Fundamental Signals for Algorithmic Trading
MS&E 448 Final Project

James Wong, Yanni Souroutzidis, Max Lai, Evelyn Mei, Ankit Sagwal
Dr. Lisa Borland (supervisor)

June 6, 2016

Abstract

The project aims at analyzing the impact and making use of fundamental signals to predict the future stock price performance of equities. We looked at a number of signals - firm's financial performance metrics and ratios (fundamental signals), critical events and public sentiment. After discussing potential approaches, we decided to start with the Healthcare sector to develop the core algorithm before moving on to other sectors (Technology). The strategy hypothesizes that fundamental and sentiment signals could be good predictors of stock price performance. The engine is based on machine learning - we trained and optimized a random forest model to make predictions. The project has given us a good understanding of which factors are important in making predictions. But more importantly, through this effort, we were able to develop a framework to test fundamental based algorithms for predictive power. This was critical since outperformance in test scenarios is not a sufficient condition for a great algorithm.

1 Data Collection

We started our project focusing on the Healthcare sector. The impetus for this was because we originally wanted to explore event-driven trading which would be done through mining clinical trial data in addition to other datasets available to us. After developing a reasonable model for the Healthcare sector, we extended our work to other sectors such as Technology.

The features we explored include: fundamental data, sentiment data, clinical trial data, and stock price and volume data. Raw feature data is very skewed and not predictive by itself. For example in Figure 1 we can see that there is a wide range of values for working capital fundamental values, and the predictability of short term returns or long term enterprise value has a high variance. To resolve this we derived the stock's feature value normalized with the particular stock's past feature values (rather than taking the raw feature value) and took the z-score to get a cleaner signal.

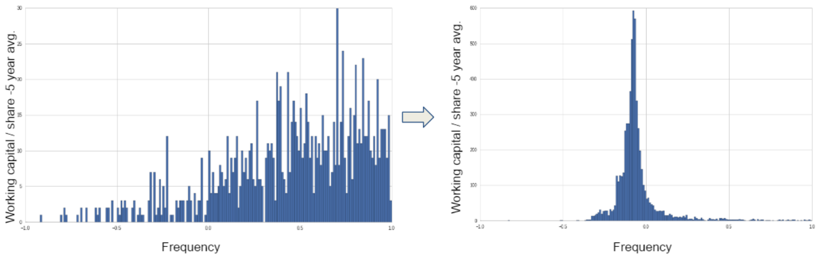


Figure 1: Processing raw feature data to obtain cleaner, more predictive signals.

1.1 Fundamental Data

Fundamental data is provided by Morningstar which is integrated with Quantopian. There are 13 categories of fundamental data and over 600 total metrics. To select the top fundamentals to use in our final model, we (1) trained random forest (and other) models on each category of fundamental data, (2) selected the top 10 to 15 features from each category by feature importance (feature importance will be described in Section 3.1: Feature Selection), and (3) retrained the models using the top features within each category to obtain the overall top features. The reason to obtain the top features within each category and then merging was because of a technical limitation to the number of features we could train at one time using the scikit-learn library.

1.2 Sentiment Data

The sentiment data was provided by PsychSignal which derives bullish and bearish scores from Twitter and StockTwits data. Categorization of whether an individual message is bullish or bearish is determined by PsychSignal's algorithm and a score is provided as to how bullish or bearish a message is. To obtain a clearer and more consistent signal, we took the 30-day average of the message scores.

1.3 Clinical Trial Data

The clinical trial data was provided by Quantopian as a .csv file containing the phases, outcomes, and indications of each clinical trial update. The data was too sparse as some companies were missing clinical trial events or did not have complete coverage, and loading the .csv file required a lot of computational overhead which caused the backtester to timeout, so we did not use the data as part of our model.

1.4 Stock Price & Volume Data

Features were also derived from the stock's price and volume data. Important features included the 30-day moving average of stock price and volume averaged over the past month, average volatility over the past month, and average return percentage over the past month.

2 Machine Learning Models

The types of machine learning models we explored include the random forest, extremely random trees, and logistic regression models. The machine learning models are used to predict short term returns and long term change in enterprise value. For the two tree-based models, we experimented with both regression and classification models (regression models predict continuous values whereas classification models predict class or category labels), and for logistic regression we experimented only with a classification model.

With the tree-based models, we tuned the models by modifying the depth of each tree, the total number of trees in the model, and the number of variables to choose from at each node split. Both of the tree-based models we experimented with are ensemble learning methods consisting of multiple decision trees. The individual subtrees of the random forest model are generated by optimizing for information gain where each node split is determined by the split resulting in the largest reduction of entropy.

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Figure 2: Entropy and information gain in a decision tree.

This is in contrast to the training of the extremely random trees model where each node is split randomly [2]. The final prediction of the ensemble is the mode or mean of the individual decision trees depending on whether we are modeling classification or regression, respectively.

In machine learning, there is a tradeoff between bias and variance. Bias causes the algorithm to miss relevant information embedded in the features (underfitting), and variance causes the algorithm to overcompensate for small intricacies or noise of the training data (overfitting). Decision trees typically suffer from overfitting (low bias, high variance) especially when the depth size or number of nodes is high. However randomness of the random forest, where the subset of features used to train each subtree is random, and extremely random trees, where the node splits are also random, help the overall ensemble achieve lower variance at the cost of slightly higher bias.

The machine learning models were used to predict the following: (1) long term changes in enterprise value (over six months) using signals from fundamental data, and (2) short term returns (over one month) using signals from the long term model and sentiment, clinical trial, price, and volume data. The next section further describes how we collected our training data. From our work, we found the random forest to perform the best so we spent the most time on random forests throughout our project. Extremely random trees were significantly faster to train and was comparable to the random forest but had higher variance. The logistic regression model did not perform well relative to the tree-based models even if we compared amongst classification models.

2.1 Training Data Collection

Collecting training data for the classification and regression models was similar. For the short and long term models, we used data from the past few years and computed the percentage change in stock price (returns) over one month and percentage change in enterprise value over six month time intervals, respectively. For regression models, the raw percentage change was used as the label, and for classification models we assigned labels of -1, 0, and 1 for when the stock price or enterprise value decreased by over 3%, stayed within 3%, and increased by over 3%; respectively; within the time interval. We empirically found the 3% threshold to be the most effective. All the training data was collected out of sample and used to predict future values in both the backtester and research mode. We also experimented with increasing the amount of training data by increasing how far back we collected training data from. For sentiment data, we were constrained because data is only available starting in 2010. The clinical trial data coverage was from 2007 to 2015, and we did did not end up using it due to its sparseness.

3 Residual Analysis and Forecast

Various approaches were used to assess each model before going into backtest runs and to get an understanding of the model.

3.1 Feature Selection

To understand the predictability of each feature, we calculated feature importance within the ensemble tree-based methods. We trained models and analyzed the feature importance which is a measure of the information gain given by each feature [3].

$$Imp(X_m) = \frac{1}{N_T} \sum_{T} \sum_{t \in T : v(s_t) = X_m} p(t) \Delta i(s_t, t)$$

Figure 3: Feature importance calculation.

The five most important fundamental features as shown below in Figure 4 were book value per share, market cap, book value yield, cash flow yield, and sales yield. While the feature importance ranking varied from sector to sector, these features generally made it to the top across most sectors.

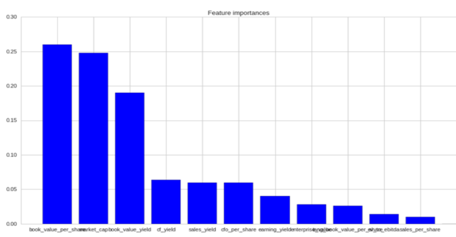


Figure 4: Feature importances for fundamentals on the Healthcare sector.

3.2 Model Assessment

To assess the model before going into backtests, we analyzed the (1) directionality accuracy of the model (e.g. if the model prediction and the actual outcome align in positive, negative, or neutral direction), (2) correlation between predicted and actual returns or change in enterprise value (R-squared), and (3) overall distribution of predicted vs. actual returns or change in enterprise value. These metrics are illustrated in Figures 5-7 below.

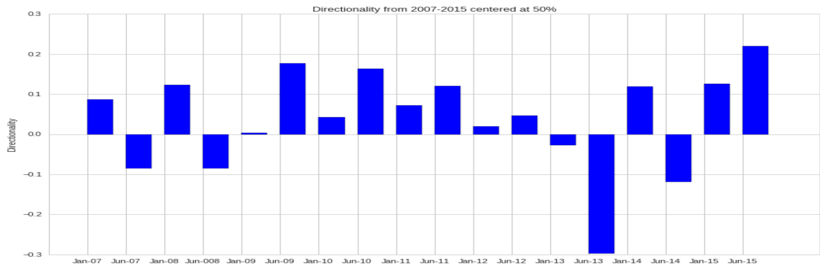


Figure 5: Directionality accuracy of the enterprise value long term model over time. In general, the model is directionally accurate for most months.

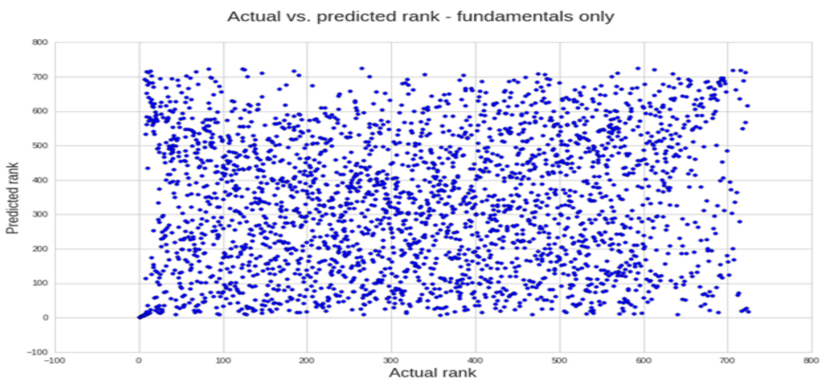


Figure 6: Correlation between predicted vs. actual short term returns. The correlation here is quite low with a R-squared value of 0.009.

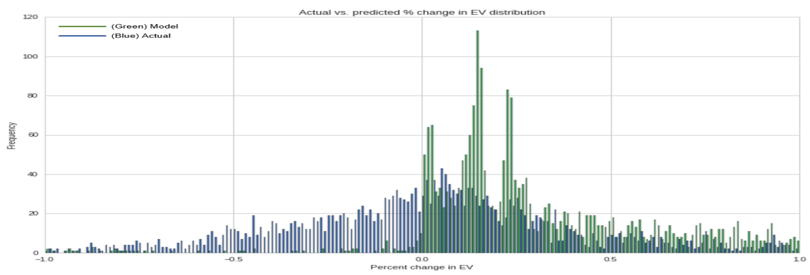


Figure 7: Overall distributions of predicted vs. actual change in enterprise value.

When tuning the machine learning models, we also leveraged the three metrics to find the best training parameters of the random forest. As we increased the number of trees and the tree depth, we found the correlation between predicted and actual enterprise value to increase and the overall distributions of predicted and actual enterprise value to also converge, as shown in Figures 8 and 9.

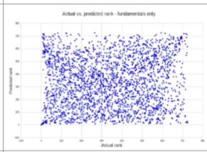
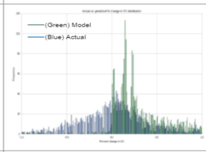
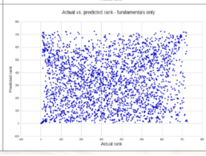
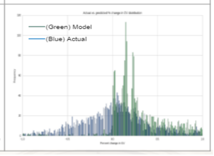
Model	Directionality	R^2	Slope	Int	Predicted rank vs. actual	Distribution of prediction
500 trees, depth of 15 (2014-2015 : N = 4)	57% +/- 11%	.009	.10	300		
1000 trees, depth of 15 (2014-2015 : N = 4)	57% +/- 11%	.009	.10	300		

Figure 8: Healthcare sector results for random forest models using 500 trees of depth 15 and 1000 tress of depth 15.

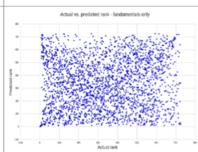
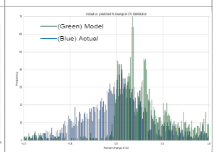
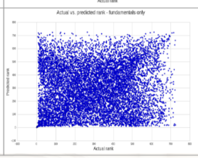
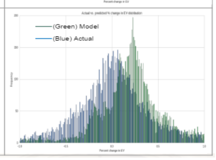
Model	Directionality	R^2	Slope	Int.	Predicted rank vs. actual	Distribution of prediction
1000 trees, depth of 20 (2014-2015 : N = 4)	56% +/- 11%	.012	.11	296		
2000 trees, depth of 25 (2007-2015 : N = 18) Run time: ~7 hours	54% +/- 12%	0.012	0.1	267		

Figure 9: Healthcare sector results for random forest models using 1000 trees of depth 20 and 2000 tress of depth 25.

Extending these models into the Technology sector, we obtained similar results as those of the Healthcare sector in Figure 10. The overall feature importances were also similar to those for the Healthcare sector.

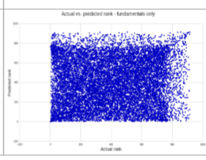
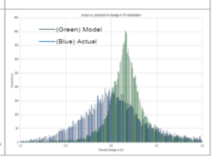
Model	Directionality	R^2	Slope	Int	Predicted rank vs. actual	Distribution of prediction
2000 trees, depth of 25 (2007-2015 : N = 18)	54% +/- 12%	.004	.07	373		

Figure 10: Technology sector results for a random forest model using 2000 trees of depth 25.

4 Core Model & Trade Execution

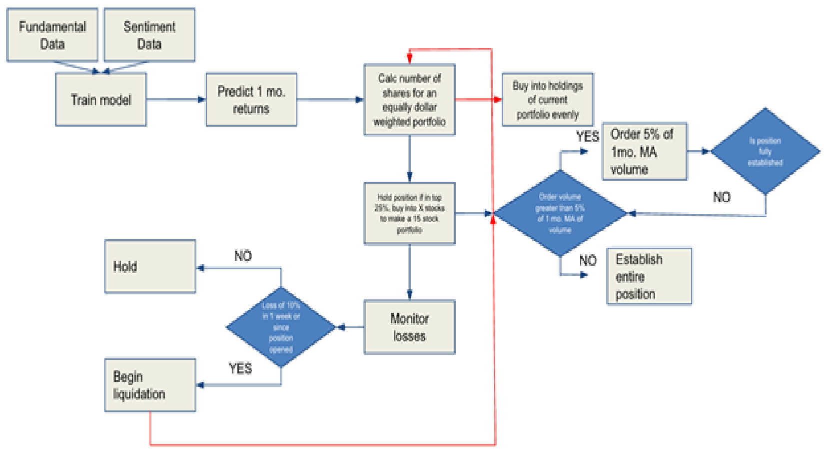


Figure 11: Flow diagram of the trading algorithm

We considered three distinct elements when executing our trades: signal generation, risk management and portfolio optimization. The algorithm implements these elements as follows (also illustrated in Figure 11):

4.1 Signal Generation

We use a machine learning model to make predictions. The input to the machine learning model is a set of fundamental and sentiment signals, and the return generated by a stock over one month periods (the time is flexible, as you will see in the multiple approaches we tried). We then train our machine learning to predict monthly returns in the future. At the start of every month, our model outputs a prediction of monthly returns expected in each stock, based on the signals observed in them. This drives the trade decisions - higher the prediction from the model, more we want to buy that particular stock.

4.2 Portfolio Construction

We consider the top 15 recommendations from the model and allocate capital equally between the stocks. In subsequent trades, if a particular stock is already in the portfolio, then we relax our criteria of top 15 for that stock, and include it as long as it falls within the top quartile of buy recommendations. The buying decision is also based on the liquidity available in the stock. Whenever we see small trading volumes in comparison to the amount we want to buy, we divide our execution over multiple days to avoid slippages (or failed execution) in the Quantopian backtest environment.

4.3 Risk management

If a particular stock in the portfolio performs poorly during the month (when the model is not making predictions), we trigger our stop loss and exit that particular position. Currently, the two triggers in place are - Loss of more than 10% in value in a week, or loss of more than 10% since taking the position. The intuition here is to avoid getting into traps where a stock falls significantly due to external reasons and we lose a lot of value before our model gets to the next month's prediction.

4.4 Future Improvements

If we were to develop this model further in the coming months, some important areas to improve on would be (also refer Section 7: Future work) - Dynamically decide number of positions to hold each month, and allocate capital based on the strength of the signal output from the prediction model. Dynamic stop loss determination, which the model can decide based on the sector, volatility and market capitalization of the stock. Manage portfolio risk through covariance analysis of existing positions. This could also serve as one of the features inside the prediction model. The feature would depend on the existing portfolio, and would aim to minimize risk in the portfolio. Index shorting can also be employed to hedge against risk.

5 Strategy Performance

The performance of our short-term monthly updating algorithm is shown below. We tested our algorithm by backtesting on the Pharmaceutical sector in the Quantopian platform and taking different combinations of the datasets available (twitter data, clinical trials data and fundamentals data). Although the long-term enterprise model is more rigorous, we are unable to implement this on the Quantopian backtest framework since it times out before the tests can be completed. Hence, we will just show performance results of the short-term monthly algorithm as shown in Figure 13. The 4 different datasets combinations that we tested on our algorithm are:

- Twitter data only
- Fundamentals data only
- Twitter data and fundamentals data only
- Twitter data and clinical trials data and fundamentals

	Alpha	Beta	Sharpe	Volatility
Twitter Only	-0.08	0.94	0.27	0.25
Fundamentals Only	0.27	0.93	1.54	0.28
Twitter and Fundamentals Only	0.12	0.82	1.1	0.24
Twitter and Clinical trials and Fundamentals Only	-0.04	0.83	0.43	0.25

Figure 12: Summary of the performance of different dataset combinations.

We see that if we just use the Twitter data, we do not achieve strong predictions, as we obtain a negative alpha and low sharpe ratio combined with a

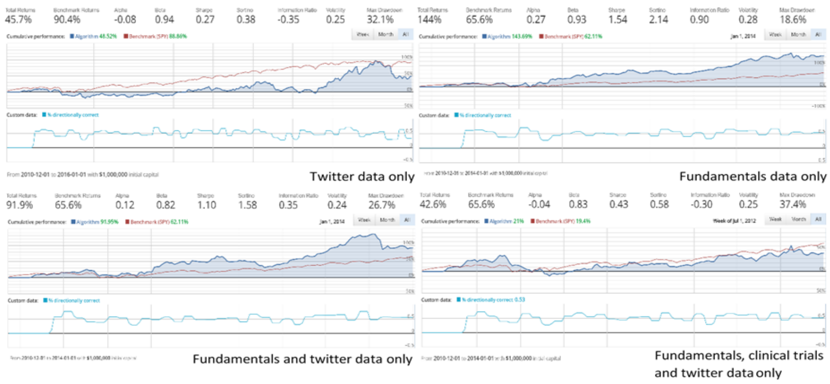


Figure 13: Quantopian backtests of the different dataset combinations.

relatively high beta. Using just the fundamentals data, we obtain a much higher alpha and sharpe ratio with the same beta.

We investigated and found that combining more datasets in our algorithm does not improve the predictions generated. We tested different combinations of the dataset, two of which are shown in the bottom of Figure . We see from the statistics in Figure 12 that adding additional datasets to the fundamentals data actually decreases alpha and sharpe without improving beta and volatility. We believe that this is because there is too much noise in the Twitter and clinical trials dataset, which results in the generation of a weaker predictive model (overfitting to the nuasances of the training data and noise).

From our results, it seems that only using the fundamentals data generates the best predictive model. It has a strong sharpe ratio of 1.5. However, it should be noted that past performance is not indicative of future results - all that glitters is not gold [5]. From the backtest, we see that the model essentially performed the same as the S&P 500 for prolonged periods of time and the outperformance was due to two stretches of spectacular returns. It is not known if this can be repeated in a future time period.

To see if this performance can be generalized to different sectors, we applied the fundamentals data only model onto the Technology sector. The backtest is shown in Figure 14. We can see that the performance of the model remains steady. This supports the notion that we have built a robust model that can be generalized to other sectors.

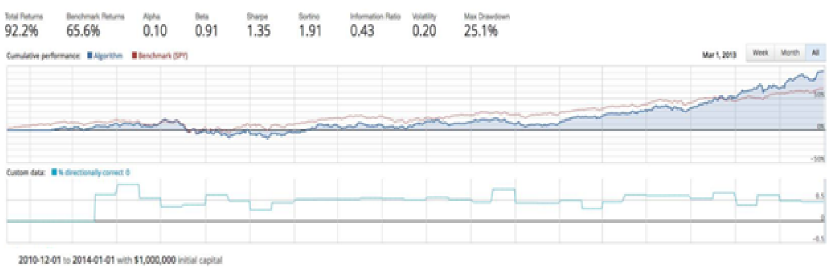


Figure 14: Fundamental data only backtest on Technology sector.

6 Summary

Using publicly available data, we have built an algorithm that generates mildly accurate predictions on the future stock price of companies. While we do not believe that substantial profits can be generated from the algorithm, we trust that it can be added to an investor's arsenal of indicators to generate better returns. It makes intuitive sense that any model based on publicly available fundamental signals to predict stock price performance has to be reasonably complicated and many easy predicted outcomes would already have been exploited by quantitative hedge funds to make returns. Most of these potential profits would then be arbitraged away by an efficient market.

If we were to undertake this project once again, we would have proceeded slightly differently to save a considerate amount of time. We would have focused on implementing the algorithms on the research environment instead of the backtest environment, which often had technical issues. We would have also spent less time on trying to build a model based on the clinical trials dataset, which was ultimately unsuccessful due to the technical capacity of the Quantopian backtest environment.

Overall, we are pleased to better understand the space of fundamental based algorithm trading and to have developed a mechanism to check future algorithms for robustness, rather than depending on only backtest runs. In the future, we would like to explore further and hope to develop more predictive power in our algorithms.

To deploy our model and system into the real world, Quantopian allows for algorithms to go live with a linked Robinhood or Interactive Brokers account. After some adjustments, we would be interested in deploying our system into a Quantopian contest or for live trading.

7 Future Work

While we have accomplished much during the 10-week project, we believe that there is still ample room for further exploration in the journey of developing a consistent high sharpe ratio algorithm. We have identified three areas that we believe could be further explored given a longer project timeframe.

One improvement that can be made to our project is to perform better data pre-processing. Specifically, we believe that there is substantial room for improvement in the way we treat gaps in the dataset. Currently, we have replaced all gaps with -2, an arbitrary number. This does not really make sense for some features, such as market capitalization, as they cannot realize a negative number. Additionally, replacing substantially different rows that have the same gap with the same constant can seriously distort the regression results. To improve we can use machine learning algorithms such as k-nearest-neighbors and local regression to improve our prediction of these values based on the values of the rest of the features. This is not a simple problem, as we also need to figure out how to account for the fact that different rows have different amounts of data missing on different features.

Another improvement that can be made is to look at additional machine learning algorithms. During the design process, we looked at several models, including linear regression, interaction terms and random forests. From the test results, we see that of all the machine learning techniques we have implemented a tuned random forest predictor performs the best. However, there are plenty of other machine learning algorithms that we can try implementing, such as Neural Networks and Support Vector Machines. We can also consider building our own adaptation of a machine learning algorithm using a unique loss function.

Given substantially more time, we also believe it would be prudent to build our own research platform to allow us to obtain substantially greater processing power for our algorithms. Currently, the Quantopian backtest environment will time out after 6 minutes. Additionally, since scripts are run on the internal Quantopian server, both the backtest environment and the research environment run (very) slowly with no conceivable method of speeding up. We frequently passed this 6 minute time limit and in fact had to abandon using the clinical trials dataset because the Quantopian framework took too long to process the input file. This is a problem, as our tests have shown that our algorithms show slight improvements in predictive ability as the depth of our trees increases. With our own research platform, which we can create by extracting the necessary data from Quantopian, we are given substantially more freedom to generate more complex learning algorithms. For example, we can quickly implement the first two points above by paralleling the computations on our own server (otherwise it could take a substantial amount of time).

References

[1] K. Daniel and S. Titman. "Market Reactions to Tangible and Intangible Information". In: *The Journal of Finance* (2006).

[2] Pierre Geurts, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees". In: (2006).

[3] Gilles Louppe et al. "Understanding variable importances in forests of randomized trees". In: (2013).

[4] J. Piotroski. "Value Investing: The Use of Historical Financial Statement Information to Separate Winners from Losers". In: *Journal of Accounting Research* (2000), pp. , 38, 1–41.

[5] Thomas Wiecki et al. "All that Glitters Is Not Gold: Comparing Backtest and Out-of-Sample Performance on a Large Cohort of Trading Algorithms". In: (2016).

8 Appendix - Valuation Methodology

Fundamental analysis and technical analysis are two major schools of practice for evaluating stocks. Fundamental analysis is a technique that attempts to determine a security's intrinsic value by examining underlying factors that affect a company's actual business and its future prospects. These factors can be both qualitative and quantitative, concerning macroeconomic, such as the overall economy and industry conditions, as well as company-specific factors, like financial condition and management. Fundamental analysis helps estimate the intrinsic value of the firm, which can be used to compare against the security's current price, based on which, long short positions are taken accordingly (if underpriced, long; if overpriced, short) In general, there are two ways of conducting fundamental analysis

8.1 Method 1: Discounted free cash flow model

$$Stock\ Price = \frac{Market\ Value\ of\ Equity}{Number\ of\ Shares\ Outstanding}$$

$$= \frac{Enterprise\ Value\ +\ Cash\ -\ Debt}{Number\ of\ Shares\ Outstanding}$$

Where the current enterprise value is

$$V_0 = \frac{FCF_1}{1 + r_{WACC}} + \frac{FCF_2}{(1 + r_{WACC})^2} + ... + \frac{FCF_N}{(1 + r_{WACC})^N} + \frac{V_N}{(1 + r_{WACC})^N}$$

And

$$FCF_N = Free\ Cash\ Flow\ in\ Year\ N$$

$$V_0 = Current\ Enterprise\ Value$$

$$r_{WACC} = Discount\ Rate\ Determined\ by\ Weighted\ Average\ Cost\ of\ Capital$$

$$V_N = Enterprise\ Value\ in\ Year\ N$$

Usually the terminal value is estimated by assuming a constant long-run growth rate for free cash flows beyond year N. Therefore,

$$V_N = \frac{FCF_{N+1}}{r_{WACC} - g_{FCF}} = \Big(\frac{1 + g_{FCF}}{r_{WACC} - g_{FCF}}\Big) * FCF_N$$

Consequently, current enterprise value can be estimated by

$$V_0 = \frac{FCF_1}{1 + r_{WACC}} + \frac{FCF_2}{(1 + r_{WACC})^2} + ... + \frac{FCF_N}{(1 + r_{WACC})^N} + \Big(\frac{1 + g_{FCF}}{r_{WACC} - g_{FCF}}\Big) * FCF_N$$

The factor with greatest uncertainty here is future free cash flow. By definition, free cash flows are derived directly from financial statements. Therefore, a firm's fundamental signals, such as revenue, cost and so on can be expected to have significant influence over the firm's enterprise value, and thus on stock price.

8.2 Method 2: Method of Comparables

The Law of One Price implies that two identical firms should have the same value. Therefore, another commonly used technique for estimating the value of a firm is comparing against other firms or investments that we expect will generate very similar cash flows in the future.

Common multiples that we look at for comparison purposes are Price-Earnings ratio, Enterprise value to EBITDA multiple and Price to Book value multiple per Share, etc.

The Comparables method makes more sense especially for firms in the same sector. Indeed, when constructing a portfolio, there is no need to estimate exact stock prices. Comparing fundamentals should give us a good sense of the relative ranking of the firms, which could guide us to take long, short positions accordingly.

In literature review, Piotroski, J. (2000) demonstrated such success. He selected a pool of stocks that has high book-to-market ratio. Then he forecasted return, bought expected winners and shorted expected losers. This strategy generated a 23% annual return between 1976 and 1996 [4].

However, we need to bear in mind that whatever data we already have only reflects the past. When predicting for the future, there is usually a "mean reversion" effect. That is to say, a top-ranking stock at present is likely to become average in the future. Therefore, holding this stock might not be profitable. This is discussed by several scholars, especially by Daniel, K. and Titman, S. (2006) [1].

In sum, the predictability of fundamental signals is still under debate. We took up this challenge with two objectives in mind – validating this hypothesis and building a robust trading algorithm. We looked back to previous scholars' work. When Piotroski, J. (2000) conducted his experiment, he manually selected financial performance signals on profitability, leverage, liquidity and source of funds, as well as operating efficiency. Then he used a composite score to evaluate each stock.

With today's technology, we think the aforementioned method can be improved. Firstly, by using a machine learning algorithm, we can utilize all signals without imposing pre-selection bias. We can also validate if the predictive power does exist by letting the computer extract a model from past performance and testing it with new datasets. Secondly, with drastically increased computational power, we can utilize not only financial data, but also other data series, such as events and sentiment, which are closely related to stock price. Since there are industry-specific trends and characteristics, we divided all stocks according to industry sectors, and ran algorithm separately.