# Factor Investing with a Linear Model and with Machine Learning

Yiyi Xu, Ben Ding, Xiaoqian Sun, Harshvardhan Solanki

April 2021

## 1 Abstract

Our project aims at comparing the performances of two portfolios constructed by factor investing with a linear model and machine learning models. All stocks are selected from the S&P 500. The period of 02/02/2015-12/31/2018 is the training set and 01/02/2019-12/31/2019 is the test set. In the linear model, 11 factor portfolios are first constructed by long $1 of each stock that has its exposure to a factor in top 50% and short $1 of each stock that has its exposure to a factor in bottom 50%. Targeting at maximizing the Sharpe Ratio for the training period, weights for factor portfolios are determined. The portfolio for the test period is a linear combination of 11 factor portfolios with their weights determined from the training period. Machine learning models predict the sign of excess return using the same 11 factors plus industry factors. The portfolio is then constructed by long $1 of each stock that is predicted to have positive excess return and short $1 of each stock that is predicted to have negative excess return. The linear model obtains a Sharpe Ratio of 0.75 for the test period. Among all machine learning models, Logistic Regression obtains an out-of-sample prediction accuracy of 50.63% and a Sharpe Ratio of 0.14. The out-of-sample prediction accuracy is improved to 50.93% by the Decision Tree Classifier and the corresponding Sharpe Ratio is improved to 1.13. This report consists of 4 main sections: Section 2 describes data sources and data preprocessing; section 3 introduces the attributes; section 4 introduces two models and demonstrates portfolio performances; section 5 explains the results from two models and includes possible improvements.

## 2 About the Dataset

Stock samples from S&P 500

1. **Time frame:** 02/02/2015 – 12/31/2020
2. **Sample size:** We select 408 stocks from the 505 stocks of S&P 500 and drop the other 67 stocks because their data are incomplete. For example, TraneTechnologies(TT) only released PE ratios of 2020 and 2021, which does not match our time frame.
3. **Factors:**
   (a) **PE ratio, PB ratio, Return on Equity (RoE):**
   We collect quarterly updated PE ratios, PB ratios and Return on Equity (RoE) of the 408 stocks from Macrotrends.

(b) **Net margin and total asset turnover:**
We collect quarterly updated total revenue, total assets, and net income of the 408 stocks from Macrotrends to calculate these two factors.

(c) **DCF premium:**
We collect the annually updated free cash flow, current liabilities, cash on hand and the number of shares outstanding of the 408 stocks from Macrotrends to calculate the DCF premium factor.

(d) **Growth factor:**
We collect the quarterly updated total revenue, operating expense and net income of the 408 stocks from Macrotrends to calculate the growth factor.

(e) **PE premium:**
We collect updated operating income, net income of the 408 stocks from Macrotrends to calculate the P/E premium factor.

(f) **Stock prices and company sector:**
We collect daily updated historical stock prices and company sectors of the 408 stocks from Yahoo Finance.

4. **Method:**

   (a) Use the *requests* package in python to get the web pages like `https://www.macrotrends.net/stocks/charts/MMM/3m/roe` (replace the mmm, 3m, roe with stock name,ticker and factors we need)

   (b) Use regular expression (*re* package) to parse the response and find all the data we want.

   (c) Clean the data and construct dataframes.

5. **Problems and solutions:**
   When collecting financial data such as total revenue and P/E ratios, we first tried using `https://elite.finviz.com/`, but the problem was that finviz provides data for at most eight periods. That means we can only get two years of quarterly data,which is not enough. Then we turned to Bloomberg, but the P/E ratios given by Bloomberg showed spurious high Sharpe Ratio. We found out that Bloomberg gave us the forward P/E ratio, which incorporates significant information about future stock prices that we won't be able to see in reality. So, the data given by Bloomberg are not suitable for our project. Finally, we found that Macrotrends could provide the data we need.

# 3 Factors

In this section, we discuss the meaning of our factors and how we derive the raw signals.

## 3.1 Fundamental Attributes

### 3.1.1 The Discounted Cash Flow Model

The Discounted Cash Flow Model (DCF model) aims to find a company's fair equity value, or Enterprise Value (EV) by making predictions about its future free cash flows (FCF) and discounting them at an appropriate rate. We make several simplifying assumptions when implementing the model:

1. All companies' FCF will grow linearly;
2. The appropriate discounting rate is the companies' weighted average cost of capital (WACC), which is the same across all S&P 500 companies;
3. All companies will last forever and follow a terminal growth rate after 10 years.

Using these assumptions and the definition of the model, the following procedures are used to calculate raw signals for every company for each year in the period 2015-2020:

- Step 1: Calculate past five year's "free cash flow" (FCF) by multiplying each year's FCF / price ratio (extracted from macrotrends.com) by each year's year-end stock price.
- Step 2: Predict FCF by using rolling regression, assuming that the FCF will grow linearly for the next 10 years. For example, we use the FCF from 2010 to 2014 to predict the FCF of the period 2015-2025; and then use the FCF from 2011 to 2015 to predict the FCF of the period 2016-2026 and so on.
- Step 3: Discount 10 year's cash flow by the company's weighted average cost of capital(WACC). We use 6.6%, the average WACC for S&P 500 companies.
- Step 4: Add the above figure by: the projected FCF in 10 years / (WACC-x%). If the company is in technology, then it's 3.5%. If the company is in consumers, then it's 2.5%. Otherwise it's 3%.
- Step 5: Calculate Enterprise Value (EV) by: the above figure - current liabilities + cash + (marketable securities or short term investments) = Enterprise value (EV).
- Step 6: Raw signal of the DCF model is given by (EV/ # of shares outstanding) - (the company's current share price). The raw signal could also be interpreted as a company's DCF premium (how high the DCF fair value is above the company's current stock price).

The DCF model is largely driven by a company's weighted average cost of capital (WACC) as well as its terminal growth rate after 10 years. The higher a company's current growth and growth prospect are, the higher DCF fair value it will have. Thus, the DCF model raw signal favors companies with high growth, so the DCF premium should be considered as a growth factor.
This factor is an annual factor, being updated yearly.

### 3.1.2 The P/E model

The P/E model aims to find a company's EV by first projecting its net income in the following fiscal year, then multiplying it by the trailing P/E ratio. Note that because there exists timing differences between a company's financial statements releases and the trailing P/E ratio, there might be look-ahead bias associated with this process. Hence, we will make a simplification to the procedures, as well be shown in Step 3 below.
The procedures of finding raw signal of the P/E model for each company in each year during the period of 2015-2020 is given by:

- Step 1: Calculate growth rate in the past five years of net income.
- Step 2: Assuming the growth rates will continue for next fiscal year, calculate forecasted net income for the next fiscal year.
- Step 3: A growth rate can be calculated by dividing this forecast by the current net income. Then, the raw signal is given by multiplying this growth rate by the current stock price per share. Note that this raw signal represents exactly (EV predicted by P/E model) - (current stock price), which can be defined as the P/E model premium.

The P/E premium is driven by two factors: a company's current growth and price per share. Both of these factors favor companies with high growth, so the P/E premium should also be considered as a growth factor. Also, it is interesting to separate these two factors and see their individual performances, as will be discussed in 1.8 and 1.9 in this section.
This factor is an annual factor, being updated yearly.
It is worth mentioning that though the DCF and P/E premium model should both be considered growth factors, they differ in some main factors that drive their raw signals. Therefore, if the theoretical foundations of our DCF and P/E premium models are correct, the raw signals (as well as the normalized signals) should have a weak, positive correlation. Indeed our data shows a correlation of 0.2 as expected in Figure1.

### 3.1.3 Return on Equity (RoE)

The Return on Equity (RoE) ratio is an overall performance measure of a company; it measures the amount of net income earned on every $1 of shareholder's equity. The raw signal is given by: (net income) / (total shareholder'sequity).
This factor is updated quarterly.

### 3.1.4 Net Margin

The Net Margin ratio measures the profitability of a company. The raw signal is given by:(net income) / (total revenue).
This factor is updated quarterly.

### 3.1.5 Total Asset Turnover

The Total Asset Turnover ratio measures the efficiency of a company in using its assets. The raw signal is given by: (total revenue) / (total assets).
This factor is updated quarterly.

### 3.1.6 Trailing P/E Ratio

The trailing P/E ratio measures the expensiveness of a company's current stock price. Its raw signal is given by: (price per share) / (EPS over the period of last 12 months).
Note that this ratio is different with the forward P/E ratio, whose raw signal is given by (price per share) / (forecasted EPS over the period of next 12 months). Therefore, historical data of forward P/E ratio is deceiving, as it incorporates significant information about future stock prices (at the time of measurement) which we should not expect to know. Therefore, although using forward P/E ratio as one of our factors yielded an extremely high Sharpe ratio, our trading model would be useless in real world live trading. Hence, the trailing P/E ratio is used instead.
The trailing P/E ratio is a growth factor as more expensive (thus high growth) companies will have higher raw signals.
This factor is updated quarterly.

### 3.1.7 Price/Book Ratio

The Price/Book Ratio (or P/B ratio) also measures the expensiveness of a company's current stock price. Its raw signal is given by: (price per share) / (book value per share).
The P/B ratio is also a growth factor as more expensive (thus high growth) companies will have higher raw signals.
This factor is updated quarterly.

### 3.1.8 Growth Factor

The growth factor's raw signal is the growth rate between a company's current net income and the forecast of its net income in the next fiscal year, as in Step 1 in the procedures shown in section 1.2. To be consistent with the P/E premium factor, this factor is an annual factor, to be updated yearly at the same time as the P/E premium factor.

### 3.1.9 Price Factor

The price factor is calculated by rolling regression, we use the past five years' prices to predict the next year's price, the last year-end date before our calculation of the P/E premium factor to avoid any look-ahead bias. To be consistent with the P/E premium factor, this factor is an annual factor, to be updated yearly at the same time as the P/E premium factor.

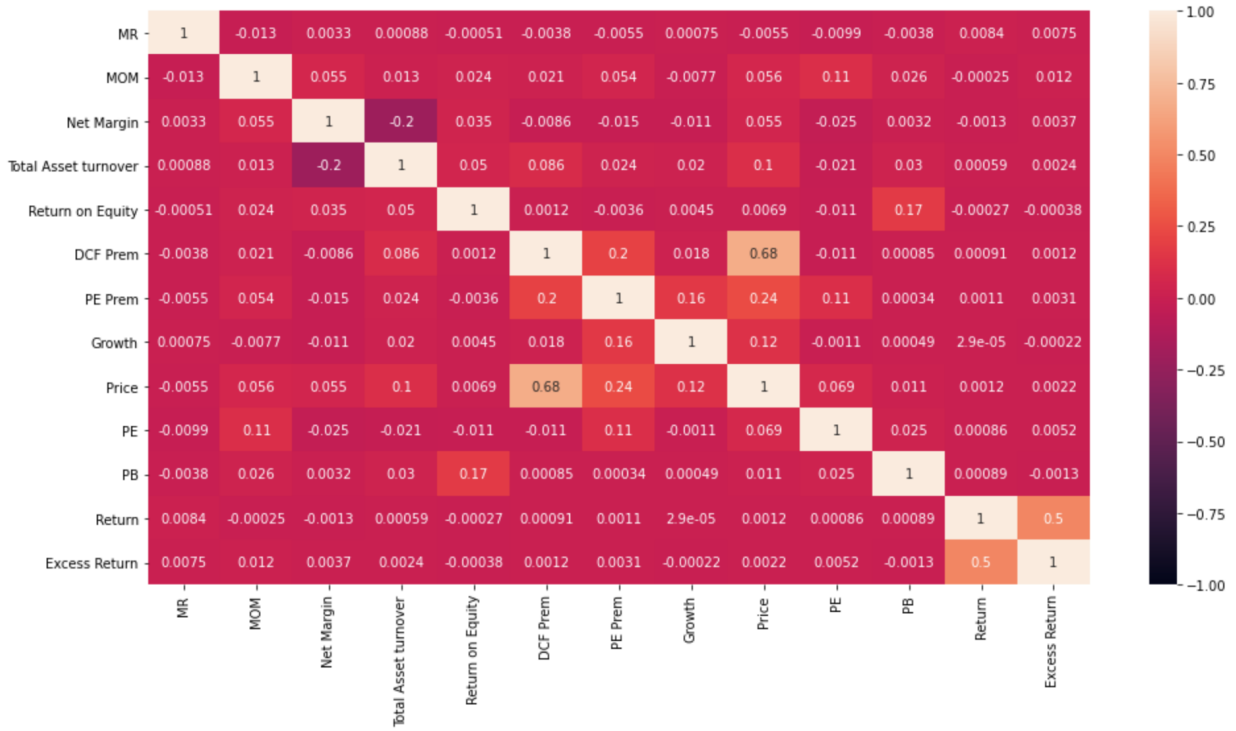



Figure 1: Correlation between attributes and return

## 3.2 Market Attributes

### 3.2.1 Mean reversion

Mean reversion strategy assumes short-term mean reversibility of stock prices. The raw signal $MR_{n,T}$ for stock $n$ on day $T$ is the return over the prior 5 trading days times -1.

$$MR_{n,T} = -\sum_{t=T-5}^{T-1} r_{n,t} \tag{1}$$

The raw signal is updated daily.

### 3.2.2 Momentum

Momentum strategy assumes that stocks which performed well will continue to do so, and vice versa. The raw signal $MOM_{n,T}$ for stock $n$ on day $T$ is return over the past year, but excluding the most recent month.

$$MOM_{n,T} = \sum_{t=T-252}^{T-22} r_{n,t} \tag{2}$$

The raw signal is updated daily.

### 3.3 Industry Attributes

All of our stocks are given industry attributes depending on which industry they are considered to be in. The industries are: Industrials, Health Care, Information Technology, Communication Services, Consumer Discretionary, Utilities, Financials, Materials, Real Estate, Consumer Staples and Energy. If the company is within the industry, it will be given a raw signal of 1; otherwise a raw signal of 0 is given. This factor is a permanent factor.

## 4 Models

This section introduces two methods for portfolio construction and analyzes the performances of resulting portfolios. For all models, the period of 02/02/2015-12/31/2018 is the training set and 01/02/2019-12/31/2020 is the test set. The linear model constructs the portfolio by maximizing the Sharpe Ratio of the training set, which obtains an in-sample Sharpe Ratio of 0.84 and an out-of-sample Sharpe Ratio of 0.75. On the other hand, machine learning models construct portfolios by taking long position in stocks with positive expected excess return and vice versa. Among all machine learning models, Decision Tree performs best, which obtains an in-sample Sharpe Ratio of 2.45 and an out-of-sample Sharpe Ratio of 1.13.

### 4.1 Linear Model

#### 4.1.1 Model Construction

- **Step 1:** As introduced in section 3, there are 11 fundamental attributes and market attributes. We construct a factor portfolio for each of the attributes by the following rule: rank all stocks' exposures to a factor, then long $1 of a stock if its exposure is among the top 50%, otherwise short $1.

**Example 1** *If we only have 4 stocks and 2 factors, on a specific day, assume their exposures to raw signals are as showed in the first two rows in the table. Then, our method will convert the exposures to factor portfolio positions as showed in the bottom two rows in the table. The bold numbers are in top 50% and thus assigned positions of +1, and the rest are assigned positions of -1. Because all factors are scaled to have zero mean, the portfolio is guaranteed to be market neutral.*

| | *Stock A* | *Stock B* | *Stock C* | *Stock D* |
|---|---|---|---|---|
| *Factor 1 (Raw Signal)* | *0.1* | *0.2* | ***0.3*** | ***0.4*** |
| *Factor 2 (Raw Signal)* | ***0.7*** | *-0.3* | ***0.5*** | *-0.1* |
| *Factor Portfolio 1* | *-$1* | *-$1* | *$1* | *$1* |
| *Factor Portfolio 2* | *$1* | *-$1* | *$1* | *-$1* |

We apply the same operation to all 408 stocks, 11 factors over the period of 02/02/2015-12/31/2020, which gives us 11 daily updating factor portfolios.

| Factor | Mean | Std | Sharpe Ratio |
|---|---|---|---|
| MR | 13.018 | 21.060 | 0.618 |
| MOM | -1.489 | 23.340 | -0.064 |
| Net Margin | 0.533 | 10.718 | 0.050 |
| Total Asset Turnover | 3.621 | 11.112 | 0.326 |
| Return on Equity | 4.674 | 13.25 | 0.353 |
| DCF Premium | 6.059 | 8.078 | 0.750 |
| PE Premium | 1.137 | 10.049 | 0.113 |
| Growth | 0.606 | 7.076 | 0.086 |
| Price | 4.689 | 10.323 | 0.454 |
| PB | 9.396 | 17.158 | 0.548 |
| PE | 2.493 | 16.398 | 0.152 |

Table 1: Mean, standard deviation and Sharpe Ratio for factor portfolios over 02/02/2015-12/31/2020

- **Step 2:** Calculate factor returns $b(t)$ by positions $p_n(t)$ and stock returns $r_n(t)$ for all k factors and T days:

$$b_k(t) = \sum_{n=1}^{408} p_{n,k}(t) r_n(t) \tag{3}$$

It follows that the expected annual return for factor portfolio k $R_k$ and its standard deviation $\sigma_k$ can be calculated by:

$$R_k = \frac{252}{T} \sum_{t=1}^{T} b_k(t) \tag{4}$$

$$\sigma_k = \sqrt{\frac{252}{T} \sum_{t=1}^{T} (b_k(t) - R_k)^2} \tag{5}$$

Sharpe Ratio (SR):

$$SR_k = \frac{R_k}{\sigma_k} \tag{6}$$

The resulted Sharpe Ratios for the whole period of 02/02/2015-12/31/2020 are showed in Table 1. After scaling returns to the same volatility of 15%, Figure 2 visualizes the performances of 11 factor portfolios. The one with higher return also has a higher Sharpe Ratio.

- **Step 3:** Optimize the weights for factor portfolios so that the optimal training portfolio as a linear combination of 11 factor portfolios has the highest Sharpe Ratio. The mean return, standard deviation and Sharpe Ratio for the training period is shown in Table 2 The position $P_n(t)$ for stock $n$ at time $t$ is determined by weights of the 11 portfolio

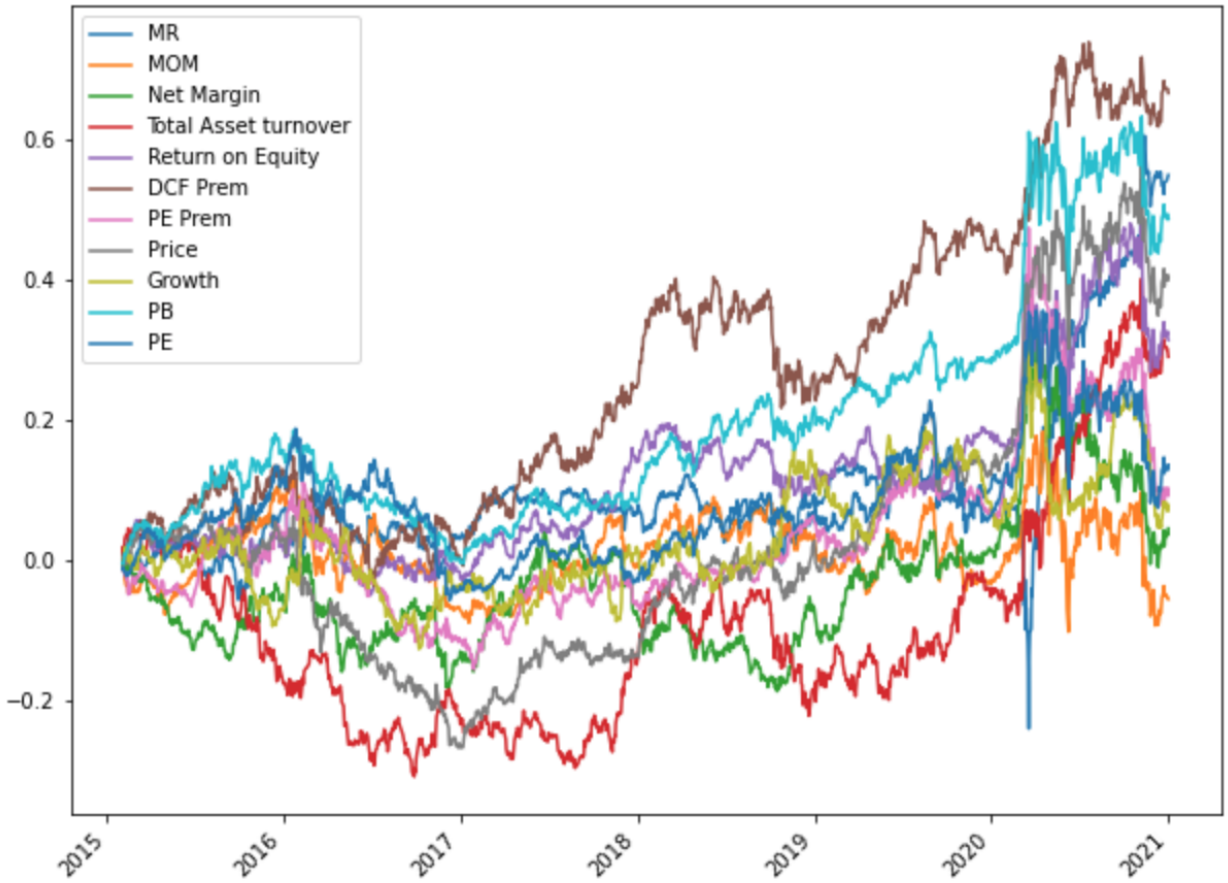



Figure 2: Return after scaling to the same volatility

| Factor | Mean | Std | Sharpe Ratio |
|---|---|---|---|
| MR | 2.190 | 12.691 | 0.173 |
| MOM | 1.411 | 15.908 | 0.089 |
| Net Margin | -1.284 | 7.673 | -0.167 |
| Total Asset Turnover | -3.505 | 8.997 | -0.390 |
| Return on Equity | 2.692 | 8.113 | 0.332 |
| DCF Premium | 3.666 | 7.550 | 0.486 |
| PE Premium | 1.076 | 6.641 | 0.162 |
| Growth | 1.621 | 5.722 | 0.283 |
| Price | 0.470 | 7.114 | 0.066 |
| PB | 6.352 | 10.198 | 0.623 |
| PE | 2.604 | 10.159 | 0.256 |

Table 2: Mean, standard deviation and Sharpe Ratio for factor portfolios over the training period

| Factor | Weight |
|---|---|
| MR | 0.074 |
| MOM | 0.000 |
| Net Margin | 0.000 |
| Total Asset Turnover | 0.000 |
| Return on Equity | 0.000 |
| DCF Premium | 0.092 |
| PE Premium | 0.000 |
| Growth | 0.493 |
| Price | 0.000 |
| PB | 0.341 |
| PE | 0.000 |

Table 3: Weights for the linear model

factors $w_k$ and their positions $p_{n,k}(t)$.

$$P_n(t) = \sum_{k=1}^{11} w_k p_{n,k}(t) \tag{7}$$

Then, the corresponding return, volatility and Sharpe Ratio are calculated.

$$R = 252 \sum_{k=1}^{11} w_k R_k \tag{8}$$

$$\sigma = \sqrt{252 \vec{W}^T \cdot \mathbf{V} \cdot \vec{W}} \tag{9}$$

where $\vec{W}$ is the weight vector, $\mathbf{V}$ is the covariance matrix of factor returns.

$$SR = \frac{R}{\sigma} \tag{10}$$

Objective: Maximize Sharpe Ratio
Constraints: $\sum_{k=1}^{11} w_k = 1$, $0 < w_k < 1$.
The optimization is done by *scipy.optimize.minimize* with solver *SLSQP* (Sequential Least Squares Programming). Instead of maximizing Sharpe Ratio directly, we minimize the negative Sharpe Ratio. The optimal weights are showed in Table 3

The table shows that the optimization assigns most weights to Growth, PB, DCF Premium and MR. These factors all have relatively high Sharpe Ratio among all factors and have low correlation with others.

#### 4.1.2 Portfolio Performance

The test portfolio is constructed as a linear combination of 11 factor portfolios with their weights determined from the training period. The Sharpe Ratio of the training and test portfolios are as the following:

| | Sharpe Ratio |
|---|---|
| In Sample | 0.84 |
| Out of Sample | 0.75 |

Since the test set was hidden while optimizing the weights, we do expect a lower out-of-sample Sharpe Ratio. Actually, if we compare the Sharpe Ratios in Table 2 and Table 1, we can tell that the Growth factor performs bad during the test period. Fortunately, the other three factors are still effective during the test period.

## 4.2 Machine Learning

### 4.2.1 Model Construction

The Machine Learning Method has mainly two steps:

- **Step 1:** Solve a classification problem:
  The predictor $Y$ is the sign of excess return defined as:

$$Y_{n,t} = \text{sign}[r_n(t) - \frac{1}{408}\sum_{N=1}^{408} r_N(t)] \tag{11}$$

  Therefore, $Y$ is a binary variable with balanced classes. We assign 1 to positive excess return, and 0 to negative excess return.
  The features $X_k$'s are the same 11 factors plus industry factors.
  We use logistic regression as the baseline and try other advanced ML algorithms: Decision Tree, Random Forest, and XGBboost.

- **Step 2:** Based on the prediction results, we long \$1 of each stock that is predicted to have positive excess return and short \$1 of each stock that is predicted to have negative excess return.

The mean return, standard deviation of return and Sharpe Ratio are calculated similarly as the Linear Model.

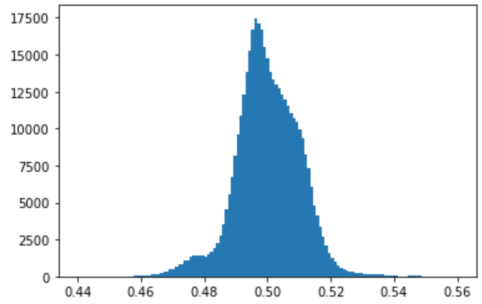
### 4.2.2 Logistic Regression

Logistic Regression takes $X_k$'s and $Y$'s from the training set to train the model. The model predicts the probability of $y_{n,t} = 1$. We set the threshold to be 0.5, which means if the probability of $y_{n,t} = 1$ is greater than 0.5, $y_{n,t}$ is predicted to be 1, which means stock $n$ is predicted to have return higher than the median of all stock returns at time $t$.
We start with using positions in factor portfolios from Linear Model as features. We have positions in factor portfolio k $p_k$ as feature $X_k$. Our goal is to use the 11 features to predict the sign of the excess return. By setting the threshold as 0.4997, the in-sample accuracy is maximized to 0.5066.
To improve the accuracy, we turn to another method that is to use raw signals as features. We have 11 features plus industry factor as dummy variables. In the previous method, the

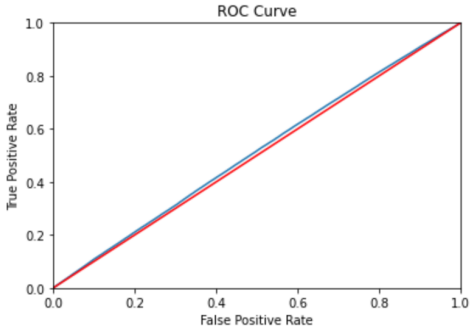
| Factor | Logistic Coefficient | Correlation |
|---|---|---|
| MR | 0.008 | 0.004 |
| MOM | 0.020 | 0.012 |
| Net Margin | -0.002 | 0.003 |
| Total Asset Turnover | 0.010 | 0.0008 |
| Return on Equity | -0.003 | -0.001 |
| DCF Premium | 0.002 | 0.001 |
| PE Premium | 0.006 | 0.004 |
| Growth | -0.001 | -0.000 |
| Price | -0.002 | 0.000 |
| PB | -0.002 | -0.001 |
| PE | 0.005 | 0.005 |

Table 4: Coefficients and correlations

features only take values of 1 and 0 with 1 meaning above median and 0 meaning below median, which could cause a lose in information from data. By using the raw signals as features, we expect a higher in-sample accuracy.
The histogram in Figure 3 shows the distribution of in-sample prediction results and the ROC. The prediction results are all around 0.5. We also tune the threshold to maximize the in-sample accuracy. The threshold is set to be 0.4996, which gives an in-sample accuracy of 0.5090 and out-of-sample accuracy of 0.5063.



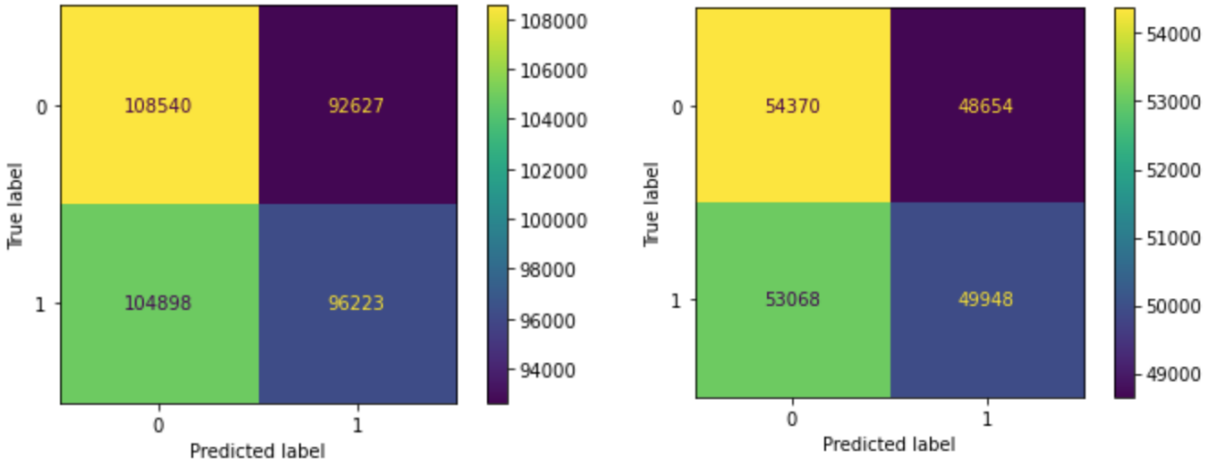

(a) Histogram of in-sample prediction results




(b) In-Sample ROC curve

Figure 3: In-Sample prediction results

By comparing the coefficients of the features from logistic regression and the correlation between features and the sign of excess return showed in Tabel 4, it is verified that the logistic model is effective. We can tell that the correlation is overall consistent with coefficients, in terms of that most factors with significantly higher coefficients in regression also have higher correlation with the sign of excess return and negative correlation matches negative coefficients. Therefore, we are confident that the regression is working as expected. The classification results are showed in Figure 4. We notice that the prediction classes are slightly unbalanced. To maintain market neutral, we multiply the long position by the amount of




(a) In-Sample

(b) Out-of sample

Figure 4: Confusion Matrix from Logistic Regression

| | Accuracy | Sharpe Ratio |
|---|---|---|
| In Sample | 0.5090 | 0.72 |
| Out of Sample | 0.5063 | 0.14 |

Table 5: Accuracy and Sharpe Ratio from Logistic Regression

predicted 0's over the amount of predicted 1's everyday so that the sum of positions is 0 everyday.
The corresponding Sharpe Ratio is showed in Table 5.

### 4.2.3 Decision Tree

We tune the hyper-parameters to be $max\_depth = 7$ and others by default, and set the threshold to maximize average accuracy from a 3-fold Cross Validation. Another advantage of using Cross Validation is to avoid overfitting in the training set, which is a common problem in tree based classifiers. The optimized threshold is 0.506.
Decision Tree shows a similar feature importance as Logistic Regression in terms of that most factors with higher absolute values of coefficients in Logistic Regression also have higher feature importance in Decision Tree as showed in Table 6. The prediction classes are still unbalanced as showed in Figure 5. We implemented the same method as in Logistic Regression to maintain market neutral. The in-sample accuracy is improved to 0.5143 and the out-of-sample accuracy is improved to 0.5093, which leads to higher in-sample and out-of-sample Sharpe Ratios.

| Factor | Logistic Coefficient | Decision Tree Feature Importance |
|---|---|---|
| MR | 0.008 | 0.125 |
| MOM | 0.020 | 0.192 |
| Net Margin | -0.002 | 0.071 |
| Total Asset Turnover | 0.010 | 0.070 |
| Return on Equity | -0.003 | 0.065 |
| DCF Premium | 0.002 | 0.055 |
| PE Premium | 0.006 | 0.052 |
| Growth | -0.001 | -0.062 |
| Price | -0.002 | 0.043 |
| PB | -0.002 | 0.079 |
| PE | 0.005 | 0.078 |

Table 6: Coefficients and Feature Importance

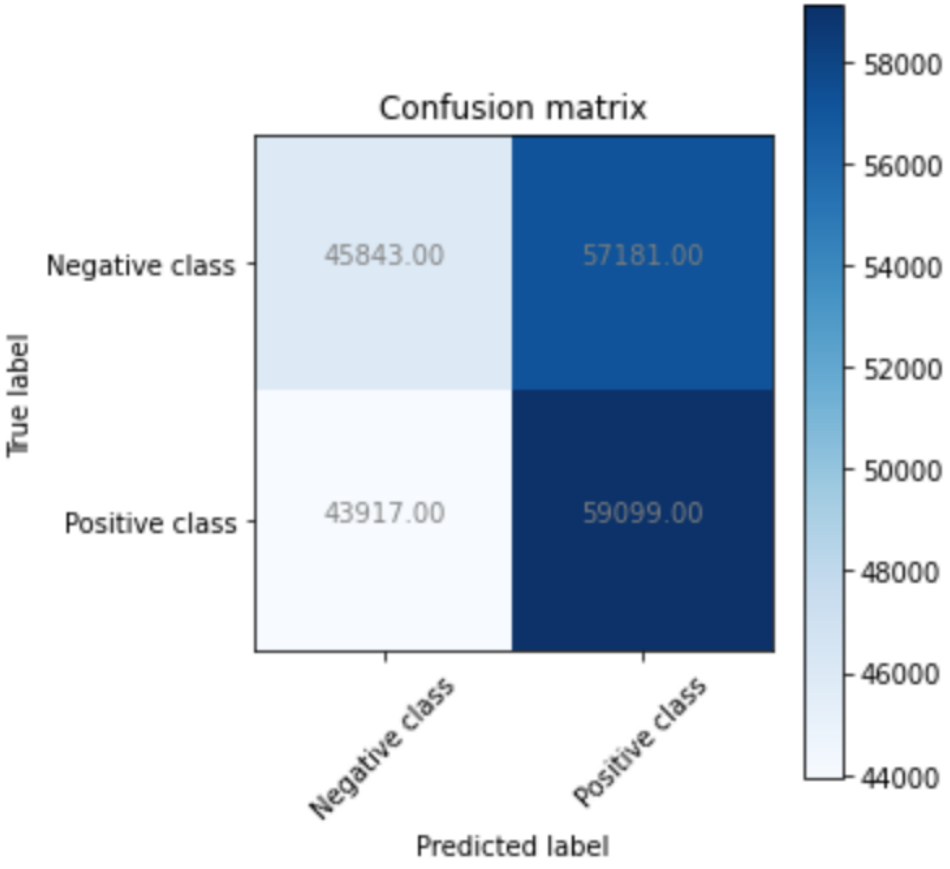



Figure 5: Out-of-sample confusion matrix from Decision Tree

| | Accuracy | Sharpe Ratio |
|---|---|---|
| In Sample | 0.5143 | 2.45 |
| Out of Sample | 0.5093 | 1.13 |

Table 7: Accuracy and Sharpe Ratio from Decision Tree

#### 4.2.4 Other Machine Learning Algos

This section shows other tryouts in Machine Learning.
Random Forest as an advanced version of Decision Tree was expected to perform better, but due to computational complexity, it is hard to optimize hyper-parameters or implement cross validation. Same as XGBoost, a tree-based classifier that uses a gradient boosting framework. Those two models both have severe overfitting problems. For example, the in-sample accuracy for XGBoost is 0.5719 but only 0.5028 out-of-sample. Therefore, we decide not to move forward with these two models at this point.

# 5 Conclusion and Other Thoughts

## 5.1 Market Analysis of the Factors

First of all, note that our DCF premium factor generally outperformed all the other factors. This is because this factor is a growth factor, and favors long-term high-growth and high price-per-share companies; these companies have generally performed well from 2015 to present. Also, as shown in figure 2, this factor performed extremely well in early-mid 2020, a period where growth stocks skyrocketed. So, the performances of our DCF premium factor are consistent with the overall market conditions.
The relatively poor performance of the PE premium factor, even though also as a growth factor similar to our DCF premium factor, is worthy of analysis. First of all, the model only takes into account a company's short-term growth, namely in one year, while the DCF premium takes into account a company's growth all the way into the future. Also, the DCF premium has a higher correlation with the price per share factor than the PE premium (0.68 vs 0.24); since from 2015, as we can see in Figure 2, the price-per-share factor has outperformed the growth factor, the better performance of the DCF premium factor could be explained.
The performances of mean reversion and momentum factors are also interesting. These two factors fluctuate significantly starting in 2020. From February 2020, the market entered a time of extreme volatility, so the mean reversion factor performed very well from February 2020. Also in the same period, the momentum factor underperformed significantly as market trends simply do not persist during this period.
Finally, it is noteworthy that the performances of some fundamental factors (such as total asset turnover, return on equity, etc.) are rather unappealing. This may suggest that from 2015, fundamental factors and fundamental analysis have not been the deterministic factor in driving stock prices.
To summarize, there are several hypotheses that we can make from observations of these

factors' performances from 2015:

1. The market overlooks the fundamental aspects of the companies. The poor performances of the fundamental factors and the growth factor is the best evidence.

2. The market favors long-term growth, rather than short-term growth. The significant outperformance of the DCF premium factor over the PE premium factor supports this argument.

3. The previous 6 years in the financial markets have been very volatile, as can be seen by the significant outperformance of the mean reversion factor over the momentum factor supports this argument. In other words, market trends do not persist, they fluctuate often in the frequency of a few months.

## 5.2 Conclusion on Models

If we compare our Linear Model and Machine Learning Models we can see results of Sharpe Ratio as the following:

| | Linear | Logistic Regression | Decision Tree |
|---|---|---|---|
| In Sample | 0.84 | 0.72 | 2.45 |
| Out of Sample | 0.75 | 0.14 | 1.13 |

The cumulative performance of our 3 strategies in the test set is showed in Figure 6.

It's critical to understand the different results from linear model and machine learning models. The reason of the lower SR of from Logistic Regression is that we only assign positions of +1, -1 to all stocks. Let's use Example 1, in such a four-stock portfolio, the position vector $p_t$ on any arbitrary day $t$ will be in the form of the following with only signs changing:

$$p_t = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \tag{12}$$

By contrast, in the linear model, by maximizing the SR for the training period, the positions for all stocks can be fractional. Still using Example 1, assume Factor Portfolio 1 is assigned a weight of 0.6 and Factor Portfolio 2 is assigned a weight of 0.4, the position vector $p_t$ on any arbitrary day $t$ will be in the form of the following:

$$p_t = 0.4 \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} + 0.6 \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} +0.2 \\ -1 \\ +1 \\ -0.2 \end{bmatrix} \tag{13}$$

It follows that in machine learning models, high excess return stocks among all stocks with

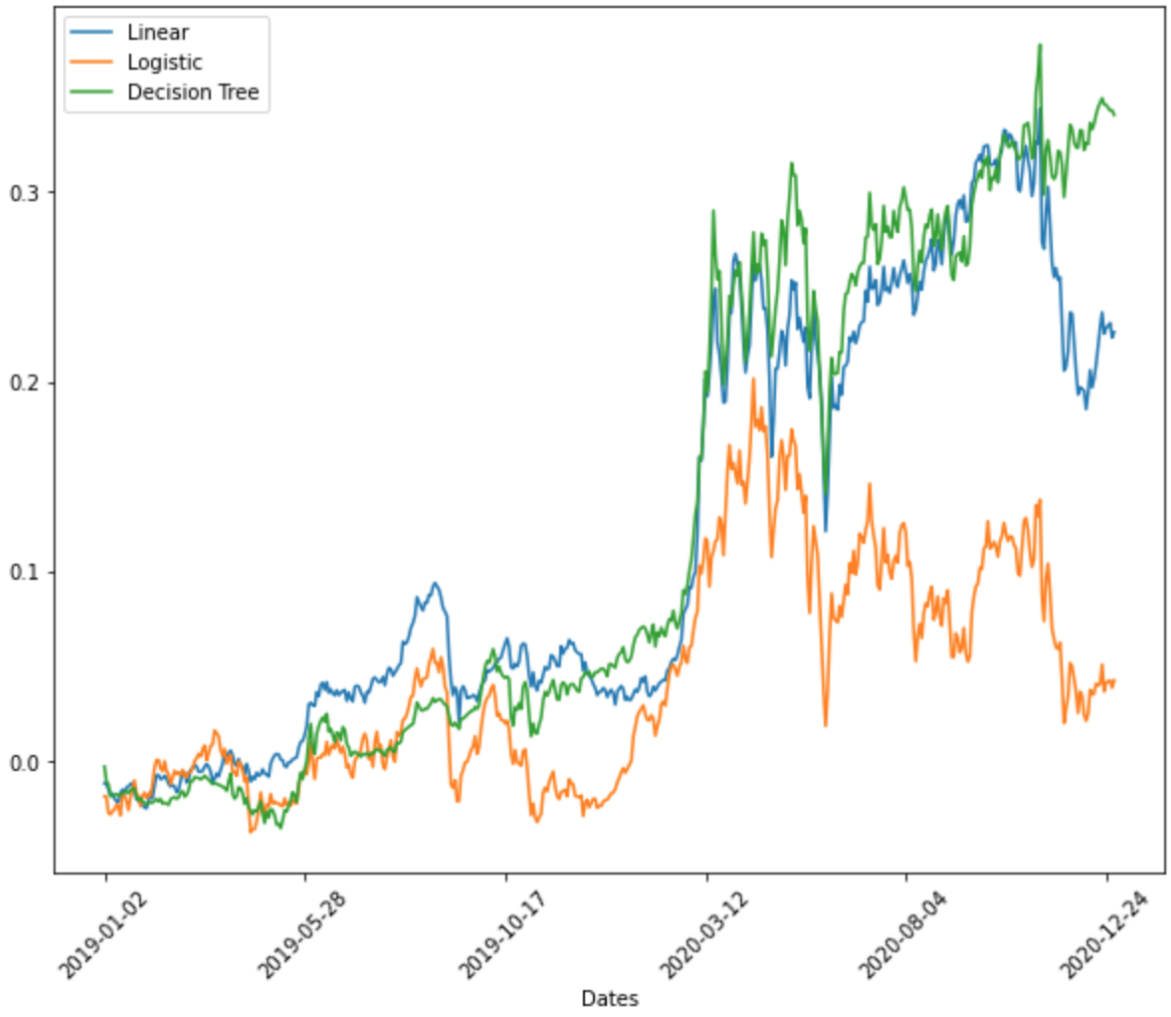



Figure 6: Cumulative performance after scaling to the same volatility

positive excess return are not distinguished, while in the linear model, a stock with higher return is very likely to be assigned a higher position which leads to an overall high SR. Therefore, to improve the SR for ML methods, we have to significantly increase the prediction accuracy like what is done by Decision Tree or by adding more powerful factors. Otherwise, we will need to do a regression instead of classification so that we can assign positions according to the magnitude of predicted excess return which I believe is more comparable to the linear method and will lead to a higher Sharpe Ratio.

All codes and data can be found here: `https://github.com/eliaiye/Equity-Team6`