Feature Engineering for Stocks in Tai Wan Stock Markets

Li Longxin
GU Enhao
DU Manhua

Table of Contents

Part 1 Introduction

During this summer, we mainly explored feature engineering tactics and models of return forecasting for stocks in Taiwan stock market, mainly one index-0050(Taiwan Top 50 ETF) and one stock-2330 (Taiwan Semiconductor Manufacturing Company Limited). The original aim of this project is to explore tactics of feature engineering for multi factor models, such as sorting of optimal features, feature combination and feature effectiveness analysis. Besides, we also generate an interest on finding the best model of predicting stock return and compare the performances of different models. Hence, our research can be divided into two parts.

The first part is feature selection and combination. We work on selecting key parameters from 48 factors, combining different factors to create new ones and compare predictive ability of model in which different factors are included. The models we have used are：

- **Lasso：** (least absolute shrinkage and selection operator) A regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model.
- **PCA:** A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables.

The second part is model selection. Our initial motivation for this part was to find for different models, the optimal length of training period and the length of the model remains useful. The models we have used are：

- **GBRT:** It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.
- **Linear regression:** The first type of regression analysis to be studied rigorously, and to be used extensively in practical applications.
- **PLS:** Partial Least Squares Regression, it finds a linear regression model by projecting the predicted variables and the observable variables to a new space.
- **LASSO:** A regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model.
- **RIDGE:** Similar to Lasso regression, commonly used to approximate an answer for an equation with no unique solution.

Part 2 Feature Selection & Creation

1. Methodology Introduction

a) T-value feature selection

Feature selection has been recognized as one of the most critical technique to boost the performance of an algorithm. T-statistics has proved its effectiveness among different methods in feature selection, which is an indicator to describe the importance of a feature. The t-statistics could be computed as follows:

$$t(X_i) = \frac{|\mu_{i1} - \mu_{i2}|}{\sqrt{\frac{\sigma_{i1}^2}{n_1} + \frac{\sigma_{i2}^2}{n_2}}}$$

where u_{ij} denotes mean of i-th feature X_i for class C_j and $sigma_{ij}$ denotes Standard Deviation of i-th feature Xi for class Cj. Thus, we could rank features' t-statistics from top to bottom for listing the most crucial features and the least important features should be dropped on the basis of the requests.

b) PCA Model

Principle component analysis (PCA) is widely used for dimension reduction to avoid the curse of dimensionality problem and the redundancy of feature vectors. A series of linearly uncorrelated components variables (principle components) is computed from the training set to maximize the variance and each one should be orthogonal to each other. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables. Actually, PCA is one of the easiest eigen-vector based multi-variable analysis, which has been extended to visualize the high dimension dataset.

c) LASSO Model

LASSO is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. Lasso is able to achieve both of these goals by forcing the sum of the absolute value of the regression coefficients to be less than a fixed value, which forces certain

coefficients to be set to zero, effectively choosing a simpler model that does not include those coefficients. The expression can be written as:

$$min_{\beta_0,\beta}\left\{\frac{1}{N}\sum_{i=1}^{N}(y_i-\beta_0 1N-x_i^T\beta)^2\right\}\ subject\ to\ \sum_{j=1}^{p}|\beta_j|\le t$$

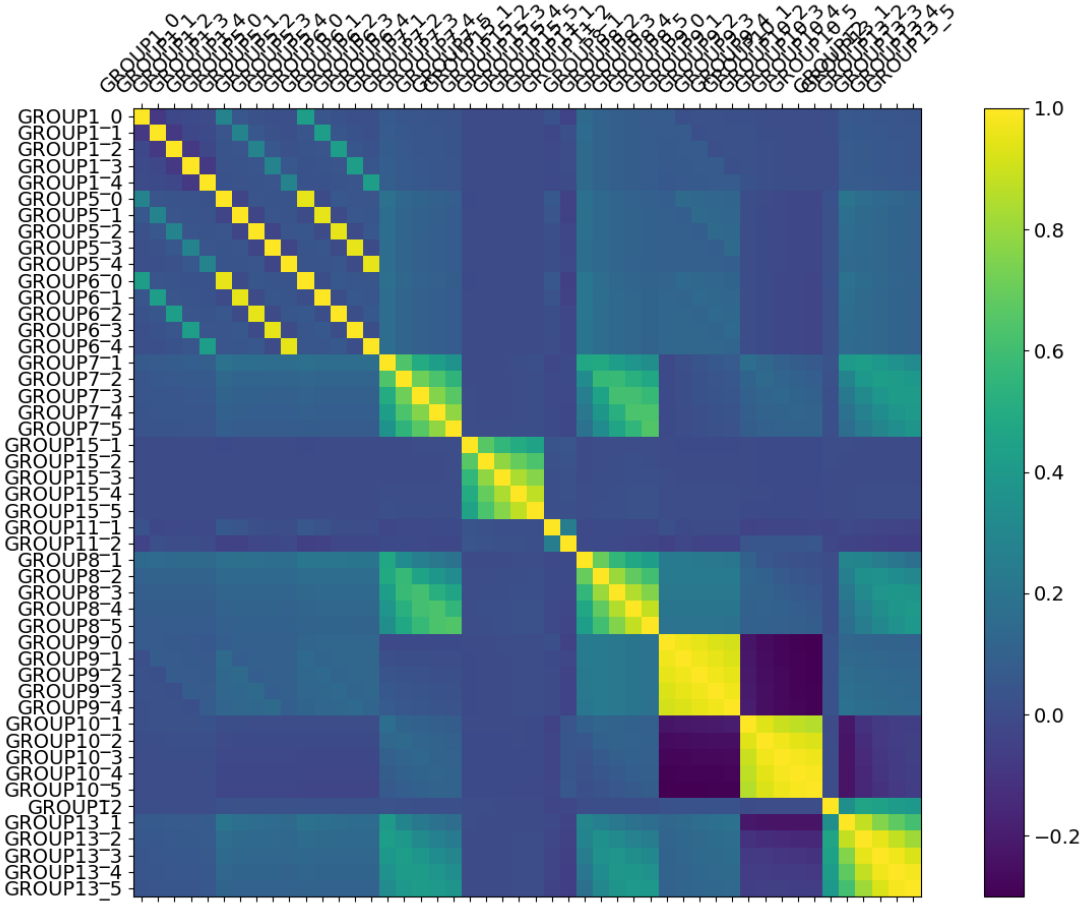
Where $\|\beta\|_p=(\sum_{i=1}^{N}\beta_i{}^p)^{1/p}$ is the standard l^p norm, and 1_N is an N*1 vector of ones.

2. Findings

a) T-value feature selection

Before performing the feature selection, we first created some new features from the combinations of original features. As it is too time-consuming to conduct comprehensive research of all the stocks, some representative stocks are chosen deliberately including ('0050', '1101', '1216', '1605', '2002', '2027', '2330').

Our first step is to compute the correlation among different features since it is needed to filter the negatively correlated features for increasing the effectiveness. The correlation plot of 0050 is attached as below:



From the correlation plot above, we can tell that the majority of inter-group features are positively correlated but exception appeared in group10 where the features are negatively correlated with each other.

After studied the relationship among features and filtered the negatively correlated feature pairs, we move to compute their corresponding t-statistics by which the importance of different features can be ranked. The top 20 features of different stocks are listed in the table below in decreasing order:

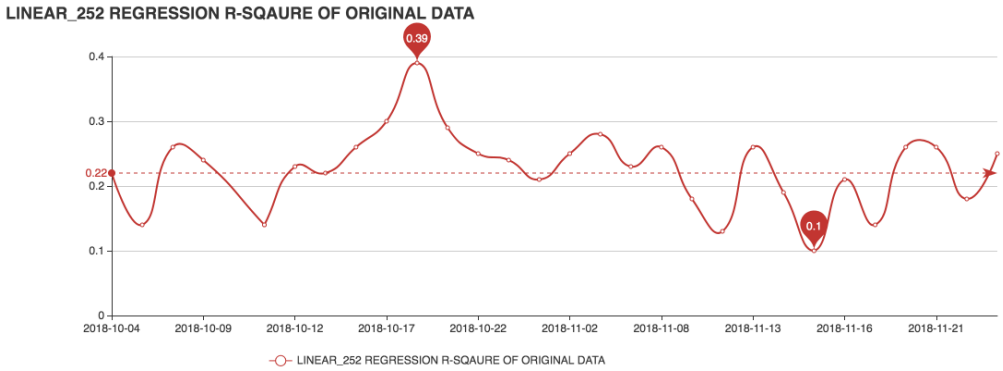
50	1101	1216	1605
GROUP8_1	GROUP9_0	GROUP9_0	GROUP10_1
GROUP9_0	GROUP1_0	GROUP10_1	GROUP9_0
GROUP12	GROUP10_1	GROUP1_0	GROUP1_0
GROUP10_1*GROUP12	GROUP1_1	GROUP6_0*GROUP13_1	GROUP1_1
GROUP8_2	GROUP1_2	GROUP1_1	GROUP8_1
GROUP1_1	GROUP1_4	GROUP6_0	GROUP1_2
GROUP1_0	GROUP1_3	GROUP8_1	GROUP1_3
GROUP10_1	GROUP8_1	GROUP6_1	GROUP1_4
GROUP1_2	GROUP6_0	GROUP6_4	GROUP6_0
GROUP1_3	GROUP13_3	GROUP1_2	GROUP8_2
GROUP6_4	GROUP6_1	GROUP11_2	GROUP6_1
GROUP8_3	GROUP11_2	GROUP1_4	GROUP15_1*GROUP10_1
GROUP1_4	GROUP6_4	GROUP6_3	GROUP15_1*GROUP13_1
GROUP8_5	GROUP8_3	GROUP1_0*GROUP13_1	GROUP11_2
GROUP6_3	GROUP13_5	GROUP5_0*GROUP7_1	GROUP8_3
GROUP6_2	GROUP6_3	GROUP6_2	GROUP6_2
GROUP6_1	GROUP6_2	GROUP1_3	GROUP13_5
GROUP11_2	GROUP12	GROUP15_1*GROUP13_1	GROUP11_1*GROUP13_1
GROUP8_4	GROUP15_1*GROUP13_1	GROUP11_1*GROUP8_1	GROUP6_3
GROUP13_4	GROUP6_0*GROUP8_1	GROUP7_1*GROUP8_1	GROUP6_4

2002	2027	2330
GROUP1_0	GROUP9_0	GROUP10_1
GROUP1_1	GROUP1_0	GROUP9_0
GROUP10_1	GROUP10_1	GROUP8_1
GROUP8_1	GROUP1_1	GROUP1_0
GROUP1_2	GROUP8_1	GROUP1_1
GROUP11_1*GROUP9_0	GROUP1_2	GROUP1_2
GROUP1_3	GROUP11_2	GROUP6_4
GROUP9_0	GROUP1_3	GROUP1_3
GROUP1_4	GROUP1_4	GROUP11_1*GROUP9_0
GROUP6_0	GROUP12	GROUP8_5
GROUP7_1*GROUP10_1	GROUP6_0	GROUP8_2

GROUP6_2	GROUP8_2	GROUP6_0*GROUP15_1
GROUP6_1	GROUP13_1	GROUP1_4
GROUP6_4	GROUP10_1*GROUP12	GROUP7_1*GROUP9_0
GROUP5_0*GROUP13_1	GROUP7_2	GROUP11_2
GROUP6_3	GROUP5_0*GROUP11_1	GROUP13_1
GROUP13_5	GROUP7_1*GROUP8_1	GROUP8_3
GROUP1_0*GROUP10_1	GROUP13_5	GROUP1_0*GROUP7_1
GROUP8_4	GROUP6_1	GROUP11_1*GROUP8_1
GROUP7_1*GROUP8_1	GROUP11_1*GROUP13_1	GROUP10_3

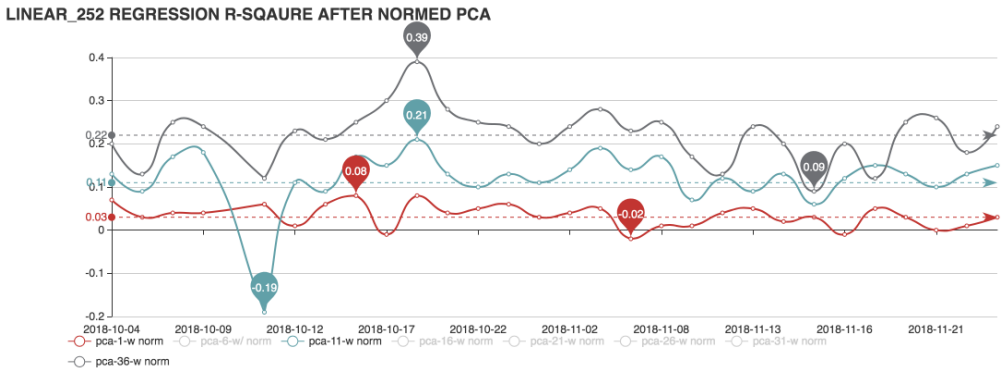
b) PCA

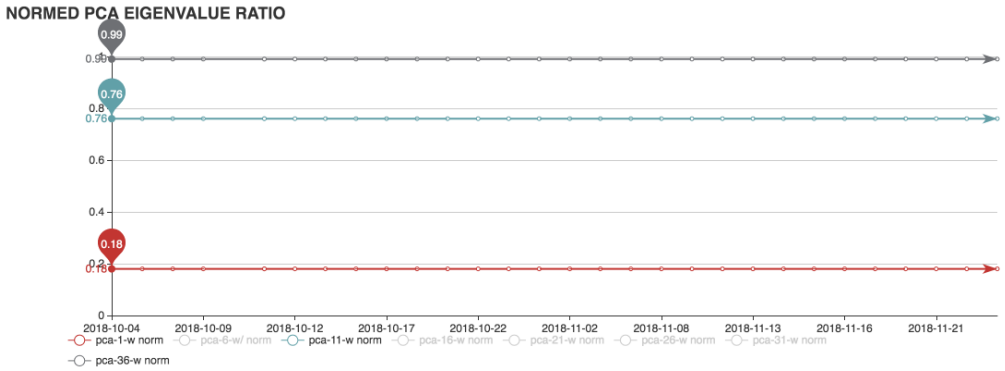
To further investigate the features' behaviour, principle component analysis was used to reduce the total number of original features and 0050 was chosen to be the target for investigation of PCA. Linear regression was first trained by original dataset with 45 features to predict the returns of next trading day, where training period is 170 days and testing period is 1 day. The r-square of each day is plotted as below:



The daily r-square plot shows that the average of r-square is 0.22 and the highest point is 0.39 on 2018-10-18. To compare the performance, PCA was then used to reduce the dimensionality to see whether we can boost the prediction performance by dropping the redundancy features.

After detailed investigation, we found out when the number of features was reduced to 36 by PCA, the linear regression can perform exactly same result of original dataset.





As shown in navy line, the average r-square equals to 0.22 when the number of features is 36 and its corresponding eigenvalue is 0.99 which means that nearly 99% information of original dataset can be explained by 36 features. With the decrease of the number of features, the eigenvalue consistently drops until there is only one feature which explained 18 percent of total variability.

Therefore, we could conclude another observation of PCA that the optimized number of 0050 features is 36 which could explain 99% of information and achieve comparable result with original dataset. However, we could not outperform the original dataset by reducing the dimensionality only and with less features reduced by PCA, the overall accuracy perform significantly worse. So PCA is useful to reduce the dimensionality instead of increasing the effectiveness according to our research.

c) Lasso Model

After the first period's research, we found that linear models generated from data in longer training period usually have better performances when it was used to predict stock returns in the next working day. Also, with more features included in models, the prediction accuracy will be higher. Based on these basic thoughts, the research on finding the most effective features is necessary. we choose the Lasso model to try to find some rules. From the lasso function in sklearn, Usually Lasso model spontaneously pick up about 22-27 features. From the table below, we could find that the R^2 didn't change much with linear regression with 48 factors, which means although at least 20 features are discarded when constructing the model, this has a little influence on the prediction accuracy rate. Also, features in group1,5,6,9,10,12 &13 are almost picked up every time.

Training -period	lasso_50	linear-50	lasso_2330	linear_2330
130	0.199	0.209	0.125	0.130
140	0.202	0.210	0.126	0.130

150	0.202	0.210	0.130	0.131
160	0.199	0.211	0.129	0.131
170	0.199	0.212	0.129	0.131
180	0.197	0.213	0.126	0.131
190	0.190	0.213	0.117	0.131

These tables show the frequency of these factors chosen as effective factors in 511 loop. We can find that features in group1,5,6,9,10,12 &13 are almost picked up every time, while some groups such as 15, 11 have never been chosen .

0050(Y_M_1):

GROUP1_0	100%	GROUP6_0	100%	GROUP15_1	0%	GROUP8_4	0%	GROUP10_4	100%
GROUP1_1	100%	GROUP6_1	100%	GROUP15_2	0%	GROUP8_5	1%	GROUP10_5	100%
GROUP1_2	100%	GROUP6_2	100%	GROUP15_3	0%	GROUP9_0	2%	GROUP12	100%
GROUP1_3	100%	GROUP6_3	100%	GROUP15_4	0%	GROUP9_1	24%	GROUP13_1	100%
GROUP1_4	100%	GROUP6_4	90%	GROUP15_5	0%	GROUP9_2	32%	GROUP13_2	100%
GROUP5_0	100%	GROUP7_1	35%	GROUP11_1	0%	GROUP9_3	50%	GROUP13_3	100%
GROUP5_1	100%	GROUP7_2	8%	GROUP11_2	0%	GROUP9_4	97%	GROUP13_4	100%
GROUP5_2	100%	GROUP7_3	0%	GROUP8_1	0%	GROUP10_1	100%	GROUP13_5	100%
GROUP5_3	100%	GROUP7_4	0%	GROUP8_2	0%	GROUP10_2	100%		
GROUP5_4	100%	GROUP7_5	0%	GROUP8_3	0%	GROUP10_3	100%		

2330(Y_M_1):

GROUP1_0	100%	GROUP6_0	100%	GROUP15_1	0%	GROUP8_4	0%	GROUP10_4	100%
GROUP1_1	100%	GROUP6_1	100%	GROUP15_2	0%	GROUP8_5	0%	GROUP10_5	100%
GROUP1_2	100%	GROUP6_2	100%	GROUP15_3	0%	GROUP9_0	0%	GROUP12	100%
GROUP1_3	100%	GROUP6_3	100%	GROUP15_4	0%	GROUP9_1	0%	GROUP13_1	100%
GROUP1_4	100%	GROUP6_4	82%	GROUP15_5	0%	GROUP9_2	0%	GROUP13_2	100%
GROUP5_0	100%	GROUP7_1	1%	GROUP11_1	0%	GROUP9_3	0%	GROUP13_3	100%
GROUP5_1	100%	GROUP7_2	0%	GROUP11_2	0%	GROUP9_4	0%	GROUP13_4	100%
GROUP5_2	100%	GROUP7_3	0%	GROUP8_1	0%	GROUP10_1	0%	GROUP13_5	100%
GROUP5_3	100%	GROUP7_4	0%	GROUP8_2	0%	GROUP10_2	17%		
GROUP5_4	100%	GROUP7_5	0%	GROUP8_3	0%	GROUP10_3	89%		

The result inspired us to reduce the parameter range further. I summarized the best 5 and 10 parameters chosen by Lasso model and use these parameters to build a linear regression model to back-test the prediction accuracy. Compared with R^2 of full parameters linear regression model, those of 5 parameters and 10 parameters linear model nearly becomes a half.

But the R squares of 5 and 10 parameters have little difference. For specific parameters, they don't change with different training periods for 0050 and 233 and are all chosen from group1, 5,7,9 & 10. But parameters for 0050 and 2330 are different.

training period	linear-50_R^2	0050_5 features_R^2	0050_10 features_R^2	linear_2330_R^2	2330_5 features_R^2	2330_10 features_R^2
160	0.209	0.127	0.136	0.130	0.077	0.102
165	0.210	0.123	0.132	0.130	0.075	0.101
170	0.210	0.134	0.143	0.131	0.073	0.101
175	0.211	0.133	0.141	0.131	0.067	0.098
180	0.212	0.134	0.143	0.131	0.068	0.094
185	0.213	0.133	0.140	0.131	0.071	0.094
190	0.213	0.129	0.141	0.131	0.066	0.087

0050_5 features	GROUP5_0	GROUP7_1	GROUP9_0	GROUP9_4	GROUP10_1
0050_10 features	GROUP1_0	GROUP1_1	GROUP1_2	GROUP5_0	GROUP7_1
	GROUP7_2	GROUP9_0	GROUP9_3	GROUP9_4	GROUP10_1
2330_5 features	GROUP1_0	GROUP1_1	GROUP9_0	GROUP9_4	GROUP10_1
2330_10 features	GROUP7_5	GROUP9_0	GROUP9_4	GROUP10_1	GROUP10_5
	GROUP7_5	GROUP9_0	GROUP9_4	GROUP10_1	GROUP10_5

Then we could conclude that lasso could help reduce redundant features but performance won't be improved with less features. The effective feature are usually in group 1, 5,7,9,10.

d) Feature Creation

Try 1: two of five most significant features multiply randomly (based on lasso model)

After the 5 best parameters are chosen, I wonder that whether better parameters can be created if two of five factors multiply randomly to form new parameters. I add these new parameters in the independent variables and still use Lasso model to pick up 5 best variables and back-test stock return prediction accuracy with these 5 parameters by linear model.

	0050_5_R^2	2330_5_R^2
160	-0.013	0.033
165	-0.018	0.025
170	-0.020	0.014
175	-0.014	0.017

180	0.004	0.021
185	0.016	0.029
190	0.013	0.024

As shown in the table, new features not only do not improve prediction accuracy, also negatively influence the predictive ability of the original factors. Hence, this method fails.

Try2: two of the 11 factors multiply randomly(based on p-value)

Observing the top 10 significant features selected by lasso model, we find most of the features are the first one in their group. Hence, I decided to choose the first feature of 11 groups, make two of them multiply randomly to create new features.

Also considering low calculation speed of lasso model, we replace it by a new model. I directly choose the ten features with smallest p-values from linear regression model , then use new features to build a linear regression model and back-test the prediction accuracy.

Firstly, we calculated the top 10 features with smallest p-values from linear regression model as control group . As the below shown, chosen features belong to group 1, 8, 9&10. And the back-test R^2 are bout 0.16 and 0.09 for 0050 and 2330 respectively.

	0050_10 features	2330_10 features
160	0.150	0.085
170	0.153	0.078
180	0.162	0.092
190	0.163	0.096

	50	2330
p-values increasing	GROUP1_0	GROUP1_0
	GROUP9_0	GROUP8_1
	GROUP8_4	GROUP8_2
	GROUP8_3	GROUP8_3
	GROUP8_2	GROUP8_4
	GROUP8_1	GROUP9_0
	GROUP8_0	GROUP9_1
	GROUP9_3	GROUP9_2
	GROUP9_4	GROUP9_3
	GROUP10_0	GROUP9_4

Then, we choose the top 10 new features with smallest p-values from a linear regression and use them to back-test return prediction accuracy.

	0050	2330
160	-0.210	-43.590
170	-0.232	-36.948
180	-0.228	-24.369
190	-0.256	-42.442

As shown in the table, the result is quite bad. Hence, this method also fails. we also try to add these features in the original variables and they are almost never chosen as top 10 significant variables.

Try 3: Three of the 11 factors multiply randomly(based on p-value)

The only difference between this try and the last is that we choose three of eleven factors (the first parameter in every group). The back-test R squares of new models show that new features do not improve the prediction performance. Besides, we also try to add these features in the original variables and they are almost never chosen as top 10 significant variables.

	0050	2330
160	0.023	0.031
170	0.027	0.027
180	0.028	0.034
190	0.027	0.037

Try4: the features add product of itself and the other feature as new features(based on p-value)

Although the result is better than before ones, new features still underperform the original features. The reason why the result in this construction way is probably the format $gorup\{i\} + group\{i) * group\{j\}$, which means the original features are included. Hence, we still did not find effective new features, which is a pity.

	0050	2330
160	0.122	0.071
170	0.125	0.066
180	0.135	0.078
190	0.132	0.080

3. Conclusion

Generally, different stocks response differently in terms of features. Reducing redundant parameters by PCA and Lasso could only maintain the similar prediction performance but will not optimize it. Group 1,8,9 & 10 are slightly more significant than other features. Also, feature combination (multiplication)cannot boost the performance.

Part 3 Model Prediction

1. Methodology Introduction

It is a common sense that the prediction of models are better when we use longer training period and closer prediction date. Our initial motivation for this part was to know, for different models, the optimal length of training period and for how long the model remains useful. However, the result shows it is not exactly the case.

To determine the optimal length of training period and life span for different models, heatmaps of average out-of-sample R square are plotted for stock 0050 and 2330, for the 5 labels, under different models, with all the features. The rows of the heatmaps corresponds to the prediction date, with y-labels as the index of that prediction date. When the index is 0, it means we use the past data to predict the immediate next trading day. The maximum is 20 trading days (4 weeks). The columns correspond to the length of training period. I have chosen past trading days of length 125, 130, 135, 140, 165, 170, 175, 180. I haven't got time to calculate the results for 145, 150, 155, 160, since it takes very long to construct the models. This part is calculated by module Scikit_Example_MProcesses_ModelComparison.py, saved in 2 folders, and plotted by module plot_R2.py.

The models include GBRT, Linear Regression (with intercept, features normalized), Partial Least Square Regression (8 components, since absolute t-values decrease fast around the top 5-10 features), LassoCV (alpha_min/alpha_max=0.01, number of alphas=100, tolerance=1e-6, cv=5-fold), and neural net work.

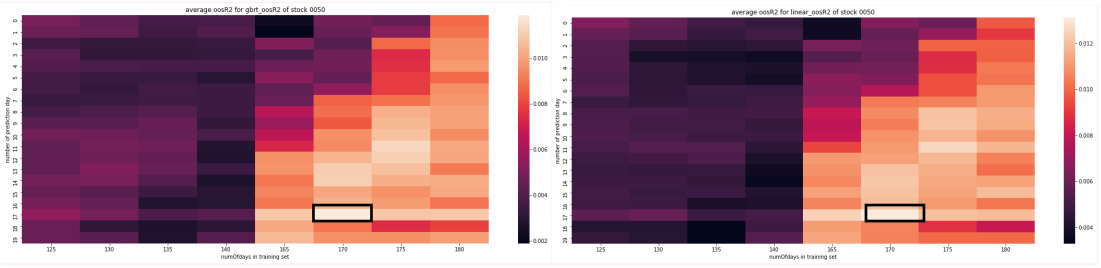
2. Findings

a) Average oosR2 for different models

The results for stock 0050 and 2330 are quite different. The following oosR2 are averaged on label Y_M_1,2,3,4,5. They correspond to model GBRT, Linear Regression, Partial Least Square Regression and Lasso, respectively.

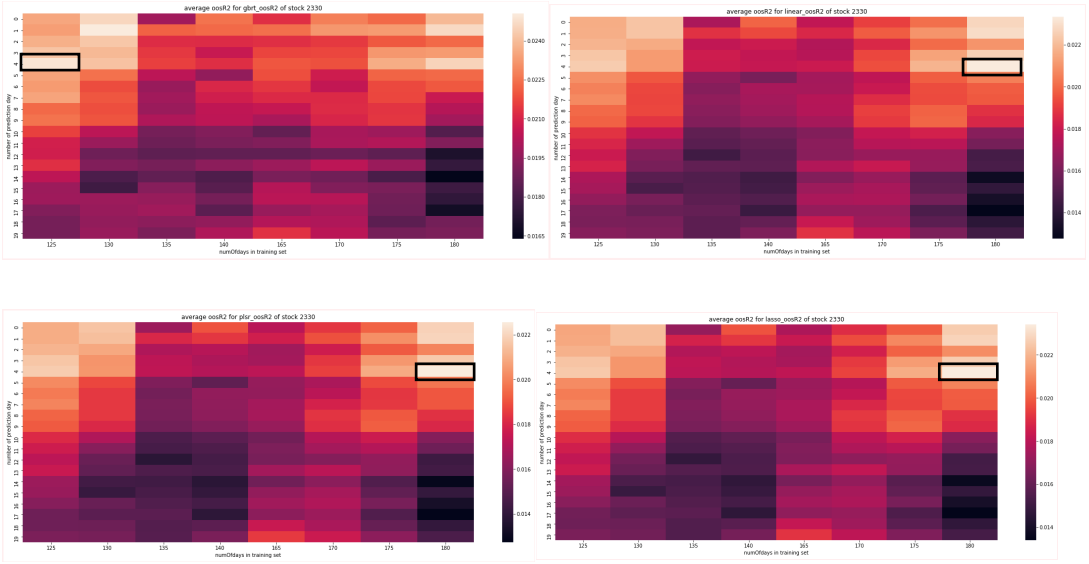
Stock 0050:

The highest oosR2 is obtained when we train our model on the past 170 days to predict the result of the 18th trading day in future.

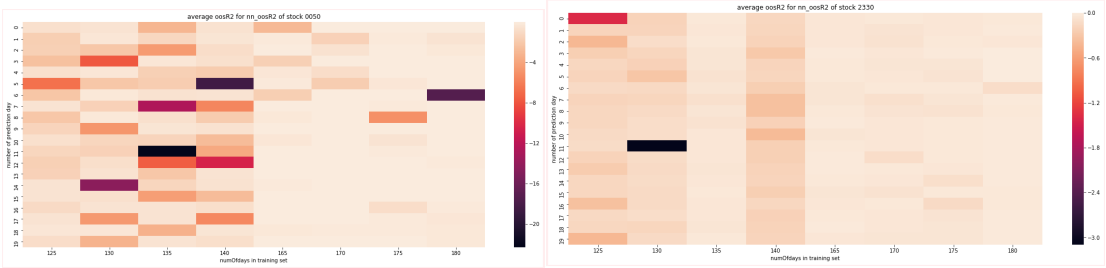


Stock 2330:

The result is quite different from that of stock 0050. When we increase the length of training period, the oosR2 decreases first and then goes back up. The highest oosR2 is obtained when we train our model on the past 180 days to predict the 5th trading day in future.



For both stocks 0050 and 2330, model of neural net work show a low average oosR2 (ranges from -20 to 0 for 0050, ranges from -3 to 0 for 2330), which explains little.

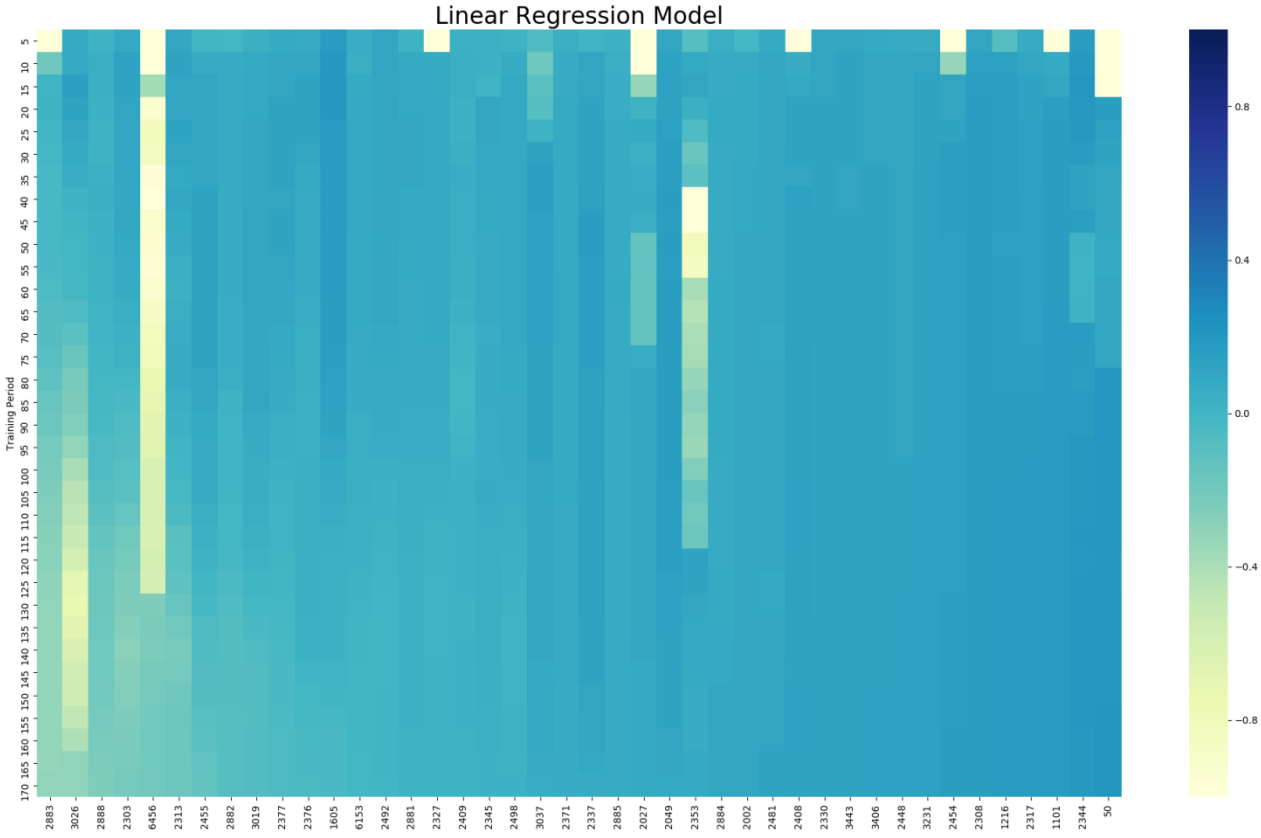


Thus, the result is not the same as what we have assumed. In general, stock 0050 and 2330 have different optimal length of training period and period of model validity.

b) Comparison between Linear and Ridge regression

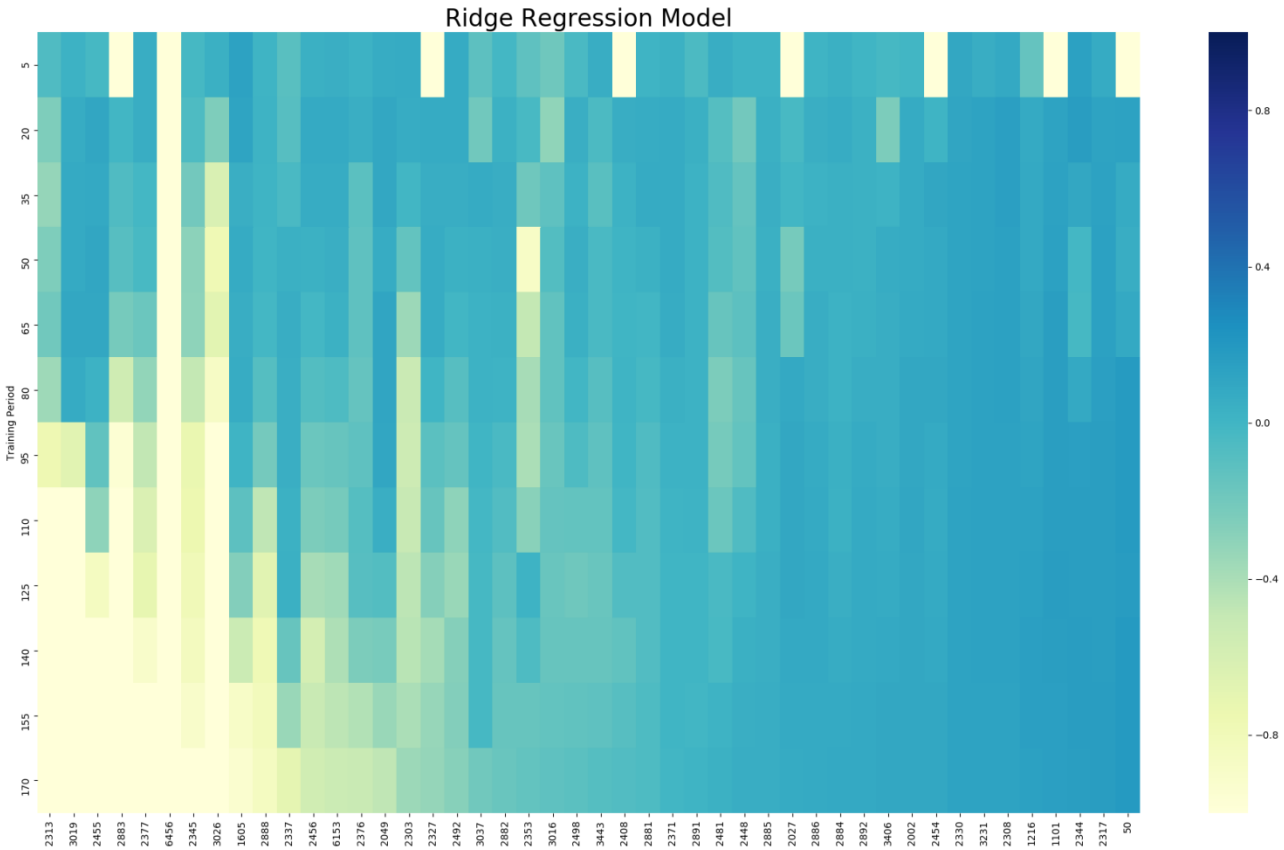
In this section, a comparison among stocks by linear and ridge regression is conducted to investigate the distinguished effectiveness and sensitivity of different stocks.

To compare the effectiveness of the model, we choose two independent variables – stocks and training period to be our research target since it is the purpose to find out how the r-square changes along with training duration, stocks and models.



The above figure demonstrates the average out-of-sample r-square produced from linear regression model trained by different stocks and training duration. The best performance achieved by stock 0050 which have the highest average r-square. Furthermore, the longer the training duration the more stable of the out-of-sample r-square can be produced, which shows that for Taiwan market, the longer training duration (around 170) is a good choice for majority of securities if using linear model as the prediction source.

Besides linear model, we also conducted the experiment to ridge model which is shown as follows:



The behaviour of ridge model in Taiwan market has a lots similarity with linear model but also some significant distinctions. A large portion of equities still tend to outperform if a longer training period is chosen, which is shown by the clustering effect at right-bottom part. Another similar pattern as the linear model is that some equities like 0050 will significantly underfit if the model is refined by 5 days' training samples.

The distinction is quite obvious that ridge model becomes overfitting for some equities listed at left-bottom part and the predicting power significant drops. One of the reason might be that ridge model was overreacting to penalize the parameter while the training samples are massively plentiful, which induces to some crucial parameters are penalized to trivial solution during training phase.

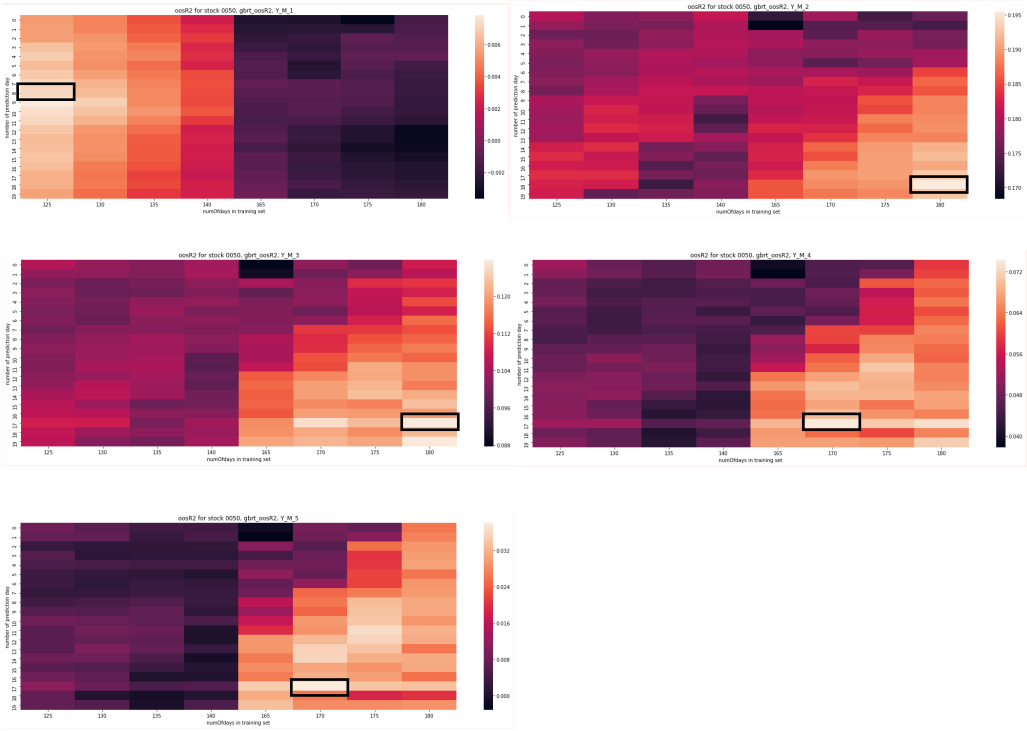
In all, there is no observable trend indicating what equities should use what kind of model or training period to predict r-square but generally, linear regression model is better than ridge model and a longer training duration will stabilize the predicting model in most of the cases at Taiwan equity market.

c) Model comparation

i. GBRT

Stock 0050:

The labels (YM1 - YM5) show different patterns. Label Y_M_1 has much smaller oosR2 than other labels.

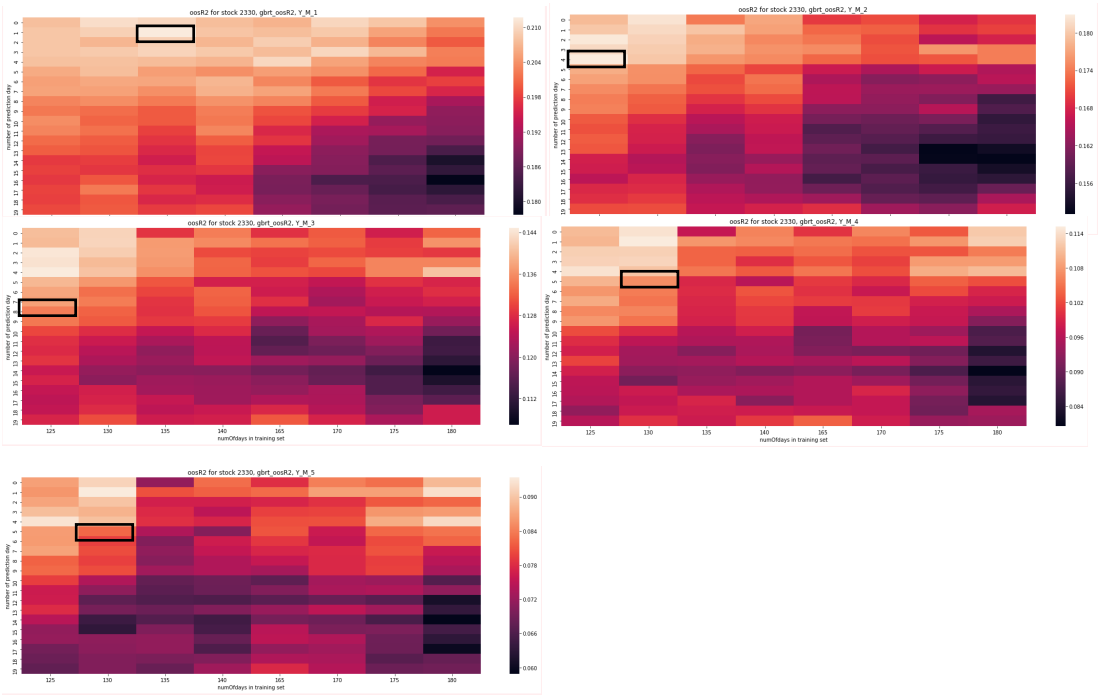


Label	Smallest oosR2	Largest oosR2
Y_M_1	-0.002	0.008
Y_M_2	0.170	0.195
Y_M_3	0.088	0.128
Y_M_4	0.040	0.072
Y_M_5	0.000	0.032

Among the 5 labels, Y_M_2 has the best results under GBRT model for stock 0050. Then, the oosR2 starts to decay, as the forecasting minute grows.

Stock 2330:

For 2330, label Y_M_1 has the highest oosR2, then it starts to decay.



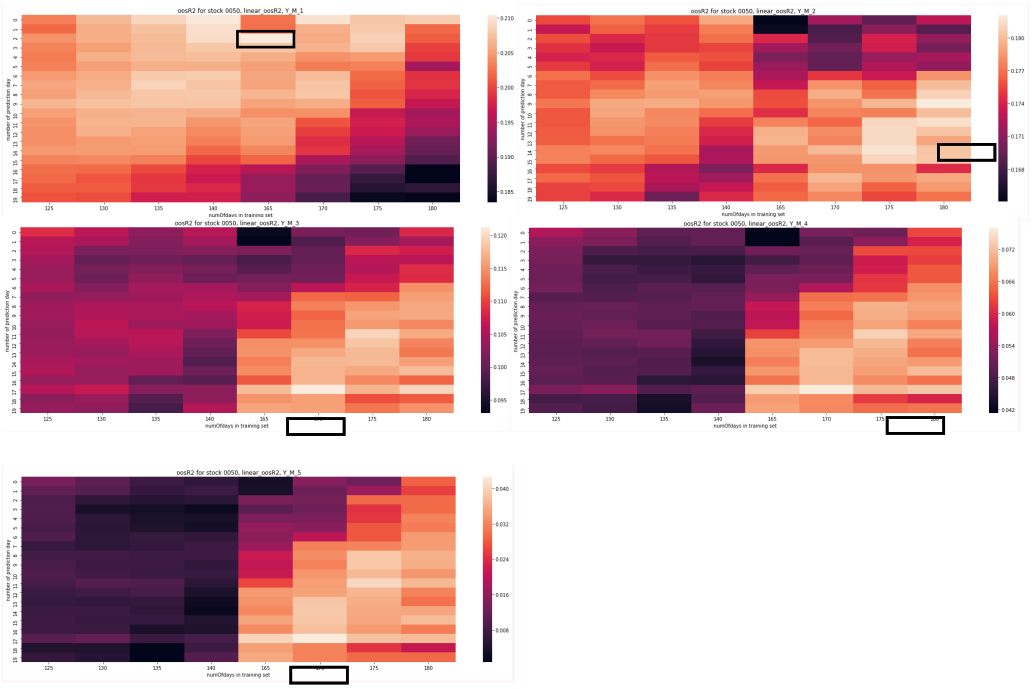
Label	Smallest oosR2	Largest oosR2
Y_M_1	0.180	0.210
Y_M_2	0.156	0.180
Y_M_3	0.112	0.144
Y_M_4	0.084	0.114
Y_M_5	0.060	0.090

Thus, GBRT works well for Y_M_2 of stock 0050 and Y_M_1 of stock 2330, according to the data we have, surprisingly.

ii. Linear Regression

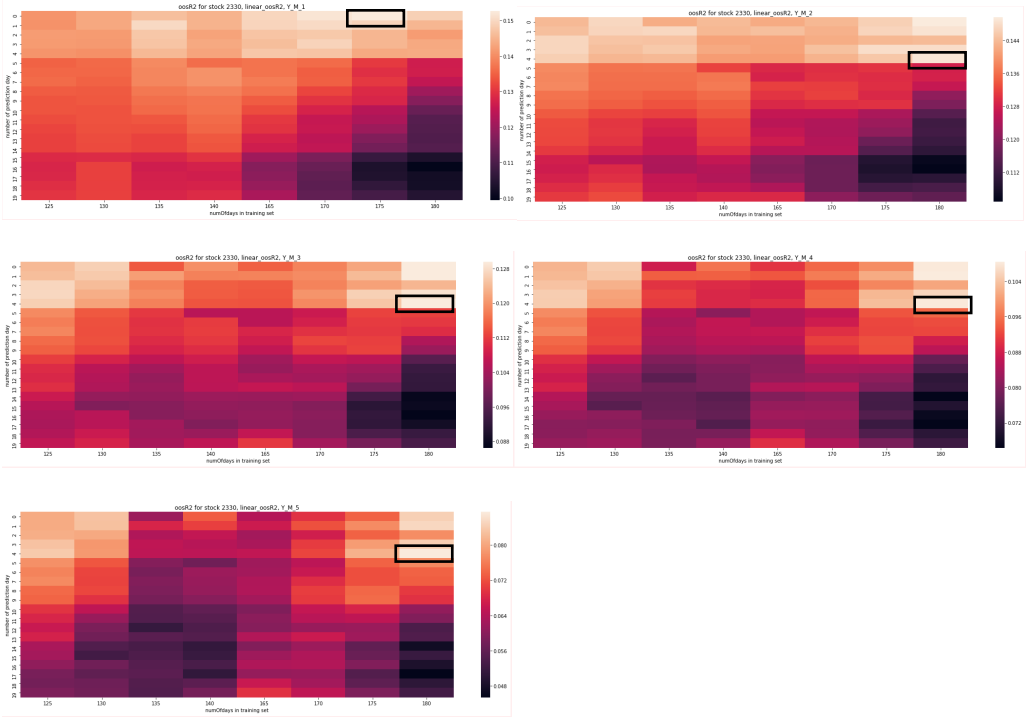
For stock 0050, label Y_M_1 has the highest oosR2. When we go on to use Y_M_2,3,4,5, oosR2 starts to decay and the heatmap pattern quickly becomes less regular. For stock 2330, however, oosR2 for Y_M_1 and Y_M_2 are both the highest. It starts to decay at Y_M_3. The patterns for stock 2330 remains unchanged.

Stock 0050:



Label	Smallest oosR2	Largest oosR2
Y_M_1	0.185	0.210
Y_M_2	0.168	0.180
Y_M_3	0.095	0.120
Y_M_4	0.042	0.072
Y_M_5	0.008	0.040

Stock 2330

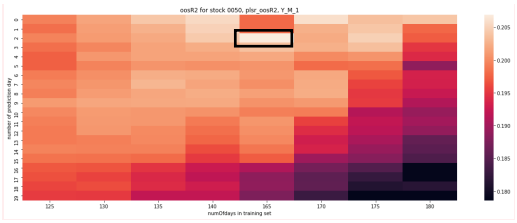


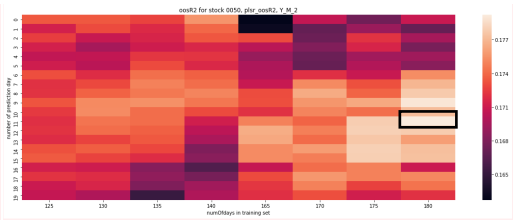
Label	Smallest oosR2	Largest oosR2
Y_M_1	0.10	0.15
Y_M_2	0.112	0.144
Y_M_3	0.088	0.128
Y_M_4	0.072	0.104
Y_M_5	0.048	0.080

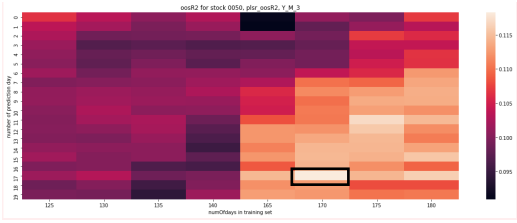
iii. Partial Least square Regression

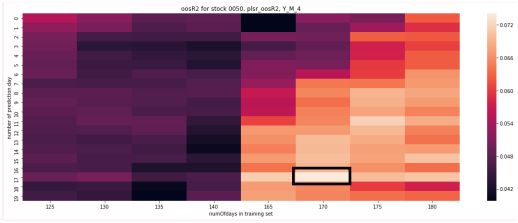
As for partial least square regression model, the oosR2 is similar to that of Linear Regression, for both stock 0050 and 2330. For 0050, The heatmap patterns also start to become irregular at Y_M_2. For stock 2330, oosR2 are also the highest at Y_M_1 and Y_M_2 and starts to decay at Y_M_3. What makes a difference is the location of maximum of oosR2. Training PLSR on past 165 trading days and using that to predict the 3rd trading days gives us the maximum, for Y_M_1. As to Y_M_2,3,4,5, the prediction day is too far and thus are unreliable.

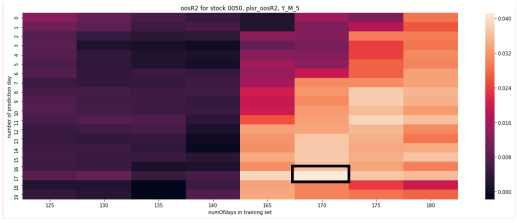
Stock 0050





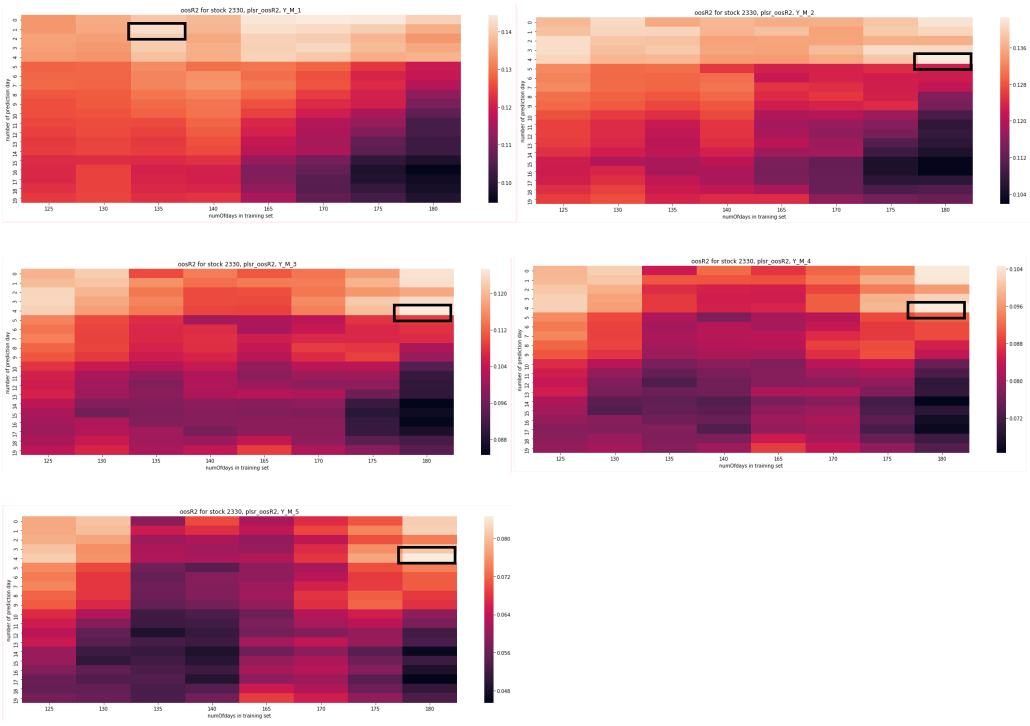






Label	Smallest oosR2	Largest oosR2
Y_M_1	0.180	0.205
Y_M_2	0.165	0.177
Y_M_3	0.095	0.115
Y_M_4	0.042	0.072
Y_M_5	0.000	0.040

Stock 2330

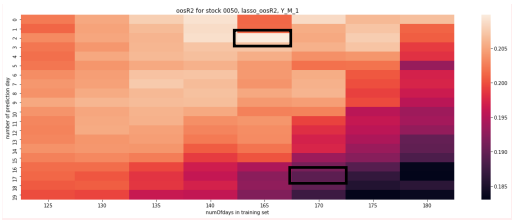


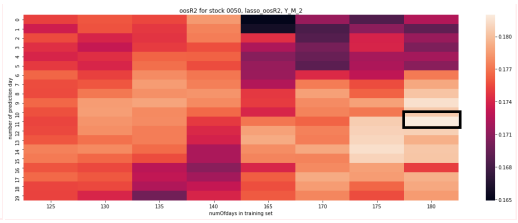
Label	Smallest oosR2	Largest oosR2
Y_M_1	0.10	0.14
Y_M_2	0.104	0.136
Y_M_3	0.088	0.120
Y_M_4	0.072	0.104
Y_M_5	0.048	0.080

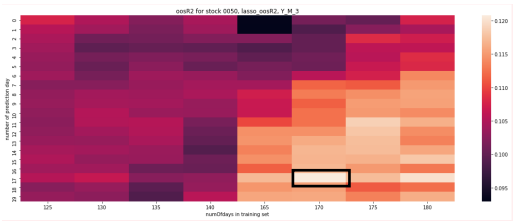
iv. Lasso

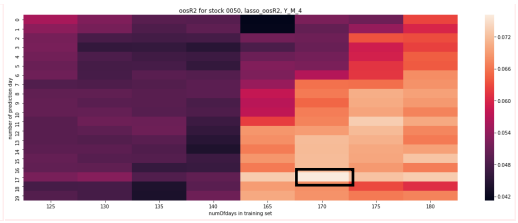
The results for Lasso are also similar to those of Linear Regression and Partial Least Square Regression. The value of oosR2 are close and the heatmap patterns are also similar.

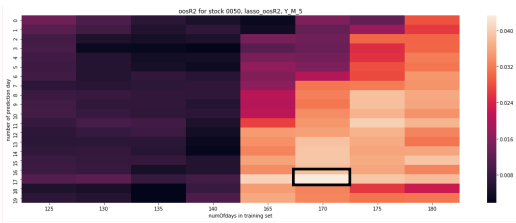
Stock 0050





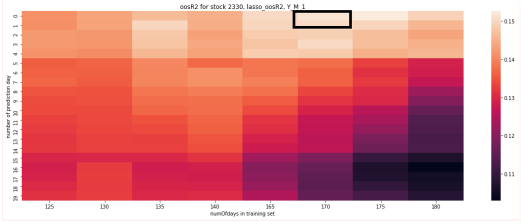


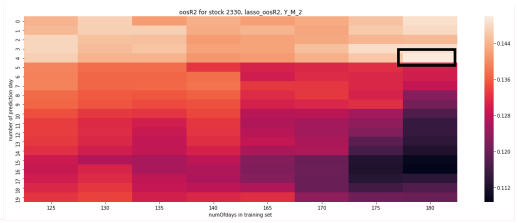


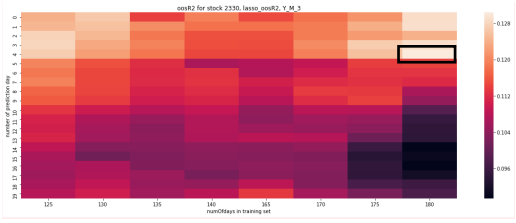


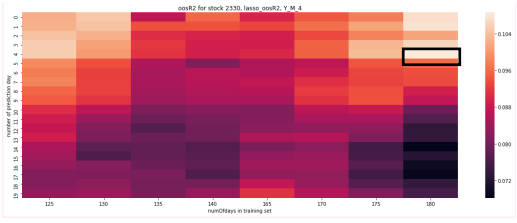
Label	Smallest oosR2	Largest oosR2
Y_M_1	0.185	0.210
Y_M_2	0.165	0.182
Y_M_3	0.094	0.120
Y_M_4	0.042	0.074
Y_M_5	0.008	0.040

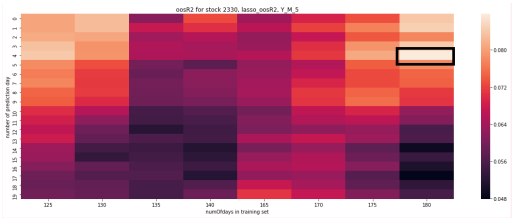
Stock 2330











Label	Smallest oosR2	Largest oosR2
Y_M_1	0.11	0.15
Y_M_2	0.112	0.144
Y_M_3	0.096	0.128
Y_M_4	0.072	0.104
Y_M_5	0.048	0.086

3. Conclusion

The results for Linear Regression, Partial Least Square Regression and Lasso are very similar. This is probably because these models are similar in nature. For these 3 models, the best label is Y_M_1 for 0050 and the best labels are Y_M_1 and Y_M_2 for 2330. As for GBRT model, the best label is Y_M_2 for stock 0050 and the best label is Y_M_1 for 2330.

In this back-end study, except for the length of training period, we also take the length of testing period and labels into consideration. For stock 0050, the performance of GBRT model for Y_M_2 is similar as the performance of Linear model family for Y_M_1. However, for stock 2330, GBRT model for Y_M_1 outperforms the Linear model family for Y_M_1 and Y_M_2 by about 5% in terms of oosR2.

Part 4 Conclusion

On the basis of our research, Y_M_1 with the shortest testing duration has the best performance among the majority of tests (except for stock 0050 under GBRT model), so we mainly conducted experiments by using Y_M_1. For front-end, t-statistics, PCA and Lasso can be useful to reduce the dimensionality without much influence on accuracy but are not able to boost the performance through dropping the redundant features. Features in group 1,8,9,10 are slightly more significant than others. Additional experiments are conducted in the back-end side by comparing the performance through different stocks, models and training duration. For stocks, models trained by 0050 generally could produce more accurate result and the longer the training duration (around 170 days) is capable to increase the averaged r-square of 0050 to around 20%. If we only predict 1 day in advance based on previous 170 training days, Linear Regression models (Linear, PLSR, Lasso, Ridge) outperform other much models we used (GBRT, neural network). If we predict more, say, 5 days in advance, considering different length of training set, GBRT surprisingly outperforms Linear Regression models.