Stock Market Prediction

Mark Dunne

January 29, 2015

Contents

Abstract

This project is an attempt to answer the question of whether stock market prices can be accurately predicted using modern Artificial Techniques. We take multiple approaches to the problem including Fundamental Analysis, Technical Analysis, and Machine Learning techniques.

0.1 Introduction

0.1.1 Introduction

0.1.2 Project motivation

0.1.3 Importance and prevalence of the problem

Explain size of the financial industry affected by the problem and why having better tools to predict the stock market may be important to the average person.

0.1.4 Project goals

0.1.5 Project outline

How the project will meet those goals

0.2 Background

0.2.1 The Stock Market

What is the stock market

How the stock market works

Components of the stock market

Terminology

Fundamental analysis

The Efficient Market Hypothesis

Throughout this area of investigation, there is a very large elephant in the room, the Efficient Market Hypothesis (EMH). The hypothesis talks specifically about an agents ability to profit from make inefficiencies, i.e when stocks and shares are mispriced by the market. Its strongest proponents would claim that the very title of this report, predicting stock market, is all but impossible. The EMH comes in three main forms. [2]

- The weak form of the efficient market hypothesis claims that prices fully reflect the information implicit in the sequence of past prices.
- The semi-strong form of the hypothesis asserts that prices reflect all relevant information that is publicly available
- The strong form of market efficiency asserts information that is known to any participant is reflected in market prices.

Informally, the weak form implies that you cannot profit using historical patterns in the share price, the semi-strong form implies that there is profit only to be made from insider trading, and the strong form says that even this is futile. Important to note is that the week form the the hypothesis does not completely rule out profitable trading on the stock market. Trading profitably based on predictions from Fundamental analysis is still possible in the week form.

Clearly for the successful application of modern AI techniques to this problem, we must hope that at least the semi-strong and strong forms of the hypothesis are wrong and do allow for a sufficiently intelligent agent to profit. Luckily, many researchers do indeed question the validity of the hypothesis. There is evidence that the stock market does not always follow EMH. Basu [1] showed that fundamental analysis could yeild information useful in future market forecasts. This result questions the semi-strong and strong forms of the hypothesis, but does not necessarily break the weak form.

Later in this report, we provide our own data and analysis that supports the view that at most only the week form of the EMH holds true.

0.2.2 Analysis of the problem

Explanation of the difficulty of the problem

Separation of profitability and accuracy

Temporal reach of prediction

Want to avoid HFT

Fundamental is long term

Machine learning focus on short/medium

Formal definition of the problem

0.2.3 Review of existing work

0.3 Methodology and Data

0.3.1 Tools Used

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 [4].

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.
The following packages were used throughout the project

Numpy A staple of scientific computing, useful for efficient data structures and linear algebra.

Pandas Useful for filtering and mutating the data into a desired form.

SciPy Extensive library of numerical routines

Matplotlib Used for graphing data

Quantopian/Zipline and Pyalgotrade

Statsmodels

0.3.2 Data Used

Data sources

Format of the data

Adjusted prices

0.3.3 Simulation of strategies

Similarity to real life

0.3.4 Defining a successful model

Statistical significance of a model

0.4 Attacking the problem - Fundamental Analysis

We begin by approaching the problem using Fundamental Analysis.

Fundamental Analysis of stocks and shares is one of the earliest and forms of market prediction. It takes the view that the market has mispriced a security, but over time the price will be corrected to its intrinsic value. If we can accurately calculate the intrinsic value of a security, e.g how much is one share of company X actually worth, then we can choose to invest based on the difference between the current price and intrinsic value.

Graham et al. [3] 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. [6]

0.4.1 P/E Ratio

One of the simplest and possible most well known approach to gauging the intrinsic value of a stock is to use the P/E Ratio. P/E Ratio is defined as follows

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

The intuition on why this metric might be used is that if the ratio is high, it means that investors are willing to pay more for every dollar the company earns; they have more faith in the company. For example, if a company is trading at \$20, and the Earnings Per Share is \$100, then the P/E Ratio is 5. This translates to the investor willing to pay \$5 for every dollar of the company's earnings.Although simplistic in nature, we were able to achieve surprising results based on this ratio.

We analysed 497 companies in the S&P index from 1990 to 2014 and compared their P/E Ratio to the change in their stock price the following year. If the P/E Ratio is indicative of future performance, we should see larger, positive, changes in the stock price of companies with a high PE Ratio compared to those with a lower PE Ratio.

Before cleaning the data, there are large anomalies. These are not incorrect data points, but we should not include them if we want to understand the market in general. Figure 1 plots the data before cleaning and analysis.

To control for anomalies, the data was filtered aggressively to look only at data with one standard deviation of the mean on both axis. After controlling for anomalies, the data is partitioned into high and low P/E ratios. High P/E Ratios are defined as those above one standard deviation away from the mean P/E Ratio, and low P/E ratios are the complement.

We found that the groups differed by an average of 12.61%. A T-test was carried out to test the statistical significance of the finding. The results of the T-test gave a t-statistic of 1.8825 and a corresponding p-value of 0.06. This is

Figure 1: Graph with anomalies present. Some companies show annual gains in excess of 700%

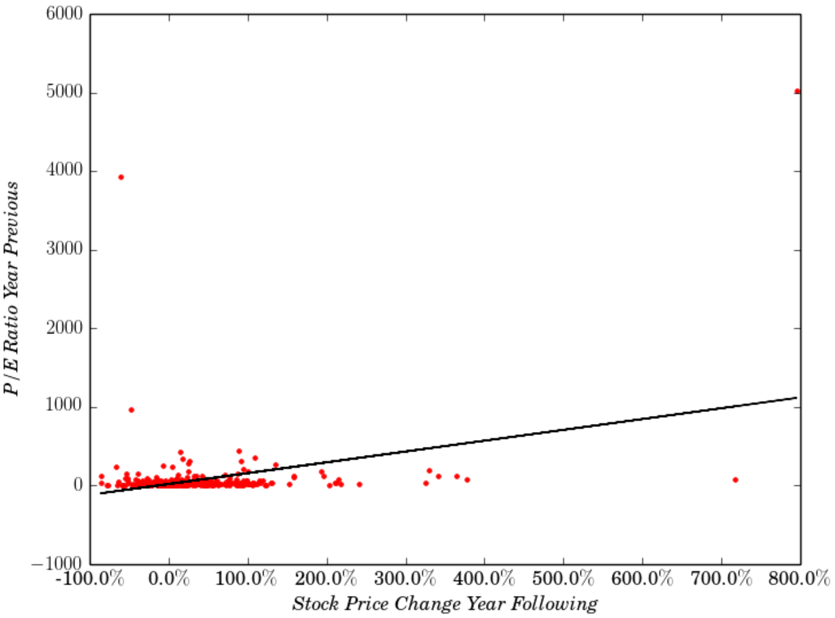
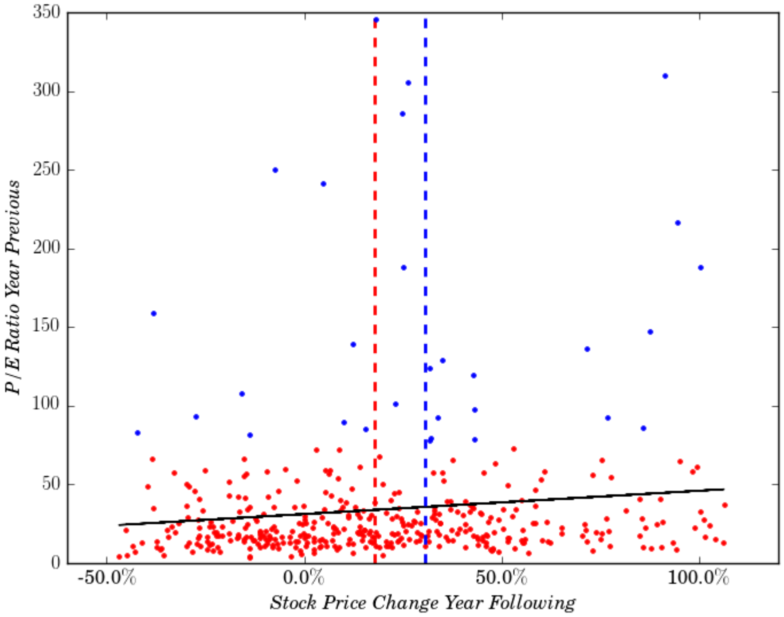


Figure 2: Filtered Data. Red dots are within one standard deviation σ of mean P. Dashed lines indicate the mean of their respective groups

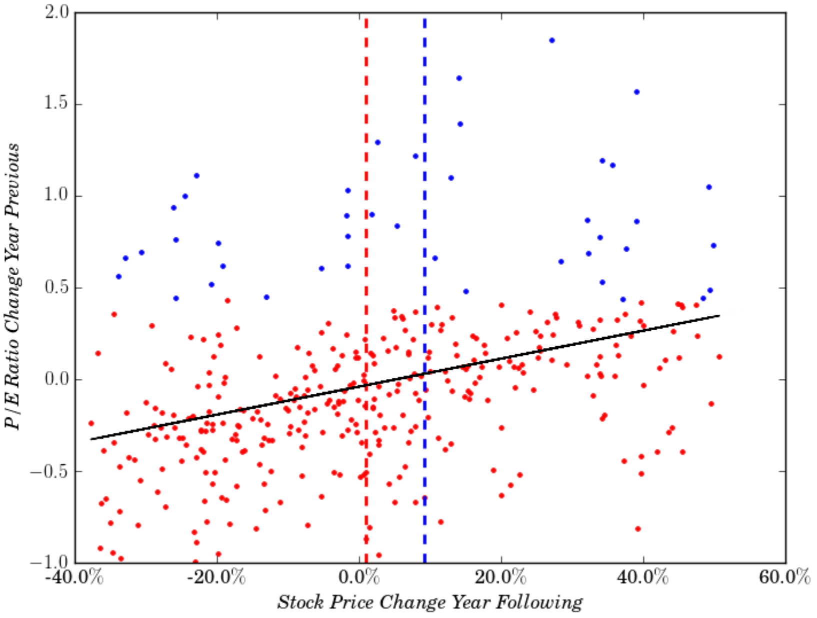


very close to the standard threshold of statistical, 0.05. Figure 2 plots the data after cleaning and analysis.

In an effort to cross the statistically significant threshold, a different experiment was proposed. Instead of looking at the absolute P/E Ratio, we look at the change from the previous year. The intuition builds on that of the original P/E Ratio. If the P/E Ratio has increased year on year, we may be able to conclude that investor confidence has increased. In a similar manner to the previous method, we partition the dataset into high and low P/E Ratio changes based on standard deviation from the mean. Figure 3 plots the dataset for this method.

With the new method, we found that the groups differed by an average of 8.2%. The results of the T-test gave a t-statistic of 2.246 and a corresponding p-value of 0.025. This indicated that the result is statistically significant, and the null hypothesis (i.e that there was no information to be gained from the P/E Ratio) can be rejected.

Figure 3: Using change in P/E Ratio year on year



0.5 Attacking the problem - Technical Analysis

0.5.1 Hobbyist Approaches

0.5.2 Review of Metrics

0.5.3 OLMAR algorithm

0.5.4 StatsModels

0.6 Attacking the problem - Machine Learning

0.6.1 KNN on metrics

[5]

Bibliography

[1] Sanjoy Basu. The relationship between earnings' yield, market value and return for nyse common stocks: Further evidence. *Journal of financial economics*, 12(1):129–156, 1983.

[2] Elroy Dimson and Massoud Mussavian. A brief history of market efficiency. *European financial management*, 4(1):91–103, 1998.

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

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

[5] pybrain.org. Classification with feed-forward neural networks, January 2015. URL `http://pybrain.org/docs/tutorial/fnn.html`.

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