



SENIOR RESEARCH

Topic: The predictability of Stock Exchange of Thailand index using text mining on news articles

Name: Chanyanit Hanthongkam

5845946329

Advisor: Asst. Prof. Dr. Yong Yoon, Ph.D.

Date: December 13, 2018

Senior Research Submitted in Partial Fulfillment of the Requirements  
for the Bachelor of Arts degree in Economics (International Program)

The Bachelor of Arts Program in Economics

Faculty of Economics

Chulalongkorn University

Academic Year 2018

Approve

---

(Assoc. Prof. Sothitorn Mallikamas, Ph.D.)

Chairman

Date of Approval \_\_\_\_\_

## **Abstract**

Stock market prediction has been a topic of interest in the field of finance. Despite commonly-used traditional approaches which mostly rely on historical time-series and financial variables, text mining approach has continually become an emerging technique for stock price forecasting. Given its ability in extracting unstructured data and transforming it into numerical values, the technique could provide some additional insights. The fact that there exist only a few studies based on this approach in emerging financial markets triggered the researcher's curiosity to explore in the context of Thailand's stock market.

This research has investigated the predictability of the Stock exchange of Thailand index (SET50) using this text mining on online news articles. A review on the literature suggests that there is a variation in methods and approaches used under this technique. In contrast to previous studies which mostly focuses on only one particular approach, this research applied four supervised machine learning algorithms, including decision tree, Naïve Bayes, k-NN and SVM, to construct a predictive model. Their forecasting results are evaluated among three different feature selections, which are TF-IDF vector representation, sentimental analysis, and the combination of the two former approaches.

The study demonstrated that sentimental analysis yields the least preferred performance among all three proposed approaches; nevertheless, it is still disputing which one of the others performs best as their rank tends to vary across algorithms. Regarding the forecasting powers of algorithms, since they also differ across approaches, there is no conclusion for a single algorithm that gives the highest performance in every approach. However, the evaluation based on accuracy measure suggests that SVM produces most precise prediction in the both TF-IDF frequency and sentimental analysis approach whereas k-NN has the best forecasting ability in the last approach. Moreover, a comparison with existing literatures reveals that the proposed approaches generate a relatively better performance.

## **Acknowledgement:**

This research is partially supported by Asst. Prof. Dr. Yong Yoon, Ph.D. whom I would like to express gratitude to and attribute the completion of the research to his guidance, supervision and patience.

## Table of Contents

1. Introduction.....	4
2. Literature Review.....	5
3. Data.....	6
4. Methodology.....	7
4.1) Data Preparation.....	8
4.1.1) Obtaining market data.....	8
4.1.2) Data Labelling.....	8
4.1.3) Extracting news articles.....	9
4.1.4) Text cleaning.....	9
4.1.5) News and prices alignment.....	9
4.2) Text Preprocessing.....	10
4.2.1) Tokenization.....	11
4.2.2) Transforming cases.....	11
4.2.3) Filtering Stopwords.....	11
4.2.4) Filtering tokens by lengths.....	12
4.2.5) Stemming.....	12
4.2.6) Sentimental Analysis.....	12
4.3) Applying machine learning techniques.....	13
4.3.1) Applying machine learning algorithms.....	13
4.3.2) Evaluating the forecasting performance.....	18
5. Results.....	21
5.1 Results from news and price alignment.....	21
5.2 Word representation.....	22
5.3 Prediction of algorithms.....	23
5.4 Evaluation of algorithms.....	23
5.4.1) Evaluation using confusion matrix.....	23
5.4.2) Evaluation using ROC curve.....	26
6. Conclusion.....	27
Bibliography.....	28
Appendix A.....	30
Appendix B.....	31

## 1. Introduction

“We have long felt that the only value of stock forecasters is to make fortune-tellers look good.”— Warren Buffett

Stock market is one of the most appealing yet complexed financial markets in the world. It not only attracts over millions of investors across countries, but it also lures as many academics to understand its mechanism. Despite their numerous attempts to forecast the stock market movements, it is still a subject of controversy among these scholars and stock experts whether stock prices can be predictable or not. As demonstrated in a quote by Warren Buffet, even the world’s successful investor who makes a trillion-dollar fortune from the stock market still believes that stock market prediction is always poorly performed and therefore, should be avoided.

Regardless of being an ongoing debate, the topic of stock market prediction has been receiving much attention in recent years. Many methods were exploited in explaining the stock market behavior, a majority of them was based on either historical prices or related financial variables. Although forecasting stock prices based on these data often provides promising results, given that, nowadays, the world is overwhelmed by a massive flow of inaccessible information, being able to access a great volume of unstructured data could provide some additional insights. Thanks to rapid technological advances in computer science, a so-called technique of “knowledge discovery from text” or “text mining” was developed to tackle this data rich but information poor dilemma. It refers to a process of transforming unstructured data into structured one and extracting a high-quality information for further analysis. Since there is a strong evidence for the influence of news articles on stock prices, text mining has become an emerging technique for stock price forecasting. While there is a number of researches conducted based on this method in developed countries’ financial market, much fewer are tested in those of developing ones. Hence, it is of value exploring the forecasting ability of news article on stock prices in the context of emerging economies, part of which includes Thailand.

The goal of this paper is to forecast the daily movement of the Stock Exchange of Thailand Index, SET50 index, using text mining on online news articles covering 8-year period. This research applies four supervised machine learning algorithms to construct the forecasting model. In addition, the performance of each algorithm will be evaluated and compared across three different approaches. Particularly, this study poses two questions. First, it questions how effective this proposed approach is in forecasting the stock index movement. Second, it identifies whether which approach or which algorithm yields the best forecasting performance.

The remainder of the paper is organized as follows. The following section summarizes a variety of existing studies on stock prediction using text mining approach. Section 3 describes the datasets. Section 4 illustrates how to conduct the research in details including explaining different types algorithms and evaluation measures. The forecasting performance of each model is then evaluated in section 5. Lastly, section 6 provides a conclusion, limitations and extensions of the study.

## **2. Literature Review**

The skepticism about the predictability of stock price can be traced as far back as 1960s when the Capital Asset Pricing (CAPM) model was introduced. At the time, as an opposition to the CAPM theory, the Efficient Market Hypothesis (EMH) proposed that the stock market is informationally efficient in a way that its prices always fully reflect all available information (Fama, 1965). Specifically, any news prevailing to the public is incorporated into the current market price. Stock price movement can be characterized as following a random walk and therefore, cannot be predicted (Malkiel, 2003). Moreover, the Efficient Market Hypothesis classifies the stock market efficiency into different degrees, which are weak, semi-strong, and strong form. The weak form indicates that all historical prices are fully integrated into stock prices, whereas the semi-strong one asserts that all publicly available information is incorporated into prices. In contrast, the strong form assumes that all information, either publicly or privately available, is included in prices.

Although the Efficient Market Hypothesis enjoyed a great popularity in the past, two contradicting theories constitute a major school of thoughts in a modern-day finance, namely fundamental and technical analysis. Despite the fact that both of them concludes that stock prices can be predictable, they are on the opposite sides of the spectrum. Fundamental analysis is an attempt to determine the intrinsic value of stock based on relevant financial information (Abad, Thore, and Laffarga, 2004). Technical analysis, on the other hand,

focuses on predicting the stock price through the use of historical data, mostly price and volume (Brown and Jennings, 1989).

In addition to two traditional practices, the rising era of big data has led text mining to become another ubiquitous tool for stock market prediction. This is reflected in a large number of researches investigating this particular topic; however, there are a variety of datasets, methods, and algorithms among researches, culminating in a variation in results as well. The first application of machine learning techniques on validating the predictive power of new articles on stock prices was conducted by Wuthrich et al. in 1998, in which he correlated the frequency of keywords in The Wallstreet Street Journal articles with major stock market indices (Wuthrich et al., 1998). In spite of low accuracy, it triggered interest of other academics as they investigated by exploiting a wide range of algorithms, for example, Lavrenko et al. (2000) employed Naïve Bayes algorithm, a probabilistic classifier based on an assumption of independence between predictors, to analyze over 30,000 of news articles. On the other hand, Mittermayer (2004) implemented a trading system based on Support Vector Machines (SVM) to predict stock trends immediately after the publication of press releases. Additionally, some researchers also extract emotional stance and integrate it into a model. This approach of identifying the polarity of opinion is known as Sentimental Analysis. The work of Das and Chen (2007) illustrates that the extraction of investors' sentiment from stock message boards increases the quality of sentiment index. Furthermore, several pieces of researches extend the sources of textual data beyond news articles to social network, online finance forums and financial reports. The study by Bollen et al. (2011) shows that the inclusion of public mood states derived from Twitter feeds could improve a performance of the predictive model of Dow Jones Industrial Average (DJIA) index. A more-detailed review on relevant studies including their methods, types of algorithms, and findings are summarized in Appendix A.

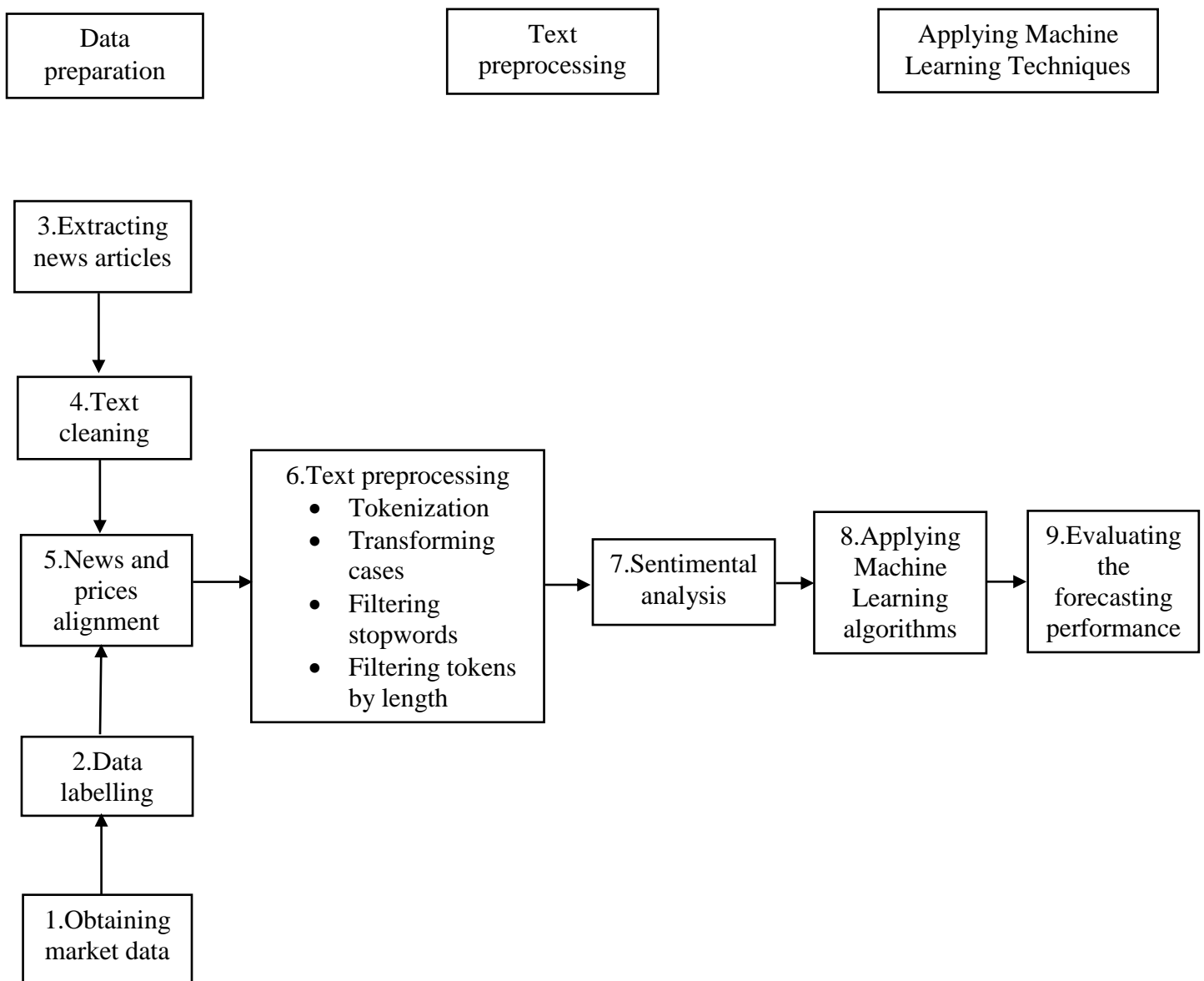
### **3. Data**

Two datasets were used in this research. The first dataset is the 10-year-period market data of daily Stock Exchange of Thailand index (SET50) ranging from November, 7<sup>th</sup> 2008 to November, 7<sup>th</sup> 2018. Despite being provided on a daily basis, data on the days when the exchange is closed are excluded as they are not available. There are 2442 observations in total. The dataset is obtained from the Bloomberg terminal and is presented in Open-High-Low-Close form, which represents the opening, highest, lowest, and closing price in each specific day. The second data, on the other hand, is online news articles published throughout

the period. They were retrieved from influential news website, namely CNBC<sup>1</sup>, with the help of RapidMiner, a data science software. Each news record extracted includes the publish date, URL link, title, author and article. The total number of news amounts to 846 articles.

[1] <https://www.cnbc.com>

#### 4. Methodology



The diagram above portrays that the methodology of this research can be divided into 3 main processes – data preparation, text preprocessing, and applying machine learning

algorithms. Each of them contains multiple subprocesses. The first process, data preparation, involves acquiring, labeling, cleaning and aligning data with a purpose of facilitating further analysis. Secondly, text preprocessing is carried out to transform textual data into a pattern readable by machine learning algorithms. Alternatively, sentimental analysis can also be performed to extract emotion stance from the text. Finally, machine learning algorithms are applied to the prepped data in order to build a predictive model upon it. The performance of the model is then evaluated using different measures.

The methodology of this research is entirely conducted through a data science platform named “RapidMiner” and its extensions including Web Mining, Aylien Text Analysis, Rosette Text Analytics and Text Processing. RapidMiner provides a wide range of tools for data preparation, text mining, machine learning, and predictive analytics through a graphical interface which enables users to accelerate procedures. Hence, operators identified in this and the following section is all referred to those in RapidMiner. All processes will be discussed in turn as illustrated in the flow diagram.

#### **4.1) Data Preparation**

This is the first step in text mining in which data is prepared for further analyses. This process starts from collecting raw data, cleaning, and transforming it into a form that can be easily processed later on. This includes a number of subprocesses which can be explained as follows.

##### **4.1.1) Obtaining market data**

In this process, the dataset of SET50 index is acquired from the Graph Price (GP) function on the Bloomberg terminal and converted into a Microsoft Excel file. The file is then imported into RapidMiner through Read Excel in which its cells can be selected via Import Configuration Wizard.

##### **4.1.2) Data Labelling**

The closing price of the index is used to calculate price change according to specified formula.

$$PRICE\_CHANGE = PRICE_t - PRICE_{t-1}$$

*where PRICE is the closing price*



This process could be executed easily in RapidMiner via the operator of Lag Series to create  $P_{t-1}$  and Generate Attribute to create a new variable,  $PRICE\_CHANGE$ , using the specified formula. The variable  $PRICE\_CHANGE$  is then assigned as a label and classified into a binary variable using Set Role and Classify by Trend operator, respectively.

$$PRICE\_CHANGE = \begin{cases} UP & \text{if } \geq 0 \\ DOWN & \text{if } < 0 \end{cases}$$

The result is that the class of the variable is generated as a new attribute called “label”, meaning that it is a variable that we need the machine learning algorithms to take as an input in building a predictive model.

#### **4.1.3) Extracting news articles**

This process involves retrieving news articles from online news websites. News articles can be easily extracted by using Crawl Web operator offered in Web Mining extension in which it will fetch all the text prevailing in a specified URL and write them in a set of documents or corpus in which each document refers to each article and its respective publish date, link, title, and author are represented as a column or attribute in RapidMiner.

#### **4.1.4) Text cleaning**

When news articles are extracted from websites, they are usually not ready-to-use as some of them may contain duplicates and irrelevant data such as advertisement and HTML links. As a result, it is necessary to preliminary clean the data. Filter Example operator is applied to remove document with such data. Moreover, unused attributes, including link, title, and author are removed through Select Attribute operator for the sake of simplicity.

#### **4.1.5) News and prices alignment**

In order to apply supervised machine learning algorithms, each document must be classified. Hence, news and its corresponding stock price change on the same day are aligned. Particularly, each news articles will be labeled as either up or down, consistent with the respective change in the index on the same day. Nevertheless, there are some news articles that were issued on the day where trading session is closed and the index quote is not available; hence, they were disregarded in this case. This step can be accomplished on the platform by Merging.

## 4.2) Text Preprocessing

Text preprocessing is a core task in text mining. It must be undertaken prior to applying machine learning techniques in order to transform prepped data into a format that is understandable by algorithms. Specifically, it converts the input corpus or set of documents into a “word vector” which numerically represents the document. Thus, text preprocessing is consistent with a concept of treating a text as a “bag of words”, implying that it only focuses on the word frequency whereas the grammar and order of the words are ignored. There are many criteria regarding the vector creation such as term frequency and term frequency-inverse document frequency (TF-IDF). While term frequency simply counts how many times a certain word appears in an individual document, TF-IDF compares the frequency in each document as opposed to the frequency of the term in all documents. TF-IDF can be computed as a multiplication of term frequency (TF) and inverse document frequency (IDF), which can be specified as follows:

*For a term  $t$  in document  $d$ ,*

$$\begin{aligned} TF - IDF (t, d, D) &= TF (t, d) \times IDF (n, dt) \\ &= \frac{\text{number of occurrences of } t \text{ in } d}{\text{number of terms in } d} \times \log\left(\frac{n}{dt}\right) \end{aligned}$$

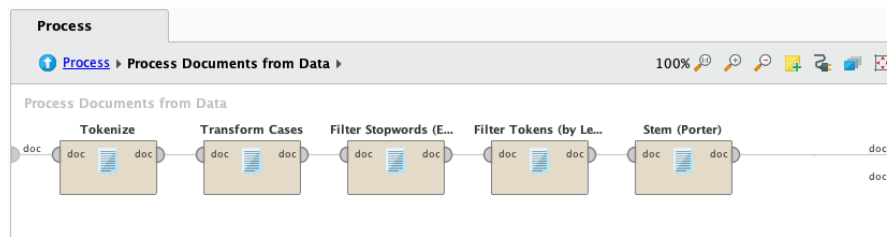
*where  $n$  is the total number of documents*

*$dt$  is the total number of documents containing the term  $t$*

Given that the more frequently the word appears across documents, the less meaningful it becomes, TF-IDF word representation seems to be an appropriate one. Therefore, this research will employ this feature selection. In addition, some word might appear too frequently or too infrequently; thus, prune method can optionally be applied where the word vector with its frequency is above or below the certain threshold will be removed. This study has filtered words with frequency less than 3 percent as they tend to be insignificant.

A wide range of operators is applicable to this process, most common ones are tokenizing, transform lowercase, filter stopwords, stemming, generating n-grams and so on. However, various approaches regarding choices of operators are taken among researches, depending on their objectives. This paper selects the use of tokenizing, transform lowercase, filter stopwords, filter token by length and stemming, which will be described separately below. This process can be carried out in RapidMiner by using Process Document from Data

operator, in which selected operators are to be placed within this operator, as shown in figure 1.



*Figure 1: Process Document from Data on RapidMiner platform*

#### **4.2.1) Tokenization**

Tokenization is a process where a textual data is separated into strings of single words or tokens which are in turn used to generate a word vector. Tokenizing primarily removes all the punctuations, numbers, and symbols. It can be performed in the program under the operator with the same name, Tokenize. In this operator, there are several options to identify the splitting points including specify characters and non-letters. Since the goal is to transform textual documents into word vectors, it is appropriate to use non-letters.

This process will result in a vector model represented in a form of matrix, called “vector representation” or, in particular, “bag-of-word representation”, in which each document corresponds to a row (example) and each word vector corresponds to a column (attribute). Each element of the matrix represents the TF-IDF value of word frequencies. The matrix is considered as a sparse matrix as most of the elements are zero.

#### **4.2.2) Transforming cases**

According to the bag of words approach, whether the word is written in uppercase or lowercase does not significantly matter. Moreover, a word starting with a capital letter is interpreted as a different term from the same word beginning with a small one; as a result, to avoid counting the same word twice, all word vectors are transformed into either lowercase or uppercase alphabets using Transform cases operator.

#### **4.2.3) Filtering Stopwords**

Stopwords are commonly used words that do not convey any meaningful information rather than serve for a grammatical function such as a, an, and, the and so on. Therefore, these words are filtered out as they do not contribute a useful context to the documents. Given that their high frequencies might obscure the understanding of the context by the

machine learning algorithms, moving such words is found to speed up and improve the performance of the process (Gurusamy and Kanna, 2014). On the platform, Filter Stopwords (English) is applied to remove these English commonly-used words from the documents.

#### **4.2.4) Filtering tokens by lengths**

Although most of stopwords are already dropped out, other irrelevant words might still remain, for example, a single alphabet or a word containing couples of alphabets often appear. These words can be eliminated using Filtering tokens by lengths operator. It filters the term based on the number of characters, where the minimum and maximum characters can be specified. In this case, the word with length less than 2 will be dropped.

#### **4.2.5) Stemming**

Stemming is the process of transforming the word into its roots by removing all the suffixes and counting only stems, for example, the word “increase”, “increased”, “increasing” are all reduced into one word, “increas” and all three words’ frequencies are combined. Stemming reduces the dimensionality of the vector representation by lowering the total number of word vectors and hence, reduce the complexity of the model (Porter, 1980). There are numerous algorithms used for stemming, the most popular ones include Porter, Lovin, Snowball. This paper has used the one by Porter (1980). This process can be executed through the operator Stem (Porter) on the software. Although the use of stemming might transform word vectors into a peculiar root, it is found that it still preserves its meaningful context and therefore, does not distort the prediction.

#### **4.2.6) Sentimental Analysis**

Relying solely on the word frequency could be misleading; thus, another technique widely used in text mining called “Sentimental Analysis” or “Opinion Mining” is applied to extract the overall emotional stance from the data (Bholat et al., 2015). The sentimental analysis can be seen as a classification of sentiment into positive, negative, or neutral by identifying and comparing positive and negative words. On RapidMiner, this task can be performed through Analyze Sentiment operator provided by the extension, Rosette Text Analytics. The extracted sentiment is then shown in a new attribute named sentiment.

### 4.3) Applying machine learning techniques

This process consists of 2 subprocesses. The first step is to apply various machine learning algorithms to forecast the stock price change. The second step is to evaluate of the performance of each model across different approaches.

#### 4.3.1) Applying machine learning algorithms

The field of machine learning can be categorized into two major tasks, unsupervised and supervised learnings. Unsupervised learning involves inferring patterns from the dataset without classification or label. Supervised learning, on the other hand, is related to learning and analyzing from labeled data to classify the new dataset. Regression and classification fall into the category of supervised learning. Since this study is one application of classification problem; hence, numerous supervised machine learning algorithms or so-called classifiers will be applied. Although there are a wide range of supervised machine learning algorithms, this work has selected four widely-used classifiers, all of which is elaborated below.

- **Decision Tree (DT)**

Decision Tree is a ruled-based algorithm which performs a classification by sorting based on feature values (Kotsiantis et al., 2007). It is referred as a hierarchical model consisting of nodes and branches. Each node represents a feature or attribute used to classify while each branch refers to its respective value (Omary and Mtenzi, 2010). The classification process is a top-down approach starting from the root node and splitting recursively into internal nodes until the leaf node is reached. To illustrate how it works, a simple decision tree in the context of news article is drawn in figure 2 using vector representation from table 1. This table shows that each document is represented by two attributes, sentiment and word vector 'boom' in which its respective value refers to the frequency. As it can be seen from the table, documents labeled as positive are always associated with upward price change. Thus, it is better to start the classification using sentiment as a root node. It can be interpreted as when the document is detected as having positive emotion stance, it will be predicted as associating with upward price change. In contrast, if the document's sentiment is classified as negative, whether the price change can be predicted as either up or down depending on the certain value of frequency of the term boom.

document no.	sentiment	boom	price change
1	positive	0.05	up
2	positive	0.07	up
3	negative	0.02	down
4	negative	0.06	up

Table 1: A simple vector representation used for decision tree

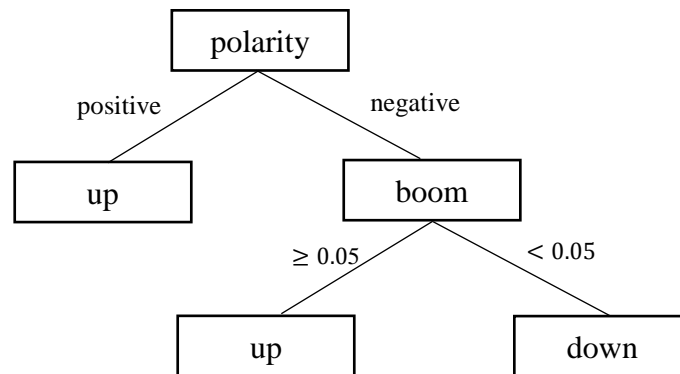


Figure 2: A simple decision tree for news articles

Decision tree is non-parametric due to the fact that it does not make any underlying assumption about the frequency distribution of the data. However, criteria in splitting such as information gain and gain ratio can also be optimized.

- **Naïve Bayes**

Naïve Bayes is a probabilistic model based on Bayes theorem with a ‘naive’ assumption that each attribute is independent in terms of determining the classification. Let  $C$  refer to the class and  $x$  refer to the feature attribute. Bayes theorem can be specified as

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

To demonstrate more clearly, consider an example described in the likelihood table (table 2),

Likelihood table		Price change		
		up	down	
Sentiment	Positive	6/9	2/5	8/14
	Negative	3/9	3/5	6/14
		9/14	5/14	

Table 2: The likelihood table for Naïve Bayes

Using the formula above,  $P(up|positive) = \frac{P(positive|up)P(up)}{P(positive)} = \frac{\frac{6}{9} \times \frac{9}{14}}{\frac{8}{14}} = 0.75$

Similarly,  $P(down|positive) = \frac{P(positive|down)P(down)}{P(positive)} = \frac{\frac{2}{5} \times \frac{5}{14}}{\frac{8}{14}} = 0.25$

Since the classifier predicts the class with higher probability, the prediction of document labeled as positive is upward price change.

By generalization to include attributes from  $x_1$  to  $x_n$

$$P(C|x_1, x_2, \dots, x_n) = \frac{P(x_1, x_2, \dots, x_n|C)P(C)}{P(x_1, x_2, \dots, x_n)}$$

Using the independence assumption that

$$P(x_1, x_2, \dots, x_n|C) = P(x_1|C)P(x_2|C) \dots P(x_n|C)$$

The formula is simplified to

$$P(C|x_1, x_2, \dots, x_n) = \frac{P(C) \prod_{i=1}^n P(x_i|C)}{P(x_1, x_2, \dots, x_n)}$$

Since the denominator is constant, it can be removed. Therefore,

$$P(C|x_1, x_2, \dots, x_n) \propto P(C) \prod_{i=1}^n P(x_i|C)$$

The algorithm classifies by computing the probability of all class variables C given sets of attributes and choosing the highest value. Hence, the Naïve Bayes classifier can be expressed mathematically as

$$C = \operatorname{argmax}_c P(C) \prod_{i=1}^n P(x_i|C)$$

- **K-Nearest Neighbor (k-NN)**

K-Nearest Neighbor is a non-parametric algorithm (Cover and Hart, 1967). It is called ‘lazy’ algorithm due to its simple and straightforward process. In classification problem, the algorithm assigns the target unlabeled example based on a majority of classes among its k nearest neighbors. Note that k is an integer and a parameter which can be optimized. In order to classify, the algorithm calculates the distance from its target to all examples. The Euclidean distance metric is commonly used for determining the distance in the case of continuous variable. The simple illustration of k-NN classifier is shown in figure 3.

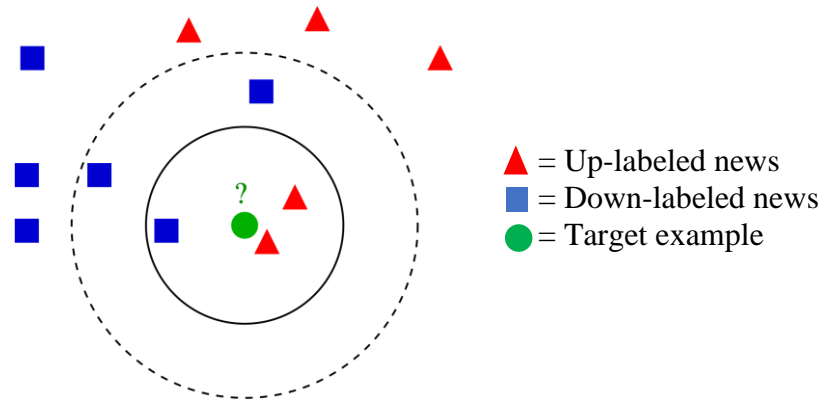


Figure 3: A simple illustration of  $k$ -NN classifier

(retrieve from: [https://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm))

To demonstrate more clearly, consider a red triangle as a news classified as up and a blue square classified as down. Target example presented as a green circle at the center can be classified as either up or down. For the case when  $k=3$ , which is represented in the solid inner circle, the  $k$ -NN classifier will classify the target as up since the number of news labeled as up outweighs those labeled as down. In contrast, when consider the outer dashed circle with  $k=5$ , the target example will be assigned as down instead due to the majority of down classes.

- **Support Vector Machine (SVM)**

Support Vector Machine is used to perform binary classification by constructing a separating hyperplane that maximizes a margin or the distance between two closest data points from each class and hence, minimizes the classification error. The solution of SVM, therefore, depends only on these data points, which are called ‘support vector’. Figure 4 illustrates how SVM fundamentally works. Note that the diagram shows SVM in performing linear classification. In fact, SVM can also solve for non-linear classification, which will be represented in multi-dimensional spaces.

positive  
news



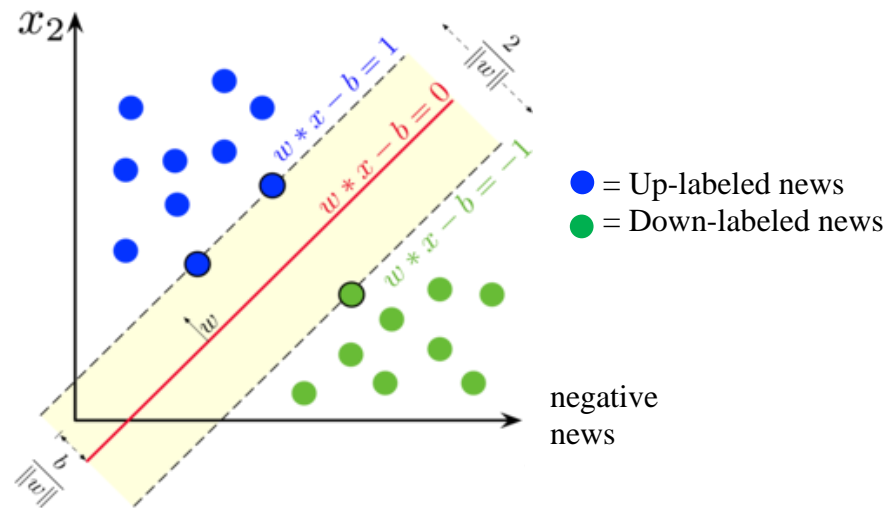


Figure 4: A simple illustration of SVM

(retrieved from: [https://en.wikipedia.org/wiki/Support\\_vector\\_machine](https://en.wikipedia.org/wiki/Support_vector_machine))

In two-dimensional space as illustrated in figure 4, by assuming the horizontal axis represents news labeled as positive while the vertical axis represents the opposite, the co-ordinate (x,y) determines classes. Therefore, green dots on the graph belongs to up classes while blue dots refer to down classes. Given these labeled data, SVM trains on this dataset by forming an optimal separating line (as shown as a red line on the graph) which divides dataset into two groups and using this line as a criterion for assigning classes to new examples.

Moreover, the classifier contains the parameter C which stands for complexity. Complexity is a degree to which the algorithm can tolerate the misclassification. It also reflects the trade-off between misclassification error and model generalization. A value of C ranges for zero to infinity in which zero implies that the model offers no penalty for misclassification. On the other hand, very large C implies that the model allows little tolerance for misclassification. C is one of the parameters that can be optimized to enhance the performance; therefore, this research will optimize the value of c.

Supervised machine learning can be divided into two subprocesses. The first subprocess is related to training an algorithm to construct a predictive model by learning the correlation between attributes and the corresponding class from labeled data. This is called 'training data'. Another subprocess is 'testing data' which involves applying the model derived from the training process to classify a new unlabeled data and evaluating the model.

As an analogy to training and testing process, there are two types of dataset. Training dataset is a sample of data in which the algorithm is learned and trained from whereas testing dataset is set of examples where the prediction of the algorithm is compared to in order to

assess the performance. Thus, how training and testing dataset is split (can be referred as split ratio) matters for the evaluation. One of useful technique is called “Cross Validation” or, specifically, “K-fold Cross validation”. It is related to randomly partitioning the dataset into k equal-size groups. One of them is used as testing set while the remaining K-1 sets are used as training ones. The model then iteratively constructs on the training data and evaluates on the other set K times in total. The estimated performance will be an average of all values. Due to its advantage in reducing biasness, this research employed this method for the evaluation.

In RapidMiner, Cross Validation can be applied using the operator under the same name, Cross Validation. This operator is divided into two subprocesses, training and testing. All supervised algorithms must be trained in the training process; as a result, the algorithm operator is placed in this area. In contrast, in testing process, the operator Apply Model and Performance are applied. While the former allows the algorithm to make a prediction, the latter evaluates the algorithm by comparing predicted classes to actual classes and therefore, produces the resulted performance values in various measures. The process of Cross Validation on RapidMiner is shown in figure 5.

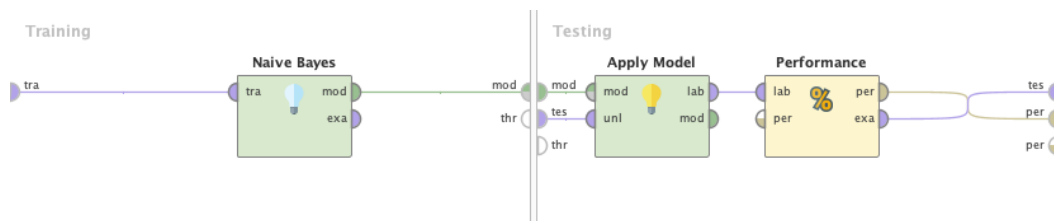


Figure 5: Cross validation on RapidMiner

#### 4.3.2) Evaluating the forecasting performance

The performance of machine learning algorithms can be evaluated using different measures, namely confusion matrix and ROC curve.

- **Confusion matrix**

In a binary classification problem, the prediction by the classifier can be represented in a two-by-two matrix called “confusion matrix” or “contingency table”. The confusion matrix illustrates the number of documents correctly and incorrectly predicted by the algorithm, as shown in table 3.

Predicted class	True class	
	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)

Negative	False Negative (FN)	True Negative (TN)
----------	---------------------	--------------------

*Table 3: Confusion matrix*

where True Positive = the number of positive class correctly classified

False Positive = the number of negative class incorrectly classified as positive

False Negative = the number of positive class incorrectly classified as negative

True Negative = the number of negative class correctly classified

**Accuracy:**

Accuracy is the ratio of correctly predicted documents to the total number of documents, which can be defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

The higher value of Accuracy, the better performance of the algorithm. However, Accuracy is an appropriate measure only when the dataset is balanced. In the case of unbalanced dataset where one class is significantly lower than another, accuracy could be biased in a way that it is skewed towards the majority class (Falinouss, 2007). Thus, under this circumstance, other measures, specifically, precision and recall are more appropriate due to more sensitivity to their own class (Landgrebe et al., 2006).

**Precision:**

Precision is considered as a measure of exactness where a high precision implies most of the documents classified as positive are predicted correctly. It is the ratio of the number of documents correctly classified as positive divided by the total number of documents being classified as positive, which is computed as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

**Recall:**

Recall, on the other hand, can be seen as a measure of completeness in which a high recall indicates that a large number of positive documents is labeled correctly. It can be estimated as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The higher the precision is, the lower the recall becomes, reflecting the Precision-Recall trade-off.

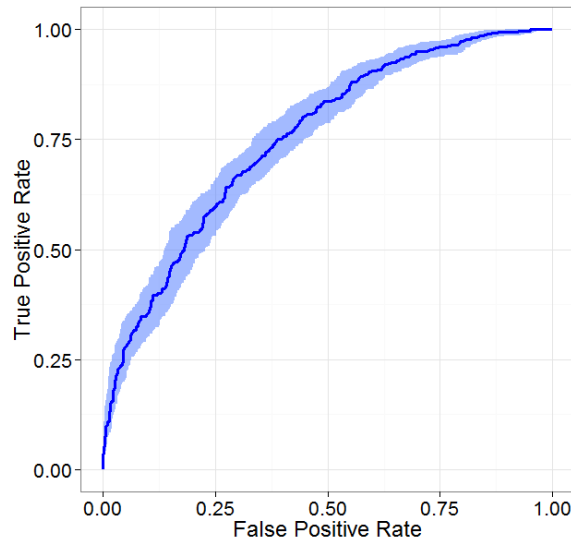
- **Receiver Operating Characteristic (ROC) curve**

ROC curve is another useful technique to visualize the performance of classifiers. It captures all elements from the confusion matrix (Swets, 1988). As demonstrated in figure 6, ROC curve represents the trade-off between True Positive Rate (also referred as hit rate) and False Negative Rate (also referred as false alarm rate), which are plotted on the vertical and horizontal axis, respectively. True Positive Rate measures the ratio of positive classes correctly classified to the total number of positive classes while False Negative Rate indicates the ratio of negative classes incorrectly classified to the total number of negative classes. Both rates are calculated as

$$\text{True Positive Rate (TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{False Postive Rate (FPR)} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

The performance of the classifier is better when True Positive Rate is high while False Negative Rate is low. Therefore, generally speaking, the further the curve to the northwest direction, the better the performance of the classifier. Theoretically, the performance can be evaluated in terms of the area under the curve (AUC) in which the larger the area is, the better the model performs. Furthermore, ROC curve is relatively more beneficial than confusion matrix in terms of visualization. The confusion matrix only shows the evaluation of one classifier at a time. ROC curve, by contrast, allows the comparison of performance among algorithms as it can plot multiple ROC curves onto the same graph. Hence, this research will exploit this feature of ROC curve by using it to compare the performances of different types of algorithms.



*Figure 6: ROC curve*

(retrieved from: <https://www.epeter-stats.de/roc-curve-analysis-with-fbroc>)

## 5. Results

This section provides results to steps in methodology and evaluation of the algorithms among three different approaches.

### 5.1 Results from news and price alignment

After the completion of the data preparation, news and price change on the same date are aligned together, resulting in corpus consisting of 846 news documents and 2 attributes including label and publish date. The result from RapidMiner is demonstrated in figure 7.

publishDate	article	label
08/03/2018	The economic growth outlook is positive for Thailand due to strong commitmen...	up
08/03/2018	There is sufficient public and social pressure this time to ensure that elections ...	up
08/03/2018	An increase in the political space could mobilize the silent majority to vote in th...	up
28/02/2018	Mark Matthews of Bank Julius Baer says the major thing that is being watched i...	up
19/02/2018	Pent-up demand in Thailand saw a rebound across all business segments, say...	up
16/02/2018	Uber is preparing to sell its Southeast Asia business to Singapore's Grab in exc...	down

*Figure 7: Results from news and price alignment on RapidMiner*

The research examines further into the statistics of this dataset, which is shown in figure 8. As it can be seen, among the total number of 846 documents, 484 of them are labeled as up while the other 362 are classified as down. This suggests that the dataset exhibits some degree of unbalance in which the down classes outweigh the up classes. Thus,

it is appropriate to use other different measures apart from accuracy to validate the performance of the classifier.

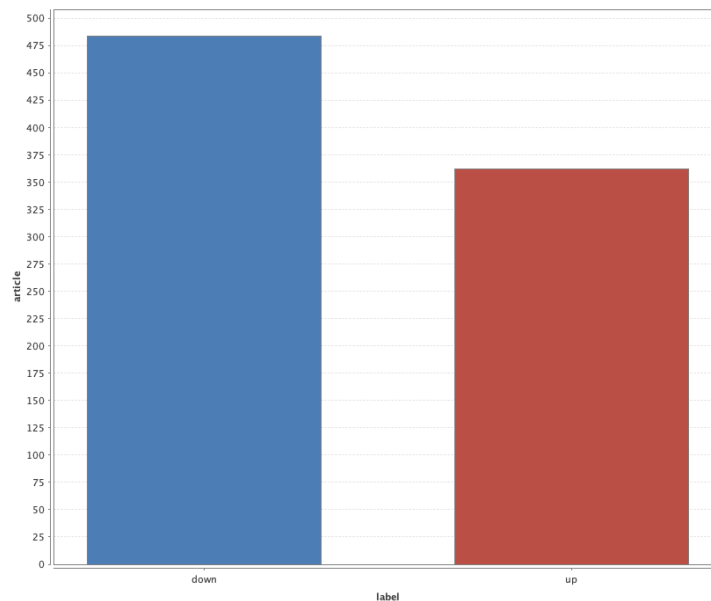


Figure 8: a bar chart for up and down classes from RapidMiner

## 5.2 Word representation

After the completion of the entire text preprocessing, the result is generated in a TF-IDF word representation with 846 documents, 7319 regular attributes, and 1 special attributes or label. However, the matrix is still considered large in dimensionality; therefore, the pruning method with minimum frequency of 3 percent is then applied and thereby, managed to reduce the size of the matrix to only 584 significant regular attributes. In particular, this research found that by using the pruning method, the accuracy of the prediction improves, and thereby, decided to apply this method. Moreover, when applying sentimental analysis, the extracted sentiment is shown in sentiment attribute in which 'pos', 'neu', and 'neg' stands for positive, neutral, and negative sentiment, respectively. The TF-IDF word representation is demonstrated in figure 9 below.

Row No.	label	Sentiment	access	accord	account	acquisit	activ	ad
1	down	pos	0	0	0	0	0	0
2	down	neu	0	0	0	0	0.089	0
3	up	neu	0	0.128	0	0	0.178	0.061
4	up	pos	0	0	0	0	0	0
5	up	pos	0	0	0	0	0	0.214

Figure 9: a TF-IDF word presentation from RapidMiner

### 5.3 Prediction of algorithms

The TF-IDF word representation is then used as an input to the machine learning algorithms in order to build a predictive model. The prediction from the model is displayed in additional attributes in the matrix, which are prediction, confidence level of predicting up, and confidence level of predicting down. The word representation with prediction is demonstrated in figure 10.

Row No.	label	prediction(label)	confidence(down)	confidence(up)	Sentiment	access	accord	account
1	up	down	1.000	0.000	neu	0	0.128	0
2	down	down	1.000	0.000	neu	0	0	0
3	down	up	0	1	neu	0	0.135	0.100
4	down	down	1	0	neg	0	0	0
5	down	up	0	1	neg	0	0	0

Figure 10: a prediction of the algorithm from RapidMiner

### 5.4 Evaluation of algorithms

In this research, the forecasting performance of four algorithms are evaluated among three approaches, which are TF-IDF word representation, sentimental analysis, and the combination of the first two approaches.

In addition to the prediction presented in TF-IDF word representation, Cross Validation provides a mean to evaluate forecasting performance of the algorithms based on various criteria. This research focuses the use of confusion matrix and ROC curve.

#### 5.4.1) Evaluation using confusion matrix

Consider a confusion matrix resulted from k-NN classifier in TF-IDF approach, which is exhibited in table 4. This is the result obtained after the optimization of parameter k, in which the optimal value of k equal to 1. From the matrix, it can be concluded that among 484 down classes, 304 of them are correctly predicted as down whereas the other 180 documents are incorrectly predicted as up. Likewise, given the total number of 362 news labeled as up, 185 are correctly predicted while the rest are incorrectly assigned. Moreover, the accuracy of 57.45 can be interpreted as the algorithm can predict both ups and downs correctly 57.82 percent of the time. On the other hand, when consider the recalls for both classes, the relatively higher recall for downs indicates that the classifier predicts downs 1.23 times better than ups. Similarly, class precision implies the better performance of the algorithms in classifying down category.

$$\text{accuracy} = 57.82\%$$

	true down	true up	class precision
predicted down	304	177	63.20%
predicted up	180	185	50.68%
class recall	62.81%	51.10%	

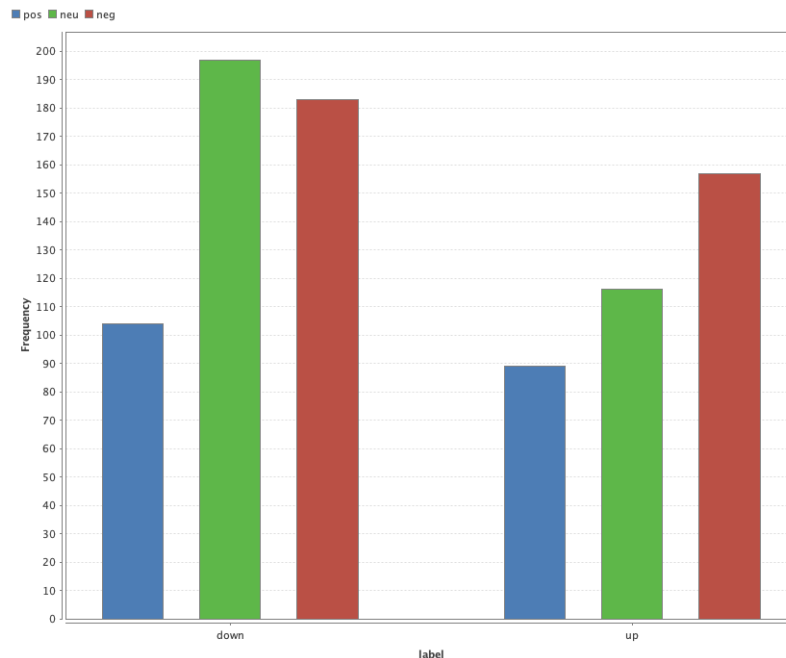
Table 4: Confusion matrix from  $k$ -NN based on TF-IDF (with  $k=1$ )

To evaluate among different approaches, it is useful to examine the confusion matrix derived from the second approach which employed only sentimental analysis. Note that under this approach, the optimal value of  $k$  is 100. From the matrix provided in table 5, recall values indicate that the classifier performs relatively better in predicting down groups while performs relatively poorer in up classes. Consistently, it is also reflected in lower accuracy compared to the one from the first approach. The fact that sentimental analysis underperforms TF-IDF can be attributed to its lower performance in predicting up categories. This is evidenced in a histogram presented in figure 11. As can be seen, among down-labeled news, a majority of them is associated with neutral and negative sentiment and these two groups are much larger than those labeled as positive. On the other hand, in up-labeled group, a majority is associated with negative while a minority is positive which does not seem sensible. Thus, it might explain the relatively poorer performance by the classifier.

accuracy = 55.43%

	true down	true up	class precision
predicted down	380	273	58.19%
predicted up	104	89	46.11%
class recall	78.51%	24.59%	

Table 5: Confusion matrix from  $k$ -NN based on sentimental analysis (with  $k=100$ )





*Figure 11: a histogram for two classes: up and down*

The confusion matrix from the third approach which combines TF-IDF and sentimental analysis together is described in table 6. The accuracy is 59.11 percent which is higher than both obtained from the first and the second approach. In comparison to the first approach, recalls and precisions are relatively higher as well. Thus, the outperformance of this approach can be attributed to the greater ability of sentimental analysis in predicting the down classes.

accuracy = 59.11%

	true down	true up	class precision
predicted down	310	172	64.32%
predicted up	174	190	52.20%
class recall	64.05%	52.49%	

*Table 6: Confusion matrix from k-NN based on TF-IDF and sentimental analysis (with k=1)*

The performance of four algorithms among three approaches are summarized in table 7. The table is constructed using accuracy measures obtained from confusion matrices. For k-NN and SVM classifier, their respective parameter which are k and C are optimized whose values may differ among approaches.

Machine Learning Algorithms	Approaches		
	TF-IDF	Sentimental Analysis	TF-IDF and Sentimental analysis
Decision Tree	57.32%	57.21%	57.32%
Naïve Bayes	57.45%	57.21%	57.33%
k-NN (with optimization)	57.82%	55.43%	59.11%
SVM (with optimization)	58.39%	57.45%	58.86%

*Table 7: The comparison of accuracy measure among different approaches*

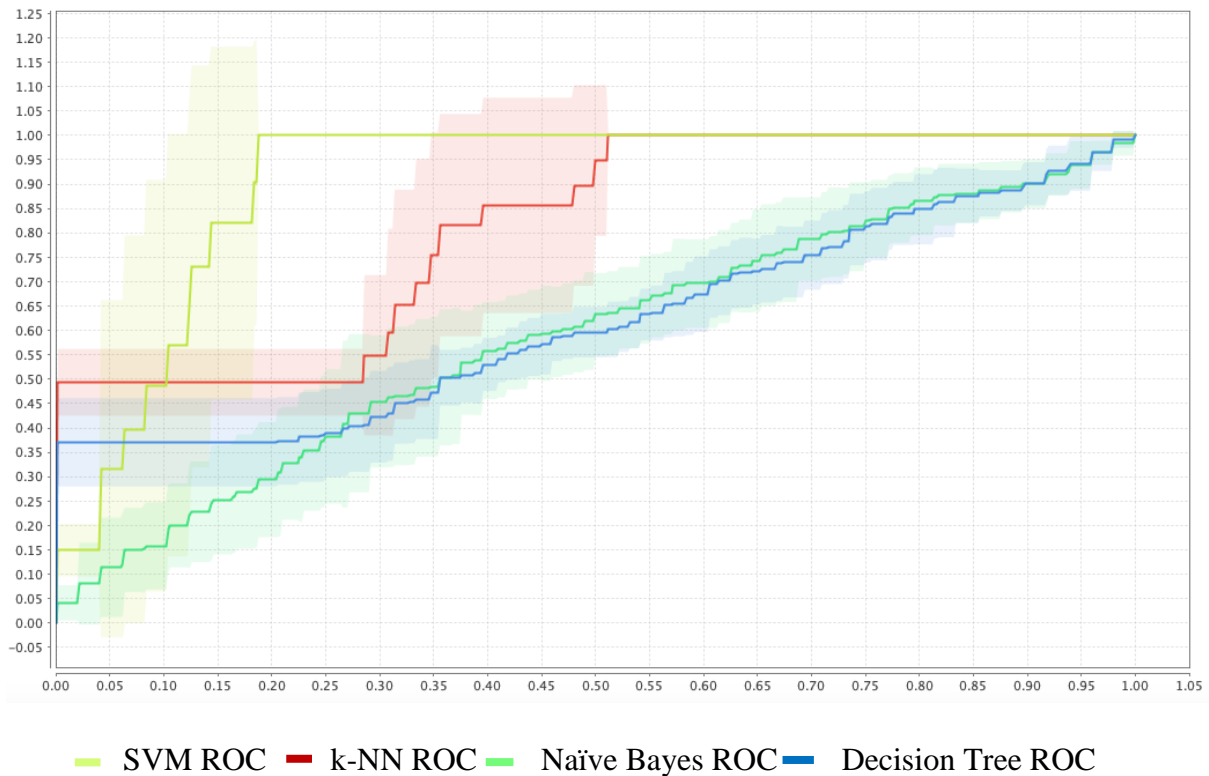
*(All values are obtained from confusion matrices which are provided in the Appendix B)*

The table reflects two important findings which provide answers to the second research question. First, the comparison across three approaches reveals that sentimental analysis yields the lowest performance regardless of the algorithm; however, it is still controversial which one results in the best performance, for example, TF-IDF representation performs best in Naïve Bayes while the mixture of the two works best in SVM and k-NN. Second, in order to evaluate across different types of algorithms, it is ambiguous to determine which a single algorithm has the highest forecasting ability as the performance of each algorithm varies across approaches. The result suggests that SVM produces the most precise prediction in the both TF-IDF frequency and sentimental analysis approach whereas k-NN has the best forecasting ability in the last approach.

Regarding to the first research question, in order to evaluate how well the proposed method predicts the stock price movement, the research compares the resulted algorithms' performance to those in existing literatures. According to Wuthrich et al. (1988), the accuracy measures of k-NN are approximately 42, 47, and 40 percent for FTSE100, Nekkei25, and DJIA index, respectively. Therefore, this proposed approach based on this algorithm tends to give reasonably good performance. The study by Lavrenko et al. (2000) based on Naïve Bayes classifier resulted in 40 percent of recall; therefore, recall measures (provided in Appendix A) are relatively better in comparison.

#### **5.4.2) Evaluation using ROC curve**

As an alternative to comparing back and forth across confusion matrices, ROC curve serves as a useful tool in visualizing the validation of the classifier. ROC curve for the first approach, TF-IDF vector representation, is illustrated in figure 12.



The evaluation demonstrated in the ROC curve is consistent with the results derived from accuracy measures. As it can be seen, under TF-IDF approach, SVM classifier yields the highest performance as it is the furthest to the (0,1) and it has the largest area under the curve. The next best performing algorithm is k-NN, followed by Naïve Bayes and Decision Tree, respectively.

## 6. Conclusion

This research has applied text mining technique to investigate whether Stock Exchange of Thailand index (SET50) can be forecasted by online news articles. Using two datasets of daily SET50 index and online news articles, it applies various feature selections and machine learning algorithms to address two questions including whether the proposed methodology is able to predict the stock price movement and if it does, which approach and which algorithm produces the highest forecasting ability.

The results from the study suggests that although it is not concludable which approach gives the highest performance as they tend to differ across algorithms; nevertheless, in every classifier, sentimental analysis is found to yield the least predicting ability. On the other hand, when consider among classifiers, SVM performs best in the both TF-IDF frequency and sentimental analysis approach while k-NN yields the highest forecasting power in the last

approach. Furthermore, when assessing the results from this study using the relevant literatures, the proposed approaches in the research tend to give a relatively good result.

Finally, despite the interesting findings, this research contains few caveats, particularly, the data limitation in terms of diverse sources and frequency. The results relied on news samples obtained from only one provider and news are not published on everyday basis. Hence, one of further extensions of this research is extending news articles to capture longer time frame, greater number of sources and higher frequency, which can possibly lead to better forecasting performance. Consequently, the size of news samples may become larger; hence, more feature selections are required to drop insignificant word vectors. Moreover, despite optimizing some parameters, there are still more parameters and functions in the algorithms; hence, fine-tuning each algorithm would be another way to further the future work.

## **Bibliography**

Abad, C., Thore, S. A., & Laffarga, J. (2004). Fundamental analysis of stocks by two-stage DEA. *Managerial and Decision Economics*, 25(5), 231-241.

Bholat, D. M., Hansen, S., Santos, P., & Schonhardt-Bailey, C. (2015). Text mining for central banks.

Brown, D. P., & Jennings, R. H. (1989). On technical analysis. *The Review of Financial Studies*, 2(4), 527-551.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of computational science*, 2(1), 1-8.

Cover TM, Hart PE. Nearest neighbor pattern classification. *IEEE Trans Inf Theory*. 1967;13(1):21–27. doi: 10.1109/TIT.1967.1053964.

Das, S. R., & Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management science*, 53(9), 1375-1388.

Falinoouss, P. (2007). Stock trend prediction using news articles: a text mining approach.

Feldman, R., & Dagan, I. (1995, August). Knowledge Discovery in Textual Databases (KDT). In *KDD* (Vol. 95, pp. 112-117).

Gurusamy, Vairaprakash & Kannan, Subbu. (2014). *Preprocessing Techniques for Text Mining*.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160, 3-24.

Landgrebe, T. C., Paclik, P., & Duin, R. P. (2006, August). Precision-recall operating characteristic (P-ROC) curves in imprecise environments. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (Vol. 4, pp. 123-127). IEEE.

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000, August). Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining* (Vol. 2000, pp. 37-44).

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of economic perspectives*, 17(1), 59-82.

Mittermayer, M. A. (2004, January). Forecasting intraday stock price trends with text mining techniques. In *system sciences, 2004. proceedings of the 37th annual hawaii international conference on* (pp. 10-pp). IEEE.

Omary, Z., & Mtenzi, F. (2010). Machine learning approach to identifying the dataset threshold for the performance estimators in supervised learning. *International Journal for Infonomics (IJI)*, 3(3).

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137.

Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science* 240, 1285–1293.

Wuthrich, B., Cho, V., Leung, S., Permuntilleke, D., Sankaran, K., & Zhang, J. (1998, October). Daily stock market forecast from textual web data. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218) (Vol. 3, pp. 2720-2725)*. IEEE

## Appendix A

Author	Algorithm type	Feature selection	Feature Representation	Findings
Wuthrich et al. (1988)	k-NN, Neural network	Bag-of-words	Binary	k-NN: accuracy of 42%, 47%, 40% for FTSE 100,

				Nikkei 25, and DJIA, respectively
Lavrenko et al. (2000)	Naive Bayes	Language modeling	Discrete	Achieved 40% for recall and 15% for alarm rate
Peramunetilleke and Wong (2002)	Decision Tree	Bag-of-words	Boolean, TF-IDF	Better than random chance
Mittermayer (2004)	SVM	Bag-of-words	TF-IDF	Average profit of 11 percent
Das and Chen (2007)	Combination of algorithms	Bag-of-words, Triplets	Discrete	Naïve Bayes: accuracy of 60.69% and 31.54% for in-sample and out-sample
Bollen et al. (2011)	Neutral network	Sentimental analysis	-	Accuracy of 87.6% for DJIA index

Table 1: A summary of literature review

## Appendix B

### 1. Decision tree

#### 1.1 Decision tree based on TF-IDF

accuracy = 57.32%

	true down	true up	class precision
predicted down	431	308	58.32%
predicted up	53	54	50.47%
class recall	89.05%	14.92%	

#### 1.2 Decision tree based on sentimental analysis

accuracy = 57.21%

	true down	true up	class precision
predicted down	484	362	57.21%
predicted up	0	0	0.00%
class recall	100.00%	0.00%	

#### 1.3 Decision tree based on TF-IDF and sentimental analysis

accuracy = 57.32%

	true down	true up	class precision
predicted down	431	308	58.32%

predicted up	53	54	50.47%
class recall	89.05%	14.92%	

## 2. Naïve Bayes

### 2.1 Naïve Bayes based on TF-IDF

accuracy = 57.45%

	true down	true up	class precision
predicted down	299	175	63.08%
predicted up	185	187	50.27%
class recall	61.78%	51.66%	

### 2.2 Naïve Bayes based on sentimental analysis

accuracy = 57.21%

	true down	true up	class precision
predicted down	484	362	57.21%
predicted up	0	0	0.00%
class recall	100.00%	0.00%	

### 2.3 Naïve Bayes based on TF-IDF and sentimental analysis

accuracy = 57.33%

	true down	true up	class precision
predicted down	298	175	63.00%
predicted up	186	187	50.13%
class recall	61.57%	51.66%	

## 3. K-Nearest Neighbor

### 3.1 K-Nearest Neighbor based on TF-IDF (K=1)

accuracy = 57.82%

	true down	true up	class precision
--	-----------	---------	-----------------



predicted down	304	177	63.20%
predicted up	180	185	50.68%
class recall	62.81%	51.10%	

### 3.2 K-Nearest Neighbor based on sentimental analysis (K=100)

accuracy = 55.43%

	true down	true up	class precision
predicted down	380	273	58.19%
predicted up	104	89	46.11%
class recall	78.51%	24.59%	

### 3.3 K-Nearest Neighbor based on TF-IDF and sentimental analysis (K=1)

accuracy = 59.11%

	true down	true up	class precision
predicted down	310	172	64.32%
predicted up	174	190	52.20%
class recall	64.05%	52.49%	

## 4. Support Vector Machine

### 4.1 Support Vector Machine based on TF-IDF

accuracy = 58.86%

	true down	true up	class precision
predicted down	396	260	60.37%
predicted up	88	102	53.68%
class recall	81.82%	28.18%	

### 4.2 Support Vector Machine based on sentimental analysis

accuracy = 57.45%

	true down	true up	class precision
predicted down	465	341	57.69%

predicted up	19	21	52.50%
class recall	96.07%	5.80%	

#### 4.3 Support Vector Machine based on TF-IDF and sentimental analysis

accuracy = 58.39%

	true down	true up	class precision
predicted down	363	231	61.11%
predicted up	121	131	51.98%
class recall	75.00%	36.19%	