Stock Market Prediction

Mark Dunne

April 1, 2015

Abstract

In this report we analyse existing, and explore 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 investors decisions. We demonstrate a common flaw in the methodology of Technical Analysis practitioners and show that it produces limited useful information. Based on our findings, an algorithmic trading program is developed and entered into the Quantopian trading competition.

Contents

Chapter 1

Introduction

[todo]

1.1 Project Focus

[todo]

Chapter 2

Data and Tools

2.1 Data Used

For this project, we chose the Dow Jones and its components as a representitive bundle of stocks. The dow jones is a large index traded on the New York stock exchange. It is a prices-weighted index over 30 component companies [todo show calculation]. All companies in the index are large publically traded companies, leaders in each of their own sectors. The index covers many different sectors featuring companies such as Microsoft, Visa, Boeing, and Walt Disney.

We wanted to use a set of companies already picked by someone else so that we don't open ourselves to methodology errors / fishing expeditions to find a set of companies that our algorithms do happen to work for.

The dow jones was chosen because it is well known, and has a relatively small number of compontents when compared to indices such as the S&P 500 which has over 500 components at the time of writing.

This small but representitiive set allowed for a managable dataset given limited resources. Although there were only 30 companies, there was no lack of data to study. To test many of the hypothesis laid out in this report, we were able to extract datasets many thousands of examples in size.

2.2 Data Sources

[todo]

2.3 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 enourmous community behind it. Any problems that might be encountered on the way can be easily solved with a trip to Stack Overflow. Python is amoung the most popular languages on the site which makes it very likely there will be a direct answer to any query [6].

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 such as Haskell. In my view, this is an issue that makes learning to use a new Python package more difficult than it otherwise could be.

Chapter 3

Considerations in approaching the problem

Throughout the project, there are a couple of things that should be kept in mind. All three of these ideas, in their own way, emplore us to keep an open mind in that we might not actually find a profitable way to predict market movements.

3.1 Random Walk Hypothesis

The random walk hypothesis sets out the bleakest view of the predictibility 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 [5]. Famously, he demonstrated that he was able to fool a stock market 'expert' into forcasting 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 forcasting, 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 truely random, there is little point in continuing.

3.1.1 Qualitative Similarity to Random pattern

The stock market can certainly looks random to the eye of a casual observer. To demonstrate this, we generated a random process with similar visual char-

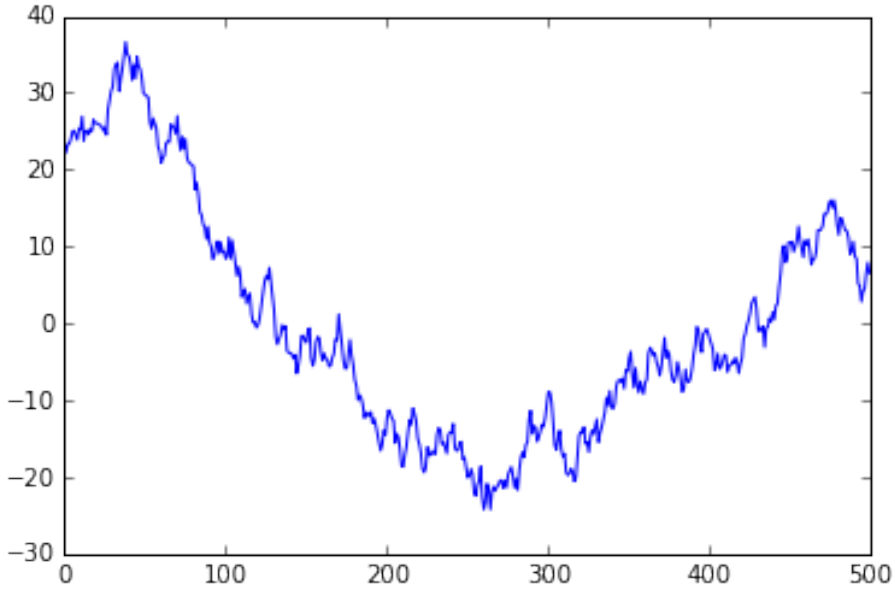
acteristics to the Dow Jones index.

We created a perfectly random process that had striking visual similarity to real stock market data using the following simple formula.

$$a_x = a_{x-1} * \rho + q_x$$
$$b_x = b_{x-1} + \rho * r_x$$
$$f(x) = a_x + b_x$$

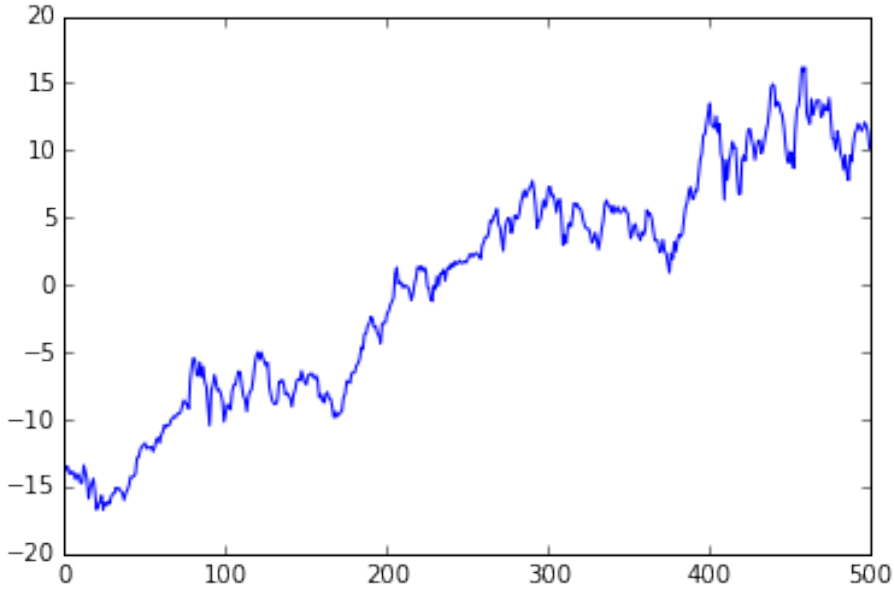
For the best results, ρ was given a value between 0.995 and 1, while q and r are random values taken from a standard normal distribution.

Figure 3.1: Example of random pattern generated



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. We will demonstrate that it is different to random in two ways. In the very next section, we will show that the price itself is fundamentally different to random data, and later we will show that the price is not as random as it may appear when take external variables into account.

3.1.2 Quantitative Difference to Random pattern

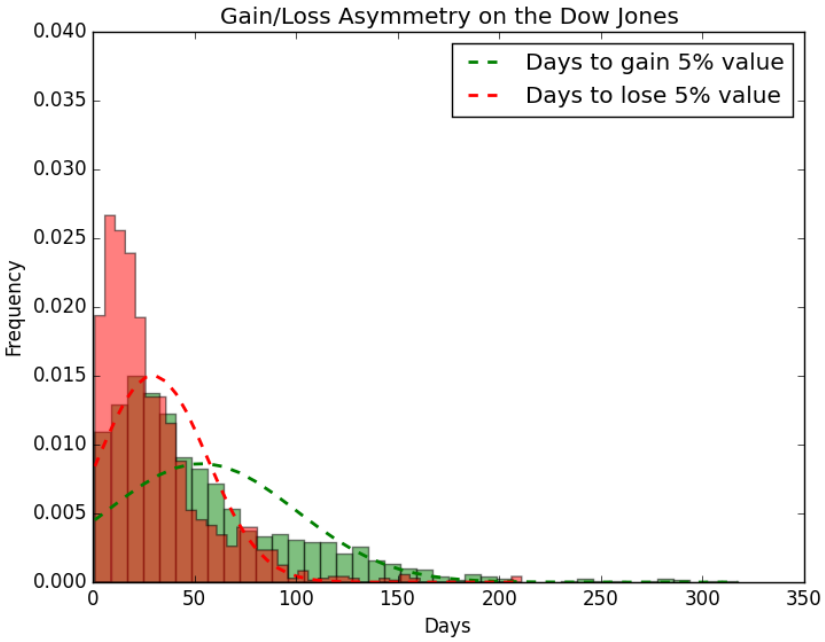
We will first show that the way in which markets move is fundamentally different to the way one would expect them to move if they were random.

Karpio et al. [4] describe an asymmetry between gains and losses on the stock market. Their research looks specifically at indices like the Dow Jones and 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 simulated an investment on the Dow Jones index. We then counted the number of days it took for the

investment 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. The graph shows 2 overlaid histograms detailing how long it took for an investment to lose of gain 5%.

Figure 3.3: Gain-Loss Asymmetry on the Dow Jones



What this graph shows is that the market generally creeps upwards but is prone to sudden drops downwards, and supports the findings described earlier. 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 says that the market is efficient at finding the correct price for the stock market.

It comes in three flavors, however it is still a matter of debate which one, if any, are 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. Later in the project, we will provide results are seem to be inline with this hypothesis but not as good as with the weak form.

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 dimize.

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 erroded away.

This is the same reason for the lack of academic papers on the topic of successfully 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

Chapter 5

Attacking the problem - Fundamental Analysis

The first approach we take at 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. [1] 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 [7]. This gives us some hope, but we should be cautious and remember that the economy 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

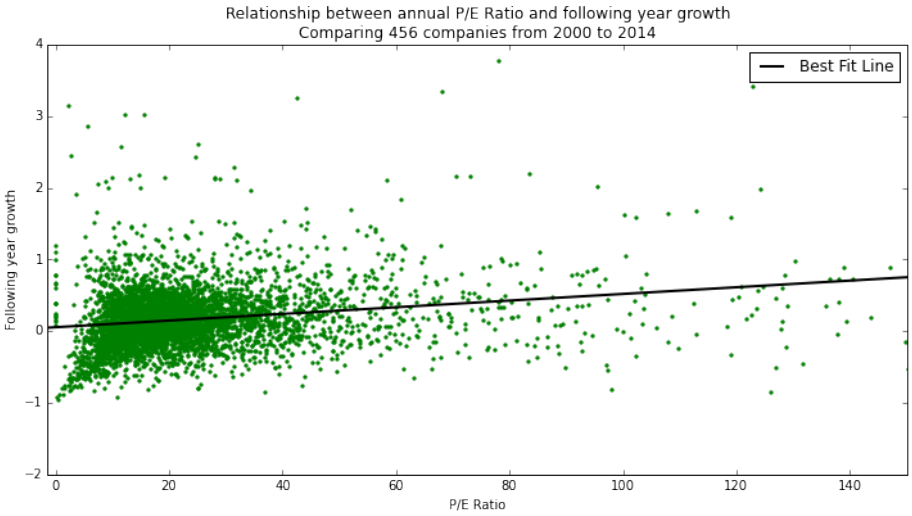
The first metric for the value of a company that we will look at is the Price to Earnings ratio. The price to earnings ratio is calculated as

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

Roughly speaking, what this 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 be a relationship between high P/E ratio and high returns in the following year.

To investigate this relationship, we plotted the P/E ratio for 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. Below is a graph of this relationship.

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

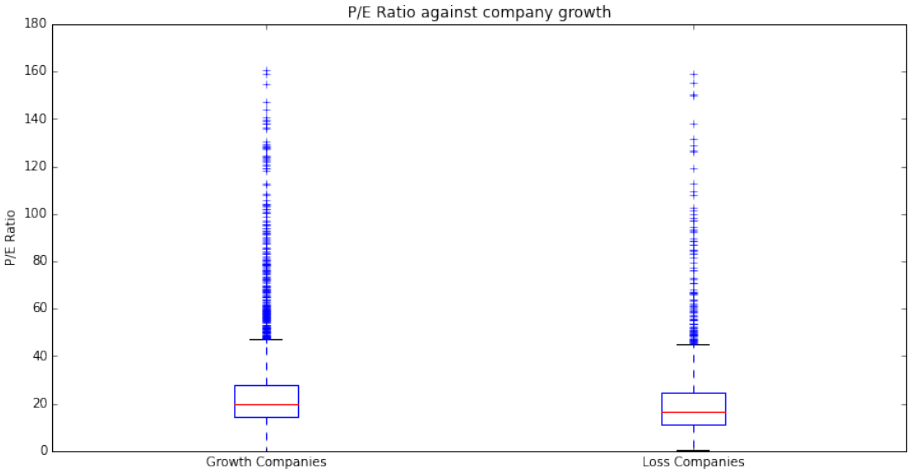


The best fit line was calculated using the standard Least Squares method. If the P/E ratio was indeed predictive, we might have expected a steeper slope in the best fit line, but we can see that there is a very weak correlation at best. It should also be noted that the more we remove outliers, the lower the slope becomes. This indicates that the the line is probably being pulled up by outliers rather than an actual correlation in the data.

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 noticable difference in the P/E Ratio distribution of companies whose share increased, and those whose share price decreased. However, this is not the case. It is clear they the P/E Ratio distribution between these categories is almost identical. We can therefore conclude that the P/E Ratio has little or no 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 is calculated as

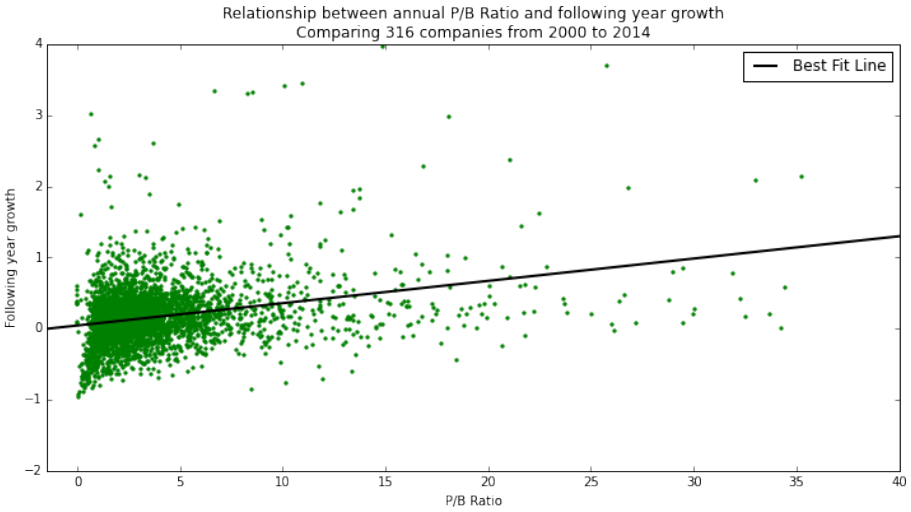
$$\text{P/E Ratio} = \frac{\text{Share Price}}{\text{Book Value of Company}}$$

Informally, what this 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/E 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. Below is a graph of this relationship.

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

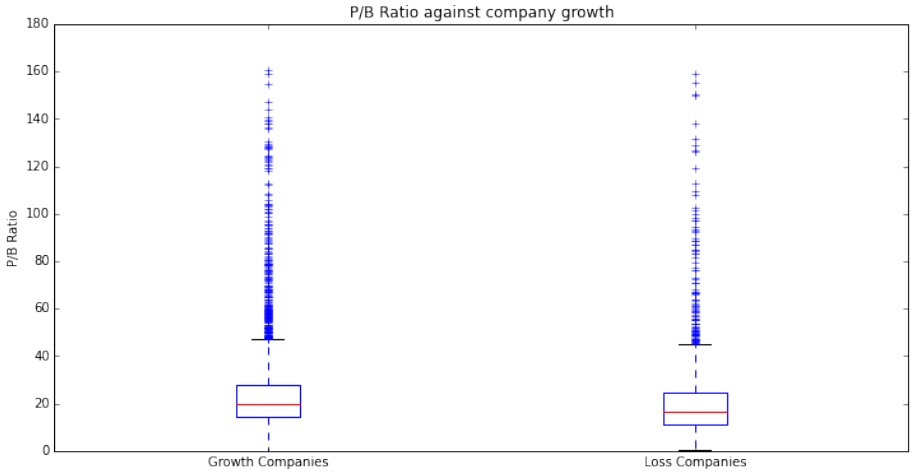


Just as in the P/B diagram, the best fit line was calculated using the standard Least Squares method. Although slope of the best fit line is greater than that of the P/E ratio, this is the opposite of what we might have expected. The data suggests that a high P/B ratio is somewhat predictive of a high growth in the stock price. This is and unexpected result directly opposed to available literature on the subject [3].

One hypothesis is that a high P/B ratio could be a signal of investor confidence like the P/E Ratio and so might be able use the argument about how investor confidence could be a predictor of growth. More likely, we suspect, is that the data used does not control correctly for the expected growth in all companies over the studied time period, from 2000 to 2014. Even accounting for the Great Recession, the stock price of most companies in our dataset did grow from year to year. However, this does not fully explain why the slope of the P/B Ratio best fit line is greater than the slope of the P/E Ratio best fit line. One would have expected at least a lesser slope if the expected slope was an inverse relation.

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 this diagram tells a very similar story to the P/E Ratio diagram. 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 or no predictive value when it comes to estimating company performance for the following year.

5.3 Fundamental Analysis limitations

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

Fundamental Analysis methods does not attempt to capture, and so it difficult to build a software solution to do so. This leaves a large gap in knowledge an algorithm could learn about a company. How should it quantify the value of a brand, the size of its customer base, or a competitive advantage?

These are three examples of some of the many things that a human investor might take into account when deciding who to invest in, but they are untouchable within the scope of this project.

Instead, we are limited to purely quantitative company metrics. We will look at two of the most common metrics, Price to Earnings ratio and Price to Book ratio.

5.4 Fundamental Analysis - Conclustion

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 were not the focus of the project, instead we wanted to focus on predicting daily trends in the market.

Because of these issues that we moved away from Fundamental Analysis.

Chapter 6

Technical Analysis

The second approach we take at solving the problem of market prediction is to use Technical Analysis. This approach tries to recurring patterns and trends within the price of the stock itself.

It should be noted that Technical Analysis goes directly against all forms of the efficient market hypothesis. As explained earlier, even the weak form of the hypothesis rules out prediction using historic price data alone.

Technical Analysis is used for daily level price prediction which was the original focus of this project.

6.1 Broad families of Technical Analysis models

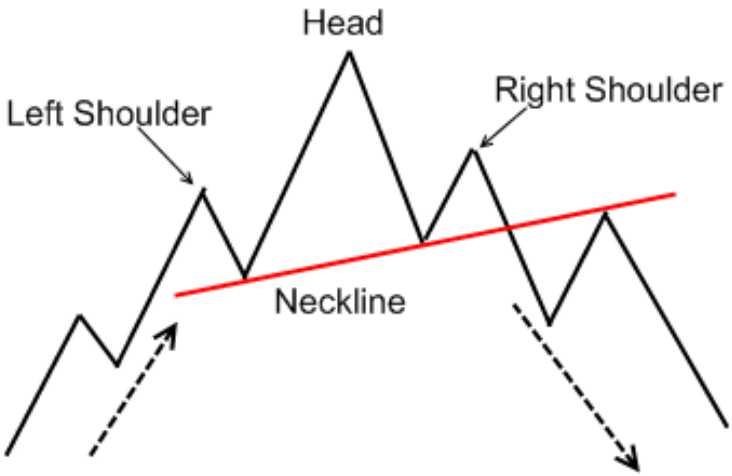
If a casual investor was to do some research into trading on the stock market using Technical Analysis they would encounter two broad categories of models. We will demonstrate that one of these is implausible in theory and in practice, while the other although sound in theory, does not work in practice.

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.

Amoung 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 [2]



The diagram 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, it can be shown that the pattern does not, and indeed cannot, provide 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 occured. 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. If the price were to rise after the right shoulder, it would not be a head and shoulders pattern. A common pattern here is that the investor does not see this as a case where the head and shoulders pattern failed, but instead a case where the head and shoulders pattern didn't exist. This is confirmation bias.

Because of the lack of theorethical support, it is easy to find many additional problems with the head and shoulders pattern. The most obvious one is that because we cannot identify the pattern until after the fact, we can never tell the 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 should of another possible head and shoulders, or even a head peak.

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 authoritive source telling them how to use this pattern, or one like it, to profit on the market.

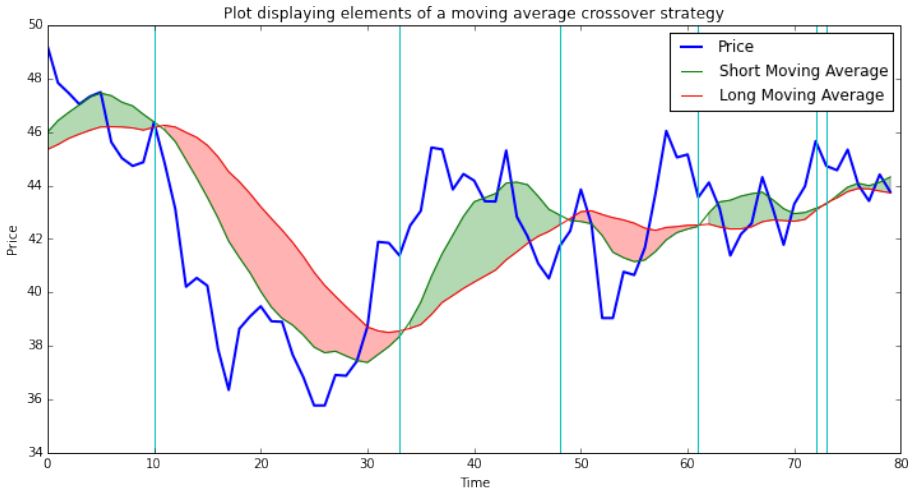
There is no shortage of similar patterns to be found in online literature, but almost all fall into the same problems as the head and shoulders pattern. All information available from these models is only useful in retrospect.

6.3 Statistical trading patterns

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 simpliest and most common model of this type is the Moving Average Crossover strategy.

The moving average crossover strategy relies on the interaction between two moving average calculations. One is a short moving average, and the other is a long moving average. The short moving average might be the mean price for the last 10 days, and the long moving average might be the mean price for the last 20 days for example. When the short moving average crosses under the long, this can be intrepreted as a negative signal that the market is trending downwards. Conversely if the short moving average crosses over the long, this can be intrepreted as a positive signal that the market is trending upwards. The points at 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. This are crossover points just as much as the ones on the left area but these are not as predictive for market trends. We need to choose a 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 (negative and positive) 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, assembled a database of daily prices of all 30 companies in the Dow Jones from the year 2000 to 2014.

This data was then augmented with 49 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 greater increased the time taken to train the model.

The data was split into a training and test set. For simplicity, we divided the data 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. There were over 74,000 data points in the training set and over 38,000 in the test set.

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. Deciding which long and short term periods to use can be seen as equivilent 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 the points between them. For each crossover point, we make a prediction based on its positivity or negativity, and compare tomorrows 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 tomorrows 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.

We can better understand what the model is doing by looking at the confusion matrix.

Figure 6.3: Cross-validation Confusion Matrix for Moving Average Strategy

		Predicted class	
		Gain	**Loss**
Actual class	**Gain**	363	333
	Loss	328	357

We can now put into context how accurate the model is. We can see a slight increase in the numbers along the principle diagonal, indicating correct predictions. This gives us the slightly better than 0.5 accuracy score. But it is also obvious that the model isn't doing much better than a random predictor. We can quantify this using the Kappa Statistic, which compares the model's performance to a random version of itself based on the confusion matrix. This model scores a kappa of 0.0427, which is not significant.

We must conclude that the Moving Average Crossover is not predictive in any meaningful way.

6.3.2 Additional Technical Analysis Models

[todo]

6.4 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 the real world. This problem is prevalent when looking for short and long terms in moving average crossover. What many practitioners appear to do is find the best terms 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 your model to preform well on a single piece of data, but this may not carry over to unseen examples.

Above, we applied the correct methodology. First we split the data into test and training sets, found the best term pair for the training set, and tested that on the test set. 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

6.5 Technical Analysis - Conclusion

It might have been expected that given the popularity of Technical Analysis for stock market trading, that there might 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 teach themselves what to look for, and learn to exploit relationships within the data that we might have otherwise missed using Fundamental Analysis or Technical Analysis.

7.1 Preceeding 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 Aanalysis, 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, but we might be waiting a while for economists to prove that concretely.

The data that we'll we using is the percentage change in closing price of the stock from the preceding 5 days. The dependent variable will be the trend of the 6th day, i.e will the stock price move up or down.

Table 7.1: Data Extract

	day0	day1	day2	day3	day4	outcome	outcome-class
0	0.0492	-0.0029	0.0176	0.0115	0.0028	-0.0142	0
1	-0.0029	0.0176	0.0115	0.0028	-0.0142	-0.0028	0
2	0.0176	0.0115	0.0028	-0.0142	-0.0028	0.0115	1
3	0.0115	0.0028	-0.0142	-0.0028	0.0115	-0.0229	0

Table 7.1 is extract of the first 4 rows in the dataset. Columns labeled day0 to day4 are the percentage change in closing price of the preceding 5 days. We will use the outcome-class as the dependent variable. This column has three classes; 0 represents a drop in the price, 1 represents an increase in the price, and 2 represents no change. This third class is rare, and is ignored for the remainder of the discussion. The dataset that we 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, it the data rows were randomly permuted to remove and 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 a couple of different 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 mechine 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 that we tried got 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 we got here. The nest 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 are not significant. 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, of course, exactly what the Efficient Market Hypothesis predicts.

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 hypothesis 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 hypothesis 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.

One possible reason for the failure of our models in the previous section is that the models we tried were not sufficiently complex. 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 simplier 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 hypotheses, but a $n - NN$ model can only produce one hypothesis.

Figure 7.1: kNN Validation Curve

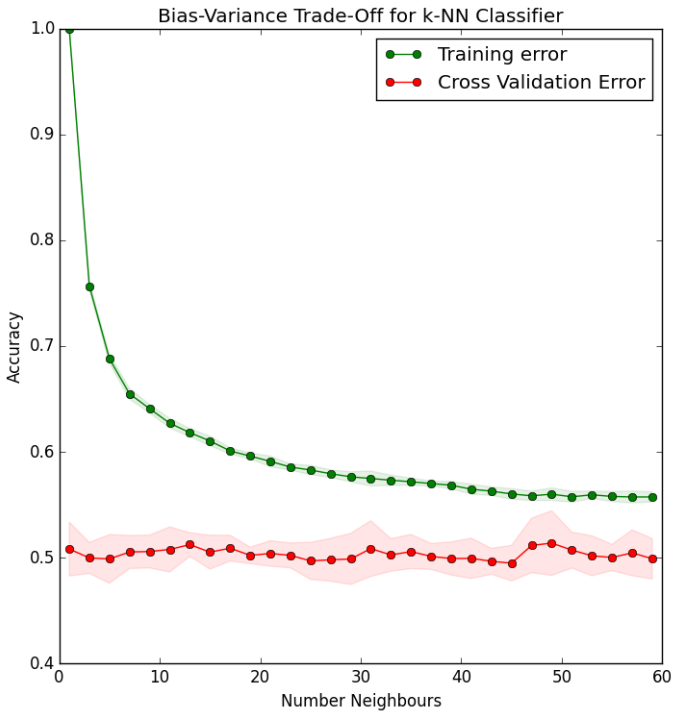


Figure 7.1 plots the validation curve for the kNN model. To generate this graph, we generated 60 kNN models with different k values. The training and test error for each value of k was then plotted. Since the complexity of the

model varys with k as explained above, we can interpret the graph as being a performance comparison of models of varying complexity. Normally in such a graph we would expect the training and test errors to converge, but this is clearly not happening for 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.

Figure 7.2: kNN Learning Curve

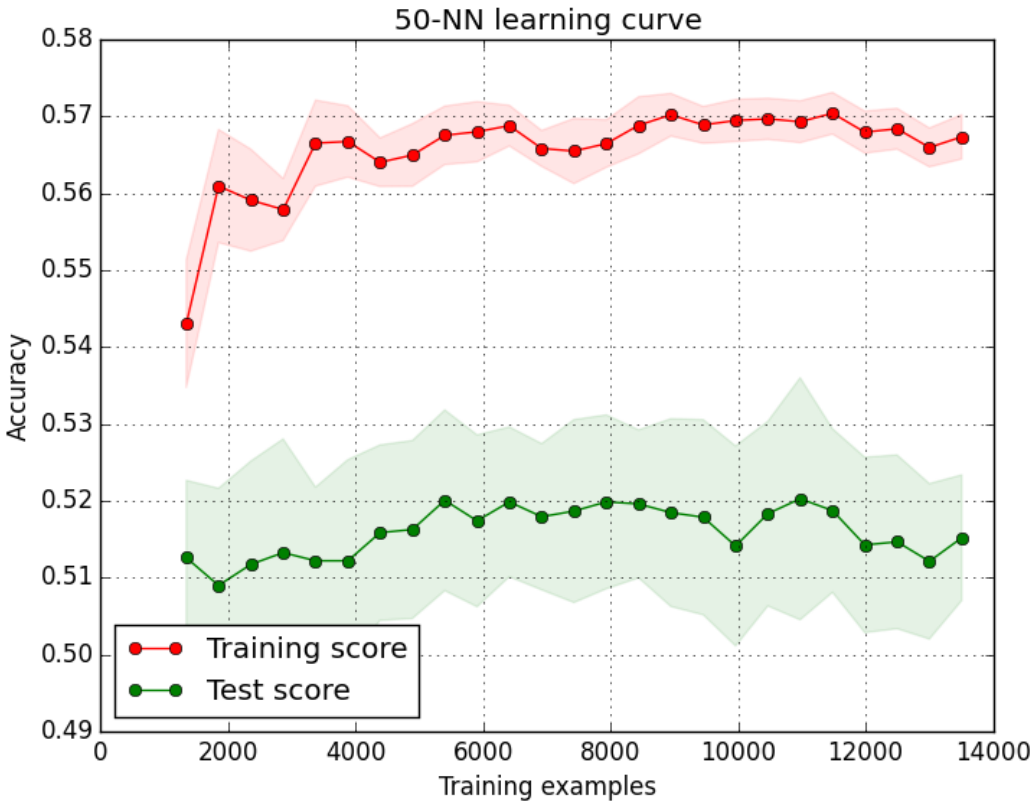


Figure 7.2 shows the effect of varying the number of examples on the model. Normally in a Learning Curve graph, you would expect the test and training errors to converge. The intuition is that as the model is presented with more data, it is better equiped to come up with better, more general, hypotheses. 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 the the problem must lie in the data itself. It would seem that the preceeding 5 day trends 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. Prehaps 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 yesterdays prices appear to have no influence on todays 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 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 as trending heavily in one directly, 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 prediction of our the trend in 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 then which we can cleanly partition 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 then forced to accept 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. This is opening us to problems of leakage, where the test set can influence the training set. It is concievable that the trend in the New York Stock Exchange can effect the trend in the London Stock Exchange, even though it only opens in the final 4 trading hours. Due to this issue we must be cautious and suspecious positive results moving forward. It is unfortunate that later in the report, when we implent 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.

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

DAX The German DAX index is essentially equivilent 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 Similar to the Dow Jones and the German DAX. Traded on the London Stock exchange, its components are a selection of 100 large UK based companies. Commonly known as the FTSE100

N225 The Nikkei 225 has 225 large Japanese companies as components as 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. It is itself traded on the Shanghai Stock Exchange.

AUD The Australian Dollar to US Dollar exchange rate

EUR The Euro to US Dollar exchange rate

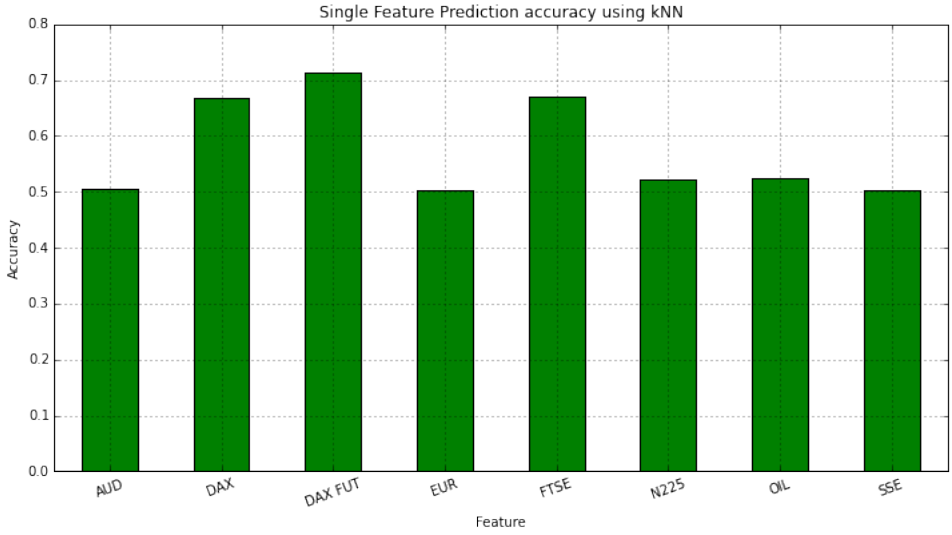
OIL Brent Crude Oil futures prices

A decision was made to use the price of the Dow Jones index rather than the price of the components, or companies, themselves as the dependent variable. This was due to the fact that the features listed above are not specific to any one company, although it could be argued that some features might influence some companies more than others. Determining the predictive value for each feature for each company would have made modeling significantly the problem more complex and computationally time consuming. For simplicity, the price of the Dow Jones index was used as the dependent variable in this section.

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 the graph 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.

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 hyperparameter 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 Specifc 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 neighbors the model should examine to make a prediction. We can then weigh those neighbors 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 nighbours 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 were 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 precisely 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 publically available intraday price data, we might have some leakage of the test set that is skewing 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 it would not work as well in the real world.

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, may 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 that these recommendations might truely 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 concievable 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 predict. For whatever reason, if these opinions can predictive 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 equivilent 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: 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

Table 7.5 Displays what the gathered data looks like.

It should be noted that in this section we will be trying to predict the daily movement of individual companies prices rather than the Dow Jones index as a whole as we did in the last section. This is because each opinion very explicitly relates 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 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 counting the number of Upgrades and Downgrades. Finally, the aggregated opinions were merged with the INTC stock price. This data was then plotted.

Figure 7.4: Visualisation of analyst opinions and INTC price

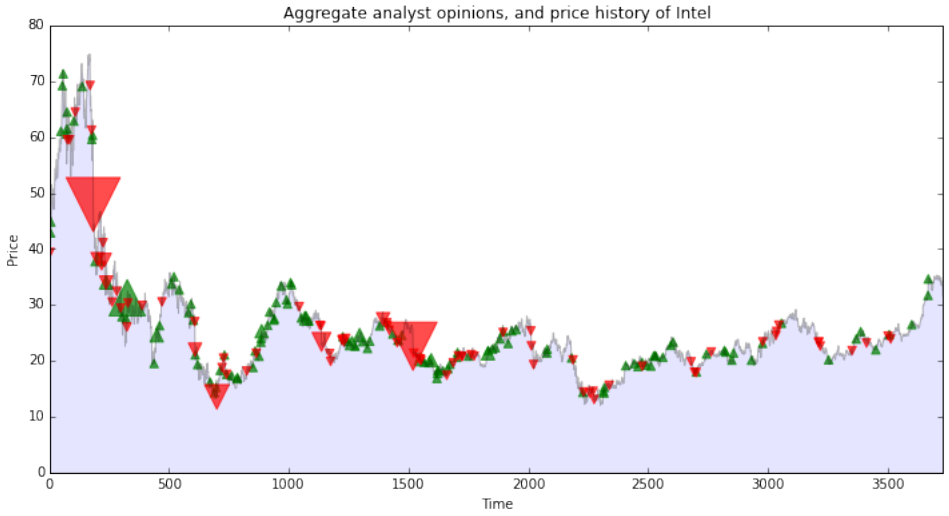


Figure 7.4 visualises the relationship between opinion sentiment (Upgrade or Downgrade) and price. Green upwards pointing triangles signal an upgrade and red downwards pointing triangles signal a downgrade. The size of the triangle represents the number of coinciding opinions across all research firms. For instance the large red down pointing triangle on the left hand side of the diagrm is a point representing 9 separate research firms all downgrading their opinion of INTC on the same day.

From visual inspection it would appear there may indeed 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 problem, 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 features are *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 accross 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 a using the Pandas python library.

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 were the hyperparameter 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

MultinomialNB being the model that won in the Error Estimation round, it should now be the model that we focus on in 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.

At the same time as we are searching over the model hyperparameter, we will again be searching for the best k features to keep.

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 leakage problems the data in the last section had, we will unfortunately see that we cannot profitably trade on the stock market using this model either.

7.4 Disasters

[todo]

Bibliography

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

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

[3] 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.

[4] 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.

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

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

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