SENG 550 Final Project Report

Stock Return Predictions Through Multi Factor Modeling

Alpha-Bets Group
December 2021

By:
Parsa Honarmand
Qi Feng (Willie) Li
Jose Menjivar
Cole Thiessen

Project Abstract

This project strives to answer questions about different economic factors that may affect stock price returns over the long term. As stock markets and economies have evolved, economists and investors have continuously attempted to improve their stock return prediction models in search of sustainable investment profits. Our project is focused on the factor investing approach, an approach that involves targeting quantifiable firm characteristics or "factors" that help predict stock returns. This is considered a sophisticated investment method used by institutional investors such as pensions and hedge funds. As several of our group members are dual majors with finance, we thought it was appropriate to explore this investment approach that has been commonly denied to retail (non-institutional or individual) investors and is only briefly taught in business schools.

We sampled a large number of stocks in the S&P 500 (S&P), a portfolio of the biggest public companies in the U.S that acts as a proxy for the whole market. We then chose a set of factors that we thought were most appropriate to explain general stock returns and we iteratively ran numerous multiple linear regressions with the different factors being the set of independent variables (our feature set) and the stock returns being the dependent variable. We checked for factors that had a regression p-value of over 0.05 (signalling non-statistical significance), removed them from the regression, and ran the regression again without said factors to get a more precise set of factor weights. We only analyze regression results with adjusted R-squared values of over 0.65, since we deem the regressions above this threshold to strongly explain stock returns. Our training data, or the data used to run the regressions was about 70% of our data, with the predictions being done on the leftover 30%, and we have used factor and stock data since early 2014. Our results were promising, showing that our found factor weights predicted real stock returns in our test data. Additionally, we ran comparative regularized linear regressions with the stocks we found had a high R-squared value in the base regression, and surprisingly, the results of the regularized regression were much worse in every case.

Project Introduction

We chose the problem of stock returns prediction. This problem is important as a majority of people in the developed world now have at least a portion of their personal wealth tied to the stock market. In fact, with the vast migration out of company pensions and towards individual pension plans (401Ks, TFSAs), individuals are now forced to invest their savings into different assets, including the stock market, to pay for their retirements in the future. Our project can help individual investors better understand the nature of their stock returns to safeguard their retirement savings as well as can help institutional investors better manage their exposure to specific economic factors. Our project can also be used for any set or combination of factors, so not only the factors we have tested can be used to predict stock prices.

There has been some work done in the space in the past, but it has been mostly in the academic/mathematical realm. The original proponents of multi-factored models, Fama and French (finance and economics academics from Dartmouth University), came up with a three-factor theory in 1992, which was later expanded to five factors by 2014. Large hedge funds and investment institutions have partially adopted factor investing as one of their strategies, but this has remained largely unavailable to individual investors due to the academic aura that surrounds it. Factor investing has now been generalized, with some institutions realizing that there may be many factors that predict stock returns besides the 3 to 5 outlined by Fama and French's theories.

The gap we are trying to fill in this sector is to create a tool that allows retail investors (through open source code) to analyze all of the stocks in the S&P in a very short period of time. Additionally, our project can be used to predict stock prices given their correlations with specific factors. Users would also be able to input their own chosen factors, as long as they are in the correct format in the 'factorDirectory' subdirectory in the repository. This would provide any investor with the tools and predictive power only institutions have access to right now. Our data analysis questions are as follows:

1) What factors reliably predict individual stock returns via multiple linear regression?
2) How does the predictive accuracy of multiple linear regression compare to regularized regression?
3) How does the predictive accuracy of a neural network compare to that of all aforementioned regressions?

Our main findings were favorable as we found that our 11 factors have statistically significant predictive and explaining power on many stocks of the S&P. Even when we would allow stocks with relatively lower r-square values into our results, there would still be meaningful return predictions over our testing period (30% of our data, or stock returns since late 2019). This is significant, more so considering the fact that stock markets have experienced large uncertainty and volatility in this time period due to the COVID-19 pandemic, and this shows how our model can remain reliable in market turbulence.

Background and related work

To fully understand this project and its implications, it is useful to understand why factor investing is important, the original asset return models, and other ways to value stocks and their expected returns.

The Capital Asset Pricing Model (CAPM) stipulates that the returns of a stock is proportional to the expected market returns and the risk free rate, which is a guaranteed rate of return in a market. Usually, this rate of return is guaranteed through the purchase of government bonds, or lending money to the government to which you are guaranteed a certain interest rate in return, with no possibility of default. Usually, market returns are calculated using the S&P as a benchmark. The model may be summarized in a single formula line:

$$Ri = Rf + Bi * (Rm - Rf)$$

Where Rf is the risk free rate, Rm is the expected market rate of return, Rm - Rf are the "excess market returns", or the expected market returns over the risk free rate, Bi is the beta of stock i, or the coefficient of excess returns in the linear regression of excess market returns and stock returns, and Ri is the expected return of the asset (stock) i.

As it is evident, CAPM assumes that the only explaining variable for stocks are excess market returns. However, it is widely accepted that stocks do not only have market risk (sensitivity to market returns) but also have company specific risks. The formal names for these two kinds of risk are systematic (market inducing) and idiosyncratic (company specific) risks. This makes sense, as a stock's returns cannot only be a function of its reaction to market movements, but also the result of company events.

Fama and French were aware of these additional (non-market) sources of risk which had the potential to explain an asset's expected returns. As such, they developed a three-factor model of assets' expected returns.

$$Ri = Rf + Bi(Rm-Rf) + s1(SMB) + s2(HML)$$

The first few terms are the same as CAPM. However, SMB and HML are other factors that may contribute to a stock's expected returns. These factors are important to understand in order to grasp the rest of the project. SMB is the return spread (return difference) between small companies and big companies, so it represents the effect that company size has on returns. HML is the return spread of cheap minus expensive stocks (the price of a company stock in relation to the amount of money it made in profit). The different s terms are coefficients in a linear regression that explains the expected returns of asset i.

What our project strives to do is find factors that can predict (explain) a stock's return. Fama and French's three factor model has proven to be generalizable, in that any arbitrary number of factors can be used to predict a stock's price. In our experiment, we use different factors not captured by Fama and French's research. We use factors such as momentum, volatility, correlation to interest rates, growth, value or environmental exposure.

Extending beyond the Fama and French Model

The Fama and French model is heavily based on return spreads. However, these are not the only available factors for market analysis. Following the basis of the model, we are looking to extend beyond the three factors identified by Fama and French. Given that markets have changed considerably since 2014 with the rise of high frequency trading, blockchain technology, and ESG accountability, it is likely that the original five factors of the Fama and French model are due for an overhaul. We have picked 11 factors that we believe will yield the best results. In no particular order of significance, the factors are listed below:

1) Dow Jones High Momentum Index
2) Dow Jones Thematic Long Small Size Index
3) ESG Quality Index
4) ESG Value Index
5) High Rate Sensitivity Index
6) Low Rate Sensitivity Index
7) Inverse Rate Correlation Index
8) High Volatility Index
9) Low Volatility Index
10) Market Pure Growth Index
11) Market Pure Value Index

Each of these factors represent important macroeconomic indicators over various time intervals. 1) represents 200 companies ranked as having the highest momentum in the S&P. For an asset, momentum refers to the inertia of a price trend to continue over a particular time frame. These names incorporate systematic risk into our model. 2) has a considerably smaller systematic risk component as these are the smallest names, by market cap, in the Dow Jones. This index adds a mix of idiosyncratic risk into the model. 3) and 4) both relate to ESG (Environment, Social, and Governance), which has seen a growth in popularity in recent years. Thus, we thought that this would be a good factor to consider and broke it down into quality and value. Quality pertains to how ESG aware the companies are, so 3) represents the top quality names. 4) considers the top ESG companies but from a value perspective, looking to minimize financial ratios such as price to earnings, which indicates that the company is trading cheaply relative to peers. 5), 6) and 7) all relate to interest rates. The biggest market in the world is the bond market, which is highly sensitive to interest rates, and historically, interest rates have had a strong effect on the performance of stocks. Each of these indices considers all companies in the market, picking based on the high, low, and inverse rate sensitivity respectively. High rate sensitivity means that a stock's price changes drastically during rate announcements while low rate sensitivity means the opposite. Inverse rate sensitivity is when a stock's price moves in negative correlation to interest rate movements. Indices 8) and 9) correlate to volatility, which is a measure of the magnitude of historical stock movement. There is no consideration for price movement within these indices, allowing us to add another unique factor into our model. High volatility stocks tend to exhibit significant deviation from their mean price while low volatility stocks tend to exhibit minimal deviation from their mean price. Finally, we decided to consider two traditional finance factors in our model. 10) focuses on growth, which takes the stocks that have shown the greatest return over the given time frame. 11) focuses on value, which takes the stocks that have favorable ratios, similar to 4) except the scope is no longer limited to ESG companies.

Although each factor is important for predicting stock movement, the goal is to create a model that is able to address most of the factors while minimizing computing time and maximizing efficiency. For that reason, we decided to cap our model to these 11 factors, with potential to remove certain factors if it turns out that the factor does not show statistical significance during testing. The basis on which these factors were chosen was diversity, as we hope that movements that are not captured by one factor will be by another.

Technical specifications

Technologies used for this project are Python and Spark. The machine learning algorithms explored are linear regression and gated recurrent unit neural networks.

Setting up the multiple linear regression

Linear regression is chosen as one of the two machine learning algorithms we explore here because it has been tried and tested in the industry with some level of success. We look to gain statistically significant prediction results, in addition to a benchmark upon which to compare with our neural network.

The initial step was acquiring and cleaning the required data. All factor data was acquired from S&P Global while stock data was collected from yahoo finance. Since the factor data and stock data are from two different sources, we had to ensure that all dates matched.

Multiple linear regression results

Once the data was cleaned, a multiple linear regression was performed on multiple stocks. This would yield p-values for each factor. Since we're looking for only the statistically significant factors, a threshold of 0.05 was

used to determine which factors would be used for the next round of regression. Factors with a threshold of more than 0.05 would be discarded for the stock. The multiple regression would then be run again. This process is repeated until no factors are dropped.

As there are over 500 stocks in the market that we are considering, we decided only to store the predictions for stocks where the regression resulted produced an adjusted R-squared value of greater than 0.65, the threshold for statistical significance. This process produced a list of stocks, with a custom combination of factors each, where the predictions were statistically significant.

One concerning metric was mean squared error, which was ~1% for all saved stocks. This is a considerable amount because this percentage translates to 1% per day. With consideration for compounding, it means that our results are only reliable for a short period of time. Considering that stocks generally show daily movement of ~5%, this translates to a whopping 20% error per day. However, this is only the worst case scenario. The results of the multiple linear regression appear relatively accurate, despite the errors, as stock movements tend to follow brownian motion. Thus, the error often cancels out over a longer period of time. The graphical results demonstrate this, which is shown in the figure below for one of the stocks.

Performance relative to regularized linear regression

While the multiple linear regression performed well, we were curious if it suffered from overweighting for certain factors. As a result, we ran a regularized linear regression for all stocks that were statistically significant. The multiple linear regression did not suffer from overweighting as the results were far superior to the regularized linear regression. Similar to the multiple linear regression, the mean squared error was very similar for all stocks. However, instead of a value ~1%, the mean squared error for regularized linear regression was ~3%. Again, as stocks follow brownian motion, the error fails to compound over time. However, the graphical results are in line with the significant increase in mean squared error, which are shown below. It is important to note that analysis was done with several different choices for the regularization hyperparameter (lambda). The demonstrated regularized result uses a lambda hyperparameter of 0.1. Tuning this hyperparameter we realized that the best results were obtained when lambda was closest to 0, meaning that our unregularized model was not overfitting to the data initially, and regularization is not necessary.

While the results from the regression prediction may be used with reasonable confidence, the results from the regularized regression are unusable as the testing error is too high. We show our results by choosing one of the output stocks from our regression, TFC, or Truist Financial.

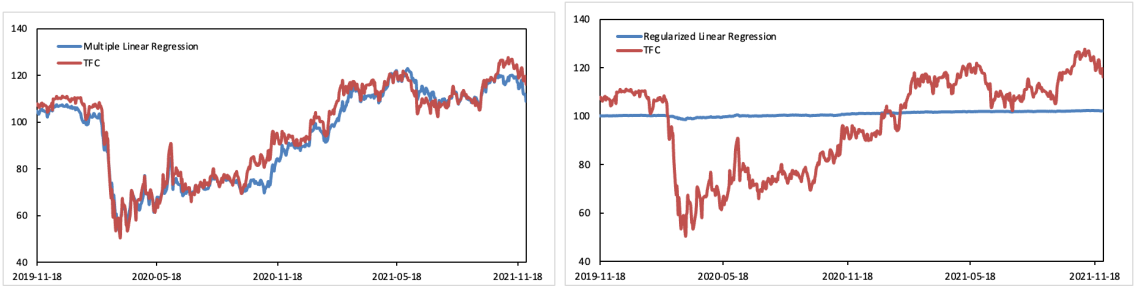

Figure 1: multiple Unregularized Linear (Left) and Regularized Linear Regression (Right) prediction results
Note: Returns indexed to $100 on the starting date

Optimizations with Spark

In the world of financial markets, timing can be extremely important and accurate information that has incorrect timing is effectively useless. Thus, it was in the interest of the project to optimize the linear regression using spark. Initially, the linear regression was completed using pandas and numpy python libraries. However, while pandas is good for working with data, Spark is vastly superior with respect to speed due to the ability to parallelize by partitions. Spark was not immediately implemented in the original version of the project as the team chose to follow agile methodology.

The introduction of spark was able to cut the linear regression time by 7.8x for 300 stocks, from 351 seconds with one partition, down to 45 seconds with 8 partitions. We were successful in partitioning our data pulls as well as our regressions, where our data pulls were the most costly to our performance. Nonetheless, we demonstrated that our approach is scalable to any number of stocks for any time frame using Spark.

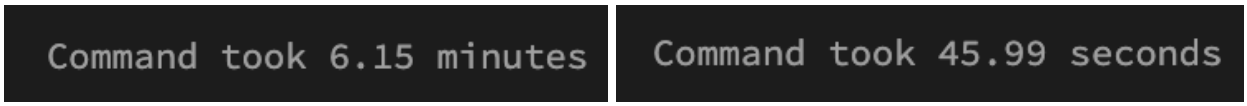

Figure 2: 1 partition (Left) and 8 partitions (Right) timing

A promising neural network

We built out a neural network using only historical stock prices as the inputs. From financial theory, historical stock prices should not have an impact on future performance although empirical evidence suggests otherwise. Thus, it was surprising when the neural network prediction results were comparable to those of the linear regression model.

The mean squared error for the neural network was ~1%, which is on par with that of the multiple linear regression. The visual results are comparable as well, which is shown below. The same time period for the multiple linear regression is also provided.

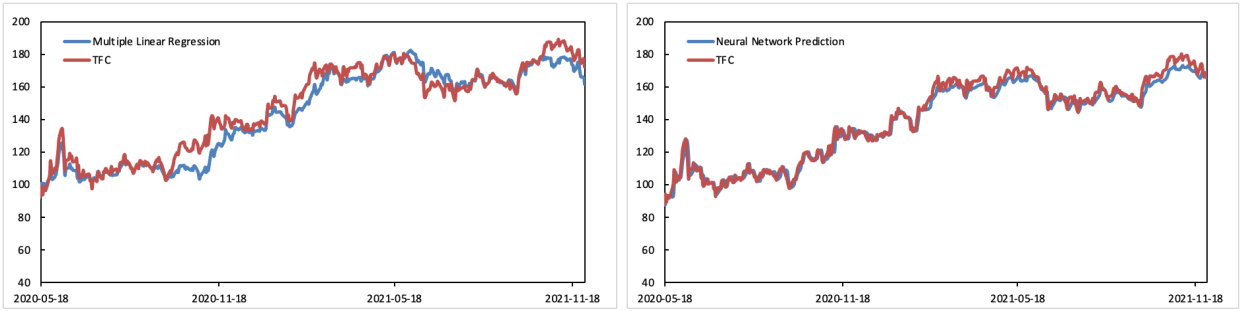

Figure 3: Unregularized Linear (Left) and Neural Network (Right) prediction results
Note: Returns Indexed to $100

Although both results had similar mean squared errors, it is evident from the above figures that the neural net is more accurate than the linear regression, with almost no periods of visible deviation from the actual price.

Potential future improvements and conclusion

As this project has shown promising prediction results, there is reason to extend the model. Two methods that are immediately obvious are multiple linear regression and a multi-factor neural network.

The multiple linear regression filtered factors based on their p-value to ensure statistical significance. However, this does not explore all possible combinations with all factors. Given the 11 factors chosen, there are 2047 different combinations of these factors for each stock, the sum of all possible combinations with 1 to 11 factors. Given that there are over 500 stocks in the market, determining every combination of factors would have a very high time complexity, but the optimization in Spark, assuming optimal parallelization, would potentially allow for a 2047 times speed up as each regression can be performed in a partition.

The neural network yielded superior results to the multiple linear regression using only historical stock price data. It is possible that a neural network that had multiple factor inputs would perform considerably better, potentially resulting in an even lower mean squared error, significantly increasing prediction accuracy.

The future iteration would likely look to implement both of these improvements, as benefits of spark are not limited to the multiple linear regression. Multiple instances of the neural network can be trained across different partitions, improving the neural network's speed complexity, allowing for extreme accuracy in record times.