Stock Market Prediction

Student Name: Mark Dunne

Student ID: 111379601

Supervisor: Derek Bridge

Second Reader: Gregory Provan

Declaration of Originality

In signing this declaration, you are confirming, in writing, that the submitted work is entirely your own original work, except where clearly attributed otherwise, and that it has not been submitted partly or wholly for any other educational award.

I hereby declare that:

- This is all my own work, unless clearly indicated otherwise, with full and proper accreditation;
- With respect to my own work: none of it has been submitted at any educational institution contributing in any way towards an educational award;
- With respect to another's work: all text, diagrams, code, or ideas, whether verbatim, paraphrased or otherwise modified or adapted, have been duly attributed to the source in a scholarly manner, whether from books, papers, lecture notes or any other student's work, whether published or unpublished, electronically or in print.

Name: Mark Dunne

Signed:

Date:

Abstract

In this report we analyse existing and new methods of stock market prediction. We take three different approaches at the problem: Fundamental analysis, Technical Analysis, and the application of Machine Learning. We find evidence in support of the weak form of the Efficient Market Hypothesis, that the historic price does not contain useful information but out of sample data may be predictive. We show that Fundamental Analysis and Machine Learning could be used to guide an investor's decisions. We demonstrate a common flaw in Technical Analysis methodology and show that it produces limited useful information. Based on our findings, algorithmic trading programs are developed and simulated using Quantopian.

Contents

1 Introduction **3**
1.1 Project Goals and Scope 3

2 Data and Tools **5**
2.1 Data Used 5
2.1.1 Choosing the Dataset 5
2.1.2 Gathering the Datasets 5
2.1.3 Limitations of the Data 7
2.2 Tools 7

3 Considerations in Approaching the Problem **9**
3.1 Random Walk Hypothesis 9
3.1.1 Qualitative Similarity to Random pattern 9
3.1.2 Quantitative Difference to Random pattern 11
3.2 Efficient Market Hypothesis 12
3.3 Self Defeating Strategies 13
3.4 Conclusions 13

4 Review of Existing Work **14**
4.1 Article 1 - Kara et al. [10] 14
4.2 Article 2 - Shen et al. [19] 16

5 Attacking the Problem - Fundamental Analysis **18**
5.1 Price to Earnings Ratio 18
5.2 Price to Book Ratio 20
5.3 Limitations of Fundamental Analysis 22
5.4 Fundamental Analysis - Conclusion 22

6 Attacking the Problem - Technical Analysis **24**
6.1 Broad Families of Technical Analysis Models 24
6.2 Naive Trading patterns 24
6.3 Moving Average Crossover 26
6.3.1 Evaluating the Moving Average Crossover Model 27
6.4 Additional Technical Analysis Models 29
6.4.1 Evaluating the Indicators 30

6.4.2 Data Preparation . . . 31
6.4.3 Error Estimation . . . 31
6.5 Common Problems with Technical Analysis . . . 32
6.6 Technical Analysis - Conclusion . . . 33

7 Attacking the problem - Machine Learning 34
7.1 Preceding 5 day prices . . . 34
7.1.1 Error Estimation . . . 35
7.1.2 Analysis of Model Failure . . . 36
7.1.3 Preceeding 5 day prices - Conclusion . . . 39
7.2 Related Assets . . . 39
7.2.1 Data . . . 39
7.2.2 Exploration of Feature Utility . . . 40
7.2.3 Modeling . . . 41
7.2.4 Related Assets - Conclusion . . . 43
7.3 Analyst Opinions . . . 43
7.3.1 Data . . . 44
7.3.2 Data Exploration . . . 44
7.3.3 Data Preparation . . . 45
7.3.4 Error Estimation . . . 47
7.3.5 Model Selection . . . 47
7.3.6 Analyst Opinions - Conclusion . . . 48
7.4 Disasters . . . 48
7.4.1 Data Preparation . . . 48
7.4.2 Predictive Value of Disasters . . . 49
7.4.3 Disasters - Conclusion . . . 50

8 Quantopian Trading Simulation 52
8.1 Simulation 1 - Related Assets . . . 52
8.2 Simulation 2 - Analyst Opinions . . . 54

9 Report Conclusion 57

Chapter 1

Introduction

Predicting the Stock Market has been the bane and goal of investors since its existence. Everyday billions of dollars are traded on the exchange, and behind each dollar is an investor hoping to profit in one way or another. Entire companies rise and fall daily based on the behaviour of the market. Should an investor be able to accurately predict market movements, it offers a tantalizing promises of wealth and influence. It is no wonder then that the Stock Market and its associated challenges find their way into the public imagination every time it misbehaves. The 2008 financial crisis was no different, as evidenced by the flood of films and documentaries based on the crash. If there was a common theme among those productions, it was that no one was really sure how the market works or reacts. Perhaps a better understanding of stock market prediction might help in the case of similar events in the future.

1.1 Project Goals and Scope

Despite its prevalence, Stock Market prediction remains a secretive and empirical art. Few people, if any, are willing to share what successful strategies they have. A chief goal of this project is to add to the academic understanding of stock market prediction. The hope is that with a greater understanding of how the market moves, investors will be better equipped to prevent another financial crisis. The project will evaluate some existing strategies from a rigorous scientific perspective and provide a quantitative evaluation of new strategies.

It is important here to define the scope of the project. Although vital to any investor operating in the real world, no attempt is made in this project at portfolio management. Portfolio management is largely an extra step done after an investor has made a prediction on which direction any particular stock will move. The investor may choose to allocate funds across a range of stocks in such a way to minimize his or her risk. For instance, the investor may choose not to invest all of their funds into a single company lest that company takes unexpected turn. A more common approach would be for an investor to

invest across a broad range of stocks based on some criteria he has decided on before.This project will focus exclusively on predicting the daily trend (price movement) of individual stocks. The project will make no attempt at deciding how much money to allocate to each prediction. More so, the project will analyse the accuracies of these predictions.

Additionally, a distinction must be made between the trading algorithms studied in this project and high frequency trading (HFT) algorithms. HFT algorithms make little use of intelligent prediction and instead rely on simply being the fastest algorithm in the market. These algorithms operate on the order of fractions of a second. The algorithms presented in this report will operate on the order of days and will attempt to be truly predictive of the market.

Chapter 2

Data and Tools

2.1 Data Used

2.1.1 Choosing the Dataset

For this project, we chose the Dow Jones and its components as a representative bundle of stocks. The Dow Jones is a large index traded on the New York stock exchange. It is a price-weighted index over 30 component companies. All companies in the index are large publicly traded companies, leaders in each of their own sectors. The index covers a diverse set of sectors featuring companies such as Microsoft, Visa, Boeing, and Walt Disney.

It is important to use a predefined set of companies rather than a custom selected set so that we do leave ourselves open to methodology errors or accusations of fishing expeditions. If we had selected a custom set of companies, it could be argued that the set was tailored specifically to improve our results. Since the aim of the project is to create a predictive model of stock markets in general, this is a claim that we want to avoid.

The Dow Jones was chosen because it is well known, and has a relatively small number of components. The leading index, the S&P 500, has over 500 components, but ultimately analysing 500 companies would be too computationally expensive for the resources at hand. The Dow Jones components provided a good balance between data available and computational feasibility. Although there were only 30 companies, for many of the experiments carried out in this report, datasets many thousands of examples in size were available.

2.1.2 Gathering the Datasets

A primary dataset will be used throughout the project. The dataset will contain the daily percentage change in stock price for all 30 components of the Dow Jones.

Luckily, daily stock price data is easy to come by. Google and Yahoo both operate websites which offer a facility to download CSV files containing a full

daily price history. These are useful for looking at individual companies but cumbersome when accessing large amounts of data across many stocks.

For this reason, Quandl[5] was used to gather the data instead of using Google and Yahoo directly. Quandl is a free to use website that hosts and maintains vast amounts of numerical datasets with a focus specifically on economic datasets. Stock market data which is backed by Google and Yahoo are both avilable through Quandl. Quandl also provides a small python library that is useful for accessing the database programmatically. The library also provides a simple method for calculating the daily percentage change daily in prices. This calculation is defined in equation 2.1 where p_t is the closing price on day d, and δp_d is the resulting percentage change.

$$\delta p_d = \frac{p_d - p_{d-1}}{p_{d-1}} \tag{2.1}$$

To build the primary dataset, a simple program was built to query the Quandl API for each of the 30 Dow Jones components using the transformation outlined in equation 2.1. Before finally saving the data, the gathered data was augmented with an additional column containing the classification of the daily percentage change. This augmentation is defined in equation 2.2 where $trend_d$ is the price movement direction on day d and δp_d as defined in equation 2.1.

$$trend_d = \begin{cases} Loss & \text{if } \delta p_d < 0 \\ Gain & \text{if } \delta p_d > 0 \\ Neutral & \text{if } \delta p_d = 0 \end{cases} \tag{2.2}$$

The final step is shifting all of the δp_d and $trend_d$ data points backwards by one day. The data we have for any day should be used to predict the trend tomorrow, not the same day. By shifting δp_d and $trend_d$ backwards, data gathered in later sections will be paired with the correct dependent variable. For instance, data we gather for a Monday will be matched with, and try to predict, Tuesday's trend.

This dataset was then saved in CSV format for simple retrial as needed throughout the project. This dataset containing the daily trends of companies will serve as the core dataset that will be used in most experiments later in the report. When we want to use the dataset later with the extra data we collect for each experiment, we only need to do a simple database join operation on the company and date. The dataset contains 122,121 rows covering all 30 companies daily since January 1st 2000. Table 2.1 shows an extract of the number of rows from this dataset.

Figure 2.1: Data Extract of Primary Dataset

Date	Symbol	δp_d	trend
2000-01-05	DD	0.04230	Gain
2000-01-05	DIS	0.03748	Gain
2000-01-05	GE	-0.00749	Loss
2000-01-05	GS	-0.04248	Loss
2000-01-05	HD	-0.00097	Loss
2000-01-05	IBM	0.03515	Gain

With the primary dataset prepared, we will combine it as needed with additional datasets to carry out individual experiments.

2.1.3 Limitations of the Data

Although the data gathered in the previous section is certainly a good start, it is admittedly far behind what any serious investor more than likely has access to.

One obvious piece of missing data that is the intraday prices, i.e the prices minute by minute. It is possible that this data could be used to guide and investors decisions on the interday level. However, intraday prices are not as freely available as interday prices and are considered a commodity in themselves. To get hold of such a dataset would incur a large cost, one that is not in the budget of a project such as this. Later in the project we will evaluate a strategy in which this limitation becomes significant.

Another important piece of missing data is the order book. The order book is a record of live buy and sell orders for a particular stock. It consists of the amount of stock each trader is willing to buy or sell, as well as their price. Successful orders are matched off against the order book by the exchange. The price of a stock is usually considered to be half way between the highest buying price and the lowest selling price. It is easy to imagine that the order book contains useful data. For instance, the weighted average of orders might be predictive of the price. However access to this data is extremely costly and far beyond what most casual investors can afford, let alone the budget for this project's.

With no way around these limitations, we use the data provided by Quandl.

2.2 Tools

Python and associated packages

Python was the language of choice for this project. This was an easy decision for the following reasons.

1. Python as a language has an enormous community behind it. Any problems that might be encountered on the way can be easily solved with a trip to Stack Overflow. Python is among the most popular languages on the site which makes it very likely there will be a direct answer to any query [17].

2. Python has an abundance of powerful tools ready for scientific computing. Packages such as Numpy, Pandas, and SciPy are freely available, performant and well documented. Packages such as these can dramatically reduce and simplify the code needed to write a given program. This makes iteration quick.

3. Python as a language is forgiving and allows for programs that look like pseudo code. This is useful when pseudo code given in academic papers needs to be implemented and tested. Using Python, this step is usually reasonably trivial.

However, Python is not without its flaws. The language is dynamically typed and packages are notorious for Duck Typing. This can be frustrating when a package method returns something that, for example, looks like an array rather than being an actual array. Coupled with the fact that standard Python documentation does not explicitly state the return type of a method, this can lead to a lot of trial and error type testing that would not otherwise happen in a strongly typed language. In my view, this is an issue that makes learning to use a new Python package or library more difficult than it otherwise could be.

Chapter 3

Considerations in Approaching the Problem

Throughout the project, there are three ideas that warn us that we might not find a profitable way to predict market trends.

3.1 Random Walk Hypothesis

The random walk hypothesis sets out the bleakest view of the predictability of the stock market. The hypothesis says that the market price of a stock is essentially random. The hypothesis implies that any attempt to predict the stock market will inevitably fail.

The term was popularized by Malkiel [13]. Famously, he demonstrated that he was able to fool a stock market 'expert' into forecasting a fake market. He set up an experiment where he repeatedly tossed a coin. If the coin showed heads, he moved the price of a fictitious stock up, and if it showed tails then he moved it lower. He then took his random stock price chart to a supposed expert in stock forecasting, and asked for a prediction. The expert was fooled and recommended that he buy the stock immediately.

It is important for the purpose of this project to confront the Random Walk Hypothesis. If the market is truly random, there is little point in continuing.

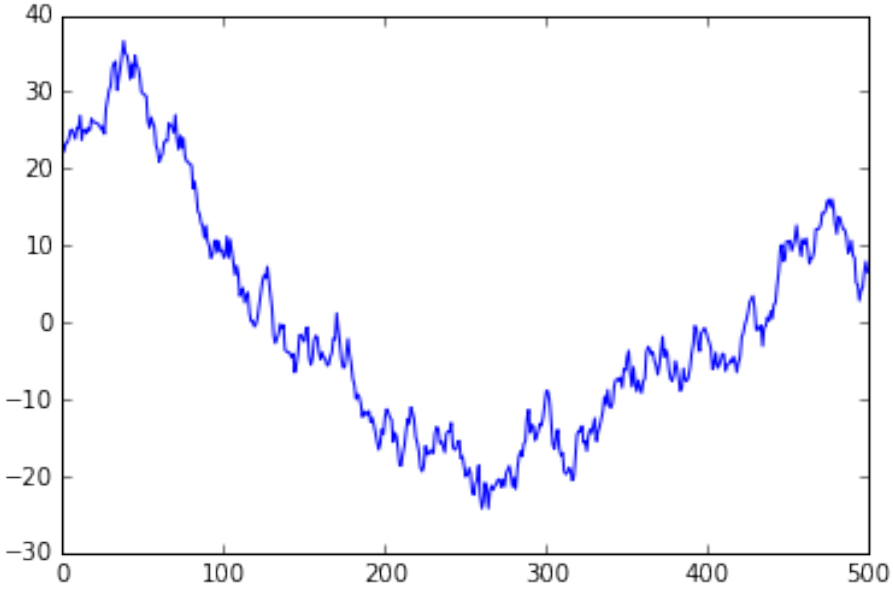
3.1.1 Qualitative Similarity to Random pattern

The stock market can certainly look random to the eye of a casual observer. To demonstrate this, we created a perfectly random process that had striking visual similarity to real stock market data. The random process used to generate this is defined in equation 3.1 where $a_0 = 0$, $0.995 \leq \rho < 0$, q and r are random values taken from a standard normal distribution. b_0 can be initialised at any desired number

$$
\begin{aligned}
a_n &= a_{n-1} * \rho + q_n \\
b_n &= b_{n-1} + \rho * r_n \\
f(n) &= a_n + b_n
\end{aligned} \tag{3.1}
$$

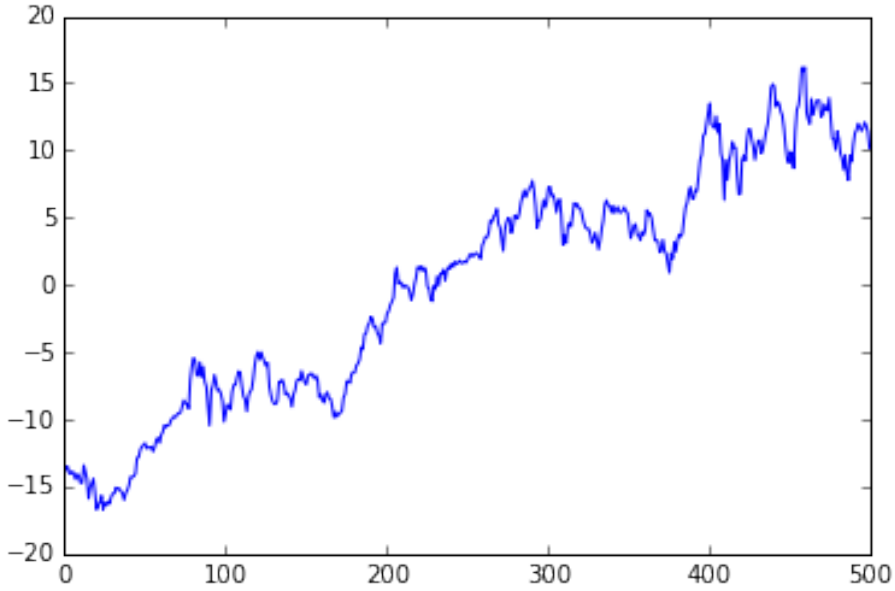
A sequence generated using equation 3.1 is displayed in figure 3.1. The graph plots an example generation of $f(0), f(1), ..., f(500)$.

Figure 3.1: Example of a randomly generated pattern



We then compare this random process to a real piece of market data.

Figure 3.2: Centered APPL stock price, some time after 2010



Presented with both of these diagrams, and without the aid of time scales or actual prices, most people would find it impossible to differentiate the diagrams. Using visual inspection alone, either of these diagrams could just as likely be a real piece of stock market data.

This gives us pause as there is little point in moving forward if the stock market is truly random and there is nothing to predict. However, this does not turn out to be the case.

3.1.2 Quantitative Difference to Random pattern

[todo external vars]

In this section, we will demonstrate that prices, specifically the way they move, are fundamentally different to random data.

Karpio et al. [11] describe an asymmetry between gains and losses on the stock market. Their research looks specifically at indices like the Dow Jones. They describe how "you wait shorter time (on average) for loss of a given value than for gain of the same amount". However, this research was conducted in 2006, before the Great Recession. It is conceivable that the market conducts itself differently since then, and therefore we tried to replicate their findings.

On every day from the year 2000 to 2014, we recorded the value of the Dow Jones index. We then counted the number of days it took for the value to gain or lose 5% of its original value. When it lost 5% of its value, it was put into the red set, when it gained 5% of its original value, it was put into the green set.

Figure 3.3 shows two overlaid histograms detailing how long it took the red and green sets to lose or gain 5% respectively.

Figure 3.3: Gain-Loss Asymmetry on the Dow Jones

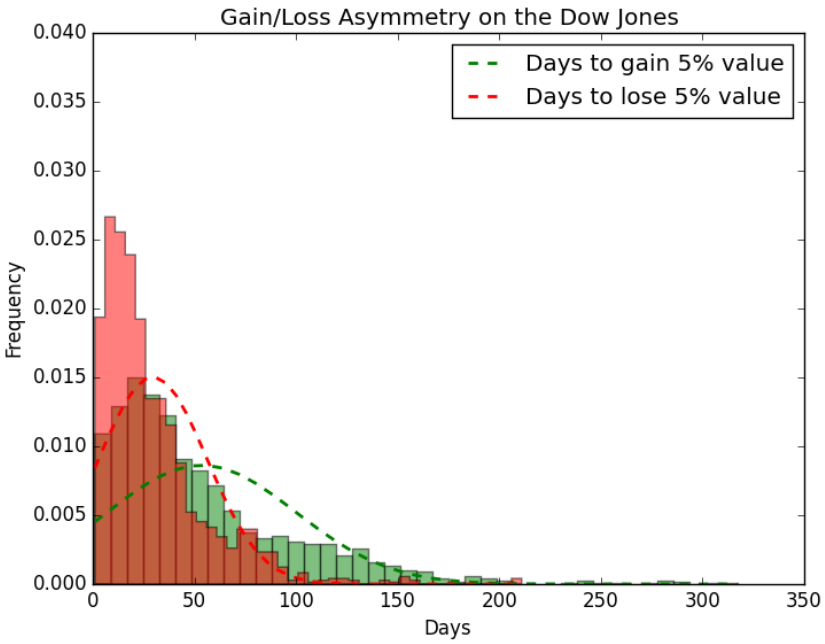


Figure 3.3 shows that the green set, the instances that gained 5% of their original value, took on average longer to reach that point than the red set, the instances that lost 5% of their original value. This indicates that the market generally creeps upwards but is prone to sudden drops downwards, and supports the findings of Karpio et al. [11].

This demonstrates that the stock market is fundamentally different to random data. This gives us hope for the remainder of the project. If the market price is not random, then it might be worth investigating and trying to predict.

3.2 Efficient Market Hypothesis

Another concept to keep in mind while working on the project was the Efficient Market Hypothesis. Informally, the Efficient Market Hypothesis says that the market is efficient at finding the correct price for the stock market.

It comes in three flavours. However it is still a matter of debate which one, if any, is correct.

Weak-form Efficient Market Hypothesis The weak form of the hypothesis

says that no one can profit from the stock market by looking at trends and patterns within the price of a product itself. It is important to note that this does not rule out profiting from predictions of the price of a product based on data external to the price. We will see examples of prediction based on both in sample and out of sample data, and provide evidence in support of the weak form.

Semi-Strong Efficient Market Hypothesis The semi-strong form rules out all methods of prediction, except for insider trading. This means that if we are only to use public domain information in our prediction attempt, the semi-strong form says that we will be unsuccessful.

Strong form Efficient Market Hypothesis The strong form says that no one can profit from predicting the market, not even insider traders.

Clearly, if we are to predict the stock market using only public information, we must hope that at most the weak form of the Efficient Market Hypothesis is true so that at least then we can use external data to predict the price of a product.

3.3 Self Defeating Strategies

Finally there is the idea of a successful model ultimately leading to its own demise.

The insight is that if there were a simple predictive model that anyone could apply and profit from themselves, then over time all of the advantage will be traded and eroded away.

This is the same reason for the lack of academic papers on the topic of profitably predicting the+ market. If a successful model was made widely known, then it wouldn't take long until it wouldn't be successful any more.

3.4 Conclusions

The three preceding ideas ask us to keep an open mind on stock market prediction. It is possible that we will not be able to do it profitably.

Chapter 4

Review of Existing Work

In this section, we will review existing academic literature in regard to predicting the stock market. We will look at two articles, one in support of technical analysis methods and another in support of machine learning methods. We will see that both leave room for improvement.

4.1 Article 1 - Kara et al. [10]

The first article we will review is *Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange* by Kara et al. [10].

The article uses technical analysis indicators as features to predict the direction of the ISE National 100 Index, a stock index traded on the Istanbul Stock Exchange. The article claims impressive results, up to 75.74% accuracy. The team uses a set of 10 technical analysis indicators which are listed below.

- Simple 10-day moving average
- Weighted 10-day moving average
- Momentum
- Stochastic K%
- Stochastic D%
- Relative Strength Index
- Moving Average Convergence Divergence
- Larry Williams R%
- Accumulation Distribution Oscillator
- Commodity Channel Index

The daily values for the indicators are calculated and coupled with the daily price movement direction, the dependent variable. Two types of model are tested; a support vector machine and a neural network. Results are cross-validated using a single-holdout method. This method of cross-validation is

known to be inferior when compared to other techniques such as k-fold cross-validation [12], but it is unlikely that this would have a drastic effect on the results presented in the article. The team's neural network model had an accuracy of 75.74% and their support vector machine had an accuracy of 71.52%.

Such success using technical analysis is in sharp contrast with work that will be presented later in the report. One reason for the difference in results is that the ISE National 100 index may be more accommodating to technical analysis indicators than the 30 Dow Jones components reviewed in this report. This is understandable since all of the Dow Jones components are traded on the New York Stock Exchange, which is the most computerised exchange in the world. It is likely that there are less trading algorithms competing on the Istanbul stock exchange and it is therefore easier to leverage algorithmic approaches.

However, there is another flaw however in the methodology of the article that is far more likely to have dramatically increased the performance of their algorithms. In section three of the article, the team describes how they prepared their research data. Specifically,

> *The direction of daily change in the stock price index is categorized as "0" or "1". If the ISE National 100 Index at time t is higher than that at time $t-1$, direction t is "1". If the ISE National 100 Index at time t is lower than that at time $t-1$, direction t is "0".*

At first this would appear sensible, but their features also make use of information from time t. For example, their moving average calculation is defined in equation 4.1 where C_t is the closing price at time t.

$$\text{Simple 10-day moving average} = \frac{C_t + C_{t-1} + ... + C_{t-10}}{10} \tag{4.1}$$

In 4.1, the simple moving average is calculated using the closing price at time t and $t-1$. In fact all features used in the article use some information from time t and $t-1$, as well as others. But recall that the dependent variable is a classification of the difference between prices at times t and $t-1$. This means, in effect, that they are using information about tomorrow to predict tomorrows price. In a real world situation, no trader or trading algorithm would have the same level of information about tomorrow when they are making their predictions. If we wish to build a stock market prediction model that is useful in the real world, then we must only provide the model with information it could have in actuality. In reality, no model would have access to information about tomorrow when making a prediction. Clearly, this is ignored by the article which provides its models with features which contain a large amount of information about tomorrow. This can not be ignored when considering the team's reported success.

Nevertheless, it can be argued that the work may still be useful. Although the impressive figure of 75.74% prediction accuracy would be impossible to

replicate in a real world model because of the problems described above, perhaps it tells us something else. We could reinterpret the 75.74% accuracy result not as predictive accuracy, but how well the technical analysis indicators agreed with true price movement on a given day. In other words, even though the indicators ultimately had access to information which included the true price movement, they agreed with it 75.74% of the time.

Using this somewhat liberal interpretation, Kara et al. [10] have not demonstrated the predictive power of technical analysis, but demonstrated that it might be worth investigating. Later in this report, we will apply the corrected methodology and attempt to build a model using some of the same features.

4.2 Article 2 - Shen et al. [19]

The second article we will look at is *Stock Market Forecasting Using Machine Learning Algorithms* by Shen et al. [19].

The article makes a case for the use of machine learning to predict large American stock indices, including the Dow Jones. The article boasts a 77.6% accuracy rate for the Dow Jones specifically.

The team uses a set of 16 financial products and and uses their movements to predict movements in American stock exchanges. Some of the financial products used are listed below.

- FTSE index price
- DAX index price
- Oil price
- EURO/USD exchange rate

The article makes good use of explorative methods in their data preparation stage. They show using graphs there there is a some features may have predictive power because of their correlation to the NASDAQ index.

They then go on to perform feature selection based on the predictive power of each feature on its own. The results presented in this section, the predictive power of single features, are very similar to results that we will show later in this report.

After they have selected their top 4 features they compare a Support vector machine model to a Multiple Additive Regression Trees model for predicting the daily NASDAQ trend. Their winning model is the SVM with 74.4% accuracy.

While the results presented by Shen et al. [19] in this article appear to be very much in line with results that will be later presented in this report, it is quite vague in terms of the methodology that was used. For instance, there is no record of what model was used in the feature selection step or how cross-validation was carried out to calculate any of their results. However their results will be supported by results in this report, so it is safe to assume that no critical errors were made in these steps.

After they successfully train and test a model, they use it to simulate a real life trading environment. Their models perform extremely well on the simulated

trading example with an average of an 8% return rate every 50 days. This is where the results of this article and the results of our report diverge. The simulation in the article fails to account for overlapping trading hours. For instance the FTSE, which is traded in London, and the Dow Jones, which is traded in New York, are both trading simultaneously for three to four hours each day. This is enough time for the New York stock exchange to influence the London Stock stock exchange. It follows then that it is incorrect to use the closing prices in London to predict New York's movements.

While training and testing the models, both the article and this report ignore the overlapping of trading hours. However, this should certainly not be ignored when simulating real world trading. Later in this report, we will simulate real world trading using Quantopian. In Quantopian we will have access to intraday prices which will allow us to correctly account for the overlapping trading hours.

Chapter 5

Attacking the Problem - Fundamental Analysis

The first approach we took to solving the problem of market prediction is to use Fundamental Analysis. This approach tries to find the true value of a company, and thus determine how much one share of that company should really be worth. The assumption then is that given enough time, the market will generally agree with your prediction and move to correct its error. If you determine the market has undervalued a company, then the market price should rise to correct this inefficiency, and conversely fall to correct the price of an overvalued company.

Graham et al. [7] laid the groundwork for the field with the book *Security Analysis*. He encouraged would-be investors to estimate the intrinsic value of a stock before buying or selling based on trends, a novel idea at the time. It stands as testament to his approach that his only A+ student was Warren Buffet, who methodically applied the strategy and has enjoyed renowned success since [18]. This gives us some hope, but we should be cautious and remember that the market might behave differently today than it did before.

It should be noted that Fundamental Analysis is compatible with the weak form of the Efficient Market Hypothesis. As explained earlier, the weak form does not rule out prediction from data sources external to the price, which is what we will use to determine our fair market price.

We will look at two of the most common metrics used in Fundamental Analysis, Price to Earnings Ratio, and Price to Book Ratio to try and predict long term price movements on a year to year basis. This is the typical prediction range for Fundamental Analysis.

5.1 Price to Earnings Ratio

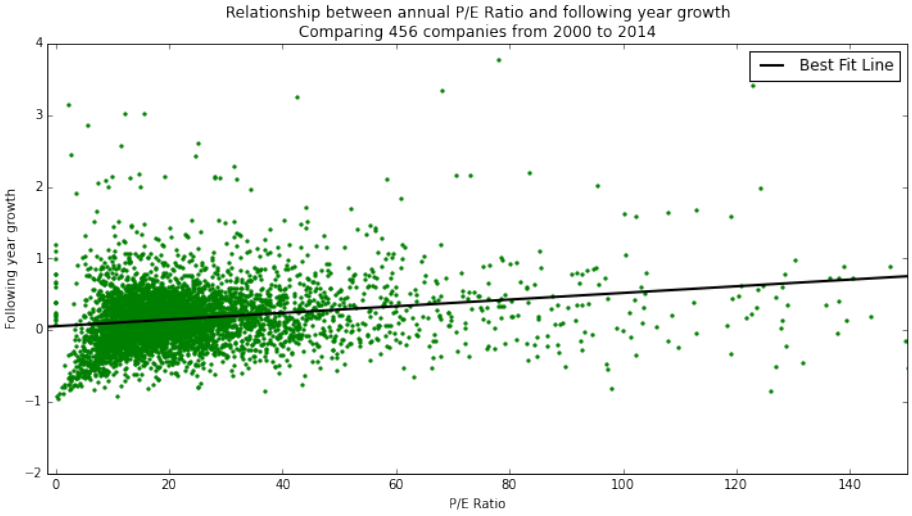
The first metric for the value of a company that we will look at is the Price to Earnings Ratio. The Price to Earnings Ratio calculation is defined in equation 5.1.

$$\text{P/E Ratio} = \frac{\text{Share Price}}{\text{Earnings Per Share}} \tag{5.1}$$

Roughly speaking, what the P/E Ratio calculates is the price an investor is willing to pay for every $1 of company earnings. If this ratio is high, it might be a sign of high investor confidence. If investor confidence is high, that might mean they expect high returns in the following year. We should then expect to see a relationship between high P/E ratio and high returns in the following year.

To investigate this relationship, we plotted the P/E ratio of 456 companies on the 31st of December against the change in stock price for the following year. We gathered these data points from the year 2000 to 2014. Figure 5.1 plots of this relationship.

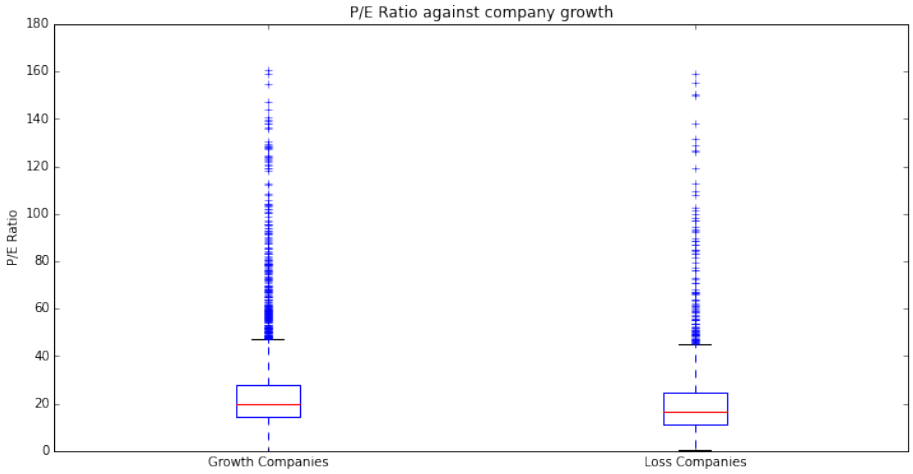
Figure 5.1: Relationship between P/E Ratio and following year growth



The best fit line was calculated using the standard Least Squares method. If the P/E ratio was indeed predictive, we might have expected a better fitting line.

We can investigate the data further using a boxplot. Figure 5.2 divides companies into two categories. The first category, *Gain Companies*, are companies whose share price increased in a given year, and the second category, *Loss Companies*, are companies whose share price fell in a given year. The box plot shows the distribution of P/E Ratios for each category.

Figure 5.2: Investigation of P/E Ratio predictive value using Box plot



If the P/E Ratio was predictive, we would have expected a noticeable difference in the P/E Ratio distribution of companies whose share increased, and those whose share price decreased. However, this is not the case. The P/E Ratio distribution between these categories is almost identical. We can therefore conclude that the P/E Ratio has little predictive value when it comes to estimating company performance for the following year.

5.2 Price to Book Ratio

The second metric for the value of a company that we will look at is the Price to Book Ratio. The price to Book Ratio calculation is defined in equation 5.2.

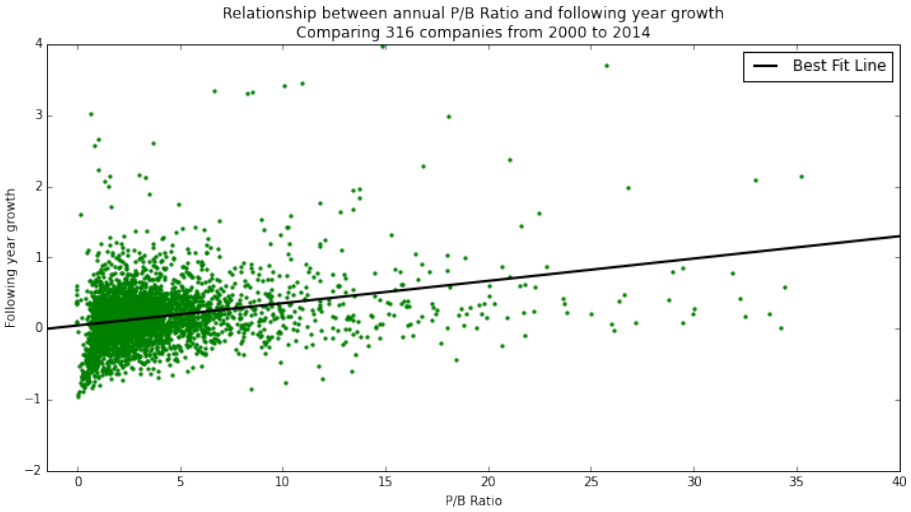
$$\text{P/B Ratio} = \frac{\text{Share Price}}{\text{Book Value of Company}} \tag{5.2}$$

Informally, what the Price to Book Ratio calculates is the ratio between the value of a company according to the market and the value of the company on paper. If the ratio is high, this might be a signal that the market has overvalued a company and the price may fall over time. Conversely if the ratio is low, that may signal that the market has undervalued the company and the price may rise over time. We should then expect to see a relationship between high P/B ratio and low returns in the following year.

To investigate this relationship, we plotted the P/B ratio for of 316 companies on the 31st of December against the change in stock price for the following

year. We gathered these data points from the year 2000 to 2014. Figure 5.3 plots of this relationship.

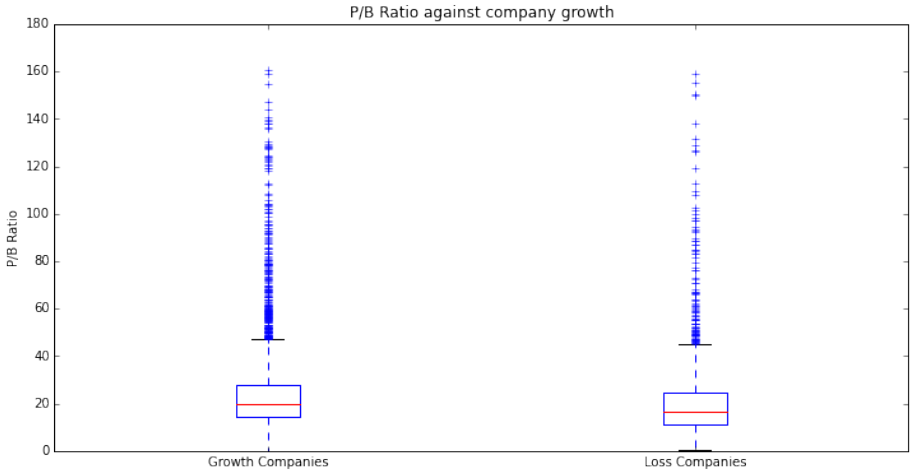
Figure 5.3: Relationship between P/B Ratio and following year growth



Just as in the P/B diagram, the best fit line was calculated using the standard Least Squares method. The slope of the best fit line here is greater than that of the P/E ratio which is the opposite of what we might have expected. The data suggests that a high P/B ratio is somewhat predictive of a high growth in the stock price. This is an unexpected result and is directly opposed to available literature on the subject [9]. This suggests that a high P/B ratio may act as a signal of investor confidence, a better signal than the P/E Ratio.

To better understand the predictive value of the P/B Ratio we can use a box plot. Figure 5.4 divides companies into two categories, exactly as in the earlier P/E Ratio example. The box plot shows the distribution of P/B Ratios for each category.

Figure 5.4: Investigation of P/B Ratio predictive value using Box plot



It is evident that figure 5.4 tells a very similar story to figure 5.2, the P/E Ratio box plot. We can see that companies that grew and companies that shrank had an almost identical distribution of P/B Ratios. If it were predictive, we would have expected different distributions for each category. We can therefore conclude that the P/B Ratio also has little predictive value when it comes to estimating company performance for the following year.

5.3 Limitations of Fundamental Analysis

There is an obvious problem in Fundamental Analysis. We are trying to quantify the true value of a company when almost every company has in some way or another some purely qualitative value.

Fundamental Analysis methods do not attempt to capture these qualitative values. How should one quantify the value of a brand, the size of its customer base, or a competitive advantage? Until these values are quantified, it leaves a large gap in what an algorithmic style approach can achieve. What algorithm, for instance, would have valued WhatsApp at $22 billion while still making a year-on-year loss? [3]

These are three examples of some of the many qualitative values a human investor might take into account when deciding who to invest in. In fundamental analysis we are limited to purely quantitative company metrics.

5.4 Fundamental Analysis - Conclusion

We evaluated two Fundamental Analysis metrics and found no conclusive proof of their predictive value.

These predictions are also very long term, looking one year into the future. Predictions on this time scale are not the focus of the project. Instead we will focus on predicting daily trends in the market. Due to these issues that we moved away from Fundamental Analysis.

Chapter 6

Attacking the Problem - Technical Analysis

The second approach we take at market prediction is Technical Analysis.

This approach tries to find recurring patterns and trends within the daily price of the stock itself. Technical Analysis goes directly against all forms of the Efficient Market Hypothesis. As described earlier, even the weak form of the Hypothesis rules out prediction using historic price data alone.

Nevertheless, Technical Analysis remains popular in online non-academic literature. This is likely because it is simple, intuitive, and what many casual investors imagine a professional trader might use. The quintessential image of what a professional trader does is analysing a historic price graphs using sophisticated diagrams and charts. This is Technical Analysis.

6.1 Broad Families of Technical Analysis Models

If a casual investor was to do some research into trading on the stock market using Technical Analysis they would, perhaps unknowingly, encounter two broad categories of models. We will demonstrate that one of these is implausible in theory and in practice, while the other although sound in theory, does not work in practice.

6.2 Naive Trading patterns

The first family of Technical Analysis methods we will look at are those that do not work in theory or in practice. These methods are based on looking for very high level patterns in the stock market price and using these patterns in an attempt to predict the following price movements.

Among the most common of these patterns is the Head and Shoulders pattern, and it is one of the worst offenders of poor methodology in the Technical Analysis of stock market field.

Figure 6.1: Head and Shoulders Pattern [8]

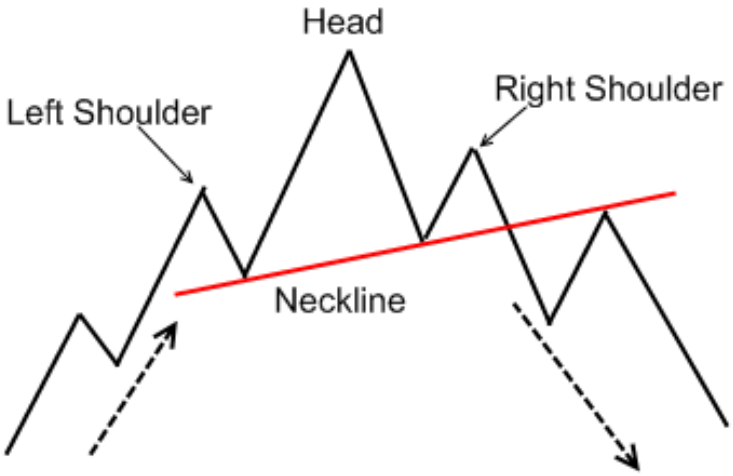


Figure 6.1 shows a bearish head and shoulders pattern. In this context, bearish is taken to mean falling share prices. The idea is that if a trader sees this pattern, they can expect the market price to then fall. To spot this pattern, a trader is supposed to look for two smaller peaks (the shoulders) surrounding a larger peak (the head). However, Neftci [16] shows that the pattern does not, and indeed cannot, provide any useful information.

The first issue is that the pattern cannot be identified until after it has happened. Not until the price falls away below the right shoulder, does it become apparent that a head and shoulders pattern has just occurred. But this information needed to identify a the head and shoulders pattern is exactly what it was supposed to predict. This leaves no useful information for the trader.

Because of the lack of theoretical support, it is easy to find many additional problems with the head and shoulders pattern. The most obvious problem is that because we cannot identify the pattern until after the fact, we can never tell which way the market should move even if the pattern was predictive. Suppose we have observed a series market movements that appear to be similar to those in the diagram up to peak of the right shoulder. We have no way of telling whether the market will continue upwards, or follow the head and shoulders pattern downwards. If the pattern moved upwards when it was supposed to be at the right peak, the right peak could turn out to be a left shoulder of another possible head and shoulders, or even a head peak.

A common error made using patterns such as this is that the analyst does not recognise instances where the pattern failed, but instead would argue that

the pattern never existed. For example, if the price were to rise after the right shoulder, many investors would tell you that it was not be a head and shoulders pattern in the first place. This is a systematic acceptance of confirmation bias.

In short, it is impossible to get any useful information from the head and shoulders pattern. However this does not appear to stop investors attempting to use it. A casual investor doing an internet search about trading patterns will more than likely bump into a blog post or an apparently authoritative source telling them how to use this pattern, or one like it, to profit on the market.

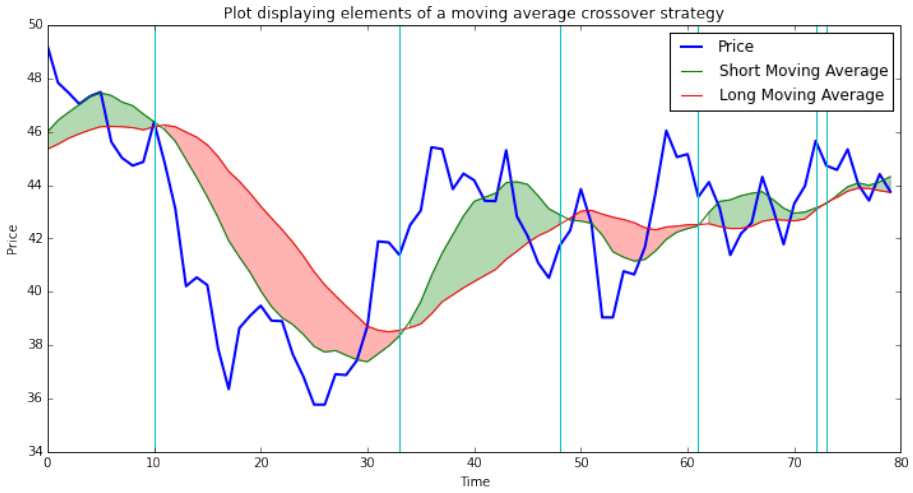
There is no shortage of similar patterns to be found in online literature, but almost all fall into the same problems as the head and shoulders pattern. All information available from these models is only useful in retrospect.

6.3 Moving Average Crossover

Next, we move to Technical Analysis models that are sound in theory. These models work on a statistical basis rather than patterns and make explicit predictions about the future. One of the simplest and most common models of this type is the Moving Average Crossover strategy.

The moving average crossover strategy relies on the interaction between two moving averages of different periods. One is a moving average over a short period, and the other is a moving average over a longer period. For example, the short moving average might be the mean price over a period of the last 10 days, and the long moving average might be the mean price over a period of the last 20 days. When the short moving average crosses under the long, this can be interpreted as a negative signal that the market is trending downwards. Conversely if the short moving average crosses over the long, this can be interpreted as a positive signal that the market is trending upwards. Accordingly, the points where these events happen are called the crossover points and can be categorised into negative and positive crossovers points.

Figure 6.2: Moving Average Crossover



In figure 6.2, the red areas are where the short moving average is below the long moving average and the green areas are where the short moving average is above the long moving average. The diagram seems to give us hope for this strategy. The large green and red areas on the left of the diagram do indeed appear to be predictive of market upward and downward trends respectively.

However, while it is attractive to look at the crossover points on the left of the diagram, one should not ignore the less significant crossover points on the right of the diagram. These are crossover points just as much as the ones on the left area but these are not as predictive for market trends. The goal is to to choose long and short periods moving average to maximize the predictive value.

6.3.1 Evaluating the Moving Average Crossover Model

To evaluate the predictive value moving average crossover model, we attempted to build a predictor using these crossover signals (positive and negative) as the input features and the market trend for the following day as the dependent variable we are trying to predict. We performed a rigorous evaluation of long and short term pairs on a training set, and tested the winning long/short term pair against an independent test set.

To perform this analysis, the database previously assembled in the Data and Tools section was used.

This database was then joined with 50 new moving average columns. The first of these columns contained the 1-day moving average price, the second contained the 2-day moving average price, etc., up to the 50-day moving average price. This precomputation of the moving averages greatly decreased the time taken to train the model.

The data was split into a training and test set. For simplicity, the data was

divided based on company. We chose 20 random companies and used them as the training set. The remaining 10 companies were used as a test set. This is similar to the single-holdout method, which under normal circumstances is not considered to be statistically credible. However there was sufficient data in this case for single-holdout to be viable. The training set contained 74,000 trading days between all 20 companies the test set contained 38,000 trading days between the 10 companies.

The model itself was purposefully kept extremely simple so as to remain true to the intended usage of the moving average strategy. When a positive crossover occurred (short crosses over long), the model predicted the stock price would increase tomorrow, and when a negative crossover occurred it predicted the stock price would fall tomorrow. Finding the optimal pair of long and short moving average periods is equivalent to finding the best hyperparameters for the model. This is model selection.

We perform a grid search over all possible long and short period pairs. For each period pair, we find the crossover points between them. For each crossover point, we make a prediction based on its positivity or negativity, and compare tomorrow's predicted trend against the actual trend. We can then calculate the accuracy of this short/long period pair and remember it if it is the best so far.

When we have iterated over all possible long and short pairs, we will have found the best period pair for predicting tomorrow's trend in the training set. Table 6.3.1 displays the top 5 short and long period pairs and their test accuracy.

Rank	Short Period	Long Period	Test set accuracy
1	8	35	0.5229
2	8	34	0.5219
3	8	33	0.5127
4	2	37	0.5073
5	2	36	0.5016

The winning pair after model selection was 8, and 35 for the short and long period respectively. We then cross validated this against our test set. Our accuracy on the test set using the 8, 35 pair was 0.5157. This is slightly lower than our training score, as should be expected.

However, both of these scores are still extremely poor. They perform only marginally better than a coin toss. We can put into context how well the model performs using the Kappa Statistic. The Kappa Statistic compares the model's performance to a random version of itself. The Kappa Statistic is defined in equation 6.1 where $P(A)$ is the observed proportion of agreement and $P(E)$ is expected proportion of agreement [6].

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{6.1}$$

This model scores a kappa of 0.0427, which is far from statistical significance; the model performs no better than random in any significant way. We must conclude that the Moving Average Crossover is not predictive in any meaningful way, at least for predicting the movement of companies on the Dow Jones.

6.4 Additional Technical Analysis Models

Although we had no success using moving average crossovers, there are many other technical analysis indicators that are regularly used by traders. We will evaluate a further five of the most common indicators. These indicators are described below.

Weighted Moving Average A weighted moving average is the average where later data points are given more weight than earlier data points. In terms of this experiment, the data points will be the prices of individual days. The intuition in using the weighted moving average indicator follows an idea of reverting to the mean, but in this case the mean is calculated using weighted prices. Equation 6.2 defines weighted moving average wma_d^n, on day d using the previous n days where p_d is the price on day d.

$$wma_d^n = \frac{n * p_d + (n-1) * p_{d-1} + ... + 1 * p_{d-n}}{n + (n-1) + ... + 1} \tag{6.2}$$

Momentum A momentum indicator calculates the relative change in price over the last n days. This is based on a straight forward percentage change. Equation 6.3 defines $momentum_d^n$ which calculates momentum on day d by looking at the price difference from n days ago.

$$momentum_d^n = \frac{p_d - p_{d-n}}{p_{d-n}} \tag{6.3}$$

Bollinger Bands A Bollinger Band indicator calculates the number of standard deviations a price is away from a moving average. The intuition is that if the current price is far away from the mean price, then it might signal an upward or downward trend is to follow. Equation 6.4 defines $bollinger_d^n$ where μ_d^n is the mean price on day d of the last n days and σ_d^n is the standard deviation in price on day d of the last n days.

$$bollinger_d^n = \frac{p_d - \mu_d^n}{\sigma_d^n} \tag{6.4}$$

Relative Strength Index The Relative Strength Index (RSI) attempts to capture the trading strength of a stock, i.e will how likely a stock price is to continue rising or falling. Borrowing from Wong et al. [20] for the intuition,

> *Readings of 100 imply that there are pure upward price movements, while readings of 0 imply that there are pure downward price movements. Hence a reading close to 100 indicates an overbought market, while a reading around 0 indicates an oversold market.*

Much like our other indicators then, RSI should act as an indicator that the market price is about to revert back to a some kind of mean.

Calculation of RSI is done in two parts. First RS_d^n is calculated. RS_d^n is, informally, the average price increase over the last n days divided by the average price decrease over the last n days. RS_d^n is then used to calculate RSI_d^n which is the relative strength index on day d using the preceding n days.

$$RS_d^n = \frac{\frac{1}{n}\sum_{k=d-n}^{d}(p_k - p_{k-1} \mid p_k > p_{k-1})}{\frac{1}{n}\sum_{k=d-n}^{d}(p_k - p_{k-1} \mid p_k < p_{k-1})} \tag{6.5}$$

$$RSI_d^n = 100 - \frac{100}{1 + RS_d^n} \tag{6.6}$$

6.4.1 Evaluating the Indicators

In the previous section, when looking at the moving average crossover strategy, we built an extremely simple model that predicted the trend based on a signal. This signal was relatively easy to define, there was a signal when the long and short moving averages crossed over on each other. However with the indicators mentioned above, the definition what constitutes a signal and what doesn't is far more ambiguous. For instance, in regard to the Bollinger Band indicator there is no agreed upon threshold of what should signal an upward or downward trend.

Training many different models for every possible set of signal thresholds and parameters would have been extremely computationally expensive. A different approach was required. Instead of generating all signals thresholds and parameters and using a simple model, the raw values from the indicators were used in a more complex model. A more complex model should be able to hypothesise what these signals should be, which avoids having to predefine them.

These models will employ machine learning techniques and will attempt to learn the important signals from our technical analysis indicators. If these models perform well, given the indicator data as input, it will mean that the indicators is indeed useful. If, on the other hand, these models are not successful it probably (but not certainly) means that the indicators are not useful.

6.4.2 Data Preparation

Before we begin training the models, we must prepare the data to feed them. Because no signals had to be defined, this step was relatively straight forward.

The primary price database originally prepared in the Data and Tools chapter was used for this experiment. For each company, its price history was iterated over and the value of each indicator for each day was calculated. The results of these calculations were stored in a temporary collection.

When iteration was completed, the data for each company was individually standardised so that both high and low priced companies could be used in training and testing the same model. Standardisation shifts and scales the data so that it has a mean of 0 and a standard deviation of 1. It should be noted that standardising in this manner is not correct methodology. Ideally, the training data should be standardised before the test data, and the test data should be standardised using calculations based on the training data. This is done to avoid leakage, i.e when the test data influences the training data. However, this would be difficult to do in this situation as each each company needed to be standardised separately and no division of test and training data has been made yet. Since this error will usually artificially increase the model's performance, and our models will later fail to perform anyway, this error is ignored.

With each the indicator data of each company standardised, the data is stored for later use in testing models. There are 95,407 examples in the database.

6.4.3 Error Estimation

In this step, we determine which class of model should be used on this data. We will compare three types of model, a k-Nearest Neighbours model, a Logistic Regression model, and a Naive Bayes based model. We also need to search a range of hyperparameters for each model. For instance, what value of k do we use in k-NN? To get an error estimation of each model type, we do a grid search over all hyperparameter values and test using nested kFold cross validation. In this case, we split the data into 10 outer folds and 10 inner folds. The inner folds determine the winning hyperparamter which is cross validated by the outer fold. In total, for each hyperparameter there are 100 models trained. Table 6.1 details the models tested, the hyperparameters searched, and their estimated accuracy.

Table 6.1: Error Estimation Scores

Model Name	Model Specific Hyperparameters	Estimated Accuracy
LogisticRegression	Norm penalization: l1 and l2	0.5143
KNeighborsClassifier	k: 5 to 15, weights: uniform and distance	0.5034
GaussianNB	-	0.5131

It is evident from table 6.1 that none of these models were terribly successful. This is a strong sign that the input data, the indicator values, were not predictive of the following day's price trend.

This does not bode well for technical analysis in general. All technical analysis methods that we have tested have failed. However it would seem that our work is supportive of most serious academic literature on the subject. Prominent economist Burton Malkiel summarises the academic attiude towards technical analysis quite well [14].

> *Obviously, I am biased against the chartist. This is not only a personal predilection, but a professional one as well. Technical analysis is anathema to the academic world. We love to pick on it. Our bullying tactics are prompted by two considerations: (1) the method is patently false; and (2) it's easy to pick on. And while it may seem a bit unfair to pick on such a sorry target, just remember': it is your money we are trying to save.*

Whether deserving of it or not, it would be right to analyse what went wrong in the tests that were carried out. Does the fault indeed lie with lack of predictive power in the technical analysis indicators? Perhaps it is that our models were not complex enough to capture their signals.

This question will be answered in the following chapter where we will question whether there is any useful information in the price history at all. Ultimately, we will provide strong evidence that there is in fact no useful information in price history. It logically follows then that any method that tries to leverage price history is doomed to fail. This includes technical analysis models.

6.5 Common Problems with Technical Analysis

For a casual investor, navigating online literature in this area poses a significant challenge. An extremely common theme in this literature is the poor methodology applied to evaluating trading patterns.

We have seen two examples of confirmation bias when we looked at the Head and Shoulders pattern and when we looked at Moving Average crossover points. In the former, patterns that didn't fit the narrative were simply ignored and in the latter people focused too heavily on the instances where it did work.

Even when there is no confirmation bias present, there is very rarely any proper separation of training and test set. Correct methodology would separate these examples so that one could accurately estimate how the model would perform given unseen examples, like it would have to do in reality. This problem

is prevalent when looking for short and long periods in moving average crossover. What many practitioners do is find the best period pair for their given time period and expect that to be just as predictive in future periods. This is incorrect methodology. You will always be able to overfit a model to perform well on a single piece of data, but this may not carry over to unseen examples.

Above, we applied the correct methodology. In our experiments, the data was split into test and training sets and hyperparameters were determined through cross-validation. This gives us a true estimate of how our best estimator carries over to future data. This proper methodology is not common in online literature.

6.6 Technical Analysis - Conclusion

It might have been expected, that given the popularity of Technical Analysis for stock market trading, that there would have been a more positive result. However, somewhat surprisingly, the data shows that there is little predictive value to be found in Technical Analysis.

Chapter 7

Attacking the problem - Machine Learning

Our final approach to attacking the problem of stock market prediction is to look at machine learning. With machine learning, we will be building models that learn to exploit relationships within the data that we might have otherwise missed using Fundamental Analysis or Technical Analysis.

7.1 Preceding 5 day prices

In Technical Analysis, we attempted to find patterns and trends in the data that we could use to predict the price movement, the trend, for the following day. Ultimately Technical Analysis failed to produce any notable results, but perhaps it was because the models are not complex enough to capture any hypothetical pattern that might exist in market data.

Similarly to Technical Analysis, in this section we will try to apply machine learning techniques to the price of the stock itself. Again, the Efficient Market Hypothesis says that it should not be possible to gain any predictive value from the price alone.

The data that we will be using is the percentage change in closing price of the stock of the preceding 5 days. The dependent variable will be the trend of the 6th day, i.e whether the stock price fell or rose. This dataset is built using the dataset assembled in the Data and Tools section. Table 7.1 is an extract of the first 4 rows in the dataset where *trend* is the encoding of the 6th day according to equation 2.2.

Columns labelled day1 to day5 are the percentage change in closing price of the preceding 5 days. We will use the 6th day trend as the dependent variable. The dataset gathered contained 206635 examples. It was gathered from the the daily closing price of companies in the Dow Jones from the year 2000 to 2014. Before feeding the data into the models as discussed later, the data rows were

Table 7.1: Data Extract

	day1	day2	day3	day4	day5	6th day trend
0	0.0492	-0.0029	0.0176	0.0115	0.0028	Loss
1	-0.0029	0.0176	0.0115	0.0028	-0.0142	Loss
2	0.0176	0.0115	0.0028	-0.0142	-0.0028	Gain
3	0.0115	0.0028	-0.0142	-0.0028	0.0115	Loss

randomly permuted to remove any bias the model could learn from an ordered dataset.

7.1.1 Error Estimation

After we have gathered the data, the next step in building our model is choosing which base model we should work with. This step is called Error Estimation. It involves training and testing the performance of various models on the dataset to see which one we should focus on optimizing.

We tested each model by doing a nested kFold test. In this case, we split the data into 10 outer folds and 10 inner folds. The inner folds determine the winning hyperparamter which is cross validated by the outer fold. In total, for each hyperparameter there are 100 models trained.

Table 7.2: Error Estimation Scores

Model Name	Hyperparameters Tested	Nested KFold Accuracy
LogisticRegression	Norm penalization: l1 and l2	0.5343
KNeighborsClassifier	k: 1 to 10, weights: uniform and distance	0.5172
GaussianNB	-	0.5292

Table 7.2 shows the accuracy scores of each model that we tried on the given data. A support vector machine based model was also tried, but it proved too slow to train with the computational resources at hand. However in smaller trials the SVM model did not seem to be significantly better than any of the model types presented above.

It is immediately clear that there is a problem here. None of the models tested scored significantly above 0.5% accurate in our classification test. A coin toss predicting the outcome of the dependent variable would perform similarly to what these models did. Although some models appear to be slightly better, we cannot place too much value in the actual value cross validation score in error estimation. The nested kFold method does not give us the true accuracy of the model, it only gives us an estimate which should be good enough to choose which model we carry forward to Model Selection. Differences of 1% or 2% at this point do not point to any clearly superior model type. We conclude that none of these models will work at predicting the the trend on the 6th day given the trends of the previous 5 days.

7.1.2 Analysis of Model Failure

The failure of all models there were tested gives us a strong indication that something is deeply wrong somewhere. In this section, we will show that there is good evidence that the problem is in fact that there is no information to be found from the previous 5 day prices. This is entirely supportive of the Efficient Market Hypothesis.

Model Complexity

It is important here to clarify what we mean by model complexity. The complexity of a model is the size of the set of hypothesis that it can produce. A hypotheses is a guess that the model can make about the relationship between input features and the dependent variable. A linear model would guess, or hypothesize, that the relationship is a linear combination of the input features. A nearest neighbours model would guess that the relationship is going to copy previously seen examples. A model that can produce a larger number of hypotheses is more complex than one which can produce a smaller set. It is also important to note that these hypothesis sets do not necessarily overlap. One model may be more complex than the other, but that is not to say that the best hypothesis does not belong to the model of lesser complexity.

It is possible that the models in the previous section failed because the models tested were not sufficiently complex, or that they did not cover the optimal set of hypotheses. However, the models and hyperparameters that were tested covered a very large hypothesis base. If there was a good hypothesis to find, it is reasonably that one of the models should at least have been better than a coin toss at predicting the trend.

We can also inspect the effect of model complexity visually using graphs. For convenience we choose to model the kNN model here. Analysis of the kNN model is simpler and no less consequential than the other models.

The complexity of kNN can be varied by changing k, where k is the number of neighbours the model will consider to make its prediction. A low value of k means the model will look at only a few of the closest neighbours to attempt to infer the value of the input, and conversely a high value of k means that the model will consider a larger number of neighbours. Recall that our measure of complexity is the size of the hypothesis set. It follows that the complexity of kNN is inversely proportional to k. Given n examples, a uniformly weighted 1-NN model is able to produce n different predictions, but a $n - NN$ model can only produce one prediction.

Figure 7.1: kNN Validation Curve

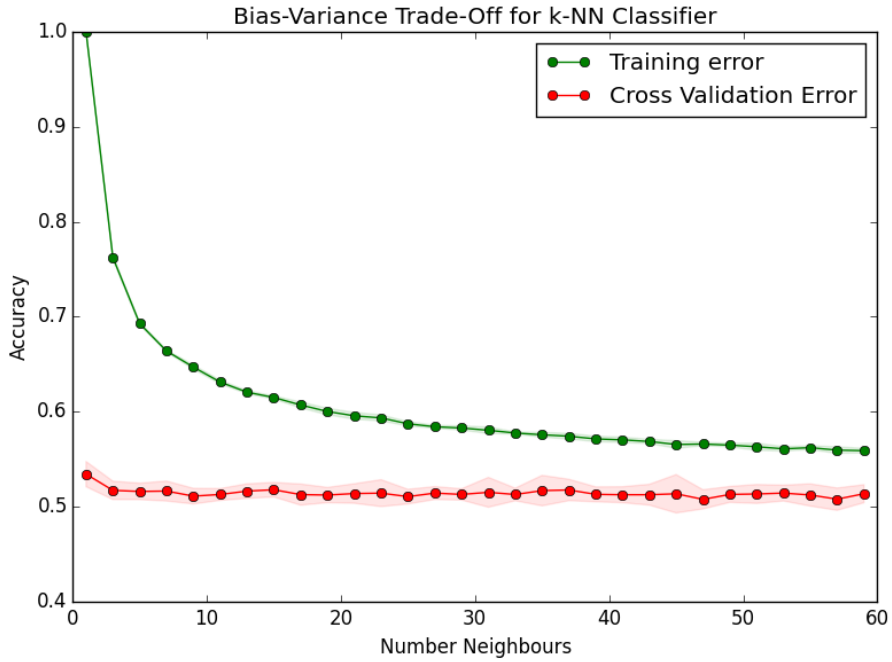


Figure 7.1 plots the validation curve for the kNN model. To generate this graph, we generated 60 kNN models with different k values. The training and test error for each value of k was then plotted. Since the complexity of the model varies with k as explained above, we can interpret the graph as being a performance comparison of models of varying complexity. On the left are models of high complexity and high variance (leading to high training score) and on the right are models of low complexity and high bias. Normally in such a graph we would expect the training and test errors to converge somewhere in the middle, indicating the optimum trade-off between bias and variance. However there is no such clear point in this graph. No matter what the complexity of the model, the Cross Validation Accuracy never rises significantly above 0.5%. We can then conclude that the problem is not due to the complexity of the model.

Training Data

If the problem does not lie with the complexity of our models, then maybe the problem is due to the amount of training data that we have. Perhaps if we had more data, we could build better models.

As mentioned previously, the dataset we used for training and testing these models had over 200,000 examples. This seems like it should be enough, but maybe the stock markets are so complex they need more.

To properly diagnose this, we can plot the learning curve. In the validation curve, we varied the complexity of the model. In the learning curve, we will vary the number of examples presented to the model. Because of computational resources available, we were forced to constrain the number of examples in the learning curve to approximately 15,000. This is much smaller than the 200,000 examples in the dataset, but certainly not too small to ignore the findings. A 50-Nearest Neighbours model was used because there appears to be a slight upwards bump in the validation curve around 50NN.

Figure 7.2: kNN Learning Curve

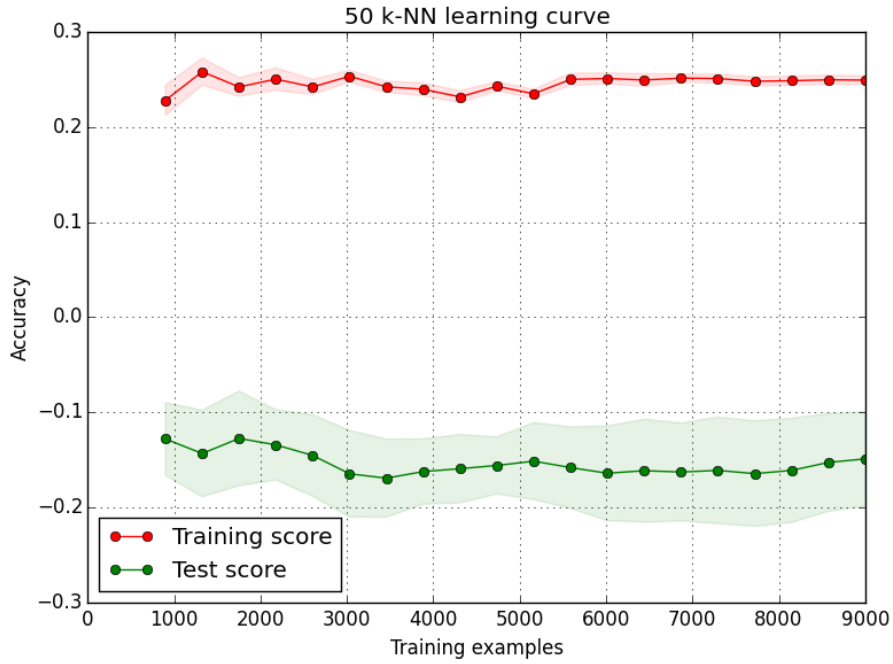


Figure 7.2 shows the effect of varying the number of examples on the model. Normally in a learning curve graph, you would expect the test and training errors to converge. The intuition is that as the model is presented with more data, it is better equipped to come up with better, more general, hypotheses. Because these improved hypotheses are more general we expect the training accuracy to decrease (less overfitting), but the test accuracy to increase (less underfitting). However we can see that this is not the case for our graph. The graph shows that for our model, the test and training accuracy remain apart and do not begin to converge no matter how much data they were given. We can then conclude that the problem is not due to lack of data for training and testing our model.

7.1.3 Preceeding 5 day prices - Conclusion

In this section, we attempted to train a model to predict the trend in the 6th day given the trends of the previous 5 days but all models were unsuccessful. We then analysed why they might have failed and concluded that it was a problem with neither the complexity of the models nor the amount of data we had to train the models.

We must then conclude that the problem must lie in the data itself. It would seem that the preceding 5 day prices contain no information useful in predicting the following day trend. This supports our findings in the Technical Analysis approach and the Efficient Market Hypothesis.

Clearly we must begin to look at using external data.

7.2 Related Assets

With the failure of using the price itself to predict stock movements, we turn to other sources of predictive value. Perhaps the most obvious source we should seek to use is the movement of assets related to the Dow Jones.

Intuitive, one might suppose that when the price of Oil rises, that is a good sign for the Dow Jones and we can expect it to rise too. Similarly, if the price of Oil falls we might expect the price of the Dow Jones to fall with it. In this section, we will search for features that rise and fall with the Dow Jones index.

7.2.1 Data

Since in the last section we showed that yesterday's prices appear to have no influence on today's prices, it is unreasonable to expect yesterday's prices of related assets to influence today's Dow Jones price. Because of this, instead of predicting based on yesterday's prices, we will predict based on price movements in assets that are traded earlier in the day than the Dow Jones. For example, we will use the price movements of assets that are traded in Europe and Asia to predict the trend in the Dow Jones, which is traded in New York. If markets in Europe and Asia are trending heavily in one direction, it might follow that the Dow Jones will also trend that way when the markets open. What is important is that in the real world, we can observe the trends of our related assets before we need to make a trend prediction for the Dow Jones.

However, some market times overlap. For instance, the London Stock Exchange and the New York Stock Exchange are both trading at the same time for approximately 4 hours daily. It would be ideal then to have price data which can be cleanly partitioned into intraday prices before and after the Dow Jones begins. However, this data is not easily available in the public domain. Intraday price data is a commodity and not something which is distributed as freely as interday price data. Quandl, which has been our source of much of the data up to this point, does not offer intraday price data.

We are forced to use subpar data which does not allow for proper preparation. We will continue to use the data provided by Quandl from which we can extract

the daily closing price of each asset and index. It is conceivable that the trend in the New York Stock Exchange can affect closing prices on the London Stock Exchange, even though it only opens in the final 4 trading hours. This means that we have a dependent variable that potentially influences our independent variables. This is bad data preparation. Due to this issue we must be cautious and suspicious of positive results moving forward. It is unfortunate that later in the report, when we implement the trading algorithms in Quantopian, we will show that positive results mentioned in this section do indeed appear to be because of this error in the data.

Temporarily ignoring the aforementioned issues, we gathered data for 8 features from Quandl. These features are detailed below.

DAX The German DAX index is essentially equivalent to the Dow Jones except that its components are 30 major German companies traded on the Frankfurt Stock Exchange.

DAX FUT We also considered DAX futures. Futures are a stock market derivative product.

FTSE The FTSE is London's equivalent to the Dow Jones German Dax. It is traded on the London Stock exchange, and its components are a selection of 100 large UK based companies. It is more commonly known as the FTSE100.

N225 The Nikkei 225 is Japan's equivalent. Its components are 225 large Japanese companies and it is traded on the Tokyo Stock Exchange.

SSE The SSE Composite Index is an index covering all stocks that are traded on Shanghai Stock Exchange.

AUD The Australian Dollar to US Dollar exchange rate.

EUR The Euro to US Dollar exchange rate.

OIL Brent Crude Oil futures prices.

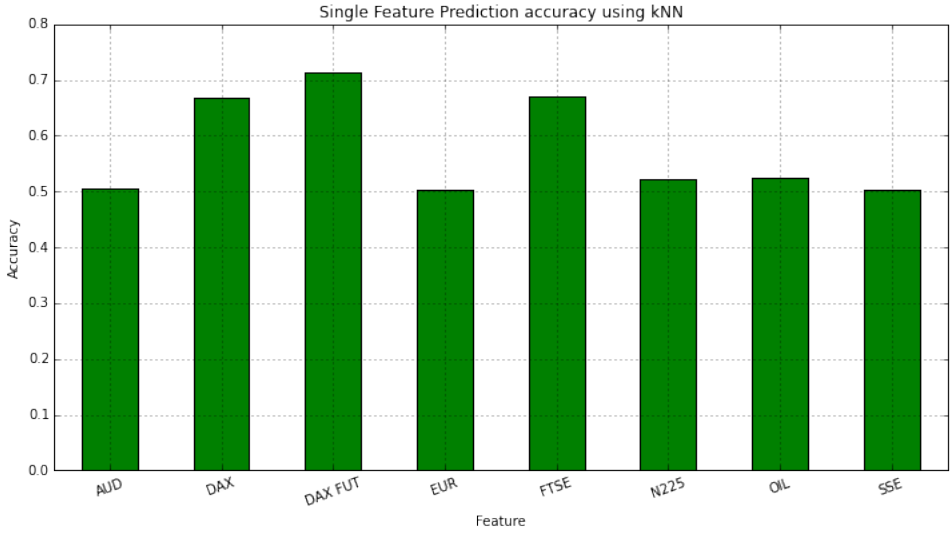
In this section, the value of the Dow Jones index itself is used as the dependent variable rather than price of its component companies. This was due to the fact that the features listed above are not specific to any one company. It could be argued that some features might influence some companies more than others, but determining the predictive value of each feature for each company would have made modelling the problem significantly more complex and computationally expensive. For simplicity, the price of the Dow Jones index was used.

7.2.2 Exploration of Feature Utility

The next step in creating a model to predict the daily trend of the Dow Jones Index is to determine the predictive value of the features. To do this in an

exploratory and qualitative manner, we first analysed the predictive value of each feature on its own. We trained 8 models, one for every feature. Each of these models used exactly one of the features to try and predict the trend for the Dow Jones Index. For each feature a, kNN model was trained and cross validated. Figure 7.3 shows the cross validation results for each feature.

Figure 7.3: Single Feature Prediction Results



It is evident from figure 7.3 that there is indeed some predictive value in this dataset. There are three features that appear to be highly predictive. The German DAX, DAX futures, and the FTSE100 are all roughly 70% accurate without doing any work whatsoever. These results are quite similar to results presented by Shen et al. [19] as discussed earlier.

7.2.3 Modeling

Now that we know that there is at least some predictive value in our dataset, we can begin to create our model.

Error Estimation

Similar to what we attempt to do with the model based on the historical prices, the first step in creating our model is error estimation. In this step we estimate the performance of various classes of model.

We used a nested kFold method to perform the error estimation. The inner kFold performed a Grid Search over a set of hyperparameters. The outer kFold was responsible for the cross validation of the set of hyperparameters found within the inner kFold. The Grid Search searched over two domains. The first

domain was the hyperparameter determining how many of the k best features to keep. For instance, the search might determine that it was best to keep the top 3 most predictive features. The second domain were the hyperparameters specific to the model being tested. Table 7.3 details the model class tested, the model specific hyperparamters searched over, and the cross validation score.

Table 7.3: Error Estimation Scores

Model Name	**Model Specific Hyperparameters**	**Estimated Accuracy**
LogisticRegression	Norm penalization: l1, l2	0.5717
KNeighborsClassifier	$0 < k \leq 25$, weights: uniform, distance	0.7251
GaussianNB	-	0.7208

The scores from table 7.3 look hopeful. We can see that the kNN model has the best estimated accuracy, albeit not by much. This means that this is the model we should bring forward to Model Selection to further optimise it.

Model Selection

Now that we have our winning model class, kNN, we now need to estimate the optimum hyperparameters for our model.

A kNN model has two important hyperparameters, k and how to weigh examples in the neighborhood. k dictates how many neighbours the model should examine to make a prediction. We can then weigh those neighbours either uniformly or by their distance to our input. We can search over the same value space as we did in Error Estimation for kNN, $0 < k \leq 25$ and weights $uniform$ or $distance$. We will also be searching for the best number of features to keep in the model. This can be treated as another hyperparameter.

After performing a grid search over the value space and cross validating each combination of hyperparameters, we find that the best combination of hyperparameters gives us a cross validated accuracy of 74.63% at predicting Dow Jones Index daily trends. Table 7.4 shows the best combination of hyperparameters found.

Table 7.4: Model Selection Results

Hyperparameter	**Best value**
Number of features to keep	3
Number of neighbours to examine	14
Method of weighing neighbours	uniform

Table 7.4 indicates that the search determined that keeping only three of the original 8 features gave us the best cross validation score. This is in line with what we might have expected from the exploration of the predictive value of the features where we pointed out three features that appeared to be highly predictive already.

By training the model on the full dataset with the winning hyperparameters and inspecting the 3 features the feature selector decided to keep, we can see that they are indeed the three features we assumed would be predicted. The German DAX, DAX futures, and the FTSE100 are the features selected in the final model.

7.2.4 Related Assets - Conclusion

In this section we gathered data for 8 features that could conceivably have some predictive value for the Dow Jones. We explored the predictive value of each feature visually to gain some sense of the data and then followed the correct methodology to end up with an optimal model. The final model had an estimated accuracy of 74.63%, which is significant for stock market data.

However, as warned by the data section, we should be cautious of such high accuracy figures. Due to lack of publicly available intraday price data, we might have a dataset that is artificially increasing our result. Later in the report, when we try to translate this model to a trading simulation, we will demonstrate that this error is most likely what is giving such a high accuracy score and that the model would not work as well in actuality.

7.3 Analyst Opinions

With the apparent success of using related assets to predict daily movements in the Dow Jones, we continue our search for additional predictive features. In this section we will attempt to use analyst opinions to predict the same day trend.

An analyst opinion is a prediction of a particular research firm on a particular stock. For instance, a large investment research firm such as JP Morgan might issue an opinion on Intel stock. They may upgrade or downgrade their estimates of stock performance and recommend buying or selling the stock at the current price.

It would seem intuitive that these recommendations should be predictive. There are two reasons why that might be true. The first reason is that these recommendations might truly be predictive of the price. One can imagine that large research firms such as JP Morgan would employ skilled analysts that are able to make accurate predictions. The second reason is less optimistic and assumes that these predictions are self-fulfilling prophecies. If multiple large investment firms upgrade or downgrade their opinion of a stock on the same day, it is certainly conceivable that the opinions themselves are enough to shift the market regardless of their true predictive value.

However, it should not matter to us why they are predictive. If these opinions can predict price trend, for whatever reason, then we should be able to build a model to utilise them.

7.3.1 Data

Conveniently, Yahoo Finance provides an open database of opinions issued by large research firms. The database covers all major stocks including the components of the Dow Jones, the 30 companies of interest to the project.

One small difficulty in obtaining this data is that it is provided in a HTML table on the Yahoo Finance website. This means that it was not directly downloadable as a csv file or an equivalent easy to use file format. Because of this, a small web scraper had to be constructed.

The web scraper was constructed in python. The scraper began by sending 30 HTTP get requests to Yahoo Finance, one for each company. The responses were then parsed using the BeautifulSoup4 HTML parsing library. After the HTML was parsed it was trivial to extract the data in a more usable format. The extracted data for all companies was cached in a single CSV file. In total, 3584 analyst opinions were gathered from the year 2000 to 2014. Table 7.5 shows an extract of the data collected.

Table 7.5: Analyst Opinion Data

Date	**Research Firm**	**Action**	**From**	**To**	**Symbol**
2001-04-25	First Union Sec	Downgrade	Strong Buy	Buy	CSCO
2001-04-25	AG Edwards	Downgrade	Accumulate	Maintain Position	VZ
2001-05-01	Salomon Smth Brny	Upgrade	Neutral	Outperform	PG
2001-05-07	Prudential	Downgrade	Hold	Sell	JPM
2001-05-08	Mrgn Stnly	Upgrade	Neutral	Outperform	CSCO

In this section, predictions will be tailored towards individual companies rather than the Dow Jones index as in the previous section. This is preferable because each opinion relates explicitly to a single company.

Also note that these opinions are almost always issued before the market opens. It is therefore acceptable to use these opinions to predict the movement on the market for that day.

7.3.2 Data Exploration

As we did in the previous section, to get a better sense of our data we will explore it a little further.

The company with the most opinions in the gathered dataset is Intel (INTC) which had 326 individual analyst opinions. To get a sense of whether the data might be useful, we can plot these opinions in relation to the INTC price.

First we filtered the dataset so that we only consider INTC opinions. We then aggregated the opinions that were issued on the same day by different research firms by summing the number of Upgrades and Downgrades. Finally, the aggregated opinions were merged with the INTC stock price. This data was plotted in figure 7.4. Green upwards pointing arrows signal an upgrade and red

downwards pointing arrows signal a downgrade. The size of the triangle represents the number of coinciding opinions across all research firms on a particular day.

Figure 7.4: Visualisation of analyst opinions and INTC price

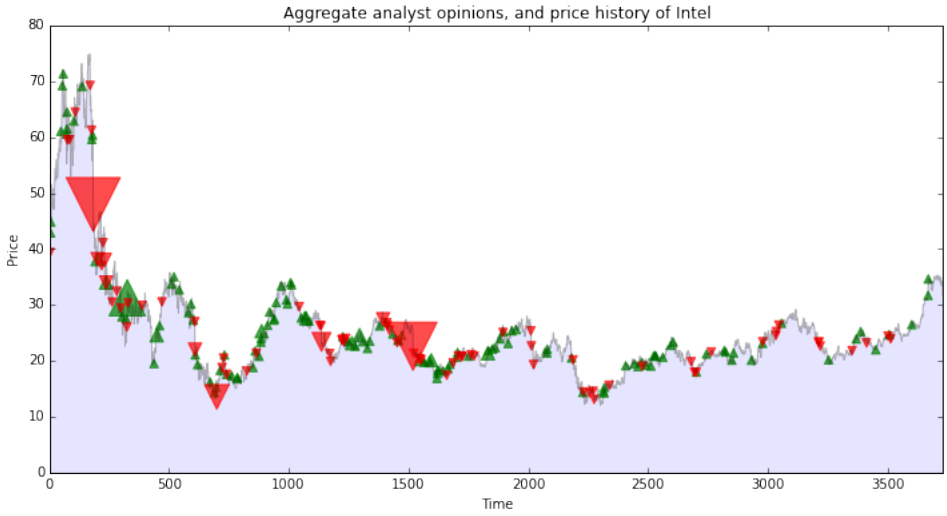


Figure 7.4 provides a visualisation of the relationship between opinion sentiment (Upgrade or Downgrade) and price. The large red down pointing triangle on the left hand side of the diagram is a point representing 9 separate research firms all downgrading their opinion of INTC on the same day. The continued sharp fall in price after this point is evidence that the opinions were predictive.

From visual inspection it would appear there may be some value in these opinions, but we cannot be certain until we attempt to build a model.

7.3.3 Data Preparation

As evident from Table 7.5 all of the feature values in the dataset are labels and categories, not numerical data. This presents a problem. The machine learning library used in this project, sklearn, does not provide models which can handle non-numeric data. We must therefore transform our data into numeric form.

The correct way to transform a label type feature is to use One-hot Encoding. One-hot Encoding will transform a single feature with n unique values into n different features with binary values.

Figure 7.5: One-hot Encoding Example

	Location
0	London
1	Paris
2	New York

(a) Before

	London	Paris	New York
0	1	0	0
1	0	1	0
2	0	0	1

(b) After

Figure 7.5 demonstrates the concept of One-hot encoding. After One-hot encoding, the original feature is removed but no information is lost. To signify a value in the original column, a 1 is placed in the new corresponding column.

For the opinions dataset, it was decided that the most important original columns were *Research Firm*, *Action*, and *To*. *Research Firm* might be important because an opinion from some firms may have more of an influence on the movement than others. The *Action* column is intuitively important because it summarises the sentiment of the opinion into Upgrade or Downgrade. The *To* column is the recommendation of the Research Firm. This is also intuitively important because it could be a recommendation to Buy or Sell.

It was then necessary to One-hot encode each of these three features features. This resulted in a total of 266 new feature columns, one new column for each unique value in the original three columns. Now that all the features have been one hot encoded, we can aggregate rows where more than one research firm issued an opinion for the same company on the same date. We will use a sum operation to aggregate the rows across all columns. For example, if both research firms upgraded a company on the same day then we will merge these rows and place a value of 2 in the upgrade column.

The entire data preparation stage can be done very concisely using the Pandas python library. The code used for this step is shown in listing 7.3.3.

Listing 7.1: Data Preparation Using Pandas

```
dataset = pd.merge(opinions,prices,on=['Date', 'Symbol'])

X = dataset[['Date', 'Symbol']] \
    .join(pd.get_dummies(dataset['Research Firm'])) \
    .join(pd.get_dummies(dataset['Action'])) \
    .join(pd.get_dummies(dataset['To'])) \
    .groupby(['Date', 'Symbol']).sum()

y = [frame['Trend'].values[0] for index, frame in \
        dataset.groupby(['Date', 'Symbol'])]
```

7.3.4 Error Estimation

With our data prepared, we must decide which class of model to use.

We used a nested kFold method to perform the error estimation. The inner kFold performed a Grid Search over a set of hyperparameters. The outer kFold was responsible for the cross validation of the set of hyperparameters found within the inner kFold. The Grid Search searched over two domains. The first domain was the hyperparameter determining how many of the k best features to keep. The second domain is the hyperparameters specific to the model being tested. Table 7.6 details the model class tested, the model specific hyperparamters searched over, and the cross validation score.

Table 7.6: Error Estimation Scores

Model Name	**Model Specifc Hyperparameters**	**Estimated Accuracy**
LogisticRegression	Norm penalization: l1, l2	0.6627
KNeighborsClassifier	$0 < k \leq 20$, weights: uniform, distance	0.6564
MultinomialNB	-	0.6729

The scores from table 7.6 look positive. We can see that the MultinomialNB model has the best estimated accuracy. This means that this is the model we should bring forward to Model Selection.

7.3.5 Model Selection

The MultinomialNB model is a naive bayes classifier and only has one hyperparameter. The hyperparameter, called alpha in sklearn, controls the additive smoothing in the model. What exactly additive smoothing is is out of the scope of this report, but all we need to know is that it is a hyperparameter that should be searched over to optimise the model. Additive Smoothing is a technique used to smooth categorical data [15].

At the same time as we are searching over the model hyperparameter, we will again be searching for the best k features to keep. This is an important step in this case because we have a large number of features, 266 in total, to begin with.

Table 7.7: Model Selection Results

Hyperparameter	**Best value**
Number of features to keep	11
MultinomialNB alpha	0.7333

Table 7.7 shows the best set of hyperparameters found in the search. The winning model had a cross validation accuracy of 67.40%.

7.3.6 Analyst Opinions - Conclusion

In this section we gathered a list of analyst opinions and used them to build a model to predict the same day price movement for companies in the Dow Jones. Although concise in its final version, the data preparation for this section proved difficult to get right. The final model had an estimated accuracy of 67.40% which is an extremely positive result.

Although the data used in this section does not have the same problems the data in the previous section had, we will see similar results when it is simulated in a realistic trading environment. The model does not fair as well in the real world as our results may suggest.

7.4 Disasters

The final source of predictive data we will consider are natural disasters in the United States.

It is easy to imagine why these might be predictive of changes in the stock price. A large storm could damage machinery or interfere with manufacture and logistics. In turn, this could have an impact a company's profit and therefore the stock price.

7.4.1 Data Preparation

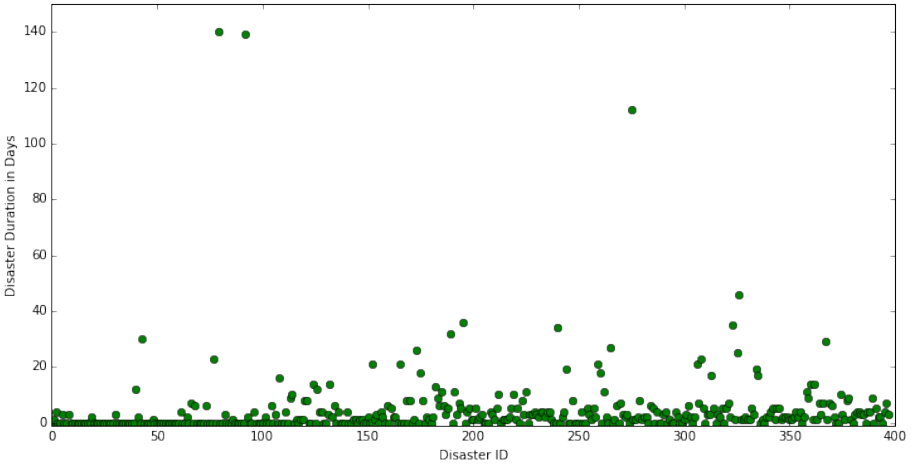
The first task was to gather a database of natural disasters in the United States. This data is provided by EM-DAT, The International Disaster Database [1]. Disasters in this dataset are guaranteed to be relatively large. The website states the qualifications necessary for a disaster to be entered into the database

> *In order for a disaster to be entered into the database at least one of the following criteria has to be fulfilled:*
>
> - *10 or more people reported killed*
> - *100 people reported affected*
> - *A call for international assistance*
> - *Declaration of a state of emergency*

All disasters occurring in the United States from the year 2000 to 2014 were extracted. Before any further preparation, there were 423 entries in this dataset.

Disasters with missing or invalid dates were then removed. Following this, disasters with anomalous durations were removed. Figure 7.6 shows that there were three clear outliers with durations exceeding 100 days. These were removed from the dataset. After all preparation was completed on this dataset, there were 395 disaster events remaining.

Figure 7.6: Anomolies in the disaster data



Similarly to the Related Assets section, this data does not relate specifically to any one company. Because of this, we will look at price changes of the Dow Jones index, rather than any of its component companies.

7.4.2 Predictive Value of Disasters

We will first determine the predictive value of the disaster events. The goal of this experiment is to decide whether the price movements of the Dow Jones during disaster times are any different to non-disaster times. If disasters contain any predictive value, positive or negative, we should be able to observe a difference in market behaviour during disaster times.

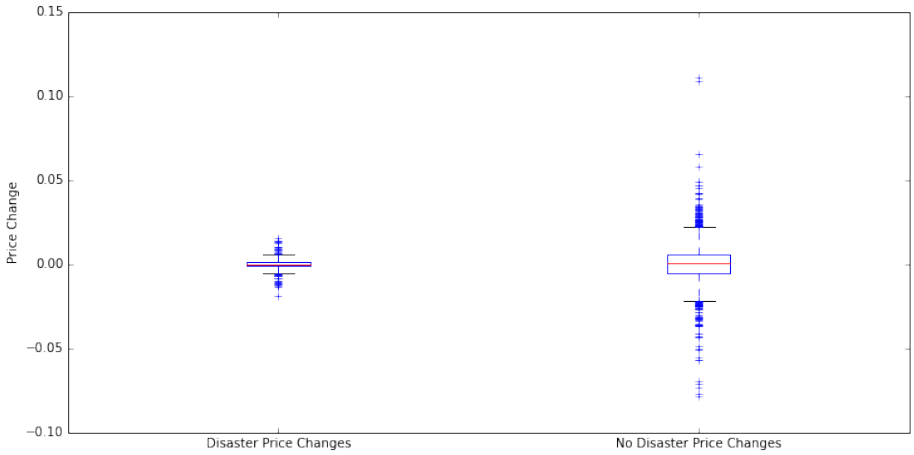
For each disaster, we determined the percentage change in the value of the Dow Jones index over the duration of the disaster. Equation 7.1 shows how this was calculated. $P_{start-1}$ is price of the Dow Jones the day the before the disaster began and P_{end} is the price of the Dow Jones the day the disaster ended. The varying length of disasters is controlled for by diving by the change in price by the length of the disaster. This equates to an average price movement for each day of the disaster.

$$\text{Disaster price change} = \frac{p_{(end)} - p_{(start-1)}}{p_{(start-1)}} * \frac{1}{DisasterDuration} \tag{7.1}$$

Next, daily percentage changes in the Dow Jones were calculated for days where no disaster was occurring. At this point we have two independent, but comparable, datasets. One contains average daily price movements of the Dow Jones in times of Disaster. The other contains daily price movements of the Dow Jones in times of no Disaster.

To visualise these datasets, figure 7.7 shows a boxplot of their distributions. From visual inspection, we can see immediately that the data is probably not useful for prediction. If it were useful, we would expect to see a difference in their distributions. However it is clear that the disaster time price changes distribution is almost entirely contained within one standard deviation from the mean of the non-disaster time price changes.

Figure 7.7: Disaster and no disaster price change distributions



To quantify this, we perform a t-test. This results in a t-value of -0.4783 and a two-tailed p-value of 0.6324. Because the p-value is large, we cannot rule out the null hypothesis, that any difference in the distribution of disaster time price movements and non disaster time price movements is due to random chance.

Because there is no significant difference between these two datasets, we will not be able to use disasters to predict price movements in the Dow Jones index.

7.4.3 Disasters - Conclusion

Failing to find any predicted value in disasters might be expected in hindsight for three reasons.

1. Although all companies are traded on the New York Stock Exchange and have headquarters located in the United States, they are all multinational corporations. Multinational corporations are unlikely to be substantially impacted by any single disaster. Furthermore, if there are any set of companies that are in a position to mitigate themselves from disaster, it is probably this set of companies. All of this companies are among the largest and most well resourced companies in the world.

2. The second reason that these companies might be unaffected is that they are companies which are largely divorced from whether conditions. It is

hard to imagine that companies such as Microsoft, Visa, or Walt Disney are terribly concerned about extreme weather conditions except in very specific locations.

3. Finally, it is possible that we simply did not have enough data to begin with. After preparation we had 395 examples spanning 14 years. Perhaps with more data we could target disasters by type or location to specific companies rather than the Dow Jones index as a whole. It is easy to imagine this might have more predictive value.

Chapter 8

Quantopian Trading Simulation

We have presented two apparently successful strategies in this report. The first used correlated related financial products to predict the price movement in the Dow Jones Index. The second used analyst opinions to predict the price movement in the component companies of the Dow Jones. These strategies had an estimated accuracy of 74.63% and 67.40% respectively. With accuracies as high as these, it would seem logical then that one could profit their use on the real market.

To test this we will use Quantopian [2]. Quantopian allows users to create trading algorithms and test them rigorously against historical data (a process known as backtesting). Quantopian allows for much more realistic simulation than one might otherwise be able to carry out. Quantopian will correctly model costs a real trading algorithm would incur such as commission charged by the stock broker and slippage. Slippage is a difference between the expected price of a trade and the price the trade actually executes at [4]. Slippage might occur when placing a large order that could shift the market, or when market volatility is high. Quantopian has the added benefit on being able to use minute by minute or daily price. This will be useful when working with overlapping trading hours.

Algorithms are constructed in Quantopian using their web-based python IDE. This is a slow and tedious process which is why Quantopian was not used before this point. The web IDE does not allow the same level of code hinting and inspection that modern tools like IPython Notebooks allow.

8.1 Simulation 1 - Related Assets

The first model we will simulate in a trading environment is the model based on related financial assets.

The only external data that can be imported into the Quantopian web IDE

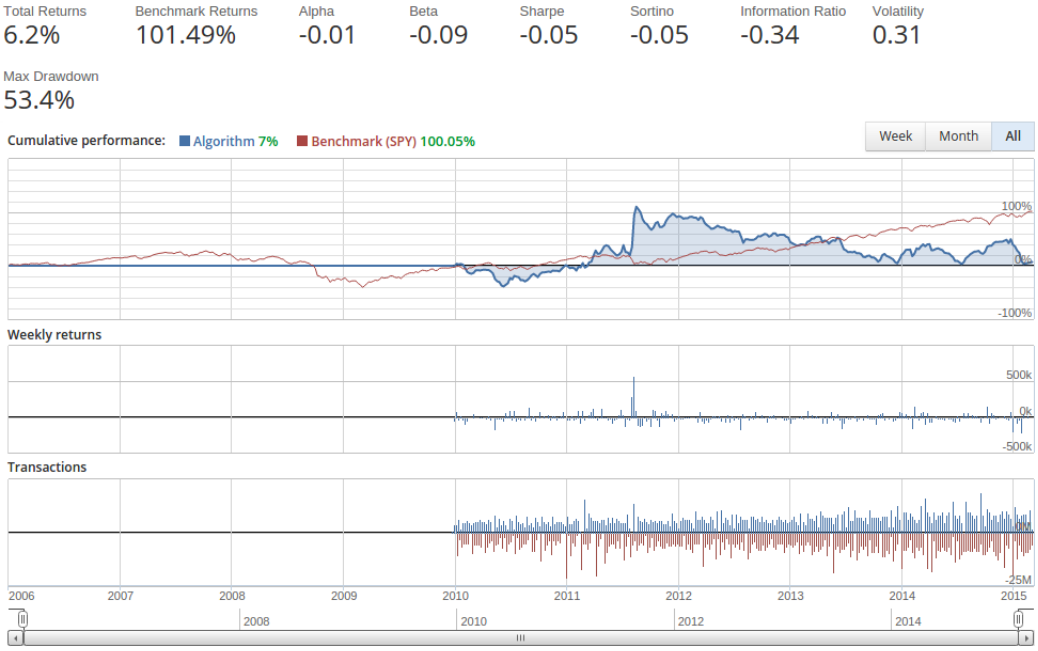
are CSV files. This means that we can not easily import our final fully trained model into Quantopian. Instead, we will take the optimal hyperparameters we learned from the model selection process and train a new model using these hyperparameters for the first half of the backtest period. This isn't optimal, but it is the best that can be done under the circumstances.

An important difference between the training of this model and the training of the model earlier is that this model were given the price of London and German assets at the exact moment the Dow Jones index began trading. The earlier model was given access to the closing prices of the London and German assets. As discussed earlier, because of overlapping trading hours, access to the closing price might have provided more information than the models were entitled to in reality. This risk is mitigated in the Quantopian simulation.

After the training period has been completed, the model will begin to make predictions. Quantopian allows the use of short selling, so that the algorithm can make use of negative predictions. If the prediction is that the price of the Dow Jones index will rise, the algorithm attempts to buy shares in the Dow Jones index. If the prediction is that the price of the Dow Jones index will fall, the algorithm attempts to sell shares in the Dow Jones index. The algorithm issues all orders in the opening minutes of the Dow Jones and liquidates all positions in the final closing minutes of trading. The profit the algorithm makes will be determined by the difference in price at these two points.

The algorithm was simulated from January 1st 2006 to March 2nd 2015. The performance charts generated by Quantopian are presented in figure 8.1. The benchmark performance line in red is the S&P 500 index. This is a general indicator of the performance of a passive buy and hold strategy. The lack of activity by up to the approximate half way point is the training period of the model.

Figure 8.1: Quantopian Simulation 1 Performance



Although the algorithm did have a positive return of about 6%, this is probably due to random chance more than any success of the model. It should also be noted that in the eyes of an investor, any return less than a low risk buy and hold strategy is considered a loss. In the same time our algorithm generated a return of 6%, the S&P 500 generated a return of just over 100%.

Clearly there is a big difference between the performance of the model trained earlier and the one simulated here. There are few differences between the models, except that in the Quantopian simulation the problem of overlapping trading hours was corrected. One must conclude that this is to blame for the difference. It follows then that the model trained earlier scored so highly only because of this error.

8.2 Simulation 2 - Analyst Opinions

The next model that will be simulated is the model based on analyst opinions.

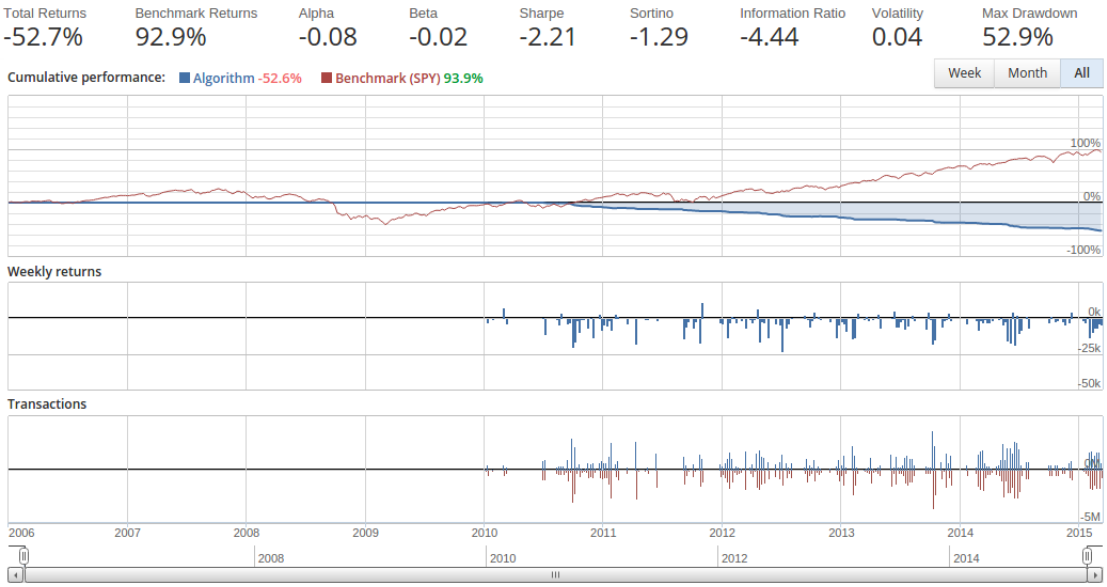
Similarly to the previous section, we can not import the model that was trained earlier because of the limitations of Quantopian. Instead, we must train the model during the backtest.

Trading logic is exactly equivalent to the previous algorithm, but is now applied to individual companies rather than the Dow Jones index. If the prediction is positive for a company, then the algorithm attempts to buy a share in

the company. If the prediction is negative, the algorithm attempts to sell shares in the company. The algorithm issues all orders in the opening minutes of the Dow Jones and liquidates all positions in the final closing minutes of trading.

The algorithm was simulated from January 1st 2006 to December 31st 2014. The performance charts generated by Quantopian are presented in figure 8.2. The lack of activity by up to the approximate half way point is the training period of the model.

Figure 8.2: Quantopian Simulation 1 Performance



With total returns of -52.7%, it is obvious that 8.2 that the model performs poorly in a real world simulation.

We offer two explanations as to why this algorithm performs so poorly.

1. By the time the algorithm places its orders, the market may have already shifted to reflect these analyst opinions. It is plausible that there are many similar traders that have already read these analyst opinions before the market opens. These traders are then ready to immediately place their orders when the market opens. Quantopian offers no guarantees on when an order will be fulfilled. It is therefore possible that by the time our model places an order on the market, the price has already moved to reflect the opinions and they are no longer relevant. The algorithm is then effectively trading on outdated information.

2. Another possible explanation is that there are some traders taking advantage of other traders that are employing a similar strategy. After all the training has been completed, the model ultimately ends up using an

extremely simple strategy; buy when there is a positive opinion and sell on a negative opinion. It does not take a terribly sophisticated model, or trader, to come up with such as strategy. It is therefore credible there are many traders also trying to apply the same strategy. It follows then that there will also be traders trying to take advantage of them. Such traders could artificially shift the market more than the opinion might originally merit, leave time for naive traders (including our model) to buy into the shifted price, and profit by exiting soon after. This would effectively invert any predicted profit of the naive traders.

Chapter 9

Report Conclusion

A large body of work was presented in this report.

Two of the most widely used methods, Fundamental Analysis and Technical Analysis showed little promise in the experiments carried out. Technical Analysis specifically shows little to no potential of ever producing any statistically significant result when the correct methodology is applied.

Machine learning methods were then tested on a wide range of data sources. The result of some models looked hopeful, but ultimately failed when they were put through realistic trading simulations. This highlights that the stock market is prone to differences between theory and practice.

If there is anything that this report shows, it is that profitable stock market prediction is an extremely tough problem. Whether it is possible at all ultimately remains an open question.

After completing the project, it is the firm belief of the author that the only viable trading strategy for a casual investor is a passive buy and hold strategy in index funds and ETFs.

Bibliography

[1] EM-DAT the international disaster database. `http://www.emdat.be/database`. Accessed: 2015-02-26.

[2] Quantopian. `https://www.quantopian.com/`. Accessed: 2015-03-15.

[3] Facebook 22*billionwhatsappdealbuys*10 million in sales.

[4] Investopedia.com. Apr 2015. URL `http://www.investopedia.com/terms/s/slippage.asp`.

[5] Quandl, January 2015. URL `https://www.quandl.com/`.

[6] Michael Cunningham. More than just the kappa coefficient: a program to fully characterize inter-rater reliability between two raters. In *SAS global forum*, pages 242–2009, 2009.

[7] Benjamin Graham, David Le Fevre Dodd, and Sidney Cottle. *Security analysis*. McGraw-Hill New York, 1934.

[8] Online Stock Trading Guide. Head and shoulders pattern, March 2015.

[9] Gerald R Jensen, Robert R Johnson, and Jeffrey M Mercer. New evidence on size and price-to-book effects in stock returns. *Financial Analysts Journal*, 53(6):34–42, 1997.

[10] Yakup Kara, Melek Acar Boyacioglu, and Ömer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert systems with Applications*, 38(5):5311–5319, 2011.

[11] Krzysztof Karpio, Magdalena A Załuska-Kotur, and Arkadiusz Orłowski. Gain–loss asymmetry for emerging stock markets. *Physica A: Statistical Mechanics and its Applications*, 375(2):599–604, 2007.

[12] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[13] Burton Gordon Malkiel. *A random walk down Wall Street: including a life-cycle guide to personal investing*. WW Norton & Company, 1999.

[14] Burton Gordon Malkiel. *A random walk down Wall Street: the time-tested strategy for successful investing*. WW Norton & Company, 2003.

[15] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[16] Salih N Neftci. Naive trading rules in financial markets and wiener-kolmogorov prediction theory: a study of" technical analysis". *Journal of Business*, pages 549–571, 1991.

[17] Stephen O'Grady. The redmonk programming language rankings: January 2015, January 2015. URL `http://redmonk.com/sogrady/2015/01/14/language-rankings-1-15/`.

[18] Alice Schroeder. *The snowball: Warren Buffett and the business of life*. Random House LLC, 2008.

[19] Shunrong Shen, Haomiao Jiang, and Tongda Zhang. Stock market forecasting using machine learning algorithms, 2012.

[20] Wing-Keung Wong, Meher Manzur, and Boon-Kiat Chew. How rewarding is technical analysis? evidence from singapore stock market. *Applied Financial Economics*, 13(7):543–551, 2003.