29 April 2022

Announcement Date Investing with Machine Learning

Oxford Alpha Fund - Quantitative Strategies

Contents

Abstract

In this project, we extend previous work [1] on announcement day strategies based on the Earning Announcement Premium. Specifically, we investigate Machine Learning (ML) on corporate announcements with the following changes: we use announcements dates as a feature instead of a strict precondition for investment, we added features such as P/B and return in previous announcement months inspired by the predictive factors in the literature, and used a more complex ML model.

1 Literature Overview

1.1 Corporate Earnings Announcements and Equities

All public companies are required by law to publish their earnings periodically to keep investors informed about the state of their investments. For the US, this is regulated by the SEC and earnings reports are expected to be published for each fiscal quarters and the fiscal year end. Accompanying these reports are often press releases which boil down the report contents and provide additional company sentiments on its outlook in the form of a digestible summary. Announcements from large companies or expected to be impactful often are reported on by prominent financial new sources such as The Wall Street Journal or The Financial Times.

Abnormal returns, sometimes referred to in the literature as the ”Earning Announcement Premium” (EAP), around earning announcements is a well documented phenomenon, with some of the earliest studies dating back at least to Beaver’s 1963 paper. In many subsequent papers, it has been shown that this phenomenon is both statistically significant and substantial. In one study it was presented that monthly trading strategies based on EAP produced 7 - 18 % annual returns and Sharpe Ratio higher than other anomalies.[2] And in another paper, it was presented that 16 % of annual returns occur in the 2 days prior to announcements for select small companies.[3] Clearly these findings present potential investment opportunities, if they can be well understood and predicted.

Although the announcement phenomenon is well documented, the predictive factors and causes of this phenomenon remains an area of active research with competing different schools of thoughts.

One school of thought argued the premium arise from the compensation for the disclosure of information and associated risk from the announcements and subsequent market adjustments. as different sectors would have different degree of information availability; Following this school; Chari et al. (1988)[3] investigated the EAP based on company size, La Porta et al. (1997)[4] investigated EAP between ”value” and ”glamour” stocks classified using market-to-book, cash-flow-to-price, and growth rate metrics. Hsieh et al. (1999)[5] found increases in the daily beta values around announcements; Cohen et al. (2007)[6] explored and argue for an idiosyncratic risk explanation; whilst Barber et al. (2013)[2] explore the EAP in different international markets and argues the effects is larger in countries with more idiosyncratic volatility.

Another and more recent school focused on the ”attention grabbing effect”, where it is hypothesises attentions creates disproportional buying pressure from individual investors. Lee (1992)[7] found that there is abnormal increase in small investors buying around investments regardless of the signal of the announcement. Frazzini et al. (2007)[8] Investigated the relationship between abnormal trading volume, and suggest a strong relationship, Whilst Kimball (2018)[9] tries to reconcile the two schools investigating at announcement notifications, suggesting that different models dominates different sections of EAP phenomenon.

Whilst there are multiple different schools of thoughts, the evidence is not conclusive and sometimes contradictory, for example Cohen et al. (2007)[6] found that the volume relationship is sensitive to the definition of ”abnormal volume” whilst Barber et al (2013)[2] suggested pre-announcement trading volume is negatively associated with EAP.

In this investigation, we then aim to investigate the viability of a EAP based strategy by predicting abnormal returns from the announcement phenomenon using past data and machine learning techniques. Previously we have already began investigations under this approach, using Monthly Market Caps, Price and Beta as our predicting factors, and used simple machine learning techniques such as linear and logistic regression, Naive Bayesian approaches and have found some success.[1]

One advantage of our machine learning approach is that it enables us to be agnostic to the different schools of thoughts. Building upon this prior work we then aim to seek improvement by collating a larger range of potential predictive factors for EAPs from the literature and using more complex machine learning models by using Neural Networks (Multi Layer Perceptrons).

2 Methodology

2.1 DataFrame Generation

We Generated different input DataFrames for our Machine Learning models by selecting combination of different prediction factors or features ”inspired” by the EAP Literature. We did not use the exact predictor factors as our input features as we have had some difficulties collecting some of the raw data needed.

We first collected raw data of Market Cap, Market Price, Beta,Market Price to Book Ratio Announcement Dates, Performance and Announcement Surprise of constituent stocks of the S&P 500 Index form January of 2001 to January of 2021. Membership of the S&P 500 Index is sampled yearly on the last trading day of the year. Most other raw data is sampled monthly at the start of the month, whilst Announcement Dates, Performance and surprise is sampled when it has occured. All data is collected from a Bloomberg Terminal.

We then generate the following features from the raw data: **Past N months % difference**, which is the percentage difference of a value of the current month and N months prior; **Value N Months or Announcements Prior**, which is the value of the feature but N month or Announcements ago; **Average over past N Months**; **Value Abnormality**, measure the % change from the average over the past 12 months; And finally this is used to predict the next month returns, which is the percentage difference of the price between the current month and 1 month later. Among the data set a **Announcement Present** feature was also included which denotes the presence of at least 1 announcement in the up-coming month which is represented by a binary value. (Note that these operations was combined and applied to different raw data features and see Appendix 5 for calculation details.)

In our experiments, we gradually increase the amount of feature included in our DataFrames to observe their impact. Tables 1-3 below detail all the DataFrames that have been generated and the features that they contain.

2.2 Machine Learning Models Hyper-parameter Selection

Our Methodology is then as such, We then split all generated Dataframes into 3 sub-DataFrames, training (60%) (Jan 2000 - Feb 2013), testing (20%) (Mar 2013 - Jul 2017) and Results sub-DataFrames (20%) (Aug 2017 - Dec 2021).

DataFrame version	Description
DataFrame: V1	Basic DataFrame
	Market Cap
	Beta
	1 Month Return
	2 Month Return
	3 Month Return
DataFrame: V2	Expanded with Announcement Data
	All Features in **V1**
	Announcement Present
	1 st Past Announcement Performance
	1 st Past Announcement Surprise
DataFrame: V3A1	Expanded with More Past and Announcement Data
	All Features in **V2**
	4 Month Return
	5 Month Return
	2 nd Past Announcement Performance
	2 nd Past Announcement Surprise
	3 rd Past Announcement Performance
	3 rd Past Announcement Surprise
DataFrame: V3A2	Version investigates inclusion of 6 Month Return
	All Features in **V3A1**
	6 Month Return

Table 1: **V1-V3** DataFrames

The Training sub-Dataframe is then input into a Machine Learning Trainer. The training data is further split up into 5 sections for Cross Validation, with sklearn's "TimeSeriesSplit" scheme [10] which trains on the first k fold and cross validate on the $k+1$ section until the last fold (5) has been cross validated.

Hyper-parameters are searched in a grid like fashion for different MLP geometries {(100); (100,50); (100,50,30); (100,50,30,30)}, where each number denotes the number of ReLu "neurons" in each hidden layer and "Alpha" {0.00001; 0.0001; 0.0005; 0.001; 0.01; 0.1}, a scale factor of the magnitude of the MLP parameter weights in the cost function. The best parameters are then fitted with the prediction returning predictions for Next Month Returns.

Our strategy then selects the top n Stocks (in our case n = 6) with the highest non-negative predicted

DataFrame version	Description
DataFrame: V4A1	Expanded with Abnormal Returns and Annual Data All Features in **V3A1** 12 Month Return 12 Months Average Return 1 st Past Announcement Abnormal Return 2 nd Past Announcement Abnormal Return 3 rd Past Announcement Abnormal Return 4 th Past Announcement Performance 4 th Past Announcement Surprise 4 th Past Announcement Abnormal Return
DataFrame: V4A2	Version investigates inclusion of 6 Month Return All Features in **V4A1** 6 Month Return
DataFrame: V4B1	Inclusion of Price to Book Ratio All Features in **V4A1** Price to Book Ratio
DataFrame: V4B2	Version investigates inclusion of 6 Month Return All Features in **V4B1** 6 Month Return
DataFrame: V4C1	Expanded with Abnormal Annonucemnt Beta All Features in **V4B1** 1 st Past Announcement Abnormal Beta 2 nd Past Announcement Abnormal Beta 3 rd Past Announcement Abnormal Beta 4 th Past Announcement Abnormal Beta
DataFrame: V4C2	Version investigates inclusion of 6 Month Return All Features in **V4C1** 6 Month Return

Table 2: **V4** DataFrames

DataFrame version	Description
DataFrame: V5	Expanded With Past Yearly Data to investigate periodicity
	All Features in **V4B1** **Replace** Beta with 12 Months Average Beta 6 Month Return 1 Month Return 1 Year Ago 1 Month Return 2 Year Ago 1 Month Return 3 Year Ago 8 th Past Announcement Performance 8 th Past Announcement Surprise 8 th Past Announcement Abnormal Return 8 th Past Announcement Abnormal Beta 12 th Past Announcement Performance 12 th Past Announcement Surprise 12 th Past Announcement Abnormal Return 12 th Past Announcement Abnormal Beta

Table 3: **V5** DataFrame

returns, and weight them proportionally to the magnitude of their predicted Returns. Different to our prior work[1], we do not limit the selection of stocks to those that has an announcement in the coming month.

We then observe the performance of each Dataframe on the testing data set, to determine the best performing feature set. Finally we further test the different Dataframes on the reserved ”results” sub-dataframe to validate our findings.

To evaluate the perforamnce of each Dataframe in the Test and Result sets, We consider 3 performance metrics; Mean Annual Returns, Sharpe Ratio and Maximum Draw down as a combined score, with and without transaction costs (we selected 5 bps commission for large-cap US stocks, which is slightly higher than data [11] see the appendix 5 for calculation details). We compare our stratergies against an equal weighted SPX 500 as a baseline. The combined score metric is calculated as

$$Score = \frac{MeanAnnualReturns \times SharpeRatio}{MaximumDrawDown}$$

to provide a simple unified comparison value.

3 Results

DataFrame Version	Mean Annual Returns	Sharpe Ratio	Maximum Draw Down	Combined Score
V4B1	0.24454	1.79110	0.08619	5.08191
V4C2	0.27143	1.70870	0.15561	2.98045
V4C1	0.23296	1.60827	0.08437	4.44077
V4A1	0.20857	1.42326	0.11230	2.64329
V4A2	0.17762	1.38749	0.09591	2.56953
V1	0.17408	1.24895	0.09203	2.36242
V4B2	0.32144	1.23773	0.21870	1.81920
SPX	0.12671	1.23745	0.11063	1.41733
V5	0.23539	1.04551	0.34249	0.71856
V3A2	0.09852	0.96437	0.09940	0.95584
V3A1	0.10155	0.74863	0.17439	0.43594
V2	-0.01021	-0.02702	0.72761	0.00038

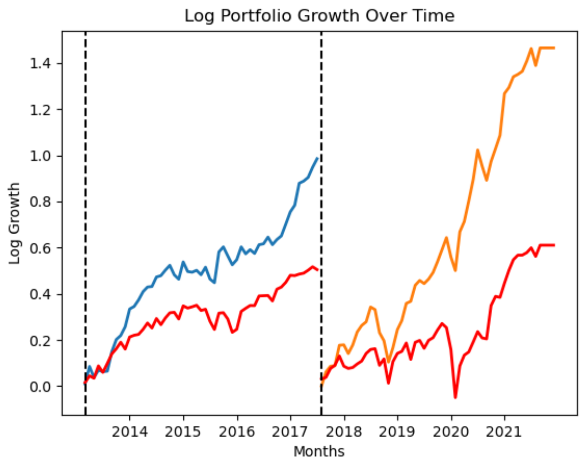
Table 4: Performance in the Test Period (Mar 2013 - Jul 2017) without transaction costs, sorted by Sharpe Ratio

From table 5 we can observe that most of the DataFrames outperformed our baseline S&P 500 Index (**SPX**), but increasing the number of features in our DataFrames did not always improve the performance (this can be show comparing figure 1c and 1d)

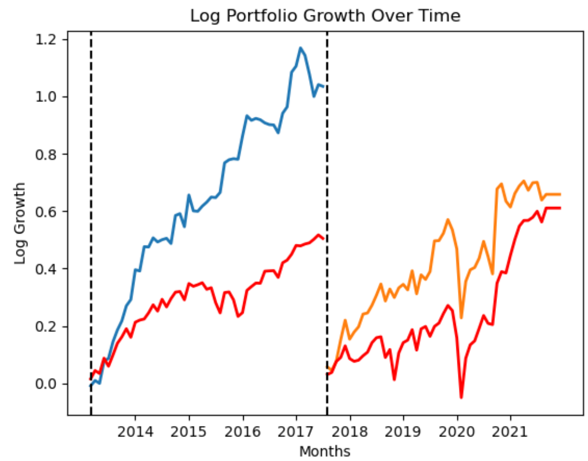
DataFrame Version	Mean Annual Returns	Sharpe Ratio	Maximum Draw Down	Combined Score
V4B1	0.42141	1.97523	0.21217	3.92321
V4C2	0.17113	**0.67607**	0.28991	0.39909
V4C1	0.31887	1.56790	0.19332	2.58616
V4A1	0.22689	0.85749	0.27443	0.70895
V4A2	0.18603	0.94009	0.19415	0.90076
V1	0.35528	1.69447	0.19841	3.03412
V4B2	0.17645	**0.63446**	0.23050	0.48569
SPX	0.15782	0.81971	0.27449	0.47130
V5	0.06241	0.18715	0.45187	0.02585
V3A2	0.25281	**1.04651**	0.34802	0.76020
V3A1	0.15173	0.75285	0.21228	0.53810
V2	0.30229	0.44646	0.46183	0.29223

Table 5: Performance in the Results Period (Jul 2017 - Dec 2020) without transaction cost, sorted by Test Period Sharpe Ratio. Red and Green denotes decreases and increases in Sharpe Ratio; **Bold** denotes changes relative to the SPX

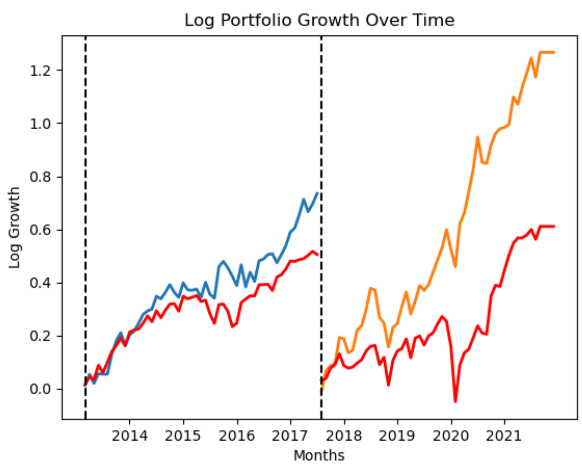
Table 4 then shows the performance metrics of these DataFrames from the results period, although some DataFrames sees a significant drop (see figure 1b) in Sharpe Ratio and there is decrease in performance overall , findings from the test periods are reinforced and the top performing DataFrame remained consistent (V4B1). furthermore, in this period it is more clear that the introduction of the 6 month returns seems to have a significant impact on performance (comparing DataFrames ending in 1 vs 2, i.e. V4C1 vs V4C2 etc.).



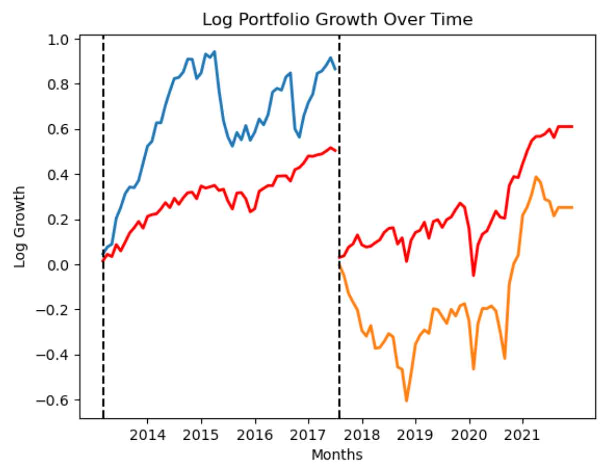
(a) Best performing DataFrame V4B1



(b) V4C2, drop in performance in the results period

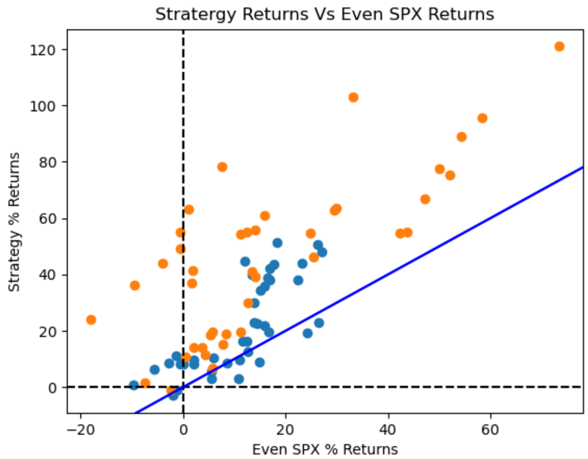


(c) V1, good performance for the simplest DataFrame

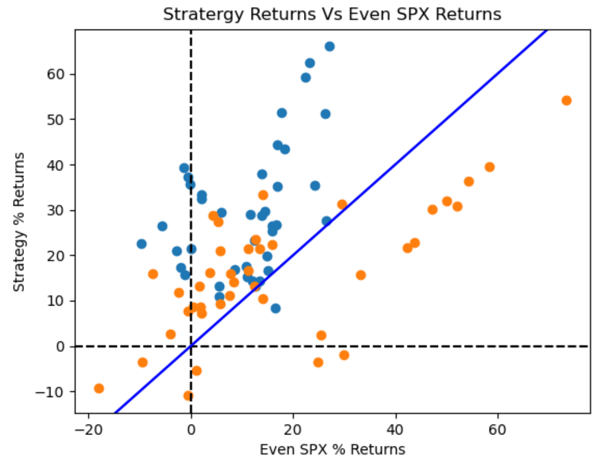


(d) V5, poor performance for the most complex DataFrame

Figure 1: Scatter Plots of the Portfolio Returns against Equally Weighted S&X 500 Returns, blue from the Test Period, Orange from the Results Period

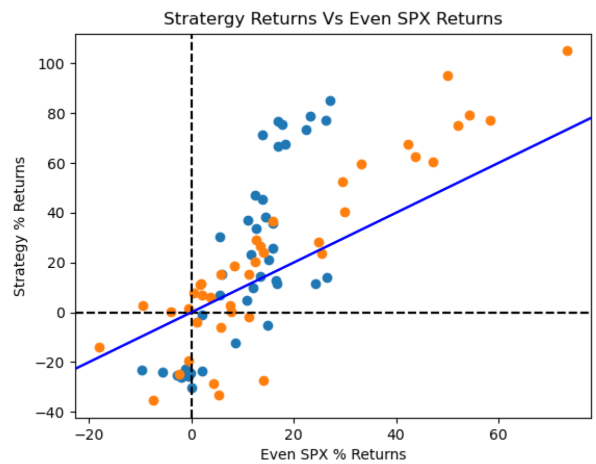


(a) Best performing DataFrame V4B1



(b) V4C2, drop in performance in the results period

(c) V1, good performance for the simplest DataFrame



(d) V5, poor performance for the most complex DataFrame

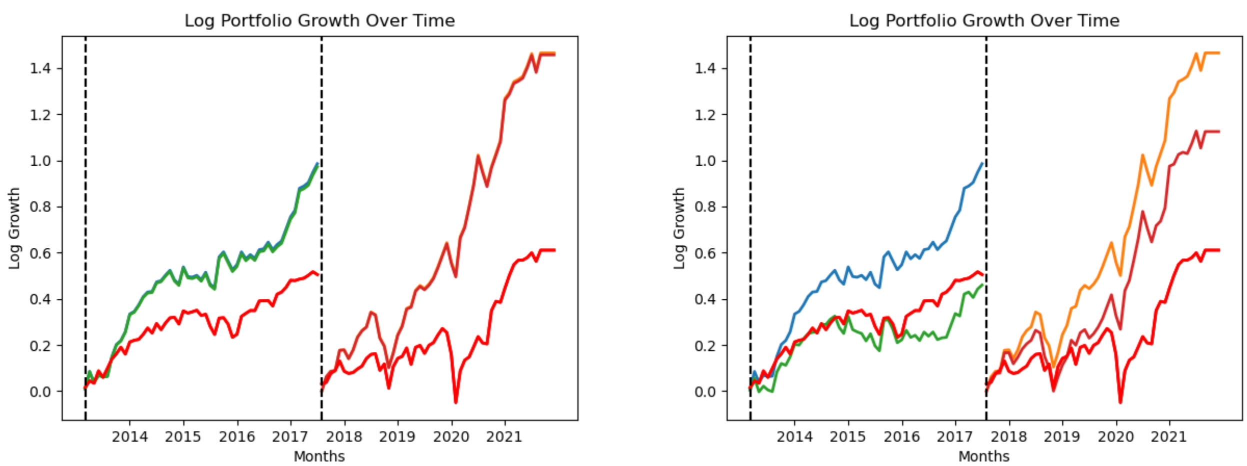
Figure 2: Scatter plots of portfolio returns against Equally Weighted S&X 500 (Red)

Finally, we have also conducted a Fama-French 3 Factor model analysis of the stratergies for the Overall Period (Mar 2013 - Dec 2020) with table 7 showing the results, where our best strategy in the Test period achieving an annualised alpha of 15%.

Focusing on the Best DataFrame and demonstrating the effects of Transaction Costs Table 6 and figure shows that our Strategy are very robust to transaction costs.

Test Period	Mean Annual Returns	Sharpe Ratio	Maximum Draw Down	Combined Score
0 bps	0.24454	1.79110	0.08619	5.08191
5 bps	0.2409414	1.7646193	0.0875325	4.85728
200 bps	0.1073999	0.7633943	0.1394095	0.58811
Results Period	Mean Annual Returns	Sharpe Ratio	Maximum Draw Down	Combined Score
0 bps	0.42141	1.97523	0.21217	3.92321
5 bps	0.4185351	1.9621350	0.21266694	3.86154
200 bps	0.3097653	1.4478250	0.23188350	1.93410

Table 6: Performance Measures of V4B1 DataFrame with Different Transaction Costs



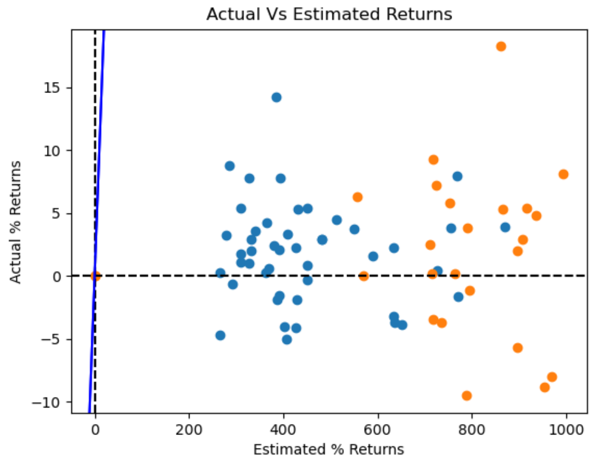
(a) V4B1 under 5bps (Green & Maroon) (b) 4B1 under 200bps (Green & Maroon)

Figure 3: Plots of the Natural Log of Cumulative Returns against Equally Weighted S&X 500 (Red)

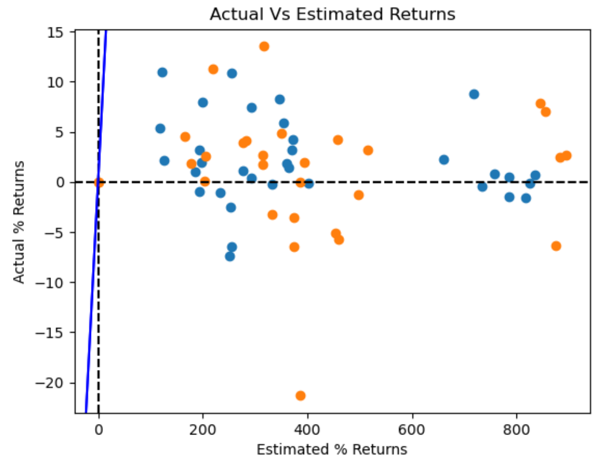
DataFrame Version	Alpha	Index Risk Premium	SMB Beta	HML Beta	R2 Score	Annualised Alpha
V4B1	0.012327968	0.788862195	0.236044123	-0.031164799	0.11994963	0.158390068
V4C2	0.005832112	0.949460635	-0.088329182	0.117543397	0.160029084	0.072274457
V4C1	0.011039579	0.758971816	0.303728999	-0.104742325	0.119275001	0.140821996
V4A1	0.005554784	0.964588206	0.155568089	0.134033696	0.137222649	0.068732062
V4A2	0.005447249	0.782546234	0.082510158	-0.042850597	0.098608626	0.067361373
V1	0.010344042	0.788838099	0.128103288	-0.084143393	0.139032374	0.131439711
V4B2	0.006475027	1.078873775	-0.256042507	0.07580503	0.319703332	0.080528041
V5	0.004430206	1.066656054	0.299969069	0.249927305	0.276650748	0.054477157
V3A2	0.002205547	0.701398733	0.046999643	0.025264682	0.068441015	0.026789989
V3A1	0.002007583	0.879875922	0.013476103	0.145596322	0.073141004	0.02435879
V2	-0.011195403	2.501734963	0.144247948	-0.086899603	1.090713744	-0.126373656

Table 7: Fama-French 3 Factor Model Analysis of different DataFrames (Mar 2013 - Dec 2020)

Although the results above seems very promising there are also some concerns from our results. figure 4 shows that the machine learned predicted portfolio returns often have little correlation with the actual portfolio returns, as well as different machine learning initialisation often very result in different results.



(a) Best performing DataFrame V4B1



(b) V4C2, drop in performance in the results period

Figure 4: Plots of the Natural Log of Cumulative Returns against Equally Weighted S&X 500 (Red)

Looking closer at Table 8 which shows the relationship between performance metrics, we also find that of the originally 3 metrics only Maximum Draw Down demonstrated a significant correlation. Although surprisingly then the combined score metric shows a strong and significantly correlation.

	Mean Annual Returns	Sharpe Ratio	Maximum Draw Down	Combined Score
Correlation	-0.113763932	0.532187606	0.773441172	0.766103754
P Vlaue	0.253618921	0.133053713	0.03642213	0.007710871

Table 8: Correlation and P value of different metrics between the Test and Results Periods

4 Conclusion

Under our approach, we have found promising results robust to transaction costs, with the best results obtaining 26% excess returns compared to an equally weighted S&P 500 Index of the same period and Sharpe Ratio of 1.96 under 5bp transaction costs between Jul 2017 - Dec 2020. It has also achieved a 15% alpha under the Fama-French 3 Factor Model over the period Mar 2013 - Dec 2020.

However, Although our best DataFrame is the same in both periods, at large there is no correlation between Sharpe ratio across the periods, which could present an issue when selecting a suitable DataFrame. Whilst inconclusive, it is very interesting then that the ”Combined Score” metric showed a strong and significant correlation, as it could bypass this problem.

Finally and surprisingly, given the generally promising results, we observe that there is little to no correlation between our predicted portfolio returns and actual portfolio returns. Together with concerns over convergence, they represent the remaining risk and challenges to the otherwise promising result.

5 Appendix

Past N months % difference:

This is the percentage difference between the value of a feature at current month k and a N prior months ago $k - N$

$$R_k^{[N]} = \frac{(P_k - P_{k-N})}{P_{k-N}}$$

this is mostly used for Monthly Price and Returns, in which case will be refereed to as "N Months Return"

Value N Months or Announcements Prior:

This is the value of any data fields, that is from N Announcement ago. as Announcement are usually quarterly, multiples of 4 denotes values of the same announcement a year ago.

Average over past N Months:

This should be self explanatory and is simply the average value of the past N periods

$$M_k^{[N]} = \frac{1}{N}\sum_{i=0}^{N}(V_{k-i})$$

Value Abnormality:

This denotes how much does the value of any field deviate from a 12 Months Average. this is calculated as the percentage change from the average:

$$A_k^{[12]} = \frac{V_k - (\frac{1}{12}\sum_{i=0}^{12}(V_{k-i}))}{\frac{1}{12}\sum_{i=0}^{12}(V_{k-i})}$$

Data N Months Prior

This is should also be self explanatory as the value of the same field but N Months ago

Next Month Returns

This is the target of our machine learning model, and is also the output. This denotes the Percentage

Returns that we be obtained if we held said stock for the period of 1 Month.

$$R_{nm} = \frac{(P_{k+1} - P_k)}{P_{k+1}}$$

Transaction costs

To incorporate transaction Costs into our calculations and our prior work [1] [12], we define a weight matrix W and returns matrix R, both in *{Dates x Tickers}* format, such that the transaction cost-adjusted returns R_{adj} are given by,

$$R_{adj} = W^T R - k \sum_{i=1}^{n} |\Delta W|. \tag{1}$$

For default analysis, we choose k = 5 bps commission for large-cap US stocks.

6 References

[1] Adrian Penz, Daniel Gagliardi, Peiyu Liu, and Shengwei Lu. Oxford alpha fund: Announcement date project. 2022.

[2] Reuven Lehavy Brett Trueman Brad M. Barber, Emmanuel T. De George. The earnings announcement premium around the globe. *Journal of Financial Economics*, 108(1):118–138, 2013.

[3] Aharon R Ofer V.V Chari, Ravi Jagannathan. Seasonalities in security returns: The case of earnings announcements. *Journal of Financial Economics*, 21(1):101–121, 1988.

[4] Andrei Shleifer Robert Vishny Rafael LaPorta, Josef Lakonishok. Good news for value stocks: Further evidence on market efficiency. *The Journal of Finance*, 52(2):859–874, 1997.

[5] William Kross Su□Jane Hsieh, Scott I. Jerris. Quarterly earnings announcements and market risk adjustments. *Journal of Business Finance Accounting*, 36(3-4):313–336, 1999.

[6] Daniel A. Cohen, Aiyesha Dey, Thomas Z. Lys, and Shyam V. Sunder. Earnings announcement premia and the limits to arbitrage. *Journal of Accounting and Economics*, 43(2):153–180, 2007.

[7] Charles M.C Lee. Earnings news and small traders: An intraday analysis. *Journal of Accounting and Economics*, 15(2):265–302, 1992.

[8] Andrea Frazzini Owen Lamont. The earnings announcement premium and trading volume. NBER Working Papers 13090, National Bureau of Economic Research, Inc., `https://ideas.repec.org/p/nbr/nberwo/13090.html`, 2007.

[9] Chapman Kimball. Earnings notifications, investor attention, and the earnings announcement premium. *Journal of Accounting and Economics*, 66(1):222–243, 2018.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] Anastasia Petraki. What is behind transaction cost figures and how to use them. *Schroders*, 2020.

[12] Peiyu Liu, Oliver Hayman, Thomas Orton, Jerry Soong, and Raymond Zhao. Cross-sectional momentum trading with portfolio weighting, machine learning and hidden markov models. 2022.