The Stock Market

Who Should You Listen To?

1. Introduction

Anyone who can accurately predict the changes in the stock market is bound to make a large fortune. Therefore, many people have put huge amounts of time and resources into trying to understand the behaviour of the market.

This report tries to extract the expected fluctuations of these stocks, not from the market itself, but rather from the experts who have already put so much study into the market.

To do this, various data aggregation algorithms are used, which look at filtering out malicious information and considers the track record of various experts in the field. The processing leads to a final numerical score for each stock, at given periods of time.

Being able to talk and speculate about the behaviour of algorithms is great, but what's even better, is being able to apply them to real-world data. Not only can the results help prove/disprove theories, but they can also help guide the development of an algorithm. Therefore, this report takes the approach of starting with basic concepts and builds upon them by letting the test results drive the development.

2. Summary of the Data

The data used to test the algorithms and theories comes from 'Thomson-Reuter's Institutional Broker's Estimate System' [1]. This is a huge database of financial information and evaluations, for a very large number of companies.[1]

Specifically, we will look at the recommendations given by analysts for the stocks that make up the Standard & Poor's 500 Index [2]. The choice of using these stocks comes down to their overall reflection of the U.S. market and the potential for using index figures in the following algorithms.

The recommendations themselves have been reduced to a value from 1 to 5:

1) Strong buy
2) Buy
3) Hold
4) Underperform
5) Sell

3. Initial Data Processing

Although a scale of 1 to 5 is sensible, a more useful strategy is to reduce the scale to values between -1 and 1 (for reasons we will see in the following sections). The value of -1 represents sell, 0 represents hold and 1 represents strong buy.

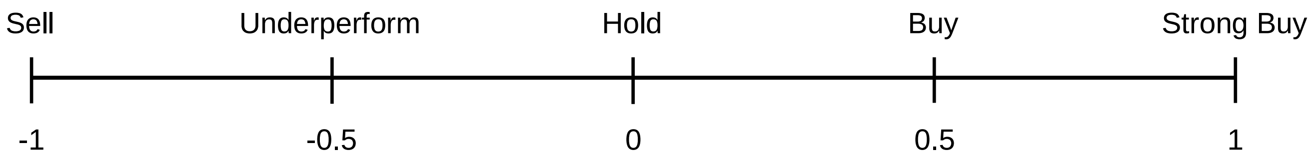


Figure 3.1 – New recommendation scale from -1 to 1.

To adhere to this new scale, all ratings were adjusted with a simple formula:

$$R_{new} = \frac{3 - R_{old}}{2}$$

Where R_{old} refers to the '1 to 5' ratings and R_{new} refers to the '-1 to 1' ratings.

Now that we have an understanding for what the data looks like, we can move on to the applications.

Note – Code for adjusting the data as described in this section is found at:
https://github.com/lp1719/Stock-Analysis/blob/master/init_scripts/rescale_rec.py

[1] *The actual data comes from the 'Wharton Research Data Services' [3].*

4. Testing Algorithms

Since we aim to let the test results dictate our focus, we must first define a suitable testing method. To do this, we must first answer some important questions.

'What will the output of our algorithms look like?'

In its simplest form, each stock will receive an adjusted rating value within the scale seen in *figure 3.1*. However, the rating of a stock will fluctuate over time due to changes in a company, changes to the current political/economic environment and many other factors.

Therefore, ratings are represented in a matrix $R(i,j)$, where the i^{th} stock is given a rating in the j^{th} period of time.

Periods are distributed equally along a fixed time frame. For example, if we consider the ratings of stocks within the time frame of a single year:

$$\begin{aligned} 1\, period \ &= \ evaluating\ annually \\ 2\, periods \ &= \ evaluating\ bi - annually \\ 4\, periods \ &= \ evaluating\ quarterly \\ 12\, periods \ &= \ evaluating\ monthly \\ 52\, periods \ &= \ evaluating\ weekly \end{aligned}$$

Now that we understand what the results will look like, let us consider the following.

'How can we validate the quality of these results?'

By considering that the output of the algorithms would traditionally be used for investment in the real world, let us simulate the potential profits/losses, when using this new data. We can then visualize the portfolio value change with respect to time, and so get a good representation of the success/failures of the algorithms.

'How will simulation occur?'

To answer this question, let us more clearly define the problem, it's results and then describe the processes.

We start by creating a new matrix, which stores the number of shares that our simulated investor owns, at the different periods of time. Namely, $S(i,j)$, which is the number of shares our investor has in the i^{th} company during the j^{th} period.

If we do not give our simulated investor any shares at the start, there is no way to evaluate selling recommendations if no buying has occurred previously. Therefore, we will give our investor \$10 000 worth of each stock initially.

Now consider that the initial price of stocks A, B and C are as follows:

$$\begin{aligned} Stock\ A &= \$100.00\ per\ share \\ Stock\ B &= \$450.00\ per\ share \\ Stock\ C &= \$925.00\ per\ share \end{aligned}$$

An integer number of shares would give our simulator:

$$\begin{aligned} Stock\ A \rightarrow\ 100\ shares &= \$10\ 000\ value \\ Stock\ B \rightarrow\ 22\ shares &= \$9\ 900\ value \\ Stock\ C \rightarrow\ 10\ shares &= \$9\ 250\ value \end{aligned}$$

This means that $850 in value has been lost, leading to an unfair investment distribution. Therefore, we will allow for fractional amounts of shares to avoid these errors.

Although this is not quite realistic, keep in mind that the purpose of the simulation is to test our algorithm development.

After completion of the testing simulation, two vectors are produced:

- $I(j)$ – The total amount invested at the j^{th} period
- $V(j)$ – The total net value at the j^{th} period

$I(j)$ is easily calculated as the buying strategies described below keep track of it. However, $V(j)$ is a bit more complicated to calculate:

$$V(j) = M(j) + \sum_{i=1}^{n} S(i,j)\ P(i,j)$$

The new variables are:

- $M(j)$ – the amount of money not invested in the stock market at period j. This comes from selling shares and the formula for calculating this value is shown a bit later.
- n – the total number of stocks.
- $P(i,j)$ – the value of the i^{th} stock at the j^{th} period of time.

From $I(j)$ and $V(j)$, we calculate the change in value, as a percentage of the amount invested:

$$C(j) = 100\left(\frac{V(j)}{I(j)} - 1\right)$$

For example:

$$\begin{aligned} C(1) = -10\ &\rightarrow\ indicates\ a\ 10\%\ loss\ after\ the\ first\ period. \\ C(3) = 120\ &\rightarrow\ indicates\ a\ 120\%\ profit\ after\ the\ third\ period. \end{aligned}$$

Now we must consider how $S(i,j)$ is calculated.

There are three investment strategies described below, but what changes between them is the way they buy shares, not the way they sell them (we use three strategies to give a better overall picture of how our data aggregation has performed).

Selling occurs when a negative recommendation is given. The recommendation value itself is the percentage of that stock to be sold. For example:

$$\begin{aligned}&If\ the\ recommendation\ is\text{:}\ \ -1.0 \rightarrow\ sell\ 100\%\ of\ the\ shares\\&If\ the\ recommendation\ is\text{:}\ -0.5\ \rightarrow\ sell\ 50\%\ of\ the\ shares\\&If\ the\ recommendation\ is\text{:}\ -\ 0.1 \rightarrow sell\ 10\%\ of\ the\ shares\end{aligned}$$

This can be translated to the following formula:

$$\begin{aligned}S(i,j) &= S(i,j-1) + S(i,j-1)\ R(i,j)\\&= S(i,j-1)\ [1 +\ R(i,j)] \qquad\qquad for\ R(i,j) < 0\end{aligned}$$

With sample values:

$$\begin{aligned}&Let\\&S(i,j-1) = 100 \quad (number\ of\ shares\ previously\ owned)\\&\quad\ R(i,j) = -0.2 \quad (the\ current\ recommendation)\\&\\&Then\\&\quad S(i,j) = 100[1 + (-0.2)]\\&\qquad\qquad = 100(0.8)\\&\qquad\qquad = 80\\&\qquad\qquad \therefore The\ investor\ now\ has\ 80\ shares\ in\ the\ i^{th}\ stock\end{aligned}$$

At this point, we can also calculate, $M(j)$, which is the money not invested in the stock market at period j:

$$\begin{aligned}&M(0) = 0\\&M(j) = M(j-1) + \sum_{i\,:\,R(i,j)\,<\,0} |\,S(i,j-1)\ R(i,j)\ P(i,j)\,| \quad for\ \ j > 0\end{aligned}$$

In words, we add the previous amount of money that is not invested and the amount that is gained from selling shares at the current period.

One might also consider that we can apply the formula for calculating $S(i,j)$ when buying shares. This would mean that a consistently high rated stock would have more shares bought than a fluctuating rated stock. This sounds reasonable.

However, consider that a company has a bad media report and is predicted to have a massive fall in share prices. Hence, all the recommendations tell you to sell and that is what you do. After a couple of months, the company is on the upswing again, and all the recommendations are now telling you to buy. However, since you have no shares left, $S(i,j-1)\ R(i,\mathrm{j})$ will always evaluate to zero and so you are unable to buy back any shares ($Since\ S(i,j-1) = 0$).

Therefore, we now consider three buying strategies for when $R(i,j) > 0$, some of which use the total sum of positive ratings:

$$T_R = \sum_{k\,:\,R(k,j)\,>\,0} R(k,j)$$

Single Fixed Investment

After the initial amount that is invested in each stock, the simulator only uses the money gained from selling shares for reinvestment into the stock market:

$$S(i,j) = S(i,j-1) + \frac{M(j)}{P(i,j)}.\frac{R(i,j)}{T_R}$$

Breaking down the formula:
$S(i,j-1)$ refers to the previous shares that the simulator owns, which should remain. $M(j)$ is the uninvested money being used to buy new shares, which is divided by the current price of each share $P(i,j)$. The result is the maximum number of shares that could be bought with this money. However, not necessarily all the money is going into this stock, so we multiply the number of shares that can be bought by the fraction $\frac{R(i,j)}{T_R}$. This means that higher rated stocks will produce a larger fractional value and so get a bigger slice of the money to go towards their shares.

Distributed Periodic Investment

For each investment period, the simulator adds another \$1 000 000 worth of investment, distributed among stocks that are worth buying:

$$S(i,j) = S(i,j-1) + \frac{1000000}{P(i,j)}.\frac{R(i,j)}{T_R}$$

Breaking down the formula:
The idea behind the formula is the same as the 'Single Fixed Investment' formula. However, this time \$1 000 000 is distributed rather than the money from selling the shares.

Fixed Periodic Investment

For each investment period, the simulator is willing to buy up to \$10 000 worth of each stock. The actual amount bought is the maximum potential of \$10 000, multiplied by the recommendation itself:

$$S(i,j) \ = S(i,j-1) + \frac{10\,000.R(i,j)}{P(i,j)}$$

Breaking down the formula:
Once again, we add the previous number of shares $S(i,j-1)$. Since the rating is a value between 0 and 1, it can be treated as a percentage of how much of a fixed limit should be bought. A highly recommended stock will have a value close to 1, hence close to 100% of the maximum limit will be spent. Similarly, a lower rated stock will get less of the maximum limit invested in it. The amount of money being spent is $10\,000.R(i,j)$ and dividing by $P(i,j)$ leaves the number of shares that are bought.

Note – The code that implements the simulation described in this section is found at:
https://github.com/lp1719/Stock-Analysis/blob/master/test_profits.py

5. Baseline Case - Simple Average

Now that we have our testing set-up, we can look at creating a baseline result, for which we aim to improve upon. 'Thomson-Reuter's Institutional Broker's Estimate System' comes with its own averaging of recommendations, in one month periods. The averages are calculated as follows:

$$R(i,j) = \sum_{A_k:\, A_k \to (i,j)} \frac{A_k(i,\ j)}{|\{A_l : A_l \to (i,j)\}|} \qquad \text{(R. 1)}$$

The variables are as follows:

- $A_k \to (i,j)$ indicates that analyst A_k has rated the i^{th} stock for the j^{th} period of time.
- $A_k(i,\ j)$ refers to analyst $A_k's$ recommendation of the i^{th} stock for the j^{th} period.

Hence, this is the usual way of calculating the mean of results, i.e. the sum of all the ratings is divided by the total number of experts who gave the ratings.

We will now see how this performs with the testing algorithms described earlier. The simulation will occur over the year of 2016, divided into monthly periods. We will also overlay the effect of having the initial stocks and not buying/selling, which is indication of how the entire market was performing.

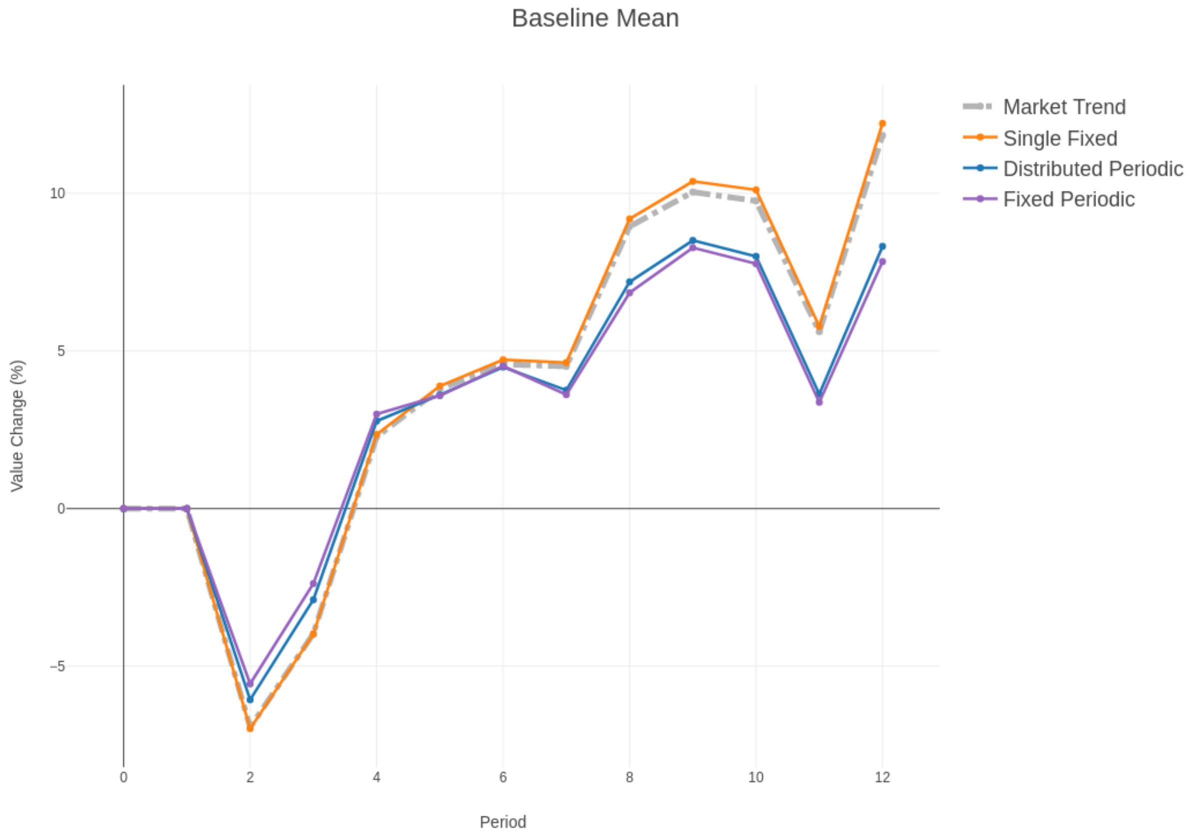


Figure 5.1 - Graph of the portfolio value change for buying and selling S&P 500 stocks using mean recommendations. The periods represent the months in the year 2016, with months 0/1 being January and month 12 being December.

S\P	1	2	3	4	5	6	7	8	9	10	11	12
M.T	0	-6.9	-4.0	2.3	3.8	4.6	4.5	8.9	10.0	9.8	5.6	11.8
S.F	0	-7.0	-4.0	2.4	3.9	4.7	4.6	9.2	10.4	10.1	5.8	12.2
D.P	0	-6.1	-2.9	2.8	3.6	4.5	3.8	7.2	8.5	8.0	3.6	8.3
F.P	0	-5.6	-2.4	3.0	3.6	4.5	3.6	6.8	8.3	7.8	3.4	7.8

Figure 5.2 - Table of the portfolio value percentage changes when buying and selling S&P 500 stocks using mean recommendations. The top row is the number of months into 2016 that the calculations were made.

Looking at *figure 5.1*, we can already get a valuable insight with how a simple averaging of recommendations can allow for portfolio value changes. Interestingly, all the investment strategies follow the trend of the market itself. However, the 'single fixed investment' manages to do even better than the market trend. From *figure 5.2*, we can see that there is exactly a 0.4% profit increase.

Both the 'Distributed Periodic' and 'Fixed Periodic' have significantly worse results. However, this could be due to more money being introduced into the market and so the newly invested money will not have the same period to increase in value, thereby lowering the overall percentage increase.

The fact that one year is not enough time to allow for potential divergence away from the market, means that our next step is to increase the number of periods. We will now apply the same testing over a ten-year period:

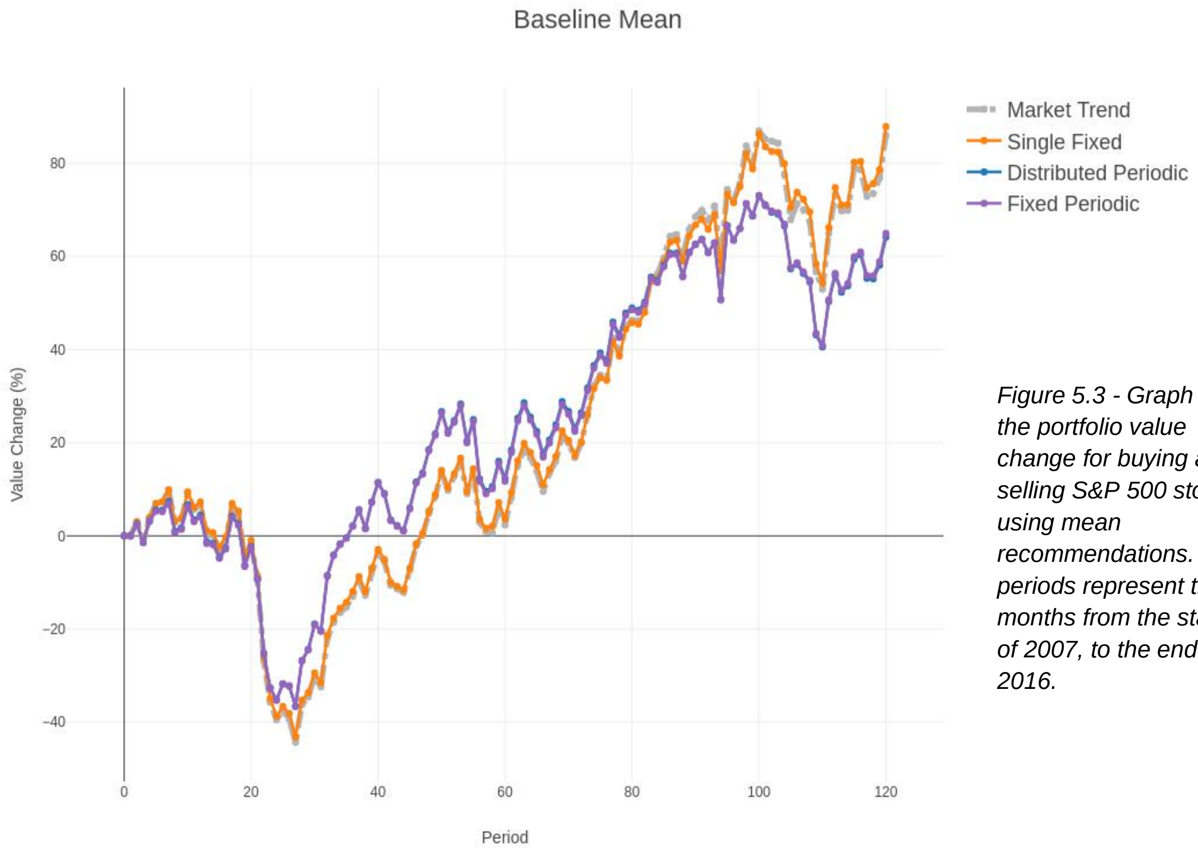


Figure 5.3 - Graph of the portfolio value change for buying and selling S&P 500 stocks using mean recommendations. The periods represent the months from the start of 2007, to the end of 2016.

With a larger time frame, we are now better equipped to evaluate trends. As we can see in *figure 5.3*, there is still a pattern of following the market. However, the periodic methods benefit from spikes in the market, as seen in the recovery period after the global financial crisis (periods 27 - 40). In this time, there was a 48% gain in a period of a year. In comparison, the single investment strategy climbed 41% in the same time. Although, the net change is quite similar, the periodic investment strategies are shown to be worse overall than a single fixed investment. This tells us that they should only be used when there is an upward trend.

However, the aim of this report is not to find an optimal investment strategy, but to find the best aggregation of expert opinions. We now have a good baseline to improve on in the following sections.

Note – The code that implements the 'baseline mean' rating generation described in this section is found at: https://github.com/lp1719/Stock-Analysis/blob/master/rating_generators/baseline_mean.py

6. Recommendation Time Frame

In the baseline case, the mean was calculated from the 'Thomson Reuters Estimates' themselves. The period of time for which they consider recommendations valid is unclear, but we can make our own expectations on how time should affect the importance of a recommendation.

Consider the scenario where there are great expectations for a company at the start of the year, so people suggest buying up the stock quickly. However, at the end of the first quarter, there are some damaging reports on how the business is being run and so recommendations have swung to the selling side. If we are taking a simple average, these recommendations would cancel out and we would be left with 'holding' a stock that is losing value.

Therefore, we will consider introducing a weighting to the stocks based on the time that the recommendation was made. To do this, we define a function to scale the recommendation. A hyperbolic/exponential function could be used as follows:

$$w(t) = \frac{1}{ct + 1}$$

$$w(t) = e^{-ct}$$

The variables are as follows:

- $w(t)$ – the weight that a rating holds with respect to time.
- t – the number of months since the rating was made.
- c – a scaling constant.

These functions can be visualized in the following graphs, which look at how the weight changes with respect to the number of months that have passed (different coloured lines have differed constant c values):

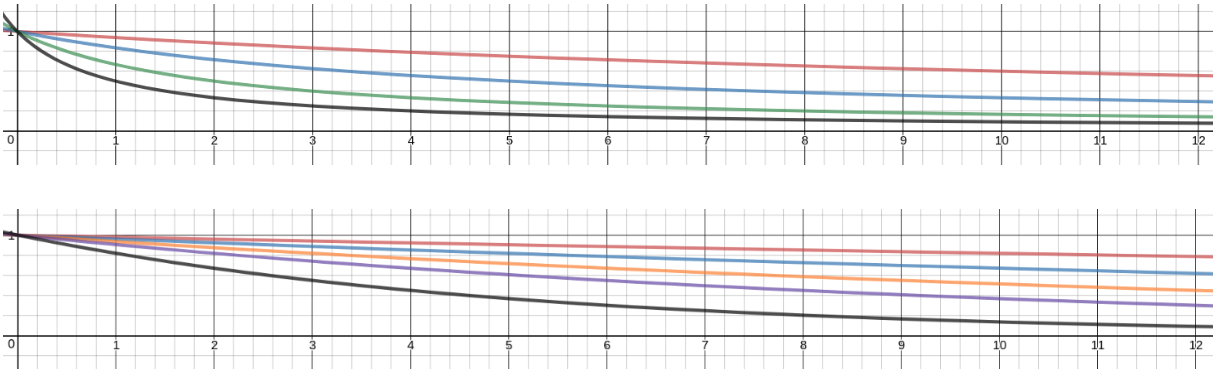


Figure 6.1 - Potential changes to rating weights. Hyperbolic functions (top) and exponential functions (bottom) are shown with varying coefficients (c). Images were sourced from Desmos online graphing software [4].

Putting it all together, we now formulate ratings of the i^{th} stock at the j^{th} period like so:

$$T_w = \sum_{\substack{A_l,\mathrm{n}\,:\,A_l \to (i,n) \\ n \leq j}} w(j-n)$$

$$R(i,j) = \sum_{\substack{A_k,m\,:\,A_k \to (i,m) \\ m \leq j}} \frac{w(j-m)}{T_w} A_k(i,m) \qquad (\mathrm{R.2})$$

Breaking down the formula:
We see that we are looking at all the ratings before and including the current period, by the summation conditions: $A_k, m \;:\; A_k \;\to\; (i,\; m)\; and\; m \;\leq\; j$
Then the weight of the period in which the recommendation was made, is divided by the total weight of periods where recommendations were made for stock i: $w(j-m)/T_w$.
This fraction is the amount of the current recommendation that should contribute to the overall rating. Hence, we multiply the fraction with the recommendation $A_k(i,m)$.
Note that the weight function is the hyperbolic/exponential function described earlier.

To prove that the theory is “as good as it sounds”, we must test it. Using the same test methods as before, we run a simulation using the ‘fixed periodic’ investment strategy:

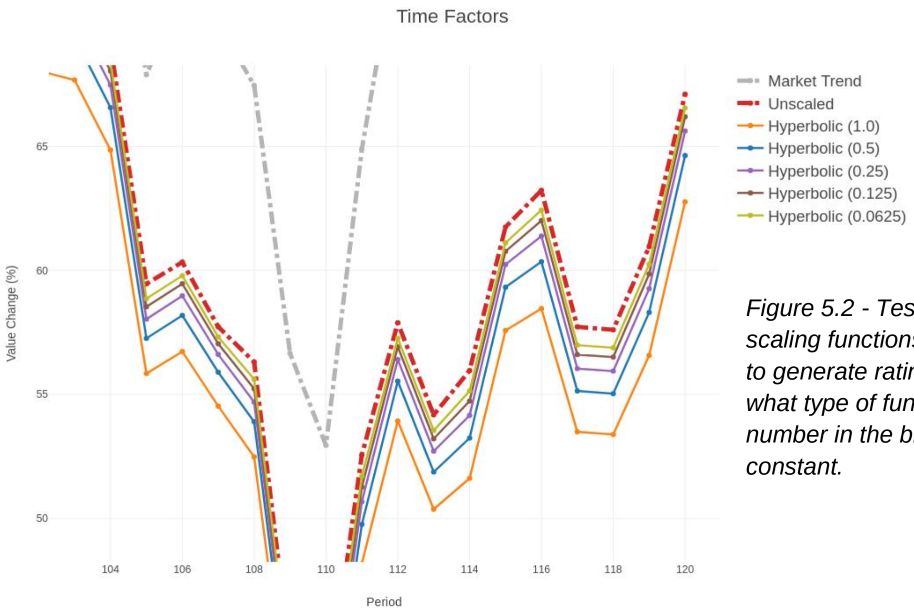


Figure 5.2 - Testing different hyperbolic scaling functions w(t), with equation R.2 to generate ratings. The legend tells what type of function was used, with the number in the brackets specifying the constant.

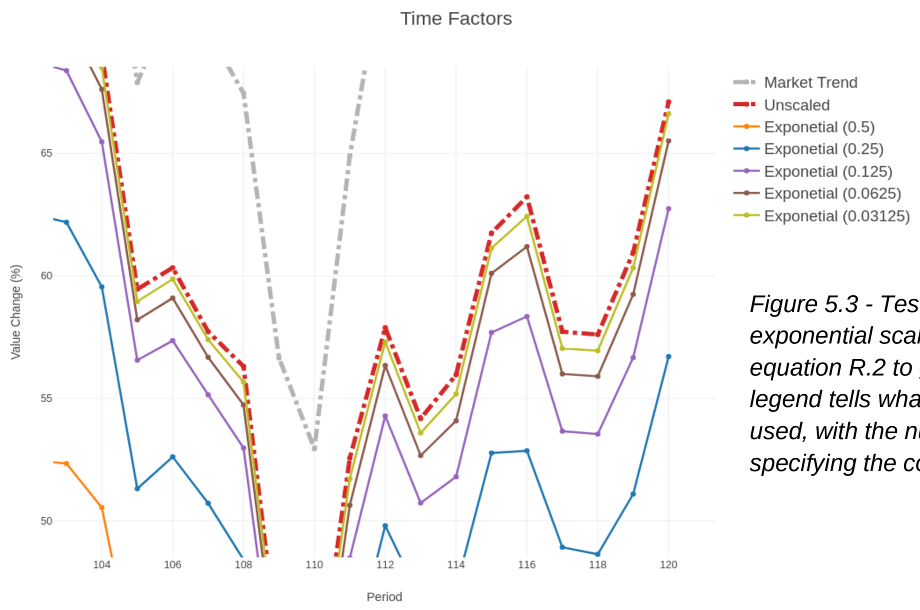


Figure 5.3 - Testing different exponential scaling functions w(t), with equation R.2 to generate ratings. The legend tells what type of function was used, with the number in the brackets specifying the constant.

The results here are quite peculiar and contradict the theory described earlier. Averaging all recommendations prior to and including the evaluation point, is better than giving higher weight to more recent recommendations. This trend was not specific just to the 'fixed periodic' strategy, but all investment strategies yielded the same results. Looking at equation R.2 again, can you think of a situation which leads to a large pitfall?

Hopefully you stopped to think about it, but if you didn't or can't see an issue, think about the following situation. For a particular stock, the ratings are very sparse. We are currently in a period where the stock hasn't been given a score for over 2 years, but they were previously all strong buys. What would be a reasonable rating to give for this period? What does our current formula return as the rating?

To answer the first question, it is unnatural to expect that the rating from 2 years ago is still the same and so the actual recommendation should be scaled down. As for the second question, since all the old ratings were strong buys and there haven't been any recent recommendations to take a larger slice of the weighting fraction, the score will still be a strong buy.

Therefore, to scale the value down, we can multiply $A_k(i,m)$ by a scaling factor that is adjustable with time. You may notice that the weight itself is such a value. Hence, our new formula is:

$$R(i,j) = \sum_{\substack{A_k,m\,:\,A_k \to (i,m) \\ m \le j}} \frac{w(j-m)}{T_w} A_k(i,m)\, w(j-m) \qquad (\mathrm{R.3})$$

Again, we test out new ratings:

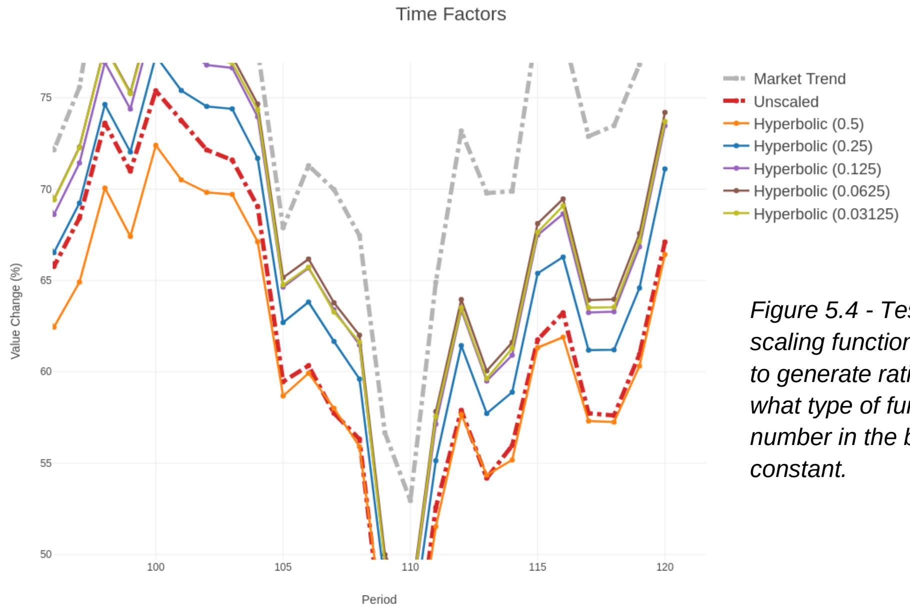


Figure 5.4 - Testing different hyperbolic scaling functions w(t), with equation R.3 to generate ratings. The legend tells what type of function was used, with the number in the brackets specifying the constant.

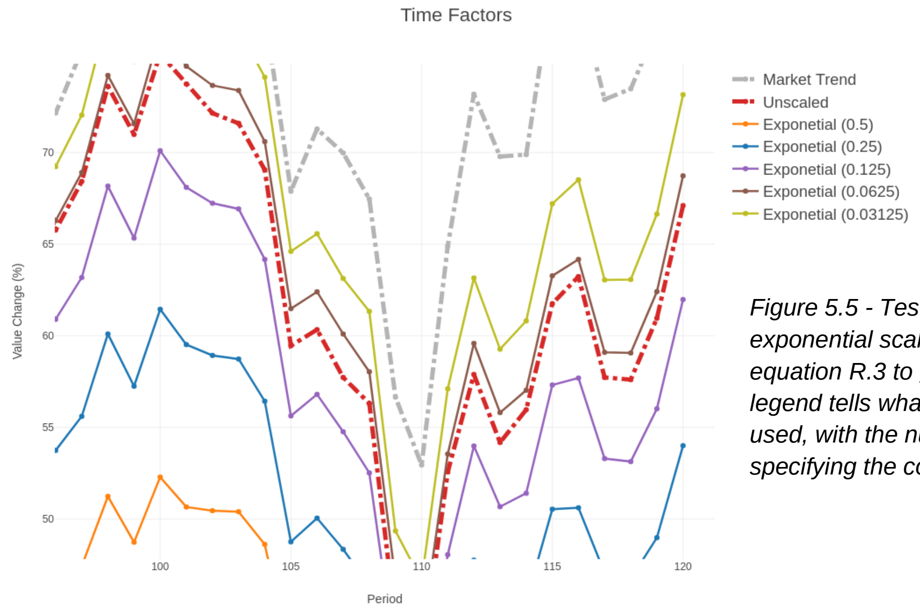


Figure 5.5 - Testing different exponential scaling functions w(t), with equation R.3 to generate ratings. The legend tells what type of function was used, with the number in the brackets specifying the constant.

These results are much more consistent with our expectations. For the hyperbolic functions, we see that the portfolio value improves as we decrease the constant factor, until we hit a specific peak point. This reduction of the constant has the effect of reducing the penalization factor for going further back in time. However, having no penalization factor (unscaled), does not produce the best results. The same can be seen for the exponential functions, however the peak point had not been seen yet.

Let us now take the best hyperbolic and exponential functions, i.e. those with constants that produce the best results and compare them with the baseline:

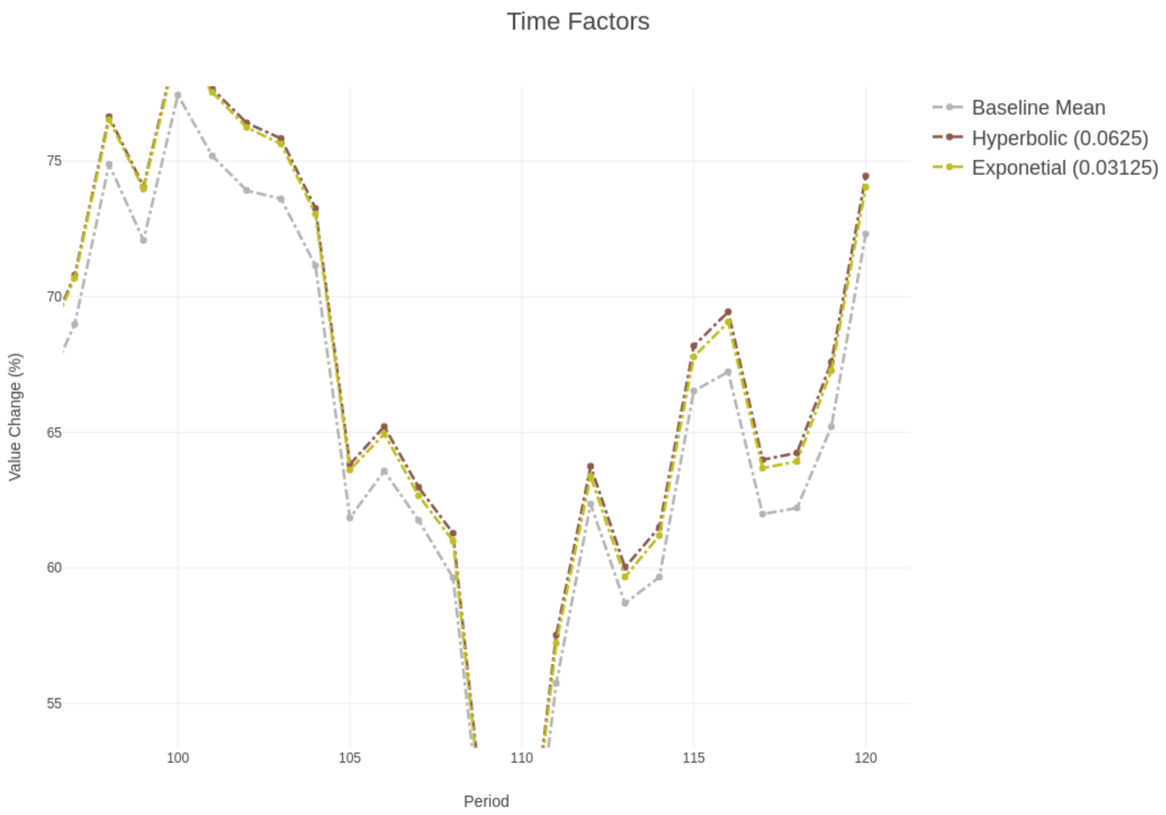


Figure 5.6 - The average portfolio value change of the three investment strategies, for different rating generators.

Therefore, considering a simple factor such as time can lead to slight improvements in recommendations. Although the percentage increase is only minor, stock markets experience an extremely large number of trades daily, with huge amounts of money moving around. Even a small percentage increase can lead to millions of dollars being gained/lost for investors.

To finish this section, let's look at a simple worked example to allow for the concept to sink in. Consider that you are trying to evaluate a stock with the following information:

- One year ago, an analyst said it was time to sell (-1)
- Six months ago, another analyst said it's time to buy (0.5).
- One month ago, another analyst said it’s a strong buy (1)

Previously, we would have taken the simple average:

$$Rating = \frac{-1 + 0.5 + 1}{3} = 0.17$$

However, our intuition tells us that it seems like now is a good time to buy. Therefore, we use our new formula to factor in time and gives value to our intuition:

$$Let\ w(t) = \frac{1}{(0.0625)t + 1}$$

$$
\begin{aligned}
Rating &= \frac{w(12)}{w(12)+w(6)+w(1)}(-1)\,w(12) + \frac{w(6)}{w(12)+w(6)+w(1)}(0.5)\,w(6) \\
&\quad + \frac{w(1)}{w(12)+w(6)+w(1)}(1)\,w(1) \\
&= \frac{0.57}{0.57+0.72+0.94}(-1)(0.57) + \frac{0.72}{0.57+0.72+0.94}(0.5)(0.72) \\
&\quad + \frac{0.94}{0.57+0.72+0.94}(1)(0.94) \\
&= -0.14 + 0.12 + 0.40 \\
&= -0.14 + 0.12 + 0.40 \\
&= 0.38
\end{aligned}
$$

The rating has now doubled and leans more towards what we feel is a better strategy.

This idea of "weighting" scores will be continued in the next section, but this time by looking at the analysts themselves.

Note – The code that implements R2 and R3 respectively is found at:
https://github.com/lp1719/Stock-Analysis/blob/master/rating_generators/time_factor.py
https://github.com/lp1719/Stock-Analysis/blob/master/rating_generators/time_factor2.py

7. Following Community Sentiment (Adaptive Voting)

Note that the following algorithm has been adapted from Aleks Ignjatovic's lecture notes [4].

Imagine walking home from a polling booth during the 2012 U.S presidential election. On your way, you come across a crazy man on the street shouting that Mitt Romney will win. When you arrive home, you turn on the TV to find several political experts stating that Barack Obama will continue his presidency. Who will you believe?

This is where the notion of community sentiment comes in. We are more likely to trust people that make up the majority. Therefore, we will give each analyst their own 'faith rating', which represents how much faith we have in their evaluations. We will use the syntax of $F(A_k, j)$, as our level of trust in A_k during the j^{th} period.

Then calculating the rating of the i^{th} stock at the j^{th} period of time is simple:

$$
T_F = \sum_{\substack{A_l,\mathrm{n}\,:\,A_l \to (i,n) \\ j-12 \le n \le j}} F(A_l, j)
$$

$$
R(i,j) = \sum_{\substack{A_k,m\,:\,A_k \to (i,m) \\ j-12 \le m \le j}} \frac{F(A_k, j)}{T_F} A_k(i,m) \qquad (\mathrm{R}.4)
$$

You may notice that this formula is like equation R.3**.** That is because it is governed by the same principle. We get the fraction of the current analyst's faith score, over the total faith amongst all analysts ($F(A_k, j)/T_F$) and then multiply it by their rating $A_k(i, m)$. Therefore, the analysts we trust more get a bigger say on what the rating should be.

We must now define a way of calculating this faith value, which is the trickier part. To do this, we will use an iterative algorithm. At the first iteration and first time period, we don't know what the community sentiment is. Hence, we give all the analysts an equal faith rating of 1. For all periods past that, the faith rating is initialized to previous period's value:

$$F(A_k, 0)^{(0)} = 1$$
$$F(A_k, j)^{(0)} = F(A_k, j-1)^{(0)}$$

We now look to find what the general community sentiment on each type of recommendation is. To do this, we consider that for each stock, at each period, each recommendation (-1, -0.5, 0, 0.5, 1) is given a score. We will represent these values as $\rho(i, j, \alpha)$. This is the score of recommendation α for the i^{th} stock at the j^{th} period and is calculated as follows:

$$\rho(i, j, \alpha)^{(t+1)} = \sum_{\substack{A_k:\, A_k \to (i,m,\alpha) \\ j-12 \le m \le j}} \frac{F(A_k, j)^{(t)}}{T_F}$$

The notation $A_k \to i, m, \alpha$ indicates that analyst A_k recommended to do α for the i^{th} stock within a 12-month period of j. Therefore, the recommendation gets a score that is the fraction of the faith scores of analysts that chose that recommendation, over all the faith scores of people who voted for the same stock in the same time period.

Now we can re-evaluate the faith ratings for the analysts based on the newly calculated sentiment:

$$F(A_k, j)^{(t+1)} = \sum_{i,\alpha \,:\, A_k \to (i,j,\alpha)} \frac{\rho(i, j, \alpha)^{(t)}}{|\{k, \beta \,:\, A_k \to (k, j, \beta)\}|}$$

This is the sum of all the scores that the analysts' recommendations are getting, divided by the total number of recommendations the analyst has made in this time period. This means that two analysts who have made a different number of recommendations are compared equally.

We then continue calculating $\rho(i, j, \alpha)^{(t+1)}$ and $F(A_k, j)^{(t+1)}$ until a sufficient level of accuracy is achieved. For our case, this can be when the scores converge to a value accurate to 10^{-3}:

$$\left\| \rho(i, j, \alpha)^{(t+1)} - \rho(i, j, \alpha)^{(t)} \right\| < 10^{-3}$$

Finally, using the calculated faith ratings, we proceed to calculate all the stock ratings using equation R.4.

The result of this algorithm is as follows:

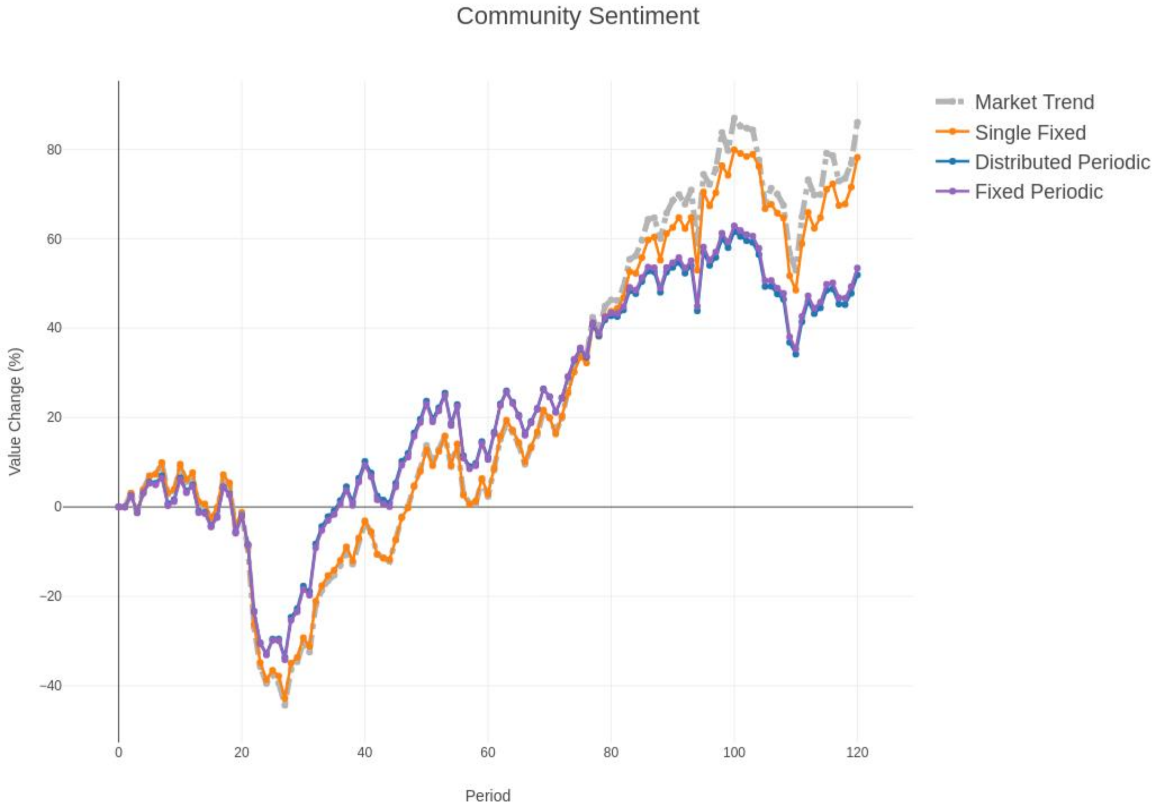


Figure 7.1 - The portfolio value change when using the community sentiment to generate stock ratings.

These results are somewhat worse than the baseline case by 10%.

It is still worth seeing how this algorithm works with a worked example, as it is the basis for the next algorithm we will look at. Consider that we have three analysts (A, B and C), giving recommendations for stocks in the same period (there is not complete overlap with the stocks they have recommended). We represent their recommendations in the following table:

$Stock\backslash Analyst$	A	B	C
1	1	−1	1
2	–	−1	1
3	0.5	−1	0.5

Our intuition tells us that B is a malicious recommender (unless they know there is going to be a global financial crisis). Therefore, our algorithm should hopefully ignore his recommendation after processing. We start by filling out the initial faith values:

$Analyst$	A	B	C
$Faith$	1	1	1

The first values are initialized to 1 and we proceed by rating the recommendations themselves:

$Stock\backslash Rec.$	-1	-0.5	0	0.5	1
1	1/3	0	0	0	2/3
2	1/2	0	0	0	1/2
3	1/3	0	0	2/3	0

Notice on this first iteration, the values are just the fraction of people who have chosen that recommendation. We now recalculate the faith rating using the table above, which is then used to recalculate the recommendation ratings, in a cycle that ends when we are happy with the accuracy:

$$Faith\ in\ A = \frac{\left(\frac{2}{3}+\frac{2}{3}\right)}{2} = \frac{2}{3}$$

$$Faith\ in\ B = \frac{\left(\frac{1}{3}+\frac{1}{2}+\frac{1}{3}\right)}{3} = \frac{7}{18}$$

$$Faith\ in\ C = \frac{\left(\frac{2}{3}+\frac{1}{2}+\frac{2}{3}\right)}{3} = \frac{11}{18}$$

$Analyst$	A	B	C
$Faith$	2/3	7/18	11/18

$$Rec. of -1\ for\ stock\ 1 = \frac{\frac{7}{18}}{\frac{7}{18}+\frac{2}{3}+\frac{11}{18}} = \frac{7}{30}$$

$$Rec. of\ 1\ for\ stock\ 1 = \frac{\frac{2}{3}+\frac{11}{18}}{\frac{7}{18}+\frac{2}{3}+\frac{11}{18}} = \frac{23}{30}$$

$$Rec. of -1\ for\ stock\ 2 = \frac{\frac{7}{18}}{\frac{7}{18}+\frac{11}{18}} = \frac{7}{18}$$

$$Rec. of\ 1\ for\ stock\ 2 = \frac{\frac{11}{18}}{\frac{7}{18}+\frac{11}{18}} = \frac{11}{18}$$

$$Rec. of -1\ for\ stock\ 3 = \frac{\frac{7}{18}}{\frac{7}{18}+\frac{2}{3}+\frac{11}{18}} = \frac{7}{30}$$

$$Rec. of\ 0.5\ for\ stock\ 3 = \frac{\frac{2}{3}+\frac{11}{18}}{\frac{7}{18}+\frac{2}{3}+\frac{11}{18}} = \frac{23}{30}$$

$Stock\backslash Rec.$	-1	-0.5	0	0.5	1
1	7/30	0	0	0	23/30
2	7/18	0	0	0	11/18
3	7/30	0	0	23/30	0

The calculations should make sense at this point, so we will just now look at the converging table values:

$Analyst$	A	B	C
$Faith$	0.77	0.29	0.71

$Stock\backslash Rec.$	-1	-0.5	0	0.5	1
1	0.16	0	0	0	0.84
2	0.29	0	0	0	0.71
3	0.16	0	0	0.84	0

$Analyst$	A	B	C
$Faith$	0.84	0.20	0.80

$Stock\backslash Rec.$	-1	-0.5	0	0.5	1
1	0.11	0	0	0	0.89
2	0.20	0	0	0	0.80
3	0.11	0	0	0.89	0

$Analyst$	A	B	C
$Faith$	0.89	0.14	0.86

$Stock\backslash Rec.$	-1	-0.5	0	0.5	1
1	0.07	0	0	0	0.93
2	0.14	0	0	0	0.86
3	0.07	0	0	0.93	0

Therefore, we clearly see that the recommendation will inevitably be:

- 'Strong Buy' for stock 1
- 'Strong Buy' for stock 2
- 'Buy' for stock 3

This is despite one analyst saying that each stock is a 'Strong Sell'.

However, as we saw from the results, "majority rules" is not the rule when it comes to the stock market. This is not the first time that such a counterintuitive idea has been presented. In fact, there is a whole investment ideology known as 'investing against the herd' [6]. However, we will stick to some more concrete data in the next section

Note – The code that implements the 'community sentiment' rating generation described in this section is found at: https://github.com/lp1719/Stock-Analysis/blob/master/rating_generators/ratings_community_sentiment.py

8. Ranking Experts By Past Success

Going back to our election example, let's consider that you heard the crazy man on the street shouting that Donald Trump will win the 2016 U.S. presidential election after voting for the 2012 election. However, the political experts on TV claim it will be "Hilary Clinton". We saw in the previous section, that following the community sentiment is not necessarily the best strategy and we all know how that election turned out. So, what piece of information are we missing to better evaluate these predictions?

In general, superficial specifications are beyond that of a computer and are in fact, often misleading. We want to be able to use concrete information. Taking the same situation, what if we now said that the crazy man on the street had predicted the last 5 election results correctly, whereas the so called “experts” on the news had only predicted 2. Who would you be more willing to believe now?

This is the basis of how we will be ranking experts in this section. It won’t be by their names, titles or star signs, but by their track record. As the saying goes, “results speak for themselves”.

The challenge is to be able to measure success. The general notion is that if an analyst says buy, the prices of the shares should go up, or if they say sell, the prices should drop. But we now need to put a numerical score for the level of success that an analyst has achieved. This value will be our faith rating $F(A_k, j)$ this time.

We now calculate the stock ratings in the same manner as R.4, but with one slight difference:

$$T_F = \sum_{\substack{A_l,\mathrm{n}\,:\,A_l \to (i,n) \\ j-12\leq n \leq j}} |F(A_l, j)|$$

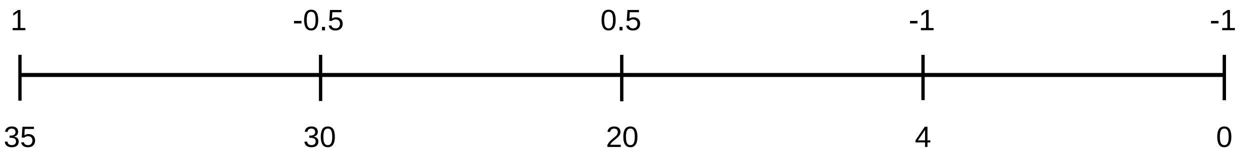
$$R(i,j) = \sum_{\substack{A_k,m\,:\,A_k \to (i,m) \\ j-12\leq m \leq j}} \frac{F(A_k, j)}{T_F} A_k(i,m) \qquad (\mathrm{R.5})$$

The difference comes in the calculation of the total faith rating, where the absolute values are taken. This is because we have not restricted these values to be positive. For example, if an analyst constantly makes poor evaluations, then they might receive a negative score, indicating we should perhaps do the opposite of what they say to do.

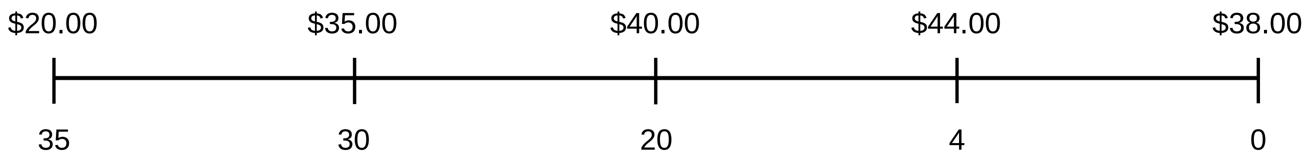
To calculate the faith scores at a specific period j, we use a ‘simple success measurement’, which is calculated in the following way:

1. Look at each stock that analyst A_k has given recommendations for, prior to period j.
2. For each recommendation, take note of the value of the stock at that time and either:
    - At the time when the next recommendation for the same stock was made (if it was within 12 months of the previous recommendation and prior to period j).
    - 12 months after the recommendation (if it was prior to period$j$).
    - At the current period j.
3. Calculate the change in value
4. Then do one of the following:
    - If the rating was > 0 (buy):
        - $F(A_k, j) \mathrel{+}= (rating)\ (change\ in\ value)$
    - If the rating was < 0 (sell):
        - $F(A_k, j) \mathrel{+}= (rating)\ (-change\ in\ value)$

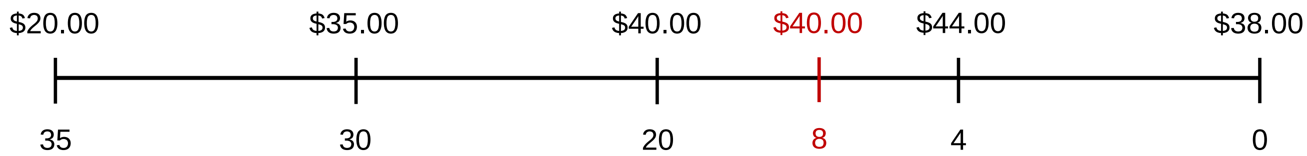
For example, consider the following timeline of recommendations with regards to one analyst and one stock. The top values are the recommendation and the bottom values are how many months in the past the evaluation was recorded:



And the following are the prices of the stock they are evaluating at these times:



However, we only look at price changes in a maximum period of 12 months. Therefore, we look at the following price timeline:



Now looking at the changes in the stock value between recommendation periods, we calculate the faith value as follows:

$$
\begin{aligned}
Faith &= (1)(15) + (-0.5)(5) + (0.5)(0) + (-1)(-6) \\
&= 15 - 2.5 + 0 + 6 \\
&= 18.5
\end{aligned}
$$

When the algorithm's results are put through the simulator the following results are produced:

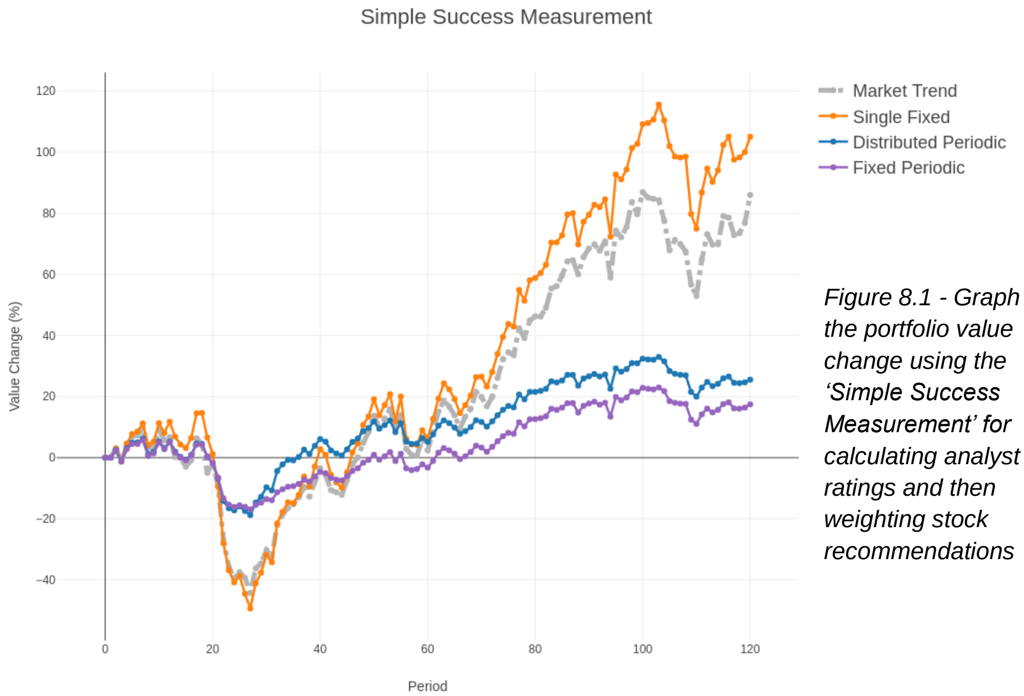


Figure 8.1 - Graph of the portfolio value change using the 'Simple Success Measurement' for calculating analyst faith ratings and then weighting stock recommendations

At first glance *figure 8.1* boasts some odd results. Somehow the 'single fixed' investment strategy has gained 17% over the baseline case, whereas the other methods have lost 40+%. How can these changes be so drastically different?

Looking back to the baseline graph in Figure 5.3**,** we notice that the 'single fixed' investment follows the market trend very closely. This indicates that very minimal buying/selling is occurring. However, we now see large variations between the two lines. Hence, more selling must be occurring, freeing up funds for reinvestment. This means that the other methods must also be selling more shares too. However, unlike the 'single fixed' strategy, the money is not reinvested. Therefore, less of the value is experiencing the gain of the overall market, lowering profits. The reduced effect of market crashes on these investment strategies can also be explained with this reasoning.

To prove this theory true, let us introduce a fourth investment strategy. This new approach will introduce 1 000 000 every period (like periodic distributed), but will also reinvest the money that came from the selling of shares (like single fixed). Therefore, we will call it 'distributed periodic investment and reinvestment':

$$S(i,j) = S(i,j-1) + \frac{1000000 + M(j)}{P(i,j)} . \frac{R(i,j)}{T_R}$$

Let us now look at the results:

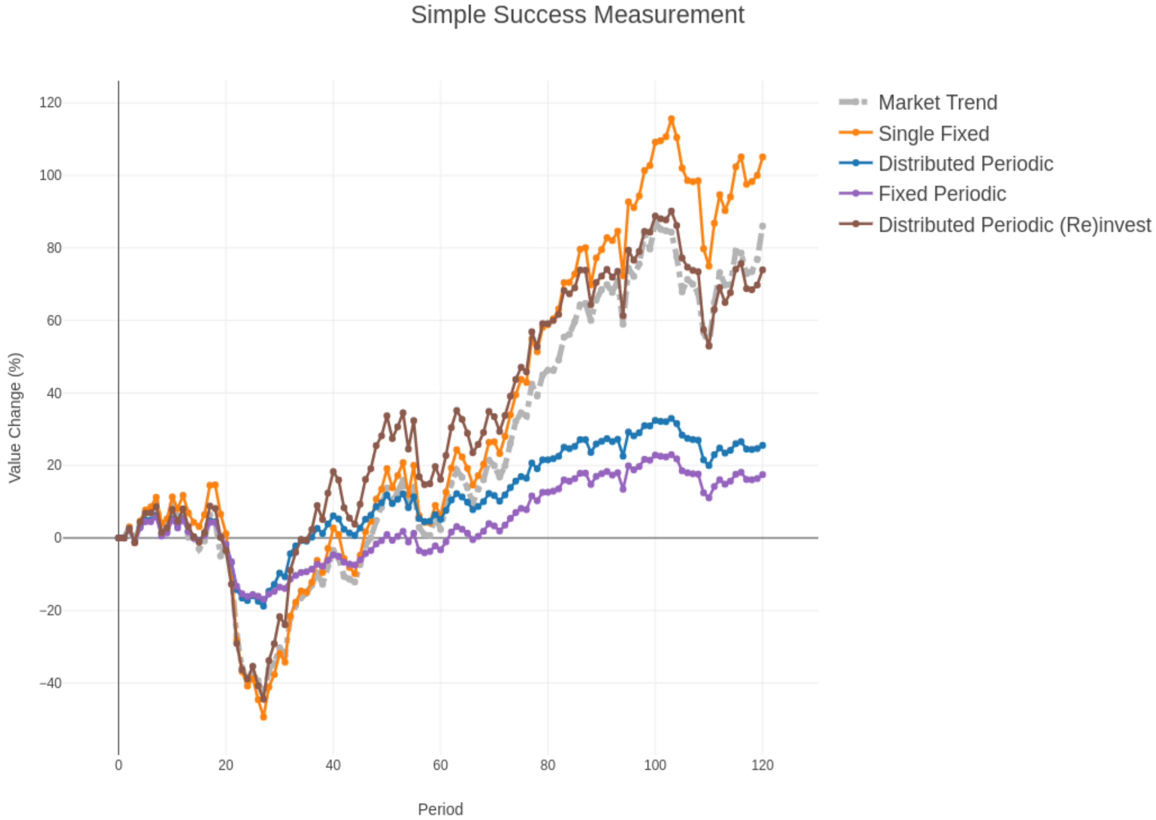


Figure 8.2 - Graph of the portfolio value change using the 'Simple Success Measurement' for ranking analysts and weighting their stock estimates accordingly. It features the new periodic distributed investment and reinvestment strategy.

Just as we suspected! *Figure 8.2* shows us a much clearer picture of how our algorithm is performing. Like before, adding more money into the market during rapid increases in the general market trend further increases portfolio value. From period 27 to 40, the new investment strategy leads to a 62% increase in value! The old baseline for this same period only saw a 48% increase in the same period.

The baseline mean with the new investment strategy yields the following results:

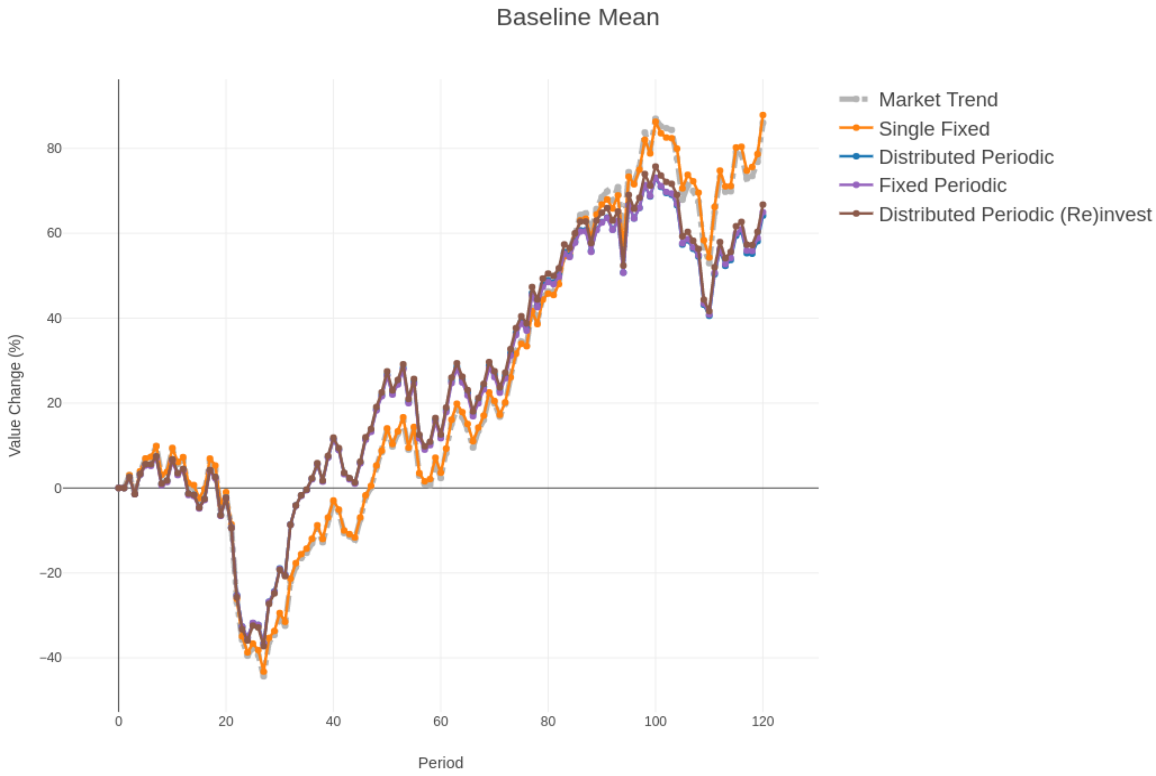


Figure 8.3 - Baseline mean that features the new distributed periodic investment and reinvestment strategy.

This should not be surprising. Since minimal selling is occurring, there is not much reinvestment occurring, so the new strategy follows the original 'distributed periodic' method.

From now on, we shall only consider the 'single fixed' investment strategy and 'periodic distributed investment and reinvestment' strategy when comparing algorithms.

Note – The code that implements the faith score generation described in this section is found at: https://github.com/lp1719/Stock-Analysis/blob/master/rating_generators/analyst_simple.py

And the code that implements the stock ratings generation is found at: https://github.com/lp1719/Stock-Analysis/blob/master/rating_generators/ratings_analyst_simple.py

9. Combining Time and Analyst Ratings

We have now made significant improvements with the stock values and it's time to put together the time factors and analyst's faith scores. The simplest way to do this would be as follows:

$$w(t) = \frac{1}{\frac{1}{16}t + 1}$$

$$T_w = \sum_{\substack{A_l,\mathrm{n} \,:\, A_l \to (i,n) \\ n \le j}} w(j-n)$$

$$T_F = \sum_{\substack{A_l,\mathrm{n} \,:\, A_l \to (i,n) \\ n \le j}} |F(A_l,j)|$$

$$R(i,j) = \sum_{\substack{A_k,m \,:\, A_k \to (i,m) \\ m \le j}} \frac{[w(j-m) + |F(A_k,j)|]\left[\frac{F(A_k,j)}{|F(A_k,j)|}\right]}{T_w + T_F} A_k(i,m) \qquad (\mathrm{R}.6)$$

The equation has the same structure as before, but now the weighting is a sum of the time and faith rating (more on this later). At first glance, the numerator may seem a little odd. However, all that is happening is a little trick to preserve the sign of the faith rating. For example, if the time weight has a value of 5 and the faith weight has a value of -5, the overall weighting should be 10, not zero. Therefore, we sum the time weight with the absolute value of the faith score: $[w(j-m) + |F(A_k,j)|]$. However, doing this means we lose the sign of the faith rating, which indicates if we should do what the analyst says or do the opposite. Therefore, we multiply it by $[F(A_k,j)/|F(A_k,j)|\ ]$, which will either be the value of 1 or -1 based on the original sign of $F(A_k,j)$.

There is currently one major flaw with this approach. Consider the case where the total time weight is 1 and the total faith weight is 99 (a bit extreme, but used to illustrate the issue). Then the values of the time weights will only have a 1% impact on the overall rating, whereas the faith weights have a 99% impact. Therefore, we will now modify the formula so that all the weights are normalized, allowing for the time and weight to have an equal impact.

$$R(i,j) = \sum_{\substack{A_k,m \,:\, A_k \to (i,m) \\ m \le j}} \frac{\left[\frac{w(j-m)}{T_w} + \frac{|F(A_k,j)|}{T_F}\right]\left[\frac{F(A_k,j)}{|F(A_k,j)|}\right]}{2} A_k(i,m) \qquad (\mathrm{R}.7)$$

Now, the total sum of $\frac{w(j-m)}{T_w}$ is 1, and so to is the total of sum of $\frac{|F(A_k,j)|}{T_F}$. Therefore, the fraction is now over 2, which is the overall total sum.

Applying equation R.7, we obtain the following results:

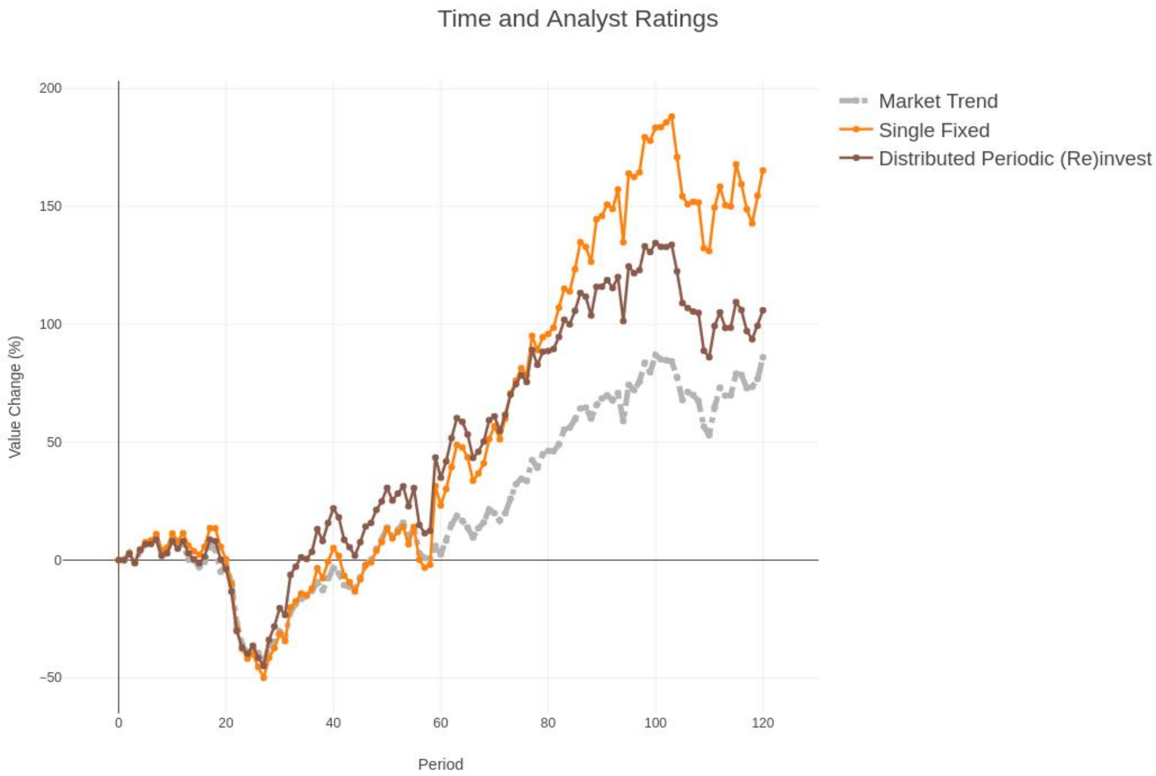


Figure 9.1 – Portfolio value change using both time and analyst faith scores to evaluate stock ratings.

This is a significant improvement. The 'single fixed' investment leads to over double the portfolio value compared to having static investments in the market!

Note – The code that implements the 'time and analyst' rating generation described in this section is found at: https://github.com/lp1719/Stock-Analysis/blob/master/rating_generators/ratings_analyst_and_time.py

10. Conclusion

Understanding the stock market is beyond the means of a mere computer science student and beyond even the experts who spend their lives studying it. However, those "economic professionals" have a bit more understanding than the average person.

Looking at the core principles of 'Time' and 'People' and creating weighted values accordingly, a reasonably simple and intuitive process was created, which was easy implementable and testable.

By the end of a steady, test driven algorithm development, we were able to leverage the information out from the "real" experts and so produce results that could have led to huge financial gains.

It is now up to you with what you do with this information!

References

[1] Thomson Reuters, 2017, *I/B/E/S Estimates* <https://financial.thomsonreuters.com/en/products/data-analytics/company-data/ibes-estimates.html>

[2] Investopedia, 2017, *Standard & Poor's 500 Index - S&P 500* <http://www.investopedia.com/terms/s/sp500.asp>

[3] Wharton University of Pennsylvania, 2017, *Wharton Research Data Services* <https://wrds-web.wharton.upenn.edu/wrds/>

[4] Desmos, 2017, *Desmos Calculator* <https://www.desmos.com/calculator>

[5] Aleks Ignjatovic, 2017, *COMP4121 Lecture Notes - Topic 3 More Fixed Point Algorithms: Online Voting and Ranking* <http://www.cse.unsw.edu.au/~cs4121/COMP4121_Lecture_Notes_3.pdf>

[6] Steve Cortes, 2011, *Against the Herd: 6 Contrarian Investment Strategies You Should Follow,* John Wiley & Sons, New Jersey