Stock Market Prediction using Ensemble Learning

by

Rakin Bin Rabbani
16101213
B. M. Fahad-ul-Amin
16101251
Sumaiya Tanjil Khan
16101212
Farjiya Benta Mahbub
16301033

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering.

Department of Computer Science and Engineering
Brac University
April 2020

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

B. M. Fahad-ul-Amin
16101251

Rakin Bin Rabbani
16101213

Sumaiya Tanjil Khan
16101212

Farjiya Benta Mahbub
16301033

Approval

The thesis titled “Stock Market Prediction using Ensemble Learning” submitted by

1. Rakin Bin Rabbani (16101213)
2. B. M. Fahad-ul-Amin (16101251)
3. Sumaiya Tanjil Khan (16101212)
4. Farjiya Benta Mahbub (16301033)

Of Spring, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on April 7, 2020.

Examining Committee:

Supervisor:
(Member)

Mahbub Majumdar, PhD
Senior Professor and Chairperson(CSE), Interim Dean(School of Sciences)
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Mahbub Majumdar, PhD
Senior Professor and Chairperson(CSE), Interim Dean(School of Sciences)
Department of Computer Science and Engineering
Brac University

Abstract

An unpredictable sector of finance market which involves three major roles: investors, buyers and sellers is called a stock market. Also stock prices may not only change the future economy of a country but also have direct effects on the current economic activities of the country. Forecasting stock market is acquiring more attention due to its expected high profit. But the prediction part of stock markets is considered as quite a challenging task. Though there are various techniques available for forecasting stock price but the number of methods for forecasting the stock market accurately is less than usual. Another way of determining future value as in rise or fall of the future stock price is known as data analysis. The purpose of this paper is to discuss how accurately the price of stocks in the US stock market can be predicted, by generating the best possible factors for particular stocks in the US stock market using machine learning algorithms. After conducting our preliminary research and then some, we found that it is quite difficult to predict the price fluctuations of stocks as the market is highly volatile. Furthermore, the number of uncertain variables in the equation which makes it hard to isolate any one or few factors that can be used to accurately predict price fluctuations. Therefore, for our trial runs, we tried to isolate the best factors that can be used to predict the prices of stocks with sufficient accuracy. For our approach, we implemented Gradient Boosting, Random Forest, Naive Bayes, AdaBoost, Logistic Regression and SVM to run on our dataset. Based on the outcome of these algorithms we will take the decision whether to go long or short for a particular stock.

Keywords: *Stock Prediction, Machine Learning, Time Series, US Stock market, stocks, Gradient Boosting, Random Forest, Naive Bayes, AdaBoost, Logistic Regression, SVM, Feature reduction*

Dedication

We dedicate this project to Allah Almighty our creator, our strong pillar, our source of inspiration, wisdom, knowledge and understanding. Allah has been the source of our strength throughout this program. We also dedicate this work to our respected teachers and our parents who has encouraged us all the way and made sure that we give it all it takes to finish that which we have started.

Acknowledgement

At first, all praise to the Great Allah for whom our thesis have been completed without any major interrupt.

We wish to express our sincere thanks to our supervisior Professor Mahbub Majumdar of the Computer Science and Engineering department at BRAC University for providing us with all the necessary facilities for the research.He always encouraged us to think innovatively and bring new ideas.He has been a constant source of encouragement and enthusiasm throughout the time of our thesis.

We are also hugely indebted to Golam Rabiul Alam, Associate Professor of BRAC University, for sharing his ideas and experiences which helped us to make right choices in the implementation phase.

We take this opportunity to express gratitude to all of the Department faculty members for their help and support. We also thank our parents for the encouragement, support and attention.Also the cooperation and understanding of our group mate has helped us to finish our thesis without any hassel. We also like thank all of our friends who always inspired and motivated us.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ϵ	Epsilon
Θ	Theta
ANN	Artificial Neural Network
$ARIMA$	Autoregressive Integrated Moving Average
D	Day
GA	Genetic Algorithm
GB	Gradient Boosting
GMM	Generalized Methods of Moments
HMM	Hidden Markov model
LDA	Linear Discriminant Algorithm
$LS-SVM$	Least square Support vector Machine
$LSTM$	Long-short term Memory
M	Month
ML	Machine Learning
OOB	Out Of Bag
PNN	probabilistic Neural Network
PSO	Particle Swarm Optimization
SVM	Support Vector Machine
SVR	Support vector Regression

Chapter 1

Introduction

1.1 Background

Public markets which are for issuing, buying and selling stocks that can be traded on a stock exchange or over-the-counter are referred as the stock market [15]. Stocks refer to equities which represent fractional ownership in a company and the stock market is a platform where ownership of investable assets can be bought or sold by the investors. An efficient stock market can create impact on economic development since the companies can influence the public and make capital from them.

1.1.1 Classification of stock

There are different kinds of parameters depending on which stocks can be classified. Some of these parameters are -size of the company, industry, risk, dividend payment, volatility as well as fundamentals.

One of the most common parameter is Stocks which are based on the ownership rules. Here, the issuing company takes decision whether it will issue preferred, common or hybrid stocks. Though preferred and common stocks are quite similar yet the difference between these two is that investors will get certain amount of dividends per year from the preferred stock whereas no such amount is given is common stock. As a result, preferred stock is more volatile in nature than the common stock in terms of price. Moreover, Hybrid stock is the combination of both preferred stock and common stock which means preferred shares that can be converted into a fixed number of common stocks at a particular time[19].

Dividend payments are another parameter for classifying stock market. Dividends are percentage of income that is generated when shares are sold at a profit. First of all, Income stocks are the stocks that provide a higher dividend in relation to their share price. So, a higher dividend means larger income. Stable companies that distribute consistent dividend are referred as income stocks. Since these companies are not that stable so the risk associated with the stock price is less. Risky stocks are more potent in making greater profits. On the contrary, Low-risk stocks generate lower returns. The second type of stock under this category is Blue-chip stocks which are considered to be safe since renowned companies hold these type of stocks. As the companies does not have much liabilities which help them to have regular dividends. Furthermore, Beta stocks are a measure to indicate the risk or volatility of stock in terms of beta. A positive value of beta refers it is in sync with the market and a negative beta value implies that it is against the market. Though the absolute value of beta is more important. Higher the beta, greater the volatility and thus more the risk. A beta value over 1 means the stock is more volatile than the market [22].

1.1.2 Stock Price Setting Process

The worth of a company which is also known as market capitalization is determined by the product of the company's stock price and the number of shares outstanding. [28] Thus setting the price for stocks is very important. There are many factors by which share price can be determined. Among these factors the most used factors are herd instinct and market news.

• Herd Instinct: The price of stock can also be influenced by what is known as herd instinct, a tendency for individuals to mimic the behavior of a larger group. For instance, as more and more people purchase a stock, driving the price higher and higher, other people will jump on board, thinking all the other investors have to be right (or know something not everyone else knows).

• Market News: Other factors may have impact on prices, and can result in sudden or temporary price changes. Types include quarterly releases, political activities, business material activities and economic news. All sorts of news or economic events do not influence all securities as in sometimes it is affected by weekly news but not by monthly status [26].

These are some of the ways to set the stock price. Apart from that Supply and Demand is also another factor to set stock price.

1.1.3 Effect of stock price

Economics demand and supply law affects stock prices. The demand and supply law says that if there is high demand, the price of that material will increase and the other way around for high supply. Likewise, if there is a lot of asking price for a stock, the price of that stock increases and if a stock has more than usual offering price that stock price will decrease. The price of a company's stock is more affected by the law of supply and demand rather than the company's success. Whether a company's stock is desirable or not depends on a variety of reasons such as it can be strength of an industry sector or the popularity of a brand.

• Cyclical stocks: In a disrupted economy some companies' growth is moderating or fastens in a booming economy thus some companies are affected by economic patterns. As a result, prices of these stocks continue to fluctuate more with changing economic conditions. Companies are rising during economic booms and dropping as the economy slows. automobile companies are the best example of cyclical stocks.

• Defensive stocks: Unlike cyclical stocks, defensive stocks are not affected by economic conditions issue. Best examples are stocks of food, beverage, pharmacy and insurance firms. These stocks are usually preferred in weak economic conditions, while cyclical stocks are preferred in booming economies.

1.1.4 Strategies of Investment

As the nature of the stock market is dynamic which makes the share market very vulnerable. So for any investor it is very important to make the right investment at the right time. The convention is to sell a share when the price is high and buy it when the price is low. There are five strategies which investors follow while investing in any share. Those are :-

• Buying Cheap and Selling Dear: Coming into the market when prices and sentiment are depressed and selling out when both are exalted.

• Long-Pull Selection: Picking out companies which will prosper over the years far more than the average enterprise also known as "growth stocks".

• Bargain Purchases: Selecting shares which are selling considerably below their true value, as measured by reasonably dependable techniques.

• General Trading: Anticipating or participating in the moves of the market as a whole, as reflected in the familiar "averages".

• Selective Trading: Picking out stocks which over a period of a year or less will do better than the market.

• Short Selling: Short-selling also known as shorting is when an investor borrows shares from a shareholder and sells them immediately, hoping that he or she will be able to buy them again at a cheaper price later on, return them to the lender and pocket the profit [17].

Each approach requires a rational, disciplined, consistent, systematic application. In a nut shell the whole idea of stock market is to make profit. Thus investors are so much focused so that they can anticipate the stock market and make their Investment accordingly.

1.2 Motivation

To make any business decision investors are required to analyze their stock market. But doing that all by their self is quite challenging and sometimes are not capable to do that. Alternatively, they have to focus on observing others investors stocks too. Again if the investors want to do it on their own, they face the difficulties of collecting the data, analytical tools as it is difficult to predict the share market. But in today's world using machine learning techniques and algorithms used in artificial intelligence it is possible to forecast the stock market quite well and the accuracy of this prediction largely depends on the available amount of data as well as how well the model has been trained. For this purpose, we are using quantopian platform which provides huge amount of financial data. To predict the dynamical stock market quantopian is used where we can provide our algorithms and test the accuracy. So the main goal of our thesis is to help investors to predict the stock market correctly.

1.3 Objective

The process of making decision based on the previous experience and available information is known as prediction. And because of the dynamic nature of the stock market predicting stock market accurately is a very difficult task to do as we discussed before. The easier way to decide for the future on an option totally depends on the accuracy of the prediction. Moreover, stock market has a positive relation with the country's finance cause if stock markets rise, the financial growth of a country would be high and vice-versa. As stock market has great impact on the economy many has tried to predict its nature previously. Also there is many available machine learning methods such as Neural Network, Evolutionary Algorithms, SVM, Neuro-Fuzzy, HMM, ARIMA, Decision Tree, Random Forest, Adaboost, GB, Linear regression, Logistic Regression, Naive Bayes etc. In this paper we have focused to create a system which can provide more accurate result and able to predict share market with less amount of hassle. To achieve our goal, we intend to use data which cost nothing as we are using quantopian platform.

1.4 Paper orientation

This section mainly focusses on the brief description of our entire paper and the steps which we have taken to implement machine learning algorithm and related works to predict the stock market. In the next chapter Literature review is described and the information from the past papers have helped us to acquire vast knowledge on this topic. Then the following section is about the machine learning algorithm . Chapter four is about the data processing and inplementation of the required algorithm. The result and the conclusion of our thesis is discussed in chapter six and seven accordingly.

Chapter 2

Literature Review

In this chapter we will discuss about the study we did on previous papers of stock market prediction and how our work is improved considering the limitations from previous papers.

2.1 Previous Researches

Stock market prediction has been the ultimate objective for many investors. The changes in the companies are dependent on how the market reacts [20]. There were quite a few different implementations of machine learning algorithms for the purposes of making stock market price predictions. Different papers experimented with different machine learning algorithms that they implemented in order to figure out which models produced the best results [29].

In [32] Tianxin Dai et al. experimented with two machine learning algorithms SVM and Logistic regression. They trained the model iteratively to make the prediction as accurate as possible. After running the baseline model of Logistic regression algorithm they got 0.0123 profit, 55.07% precision 30.05% recall 38.39% accuracy. Similarly, λ -HL of Logistic regression algorithm gave them 0.0186 profit, 59.39% precision 27.43% recall 41.58% accuracy. After training the model for one hour by the help of SVM, they discovered 0.0926 profit, 56.45% precision, 38.10% recall and 43.92% accuracy. The use of machine learning algorithm for the forecasting stock market is quite challenging since various factors and market trends must to be taken into account.

In another paper [21] Alice Zheng and Jack Jin came up with the idea that traders can use multiple analysis techniques such as fundamental analysis, technical analysis and quantitative analysis to future stock price of their company for the purpose of training. To predict stock market they choose to work with four machine learning algorithms which are SVM, Bayesian Network, Simple Neural Network and Logistic Regression. In SVM they were forced to work with smaller dataset since Gaussian Kernel maps infinite dimensional features for their predictors which lead to time complexity of $O({n^2}_{\text{samples}})$. They found out in their output, the rate of error were 40% to 50% which implies that their choice of predictors were not appropriate.

Guanting Chen et al. [31] mentions that Deep Learning has been used widely and demonstrated to be an amazing machine learning tool now a days. Among the deep learning algorithms they selected LSTM algorithm to forecast the future returns of the stock price. They used TensorFlow to experiment on their dataset. They face some constraints as they could not articulate with different neurons and layers in LSTM.

SVM is used to reduce the input dimension for load forecasting [8]. On the other hand Least Square Support Vector Machine is a very famous SVM algorithm for predicting the stock market. In [14] paper, we found that they used LS-SVM along with PSO in some stock sectors of various companies to predict the daily stock price. They concluded by stating that the error rate was less because of using both algorithms together rather than using only LS-SVM.

In [16] they build a predictive model by combining four machine learning algorithms which are Artificial Neural Network (ANN), Support Vector Machine (SVM), random forest and Naive Bayes in the Indian stock market. Firstly, to input data they considered the computation of ten technical parameters. Secondly they tried to represent these technical parameters as trend deterministic data. After the evaluation of that predictive model it has been seen that random forest algorithm dominated on the overall performance.

To determine how a financial market behaves, a hybrid model was proposed in [9] MR Hasan's paper by accommodating HMM, ANN and GA which helps to analyze the stock market in depth and optimize the required parameters.

To make stock market prediction and trend analysis automated Abraham, Nath and Mahanti established a hybrid computing method. They mainly focused on predicting one day ahead by using Nasdaq-100 index of the Nasdaq stock market along with neural networks. An encouraging result was found when they used a neuro-fuzzy system to analyze the anticipated values [5].

Chen introduced investment strategies based on a feed forward neural network, PNN. Historical data has been used to train this model to generate better results in comparison to the one which were estimated after using the random walk and parametric GMM models [7].

Besides using machine learning algorithms, data mining techniques had also been used for the prediction purpose of the dynamic stock market. Ou and Wang tried to predict the Hong Kong stock market by the help of ten data mining techniques in total. Some of those are LDA, K-nearest neighbor classification, kernel estimation based Naive Bayes, SVM, LS-SVM etc [11].

Another Prediction System was introduced for Taiwan Stock market which is basically a recurrent neural network trained by extracting features through ARIMA [3]. Autocorrelation and partial correlation function plots are examined just after differencing the raw data of the TSEWSI series. Second difference data are used to train neural networks for better prediction. The result showed that network which were trained using four-year weekly data can forecast accurately for six weeks.

In another paper [10] Sheng-Hsun Hsu et al. talked about the dynamic relationship between stock price series and their predictors. To identify this problem with the help of only one artificial technique is quite challenging. So, a two-stage architecture came into help. The first stage is SOM which is used to break the input dataset into clusters as in data points are grouped together on the basis of similar statistical distributions. Afterwards SVR is applied on these clusters to foresee financial indices. The two-stage architecture proved to provide more enhanced outcomes.

Jyotirmoy [18] in his paper proposed sentiment analysis and supervised machine learning to predict the security movements in stock market and the profit from it. Sentiment analysis is run on data collected from sources such that newspapers, articles and blogs to interpret the sentiment from the data. The limitation they face is to collect minute data because it is expensive and not available in public domain. This helps to find out the link between the market sentiment and market movement.

Marshall [25] observed best performance using a Support Vector Machine (SVM) with a radial basis kernel, when compared with Logistic Regression, Bayesian Network, and a Simple Neural Network. Due to their limited processing power, they were only able to use a subset of their data for training their model, and recommended that a more powerful processor be used to achieve better results.

On [27] it was discussed that when performing stock price prediction, [30] it came out to be that ANN the algorithm that was once popular for prediction suffers from overfitting due to large numbers of parameters that it needs to fix. This is where support vector machine (SVM) came into play and it was suggested that this method could be used as an alternative to avoid such limitations, where according the VC theory [13] SVM computes globally obtained solution unlike those that are obtained through ANN which almost in all the cases tends to fall in the local minima. It was seen that using an SVM model the accuracy of the predicted output came out to be around 57% [4].

Since a random forest ensembles multiple decision trees. In [12] it has proposed ensemble techniques in order to lessen the over-fitting problem of training data. This technique is mainly used for the analysis purpose of Stock returns.

Moreover, K. Sharif and M. Abu-Ghazaleh [33] came up with an advancements in the machine learning field which lead to ensemble methods used for forecasting. Quantopian has its simulator which allows ensemble methods for prediction and trading on a daily basis. The output from each simulation is compared against an appropriate standard and performance evaluation is done accordingly.

2.2 Our improvements

We have observed various algorithms which were implemented to predict stock market but all these algorithm had some shortcomings. So, we worked on it and tried to improvise it by using a different approach which will be efficient for the prediction. Since we chose ensemble learning so for that we selected five machine learning algorithms. These are - Support Vector Machine(SVM), Random Forest, Naive Bayes, Logistic Regression, Adaboost. All these are highly used to financial time series.

At first we chose SVM but later we did not work with it further because SVM takes a longer time and required smaller dataset is used to give the result. Then we moved to Random Forest and this gave us a more accurate result as it deals with bootstrapping and each time it works on different set of data which increases the accuracy overall. After that we tried other two algorithms as in Logistic Regression and Adaboost. But Adaboost had some limitations in providing the result up to the mark. So, for that we worked with an advanced version of Adaboost which is Gradient Boosting as it is much more flexible in terms of giving the result. Assumption of independent predictors is the main constraint of Naive Bayes. It assumes implicitly that all the attributes have mutual independence which is not possible in real life scenario. Considering all these we decided to mix and match the algorithms and then calculate the average of the results given by each of the algorithms.

Chapter 3

Machine Learning Algorithm

In this chapter side the machine learning algorithm we have chosen to use will be discussed. As there are various machine learning algorithms are available but we decide to work with Logistic Regression, Random Forest, Adaboost, Gradient Boosting, Stochastic Gradient Descent and Naive Bayes Algorithm.

3.1 Support vector Machine

Support vector machine in short SVM is a classification and regression based algorithm. It is used to maximize predictive accuracy whilst avoiding the overfitting of data. It is used for applications such as handwriting, face, text and hypertext classification, Bioinformatics etc [5]. SVM is used to achieve maximum separation between data points. Hyperplane is a part of SVM that maximize the separation of data points by increasing the line width with increments. It starts by drawing a line and two equidistant parallel lines. Next the algorithm picks a stopping point so that the algorithm does not run into an infinite loop and also picks an expanding factor close to 1 example 0.99. The main purpose of SVM is to determine a hyperplane in an N-dimensional space that distinctly classifies the data points, here n is the number of chosen features. Although there is various possible hyperplane in support

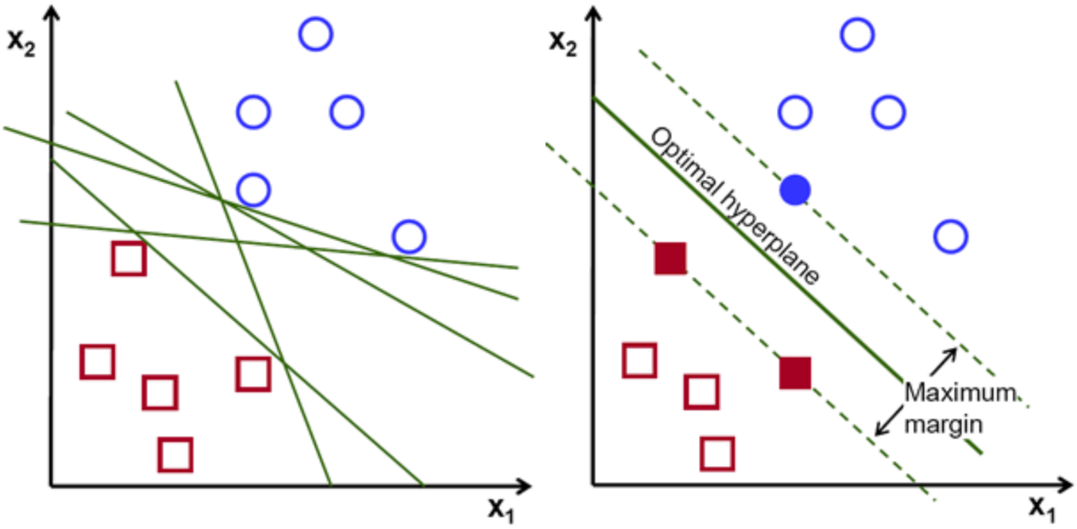


Figure 3.1: *Possible Hyperplane*

vector algorithm but the best one is that hyperplane which has maximum margin. In other word, the best hyperplane is the one which provides maximum distance between data points for both classes. As these will result in some reinforcement so that future data points can be classified with more confidence. The loss function that helps maximize the margin is hinge loss. To maximize the margin between hyperplane and data points the following equation is used:

$$c(x, y, f(x)) = 1 - y * f(x) \tag{3.1}$$

The hinge loss function is equal to 0 when y*f(x) becomes greater or equal 1. All though the support vector algorithm is very useful but it took lots of time to run in quantopian. So in spite of many advantages we chose not to use it in our ensemble learning.

3.2 Recurrent neural network

Recurrent Neural Network (RNN) is a form of deep supervised learning. RNN uses the previous memorized memory makes sense on the current data as in the output of the previous step is fed as the input for the current step. The same process continues and results in a loop. The network that is created by the help of the loop is called the recurrent network. There are three types of RNN:

1. One to Many : For one input there are multiple outputs.
2. Many to One : For many input there only one output.
3. Many to Many : For many input there are multiple outputs.

All inputs and outputs are independent of each other in conventional neural networks, but in cases such as when it is important to predict the next word of a sentence, the preceding words are needed to be remembered. Thus RNN came into being with the aid of a Hidden Layer which solved this problem. Hidden layer is the key and most significant aspect of RNN which retains certain details about a sequence. RNN is best for time series data as it can work on both stationary and non-stationary data. Traditional neural networks do not have persistence whereas RNN address this issue since they are networks with loops that allow information to persist. RNN occurs with recursive formula as it works in loop.

$$s(t) = f_w(s_{t-1}, x_t) \tag{3.2}$$

here, $\mathrm{s_t}$=state at time step t
$\mathrm{x_t}$=input at time step at t
$\mathrm{f_w}$=recursive function

$$s_t = \tanh(w_s s_{t-1} + w_x x_t) \tag{3.3}$$

here, tanh is the activation function
$\mathrm{w_x}$ and $\mathrm{w_s}$ = weight But there are some shortcomings in RNN as in when the time gap grows the density of information increases so it becomes impossible to learn to collect these information altogether. Another limitation is vanishing gradient where each layer's weight is multiplied but after certain time the updated weight will become close to zero as in it will get vanished so the whole training network will suffer for this.

3.3 Naive Bayes Algorithm

Naive Bayes, a classification technique based on Bayes' Theorem. This classifications assumes independent predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayes classifiers are formed by a group of classification algorithms based on Bayes' Theorem. It is not represented as a single algorithm rather a common principle is shared by a family of algorithm, i.e. every pair of features being classified is independent of each other.

One example will help to understand this clearly: a flower may be considered to be a rose if it is red, have round petals, and have good fragrance. Even if these features are dependent on each other as in on the existence of the other features, all of these features contribute independently to the probability that this flower is a rose and that is why it is known as 'Naive'.

The construction of Naive Bayes model is easy and it can be used for very large data sets. Along with simplicity, Naive Bayes performs really well in comparison to highly sophisticated classification methods. Bayes rule is used in Naive Bayes classification.

$$P(C_k|X) = P(C_k) * \frac{P(X|C_K)}{P(X)} \tag{3.4}$$

The above equation is defined as the probability of a particular event $\mathrm{C_k}$ which is occurring for feature vector x. To estimate the value for $\mathrm{P(C_k|x)}$ we first need to compute $\mathrm{P(x|\ C_k)}$. To find this we need the following formula:

$$P(X|c_k) = \prod_{j=1}^{d} P(X_j|c_K) \tag{3.5}$$

We assume that a particular value of $\mathrm{x_j}$ is independent of the occurrence of all other $\mathrm{x_j}$. Then the final formula which forms the x feature vector for a particular event $\mathrm{C_k}$ is given below:

$$P(C_k|X) = P(C_k) * \frac{\prod_{j=1}^{d} P(X_j|C_K)}{P(X)} \tag{3.6}$$

This formula is mainly used for the classification and to minimize the percentage error we should keep the value of $\mathrm{P(C_k|x)}$ higher.

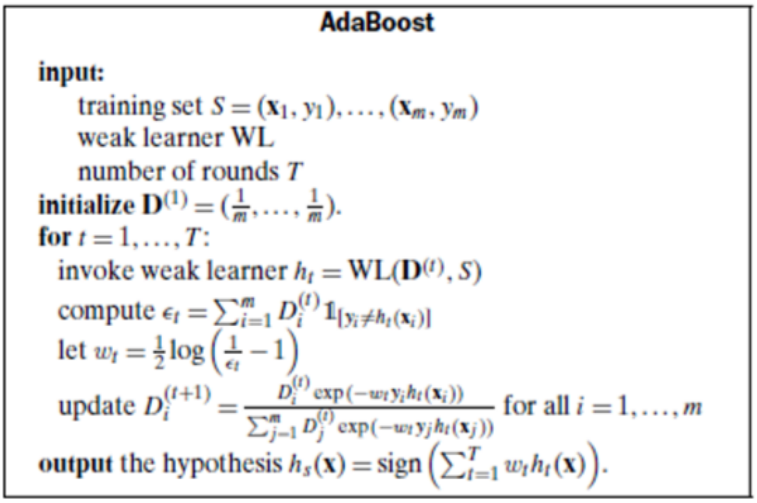
A simple probabilistic model which is based on the Bayes rule with a strong conditional independence assumption is known as a Naive Bayes classifier. The Naive Bayes model involves a simplifying conditional independence assumption. That is, given a class (positive or negative), the words are conditionally independent of each other. Due to this simplifying assumption the model is termed as "naive". It makes fast classification algorithm applicable for the problem without being affected by the assumption in the accuracy of text classification. A simple Naive Bayes classifier can be enhanced to match the classification accuracy of these more complicated models for sentiment analysis.

The advantages of using Naive Bayes as our classifier are:

- Naive Bayes classifiers due to their conditional independence assumptions are extremely fast to train and can scale over large datasets.
- They are robust to noise and less prone to over-fitting.
- Ease of implementation is also a major advantage of Naive Bayes.

3.4 Adaboost Algorithm

AdaBoost also known as "Adaptive Boosting", is the first practical boosting algorithm which were proposed by Freund and Schapire in 1996. It focuses on classification problems and the main objective is to transform a set of weak classifiers into a strong one.



```
AdaBoost
input:
    training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
    weak learner WL
    number of rounds $T$
initialize $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$.
for $t = 1, \ldots, T$:
  invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$
  compute $\epsilon_t = \sum_{i=1}^{m} D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$
  let $w_t = \frac{1}{2}\log\left(\frac{1}{\epsilon_t} - 1\right)$
  update $D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^{m} D_j^{(t)} \exp(-w_t y_j h_t(\mathbf{x}_j))}$ for all $i = 1, \ldots, m$
output the hypothesis $h_s(\mathbf{x}) = \mathrm{sign}\left(\sum_{t=1}^{T} w_t h_t(\mathbf{x})\right)$.
```

Figure 3.2: *Adaboost Algorithm Working Process*

The final equation for classification can be represented as

$$f(x) = sign(\sum_{m=1}^{m} \Theta_m f_m(x)) \tag{3.7}$$

where $\mathrm{f_m}$ stands for the $\mathrm{m_{th}}$ weak classifier and θ_m is the corresponding weight. It is exactly the weighted combination of M weak classifiers.

The main mechanism of adaboost is quite simple which can be represented for a given data set containing n point as such- $\mathrm{x}_i \epsilon R^d$; $y_i \epsilon \{-1, 1\}$
Here -1 denotes the negative class while 1 represents the positive one. Initialize the weight for each data point as:

$$w(x_i, y_i) = \frac{1}{n} \tag{3.8}$$

Here, i=1,2,3.....n. Boosting algorithm increases the accuracy of weak learners [1]. A weak learner uses a simple assumption to bring out a hypothesis that comes from an easily learnable hypothesis class and performs at least slightly better than a random guess. If each weak learner is properly implemented, then Boosting aggregates the weak hypothesis to provide a better predictor which will perform well on hard to learn problems.

The AdaBoost algorithm outputs a "strong" function that is a weighted sum of weak classifiers. The algorithm follows an iterative process where in each iteration the algorithm focuses on the samples where the previous hypothesis gave incorrect answer. The weak learner is returns a weak function whose error is ϵ_{t} such that-

$$\epsilon_t = L_{D_{\mathrm{t}}}(h_t) = \sum_{i=1}^{m} D_i^t[h_i(x_i \neq y_i)] \tag{3.9}$$

Then a specific classifier is assigned a weight for ht as follows: So, the weight given to that weak classifier is inversely proportional to the error of that weak classifier.

3.5 Gradient Boosting Algorithm

Gradient Boosting is an advanced version of AdaBoost which trains many models in a gradual, additive and sequential manner to find out the limitations of weak learners. They find the limitations by taking loss function into account. The loss function is calculated by using gradients. The equation for loss function is:

$$y = ax + b + e \tag{3.10}$$

here, e needs a special mention as it is the error term. The loss function is a way of measuring indicating how effective model's coefficients are at fitting the underlying data. For example, in order to predict the sales prices by using a regression, then the loss function can be based on the error between true and predicted house prices. The main advantage of using gradient boosting is it allows to optimize user specified cost function. It mainly converts weak learners into strong learners.

3.6 Random Forest

Random forest is a decision tree based non-linear technique [1]. Here the word random means we need to select the data randomly this method is called bootstrapping. Usually decision tree has a problem of over-fitting so to overcome this problem random forest came into help. An advantage of using Random Forest is that it can be used for both classification and regression [6]. For our purposes, we applied Random Forest for classification. The algorithm works as follows:

I. At first subset from the entire data-set is selected as in bootstrapping is done. This subset is used to make a decision tree. In this way multiple decision trees are built from multiple subsets.

II. Next, by constructing different subsets we get different decision trees and each tree gives a result and the final classification is done by a method called bagging. In order to classify, bagging choose that class which gets majority votes according to the classification given by each of the decision trees.

We can then easily estimate the error in our results in the following manner:
I. The data sample which were left out from the original dataset during bootstrapping are called "Out of bag" (OOB). Then this sample is being predicted by using the decision trees previously constructed out of bootstrapped sample.
II. We then sum the out of bag sample predictions and determine the error rate. It is called the OOB estimation of error rate. There is also no need for cross-validation or separate test set to get an accurate estimate of the test set error in random forests.

Given enough trees have been grown, the OOB estimate of error rate is quite accurate.

3.7 Logistic Regression

Logistic regression is a supervised classification algorithm. For a classification problem, the output (target variable) or the class y, can take only discrete values for a given set of inputs(features), X. Logistic regression can be represented as a sigmoid function which is basically an S-shaped curve that can take any real-valued number and map it into a range between 0 and 1.

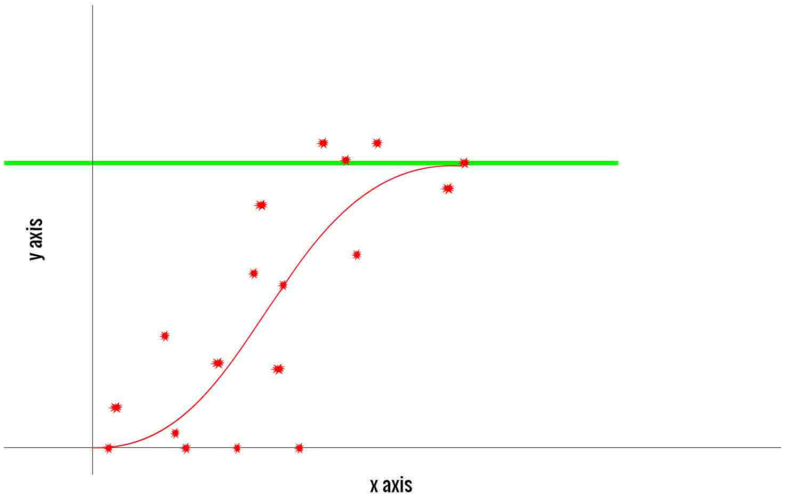


Figure 3.3: *Sigmoid Function*

Input values (x) are combined linearly using weights or coefficient values represented as Beta) to predict an output value (y). The logistic regression equation is:

$$y = e^{b0+b1*x}/1 + e^{b0+b1*x} \tag{3.11}$$

Where y is the predicted output, b0 is the bias or intercept term and b1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data.

The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b's).

Chapter 4

Methodology

In this chapter, we have discussed about how we used quantopian platform to make our Custom-Factor and using the best possible machine learning algorithms we tried to predict the stock price.

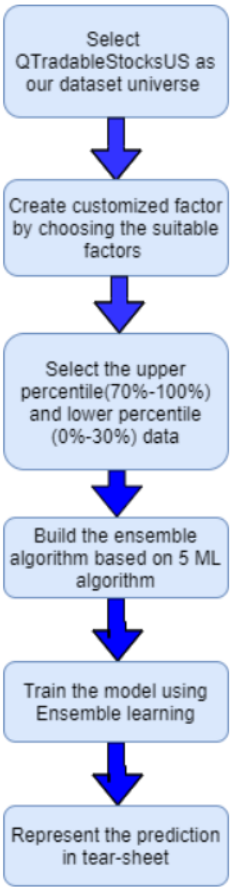

Select QTradableStocksUS as our dataset universe
Create customized factor by choosing the suitable factors
Select the upper percentile(70%-100%) and lower percentile (0%-30%) data
Build the ensemble algorithm based on 5 ML algorithm
Train the model using Ensemble learning
Represent the prediction in tear-sheet

Figure 4.1: *Workflow of Project*

One of the main advantage of quantopian is that it provides US stock market data from 2002 to till date. Total 39 factors are provided by quantopian. Though all of the factors cannot be incorporated in our algorithms because some of the factors can give positive result and negative result. So basically quantopian helps us to select the best factors among the 39 factors which get best fit in our algorithms and as a result helps us to make our customized factor. As some of the factor gives positive outcomes and some gives negative in our algorithm. Moreover, it provides us the option to implement our own algorithm in the live trading environment.

In the quantopian, platform we implemented our own strategy incorporating several machine learning algorithms into it. We began by collecting the data of 1500 of the top stocks in the market using the QTradableStocksUS() function provided by Quantopian. Next, we needed to import the factors that would go into our algorithm. We also had to make our own custom factor which will be described in later part. Some of the factors are Asset Growth 3M, Asset to Equity Ratio, Mean Reversion 1M, EBIT to Assets, EBITDA Yield, Moneyflow Volume 5D, Price Momentum 3M, Return on Invest Capital and more. These are generally considered among the best features to use of our algorithms, but we narrowed down our list through trial and error.

We defined our universe to be the US stock market, and performed a preliminary preprocessing of data to remove any NaN values in our datasets. From the dataset provided we chose only the top 70% of stocks which had the highest increase, and the lowest 30% of stocks which dropped the most to run our algorithms on. The upper 70 percent were all labelled as ‘1’ and the lower 30% were labelled as ‘-1’ and the rest were labelled as’0’. We will mainly focus on the ’1’ and’-1’ part which will eventually tell us if we should go long or short. Next we trained our algorithm based on this criteria and selection. We were trying to predict the outcome of n+5 days where n is considered as the current day, given today’s opening price and our list of factors.

We also generated our very own factor that is the custom factor using a combination of the factors which were chosen through trial and error. With the help of these customized factor, AdaBoost, Random Forest, Logistic Regression, Gradient Boosting and Naive Bayes perform the classification. The column we named ‘EnsembleAlgo’, was the average of the five algorithms which was fed into the algorithms when we were trying to make our prediction. With the help of the pipeline we are merging all the five algorithms to get our final result. We ran two trials using our model and two sets of training data. One was the time series data of the daily opening prices of our chosen stocks from ‘2018-01-01’ to ‘2019-03-01’ and the other was from ‘2015-01-01’ to ‘2016-03-01’.

After performing these, we got to realize that due to the volatility of the market just feeding all the factors into the ML algorithm will not give a very consistent result overall because a given factor can have both a positive or a negative effect on the prediction itself at different given time period with respect to the market. To overcome this problem, we had to implement feature reduction dynamically through trial and error.

Chapter 5

Data Processing and Implementations

In this chapter we discussed about how we processed the data and how the customized factors were generated.

5.1 Primary and Secondary Factors

We start with primary data as the daily (1 day) open, close, high_price, low_price, and volume of each stock in our tradeable universe. Previously Q500US and Q1500US were used but nowadays QTradeableStocksUS is in use which is a better tradable universe, bigger in size and includes both Q500US and Q1500US. Moreover, QTradeableStocksUS provides a better environment for our dataset.

Since we are working with ensemble learning we need to use pipeline to merge all the algorithm and run it in quantopian by the help of run_pipeline. USEquityPricing is our main dataset and we will be working with the latest values from it. By the help of morningstar we can extract all the financial statement, sheets and also ratios like balance sheet, cash flow statement, income statement, operation ratios, earnings report, valuation which acts as the primary factors. We are importing some built-in factors like simple moving average, AverageDollarVolume, Returns etc which helps us to customize the factor. Talib library helps us to collect financial analysis function. Numpy performs different mathematical operations, panda manipulates the dataset. Alphalen is used to measure the effectiveness of how perfectly our customized factor works. To manage performance and risk analysis we are using pyfolio library. Lastly, we are using scipy library to optimize our algorithm to get an efficient result.

As we have already extracted our primary data with the help of Morningstar so this primary data will help us to create our own factor. In quantopian though there are multiple factors available which are stored in all factor dictionary but we are not using all of them instead of that we are customizing our very own factor by filtering out from the available factors using the make_factor method. This method consists of the following factors:

Asset Growth 3M: It measures the overall growth of a stock in a three-month period. Here we are considering 22 days as a month.

$$AssetGrowth = \frac{(Asset\,value\,Prior - Asset\,Value\,Current) * 100}{Asset\,value\,Prior} \tag{5.1}$$

Asset to Equity Ratio: It is a ratio which shows the relationship between the total assets of the firm to the portion owned by shareholders. A higher ratio indicates that the firm is in huge debt.

EBIT to Asset: It represents a firm's profitability before payment of any interest and taxes. In other word it is also termed as operating earnings and operating profit.

$$EBIT = Net\,income + Interest + Taxes \tag{5.2}$$

EBITDA Yield: This value is calculated before firm distribute its earning among interest, taxes, depreciation and amortization to determine the firms profitability

Return on Invetsed Capital:In short (ROIC),which is used to examine how efficiently a firm use its capital to earn profit.This ratio gives an idea of how effectively a company is using its money to generate returns.

$$ROIC = Net\,operating\,profit\,after\,tax/Invested\,Capital \tag{5.3}$$

Mean Reversion 1M: In finance Mean reversion is considered as a theory that proposes that asset prices and historical returns eventually will return as in revert to the long-run mean or average level of the entire dataset. Interest rates and the price-earnings (P/E) ratio of a company are taken into consideration other than Percentage returns and prices[2].

$$dP = \eta P(M - P)dt + \sigma Pdz \tag{5.4}$$

Here, M is the long-run mean price which the prices tend to revert
and h is the speed of reversion.

Price Momentum 3M: It refers to how the price of a stock changes over a period of time which helps investors to trade stocks as a stock price can be rising as in bullish or can be falling as in bearish.

$$Momentum = Latest\,priceClosing\,price * Number\,of\,days \tag{5.5}$$

Vol 3M: It refers to the daily average of the cumulative trading volume over the time period of the the last three months.

Moneyflow Volume 5D: This indicator shows the time and price of stock during the purchase. It is positive when more security was purchased in uptick time compared to the downtick time.

39 Week Returns: The total money earned or lost during a transaction over a year or the ratio of profit and investment is known as returns.

$$RateOfReturns = [\frac{(Current\,Value - Initial\,Value)}{Initial\,Value}] * 100 \tag{5.6}$$

The shift_mask() method is used to process the data as in to divide our data into upper_percentile and lower_percentile. The variable upper_percentile stores the upper 30% of the data as in 70% to 100%. Whereas, the lower percentile stores the lower 30% of the data as in 0% to 30%. To generate more accurate result, we are using upper percentile and lower percentile as these data are extreme data which means it has a lot of fluctuations.

We are considering 3 different time frames which are 1 day, 5 days, 22 days to predict the future returns. The roll method is used to shift the data according to the time frame. If the given time frame is 5 then day 6 data is stored at day 1. Thus, all the data are shifted in this manner. After removing all the NAN values from both the upper percentile and lower percentile we get two variables upper and lower. Values those are greater than upper will be assigned to upper mask and values those are lesser than lower, will be assigned to lower mask. After that in the mask variable we will combine both upper and lower. Then we created a new column which is more like a binary array acting as our output column. After running our very own ensemble learning model considering customized factor the result that is generated is kept in that binary output column. After the prediction, if the result is greater than the value '0.5' we will consider it as 1 which means it is telling us to go 'Long'. If the prediction gives us a result less than 0.5 then we will put -1 in the output binary column which will tell us to go for 'Short'.

5.2 Application of Algorithms

Initially we worked with various algorithms such as SVM, RNN, Naive Bayes, Random Forest, Logistic Regression, Adaboost and Gradient Boosting. We were unable to use some of the algorithm in our classification as those could not provide the expected results. At first we try to implement SVM algorithm considering the factors but it gave us Time Limit Exceed (TLE) error. As we already know, the nature of the stock market is not consistent. SVM algorithm was ineffective for such dynamicity in quantopian platform. Again quantopian does not support RNN algorithm. As a result, we could not use the RNN algorithm as well.

Moreover, Logistic regression requires less time to perform the operation whereas Adaboost algorithm is more dominant than other algorithms in terms of performance. After considering time, performance, platform and factors we chose Logistic regression, Naive Bayes, Random Forest, Adaboost and Gradient Boosting which will be best for ensemble learning. Finally, the classification is mainly done through the average of the results provided by each of the five algorithms. So based on this classification model we do the prediction of the USA stock market.

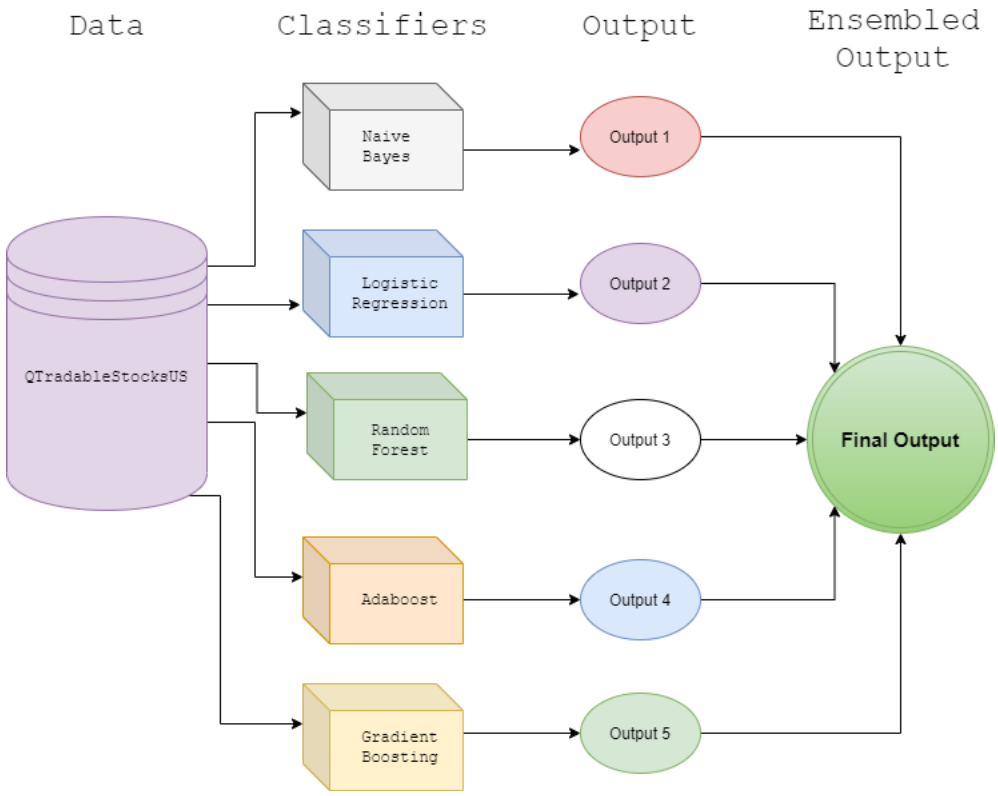


Figure 5.1: *System structure of the Model*

5.3 Factor Graphs

1. Mean Period Wise Return by Quantile: The positive y-axis of the graph represents 'long'. Whereas, the negative y-axis defines 'short'. We are considering five quantiles and each quantile is divided into three parts as in 1D, 5D and 22D. By observing the figure 5.2 we can say 5D trading gives good returns for both long and short.22D iS not giving good returns in comparison to 1D and 5D trading.

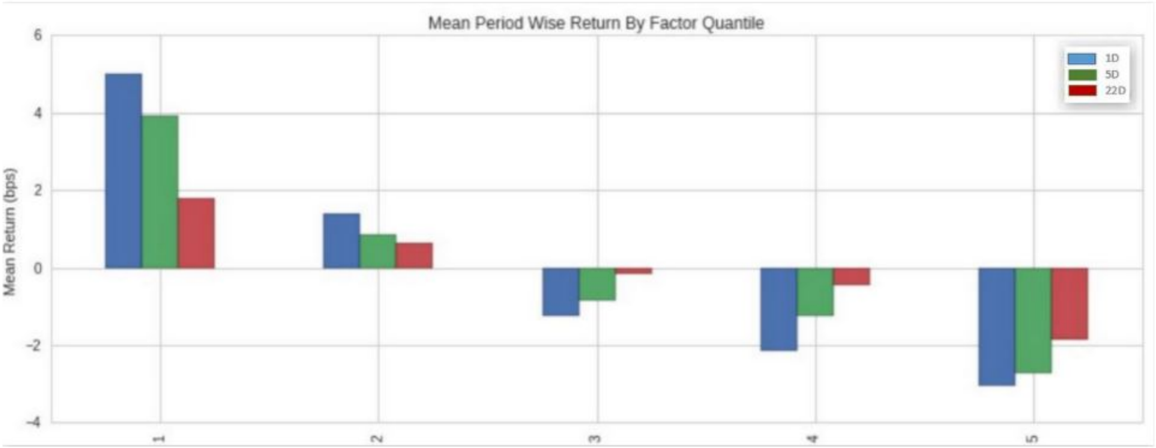


Figure 5.2: *Mean period was Return by factor Quantile*

2. Factor Weighted Long Short Portfolio Cumulative Return: As per quantile deceleration figure 5.3 shows different positions of the portfolio during 1D, 5D AND 22D.

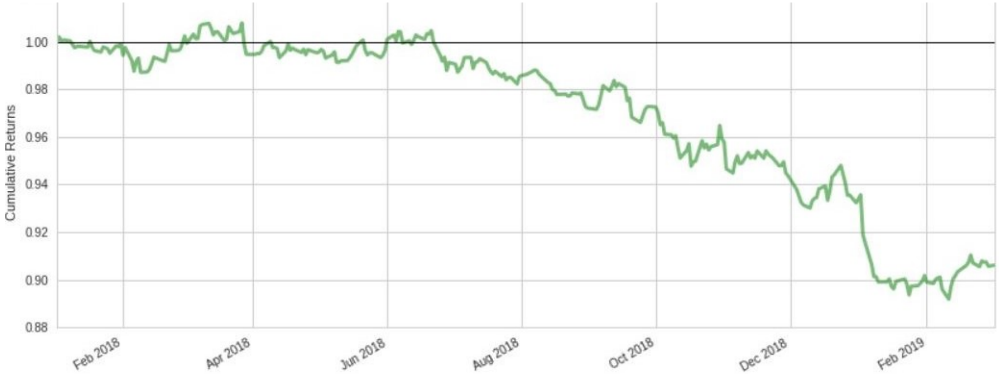


Figure 5.3: *Factor Weighted Long Short Portfolio Cumulative Return 1D*

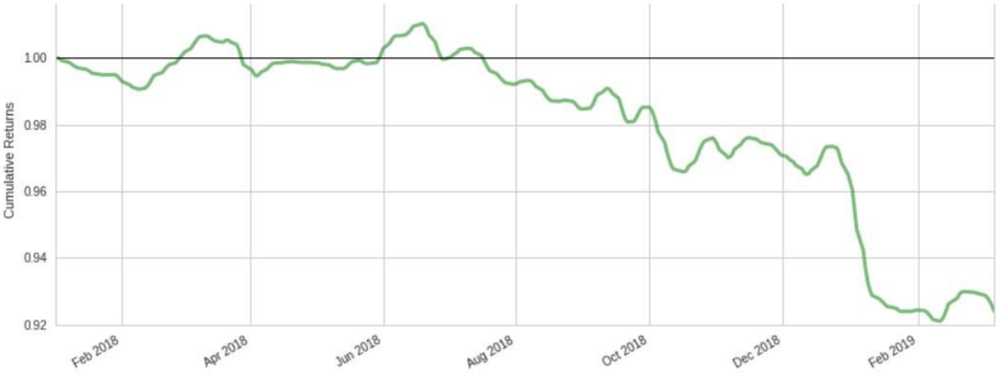


Figure 5.4: *Factor Weighted Long Short Portfolio Cumulative Return 5D*

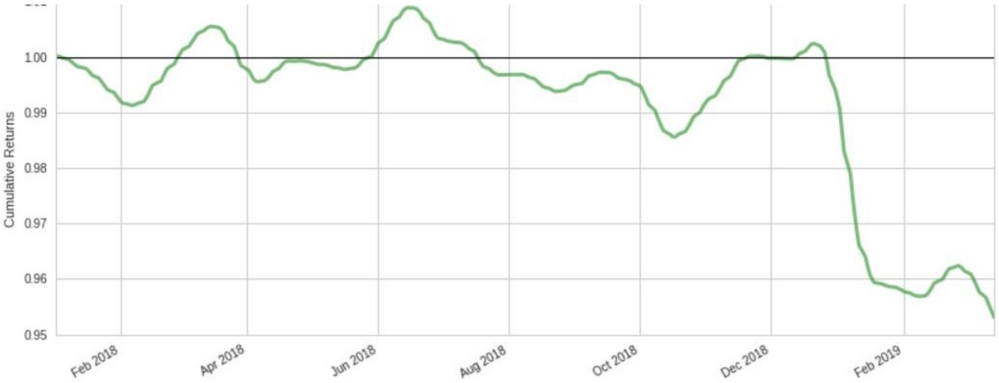


Figure 5.5: *Factor Weighted Long Short Portfolio Cumulative Return 22D*

3. Period Wise Return by Factor Quantile: The violin shaped graphs in figure 5.6 represents the comparison between the summary statistics of range of quantiles. It also provides where the returns are concentrated for each time period.

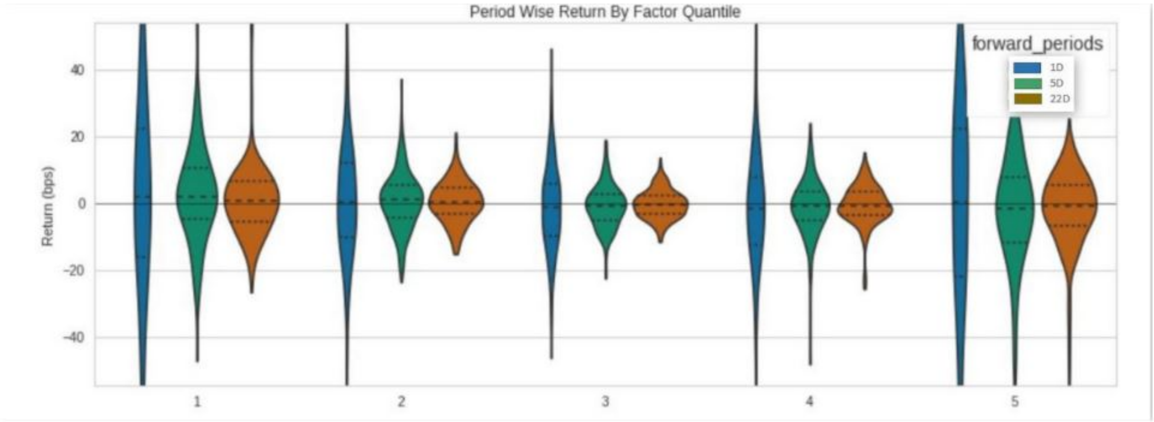


Figure 5.6: *Period Wise Return by Factor Quantile*

4. Cumulative Returns by Quantile: The main purpose of the curves given in figure 5.7, 5.8 and 5.9 is to observe the separation between the quartiles. The graph represented that the more distance and the less overlapping there is between the quartiles the more it performs better. As we proceed with time, the graph showed us that 5.9 gives better result comparing to 5.7 and 5.8 as there were less overlapping and more distance between the quartiles.

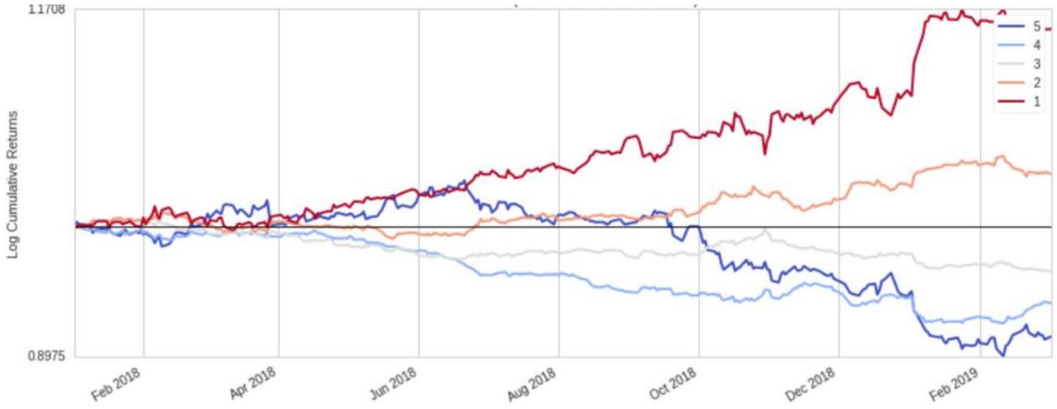


Figure 5.7: *Cumulative Returns by Quantile 1D*

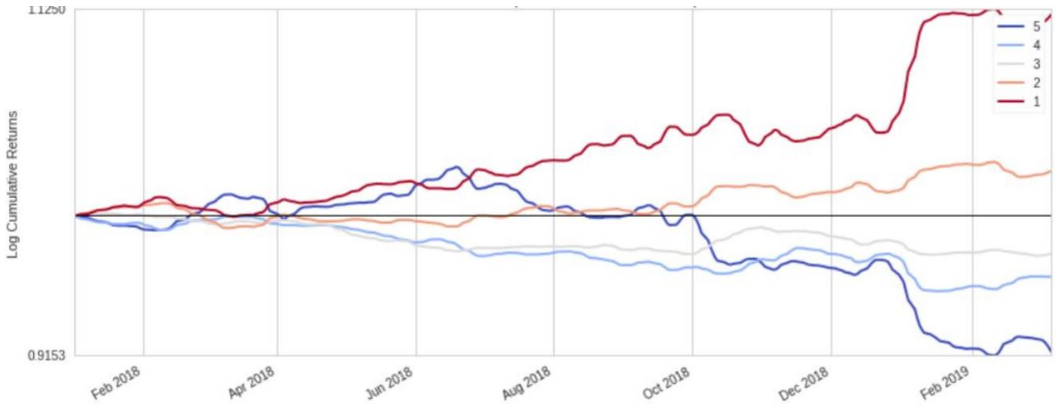


Figure 5.8: *Cumulative Returns by Quantile 5D*

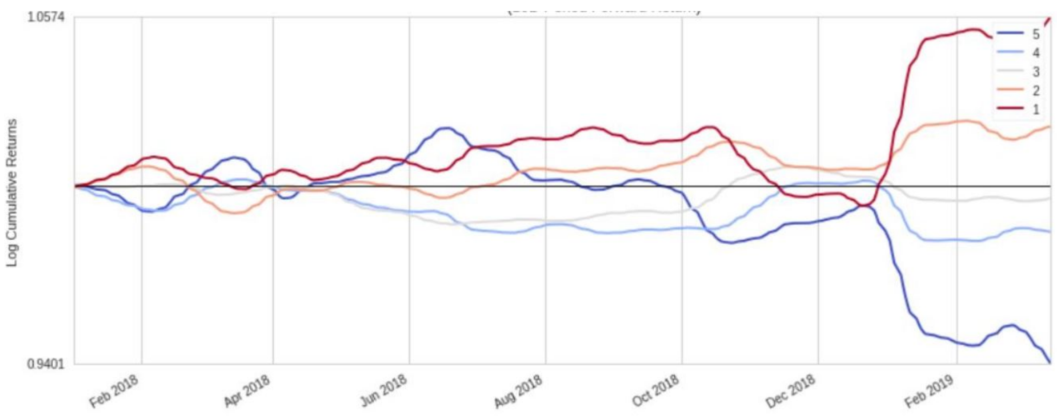


Figure 5.9: Cumulative Returns by Quantile 22D

5. Top Minus Bottom Quantile Mean: To smoothen out the result for a particular trading period the graph in figure 5.10 determines the average of the difference between top quantile and bottom quantile. The more positive the graph is, the more return we get during that particular trading period.

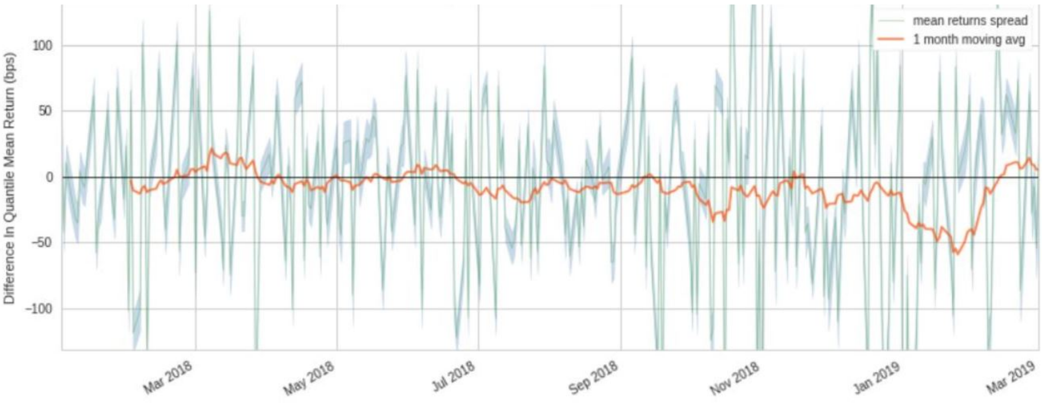


Figure 5.10: *Top Minus Bottom Quantile Mean for 1D*

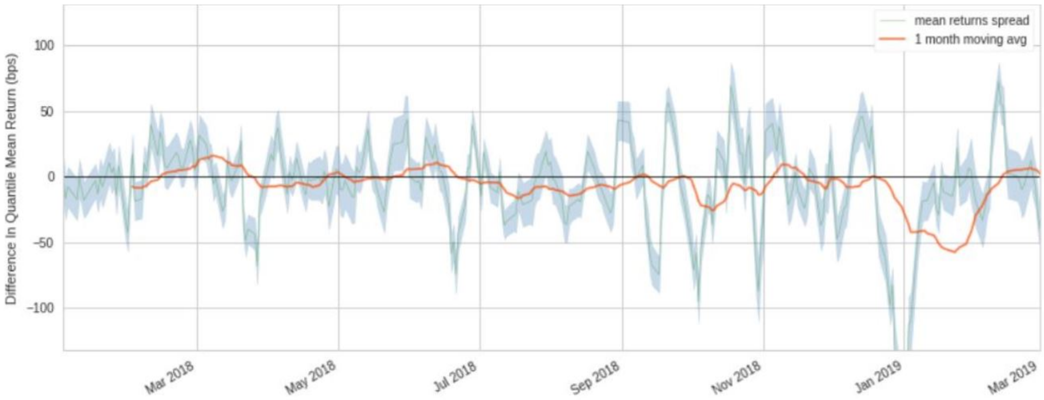


Figure 5.11: *Top Minus Bottom Quantile Mean for 5D*

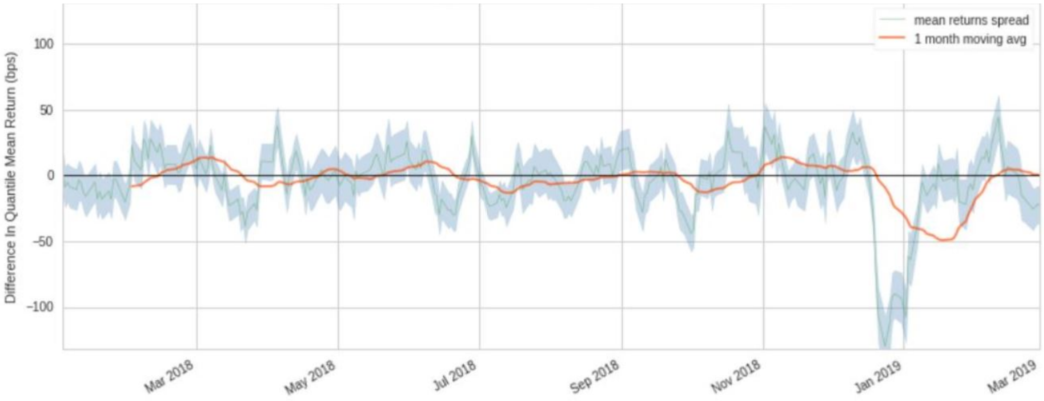


Figure 5.12: *Top Minus Bottom Quantile Mean for 22D*

6. Normal distribution Quantile: Normal distribution is the most commonly assumed form of distribution in technical stock market analysis and other forms of statistical analysis. The normal distribution has two parameters: the standard deviation and the mean [24]. From the normal distribution graph we can understand how the data points are distributed and standard deviation indicates the difference between the observed data point and mean value. Usually the less the difference is the better the prediction. From the Figure 6.1, 6.2, 6.3 it shows that 1D normal distribution quantile is providing better outcomes than 5D and 22D.

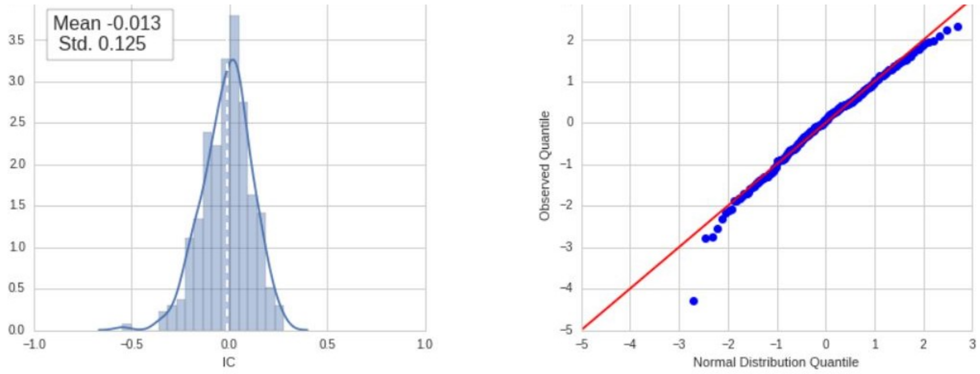


Figure 5.13: *Normal distribution Quantile for 1D*

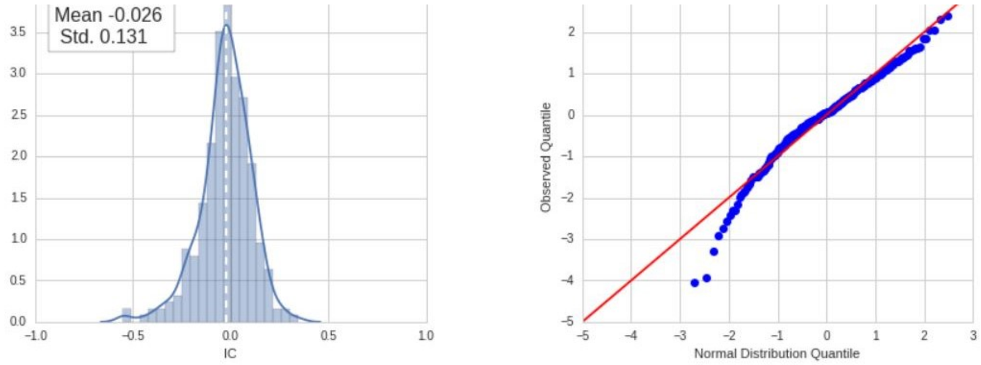


Figure 5.14: *Normal distribution Quantile for 5D*

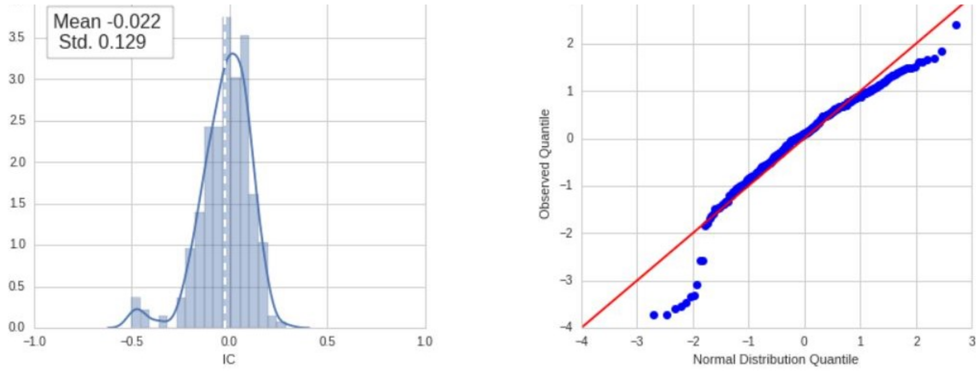


Figure 5.15: *Normal distribution Quantile for 22D*

Chapter 6

Results

In this chapter we have discussed the result we achieved through our ensemble model which we simulated from 05-01-2010 to 31-12-2018 and made comparison between the result and factor graphs. Also we have divided the result into three parts: daily(1D), weekly(5D), monthly(22D) and compared among these three to determine which one provides better returns.

6.1 Returns

Total Return: Total return is the rate of return achieved from an investment for a particular time period which includes interest, capital gains and dividends[23]. After running our ensemble model we got variation in total returns for different time-span (1D, 5D, 22D).

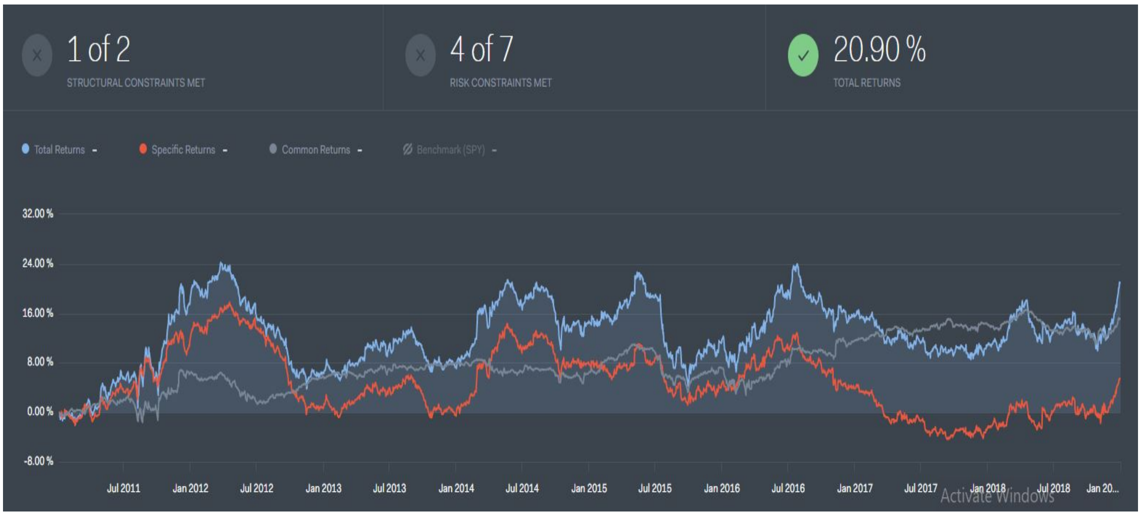
- Total return from 1D trading: 20.90%
- Total return from 5D trading: 40.58%
- Total return from 22D trading: -4.52%

Common Return: Common returns are the total returns which are associated with the common risk factors modeled by quantopian exposure to market beta, sectors, momentum, mean reversion, volatility, size, and value. Common returns should be less then total returns and it signifies the algorithm is running well.

- Common return from 1D trading: 15.06%
- Common return from 5D trading: 24.53%
- Common return from 22D trading: 7.06%

Specific Return: Quantopian defines the returns which are not correlated to any of the known risk factors like value, size, momentum etc. is known as specific return.

- Specific return from 1D trading: 5.42%
- Specific return from 5D trading: 13.18%
- Specific return from 22D trading: -11.08%

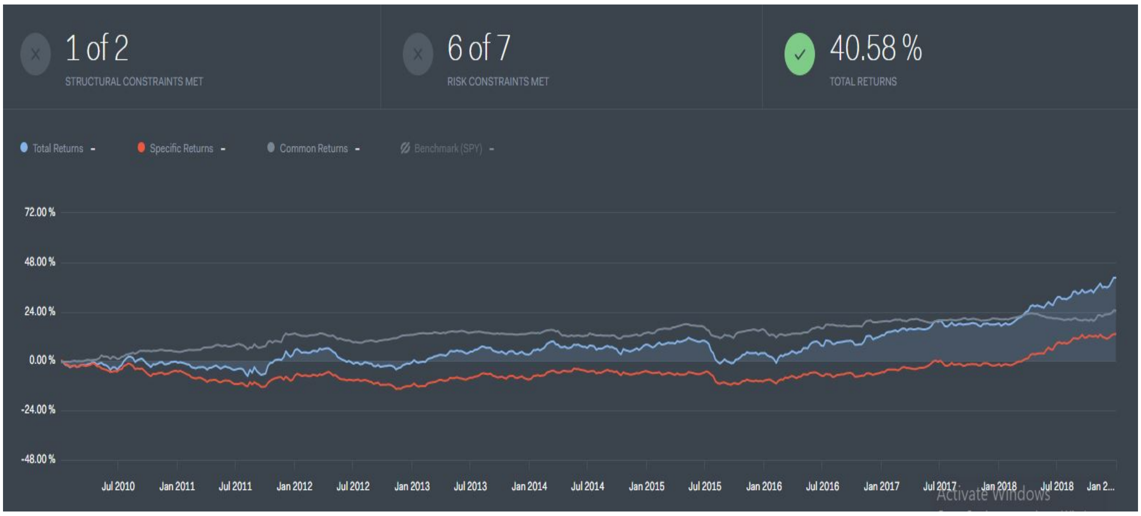


☒ 1 of 2 STRUCTURAL CONSTRAINTS MET

☒ 4 of 7 RISK CONSTRAINTS MET

✓ 20.90 % TOTAL RETURNS

Figure 6.1: *Returns for 1D*

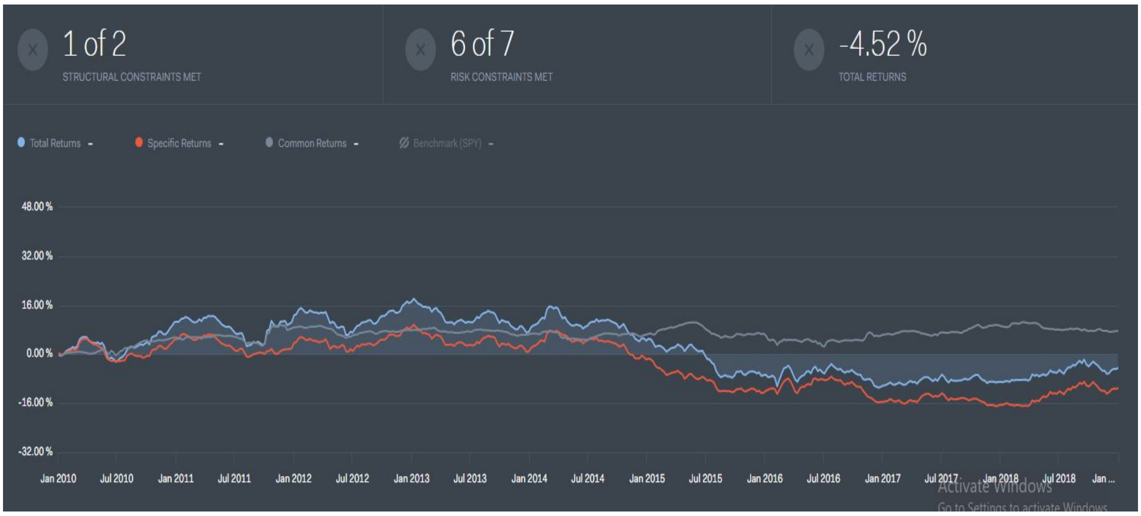


☒ 1 of 2 STRUCTURAL CONSTRAINTS MET

☒ 6 of 7 RISK CONSTRAINTS MET

✓ 40.58 % TOTAL RETURNS

Figure 6.2: *Returns for 5D*



☒ 1 of 2 STRUCTURAL CONSTRAINTS MET

☒ 6 of 7 RISK CONSTRAINTS MET

☒ -4.52 % TOTAL RETURNS

Figure 6.3: *Returns for 22D*

Sharp Ratio: It is the measurement of performance from an investment after the adjustment of its risk.

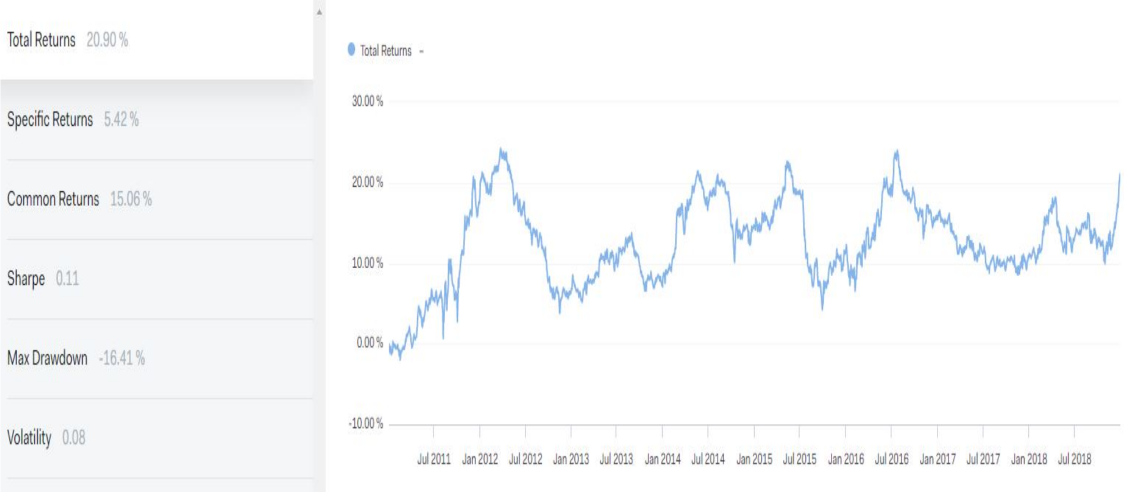
- Sharp Ratio from 1D trading: 0.11%
- Sharp Ratio from 5D trading: 0.54%
- Sharp Ratio from 22D trading: 0.11%

Max Drawdown: The maximum loss that is observed from the observed maximum point to the observed minimum point of the graph is known as max drawdown.

- Max Drawdown from 1D trading: -16.41%
- Max Drawdown from 5D trading: -12.55%
- Max Drawdown from 22D trading: -25.20%

Volatility: It measures the amount of risk.

- Volatility from 1D trading: 0.08%
- Volatility from 5D trading: 0.07%
- Volatility from 22D trading: 0.07%



Total Returns 20.90 %

Specific Returns 5.42 %

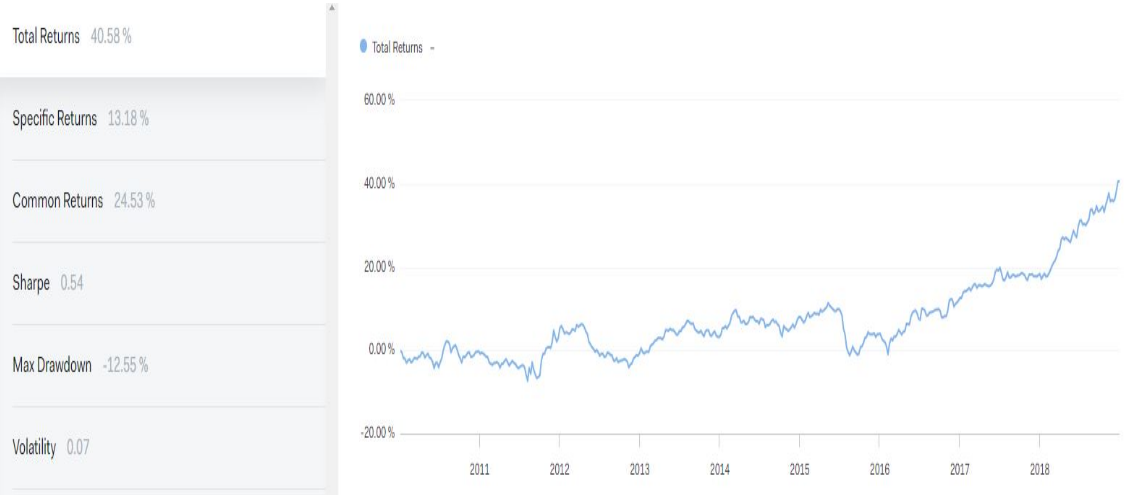
Common Returns 15.06 %

Sharpe 0.11

Max Drawdown -16.41 %

Volatility 0.08

Figure 6.4: *Total Performance for 1D*



Total Returns 40.58 %

Specific Returns 13.18 %

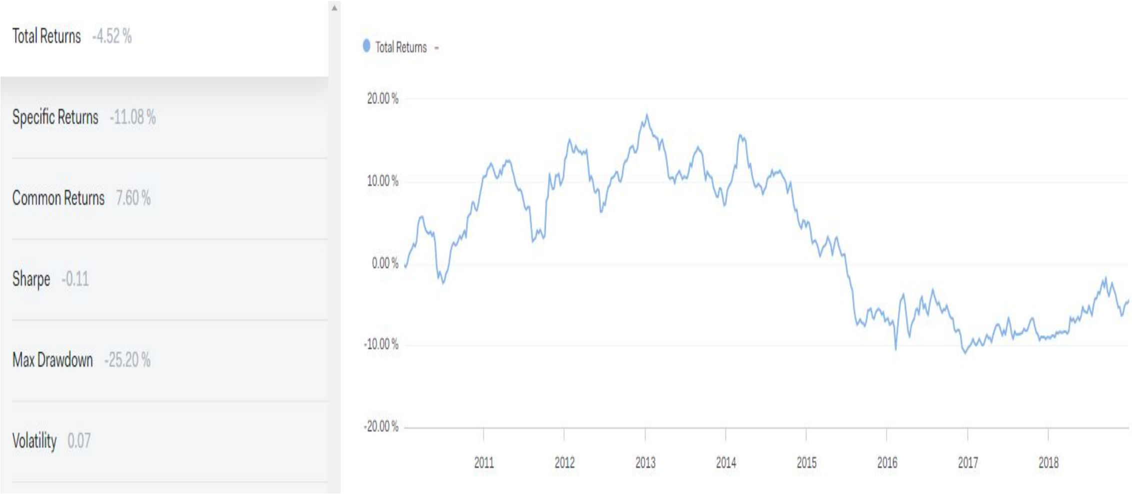
Common Returns 24.53 %

Sharpe 0.54

Max Drawdown -12.55 %

Volatility 0.07

Figure 6.5: *Total Performance for 5D*



Total Returns -4.52 %

Specific Returns -11.08 %

Common Returns 7.60 %

Sharpe -0.11

Max Drawdown -25.20 %

Volatility 0.07

Figure 6.6: *Total Performance for 22D*

6.2 Comparison On performance

From the below table, we came to a conclusion that the outcome generated during 5D trading is better in comparison with the trading of 1D and 22D time span. But previously when we generated tear-sheet from the factors we found that 1D gave better results than 5D and 22D. 5D trading performance deteriorates a little bit than 1D but better than 22D. The performance of factor's tear-sheet was based on the time frame of one year only. When we worked with our ensemble algorithm we considered the time frame to be ten years. While running our ensemble model for 10 years at first we saw that the daily returns are pretty good for 2 or 3 years but then the fluctuations started. On the other hand, the weekly returns graph was in constant increase which performed better than daily return at the end. Though the daily return was performing well than the weekly return at the beginning but with time due to the constant increase weekly return outperformed the daily return in the long run.
The output of this defined that the trading of 5D was better in performance rather than 1D and 22D.

Returns	**1D**	**5D**	**22D**
Total Returns	20.90	40.58	-4.52
Common Returns	15.06	24.53	7.06
Specific Returns	5.42	13.18	-11.08
Sharp Ratio	0.11	0.54	0.11
Max Drawdown	-16.41	-12.55	-25.20
Volatility	0.08	0.07	0.07

Table 6.1: *Outcomes of The Ensemble Model*

Chapter 7

Conclusion

In this chapter, we finish up the paper with a discussion of the outcomes and limitations of results. At long last, we talk about the potential zones in which further research should be possible.

7.1 Conclusion

Through our attempts we came to the conclusion that the US stock market is much volatile and there are too many variables to take into account when trying to predict prices of stocks. This becomes evident as discussed in the previous section, even with the implementations of all the factors we could not use some of the factors later on. We concluded with some factors that are effective during a certain period might be counterproductive during another. Thus, the human factor and the laws of demand and supply that govern the stock market make it too chaotic to predict with sufficient accuracy.

We faced some limitations such as we had limited computing power available to us. Due to our limited resources, we were not able to train our models without exceeding the time limit for some algorithms. We were also unable to implement any kind of neural networks as quantopian would not allow us to import keras for tensor flow.

Apart from Gardient Boosting, Logistic regression, Naive Bayes and Random Forest we also attempted to implement SVM (Support Vector Machine) and RNN as it is commonly used in making stock market predictions, and gives good results. We observed that Adaboost outperformed all the other four algorithms. However, training with only SVM on each dataset takes approximately 18 minutes 33 seconds to train, whereas ensemble algorithm took only 13 minutes 54 seconds to train. Moreover, RNN was not supported by quantopian platform. Thus we run each of the five algorithms and average the outputs provided by the algorithms to find out the final output as in to do the classification. While considering only the factors we found that 1D trading is giving better results when we were working for only one year basis but after running the ensemble model for the time span of 10 years we observed weekly trading gives more accurate returns.

7.2 Future work

As we are working with quantopian it has some advantages like we can access it remotely through internet and also provides US stock market dataset as well as the opportunity of Live preview of data trading. On the contrary there are some limitations of using quantopian. One of the main constraints was it does not allow to use RNN which is one of the significant algorithms. But this algorithm has proved to provide more accurate result to predict stock market previously. So if you could run this ensemble model in another dynamic platform we could have achieved better result without maintaining these constraints.

Moreover, if we could incorporate our own resources we could get more efficient results. Dynamic environment not only can predict the stock market of US but through this we can predict the stock market of other countries by only processing the data. That is why we are willing to work with dynamic environment in the near future. However, in the future if we want least time trading as in hourly and quarterly that cannot be done in quantopian since the least time trading it supports is 1D trading. To improve our existing model we can set an upper threshold for profit and lower threshold for loss. When the predicted price of a particular stock exceeds the upper threshold the stock will be sold automatically on that predicted day and when it falls under lower threshold, that particular stock should be automatically sold right at that moment. In a nut shell, even though quantopian has some limitations but for our proposed model it was able to provide a fair prediction about the stock market.

Bibliography

[1] B. G. Malkiel and E. F. Fama, “Efficient capital markets: A review of theory and empirical work”, *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[2] J. M. Poterba and L. H. Summers, “Mean reversion in stock prices: Evidence and implications”, *Journal of financial economics*, vol. 22, no. 1, pp. 27–59, 1988.

[3] J.-H. Wang and J.-Y. Leu, “Stock market trend prediction using arima-based neural networks”, in *Proceedings of International Conference on Neural Networks (ICNN’96)*, IEEE, vol. 4, 1996, pp. 2160–2165.

[4] V. N. Vapnik, “An overview of statistical learning theory”, *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[5] A. Abraham, B. Nath, and P. K. Mahanti, “Hybrid intelligent systems for stock market analysis”, in *International Conference on Computational Science*, Springer, 2001, pp. 337–345.

[6] A. Liaw, M. Wiener, *et al.*, “Classification and regression by randomforest”, *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[7] A.-S. Chen, M. T. Leung, and H. Daouk, “Application of neural networks to an emerging financial market: Forecasting and trading the taiwan stock index”, *Computers & Operations Research*, vol. 30, no. 6, pp. 901–923, 2003.

[8] X. Tao, H. Renmu, W. Peng, and X. Dongjie, “Input dimension reduction for load forecasting based on support vector machines”, in *2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies. Proceedings*, IEEE, vol. 2, 2004, pp. 510–514.

[9] M. R. Hassan, B. Nath, and M. Kirley, “A fusion model of hmm, ann and ga for stock market forecasting”, *Expert systems with Applications*, vol. 33, no. 1, pp. 171–180, 2007.

[10] S.-H. Hsu, J. P.-A. Hsieh, T.-C. Chih, and K.-C. Hsu, “A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression”, *Expert Systems with Applications*, vol. 36, no. 4, pp. 7947–7951, 2009.

[11] P. Ou and H. Wang, “Prediction of stock market index movement by ten data mining techniques”, *Modern Applied Science*, vol. 3, no. 12, pp. 28–42, 2009.

[12] C.-F. Tsai, Y.-C. Lin, D. C. Yen, and Y.-M. Chen, “Predicting stock returns by classifier ensembles”, *Applied Soft Computing*, vol. 11, no. 2, pp. 2452–2459, 2011.

[13] V. N. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[14] O. Hegazy, O. S. Soliman, and M. A. Salam, "A machine learning model for stock market prediction", *arXiv preprint arXiv:1402.7351*, 2014.

[15] R. Brown, "What is the stock market?", 2015. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/stock-market/.

[16] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques", *Expert systems with applications*, vol. 42, no. 1, pp. 259–268, 2015.

[17] D. Philip, "Why you should never short-sell stocks", 2015. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=2006&q=Why+you+should+never+short-sell+stocks&btnG=.

[18] J. Roy and A. A. R. Nayeem, "Using sentiment analysis & machine learning for security price forecasting", PhD thesis, BRAC University, 2015.

[19] Sanjay, "Shares stock market", vol. 4, 2016.

[20] M. Dunne, "Stock market prediction", *Unpublished thesis). University College Cork. Retrieved December*, vol. 26, 2017.

[21] A. Zheng and J. Jin, "Using ai to make predictions on stock market", Stanford University, Tech. Rep, Tech. Rep., 2017.

[22] Allen, "Types of stocks – common, preferred, hybrid, etc", 2018. [Online]. Available: https://cleartax.in/s/stock-types.

[23] G. Kenton, "Total return", 2018. [Online]. Available: https://www.investopedia.com/terms/t/totalreturn.asp.

[24] S. Chen, "Normal distribution", 2019. [Online]. Available: https://www.investopedia.com/terms/n/normaldistribution.asp.

[25] M. hargrave, "Return on assets", 2019. [Online]. Available: https://www.investopedia.com/terms/r/returnonassets.asp.

[26] F. Jean, "How are share prices set?", 2019. [Online]. Available: https://www.investopedia.com/ask/answers/12/how-are-share-prices-set.asp.

[27] W. kenton, "Return on invested capital – roic definition", 2019. [Online]. Available: https://www.investopedia.com/terms/r/returnoninvestmentcapital.asp.

[28] K. Leslie, "How is a company's stock price and market cap determined?", 2019. [Online]. Available: https://www.investopedia.com/ask/answers/how-companys-stock-price-and-market-cap-determined.

[29] C. Mitchell, "Percentage price oscillator", 2019. [Online]. Available: http://www.investopedia.com/terms/p/ppo.asp.

[30] ——, "Williams %r definition and uses", 2019. [Online]. Available: https://www.investopedia.com/terms/w/williamsr.asp.

[31] G. Chen, Y. Chen, and T. Fushimi, "Application of deep learning to algorithmic trading",

[32] T. Dai, A. Shah, and H. Zhong, “Cs229 project report automated stock trading using machine learning algorithms”,

[33] K. Sharif and M. Abu-Ghazaleh, “Investigating algorithmic stock market trading using ensemble machine learning methods”,