Final

Empirical Asset Pricing via Machine Learning

Shen Yiqin, Liu Jianyi, Sha Haochen

Abstract

In this paper, we try to replicate part of works by the paper "Empirical Asset Pricing via Machine Learning". We use OLS, OLS-3, PCR, PLS, GBR, Random Forest and Neural Networks(1-5) totally 11 models. The task is to predict the stock returns through recursive evaluation. First we do data pre-processing and feature engineering as the paper. Then we tune the parameter and use R^2_{OOS} to evaluate the performance of each model. For input features, we also discuss their importance among all models. Our conclusion is partly similar to the original paper.

Workload: Sha Haochen did the RF, GBR model and the part3 Evaluation part4 Conclusion of the report. Liu Jianyi did the neural network 1-5 and wrote the second part of the report. Shen Yiqin did the data cleaning and preprocessing, OSL, OLS-3, PCR, PLS coding and model tuning, wrote data description, data preprocessing and abstract in report.

1 Data

1.1 Data Description

The paper[1] investigates dataset about more than 30,000 stocks listed on NYSE, AMEX and NASDAQ, timestamp from 1957 to 2016, 60 years of financial data in total. The dataset contains 94 firm characteristics predictors and 7 information describing features.
Some firm characteristics predictors:

- "acc"–Working capital accruals
- "bm"-Book-to-market
- "chmom"-Change in 6-month momentum
- "divo"-Dividend omission
- "ear"-Earnings announcement return

Information describing features:

- "permno"–A number representing a specific stock
- "DATE"-Date
- "RET"-Return
- "prc"-Price
- "SHROUT"-Shares Outstanding
- "mve0"-Total market value at the start of the period
- "sic2"-Standard Industrial Classification(SIC) codes

Besides, there are 8 economical features which are also considered in modeling. Idea is from paper "A Comprehensive Look at the Empirical Performance of Equity Premium Prediction" by Amit Goyal & Ivo Welch. Predictors are list at following:

- "dp"–Dividend price ratio
- "ep"-Earning price ratio
- "bm"-Book to market ratio
- "ntis"-Net equity expansion
- "tbl"-Treasury-bill rate
- "tms"-Term spread
- "dfy"-Default spread
- "svar"-Stock variance

1.2 Data Preprocessing

We first select the data in time range from 1975 to 2016, totally 60 years. Then split the data into train/validation/test sets. Some features of the data have large amount of missing values. To dealing with missing values, we firstly group by the data with "permno", which is the number representing the specific stock, and filling missing values with mean for each feature. And for the result of missing values, we just fill them with zeros. After handling the missing values, we do the normalization by fitting a normalizing scalar on the training set, then transform the validation set and test set.
Models are not only be trained on the original dataset, but also on the top-1000 and the bottom-1000 dataset. Top/bottom-1000 datasets are calculated by selecting stocks through their price. We firstly group by the "DATE" and sort stocks by the "prc", so in each group, we just filter the top 1000 stocks or bottom 1000 stocks to get our top/bottom-1000 dataset and do the above preprocessing procedures again. After that, we do the modeling.

2 Model

2.1 Huber Loss and Hyperparameters Tuning

Huber loss is a loss function that is a mixture of mean absolute error (MAE) and mean squared error(MSE). It is defined as

$$\mathcal{L}_H(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} H(r_{i,t+1} - g(z_{i,t};\theta),\xi))$$

Hyperparameters means "parameters needs to be tuned" in laymen speaking, they are the parameters set before training the model. Data scientists need to find the parameter that will have best score in backtesting with the validation set, thus yielding for best prediction. Two major ways to implement this hyperparameters turning, is by grid search and random search. We manual tune the hyperparameters with the aid of grid search in our analysis.

2.2 Linear

2.2.1 OLS and OLS-3

Ordinary least squares[2] (OLS) is a type of linear least squares method for estimating the unknown parameters in a linear regression model. OLS chooses the parameters of a linear function of a set of explanatory variables by the principle of least squares: minimizing the sum of the squares of the differences between the observed dependent variable (values of the variable being observed) in the given dataset and those predicted by the linear function of the independent variable.

Geometrically, this is seen as the sum of the squared distances, parallel to the axis of the dependent variable, between each data point in the set and the corresponding point on the regression surface—the

smaller the differences, the better the model fits the data. The resulting estimator can be expressed by a simple formula, especially in the case of a simple linear regression, in which there is a single regressor on the right side of the regression equation.

The OLS estimator is consistent when the regressors are exogenous, and—by the Gauss–Markov theorem—optimal in the class of linear unbiased estimators when the errors are homoscedastic and serially uncorrelated. Under these conditions, the method of OLS provides minimum-variance mean-unbiased estimation when the errors have finite variances. Under the additional assumption that the errors are normally distributed, OLS is the maximum likelihood estimator. OLS with all covariates,OLS-3 preselects size,book-to-market,and momentum as the only covariates.

2.3 Dimension Reduction

2.3.1 PCR

Principal component regression (PCR) is a regression analysis technique that is based on principal component analysis (PCA). More specifically, PCR is used for estimating the unknown regression coefficients in a standard linear regression model.

In PCR, instead of regressing the dependent variable on the explanatory variables directly, the principal components of the explanatory variables are used as regressors. One typically uses only a subset of all the principal components for regression, making PCR a kind of regularized procedure and also a type of shrinkage estimator.

Often the principal components with higher variances (the ones based on eigenvectors corresponding to the higher eigenvalues of the sample variance-covariance matrix of the explanatory variables) are selected as regressors. However, for the purpose of predicting the outcome, the principal components with low variances may also be important, in some cases even more important.

One major use of PCR lies in overcoming the multicollinearity problem which arises when two or more of the explanatory variables are close to being collinear.PCR can aptly deal with such situations by excluding some of the low-variance principal components in the regression step. In addition, by usually regressing on only a subset of all the principal components, PCR can result in dimension reduction through substantially lowering the effective number of parameters characterizing the underlying model. This can be particularly useful in settings with high-dimensional covariates. Also, through appropriate selection of the principal components to be used for regression, PCR can lead to efficient prediction of the outcome based on the assumed model. The parameter change of PCR is as follow:

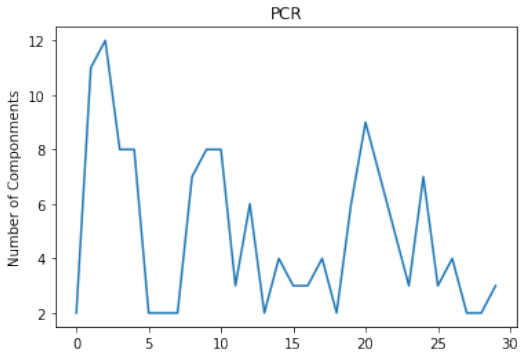

Figure 1: Parameter change of PCR

2.3.2 PLS

Partial least squares regression[3] (PLS regression) is a statistical method that bears some relation to principal components regression; instead of finding hyperplanes of maximum variance between the response and independent variables, it finds a linear regression model by projecting the predicted variables and the observable variables to a new space. Because both the X and Y data are projected to

new spaces, the PLS family of methods are known as bilinear factor models. Partial least squares discriminant analysis (PLS-DA) is a variant used when the Y is categorical.

PLS is used to find the fundamental relations between two matrices (X and Y), i.e. a latent variable approach to modeling the covariance structures in these two spaces. A PLS model will try to find the multidimensional direction in the X space that explains the maximum multidimensional variance direction in the Y space. PLS regression is particularly suited when the matrix of predictors has more variables than observations, and when there is multicollinearity among X values. By contrast, standard regression will fail in these cases (unless it is regularized). The parameter change of PLS is as follow:

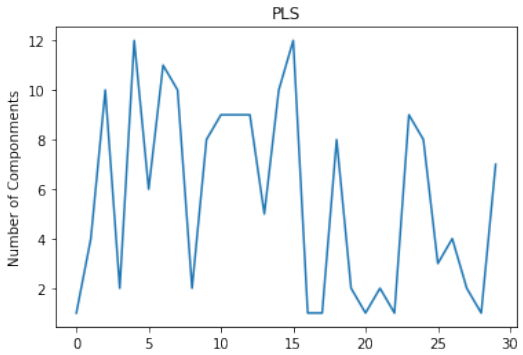

Figure 2: Parameter change of PLS

2.4 Ensemble

2.4.1 GBR

Gradient boosting Regression[4] calculates the difference between the current prediction and the known correct target value.This difference is called residual. After that Gradient boosting Regression trains a weak model that maps features to that residual. This residual predicted by a weak model is added to the existing model input and thus this process nudges the model towards the correct target. Repeating this step again and again improves the overall model prediction.

Also it should be noted that Gradient boosting regression is used to predict continuous values like house price , while Gradient Boosting Classification is used for predicting classes like whether a patient has a particular disease or not.

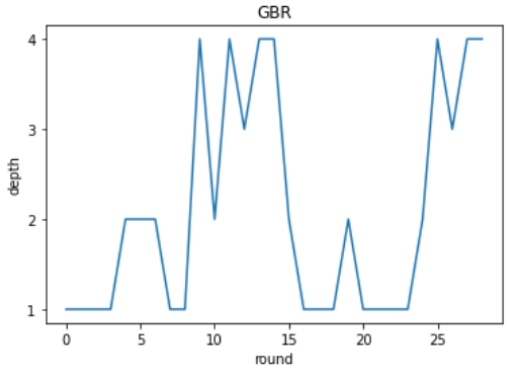

Figure 3: Parameter change of GBR

2.4.2 Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

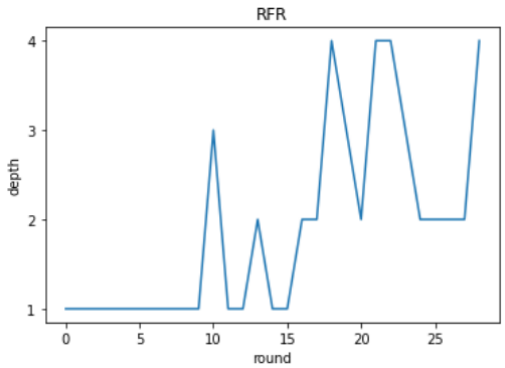

Figure 4: Parameter change of RF

2.5 Neural Network

A neural network is a network or circuit of biological neurons, or, in a modern sense, an artificial neural network, composed of artificial neurons or nodes.Thus, a neural network is either a biological neural network, made up of biological neurons, or an artificial neural network, used for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled in artificial neural networks as weights between nodes. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred to as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be 1 and 1.

These artificial networks may be used for predictive modeling, adaptive control and applications where they can be trained via a dataset. Self-learning resulting from experience can occur within networks, which can derive conclusions from a complex and seemingly unrelated set of information.

3 Evaluation

3.1 Recursive Evaluation

We used the recursive evaluation method to train and test the model. We carried out 29 round in each model and made the model to run on the training set validation set and test set. For example:

- First Round training data: 1957-1974, validation data: 1975-1986, test data:1987
- Second Round training data: 1957-1975, validation data: 1976-1987, test data: 1988
- Final Round training data: 1957-2003, validation data: 2004-2015, test data: 2016

The figure[5] Recursive Evaluation below make it easy to have an understand:

3.2 Out of Sample Performance

According to the original paper, we define a new R2. The new R2 has certain advantages compared to traditional R2. Traditional R2 will have the denominator been averaged, which usually cannot

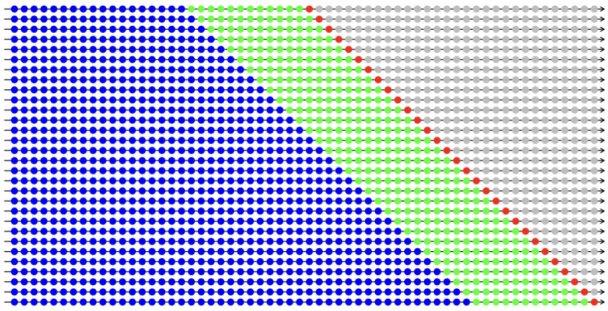

Figure 5: Recursive evaluation

achieve good results when calculating the expected return of individual stocks. Therefore, in the new R2, we directly use the square of the expected return instead of the square value of the average return in the past period as a denominator to achieve better results. The formula of R2 is shown in the figure:

$$R^2_{\text{oos}} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \widehat{r}_{i,t+1})^2}{\sum_{(i,t)\in\mathcal{T}_3} r^2_{i,t+1}},$$

We have tried totally 11 models, including linear model of OLS OLS3 PCR PLS and non-linear model RFR GBR NN1-5. Combined with the Recursive Evaluation method above, we have the average R2 obtained by each model in different years of test dataset, and get the chart shown below, tag all, top, bottom represent all stocks, 1000 biggest stocks and 1000 smallest stocks respectively:

	ols+h	ols3+h	pcr	pls	gbr+h	rf	nn1	nn2	nn3	nn4	nn5
all	-16.13	-0.08	1.67	0.62	2.64	2.43	2.89	4.42	4.64	4.68	1.40
top	-95.88	0.01	2.62	-0.19	3.04	2.74	5.30	5.17	3.65	4.55	2.21
bottom	-78.88	0.01	1.43	1.88	2.99	3.23	4.48	4.19	3.72	4.20	2.80

Figure 6: Annual out of sample performance(R2 percentage)

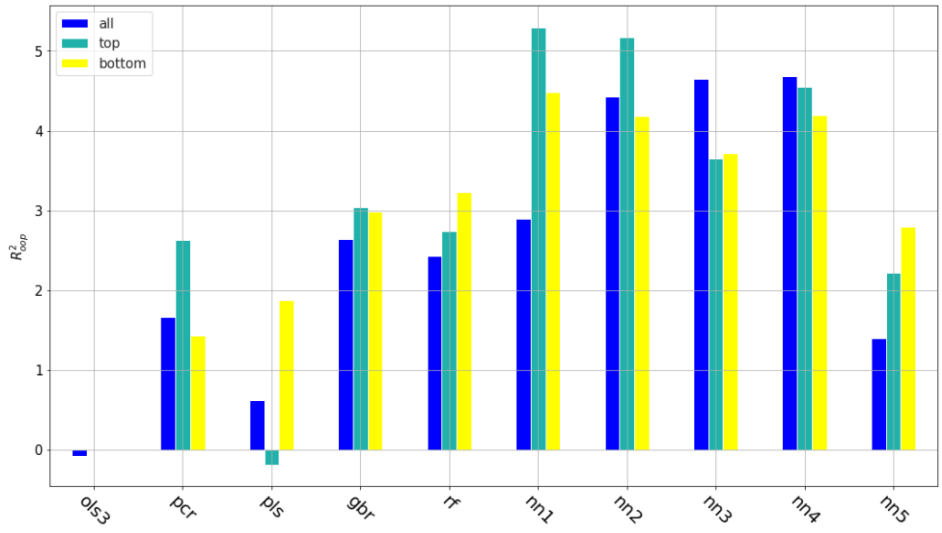

Figure 7: Annual return

From the above figure and table, we can see that the results are basically similar to the original paper. In terms of liner models, OLS model has a worst performance, because there are obvious over-fitting due to a large number of redundant variable. Through directly limit the number of features to 3, OLS3 has a better performance compared to OLS model. PLS and PCR have retained enough effective features by using dimensions reduction method. The results are further improved.

The results of non -linear models are generally better than linear models. The performance of GBR and RF has some improvement compared to the liner models. Due to the limitations of computing power and memory, the number of sub-models in the ensemble learning has some restrictions. In theory, the performance of GBR and RF has more space for improvement. Neural Network get the best performance in all models. The overall performance rise and then decrease with more layers added to the network. We also observe that the overall 3 dataset performance of NN2 in our model is the best, and NN5 has a significant decline in performance due to over-fitting. This is a little different compared to the original paper, so there is further space for discussion.

According to the Diebold-Mariano method, we also compare the performance between different models. The positive result in (i,j) represents the model in the j column is better than the model of the i row, and if negative means worse, the value reflects the size of the performance gap.The table is shown below:

	OLS+H	OLS-3+H	PLS	PCR	GBR+H	RF	NN1	NN2	NN3	NN4	NN5
OLS+H	0.0	2.75	2.86	3.04	3.21	3.17	3.25	3.51	3.55	3.56	3.00
OLS-3+H		0.0	0.12	0.3	0.47	0.43	0.51	0.77	0.81	0.81	0.25
PLS			0.0	0.18	0.35	0.31	0.39	0.65	0.69	0.69	0.13
PCR				0.0	0.17	0.13	0.21	0.47	0.51	0.52	-0.05
GBR+H					0.0	-0.04	0.04	0.3	0.34	0.35	-0.21
RF						0.0	0.08	0.34	0.38	0.39	-0.18
NN1							0.0	0.26	0.3	0.31	-0.26
NN2								0.0	0.04	0.04	-0.52
NN3									0.0	0.01	-0.55
NN4										0.0	-0.56
NN5											0.00

Figure 8: Model compare

3.3 Feature Importance

In the assessment of variable importance, we basically adopted the method of the original paper. For PLS PCR OLS, we get the importance of the characteristics by setting the variable to zero one by one and observe the change of R2. In the neural network, we also use the SSD formula described in the paper to compare the importance of characteristics. Due to the limitations of computing power, we don't use the forward method in the tree models(RF GBR), but directly call the Feature Importance function. However, basic principles of the function is similar to the above, and we have normalized all the results, so there should not be effect to the results.The features importance of different models are shown in the figure below:

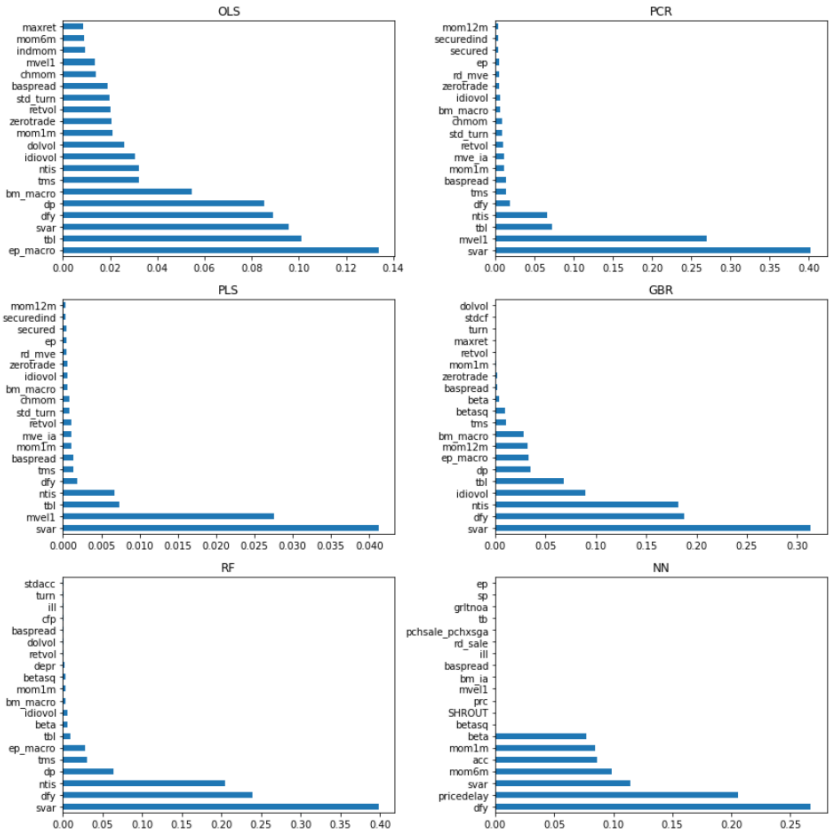

Figure 9: Feature importance

After combining the features of different models, we give the importance sorting of the features with the use of the heat map. The result is as shown in the figure 10 next page:

4 Conclusion

We have basically completed the replication of the main content of the original paper, and most of the results are similar to it. But by the restrict of computing power and time, we still have some spaces that are not good enough. For example, we have not tried the E-Net model and the linear model with regularization. Some results of the model are different from the original paper. And we have not considered the features association and interaction in the 2.3.1, 2.3.2 of the paper. These can be further improved in the future.

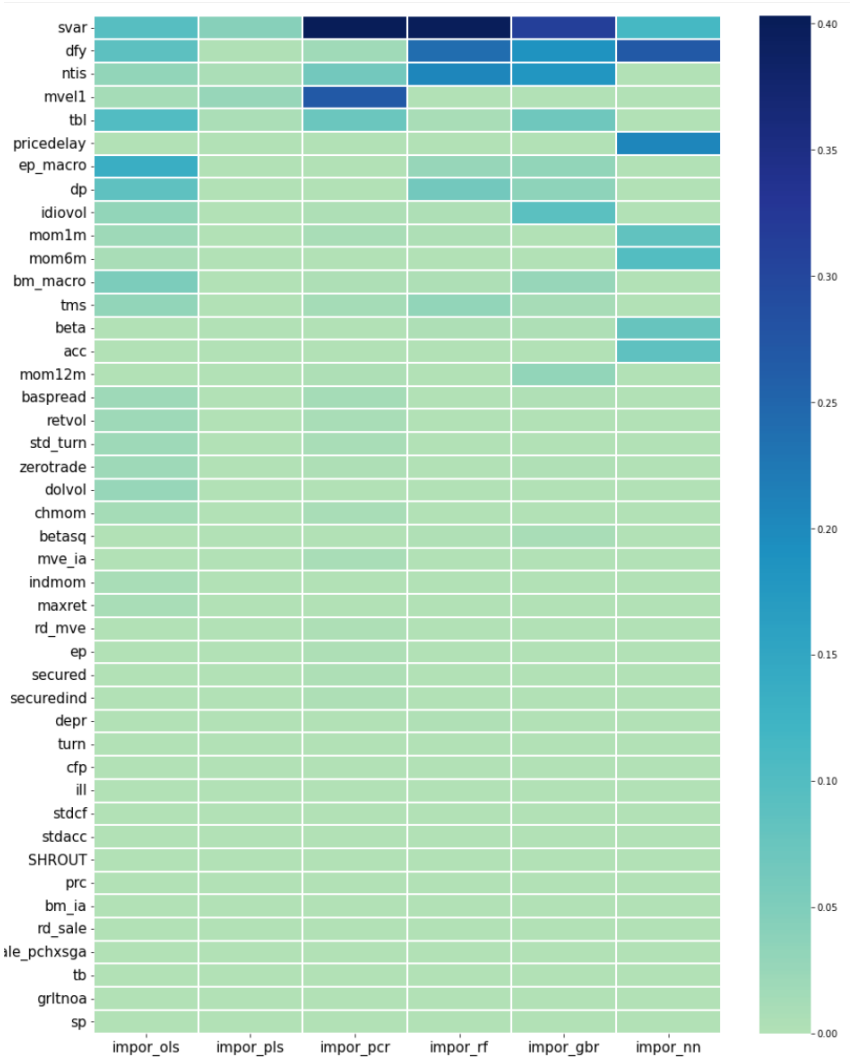

Figure 10: Feature heatmap

References

[1] https://dachxiu.chicagobooth.edu/download/ML.pdf

[2] https://www.schmidheiny.name/teaching/ols.pdf

[3] https://en.wikipedia.org/wiki/Partial_least_squares_regression

[4] https://en.wikipedia.org/wiki/Gradient_boosting

[5] https://yao-lab.github.io/course/msbd5013/2022/slides/Xiu_Papers.pdf

[6] https://yao-lab.github.io/aifin/2021/