Abstract

With the rapid advancement of the data analytics field, there are a number of readily available software frameworks and tools that assist users to build applications for making prediction via analyzing data. In this project, the team aims to explore several data analytics techniques, including Natural Language Processing and Machine Learning, to build a standalone application with modularized components. With the outputs produced by this application, the ultimate goal is to achieve more accurate stock trend prediction. This report will detail the progress that the team has made; especially on the preliminary analyses on the obtained dataset, and the adopted machine learning model. The future work will be concentrated on ways to enhance the processing of the obtained dataset and optimizing the machine learning models for more precise results.

Acknowledgement

I would like to express my gratitude towards Dr. S.M. Yiu for offering suggestions whenever the team encounters issues during the development. I would also like to thank my English instructor, Mr. Cezar Cazan, for giving clear guidelines on writing academic papers.
Table of Contents

Abstract 2
Acknowledgement 2
List of Figures 4
List of Tables 4

1. Introduction: 5
   1.1. Objectives 5
   1.2 Scope: 5
   1.3 Technical Background:
      1.3.1 Natural Language Processing: 6
      1.3.2 Machine Learning: 7
   1.4. Outline 7

2. Methodology 8
   2.1 Software setup 8
   2.2 Application architecture
      2.2.1 Obtaining Data 9
      2.2.2 Natural Language Processing 9

3 Discussion of the Current Progress and Results 12
   3.1 Pipeline creation 13
   3.2 Natural Language Processing 14

4 Limitation and Difficulties encountered 17
   4.1 Natural Language Processing 17
   4.2 Machine Learning 18

5 Future Work 18
   5.1 Natural Language Processing 18
   5.2 Machine Learning 18

6. Conclusion 19
List of Figures

Figure 1- System architectural design 8
Figure 2. Formula of TF-IDF algorithm 10
Figure 3- CBOW and Skip-gram 10
Figure 4 - Project Milestones 13
Figure 5 - System Architecture, a closer look 14
Figure 6 . Sample data output 15
Figure 7 - MSE of Test Data in SVM model 16
Figure 8 - Distribution of the test data 17

List of Tables

Table 1 - NLTK Classifier model accuracy 15
Table 2 - Classification Model Accuracy 16
Table 3 - Regression Model Accuracy 16
1. Introduction:

The project aims to build an application that incorporates technologies for better performance on stock movement prediction by analyzing the market news. The objective of this project, the relevant technical background, and the outline of the report are detailed in this section.

1.1. Objectives

The goal of this project is to build a standalone application that predicts the future stock price from financial market news, with higher accuracy and speed. The objectives of this project are stated in the follows.

1. To accurately predict the stock price moving trend, in the form of “up” and “down”, based on the financial market news data collected from the online sources.
2. To accurately predict the stock price trend in terms of percentages, based on the stock price of the specified companies collected from the online sources.

1.2 Scope:

This Fintech Application will not have a user interface, and users will have to communicate with it using a Command Line Prompt. Users will not be required to plug data into the application since it will be continuously parsing relevant data from the internet. The data to be parsed can be specified by the user, through a config file.

This application will only be able to analyze the financial news from the US energy market, and the reason will be further discussed in the Methodology section. The majority of the work will focus on the ways to conduct analysis with the data, and to make prediction accordingly.

1.3 Technical Background:

The Relevant background on how these technologies are assembled together to achieve the mentioned objectives is discussed in this section. To facilitate the understanding process for the
readers, the following paragraphs are divided into 2 subsections, to explain the background of Natural Language Processing and Machine Learning.

1.3.1 Natural Language Processing:
Natural Language Processing is the field of studying and analyzing human language, in both textual and verbal forms. Its applications include natural language understanding, speech recognition, and natural language generation.

By adopting the Natural Language Processing technique, textual analysis can be performed on financial news for extracting relevant information, and based on the extracted information, stock market movement can be further predicted by implementing the machine learning technique.

Currently there exist two approaches to predicting the financial market movement - fundamental and technical. The technical approach conducts price prediction via consuming the historical market data. However, the dispute against such approach contends that it is impossible to conduct price prediction based on the historical market trend data, due to its context-specific nature\(^1\). This argument is also supported by the Random Walk Theorem\(^2\). On the contrary, the fundamental approach consumes other types of information, such as data related to the financial environment, geopolitics and etc. In the fundamental approach of predicting the financial market movement, aside from referencing numeric data obtained from the financial reports as benchmarks, analysts also look into textual data, which is available through the published financial news.

To avoid potential problems that the technical approach might cause, the fundamental approach is adopted in this project. Since fundamental approach deals with the textual data, Natural Language Processing technique is chosen for data processing. The analyzed data is then passed to the machine learning module for further prediction.
1.3.2 Machine Learning:

Machine Learning is the field of study that enables computers to learn, with or without a given set of data, and to predict the future trend.

With the analyzed textual data available, the next step is to perform learnings on the datasets. There are a few methods of constructing models for financial prediction, based on the data availability. One approach is to use the time series data, closing stock price for example, with linear regression, to find the underlying pattern[^3]. As the data analytics field advances, more variables, including non-numeric data, were taken into the calculation and analysis. Several algorithms have also been developed for textual analysis. For instance, Thomas and Sycara developed an algorithm that predicts the future “up” and “down” by calculating the numbers of the occurrence of words with the textual posts downloaded from financial related websites[^4]. Choudhry, Rohit, and Kumkum Garg implemented Support Vector Machine along with Genetic Algorithm to achieve Stock Market Forecasting[^5]. Si et al. have adopted the autoregression technique, using time series sentiment data from Twitter as input, to perform stock prediction[^6].

The above is not a exhaustive list of what has been researched so far, and there are a lot more ongoing work in this field.

1.4. Outline

The remainder of this report proceeds as follows. First, the methodologies of this project will be illustrated, and it will be followed by the discussion on the current progress and results. Subsequently, the situations on the possible difficulties will be addressed, and finally, the report will close with elaborating how the future work is going to be conducted.
2. Methodology

2.1 Software setup

The software setup comprises the installation of the Natural Language Toolkit, Numpy, Pandas, SciKit-Learn, and the Tensorflow packages. These packages will facilitate the process of building up the application with their convenient APIs.

2.2 Application architecture

The architectural design of this application consists of several layers, which account for different functionalities. These layers include parsing news data from online sources, processing the data, and feeding them into learning algorithm for constructing models.

Figure 1 illustrates the top-level flow of the application architecture design. Details are discussed in the following paragraphs, arranged by the technique used.
2.2.1 Obtaining Data

The textual data on which the Natural Language processing technique is applied will be parsed from the credible online sources Reuters News by deploying a web crawler. The Natural Language Toolkit will be utilized to analyze the obtained data. To see whether the outputs of the application yield accurate prediction, the stock price with which the output data can be compared against, will be collected from Yahoo Finance.

2.2.2 Natural Language Processing

Natural language Toolkit offers the tokenizing functions with which the prunings of the unnecessary words can be conducted. The tokenizing functions decompose texts into a list of separate words. For example, “Today is a good day” will be broken down into “Today”, “is”, “a”, “good”, and “day”. With a tokenized list of words, the stop-words removal can be conducted to eliminate the words without concrete meaning, such as the conjunction and the transition words. Therefore, “is”, and “a” will be removed from the list, which leave us with the list of words “Today”, “good”, “day”.

In order to transform the textual data into numerical data for machine learning model, word vectorization will be implemented. A robust vectorization algorithm can represent the relationship between the words. For example, it will make sense if the vectorized result which can meet the formula “King” - “Man” + “Woman” = “Queen”.

Two algorithms are adopted in this project to vectorized the word. First is the TF-IDF score (Figure 2) \[^7\]. TF stands for term frequency and IDF stands for inverse document frequency. If a word appears in every document, it might be a meaningless word with a low IDF score such as “this”, “a”. On the other hand, if a word only appears in a few document, it is highly possible that the word is the keyword. As a result, it will have a high TF-IDF score.
Secondly, algorithms mentioned in a paper Word2Vec published by Google is implemented. The algorithms include CBO(W(Continuous Bag of Words) and Skip-gram. CBO predicts the missing word with several neighbouring words. Reversely, Skip-gram predicts the surrounding words by only one word. Both algorithms transformed from input layer to output layer by multiplying a weight matrix. The result will be optimized through tuning the weight matrix during back propagation.

Finally, the Sentiment Analysis will deal with the emotional-charge or sentiment-load of the features. For each word, a compound sentiment score will be given based on the sentimental effect of each word. Since all of the features have different levels of impact, the calculation of
the average weight and the contribution of each feature will be executed in order to determine how they affect the stock price movement. The sentiment score will be incorporated into the vectorized result of each word.

To ensure that the application will produce precise and consistent results, imposing constraints on the inputs of the data is necessary. Extracting the correct and relevant information from the internet is critical to the success of the project, and therefore it is necessary to narrow down the market news scope for model training. After researching on the attributes that different types of financial market news have, the team will only focus on the market news that comes from the US energy industry, considering the less ambiguity it contains.

### 2.2.3 Machine Learning

Based on the objectives of the project, two prediction models, direction based model and the percentage based model, will be built so that the user can see the prediction in different forms. The analyzed data from the Natural Language Processing module will first be loaded in for data preparation before the training process. Feature selection will be done here in order to extract the features that are of interest[9]. Data normalization is performed here to ensure the value of the features all reside on the same scale. The two models will then prepared data, and conduct their respective learnings accordingly. The data will be first split into two sets, one for training and the the for testing, with the train test split ratio specified by the user, or defaulted to 3:1. The train data will then be passed into the learning process and be evaluated with the test data.

After the literature review, the team has selected two machine learning algorithms for the construction of the two models, Support Vector Machine (SVM) and Multi Layer Perceptron (MLP), since they yield promising results as shown in the previous research[3][5]. The two algorithms will be built for both direction based model and percentage based model.

The main idea of SVM is as follows. To separate two different classes, a decision boundary is needed. Using SVM classifier, the line will not only separates the two classes but will also stay
as far away from the closest training instances. In other words, it will fit the widest possible
street between the classes. However, the movement of stock price is not necessarily linearly
separable, which in turns requires the help of kernel tricks. Kernel tricks are about transforming
the linear model into higher dimension that has a clear dividing margin between classes of data.
The mapping function is defined through the kernel function. Such functions are able to compute
the dot product based on the original input vectors but without knowing about the actual
transformation.

The Multilayer Perceptron model adopts the neural network architecture in which inputs are
passed through multiple layers of perceptrons. It addresses the nonlinear separation problems
that a single perceptron is not capable of solving, XOR problem for example. The learning
process, whereby the connection weight between perceptrons gets updated, is conducted with
backpropagation with gradient descent. With every training instance passed into the training
model, the connection weights get updated accordingly in order for a better fit to the training
data set.

The construction of both models is split into 2 phases, the first phase entails the implementation
of the original form of the algorithm, and the time series analysis technique will be incorporated
in phase 2. Adaptation of the algorithms will also be introduced for result optimization in phase
2.
3 Discussion of the Current Progress and Results

This section will elaborate the current progress the team has made. An overview of the project milestones is shown in Figure 4. The red circle indicates the phase that the team is currently on.

![Figure 4 - Project Milestones](image)

3.1 Pipeline creation

As discussed in the methodology section, the system architecture is realized in the components shown below in Figure 5. The application consists of two modules, NLP and ML module. The Data Set layer mentioned in the previous system architecture diagram is incorporated in the NLP module. Pipelines between the components are also established in order to facilitate the dataflow. Web crawlers are realized as reuters.py, as the news crawler, and yahoo_finance.py, as the stock indices crawler. News data is then passed into vectorize_news.py for vectorization, and along with the stockReturns data, featureMatrices for both training and testing are formed. The matrix will then be fetched by main.py, the current entry point of the application, which triggers the model training and prediction process. Depends on the inputs from the users, the model objects, MLPClassifier, MLPRegressor, SVMClassifier, and SVMRegressor, will be dynamically loaded and run.
3.2 Natural Language Processing

Preliminary experiments on the Natural Language Processing layer has been done with the financial market news, from 2007 to 2017. The news of top 8 companies of IYE index has been parsed from the Reuters News, which exceeds 2000 news and 40000 words. Historical stock prices of corresponding companies were also prepared in files of json format. Stock returns are divided into 3 parts, short-term return for 1 day, mid-term for 7 days and long-term for 28 days. The return is also netted via calculating the difference between the individual company return and S&P 500 return, which is the benchmark.

The raw textual data was then processed with tokenization and lemmatization, functions offered by the Natural Language Toolkit. Adopting the TF-IDF mentioned in methodology, a quick evaluation has been conducted. Four default classifier of NLTK package has been chosen to train and test. The average result is around 60%, which is not a satisfying result since it just exceeds the benchmark. The outcome could be attributed to the deficiency of data and the defect of the algorithm, as TF-IDF only represents the pattern of word occurrence instead of the relationship between words.
<table>
<thead>
<tr>
<th>Classifier</th>
<th>Test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultinomialNB</td>
<td>0.58</td>
</tr>
<tr>
<td>BernoulliNB</td>
<td>0.62</td>
</tr>
<tr>
<td>LinearSVC</td>
<td>0.68</td>
</tr>
<tr>
<td>SVC</td>
<td>0.52</td>
</tr>
</tbody>
</table>

Table 1 - NLTK Classifier model accuracy

With the adoption of google word2vec model, each word can be transformed into a 300 dimension vector. However, the large dimension may increase the complexity and dilute the importance of significant features. As a result, an dimension reduction algorithm called t-SNE has been implemented to shrink the dimension to 3. t-SNE is a non-linear technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. Finally, each document has been format as a 41*4 matrix. The first 40 rows are top 40 representative words in an article which comprises 3 columns of vectorized result and 1 columns of sentimental scores. The last row is the corresponding stock return. The data was passed to machine learning module for further analysis.
3.3 Machine Learning

With the data analyzed by the Natural Language Processing module, the data is then used to train the machine learning models. The test result is shown in the two tables below. The first table shows the results generated by the direction based models (classification), and the second table shows the results generated by the percentage based models (regression). Note that since the team is still in the first phase of construction, the time series analysis technique is yet to be incorporated.

<table>
<thead>
<tr>
<th>Model</th>
<th>Test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Vector Machine</td>
<td>0.6667</td>
</tr>
<tr>
<td>Multi Layer Perceptron</td>
<td>0.68421054</td>
</tr>
</tbody>
</table>

Table 2 - Classification Model Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean Square Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Vector Machine</td>
<td>0.034</td>
</tr>
<tr>
<td>Multi Layer Perceptron</td>
<td>0.0016569744</td>
</tr>
</tbody>
</table>

Table 3 - Regression Model Accuracy

The benchmark for the classification performance is 0.703 (1 - 0.297), 0.297 being the proportion of 1s residing in the test data, shown in Figure 8.
Judging from the test result and the benchmark, the classification model is performing slightly below the benchmark.
4 Limitation and Difficulties encountered

This section details the limitations of the project that are brought by the external conditions, and also the difficulties that the team has encountered during the development.

4.1 Natural Language Processing

To reach the optimal performance of the model, the information lost during the dimension reduction should be reduced. However, our current algorithm consumes too much time for computing. Additionally, the size of training data is limited due to lack of hardware support.

4.2 Machine Learning

Although building up prediction models is not a particularly difficult task, given the availability of various machine learning packages, the process of deciding which features and which hyperparameters to feed into the prediction model is rather challenging. The team has tried incorporating the sentiment score as one of the input features, but no significant prediction discrepancies were found in the output. More experiments, for example, duplicating the sentiment score column to enlarge its feature significance, are to be examined in the future.

5 Future Work

This section details the remaining work of this project, and similarly, the discussion is organized by the two modules defined in the introduction.

5.1 Natural Language Processing

The future work of Natural Language Processing module consists of the enrichment of data and the adoption of the weighting algorithm. First, with the enhancement of computing power, more data will be collected and provided for model training. Second, since each word in a document may have different level of impact on the fluctuation of stock price, a weighted score should be
calculated. For example, finance-related words may have more impact on the stock price than those unrelated, and thus should have higher weights in the output feature. More work to differentiate the importance of topic words will also be conducted in the future by making the comparison of texts between the financial news and not financial related news.

5.2 Machine Learning

The future work of the machine learning module comprises two main components, the inclusion of the time series analysis technique, and the optimization of the algorithm. As stated in the Discussion of the Current Progress and Results section, currently the prediction models do not have the ability to tell how the stock trend will proceed in the next time-step, as they do not consider the timing factor in the computation. In the following semester, the team will work on integrating the time series data into both of the models. One of the possible solutions would be to adopt RNN to retain the timing factor in each loop, e.g. using Long Short Term Memory Networks (LSTM networks) to remember information for long periods of time.

As for the other component, optimization, it is critical in terms of feature and hyperparameter selection. In our current implementation, the selection process is aided by gridsearch, which refers to exhaustively searching every possible combinations. Take the SVM model as an example. Within each kernel function, there are several hyperparameters that can be tuned to improve the prediction accuracy, while exhaustively trying out all the combinations is not only time-consuming but also inefficient. Thus, the team will explore the viability of incorporating Genetic Algorithm into the training process. As shown by Choudhry et al[5], and Seiffert et al[10], the inclusion of the genetic algorithm will enhance the accuracy of the model by providing more precise weights based on its algorithm. After incorporating the Genetic Algorithm, the performance will be evaluated to determine whether applying Genetic Algorithm is suitable under the stock prediction context.
6. Conclusion

Our team has now finished building the pipelines coordinating between the Natural Language Processing Module and the Machine Learning Module, which is capable of implementing preliminary stock price predictions, of both directional movements and discrete values, based on a feature matrix derived from the word vectorization and sentiment analysis. We will continue working on optimizations for the algorithms used and will aim to establish a frontend with which users can interact and make queries based on their needs.

References

[8] Tomas Mikolov, 2013,Distributed Representations of Words and Phrases and their Compositionality

20