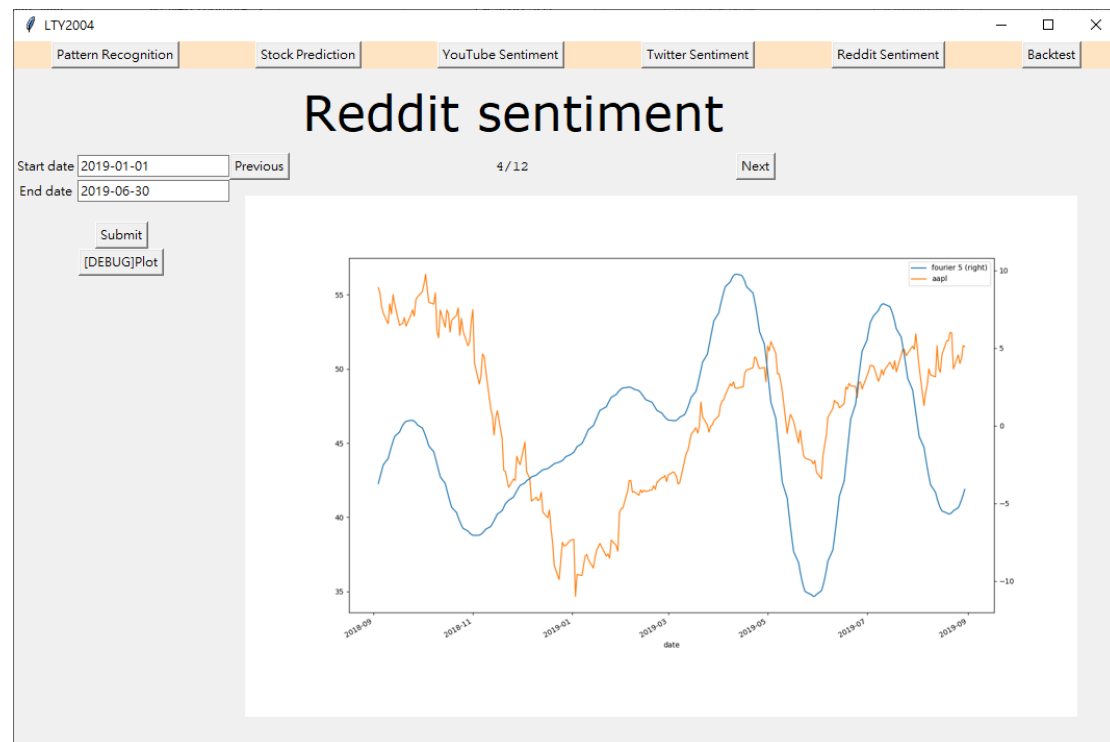


Figure 149 Example of multiple graph representation

## 5.6 Backtest

The backtest we performed in this section is not a complete backtest. The purpose of this part is to test the outcome of using the above strategies in investment. The first strategy is using the model prediction model we trained in semester one. The program will buy one share of AAPL if the model predicts it will rise on the next day, or sell all holding shares if the model predicts it will fall on the next day. Trading stock from 2019-01-01 to 2019-06-30, the overall profit is 46 dollar.

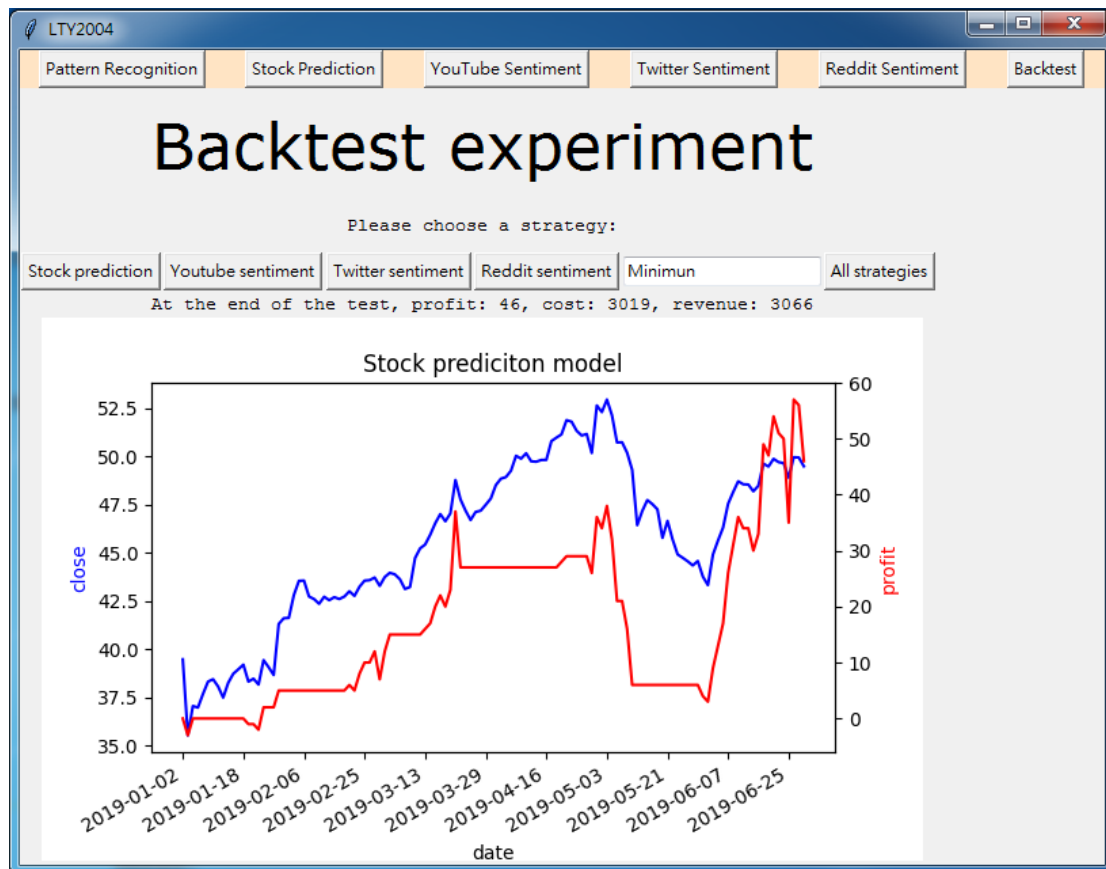


Figure 150: Using stock prediction model in backtesting

The second strategy is using YouTube sentiment in the backtest. If the average sentiment of that day is positive, then the program will buy one share. Otherwise, it will sell all the share held. The result is 16 dollar.

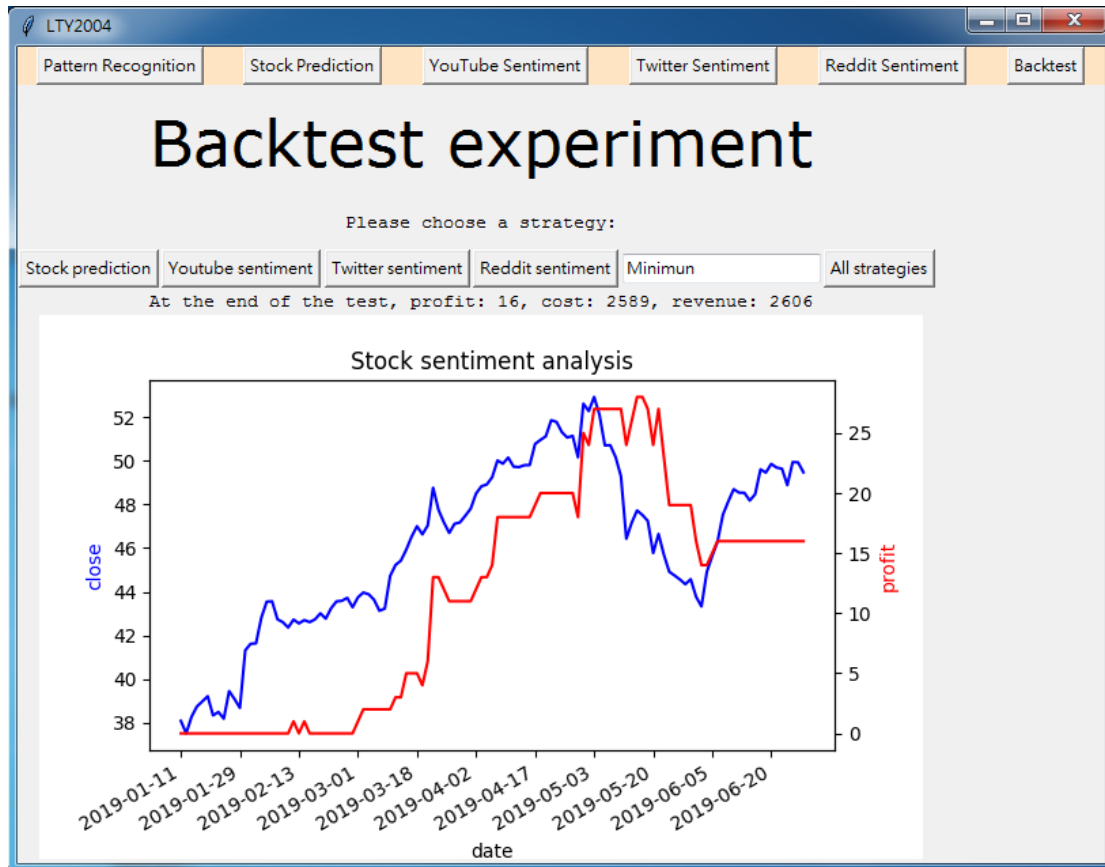


Figure 151: Using Youtube sentiment in backtesting

The last strategy is to combine all the strategies before. User can input a number from 1 to 4. For example, if the user input is 3, the program will buy one share of at least three positives from the stock prediction model, YouTube sentiment, Twitter sentiment and Reddit sentiment. Otherwise, it will sell all the shares. The result is listed in the following table.

Minimum number of positive	Overall profit
4	11
3	23
2	-21
1	-1217

Table 5 backtesting

A: profit is 11 dollar

At least three positives: profit is 23 dollar

At least two positives: profit is -21 dollar

At least one positive: profit is -1217 dollar

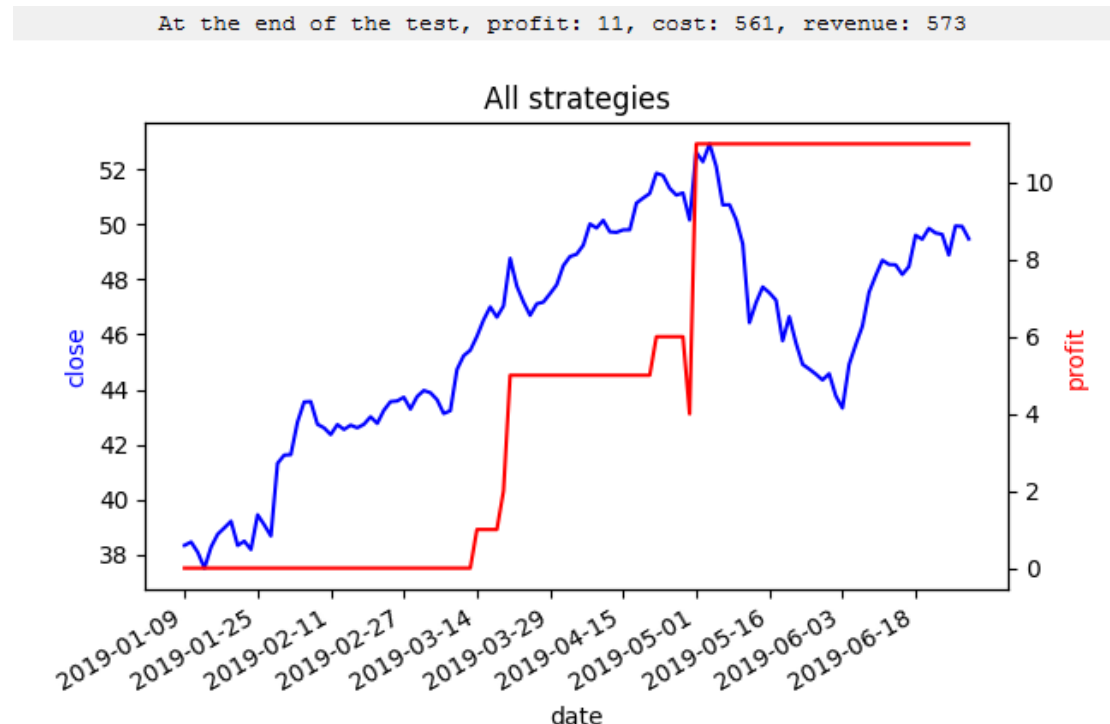


Figure 152: All positives

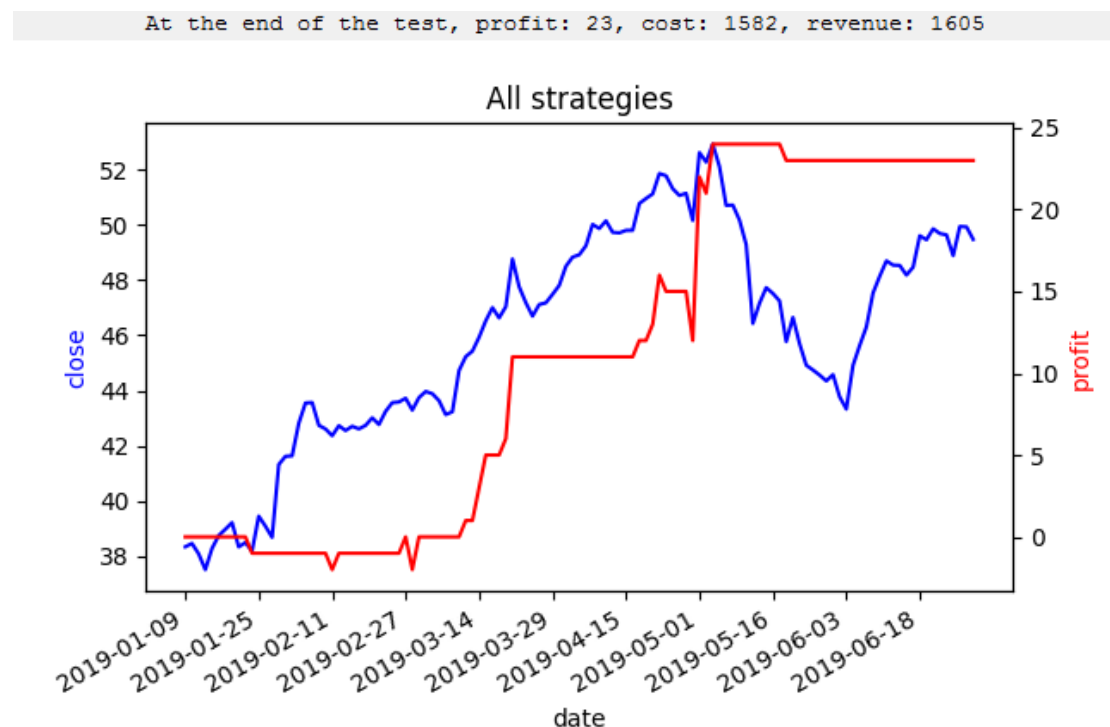


Figure 153: At least three positives

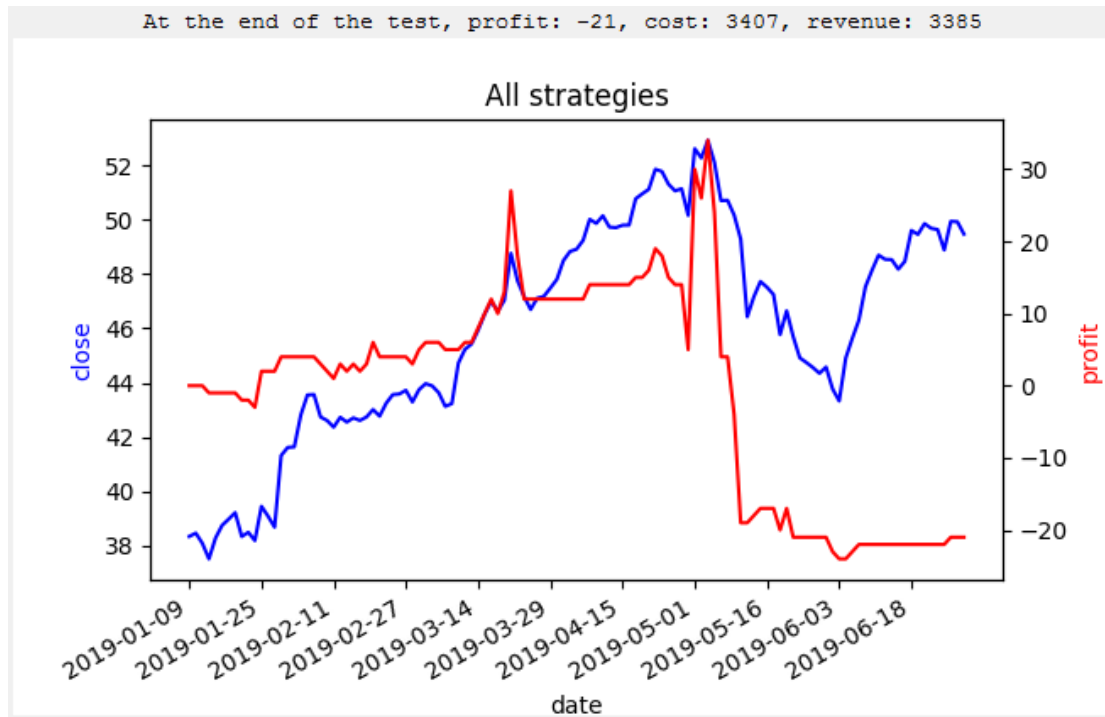


Figure 154: At least two positives

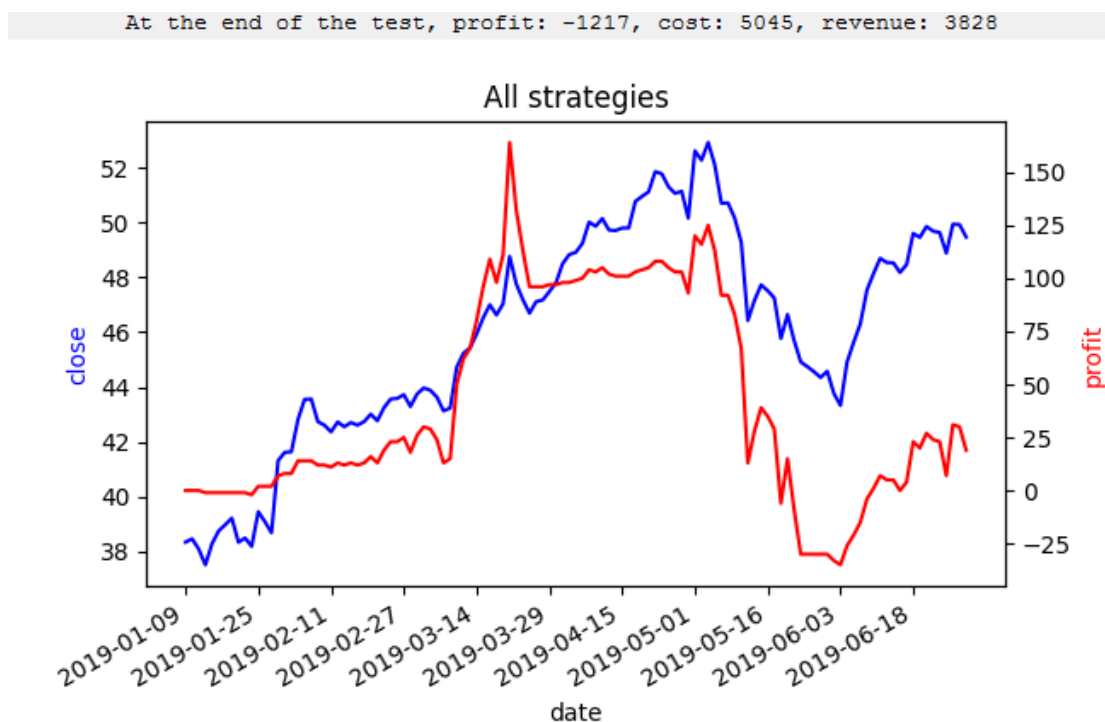


Figure 155: At least one positive

## 6. 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 textual analysis, we use TextBlob, VADAR Sentiment, Artificial Neural Network (ANN) and Bidirectional Encoder Representations from Transformers (BERT). Each of those techniques has its own pros and cons, it depends on the use case and environment. There is no perfect solution for either stock price prediction or sentiment analysis. All we can do is use the right method at the right place, at the right time.

In the aspect of numerical analysis, all the prediction of these models shows 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.

Sentiment data for general news and financial news are different. These two models cannot directly apply to each other. For example, if we train an LSTM model using financial news data, and apply prediction to general news. The polarity score may become strange or may even affect the prediction accuracy. Unfortunately, there is only financial news dataset that is pre-labelled with a 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 are able to distinguish 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 the market’s emotion and its trend.



In pattern recognition, the main reason that our model doesn't give us a good result is mainly because of the dataset. As we couldn't find a double bottom pattern dataset for machine learning, we need to build it on our own. The quality of our hard-code pattern recognizer is not good enough, turned out we had only 234 samples. It is not enough to train the model. Therefore, we cannot take the advantage of a stock pattern in this project.

In the sentiment analysis on social media, three models performed differently. For Reddit sentiment and Twitter sentiment, their correlation with a stock price trend is not obvious. We could not see it with naked eyes. However, this does not mean it is not suitable for predicting the stock price trend. Take Twitter sentiment as an example, after the Fourier transformation, we can also see there indeed have a relationship between Twitter sentiment and stock price for the long term. This also applies to Reddit sentiment, after a Fourier transformation, we can see there is an obvious correlation between the polarity and stock price using our naked eyes. So in conclusion, sentiment analysis on social media did have a correlation between stock price, and we can surely use this relationship to perform a prediction of the rise/fall of stock price.

# References

- [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] Jahidul Arafat, Mohammad Ahsan Habib,Rajib Hossain, "Analyzing Public Emotion and Predicting Stock Market Using Social Media," in *American Journal of Engineering Research (AJER)*, 2013.
- [9] M. Y. Tahir M. Nisar, "Twitter as a tool for forecasting stock market movements:," in *The Journal of Finance and Data Science* 4, 2018.
- [10] J. CHEN, "investopedia," 8 Jan 2021. [Online]. Available: <https://www.investopedia.com/terms/d/doublebottom.asp>. [Accessed 11 Apr 2021].
- [11] C. Hind, "samurai trading academy," [Online]. Available: <https://samuraitradingacademy.com/7-best-price-action-patterns/>. [Accessed 11 Apr 2021].

- [12] 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.
- [13] Mehar Vijha, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," in *Procedia Computer Science* 167, 2020.
- [14] 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.
- [15] A. Tipirisetty, "Stock Price Prediction using Deep Learning," in *Master's Projects*. 636, 2018.
- [16] Jahidul Arafat, Mohammad Ahsan Habib and Rajib Hossain, "Analyzing Public Emotion and Predicting Stock Market Using Social Media," in *American Journal of Engineering Research*, 2013.
- [17] Anshul Mittal, Arpit Goel, "Stock Prediction Using Twitter Sentiment Analysis," in *Stanford University Working Paper*, 2012.
- [18] Wikipedia, "Aarchive Team," 2021. [Online]. Available: [https://en.wikipedia.org/wiki/Archive\\_Team](https://en.wikipedia.org/wiki/Archive_Team).
- [19] "PRAW," [Online]. Available: <https://github.com/praw-dev/praw>.
- [20] Nazri Mohd Naw, Walid Hasen Atomi, M. Z. Rehman , "The Effect of Data Pre-Processing on Optimized Training of Artificial Neural Networks," in *Procedia Technology* 11, 2013.
- [21] Lexalytics Inc, "Sentiment Analysis Explained," 2020. [Online]. Available: <https://www.lexalytics.com/technology/sentiment-analysis>.
- [22] Wikipedia, "Sentiment analysis," Wikipedia, 2020. [Online]. Available: [https://en.wikipedia.org/wiki/Sentiment\\_analysis](https://en.wikipedia.org/wiki/Sentiment_analysis). [Accessed 2020].
- [23] MonkeyLearn Inc, "Sentiment Analysis: A Definitive Guide," 2020. [Online]. Available: <https://monkeylearn.com/sentiment-analysis/>.
- [24] C. Olah, "Understanding LSTM Networks," 27 08 2015. [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

- [25] 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.
- [26] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, Yoshua Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," in *arXiv*, 2014.
- [27] T. Cover, P. Hart, "Nearest neighbor pattern classification," in *IEEE Transactions on Information Theory*, vol. 13, 1967.
- [28] Sean J. Taylor, Benjamin Letham, "Forecasting at scale," 2017.
- [29] GeekforGeeks, "Clustering in Machine Learning," 23 February 2020. [Online]. Available: <https://www.geeksforgeeks.org/clustering-in-machine-learning/>. [Accessed 2020].
- [30] Wikipedia, "Cluster analysis," 2020. [Online]. Available: [https://en.wikipedia.org/wiki/Cluster\\_analysis](https://en.wikipedia.org/wiki/Cluster_analysis).
- [31] C. Hutto, "VADER-Sentiment-Analysis," 2020. [Online]. Available: <https://github.com/cjhutto/vaderSentiment>. [Accessed 2020].
- [32] A. Beri, "SENTIMENTAL ANALYSIS USING VADER," [Online]. Available: <https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>. [Accessed 2020].
- [33] R. Malde, "A Short Introduction to VADER," 8 June 2020. [Online]. Available: <https://towardsdatascience.com/an-short-introduction-to-vader-3f3860208d53>. [Accessed 2020].
- [34] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Atlanta, CA, 2014.
- [35] S. Loria, "TextBlob: Simplified Text Processing," 2020. [Online]. Available: <https://github.com/slوريا/TextBlob>. [Accessed 2020].
- [36] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, "Deep contextualized word representations," in *arXiv:1802.05365*, 2018.
- [37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in

*arXiv:1810.04805*, 2018.

- [38] P. Werbos, "Backpropagation through time: what it does and how to do it," in *Proceedings of the IEEE Vol:78*, 1990 .
- [39] "Fourier transform," Wikipedia, 2021. [Online]. Available: [https://en.wikipedia.org/wiki/Fourier\\_transform](https://en.wikipedia.org/wiki/Fourier_transform).
- [40] N. S. Chauhan, "Fourier Transformation for a Data Scientist," Feb 2020. [Online]. Available: <https://www.kdnuggets.com/2020/02/fourier-transformation-data-scientist.html>.
- [41] Wikipedia, "Jenks natural breaks optimization," 2020. [Online]. Available: [https://en.wikipedia.org/wiki/Jenks\\_natural\\_breaks\\_optimization](https://en.wikipedia.org/wiki/Jenks_natural_breaks_optimization). [Accessed 2020].
- [42] Warren S. McCulloch, Walter Pitts, "A logical calculus of the ideas immanent in nervous activity," in *Bulletin of Mathematical Biophysics* 5, 1943.
- [43] "twint," [Online]. Available: <https://github.com/twintproject/twint>.
- [44] "tweepy," [Online]. Available: <https://github.com/tweepy/tweepy>.
- [45] A. Salač, "Forecasting of the cryptocurrency market," The Netherlands.
- [46] R. B. L. H. Alec Go, "Twitter Sentiment Classification using Distant Supervision".
- [47] S. C. Q. T. H. N. C. Tien Thanh Vu, "An Experiment in Integrating Sentiment Features for Tech," 2012.