Financial Ratios, Machine Learning and Stock Picking

Steve Tulig

September, 2022

Introduction

This study examines financial ratios and machine learning models based on these ratios for their ability to predict stock returns and generate profitable trading opportunities amongst the largest 300 stocks on the Australian Securities Exchange (ASX). The study finds that whilst several financial ratios have some ability to predict future stock returns, the machine learning models fitted in this study and based on those ratios fail to improve the ability of a money manager to trade off the same information[1]. A less sophisticated model based on a subset of the ratios outperformed all the machine learning models without the same amount of computational effort. The study suggests that practitioners need to be cautious in their implementation of and confidence in stock-picking machine learning models.

The study is based on ASX and associated financial data retrieved by the author in 2014 and used at the time to write Python applications for backtesting equity investment strategies. The Python code used in this study and database details are available on the author's GitHub page[2].

The study has some limitations. As noted above, the data used has become somewhat dated, and a replication of the study with more recent data is in order. Several other promising ratios were not included because of time constraints, for example accruals, the Merton (distance-to-default) model and valuation-related measures based on analysts' forecasts. Finally, the study was limited to six of the most common machine learning algorithms; it could possibly be argued that even more-sophisticated models (e.g., deep learning models) might have overcome the failings of the six models used.

Data and Methodology

The data in this study is initially from Morningstar; it includes financial (annual report) data from the period 2000 to 2012, market (ASX) data from period 2000 to 2013 and the accumulation (total return) index for the S&P/ASX200 index. The study is limited to the largest 300 stocks by market capitalisation as measured on 31 December in each year in the study. Financial ratios are calculated either from the financial data (for example, return on equity), from both the financial and market data (for example, the book-to-price ratio), or from the market data (for example price momentum). The market data is also used to compute stock returns. We use annual stock returns (computed from December 31 each year) to compute the target variable. Daily stock returns are also used for portfolio performance details in some of the figures.

The so-called target variable is the variable we want our machine learning models to predict. Our target variable, which we refer to as *HighReturn*, is simply the classification of a stock as either a winner or a loser. It is a categorical variable which takes the value of 1 when a stock's total return for the calendar year is above the median and zero if it is below the median. Each year, we therefore have stocks classified into two categories of equal size. Outperforming stocks have *HighReturn=1*, while those which underperformed the median return have *HighReturn=0*. By defining the target variable

[1] In particular, the machine learning models in this study performed badly in 2009, ostensibly as a result of being estimated on data including the year 2008, a very bad year for equity markets.
[2] stevetulig/Machine-learning-financial-ratios-and-stock-picking (github.com)

this way, we can make use of classification algorithms, which as the name suggests attempt to predict the classification of a stock as either an outperformer or an underperformer.

The ratios in this study are based on both financial (annual report) and market (ASX) data and are well-known in the accounting and finance academic literatures for their ability to explain stock returns, either individually or in combination with other ratios. We examine all the ratios for their ability to predict future stock returns as part of an exploratory data analysis, and we then include the ratios in machine learning models to predict the next calendar year's stock return. All the ratios have outliers truncated[3] and are then standardized within each year to ensure they have zero mean and unit standard deviation. Table 1 lists all the ratios used in the study.

Table 1: List of ratios

Ratio	Meaning
B_P	Book-to-price (book value of equity divided by market value of equity)
E_P	Earnings-to-price, or earnings yield (net profit after tax divided by market value of equity; or equivalently earnings-per-share divided by share price)
ROE	Return on equity (net profit after tax divided by book value of equity)
D_A	Debt-to-assets (long-term debt divided by total assets)
CFO_A	Cash flow from operations divided by total assets
CFI_A	Cash flow from investing activities divided by total assets
CFF_A	Cash flow from financing activities divided by total assets
EBITDA_EV	Earnings before interest, depreciation and amortisation divided by enterprise value; enterprise value is defined as market value of equity plus book value of liabilities
MOM	Six-month price momentum (total return over the prior six months)
LIQ	Liquidity as defined in Standard and Poors' index methodology
MCR	Ranking by market capitalisation (with 1 denoting the largest company)

All the ratios for a stock are matched with the value of *HighReturn* (the classification as an outperformer or underperformer) in the following calendar year. The ratios based on ASX market data (*MOM*, *LIQ* and *MCR*) are computed on 31 December each year. The other ratios require the latest annual report with a balance date at least six months prior to 31 December (and therefore might be from as early as July in the previous year). The corresponding value of *HighReturn* is based on stock returns computed from the 31 December close to the 31 December close in the following year. All the ratios therefore represent information available on 31 December which might be used to predict whether the stock will outperform or underperform in the following calendar year.

We proceed by taking the perspective of an equity fund manager, who wants to estimate machine learning models and use them to make buy and sell decisions for long and short portfolios. Such a fund manager might initially conduct an exploratory data analysis to determine the suitable candidate variables to include in the machine learning models and eliminate others. We conduct this exploratory data analysis as if the date was 31 December 2005, using target (*HighReturn*) values from 2001 to 2005 and the corresponding ratios. We then fit all six machine learning models using the data available at 31 December 2005. The fitted models are then used with the ratios as at 31 December 2005 to predict the *HighReturn* variable (classification) for 2006.

All the models produce a probability that *HighReturn*=1. The fund manager then ranks all stocks by this probability, buys the top 33% for the long portfolio and sells the bottom 33% for the short

[3] We truncate outliers to within the standard definition of box plot whiskers (i.e., to either the first quartile minus 1.5 times the interquartile range or to the third quartile plus 1.5 times the interquartile range).

portfolio. The process of fitting machine learning models and trading off them is repeated on 31 December in each of the following years, with models fitted using a rolling five-year sample period. The Python Scikit-Learn library is used for all machine learning-related computations. Further details are discussed in the sections “Model-fitting and testing” and “Portfolio trading based on the models”.

Exploratory Data Analysis

In this section we perform an exploratory data analysis, using only the data observations corresponding to target (*HighReturn*) values from 2001 to 2005. Firstly, we examine the distributions and pairwise scatterplots of all variables, which are shown in Figure 1.

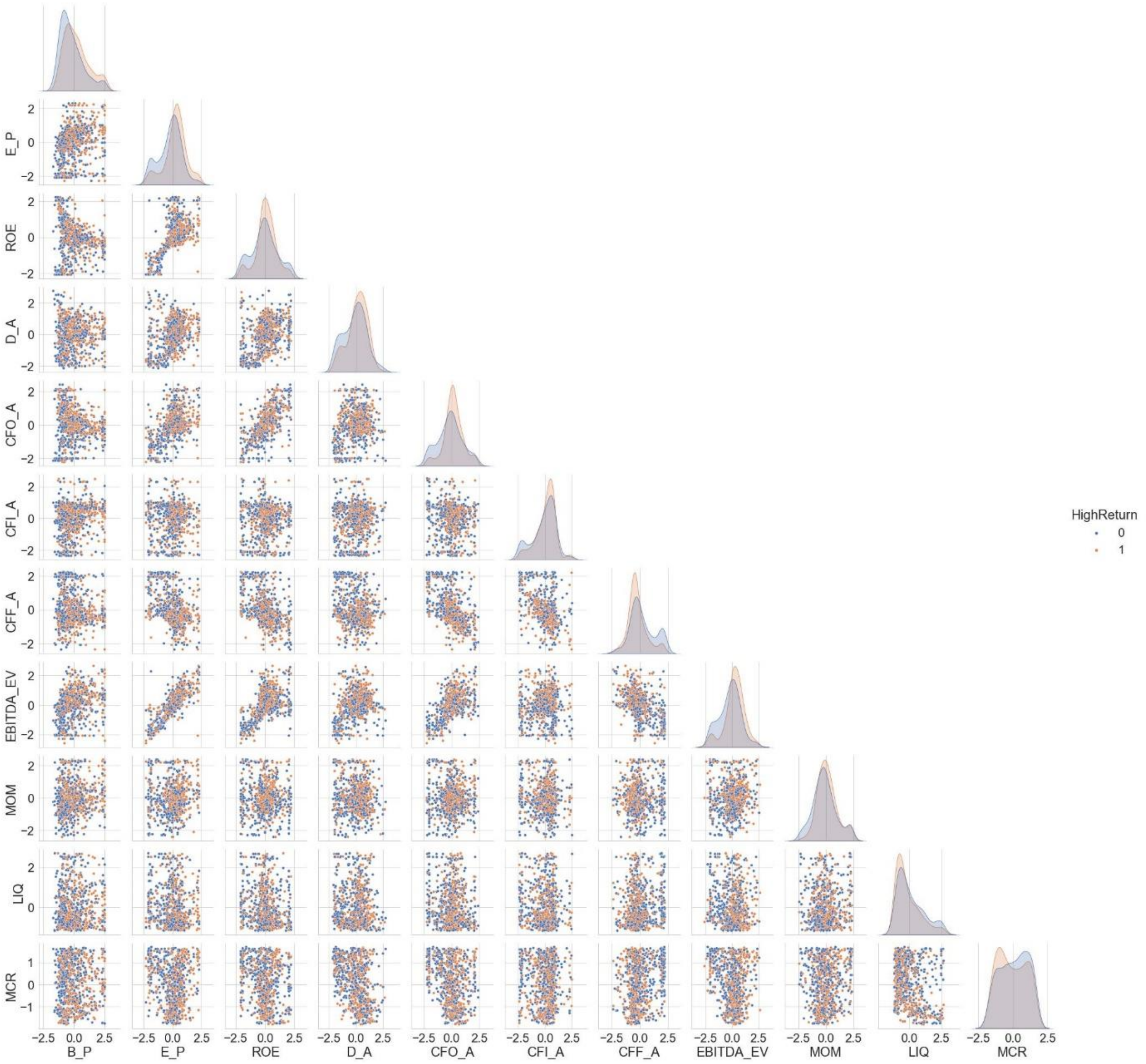


Figure 1: A ‘pairplot’ showing pairwise scatterplots and distributions of all variables

The diagonal elements in Figure 1 are the frequency distributions of the variables in the order shown on the x-axis, the first being for *B_P* and the last being for *MCR*. The off-diagonal elements are the pairwise scatterplots. Blue points and blue-shaded regions in the frequency distributions represent underperforming stocks (*HighReturn=0*), while orange points and orange-shaded regions in the frequency distributions represent outperforming stocks (*HighReturn=1*).

Some pairs of variables (e.g., *EBITDA_EV* and *E_P*) are more closely related than others (e.g., *D_A* and *MOM*). Some evidence of a relationship between each variable and future returns is evident from the frequency distributions. For example, notice the blue distributions of *B_P* and *E_P* tend to be to the left of the orange distributions. This confirms the widely known observation that, over the long term, low *B_P* and *E_P* stocks tend to underperform while high *B_P* and *E_P* (i.e., value) stocks tend to outperform.

Next, we conduct a more formal test of the ability of each variable to predict future returns. For this we divide stocks into the two groups based on *HighReturn* (i.e., outperforming stocks versus underperforming stocks) and use the non-parametric Kruskal-Wallis test to test whether the median value of each variable is the same for both groups. The p-values from this test are shown in Table 2.

Table 2: Kruskal-Wallis p-values of the test that the median of each ratio is the same for outperforming and underperforming stocks.

Ratio	Kruskal-Wallis p-value
B_P	0.000341
E_P	0.000000
ROE	0.000524
D_A	0.000270
CFO_A	0.000001
CFF_A	0.000000
CFI_A	0.148208
EBITDA_EV	0.000000
MOM	0.001190
LIQ	0.194272
MCR	0.076099

For all variables except *CFI_A*, *LIQ* and *MCR* the p-value is significant at the 5% level, and *MCR* is significant at the 10% level. Put differently, the median values of all the other variables are significantly different across the two groups (underperform versus outperform). We can therefore argue that all variables except *CFI_A* and *LIQ* have some ability to predict future returns.

Next, we look more closely into the relationships between the variables, to reduce the number of variables to include in our machine learning models. For this we examine the Spearman rank-correlation coefficients. Figure 2 shows a heatmap of the corresponding correlation matrix.

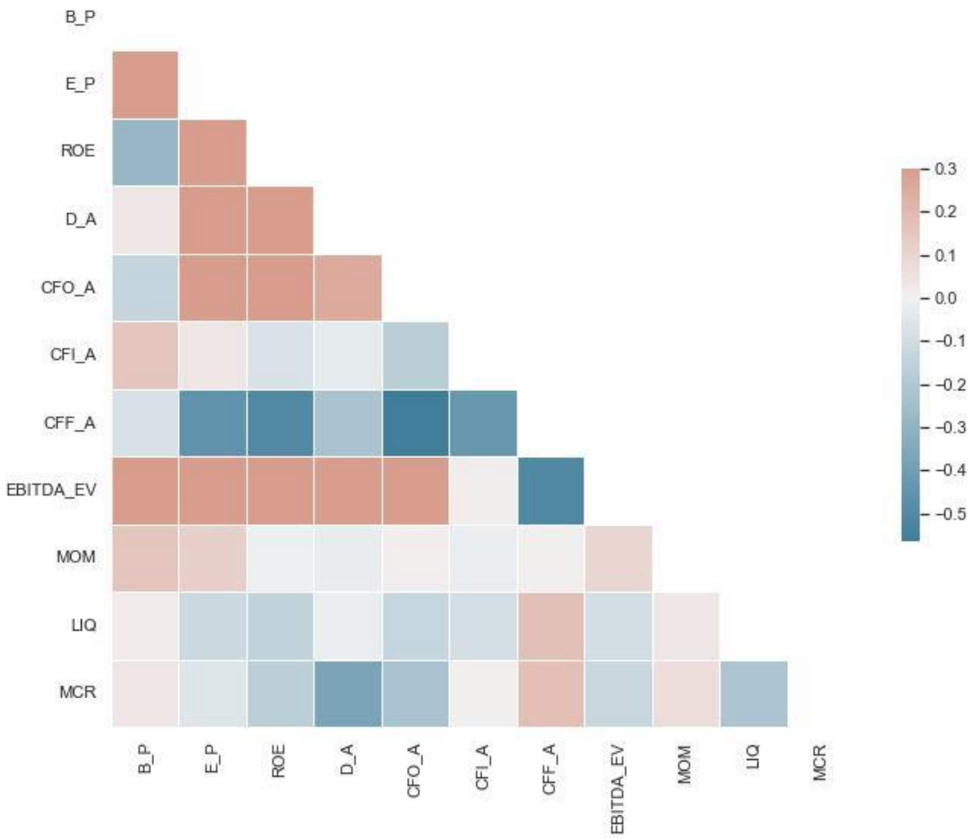


Figure 2: Rank-correlation matrix heatmap of all variables

We can see from Figure 2 that *CFF_A* (cashflow from financing activities) is the feature that is most (negatively) related to the others. This feature might also be a bit noisy because it doesn't separate flows to equity and debt, especially dividends versus interest payments in the case of outflows (in an ideal world we might include equity issuance, dividend payments, debt issuance, and interest payments as separate features). We therefore remove this feature, along with *CFI_A* (cashflow from investing activities) and *LIQ* (liquidity).

Model-fitting and testing

We use a series of rolling five-year sample periods to estimate and test models based on the following six machine learning algorithms:

- Penalised logistic regression
- K-nearest neighbours
- Support vector machines (with radial basis functions)
- AdaBoost
- Gradient boosted trees
- Random forests

The modelling process is intended to simulate the research activities of a fund manager, who estimates a model each year based on the previous five years of data, and then implements (i.e., trades on) the output of each model. Each model tries to predict whether a stock will outperform or underperform the median return over the following year (the stock's classification by *HighReturn*). On 31 December each year (starting with 31 December 2005), a model based on each algorithm is fitted to the previous five years of data. For each stock, this data consists of ratios macthed with the 'target' (*HighReturn*) for the subsequent calendar year (for example a *B_P* ratio and other ratios from 2004 paired with a *HighReturn* value for the 2005 calendar year). Once each model is fitted it is used with the latest set of ratios to predict returns in the following year. For example, on 31 December 2005 models were fitted using ratios from 2000 to 2004 and the corresponding annual returns from 2001 to 2005. On the same date, the models were used with ratios from 2005 to predict returns for 2006[4].

Each model requires tuning of hyperparameters[5], which is accomplished in the usual way through (V-fold) cross-validation. For example, many of the algorithms are based on decision trees, one hyperparameter of which is the 'maximum depth' of each tree, in other words the maximum length of a path through the decision tree. Each five-year data sample is randomly split into training and test samples using a 70%-30% split. The training set is used for cross-validation, and the test set is used to test each model on data not used for cross-validation. We record the accuracy of each model using the accuracy score, which is the percentage of stocks correctly predicted as either an outperformer or underperformer. We record the accuracy score for each model and year, for both the training set and the test set.

The training and testing of models to this point does not yet represent a true test of how the models perform in a real-life application of their intended purpose. This purpose is two-fold: firstly, to classify each stock as either a (future) outperformer or underperformer, and secondly to make money by trading off this classification[6]. To test the real-life classification ability of each model, we use the model in a practical implementation to predict the outperformers and underperformers in the year after each model is fitted. For example, on 31 December 2005 our fund manager fits and tests models using historical data, as discussed above. He then implements each model to predict returns for 2006 based on the ratios for 2005. In a temporal sense, this step is therefore the true out-of-sample application. The accuracy of each model in this step is recorded as the 'out of sample (implementation) accuracy'.

Figure 3 shows all three accuracy measures for each model, for each five-year period. As we can see by the blue bars, classification accuracy was always quite high (generally 60% or greater) when the models were initially fitted. As expected, the orange bars tell us that accuracy falls when each model is tested on data points not used to fit the models (but still from the same five-year period). This is quite noticeable for the K-nearest-neighbours classifier which always had a 100% accuracy during fitting[7], but the test accuracy fell to around 60% or less. Generally, testing accuracy for all models are between 50% and 60%. Whilst this level of testing accuracy is unacceptable in most application, it is not unexpected in equity market applications due to the inherent noisiness of the data. It should be kept in mind that even a slight statistical edge can lead to successful portfolio trading, a consequence of Grinold and Kahn's well known "Fundamental Law of Active Management".

[4] Refer to the Jupyter Notebook for a sample of the data along with explanations: Machine-learning-financial-ratios-and-stock-picking/Machine Learning, Financial Ratio, Stock Picking Notebook.ipynb at main · stevetulig/Machine-learning-financial-ratios-and-stock-picking (github.com)

[5] Refer to the Jupyter Notebook for details of the hyperparameters tuned for each model

[6] The ability to make money off each model is examined in the section "Portfolio trading based on the models".

[7] 100% training accuracy results when the number of neighbours hyperparameter equals 1. However, this was never the case as cross-validation always yielded values between 28 and 90.

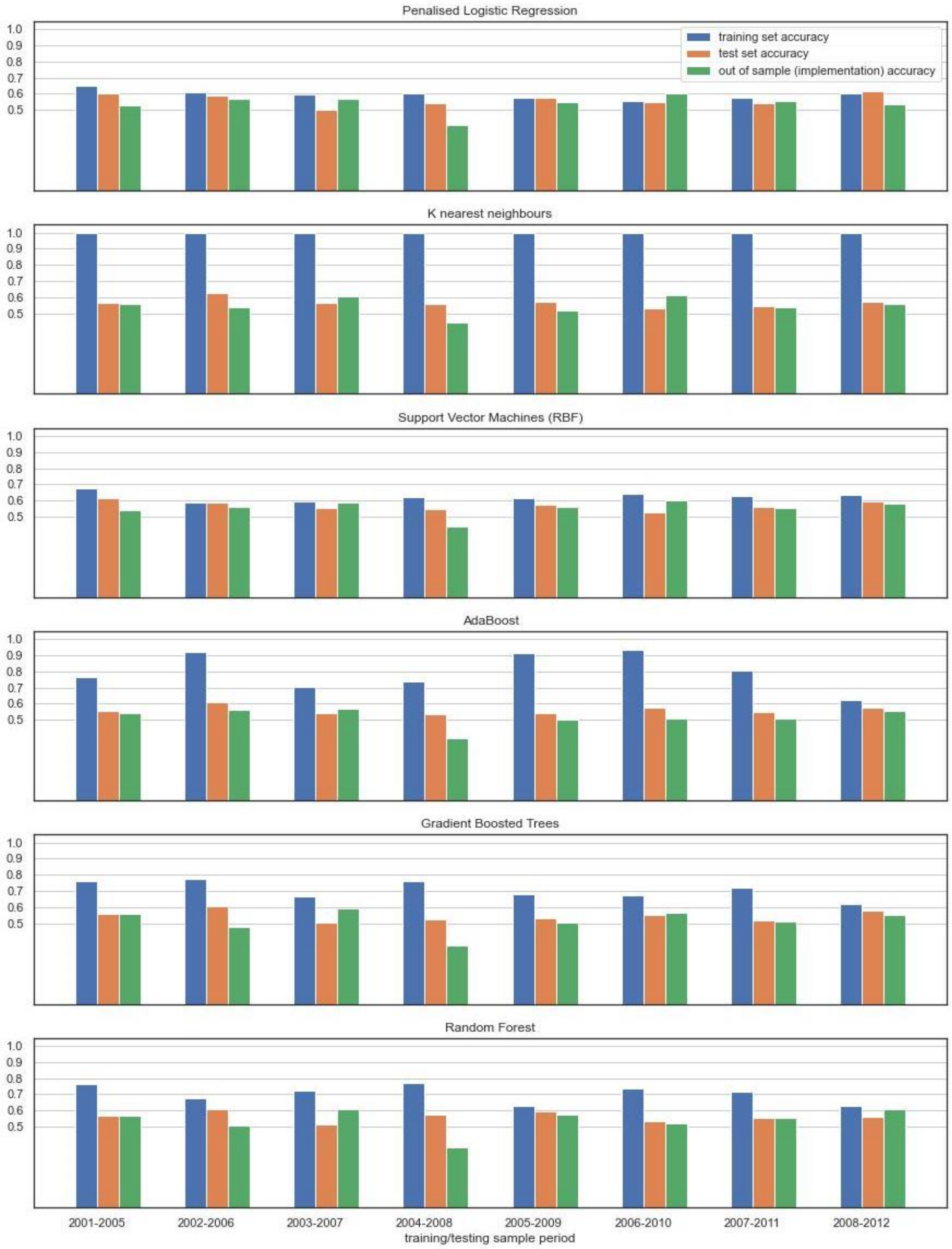


Figure 3: Model training, testing, and out of sample accuracy

The green bars in Figure 3 show the classification accuracy of each model when used 'live', in other words by implementing the model for the purpose of trading on the results. The heights of the green bars are also generally between 50% and 60%, with one glaring exception. *All* the models performed badly when used to classify stocks for the year 2009. This is illustrated by the groups of bars for the 2004-2008 training/testing sample period, indicating the models fitted on 31 December 2008 using (*HighReturn*) data from 2004 to 2008. A fund manager using any of these models on 31 December

2008 would have been wrong (much) more than 50% of time in their attempts to predict the winners and losers for 2009.

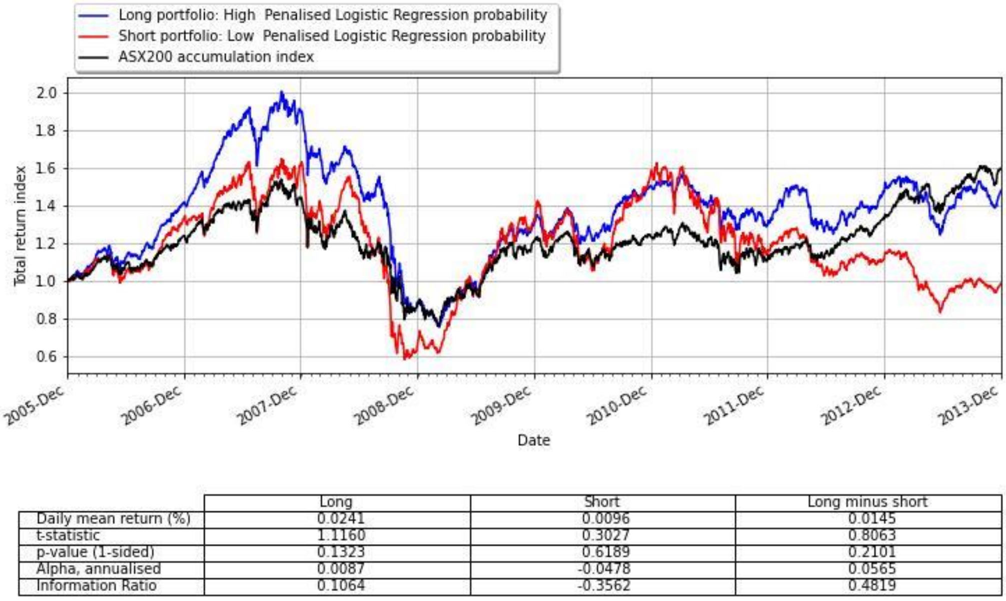
It is not difficult to understand the reason for the poor model performance for 2009. 2008 was the worst year for equity returns during the GFC period, and the 2004-2008 sample period was the first time the problematic year of 2008 enters the sample period used to fit each model.

Portfolio trading based on the models

We now investigate the use of the machine learning models in portfolio trading strategies. Specifically, we wish to determine whether we can make money by buying the stocks our models predict will outperform and selling the stocks our models predict will underperform. Each model can estimate the probability that a stock will outperform in the following year. We form portfolios by ranking all stocks by this probability of outperformance. We include the top 33% of stocks ranked by the probability of outperformance in the long portfolio, and the bottom 33% of stocks in the short portfolio. Portfolios are rebalanced on 31 December every year, using the latest fitted model. The results are illustrated in Figures 4 to 9.

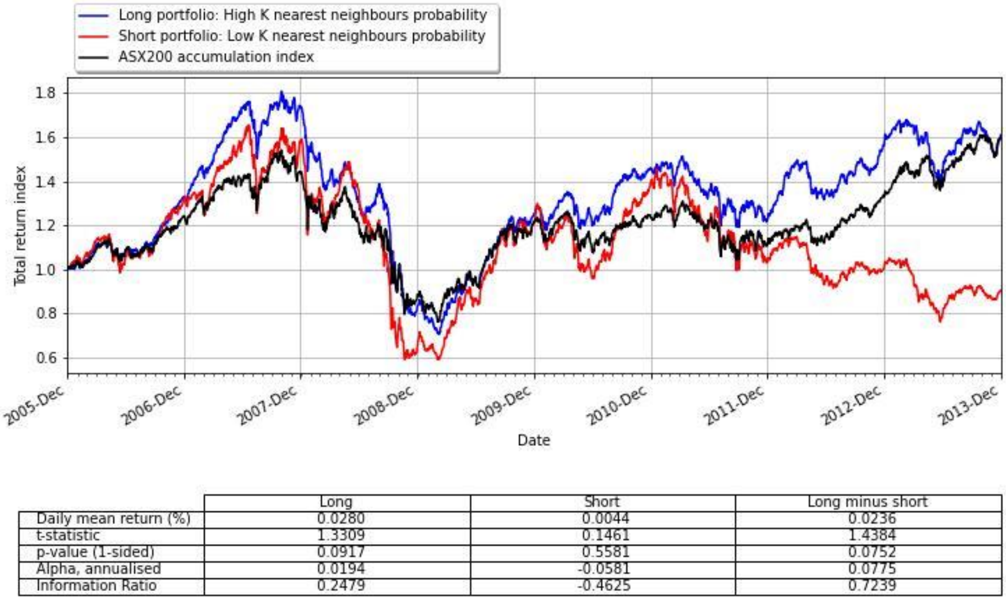
Each figure shows the cumulative total returns of the long (blue) and short (red) portfolio along with those of the ASX200 Accumulation Index (black). We can think of each curve as the value of each of the assets over time if $1 is invested on 31 December 2005 and all dividends are reinvested. Each figure also includes a table showing performance measures for the long portfolio, short portfolio and a corresponding long-minus-short portfolio. In this analysis we have not considered transaction costs, so true performance would have been somewhat worse than that suggested by the results.

We can summarise the key takeaways from Figures 4 to 9 as follows. Along with the rest of the equity market, all portfolios experienced large negative returns over 2008. This effect occurred globally and cannot be attributable to poor model performance. What is of more interest in the context of this study is the *relative* performance of the long and short portfolios, i.e., the vertical difference between the blue and red lines. Over the whole time period, all the models generated a *somewhat* profitable return differential between the long and short portfolios. The blue curve always ended up above the red curve (representing the short portfolio). The return differential is generally observable in the first two years, disappeared by the end of 2009 for four of the six models, and reappeared over the latter half of the time period. Long-minus-short trading would have made abnormal profits in the first two years, lost the gains over the subsequent two years, and again made money over the final four years.



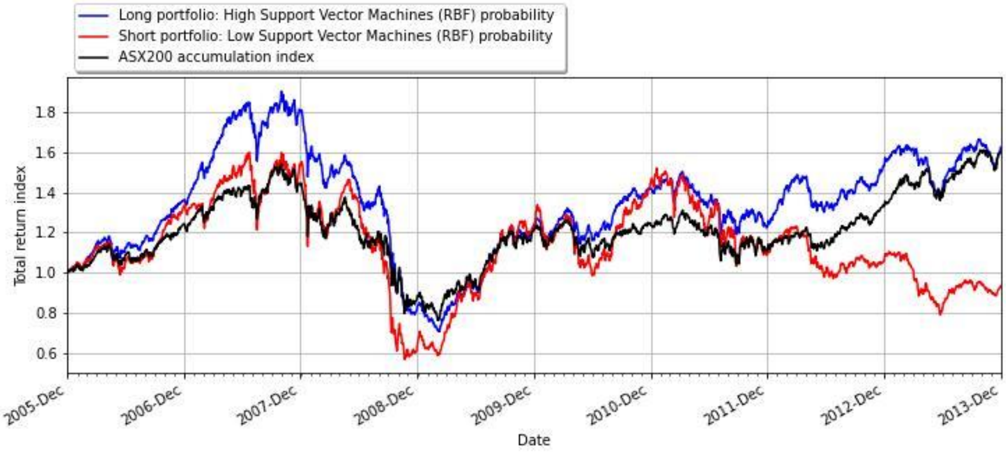
	Long	Short	Long minus short
Daily mean return (%)	0.0241	0.0096	0.0145
t-statistic	1.1160	0.3027	0.8063
p-value (1-sided)	0.1323	0.6189	0.2101
Alpha, annualised	0.0087	-0.0478	0.0565
Information Ratio	0.1064	-0.3562	0.4819

Figure 4: Portfolio trading performance based on penalised logistic regression



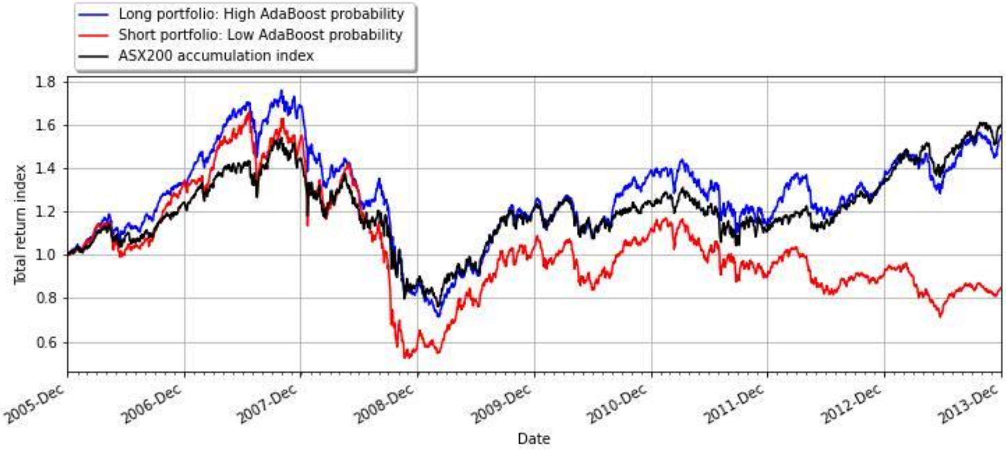
	Long	Short	Long minus short
Daily mean return (%)	0.0280	0.0044	0.0236
t-statistic	1.3309	0.1461	1.4384
p-value (1-sided)	0.0917	0.5581	0.0752
Alpha, annualised	0.0194	-0.0581	0.0775
Information Ratio	0.2479	-0.4625	0.7239

Figure 5: Portfolio trading performance based on k-nearest neighbours



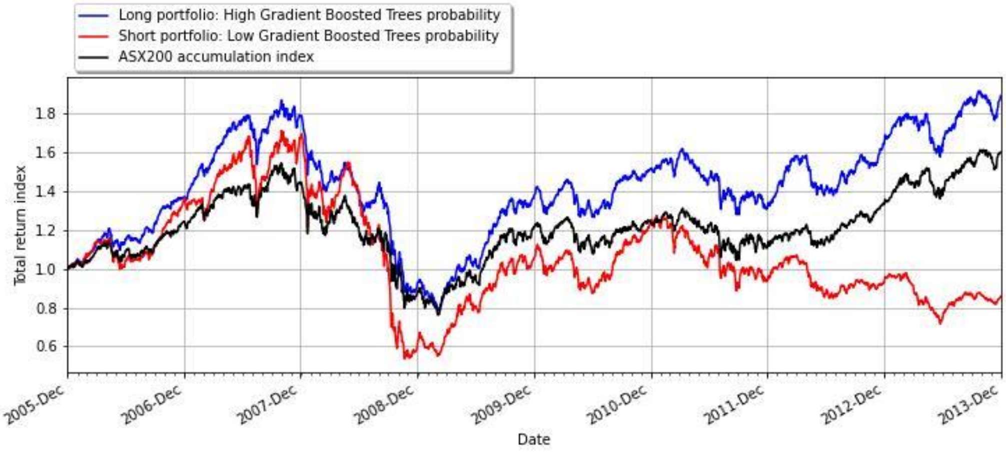
	Long	Short	Long minus short
Daily mean return (%)	0.0286	0.0064	0.0222
t-statistic	1.3362	0.2066	1.3090
p-value (1-sided)	0.0908	0.5818	0.0953
Alpha, annualised	0.0194	-0.0544	0.0738
Information Ratio	0.2519	-0.4154	0.6631

Figure 6: Portfolio trading performance based on support vector machines (with radial basis functions)



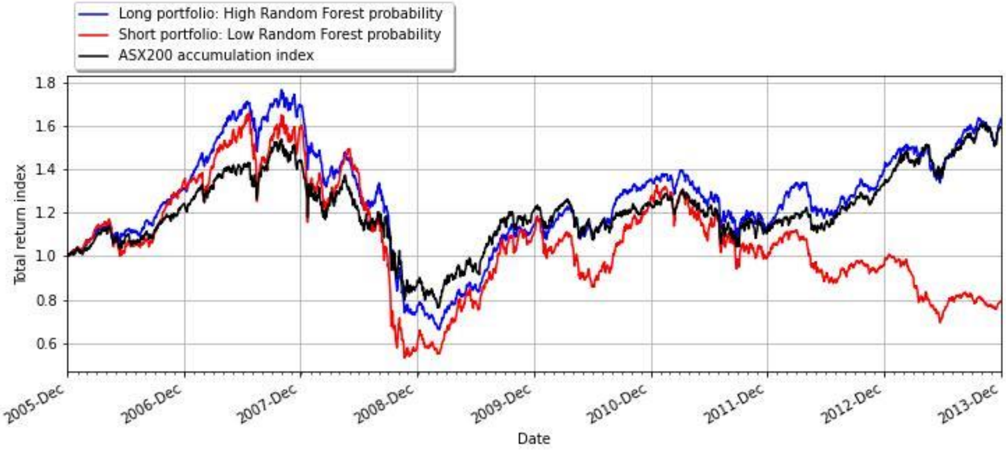
	Long	Short	Long minus short
Daily mean return (%)	0.0266	0.0005	0.0261
t-statistic	1.2061	0.0176	1.7030
p-value (1-sided)	0.1140	0.5070	0.0444
Alpha, annualised	0.0131	-0.0658	0.0789
Information Ratio	0.1627	-0.5452	0.7637

Figure 7: Portfolio trading performance based on AdaBoost



	Long	Short	Long minus short
Daily mean return (%)	0.0358	0.0020	0.0338
t-statistic	1.7161	0.0643	1.9872
p-value (1-sided)	0.0431	0.5256	0.0235
Alpha, annualised	0.0386	-0.0649	0.1035
Information Ratio	0.5181	-0.5068	0.9304

Figure 8: Portfolio trading performance based on gradient boosted trees



	Long	Short	Long minus short
Daily mean return (%)	0.0286	-0.0017	0.0303
t-statistic	1.3810	-0.0539	1.7660
p-value (1-sided)	0.0837	0.4785	0.0388
Alpha, annualised	0.0215	-0.0745	0.0960
Information Ratio	0.2819	-0.5697	0.8669

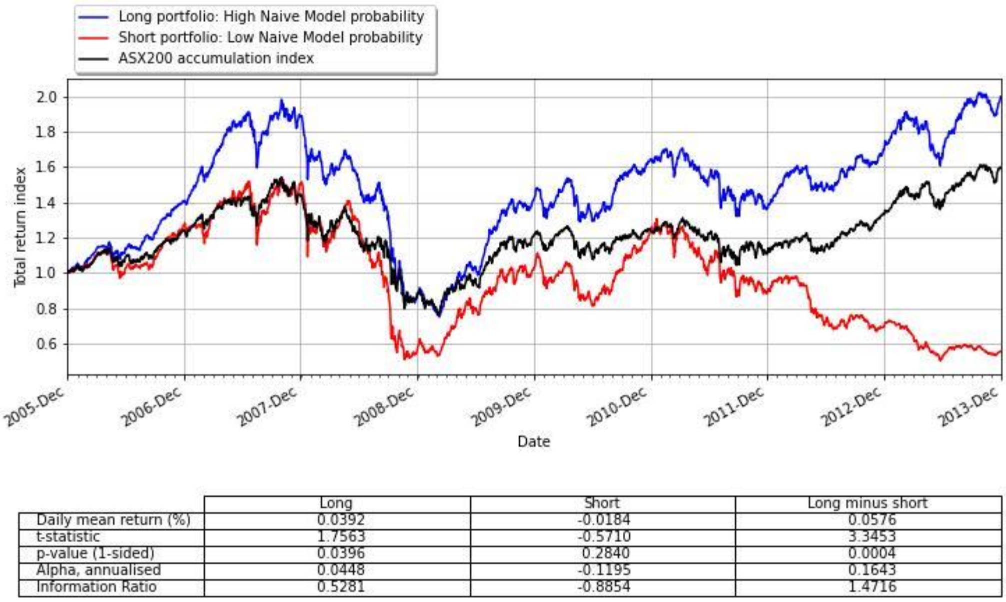
Figure 9: Portfolio trading performance based on random forests

We now turn our attention to the performance measures in the bottom section of each figure. All the models generated a positive alpha for the long portfolio and a negative alpha for the short portfolio, although some are negligible. The long portfolio alphas range between 0.9% for the penalised logistic regression model to 3.9% for the gradient boosted trees model. The short portfolio alphas range between -4.8% for the penalised logistic regression model to -7.45% for the random forests model. Long-minus-short returns are statistically significant only for the tree-based models (AdaBoost, gradient boosted trees and random forests). Information ratios range from 0.48 for the penalised logistic regression model to 0.93 for the gradient boosted trees model.

Taken as a whole, the results are not impressive, especially considering that all the ratios used as model features have explanatory power over returns (and remember, we have not included transactions costs). One possibility is that the process of building machine learning models has introduced noise into the process, and that perhaps we would have been better off with a less sophisticated trading model. We investigate this possibility next.

Performance Comparison with a Naïve Model

To determine whether the effort required to build machine learning models for equity trading is justified, we now compare the trading performance of the above machine learning models with that of a much simpler model. We refer to this model as a 'Naïve Model' because it requires much less sophistication to set up. The Naïve Model is based on a simple aggregation of some of the ratios which were already well-recognised by the early 2000s for their monotonic relationship with returns, namely *B_P*, *E_P*, *ROE*, *CFO_A*, *EBITDA_EV* and *MOM*. We then simply sum the standardised values of the ratios together to derive a new ratio we refer to as *NAÏVE*, in a similar fashion to the well-known Piotroski F-score. We then include the top 33% of stocks ranked by *NAÏVE* in the long portfolio and the bottom 33% of stocks in the short portfolio. The portfolio trading results for the Naïve model are illustrated in Figure 10.



	Long	Short	Long minus short
Daily mean return (%)	0.0392	-0.0184	0.0576
t-statistic	1.7563	-0.5710	3.3453
p-value (1-sided)	0.0396	0.2840	0.0004
Alpha, annualised	0.0448	-0.1195	0.1643
Information Ratio	0.5281	-0.8854	1.4716

Figure 10: Portfolio trading performance based on a naive model

It is quite apparent from Figure 10 that the Naïve Model resulted in better performance than any of the Machine Learning models. The long and short portfolios performed as expected both prior to and after the GFC years. The alphas of both the long and short portfolios, 4.48% and -11.95% respectively, exceed the corresponding measures for every machine learning portfolio in magnitude. The best machine learning alphas were 3.86% for the long portfolio and -7.45% for the short portfolio. The Naïve model was also the best performing model in terms of information ratio for the 'Long minus short' portfolio (1.47); the best performer amongst the machine learning models in this regard was the Gradient Boosted Trees algorithm (0.93). Unlike the machine learning models, the Naïve Model long portfolio easily beat the benchmark period over the time period considered.

Conclusion

Machine learning models based on the ratios considered did not perform as expected. Classification accuracy varied by year and was generally a little too low to be useful for portfolio trading. All the models performed quite poorly for the year 2009 when they were impacted by confounding effects of a badly performing equity market in the previous year. The observation that a Naïve Model outperformed the more sophisticated machine learning models suggests that practitioners need to be cautious in their implementation of and confidence in such models.