Predicting Stock Market Performance Using Machine Learning Techniques

**by**

Mohammad Usama Abdelnabi

Bachelor Thesis

submitted to the Department of Business Informatics

at the Faculty of Management Technology

German University in Cairo

**Student registration number/ ID**: 49-8386

**Date**: 4th of June 2023

**Supervisor**: Dr. Ayman Alserafi

Table of Contents

Table of Contents ………….………….………….………….………….………….… ii

Abbreviation List ………….………….………….………….………….……………..iii

List of Figures ………….………….………….………….………….………….…… iv

List of Tables ………….………….………….………….………….………….………v

1. Introduction ………….………….………….………….………….………….…….1
2. Literature Review………….………….………….………….………….…………. 2
   1. An Overview of Machine Learning Algorithms ………….………….………..2
   2. A Look into Previous Research ………….………….………….………….… 5
   3. Research Question of Thesis Paper ………….………….………….…………7
3. Application in Investment Management ………….………….………….…………8
4. Research Methodology ………….………….………….………….………….……9
   1. Data Retrieval and Preparation ………….………….………….………….…..9
   2. Feature Engineering ………….………….………….………….………….… 11
   3. Classification Algorithms and Ensembles ………….………….………….… 13
   4. Cross-Validation ………….………….………….………….………….…… 16
5. Results ………….………….………….………….………….………….…………18
6. Discussion………….………….………….………….………….………….…….. 21
7. Conclusion………….………….………….………….………….………….……. 25

References………….………….………….………….………….………….………….26

Appendix………….………….………….………….………….………….………….. 29

Declaration………….………….………….………….………….………….…………30

Abbreviation List

1. KNN: K-nearest neighbors
2. LR: Logistic Regression
3. LSTM: Long short-term Memory
4. ML: Machine Learning
5. NB: Naïve Bayes
6. RF: Random Forest
7. RNN: Recurrent Neural Network
8. SVM: Support Vector Machine

List of Figures

[Figure 1:A random forest structure considering multiple decision trees (Sarker, 2021, p.7) 4](#_Toc135550014)

[Figure 2:LSTM Architecture (Yadav et al., 2020, p.3) 5](#_Toc135550015)

[Figure 3: Data before smoothing 11](#_Toc135550016)

[Figure 4: Data after smoothing 11](#_Toc135550017)

[Figure 5:Code snippet of RF training mode 15](#_Toc135550018)

[Figure 6: Ensemble RF-SVM-NB training 16](#_Toc135550019)

[Figure 7:Code snippet of train-test split on one of our folds 17](#_Toc135550020)

[Figure 8:Visualization of Fold 5 21](#_Toc135550021)

[Figure 9:Visualization of Fold 6 22](#_Toc135550022)

[Figure 10:Visualization of Fold 8 23](#_Toc135550023)

List of Tables

[Table 1: Data Retrieved 2010-2019 10](#_Toc135549615)

[Table 2: Data with new features. 13](#_Toc135549616)

[Table 3: Accuracy of every model in every fold 18](#_Toc135549617)

[Table 4: Average accuracy of every fold 19](#_Toc135549618)

[Table 5: Average accuracy of every model 20](#_Toc135549619)

[Table 6: Average accuracy after Fold 5 and Fold 6 were dropped 24](#_Toc135549620)

# Introduction

The stock market is a complex and dynamic system that is influenced by various factors, including economic indicators, company performance, and investor sentiment. Predicting stock market performance has been a challenging task for financial analysts and investors for decades. Traditional asset pricing models have been used to predict stock returns based on fundamental factors such as market capitalization, book-to-market ratio, and systemic risk (Fama & French, 1993). However, with the advent of new and advanced technologies since the Fama & French paper, new methods have been introduced and utilized in the investment management world to tackle the task of accurately predicting the performance of the stock market. The most important of these technologies being Machine Learning (ML).

The potential advantages that Machine Learning algorithms can offer to investors and financial organizations are what make stock market performance predictions so important. Accurate forecasts can aid investors in making educated decisions regarding the purchasing and selling of stocks, potentially increasing returns on investment. Accurate forecasts help financial firms manage risk and make wiser investment decisions, which is another advantage.

During this thesis paper we will first conduct a more detailed Literature Review where we will take an overall look on the existing research in the area of stock market prediction using ML. From that review we will find our research gap which will guide us to our research question. We will then implement a Data Science research project to answer that research question in a comprehensive manner and present our findings in this thesis.

# Literature Review

In this section, a Literature Review will be conducted in order to establish familiarity and understanding of current research in the particular field of using ML techniques to predict stock market performance. We will use this understanding to then later on investigate a new realm of this topic that has not been as deeply discussed in previous literature.

The first section of this chapter will give an overview and brief explanation of the different ML algorithms that have been used in previous research. In the section after that we will look into how the algorithms in the first section have been used in actual previous research papers.

## An Overview of Machine Learning Algorithms

ML algorithms are divided into four main categories: Supervised Learning, Unsupervised Learning, Semi-supervised learning, and Reinforcement learning (Sarker, 2021, p. 3). However, in the area of stock market prediction, Supervised Learning is the main ML technique that is implemented by most researchers and the one that will be mainly used in this paper and hence we will be giving an overview of only these Supervised techniques.

First, we start with defining what Supervised Learning is and what it does. The goal of supervised learning is to use example input-output pairs to learn a function that maps an input to an output (Sarker, 2021, p. 3). To infer a function, as in making a prediction, it makes use of labelled training data and a variety of training samples (Sarker, 2021, p. 3).

In machine learning, classification is viewed as a supervised learning technique that also relates to a predictive modeling issue, where a class label is predicted for a given example (Sarker, 2021, p.5). There are several classification algorithms such as Naïve Bayes (NB), Logistic Regression (LR), K-nearest neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM) (Sarker, 2021, p.5-7; Ray, 2019, p. 36-39).

The naive Bayes method is based on the Bayes theorem and makes the assumption that each pair of characteristics is independent. In many real-world scenarios, such as document or text classification, spam filtering, etc., it performs well and may be utilized for both binary and multi-class categories (Sarker, 2021, p.5-7; Ray, 2019, p. 36-39).

Logistic Regression is another popular statistical model with a probabilistic basis that is used to address classification problems in machine learning. Its outcome is binomial since it yields the probability that an event will take place or not (Sarker, 2021, p.5-7; Ray, 2019, p. 36-39).

KNN is also a widely used classification ML algorithm. The "instance-based learning" or non-generalizing learning algorithm KNN is also referred to as a "lazy learning" algorithm. Instead of concentrating on creating a broad internal model, it maintains all instances corresponding to training information in n dimensions. KNN makes use of data and uses similarity metrics to categorize fresh data points (Sarker, 2021, p.5-7; Ray, 2019, p. 36-39).

In the fields of machine learning and data science, a Random Forest classifier is well recognized as an ensemble classification approach that is utilized in a variety of application domains. The "parallel ensembling" technique used in this method applies multiple decision tree classifiers simultaneously to various data set sub-samples, as illustrated in Figure 1 below, and employs majority voting or averages to determine the final result (Sarker, 2021, p.5-7; Ray, 2019, p. 36-39).

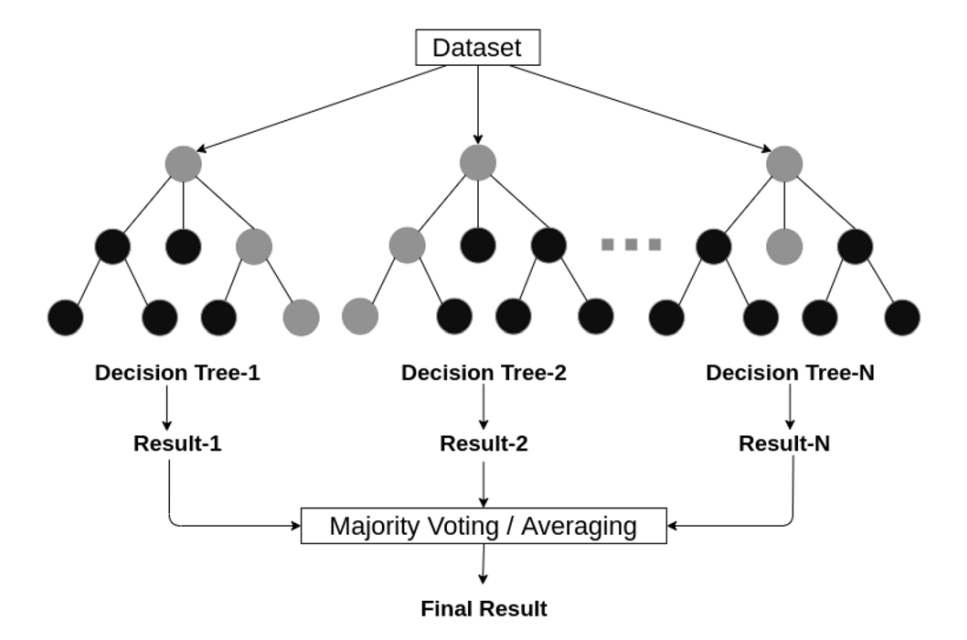


Figure 1:A random forest structure considering multiple decision trees (Sarker, 2021, p.7)

Another popular classification algorithm is the SVM. Both classification and regression issues can be handled with SVM. The decision boundary for this method is the hyperplane, which must be determined (Sarker, 2021, p. 3). A decision plane is required to divide a group of objects into their various classes when there are several of them. If the objects cannot be separated linearly, kernels—complex mathematical functions—must be used to separate the objects that belong to various classes. SVM seeks to accurately categories the objects using examples from the training data set (Sarker, 2021, p. 3).

Classification aside, another very interesting method to forecast stock market performance is through deep learning, and more specifically using a Long short-term Memory (LSTM) network (Nikou, Mansourfar, Bagherzadeh, 2019, p. 2). LSTM is a type of Recurrent Neural Network (RNN). RNNs are an effective kind of artificial neural network that can keep track of input internally (Yadav et al., 2020, p. 3). Because of this, they are especially well suited for resolving issues involving sequential data, such as a time series. But RNNs frequently experience a condition known as vanishing gradient, which causes the model learning to slow down or stop altogether. In the 1990s, LSTMs were developed as a solution to this issue. Since LSTMs have larger memory, they can learn from inputs that are far apart in time (Yadav et al., 2020, p. 3).

Three gates make up an LSTM: an input gate that selects whether to accept fresh input, a forget gate that eliminates unimportant information, and an output gate that determines what information to output (Yadav et al., 2020, p. 3). These three gates operate in the 0 to 1 range and are analogue gates based on the sigmoid function. In Figure 2 below, these three sigmoid gates are depicted. The cell state is represented by a horizontal line that passes through it (Yadav et al., 2020, p. 3).

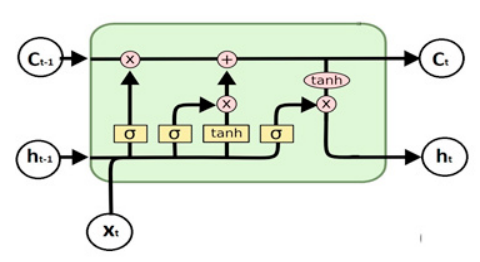


Figure 2:LSTM Architecture (Yadav et al., 2020, p.3)

After providing a comprehensive review of the ML techniques that has been cited in previous research and that will also be used in this paper, in the next section we will look more deeply at some research conducted to tackle the same problem of accurately predicting stock market performance.

## A Look into Previous Research

Different ML methods and techniques have been introduced by researchers in the realm of stock predictions. Some of the methods used classification to solve the dilemma of accurately predicting stock market performance (Ji et al., 2019). Others, for example, have put forward the use of the decision tree method in a random forest or ensemble bagging decision tree to predict returns (Tan et al, 2019; Picasso et al, 2019; Khan et al, 2019). Other researchers have chosen to study trading in the stock market through deep learning such as deep neural network, long short-term memory, recurrent neural network, and convolutional neural networks (Azhikodan et al., 2018; Yu et al, 2019; Yu & Yan, 2019; Kim et al, 2020). The results coming out of all these models have been of great interest to many stakeholders including economists, investors, portfolio managers, and even data scientists.

On the other hand, some researchers took a less numerical path and a more textual approach and implemented text mining techniques in their approach to predicting stock market data (Li et al, 2020; Khan et al 2019; Weng et al 2018; Chen et al 2018). It was also revealed through other research that the fundamentals of the company played a major role in how the stock of that company performed (Yuan et al., 2020; Hou et al., 2020)

Basak et al. (2019) proposed a novel way to minimize the risk of investment in the stock market by predicting the returns of a stock using the random forest algorithm. They used the random forest algorithm to predict the direction of stock market prices and achieved an accuracy of 78% (Basak et al., 2019, p. 11). Hernández-Nieves et al. (2021) concluded that it is possible to accurately estimate the closing value based on the current opening value of the market when Machine Learning algorithms are trained with a sufficient volume of data (Hernández-Nieves et al., 2021).

Mehtab and Sen (2021) have implemented different ML algorithms and compared them with Deep Learning techniques on historical data of the Nifty Fifty Stock Index (Mehtab & Sen, 2021, p.8). They concluded that the deep learning models performed in a superior way compared to the ordinary ML models (Mehtab & Sen, 2021, p.8). Parray et al. (2020) found that when time series supervised learning was implemented (compared to non-time-series data) there was an average increase of 2% in accuracy (Parray et al., 2020, p. 8).

A paper by Moghar and Hamiche (2020) used RNN based on LSTM to predict the performance of the stocks of Google and Nike. The results from their research were promising showing the viability of deep learning as a method to predict stock market performance (Moghar & Hamiche, 2020, p. 6).

## Research Question of Thesis Paper

After taking an overview on previous literature, we define the research gap as not enough research done in the area of using classification algorithms and ensembles of them to predict stock market performance. Hence, in this thesis, we will investigate the question of: Can classification algorithms and ensembles of classification algorithms be used to accurately predict the direction of the stock market?

# Application in Investment Management

Over the past decade or so, machine learning has led to advancements in the stock market, including the ability to use various machine learning techniques to predict stock movements in order to make the best decisions on their stock trades. This has led to recognition from by more and more stock investors as a viable method to enhance their investment strategies (Zuo, 2023, p.1).

Research shows that portfolio managers managing substantial amount of assets in the stock market for financial firms like banks, insurance companies, or investment bank benefit a whole lot from ML predictive algorithms (Henrique et al., 2019, p.3). Another research done by Faridi et al. (2022) showed that ML algorithms can be used to manage reasonable risk while also managing to score high investment returns. For these reasons many investors and portfolio managers opt to use this technology to optimize their stock portfolios (Faridi et al., 2022, p.2). Another research by Wang (2023) stresses on that previous point by concluding that ML algorithms can help investors and portfolio managers build an optimal portfolio that minimizes risk and maximizes returns (Wang, 2023, p.1).

Hence, it can be concluded that the use of ML techniques for stock market predictions can firstly help individual investors who manage their own money take better decisions in managing their own portfolios. Moreover, it can help firms on the corporate level better manage their investment funds and serve as a vital aid for portfolio managers of these funds.

The algorithms that this thesis will be presenting will test if classification algorithms and ensembles built of them can be used to predict the direction of the stock market accurately.

# Research Methodology

In this chapter, we will present the research that we have conducted in this paper. Our investigation is to find if classification methods along with ensembles built using these classification methods are a valid and accurate technique to predict the direction of the stock market on a given day.

In the first section we will look at how the data was retrieved and how it was prepared. The second section talks about how we derived new features from the existing columns we have (feature engineering). After that, we look into the classification and ensembles models that were used in this thesis and the way they were trained.

## Data Retrieval and Preparation

The stock index that we will use as a representation of the market is the well-known S&P 500, which is a market weighted index that contains the 500 most widely traded companies in the stock market of the United States. What is meant by market weighted is that the companies with the largest market capitalization have a greater impact on the index’s performance. The S&P 500 is also considered by global investors outside of the United States as an important benchmark of the global market (Xiong, 2021, p.1).

We will use the yfinance API (I) to retrieve historical stock market data for the 10-year time period spanning January 1st, 2010 till December 31st, 2019. The data retrieved is presented in Table 1 below.

Table 1: Data Retrieved 2010-2019



We conducted the appropriate Exploratory Data Analysis to clean and preprocess the data and make it ready for training and testing. A noteworthy step that was conducted that could enhance the performance of the model is smoothing. Smoothing the data can help machine learning algorithms by reducing noise and making patterns in the data more apparent. Smoothing techniques, such as exponential smoothing, can be used to remove short-term fluctuations in the data and highlight long-term trends. This can make it easier for machine learning algorithms to identify patterns and make accurate predictions. In addition, smoothing can help to reduce the impact of outliers in the data, which can improve the accuracy of the machine learning model. Figures 4 and 5 are a subset of our data before and after smoothing, respectively. As is apparent, the sharp ridges of the data have been smoothed and the trends are more apparent which will ease the job the machine learning algorithms later on in training.

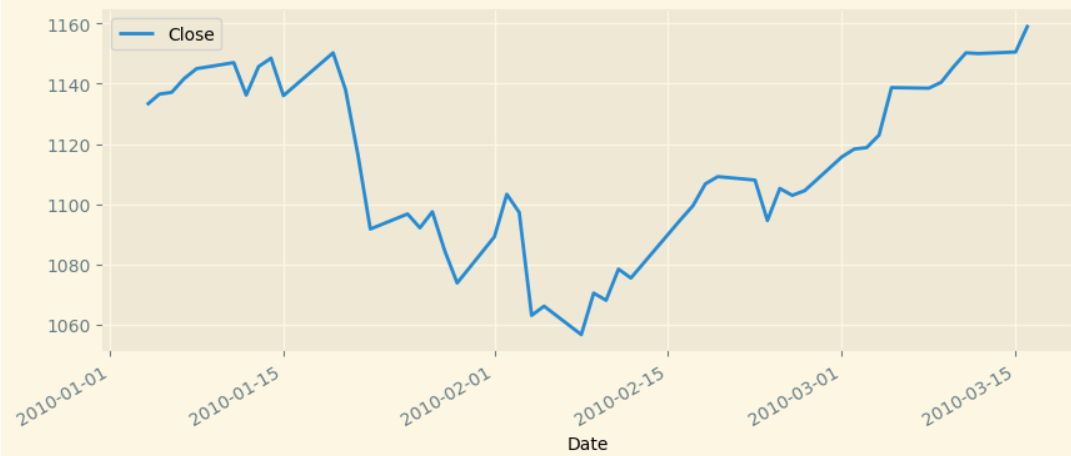


Figure 3: Data before smoothing

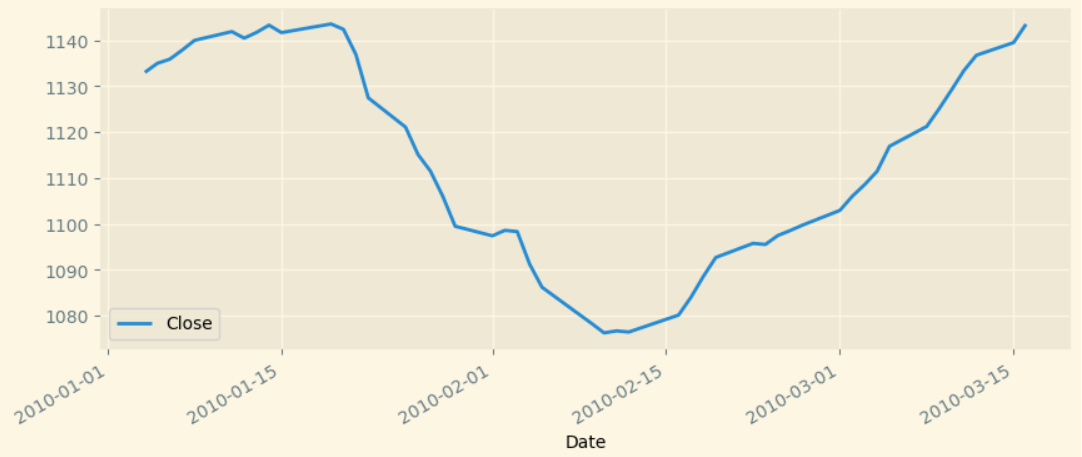


Figure 4: Data after smoothing

## Feature Engineering

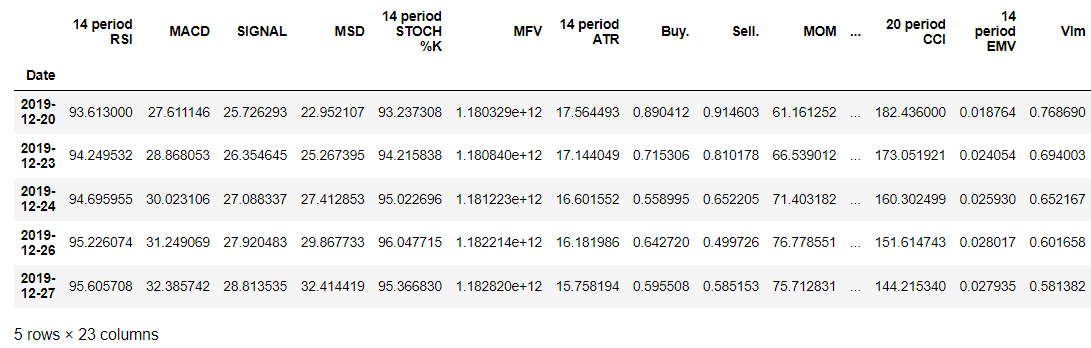
As seen from Table 1, our data has several columns. However, these columns are not good enough to provide predictive power to the ML classification algorithm that will use that data for training. Hence, we will be using the columns that we have to extract from them new features which will be of more aid to our models when it comes to training.

The new columns we will be extracting from old ones will be technical indicators of the stock index. A financial technical indicator is a mathematical formula that forecasts the direction of the financial markets using historical data on price, volume, or (in the case of futures contracts) open interest. So, what a technical indicator basically does is that it takes the closing price or the volume and runs it through a specific mathematical calculation which yields a new value that can give us new information about how the stock is performing right now and how we should expect to perform in the future. Thus, technical indicators are perfect columns that we can use as features to train our ML classification algorithms. The technical indicators that we have used in this research are: RSI, MACD, MSD, STOCH, ADL, ATR, BASP, MOM, MFI, ROC, OBV, CCI, EMV, VORTEX, EMA. Below is a brief explanation of each of them:

* RSI (Relative Strength Index): The RSI is a momentum indicator that measures the magnitude of recent price changes to evaluate overbought and oversold conditions in the price of a security.
* MACD (Moving Average Convergence Divergence): The MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price.
* MSD (Moving Standard Deviation): The MSD is a volatility indicator that measures the standard deviation of a security's price over a specified period of time.
* STOCH (Stochastic Oscillator): The Stochastic Oscillator is a momentum indicator that compares a security's closing price to its price range over a specified period of time.
* ADL (Accumulation/Distribution Line): The ADL is a volume-based indicator that measures the cumulative amount of money flowing into or out of a security.
* ATR (Average True Range): The ATR is a volatility indicator that measures the average range of a security's price over a specified period of time.
* BASP (Bollinger Bands Average Squeeze): The BASP is a volatility indicator that measures the width of Bollinger Bands over a specified period of time.
* MOM (Momentum): The MOM is a momentum indicator that measures the rate of change of a security's price over a specified period of time.
* MFI (Money Flow Index): The MFI is a volume-based momentum indicator that measures the amount of money flowing into or out of a security.
* ROC (Rate of Change): The ROC is a momentum indicator that measures the percentage change in a security's price over a specified period of time.
* OBV (On-Balance Volume): The OBV is a volume-based indicator that measures the cumulative amount of money flowing into or out of a security.
* CCI (Commodity Channel Index): The CCI is a momentum indicator that measures the difference between a security's price and its moving average over a specified period of time.
* EMV (Ease of Movement): The EMV is a volatility indicator that measures the ease with which a security's price is changing over a specified period of time.
* VORTEX (Vortex Indicator): The Vortex Indicator is a momentum indicator that measures the strength of a security's trend over a specified period of time.
* EMA (Exponential Moving Average): Exponential moving average (EMA) is a technical indicator that is used to smooth out price data by assigning a higher weighting to recent price changes. This makes EMA more responsive to changes in price than a simple moving average (SMA).

We used the Python package Finta (II) to extract from our data the aforementioned technical indicators. Our data with our new features can be seen in Table 2.

Table 2: Data with new features.



## Classification Algorithms and Ensembles

In this research 5 classification algorithms were tested individually and also in different combinations in ensembles.

The 5 algorithms are:

* RF
* KNN
* SVM
* NB
* LR

The 26 different combinations of ensembles of these 5 classification algorithms are:

* Ensemble RF-KNN
* Ensemble RF-SVM
* Ensemble RF-NB
* Ensemble RF-LR
* Ensemble KNN-SVM
* Ensemble KNN-NB
* Ensemble KNN-LR
* Ensemble SVM-NB
* Ensemble SVM-LR
* Ensemble NB-LR
* Ensemble RF-KNN-SVM
* Ensemble RF-NB-KNN
* Ensemble RF-KNN-LR
* Ensemble RF-SVM-NB
* Ensemble RF-SVM-LR
* Ensemble RF-NB-LR
* Ensemble KNN-SVM-NB
* Ensemble KNN-SVM-LR
* Ensemble KNN-NB-LR
* Ensemble SVM-NB-LR
* Ensemble RF-KNN-SVM-NB
* Ensemble RF-KNN-SVM-LR
* Ensemble RF-KNN-NB-LR
* Ensemble RF-SVM-NB-LR
* Ensemble KNN-SVM-NB-LR
* Ensemble RF-KNN-SVM-NB-LR

The individual models are trained first. We tune the hyperparameters of individual models using a method from sklearn called GridSearchCV. GridSearchCV is a technique for finding the optimal parameter values from a given set of parameters in a grid. For example, in our RF training method shown in Figure 5, we want to tune the hyperparameter ‘n\_estimators’, which is the number of trees you want to build before taking the maximum voting or averages of predictions. We gave it values ranging from 50 up until 350. For each iteration it will find the best ‘n\_estimator’ and it will use it for its predictions. The ‘cv’ parameter in GridSearchCV allows you to validate that the best parameter is indeed the best one by implementing cross-validation using it. The number you assign to that parameter is the number of folds you use in the cross validation.

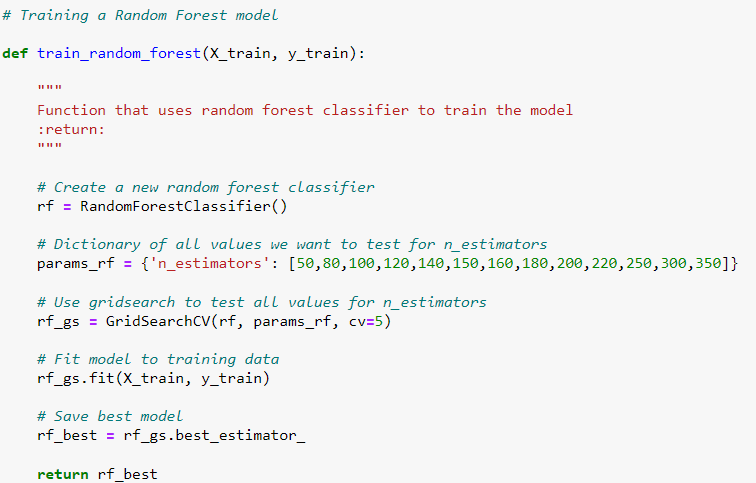


Figure 5:Code snippet of RF training mode

All individual models were built in a similar manner like that of RF shown in Figure 5 and they also had their hyperparameters tuned so that we can get the model with the best parameters. Predictions were then made and thus accuracies of each individual model were available.

Next, we need to train the ensemble models. The ensemble models have no hyperparameter tuning but we used the accuracies of the individual models that we just retrieved in the previous step as weights to the vote of each model in the ensemble. This means that the higher the accuracy of the individual model, the more weight or importance its vote will be given in the voting classifier.

Figure 6 shows an example of a method used to train an ensemble. It takes in as parameter not only the individual models, but also their accuracies to pass them to voting classifier later as weights.

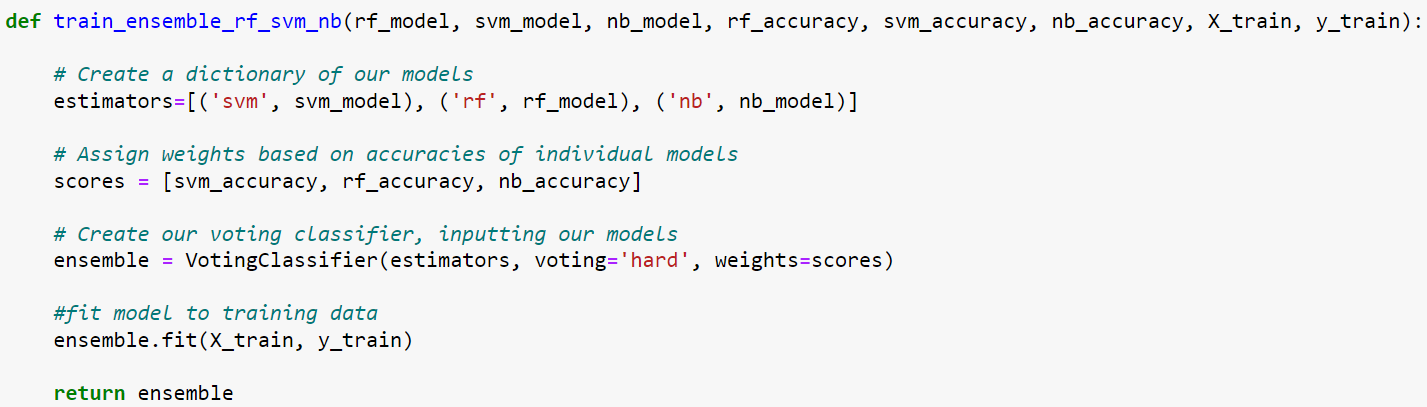


Figure 6: Ensemble RF-SVM-NB training

The remaining 25 ensembles were built in a similar manner where the models and their accuracies were passed on to a training method and a weighted classifier decided on the overall vote of the classifier.

## Cross-Validation

Due to the nature of time-series data, implementing a normal cross validation will cause the model to overperform. The reason for that is that in a normal k-fold cross-validation one fold is left out and the other k-1 are used for training regardless of whether these other k-1 folds are temporally before or after the k fold. This creates a look-ahead bias where the model is trained using data that it would normally not have as this data is in the future. For that reason, a normal k-fold cross validation will not be suitable for our case.

An alternative that we have implemented is breaking down the data into k-folds and then treating each of these folds as a separate dataset that we can have a train-test split implemented on. The most important part here is that we do not shuffle the data or any of its folds so that each fold can have the first part of it as training data and the latter part of it as test data.

As seen in Figure 7, we used the first 90% of the data as training data and the last 10% of our data as test data. We made sure that the ‘shuffle’ parameter is set to False so that the fold is not shuffled and so that it remains temporally sequential.



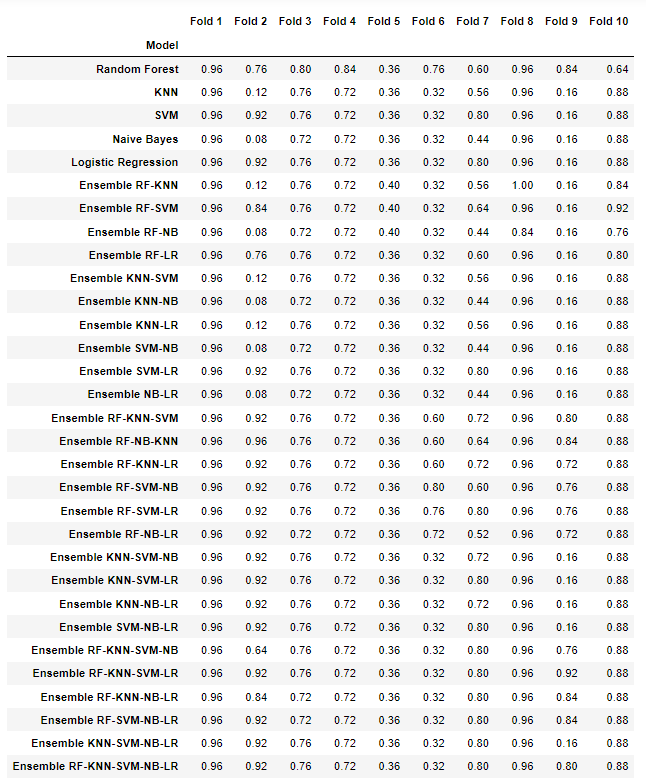
Figure 7:Code snippet of train-test split on one of our folds

We performed in this thesis a 10-fold cross validation and hence our data was separated in 10 folds where each of them had an unshuffled train-test split implemented on it and was used to train a model and a respective accuracy was calculated for each model for every fold.

# Results

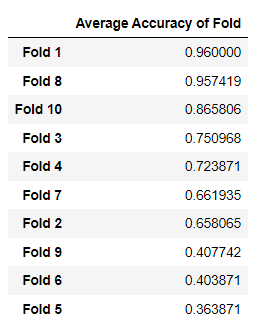
The result of each model in every fold is shown in Table 3 below.

Table 3: Accuracy of every model in every fold



The average accuracy of every fold in descending order of accuracy is shown in Table 4 below.

Table 4: Average accuracy of every fold



Finally, the average accuracy of each model in descending order of accuracy is shown in Table 5 below.

Table 5: Average accuracy of every model



# Discussion

From Table 5 we see that the best-performing model is Ensemble RF-SVM-LR with an accuracy of 78.8%. We also see that the top 5 performing models are all ensemble models and hence ensembles have proven to perform better than individual models. We also observe it is mainly 3-model ensembles that are the superstars with 4 of the 5 top spots taken by them.

It has also caught our attention how models performed substantially differently from one fold to another. As we can see, folds 5 and 6 had particularly bad accuracies on average and folds 1 and 8 performed substantially better. To get a better understanding to why that might be we visualize the data of each of folds 5, 6, and 8.

We know that fold 6 spans the time period between 2014/12/12 till 2015/12/08 and fold 5 spans the time period between 2013/12/12 and 2014/12/11. When we visualize these time periods in Figure 8 and Figure 9, we see that both periods are marked by periods of high volatility and a sharp movement either upward or downward.

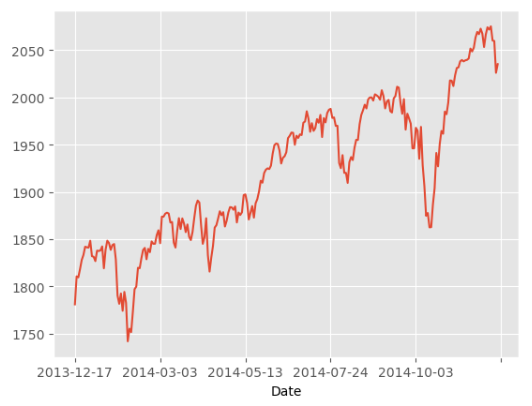


Figure 8:Visualization of Fold 5

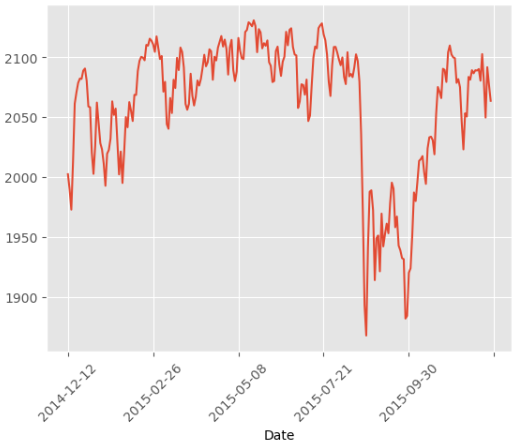


Figure 9:Visualization of Fold 6

We also know that Fold 8 spans the time period between 2016/12/05 till 2017/11/29 and if we visualize that time period, we see in Figure 10 how smooth, stable, and non-volatile the market is compared to what we have seen in Fold 5 and Fold 6.

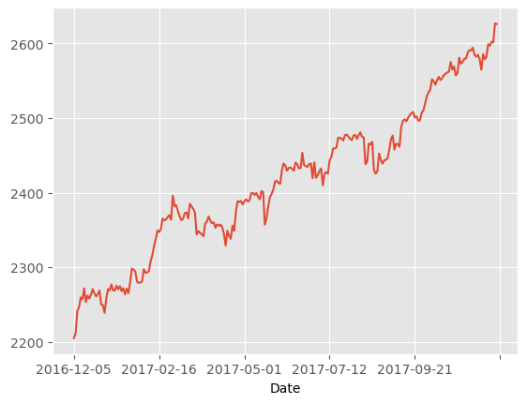


Figure 10:Visualization of Fold 8

Hence, we can conclude that the more volatile or noisy the dataset is the less likely it is that the classification algorithms or ensembles would be able to accurately predict the correct direction of the stock market. This is logical as unanticipated external information that causes a substantial increase or decrease in stock price can be hard to be predicted due to the very nature of them being unanticipated.

If the volatile folds, like Fold 5 and Fold 6, were dropped from the results table the average accuracy of each model will be significantly higher and the accuracy rankings of the models will also change. In Table 6 we see the accuracy results of the models after Fold 5 and Fold 6 were dropped sorted in descending order. We can think of this table as the accuracy results of the models during normal or non-volatile times.

What we realize is that in normal or non-volatile times 4-model ensembles and the 5-model ensemble have become the new superstars as they take the top 3 spots. The best performing model is now Ensemble RF-KNN-SVM-LR boasting an impressive accuracy rate of 86.5%.

Table 6: Average accuracy after Fold 5 and Fold 6 were dropped



# Conclusion

In this thesis, we have conducted research to test the ability of classification algorithms and ensemble learners built using them, to predict the direction of the stock market. We found that, in general, the model that performs best is Ensemble RV-SVM-LR with an accuracy of 78.8%.

We also found, through the inspection of the average accuracy of every fold, that folds which are more volatile or can be characterized as more noisy trouble all models and are very hard to predict, and hence have the lowest accuracy scores. But volatile folds aside, we found that it is actually another model that performs best in normal non-volatile times and that is Ensemble RK-KNN-SVM-LR with an accuracy of 86.5%.

Hence, we conclude the following: Under normal circumstances Ensemble models built using classification models do a great job in predicting the direction of the stock market. Ensemble RK-KNN-SVM-LR model is specifically recommended to be used during a normal non-volatile market. During a choppier market with more volatility the Ensemble RV-SVM-LR can prove to be more reliable. However, that too is not guaranteed as although it scored an accuracy of 76% with Fold 6, it scored a disappointing 32% with Fold 5.

References

1. Azhikodan, A. R., Bhat, A. G., & Jadhav, M. V. (2018). Stock trading bot using deep reinforcement learning. *Innovations in Computer Science and Engineering*, 41–49. <https://doi.org/10.1007/978-981-10-8201-6_5>. Springer Nature.
2. Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552–567. <https://doi.org/10.1016/j.najef.2018.06.013>. Elsevier.
3. Chen, W., Yeo, C. K., Lau, C. T., & Lee, B. S. (2018). Leveraging Social Media News to predict stock index movement using RNN-Boost. *Data & Knowledge Engineering*, *118*, 14–24. <https://doi.org/10.1016/j.datak.2018.08.003>. Elsevier.
4. Fama, E. F., & French, K. R. (1993a). Common risk factors in the returns on stocks and Bonds. *Journal of Financial Economics*, *33*(1), 3–56. <https://doi.org/10.1016/0304-405x(93)90023-5>. Elsevier.
5. Faridi, S., Madanchi Zaj, M., Daneshvar, A., Shahverdiani, S., & Rahnamay Roodposhti, F. (2022). Portfolio rebalancing based on a combined method of ensemble machine learning and genetic algorithm. *Journal of Financial Reporting and Accounting*, *21*(1), 105–125. <https://doi.org/10.1108/jfra-11-2021-0413>. Emerald.
6. Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226–251. <https://doi.org/10.1016/j.eswa.2019.01.012>. Elsevier.
7. Hernández-Nieves, E., Parra-Domínguez, J., Chamoso, P., Rodríguez-González, S., & Corchado, J. M. (2021). A data mining and Analysis Platform for investment recommendations. *Electronics*, *10*(7), 859. <https://doi.org/10.3390/electronics10070859>. MDPI.
8. Hou, X., Wang, K., Zhang, J., & Wei, Z. (2020). An enriched time-series forecasting framework for long-short portfolio strategy. *IEEE Access*, *8*, 31992–32002. <https://doi.org/10.1109/access.2020.2973037>. IEEE.
9. Ji, C., Zhao, C., Liu, S., Yang, C., Pan, L., Wu, L., & Meng, X. (2019). A fast shapelet selection algorithm for Time Series classification. *Computer Networks*, *148*, 231–240. <https://doi.org/10.1016/j.comnet.2018.11.031>. Elsevier.
10. Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2020). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, *13*(7), 3433–3456. <https://doi.org/10.1007/s12652-020-01839-w>. Springer Nature.
11. Khan, W., Malik, U., Ghazanfar, M. A., Azam, M. A., Alyoubi, K. H., & Alfakeeh, A. S. (2019). Predicting stock market trends using machine learning algorithms via public sentiment and Political Situation Analysis. *Soft Computing*, *24*(15), 11019–11043. <https://doi.org/10.1007/s00500-019-04347-y>. Springer Nature.
12. Kim, S., Ku, S., Chang, W., & Song, J. W. (2020). Predicting the direction of US stock prices using effective transfer entropy and Machine Learning Techniques. *IEEE Access*, *8*, 111660–111682. <https://doi.org/10.1109/access.2020.3002174>. IEEE.
13. Li, X., Wu, P., & Wang, W. (2020). Incorporating stock prices and news sentiments for stock market prediction: A case of hong kong. *Information Processing & Management*, *57*(5), 102212. <https://doi.org/10.1016/j.ipm.2020.102212>. Elsevier.
14. Mehtab, S., & Sen, J. (2021). *Stock Price Prediction Using Convolutional Neural Networks on a Multivariate Time Series*. <https://doi.org/10.36227/techrxiv.15088734.v1>. IEEE.
15. Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science*, *170*, 1168–1173. <https://doi.org/10.1016/j.procs.2020.03.049>. Elsevier.
16. Mohapatra, S., Mukherjee, R., Roy, A., Sengupta, A., & Puniyani, A. (2022). Can ensemble machine learning methods predict stock returns for Indian banks using technical indicators? *Journal of Risk and Financial Management*, *15*(8), 350. <https://doi.org/10.3390/jrfm15080350>. MDPI.
17. Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, *26*(4), 164–174. <https://doi.org/10.1002/isaf.1459>. Wiley Publications.
18. Parray, I. R., Khurana, S. S., Kumar, M., & Altalbe, A. A. (2020). Time series data analysis of stock price movement using Machine Learning Techniques. *Soft Computing*, *24*(21), 16509–16517. <https://doi.org/10.1007/s00500-020-04957-x>. Springer Nature.
19. Picasso, A., Merello, S., Ma, Y., Oneto, L., & Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications*, *135*, 60–70. <https://doi.org/10.1016/j.eswa.2019.06.014>. Elsevier.
20. Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. <https://doi.org/10.1109/comitcon.2019.8862451>. IEEE.
21. Sarker, I. H. (2021). *Machine Learning: Algorithms, Real-World Applications and Research Directions*. <https://doi.org/10.20944/preprints202103.0216.v1>. Springer Nature.
22. Tan, Z., Yan, Z., & Zhu, G. (2019). Stock selection with Random Forest: An exploitation of excess return in the Chinese stock market. *Heliyon*, *5*(8). <https://doi.org/10.1016/j.heliyon.2019.e02310>. Elsevier.
23. Wang, Y. (2023). Review: Application of machine learning to investment portfolios. *BCP Business & Management*, *38*, 3494–3498. <https://doi.org/10.54691/bcpbm.v38i.4351>. BCP.
24. Weng, B., Lu, L., Wang, X., Megahed, F. M., & Martinez, W. (2018). Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications*, *112*, 258–273. <https://doi.org/10.1016/j.eswa.2018.06.016>. Elsevier.
25. Xiong, Q. (2021). Forecast on S&P 500 index based on Har-RV model. *Proceedings of the 2021 3rd International Conference on Economic Management and Cultural Industry (ICEMCI 2021)*. <https://doi.org/10.2991/assehr.k.211209.217>. Atlantis Press.
26. Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian Stock Market. *Procedia Computer Science*, *167*, 2091–2100. <https://doi.org/10.1016/j.procs.2020.03.257>. Elsevier.
27. Yu, P., & Yan, X. (2019). Stock price prediction based on Deep Neural Networks. *Neural Computing and Applications*, *32*(6), 1609–1628. <https://doi.org/10.1007/s00521-019-04212-x>. Springer Nature.
28. Yu, S.-S., Chu, S.-W., Chan, Y.-K., & Wang, C.-M. (2019). Share price trend prediction using CRNN with LSTM structure. *Smart Science*, *7*(3), 189–197. <https://doi.org/10.1080/23080477.2019.1605474>. Smart Science.
29. Yuan, X., Yuan, J., Jiang, T., & Ain, Q. U. (2020). Integrated long-term stock selection models based on feature selection and machine learning algorithms for China Stock Market. *IEEE Access*, *8*, 22672–22685. <https://doi.org/10.1109/access.2020.2969293>. IEEE.
30. Zuo, X. (2023). Prediction of facebook and GOOG prices based on linear regression and LSTM regression. *BCP Business & Management*, *44*, 688–695. <https://doi.org/10.54691/bcpbm.v44i.4919>. BCP.

Appendix

1. Link to API used to retrieve stock market data: <https://pypi.org/project/yfinance/>.
2. Link of Python package ‘Finta’ used to extract technical indicators from data: <https://pypi.org/project/finta/>.
3. Link to GitHub repository hosting the code of the thesis project: <https://github.com/mabedelnabi/Thesis-Project>.

Declaration

I herewith declare that this report is in full accordance with the Plagiarism Guidelines of the Faculty of Management & Technology at the GUC.

Signature

Mohammad Usama