Kernel Trading Project

By: Xuyang 'Dennis' Hua

Abstract:

- The primary objective of this project is to assess the predictive capabilities of various feature and risk datasets across a selection of securities.
- Following an extensive exploration of hypotheses and ideas regarding predictive power and pattern recognition within the feature data, I proceeded to develop corresponding trading strategies. These strategies were rigorously back-tested using a comprehensive securities dataset spanning from 2010 to 2018. This evaluation involved the measurement of key statistical metrics such as the Sharpe ratio, maximum drawdown, annual yield, volatility, and more. The purpose was to discern performance disparities among different factors and strategies, thus validating the initial hypotheses.
- Subsequently, from the pool of viable strategies, I conducted an analysis of the correlation and covariance among their returns. This analysis laid the foundation for the creation of a multifactor strategy model. The allocation of weights to individual strategies within the portfolio was determined using the mean-variance theorem, with the goal of optimizing the Sharpe ratio for the overall portfolio. This step aimed to construct a diversified and efficient investment portfolio that harnesses the strengths of each strategy while mitigating risk.

Data Preprocessing:

To consolidate the data for all securities, I amalgamated their respective datasets into a unified dataset. This amalgamation involved merging 11 features and 6 risk-related data columns based on the security ID and data date. In instances where data did not exist for a particular security, the corresponding entries were populated with 'Nan' to maintain data integrity.

To bolster the statistical robustness, precision, generalizability, minimize bias, and uphold the validity of hypothesis testing, a stringent filtering process was enacted. Securities with fewer than 250 days' worth of data were systematically excluded from the analysis. Additionally, only those securities boasting a minimum of one year's worth of closed price data were retained for further analysis.

Given the substantial divergence in feature data across various securities, and the prevalence of 'Nan' values in many feature columns for most securities, a tailored approach was adopted. Each security's data was individually processed, with 'Nan' values in feature columns addressed by replacing them with the median value across all time points. Feature columns where 'Nan' values exceeded 50% were deemed unreliable and subsequently removed from consideration.

Notably, all features and risk factors data are collected post-market close. Consequently, any analysis and subsequent trading signals must be generated on the following day. To align the data appropriately, the 'ret1d' variable, representing the percentage change in today's price relative to the previous day's closing price, required a two-row shift. This adjustment ensures the data analysis is both meaningful and feasible.

Following the comprehensive preprocessing of securities within their respective dataframes, the data from all securities were consolidated into a unified, complete dataset, ready for subsequent analysis.

Transaction Cost Analysis:

To account for transaction costs, which amount to 0.01% of the trading notion, and given the dynamic nature of the portfolio with a requirement to maintain a market-neutral position, I introduced a new column called 'weights.' This column represents the allocation of capital to individual securities on a given day. For example, in the long portfolio with 250 securities, each security is assigned a weight of 1/250, and in the short portfolio with 150 securities, each security receives a weight of 1/150. This allocation ensures that the overall position remains at 0, maintaining a market-neutral stance.

To calculate the transaction cost, we determine the difference in weights for each individual security, taking the absolute value of these differences, and then multiply the sum of these absolute differences by 0.01%. This process enables us to derive the total transaction cost accurately.

Model Selection:

OLS: Linear regression can be a reasonable choice for predicting future returns, but its suitability depends on the specific characteristics of your data and whether the underlying assumptions are met.

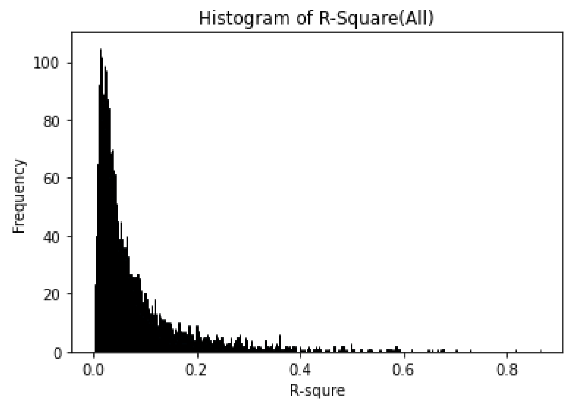
Random forest: Random Forest can be a strong choice for predicting future returns, especially when dealing with complex relationships and large datasets. However, as with any modeling approach, it is important to carefully preprocess data, tune hyperparameters.

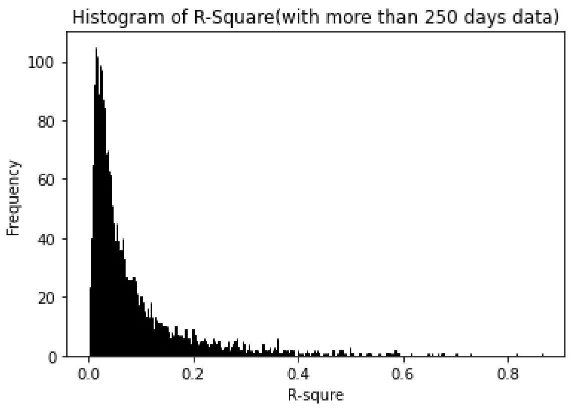
Multi-factors Model: Multifactor model attempt to find the valid factor in each feature and based on the correlation among the valid factors to form a comprehensive model which might explain the securities return.

OLS:

Since OLS is not sensitive to the missing value, I filled up all the Nan in the features data with 0 to give better sense of how those features can be related with next day securities returns. Since different securities have different kind of features and are quite independent with each other, I run the OLS on single security.

Here is the graph of R square histogram: (Left is all security OLS R Square; Right is securities with more than 250 days data)





In both cases, most of the securities R square fall in [0,0.2] range and only very small portion of the securities can be partially explained by OLS. The OLS model does not fit the data well, and a significant portion of the variation in the dependent variable remains unexplained by the model.

Random Forest:

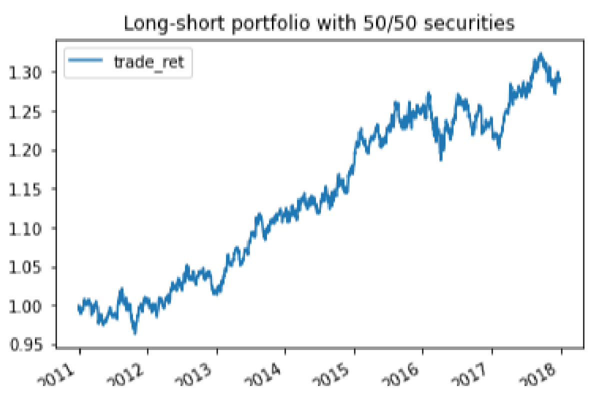
When utilizing the Random Forest algorithm to forecast future security returns, I employ a preprocessed dataset as my foundation. In this dataset, I implement a data quality criterion where any features with less than 50% valid data are removed, and for the remaining columns, I judiciously impute missing values with the respective column medians. Additionally, I apply a filter to exclude securities that possess fewer than 250 data points, ensuring that our analysis focuses on sufficiently robust cases.

Each individual security is treated as a distinct entity, warranting the creation of dedicated Random Forest models. To optimize computational efficiency, I have implemented a periodic retraining scheme. Specifically, the models are refreshed every 250 days. This approach strikes a balance between accuracy and computational feasibility, as an attempt to train the models daily proved impractical due to the prohibitively long 45-hour training time. While setting the retraining period to 250, it took 6 hours to fully trained every security’s models.

The initial 250 days serve as the foundational training set for each security. Subsequently, I employ these trained models to make predictions for the following 250 days of each security's return. These newly predicted data points are seamlessly integrated into the training dataset, forming a cumulative dataset of 500 days. This process iterates continuously until it covers the entirety of each security's data, ensuring that our models remain up-to-date and well-informed throughout the analysis.

Merge all securities dataframes. Generate the trading signal by ranking all the predicted returns. Long the top 50 securities with highest predicted return and short the bottom 50 securities.

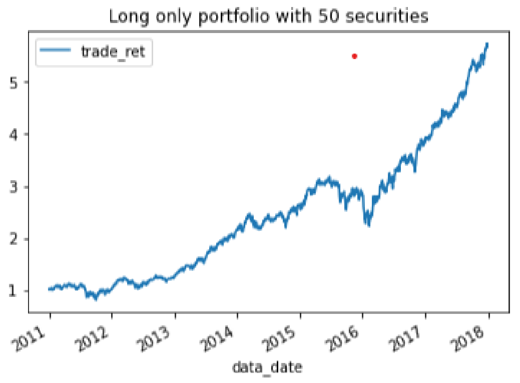
```
                                trade_ret
Annual Return %                      3.70
Sharpe Ratio                         0.66
Sortino                              0.06
Omega                                1.12
Volatility %                         5.59
semi-variance %                      0.25
VaR(30days) %                       -2.60
Max drawdwon %                       6.86
return/Max drawdwon                  0.54
positive rate %                     58.33
1 year return (rolling) %            4.23
3 year return (rolling) %           15.20
5 year return (rolling) %           24.55
1 year rolling VaR(5%) %            -1.27
```

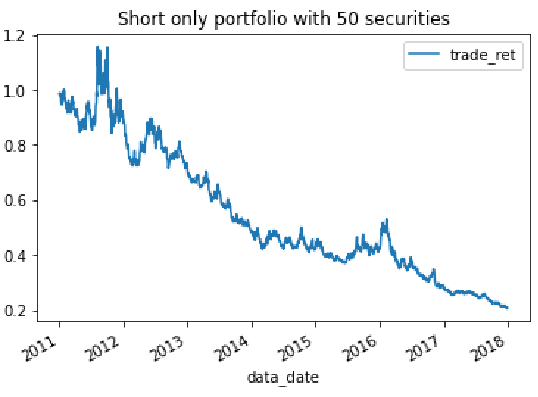


The strategy gives 3.7 annually return and 0.66 Sharpe ratio while I did not include the trading cost yet. The upward the pattern is stable with some acceptable drawdown. However, while we look at the long-only and short-only samples, the results are not quite well.

```
                                trade_ret
Annual Return %                     27.94
Sharpe Ratio                         1.31
Sortino                              0.11
Omega                                1.24
Volatility %                        21.28
semi-variance %                      0.94
VaR(30days) %                      -10.08
Max drawdwon %                      29.86
return/Max drawdwon                  0.94
positive rate %                     71.43
1 year return (rolling) %           30.59
3 year return (rolling) %          117.14
5 year return (rolling) %          256.34
1 year rolling VaR(5%) %            -3.13
```

```
                                trade_ret
Annual Return %                    -20.00
Sharpe Ratio                        -0.82
Sortino                             -0.07
Omega                                0.87
Volatility %                        24.33
semi-variance %                      1.09
VaR(30days) %                      -11.23
Max drawdwon %                      82.27
return/Max drawdwon                 -0.24
positive rate %                     40.48
1 year return (rolling) %          -18.95
3 year return (rolling) %          -45.79
5 year return (rolling) %          -64.47
1 year rolling VaR(5%) %           -34.58
```



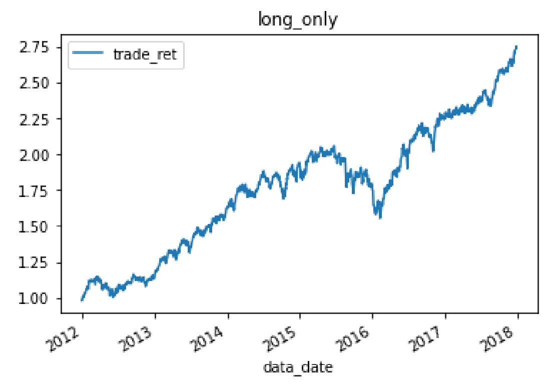


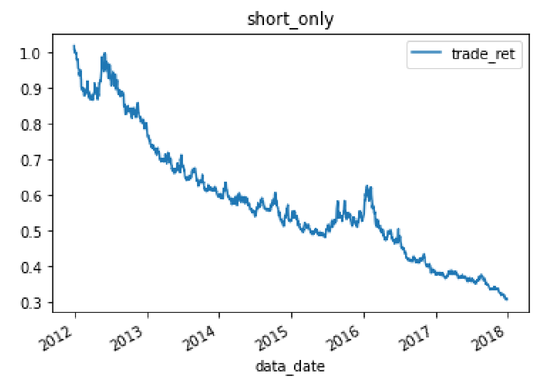
The pattern of the long-only portfolio is highly correlated with the overall the securities market price change over the past 8 years. While the short-only portfolio actually gives quite large negative returns, the short-only strategy is following the same pattern of the Long only strategy's curve with position flipped.

Simply ranking the preditive return might be questioned since the volatility for each security is different, some securities might have bigger fluactuationin daily prices which makes them always in the portfolio. Therefore, I decided to converted all the predicted returns to historical ranking, converting all the predicted return to their rank in previous 250 days rolling window and get a number between 0 and 1. Then use this standardized value to generate the traidng signal.

```
                              trade_ret
Annual Return %                   18.10
Sharpe Ratio                       1.25
Sortino                            0.11
Omega                              1.22
Volatility %                      14.51
semi-variance %                    0.63
VaR(30days) %                     -6.83
Max drawdwon %                    24.56
return/Max drawdwon                0.74
positive rate %                   69.86
1 year return (rolling) %         17.84
3 year return (rolling) %         50.52
5 year return (rolling) %        119.45
1 year rolling VaR(5%) %          -8.90
```

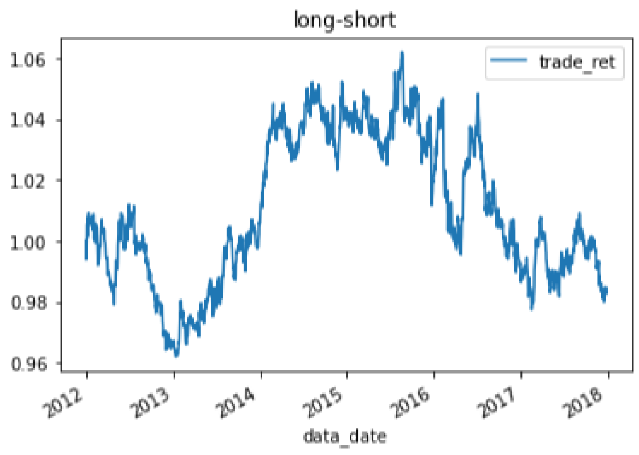
```
                              trade_ret
Annual Return %                  -17.65
Sharpe Ratio                      -1.09
Sortino                           -0.10
Omega                              0.83
Volatility %                      16.22
semi-variance %                    0.75
VaR(30days) %                     -7.90
Max drawdwon %                    69.84
return/Max drawdwon               -0.25
positive rate %                   30.14
1 year return (rolling) %        -15.87
3 year return (rolling) %        -35.96
5 year return (rolling) %        -59.19
1 year rolling VaR(5%) %         -30.04
```





The random forest model did not successfully capture the pattern from the features and the linearity of the predicted return did not generate valid trading signal for the strategy.

```
                              trade_ret
Annual Return %                   -0.28
Sharpe Ratio                      -0.07
Sortino                           -0.00
Omega                              0.99
Volatility %                       3.97
semi-variance %                    0.18
VaR(30days) %                     -1.80
Max drawdwon %                     7.95
return/Max drawdwon               -0.04
positive rate %                   45.21
1 year return (rolling) %          0.05
3 year return (rolling) %          1.16
5 year return (rolling) %          0.18
1 year rolling VaR(5%) %          -4.30
```

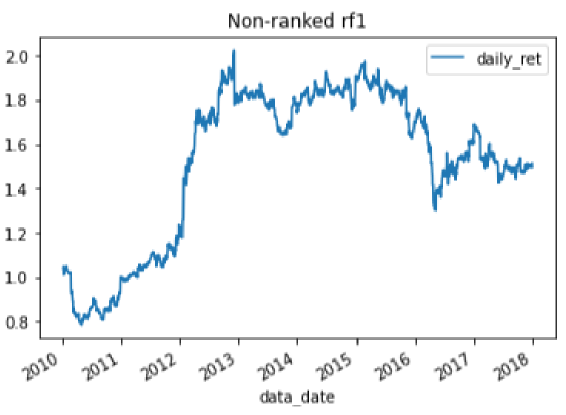


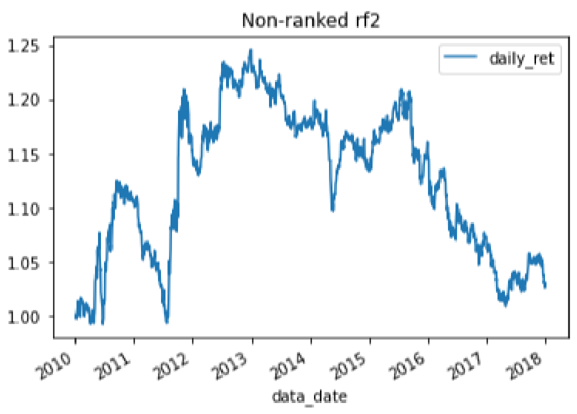
The long-short strategy got even worse result compared to directing using the predicted value. Therefore I decided to move use other way to find the relationship between features and future returns.

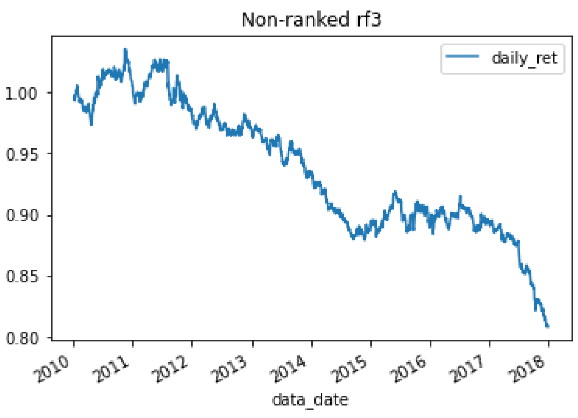
Multi-factors model:

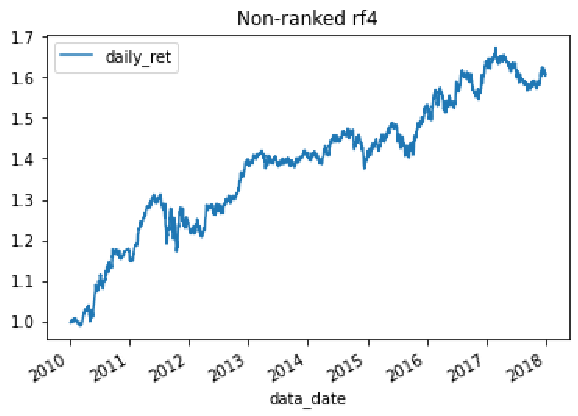
While neither OLS nor Random Forest yielded satisfactory results, I made the decision to delve deeper into the underlying risk factors. Initially, I sought to assess the performance of directly utilizing each individual risk factor to generate a trading signal. To accomplish this, I segregated all securities into individual dataframes and calculated both the mean and standard deviation for each risk factor. Subsequently, I merged these dataframes together.

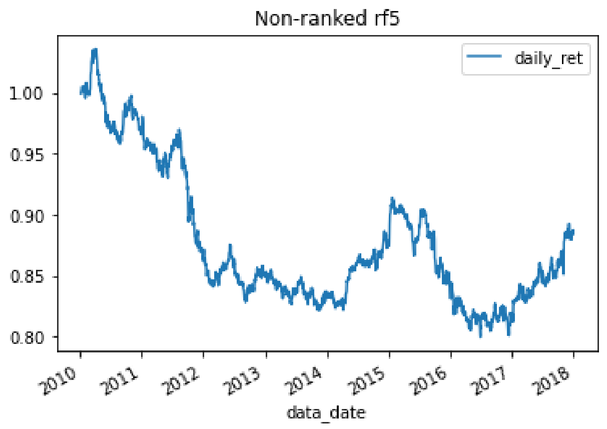
The rationale behind generating the trading signal is based on the concept that when a risk factor deviates by one standard deviation from its mean, a signal is triggered. The decision to either go long or short on securities is contingent upon the correlation between these risk factors and the 'ret1d' column. If the correlation is positive, a higher risk factor generates a long signal, while if it's negative, a higher risk factor generates a short signal. Conversely, the opposite holds true. Based on the correlation matrix r1,r2,r3 are negative correlated with next period return while r4,r5,r6 are postive correlated:

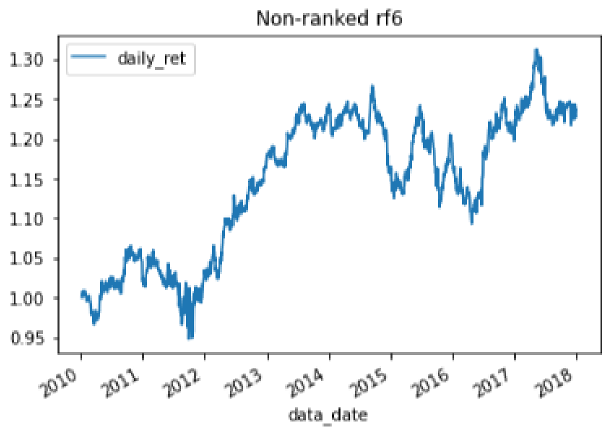




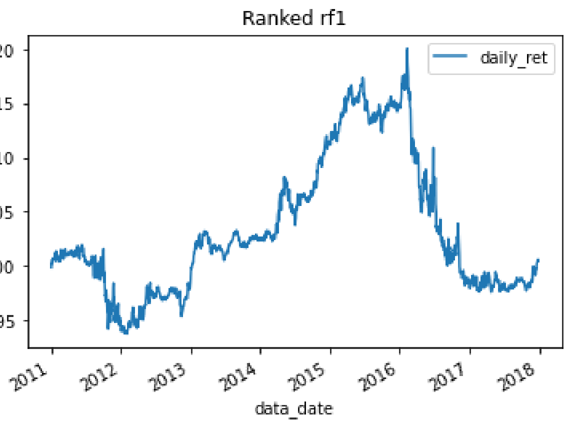


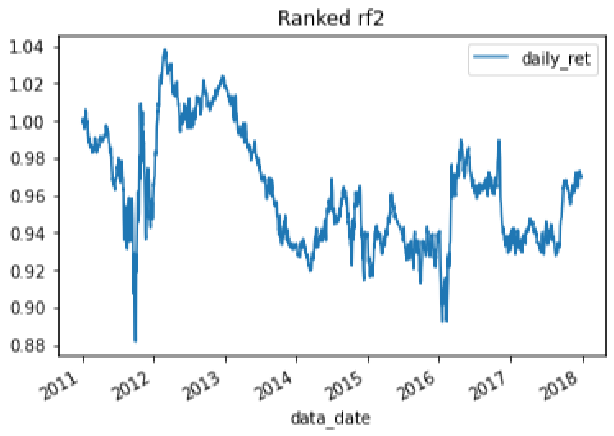


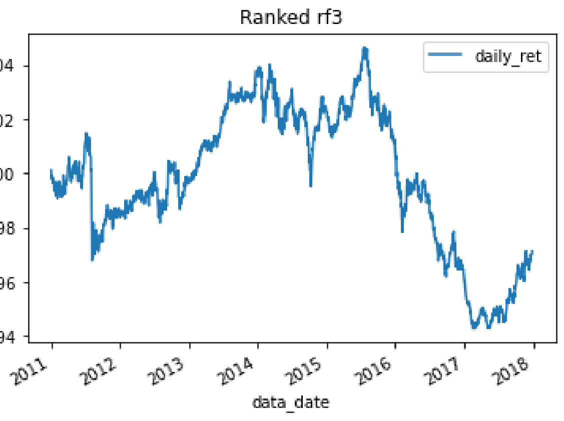


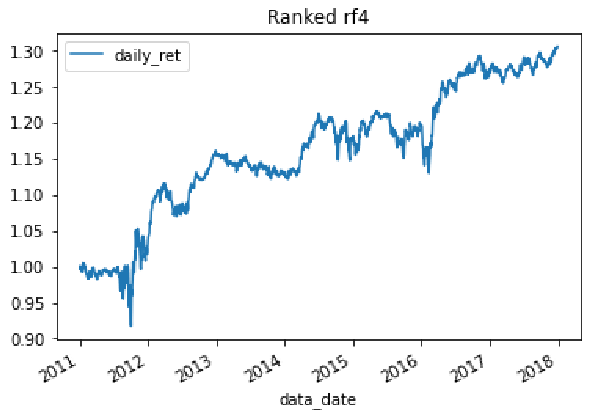


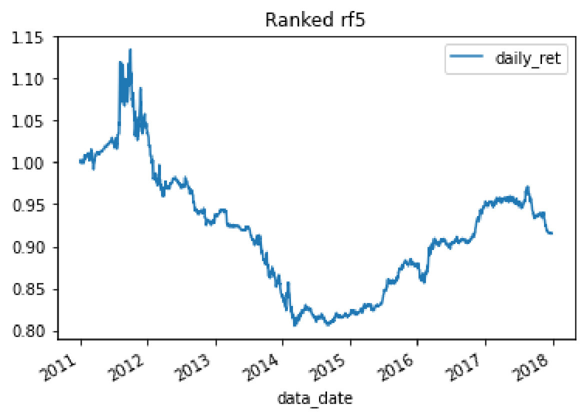
Besides using the standard deviation and mean to generate the trading signal, I also turn the risk factor into the historical ranking and use the percentile ranking in last 250 trading days to generate the signals. Similar to the previous process in the random forest model, I separate all the securities in sub-dataframe and use rolling window to convert all the risk factors to a number between 0 and 1. After merge to final dataframe, I used 0.75 and 0.25 as threshold to generate the trading signal. While the risk percentile is bigger than 0.75 or smaller than 0.25, the signals will be generated based on the correlation as we found in the last experiment.

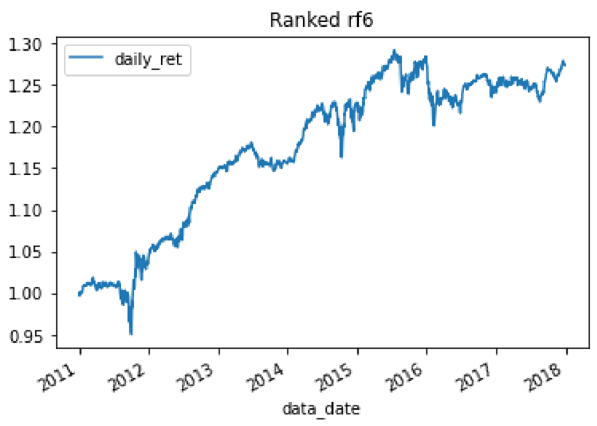




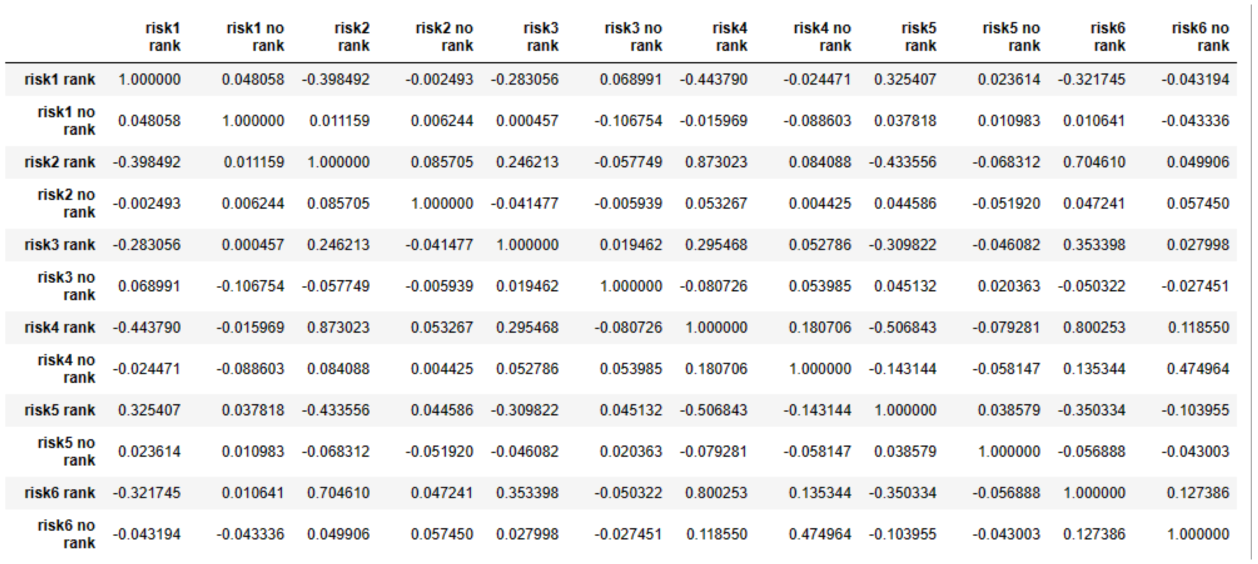






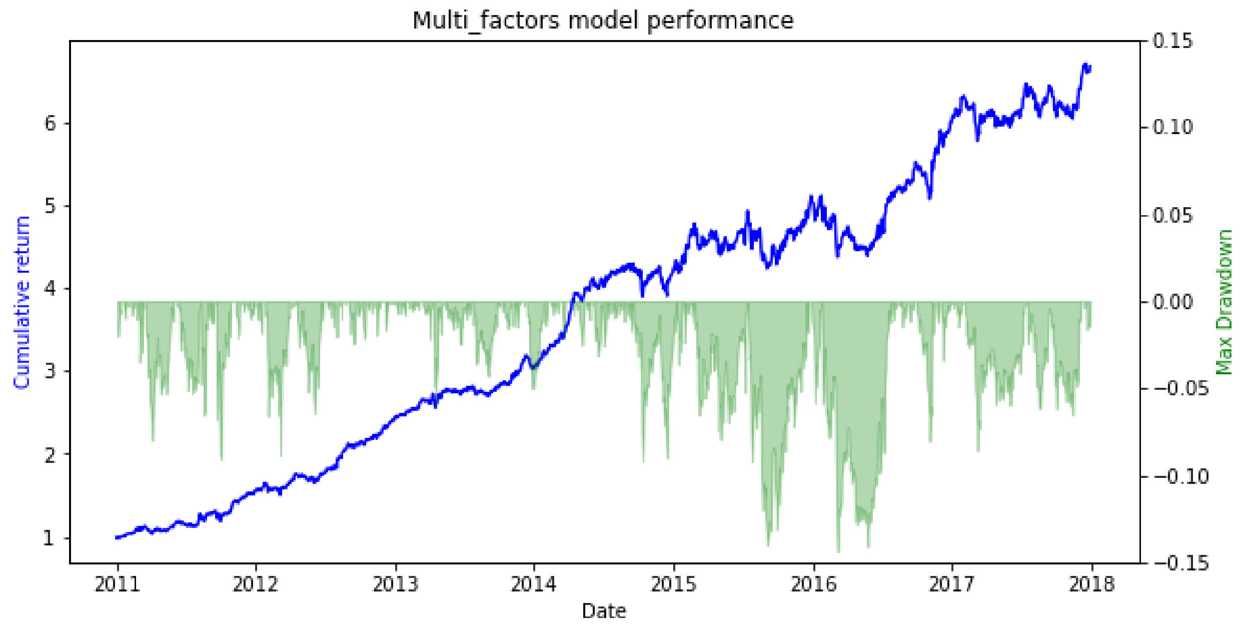


As evident from the backtest graph, certain risk factors exhibited predictive capabilities concerning future returns. Specifically, Risk Factor 4 and Risk Factor 6 displayed a consistently ascending pattern. To ascertain whether these risk factors are uncorrelated with each other, with the aim of constructing a more robust multifactor model, I examined the correlation matrix.



	risk1 rank	risk1 no rank	risk2 rank	risk2 no rank	risk3 rank	risk3 no rank	risk4 rank	risk4 no rank	risk5 rank	risk5 no rank	risk6 rank	risk6 no rank
risk1 rank	1.000000	0.048058	-0.398492	-0.002493	-0.283056	0.068991	-0.443790	-0.024471	0.325407	0.023614	-0.321745	-0.043194
risk1 no rank	0.048058	1.000000	0.011159	0.006244	0.000457	-0.106754	-0.015969	-0.088603	0.037818	0.010983	0.010641	-0.043336
risk2 rank	-0.398492	0.011159	1.000000	0.085705	0.246213	-0.057749	0.873023	0.084088	-0.433556	-0.068312	0.704610	0.049906
risk2 no rank	-0.002493	0.006244	0.085705	1.000000	-0.041477	-0.005939	0.053267	0.004425	0.044586	-0.051920	0.047241	0.057450
risk3 rank	-0.283056	0.000457	0.246213	-0.041477	1.000000	0.019462	0.295468	0.052786	-0.309822	-0.046082	0.353398	0.027998
risk3 no rank	0.068991	-0.106754	-0.057749	-0.005939	0.019462	1.000000	-0.080726	0.053985	0.045132	0.020363	-0.050322	-0.027451
risk4 rank	-0.443790	-0.015969	0.873023	0.053267	0.295468	-0.080726	1.000000	0.180706	-0.506843	-0.079281	0.800253	0.118550
risk4 no rank	-0.024471	-0.088603	0.084088	0.004425	0.052786	0.053985	0.180706	1.000000	-0.143144	-0.058147	0.135344	0.474964
risk5 rank	0.325407	0.037818	-0.433556	0.044586	-0.309822	0.045132	-0.506843	-0.143144	1.000000	0.038579	-0.350334	-0.103955
risk5 no rank	0.023614	0.010983	-0.068312	-0.051920	-0.046082	0.020363	-0.079281	-0.058147	0.038579	1.000000	-0.056888	-0.043003
risk6 rank	-0.321745	0.010641	0.704610	0.047241	0.353398	-0.050322	0.800253	0.135344	-0.350334	-0.056888	1.000000	0.127386
risk6 no rank	-0.043194	-0.043336	0.049906	0.057450	0.027998	-0.027451	0.118550	0.474964	-0.103955	-0.043003	0.127386	1.000000

Among the various correlations examined, only a few pairs exhibited relatively high correlations, reaching values as high as 0.8. In contrast, most of the strategy sets displayed low correlations with one another. Consequently, I opted to apply the mean-variance theorem to amalgamate all twelve factors in an effort to maximize the Sharpe ratio. Upon combining these factors, the overall performance is summarized as follows:



Annual Return %	30.92
Sharpe Ratio	2.11
Sortino	0.19
Omega	1.40
Volatility %	14.68
semi-variance %	0.60
VaR(30days) %	-6.48
Max drawdwon %	14.38
return/Max drawdwon	2.15
positive rate %	71.43
1 year return (rolling) %	32.81
3 year return (rolling) %	127.73
5 year return (rolling) %	274.26
1 year rolling VaR(5%) %	0.94

	Tangency Weights
risk1 rank	0.262839
risk1 no rank	0.202006
risk2 rank	-3.016596
risk2 no rank	0.136486
risk3 rank	-1.265950
risk3 no rank	-1.887674
risk4 rank	2.539639
risk4 no rank	0.875946
risk5 rank	0.054643
risk5 no rank	-0.322421
risk6 rank	1.664466
risk6 no rank	-0.243385

Conclusion:

OLS and Random Forest Model did not provide strong predictive power and explantion for the relationship between features and the future returns.

The multi-factors model gain 2.11 sharpe ratio with 30.92% annually return, which represents that the risk factors can be used as a good signal generator after we standardized and use z-score and percentile to distinguished extreme values. The mean-variance allocation can significantly improve those factors combinations' performance as factors have a relatively low correlation with each other.

Future Improvements:

- Both the random forest model and OLS have proven ineffective in extracting meaningful characteristics from the features data across files 1 to 11. Given the substantial amount of missing data for many securities in these files, I have made the decision to exclusively utilize risk factors for constructing the multifactor model. In the future, if time permits, I intend to conduct a more thorough analysis of these 11 datasets to uncover additional data patterns.
- More advanced machine learning models such as gradient boosting and convolutional neural networks (CNN) hold promise for enhancing performance. However, deploying these models demands additional time for training and a more comprehensive parameter tuning process, which I have not undertaken in this project.
- Expanding the time span of testing is crucial. In this project, my primary focus has been on strategies that change positions on a daily basis. Given more time, it would be advantageous to assess the performance of these factors and models over weekly and monthly intervals to gain a comprehensive understanding of their effectiveness.
- The transaction cost has had a noticeable negative impact on performance, reducing returns by 2% annually. To potentially mitigate this effect, I could consider making transaction costs an active parameter in the strategy design. This would involve factoring in transaction costs when determining the optimal thresholds and parameters for each factor and strategy, thereby optimizing performance while accounting for these costs.