Paper Replication:
Empirical Asset Pricing via Machine Learning
by Gu, Kelly, and Xiu (2018)

Zhen HOU
Department of Mathematics, HKUST
Clear Water Bay, HK
zhouah@connect.ust.hk

Jianda MAO
Department of Mathematics, HKUST
Clear Water Bay, HK
jmaoao@connect.ust.hk

Xiaolong WANG
Department of Mathematics, HKUST
Clear Water Bay, HK
xwangid@connect.ust.hk

Abstract

This report replicates Empirical Asset Pricing via Machine Learning by Gu, Kelly, and Xiu (2018). Measuring asset risk premiums is one of the canonical problems of empirical asset pricing. The original paper conducts a comparative analysis of machine learning approaches to predict the expected returns of a large number of assets. This replication study aims to reproduce the main results of the original paper, including data proprocessing, conducting machine learning prediction, performance evaluation and variable improtance. Our results confirm the findings of the original paper and provide further technical details.

1 Introduciton

In this work, the excess return of an asset is modeled by[1]:

$$r_{i,t+1} = \mathbb{E}_t(r_{i,t+1}|z_{i,t}) + \epsilon_{i,t+1}, \tag{1}$$

where

$$\mathbb{E}_t(r_{i,t+1}|z_{i,t}) = g^\star(z_{i,t}). \tag{2}$$

Stockes are indexed by $i = 1, \dots, N_t$ and months by $t = 1, \dots, T$. The excess return of stock i at time t is denoted by $r_{i,t+1}$, which can serve as reponse variable in language of machine learning. And $z_{i,t}$ is a vector of predictor variable of stock i at time t. The function $g^\star(\cdot)$ is the conditional expectation of the excess return, e.g. risk premium, of stock i at time $t+1$ given the predictor variable $z_{i,t}$. The main goal is to estimate $g^\star(\cdot)$, which is a typical prediction task and attractive for machine learning methods.

As a replication report, we focus more on the technical detials rather than the research contribution, compared with the original paper. The rest of the report is organized as follows. Section 2 describes the dataset and preprocessing. Section 3 introduces the machine learning models used in the original paper. Section 4 presents the performance evaluation. Section 5 discusses the variable importance. Section 6 concludes the report.

[1] Here and bolow, we use a bit more mathematical notation from the original paper.

The final project report of Course MATH 5470: Statistical Machine Learning, Spring 2024.

2 Dataset and Preprocessing

The raw dataset[2] contains almost 30000 stocks from 1957 to 2020, with average 6200 stocks per month. For per stock, there include the end day of each month(DATE[3]) t, 94 characteristics $c_{i,t}$, the first two digits of SIC code(sic2) and lag-adjusted CRSP returns(RET) $r_{i,t+1}$ which could be served as responce variable directly.

2.1 Data Cleaning and Construction

We impute missing values for each characteristic by the median of the corresponding subgroup of stocks classified by the belonging month. In addition, we include 74 industry dummies $s_{i,t}$ using one-hot encoding corresponding to the first two digits of SIC code. Following the original paper, we construct 8 macroeconomic predictors denoted as x_t. The final covariate $z_{i,t}$ in the original work is calculated by

$$z_{i,t} = [[x_t, 1] \otimes c_{i,t}, s_{i,t}], \tag{3}$$

where $\otimes$ denotes the Kronecker product and P-dimensional vector is viewed as a $1 \times P$ matrix. However, limited by the computational resources, we calculated the predictor variable $z_{i,t}$ by

$$z_{i,t} = [x_t, c_{i,t}, s_{i,t}]. \tag{4}$$

in our experiment while the code for (3) is also provided.

2.2 Data Splitting

Following the original paper, we devide the 64 years of data into 18 years of training data(1957-1974), 12 years of validation data(1975-1986) and 34 years of test data(1987-2020). We refit models every year while increasing the training data by one year and maintain the same size of validation by rolling it forward until including all the data. After each refit, we use the model to predict the next year's excess returns. We performed standardization on the training data and applied the same transformation to the validation and test data.

3 Machine Learning Models

We conducted OLS, OLS-3, ENet, PCR, PLS, GBRT, NN1, NN2, NN3, NN4 and NN5 models in our experiment. Most of them have been introduced carefully in our course and the original paper. Thus, here we just discuss some details when conducting OLS-3 and NN models.

3.1 OLS-3: Linear Regression with 3 Factors

OLS-3 is the classical empirical asset pricing linear regression model using 3 factors as predictors. The three factors are the book-to-market(bm), the size(mvel1), and the monentum(mon1m, mon6m, mon12m, mon36m) while for monetum there are 4 characteristics calculated by data from the past 1, 6, 12, and 36 months, respectively. Thus, there are totally 6 predictors appearing in OLS-3, formulated as

$$g^\star(z_{i,t}) = \theta_0 + \theta_1 bm_{i,t} + \theta_2 mvel1_{i,t} + \theta_3 mon1m_{i,t} + \theta_4 mon6m_{i,t} + \theta_5 mon12m_{i,t} + \theta_6 mon36m_{i,t}. \tag{5}$$

3.2 NN: Neural Networks

We conducted several neural networks with different, but up to 5, hidden layers. Following the original paper, the number of neurons in each layer is choosen according to the geometric pyramid rule. The detailed hidden layers setting is shown in Table 1. We use ReLU as the activation function and Adam as the optimizer. The loss function is mean squared error.

4 Performance Evaluation

To evalue predictive performance, we calculated the out-of-sample R^2 and the time-varing model complexity and conducted the Diebold and Mariano (1995) test.

[2] obtained from `https://www.dropbox.com/s/zzgjdubvv23xkfp/datashare.zip?dl=0`

[3] The characteristic name in dataset. The following cases are similar.

	NN1	NN2	NN3	NN4	NN5
neurons	32	32, 16	32,16,8	32,16,8,4	32,16,8,4,2

Table 1: Number of neurons in each hidden layer of NN models

4.1 Out-of-Sample R^2

For testing sample $\mathcal{T}_3$, the out-of-sample R^2 is defined as

$$R^2_{oos} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \hat{r}_{i,t+1})^2}{\sum_{(i,t)\in\mathcal{T}_3} r^2_{i,t+1}}. \qquad (6)$$

Notice that the denominator here is the sum of squared repsonce variables, not the sum of the squared residuals, as a naive forecast of zero typically outperfoms mean when predicting returns.

To evaluate the general prediction performance, we chose $\mathcal{T}_3$ as the whole testing sample (year 1987-2020) although prediction is conducted for each year separately. The results are shown in Table 2 and Figure 1.

	OLS	PCR	PLS	GBRT	NN1	NN2	NN3	NN4	NN5	OLS3	ENet
R^2_{oos}	-3e25	-0.02	-4.26	-0.53	-0.06	0.23	-0.19	-0.16	-0.0	0.16	0.16

Table 2: Monthly out-of-sample prediction performance (percentage R^2)

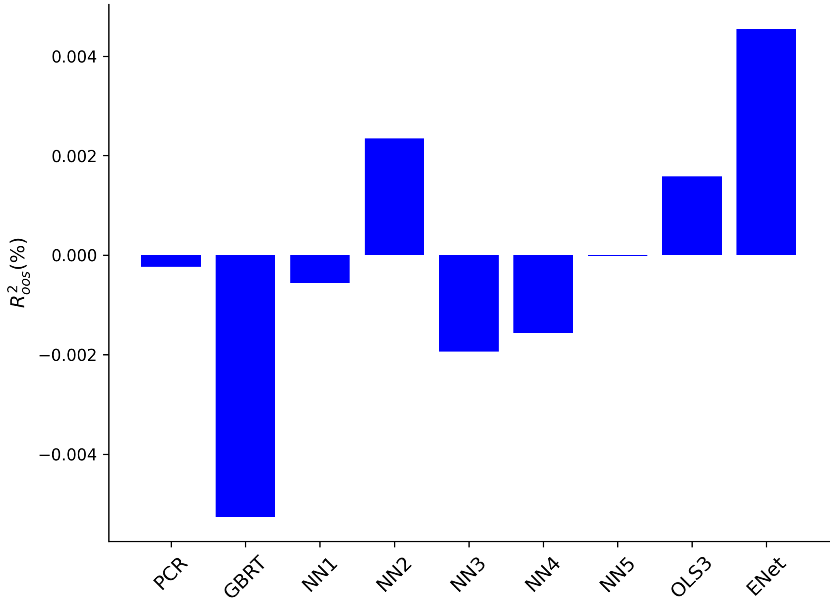

Figure 1: Out-of-sample R^2 performance of different models except OLS and PLS

The OLS's extreme large R^2_{oos} supports that due to so many parameters, OLS produces highly unstable out-of-sample forecasts. OLS3 has 0.16% R^2_{oos} which is consistent with the original paper, showing the efficiency of the classical method. The negative R^2_{oos} of PCR and PLS arise from the absence of Kronecker product, e.g. the interactions among characteristics and macroeconomic predictors, in equation 4. This supports the conclusion about the importance of interactions in the original paper, from a different perspective. Limited by the computational resources, the performance of GBRT is much worse than the original paper.

For NN models, only NN2 performs with a positive R^2_{oos}. Increasing the number of neurons in hidden layers may improve the performance of NN models. But limited by time and computational resources, we didn't conduct such experiments. We also observed the phenomenon that "shallow" learning outperforms "deep" learning, as NN2 performs best among NN models with R^2_{oos} of 0.23%. Comparing NN with OLS3, it shows that machine learning methods are potential in risk premium prediction, mainly because of the nonlinearity brought in.

The best performance model in our experiment is ENet with R^2_{oos} of 0.46%, which is a surprising result.This may be due to that predictor variable we constructed by equation (4) is simpler than the original paper, which decreases the influence of irrelevant variables and thus improves the performance of ENet.

4.2 Time-varing Model Complexity

We evaluated the time-varing model complexity of PCR, PLS and GBRT. For PCR and PLS, we use the number of components for evalutaion while the maximal depth for GBRT. The results is shown in Figure 2.

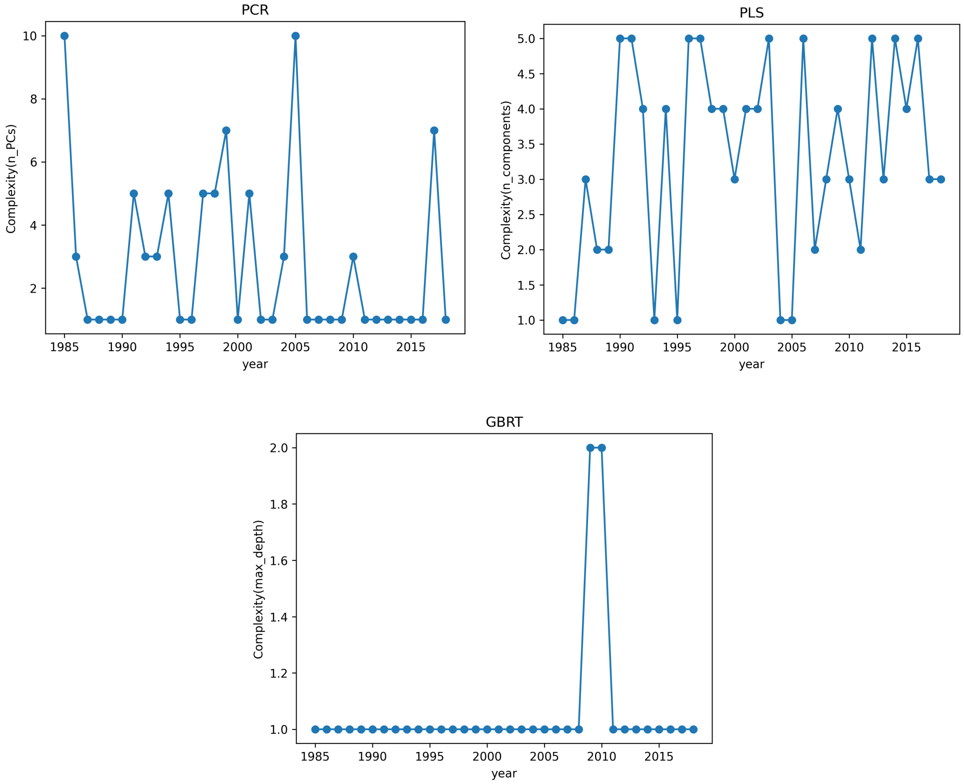

Figure 2: Time-varing model complexity

For PCR, the numbers of components are mostly below 5, while for PLS, the numbers are mostly below 4. For GBRT, the maximal depth is always 1 or 2. The results are consistent with the original paper, showing that the models are relatively simple.

4.3 Diebold and Mariano Test

Diebold and Mariano test is used to compare the out-of-sample predictive performance of model (1) versus model (2). Denote $\hat{e}^{(1)}_{i,t}$ and $\hat{e}^{(2)}_{i,t}$ as the prediction errors for stock i at testing month t of model (1) and model (2) respectively. Denote $\mathcal{T}_{3,t}$ as the set of stocks in the testing month t. Let

$$d_{12,t} = \frac{1}{|\mathcal{T}_{3,t}|} \sum_{i \in \mathcal{T}_{3,t}} \left((\hat{e}^{(1)}_{i,t})^2 - (\hat{e}^{(2)}_{i,t})^2 \right). \tag{7}$$

The test statistic is defined as $DM_{12} = \bar{d}_{12}/\hat{\sigma}_{12}$, where $\bar{d}_{12}$ is the average of $d_{12,t}$ and $\hat{\sigma}_{12}$ is the Newey-West standard deviation, calculated by

$$\hat{\sigma}_{12} = \frac{1}{T}\sqrt{\sum_{t=1}^{T}(d_{12,t} - \bar{d}_{12})^2 + 2\sum_{l=1}^{L}\sum_{t=l+1}^{T}(1 - \frac{l}{1+L})(d_{12,t} - \bar{d}_{12})(d_{12,t-l} - \bar{d}_{12})}. \quad (8)$$

Here we choose $L = T - 1$. And notice that σ_{12} is the autocorrelation-adjusted estimate of standard deviation of $\bar{d}_{12}$. Thus, under the null hypothesis, DM_{12} follows a standard normal distribution. We calculated the Diebold-Mariano statistic for each pair of models and the results are shown in Table 3.

	PCR	PLS	GBRT	NN1	NN2	NN3	NN4	NN5	OLS-3	ENet
OLS	$\mathbf{1.73}$	$\mathbf{1.73}$	$\mathbf{1.73}$	$\mathbf{1.73}$	$\mathbf{1.73}$	$\mathbf{1.73}$	$\mathbf{1.73}$	$\mathbf{1.73}$	$\mathbf{1.73}$	$\mathbf{1.73}$
PCR		$\mathbf{-2.52}$	-1.56	-0.38	$\mathbf{3.28}^\star$	-0.72	-1.56	0.06	$\mathbf{3.09}^\star$	$\mathbf{6.35}^\star$
PLS			$\mathbf{2.27}$	$\mathbf{2.54}$	$\mathbf{2.59}^\star$	$\mathbf{2.66}^\star$	$\mathbf{2.44}$	$\mathbf{2.39}$	$\mathbf{2.64}^\star$	$\mathbf{2.86}^\star$
GBRT				$\mathbf{1.68}$	$\mathbf{2.16}$	1.20	1.30	1.40	$\mathbf{2.26}$	$\mathbf{3.06}^\star$
NN1					$\mathbf{2.07}$	-0.55	$\mathbf{-2.09}$	0.29	$\mathbf{2.81}^\star$	$\mathbf{4.90}^\star$
NN2						-1.55	$\mathbf{-3.20}^\star$	$\mathbf{-2.31}$	-0.52	$\mathbf{2.26}$
NN3							-0.04	0.53	$\mathbf{1.70}$	$\mathbf{3.22}^\star$
NN4								1.17	$\mathbf{4.38}^\star$	$\mathbf{6.28}^\star$
NN5									1.34	$\mathbf{3.88}^\star$
OLS3										$\mathbf{3.17}^\star$

Table 3: Comarison of monthly out-of-sample prediction using Diebold-Mariano tests

Following the original paper, we conducted two type of tests. (1) Individaul pairwise models comparison test, with the null hypothesis that the two models have the same out-of-sample performance versus the alternative hypothesis that the first model has better performance. The rejection region is $DM_{12} > \Phi^{-1}(1 - 0.05) = 1.645$ for 0.05 significance level. (2) Joint pairwise models comparison test, with the null hypothesis that all models have the same out-of-sample performance versus the alternative hypothesis that at least one model has better performance. Under a conservative Bonferroni correction, the rejection region is $DM_{12} > \Phi^{-1}(1 - 0.05/10) = 2.576$ for 0.05 significance level amid 10-way[4] comparisons.

In Table 3, a positive value indicates the column model outperforms the row model while a negative value indicates the row model is better. By symmetry, we just show the upper triangle of the table. The bold values indicate the difference is significant at 0.05 level. The star symbol indicates significance at 0.05 level for 10-way comparisons. Concluded from Table 3, the ENet model outperforms all other models significantly.

There may be a confusion about that the D-M statistics between OLS and other models are same and relatively small, e.g. the first row of Table 3. The reason is that as shown in equation (7), $d_{12,t}$ is the difference of month t MSE between two models. In our experiment, the any month t MSE of OLS is much larger than that of other models, which leads to the D-M statistics. This is why the D-M statistics are almost same. But unfortunately, the magnitude of OLS's MSE is not same between different months. A few of them are much larger than any other. Thus, the D-M statistics are relatively small.

5 Variable Importance

We use permutation feature importance algorithm to study the variable importance. The results are shown in Figure 3. Generally, the bid-ask spread(baspread) and macroeconomic predictors(svar, d/p, etc) are important in our result.

[4] 12-way in the original paper.

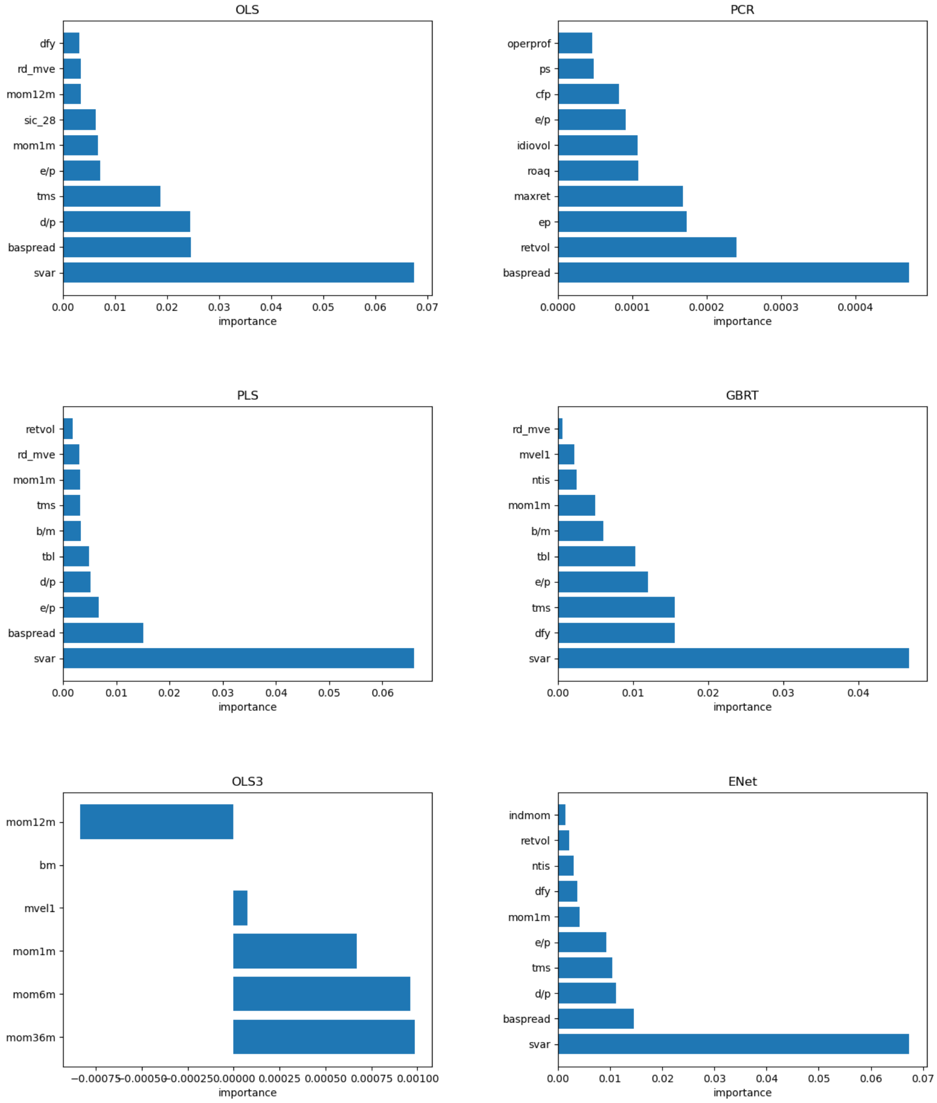

Figure 3: Variable importance

6 Conclusion

In this report, we replicated the main results of the original paper Empirical Asset Pricing via Machine Learning by Gu, Kelly, and Xiu (2018). We conducted data preprocessing, machine learning prediction, performance evaluation and variable importance. Our results confirm the findings of the original paper.

Furthermore, as a final project report for a math course, we rewrite some notations more mathematically and provide more technical details, which is helpful for readers to understand the original paper and reproduce the results.

Limited by time and computational resources, we did not conduct the full replication of the original paper, such as the Kronecker product of raw data and the performance of top-1000 and bottom-1000 stocks. However, we provided more codes than presented in this report, which could be found in our GitHub repository[5] for the interested.

[5] `https://github.com/hdsfade/math5470-final-project`

Contribution

Zhen HOU is responsible for the code of PCR, performance evaluation, presentation and report writing.

Jianda MAO is responsible for data preprocessing, training framework implementation, code of ENet, GBRT, NNs and variable importance.

Xiaolong WANG is responsible for the code of OLS, OLS-3, PLS and running the experiments.

References

[1] Shihao Gu, Bryan Kelly, Dacheng Xiu, Empirical Asset Pricing via Machine Learning, The Review of Financial Studies, Volume 33, Issue 5, May 2020, Pages 2223-2273, `https://doi.org/10.1093/rfs/hhaa009`