Paper Replication: Empirical Asset Pricing via Machine Learning

ZHAO JUNDA
MSc of Financial Mathematics
The Hong Kong University of Science and Technology
jzhaobr@connect.ust.hk

LI MINGLUO
MSc of Financial Mathematics
The Hong Kong University of Science and Technology
mlicv@connect.ust.hk

HE HAOKAI
MSc of Financial Mathematics
The Hong Kong University of Science and Technology
hheap@connect.ust.hk

HUANG WENJIN
MSc of Financial Mathematics
The Hong Kong University of Science and Technology
whuangbk@connect.ust.hk

Abstract

The paper of Empirical Asset Pricing via Machine Learning discusses a canonical problem of empirical asset: measuring asset risk premium, and show a comparative analysis of machine learning methods. In this report, we try to replicate the paper and focus on the analysis of the variable importance using those different machine learning methods. And in the paper replication, we make some simplifications and adjustments for the preliminary study based on our understanding and the limitation of the machine environment capability.

1 Introduction

Generally speaking, to price the empirical asset, we need to find out and understand the behavior of risk premiums. However, in the realistic financial market, the market efficiency forces the return variation of the asset to be dominated by those unforecastable news which obscures risk premiums and makes them hard to be measured. In this case, the paper[1] conduct a comparative analysis of machine learning methods for this financial problem.

It is clear that return prediction is economically meaningful. However, even if it is perfectly observed, the risk premium is still notoriously difficult to measure because of the market efficiency. To work further on this situation, the paper highlights gains that can be achieved in the return prediction and identifies those most important predictor variables. Also, comparing to the traditional empirical asset

[1] In this report we try to replicate the paper of Empirical Asset Pricing via Machine Learning, see Gu et al. (2020).

2nd Project MAFS 6010Z AI in Fintech (Worked by Team LI/HE/HUANG/ZHAO 2021), Hong Kong, China.

pricing methods, machine learning accommodates far more expansive list of those potential predictor variables. So in this paper replication, we would focus on the part of analyze the importance of the predictor variables which selected by those different kinds of machine learning methods based on the data of monthly total individual equity returns for all firmed listed in NYSE, AMEX, NASDAQ for the last 60 years from 1957 to 2016.

2 Preliminary Study

2.1 Data Construction

To replicate the paper, we directly download part of the data [2] from the supplementary material of the paper. Then build a large collection of stock-level predictive characteristics based on the cross-section of stock returns literature which includes 94 characteristics [3] and 74 industry dummies corresponding to the first two digits of Standard Industrial Classification (SIC) codes.[4] And according to Gu et al. (2020), for the missing characteristics, we should replace with the cross-sectional median at each month for each stock, respectively. We have tried to replace the missing data in such way, however, we found that this method was limited by the machine environment capability even in the Google Colab. In this case, after analyzing on the characteristic, for most of them are updated annually, we replace those missing characteristics with the annually median for each stock respectively.
And according to Gu et al. (2020), there are also eight macroeconomic predictors[5] are constructed. Following the variable definitions detail from Welch et al. (2007), we construct the predictors with equations,

$$dp = log(D12) - log(Index)$$
$$ep = log(E12) - log(Index)$$
$$tms = lty - tbl$$
$$dfy = BAA - AAA$$

and we can get those eight macroeconomic predictors in Table 1.

Table 1: Macroeconomic Predictors

Predictor	Description
dp	dividend-price ratio
ep	earnings-price ratio
bm	book-to-market ratio
ntis	net equity expansion
tbl	Treasury-bill rate
tms	term spread
dfy	default spread
svar	stock variance

2.2 Features Analysis

Since that all of the machine learning methods we consider are designed to approximate the overarching empirical model $E_t(r_{i,t+1}) = g^*(Z_{i,t})$ for

$$r_{i,t+1} = E_t(r_{i,t+1}) + \epsilon_{i,t+1}$$

[2] Gu et al. (2020) obtain monthly total individual equity returns from CRSP for all firms listed in the NYSE, AMEX, and NASDAQ. The sample begins in March 1957 (the start date of the S&P 500) and ends in December 2016, totaling 60 years. And there are totally almost 30,000 stocks in the sample, with average 6,300 stocks per month. And they also obtain the Treasury-bill rate to proxy for the risk-free rate from which we calculate individual excess returns.

[3] The paper choose 61 of which are updated annually, 13 are updated quarterly, and 20 are updated monthly. It cross-sectionally rank all stock characteristics period-by-period and map these ranks into the [-1,1] interval.

[4] The details of these characteristics see Table A.6 in the Internet Appendix of Gu et al. (2020).

[5] We construct the monthly data from the web site of Goyal, and the variable definitions detailed following Welch et al. (2007).

In the paper, it defines the baseline set of stock-level covariates $z_{i.t}$ as in equation

$$z_{i,t} \;=\; X_t \;\otimes c_{i,t+1}$$

where $c_{i.t}$ is a $P_c \times 1$ matrix of characteristics of each stock i, and x_t is a $P_x \times 1$ vector of macroeconomic predictors. Thus, we need to consider the $P = P_c P_x$ which includes interactions between the stock-level characteristic and macroeconomic state variables. In this way, there would be about $94 \times (8 + 1) + 74 = 920$ covariates in each machine learning methods.

For one thing, the machine environment capability may not handle the data set with such huge size, and for another, some of the machine learning methods such as Random Forest, Neural Network have already considered the interaction among different features. In this case, to simplify the features construction and improve the feasibility of training the models, we decide to analyze the variables importance of those 94 stock-level individual characteristics and the 8 macroeconomic variables with the following machine learning methods respectively.

3 Methodology

In this section, to firstly check the feasibility and the performance of the different machine learning methods, the whole data set is involved to perform regression examples. We divide the data set into 3 parts namely the training set, the validation set and the test set. The first 18 years of the data set is assigned to the training set. Data set ranged from the 19th year to the 30th year is assigned to the validation set while the last 30 years of the data set is assigned to the test set. In regression examples of Gradient Boosted Regression Trees, Random Forest and Neutral Networks, the validation set is ignored and the 19th year of the data set is assigned to the test set. The training set, the validation set and the test set mentioned below is referred to the 94 stock-level characteristic variables by default unless otherwise indicated.

3.1 Performance Evaluation

According to Gu et al. (2020), the out-of-sample R^2, as shown in equation

$$R^2_{oos} = 1 - \frac{\sum_{(i,t)\in\tau_3}(r_{i,t+1} - \hat{r}_{i,t+1})^2}{\sum_{(i,t)\in\tau_3} r^2_{i,t+1}}$$

is calculated to assess the predictive performance for individual excess stock return forecasts. However, after calculating and comparing between the R^2_{oos} and the score in different machine learning regression methods respectively, we find that the differences are too small to affect the final result. In this case, to simplify the calculating progression, it is reasonable and feasible to directly use the regression score to evaluate the prediction performance, and the regression score calculation could be adjusted based on different machine learning methods. More details would be stated in the later regression methods discussion.

3.2 OLS

Compared with models that have non-linear relation to their parameters, estimation of unknown parameters in linear models is less complex. In addition, it's easier to calculate the statistical properties of estimators in linear models.

With the original least square loss function, we have a score of 0.0495364 for the training set and a score of 0.0214496 for the test set.

Extension: Huber loss

As shown in Gu et al. (2020), replacing the least square loss function with the Huber loss function as shown in equation,

$$H(x;\xi) = \begin{cases} x^2, & if\ |x| \leq \xi \\ 2\xi|x| - \xi^2, & if\ |x| > \xi \end{cases}$$

and assigning $\xi = 1.35$, we have a score of 0.0145323 for the test set.
As for the OLS3 + H regression, we select "mvel1", "bm" and "mom1m" as independent variables and get a score of -0.0035302 for the test set.

3.3 Ridge

The lambda set for the Ridge Regression is a series of power of 10 multiplied by 0.5 where the exponents are the first 100 terms of an arithmetic sequence with an initial value of 10 and a common difference of -0.12. We select the lambda that has the best R2 score in the validation set to fit the training set. We have a score of 0.0495363 for the training set and a score of 0.0214306 for the test set.

$$J(\theta) = \frac{1}{2}\sum_{j}^{m}(y^{(i)} - \theta^T x^{(i)})^2 + \lambda \sum_{j}^{n} \theta_j^2$$

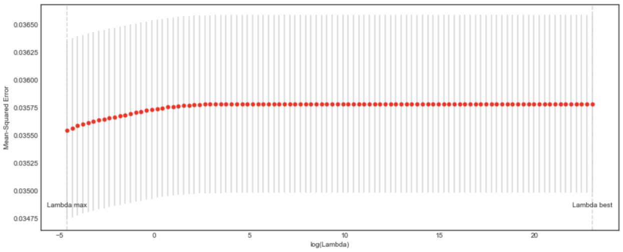

Figure 1: Ridge: log(lambda) vs MSE

3.4 Lasso

We applied coordinate descent to reach the specified tolerance for the optimal lambda of Lasso Regression for the validation set with a maximum iteration of 10000 and a model selection method of 10-fold cross validation.
In this subsection we perform Lasso Regression for both the stock-level characteristic variables and macroeconomics variables respectively. Applying the optimal lambda for the Lasso Regression we have a score of 0.021518 for the test set of 94 stock-level characteristic variables and a score of 0.0016001 for the test set of macroeconomics variables.

$$J(\theta) = \frac{1}{2}\sum_{j}^{m}(y^{(i)} - \theta^T x^{(i)})^2 + \lambda\rho \sum_{j}^{n} |\theta_j|$$

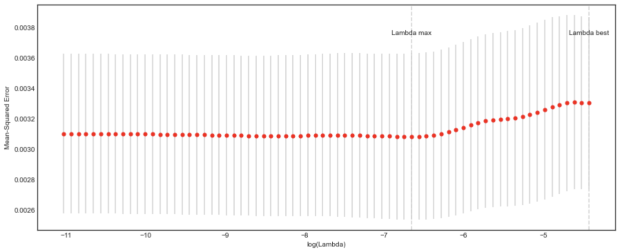

Figure 2: Lasso (Macroeconomics): log(lambda) vs MSE

3.5 Elastic net

As shown in the equation,

$$J(\theta) = \frac{1}{2}\sum_{j}^{m}(y^{(i)} - \theta^T x^{(i)})^2 + \lambda(\rho \sum_{j}^{n} |\theta_j| + (1-\rho)\sum_{j}^{n} \theta_j^2)$$

we assigned 0.5 to alpha and estimate the optimal lambda of Elastic Net Regression for the validation set with a model selection method of 10-fold cross validation.
Applying the optimal lambda for the Elastic Net Regression we have a score of 0.0283556 for the test set.

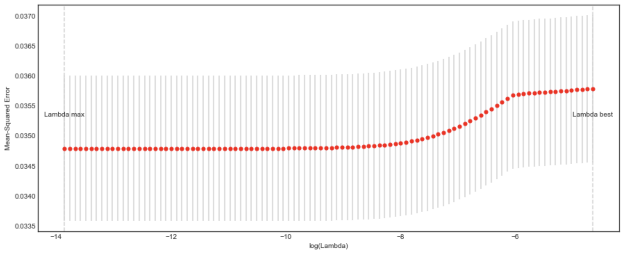

Figure 3: Elastic Net: log(lambda) vs MSE

3.6 Principal Components Regression

In regression analysis, the larger the number of explanatory variables allowed, the greater is the chance of overfitting the model, producing potential poor prediction score for the test set. PCA is an effective way to reduce the chance of overfitting by dimensionality reduction while preserving the data's variation as much as possible.
After applying the PCA, the OLS provides us with a score of -0.083994 with least square loss and a score of -0.1705276 with Huber loss for the test set respectively.

3.7 Gradient Boosted Regression Trees

We assign 2 to the max depth of trees and 20 to "random_state" for the Gradient Boosted Regression Trees. Besides, we apply Huber loss to evaluate the loss. Finally we have a score of 0.189715 for the training set and a score of 0.1594216 for the test set.

3.8 Random Forest

Random Forest reduces probability of over-fitting in decision trees by reducing the correlation among trees. Generally it outperforms decision trees but it's not as accurate as gradient boosted trees.
We define 300 trees with a maximum depth of 6 in the forest. We also assume 10 features to consider when looking for the best split and assign 20 to "random_state". Finally we have a score of 0.052631 for the training set and a score of 0.0233687 for the test set.

3.9 Neural Networks

We establish 5 Neural Networks models that have different layers. We set the early stopping to be 10 which means the minimum validation sample errors should be updated in no more than 10 steps otherwise the optimization is terminated. In addition, we select model that has the minimum validation sample errors among 5 models to be the decisive model. Regression arguments are listed in Table 2. In this regression example, we have a score of 0.551291 for the training set and a score of 0.5065095 for the test set.

And about the variables importance part of Neural Networks, for example, let us focus on NN-3, which is in the form of

$$(f \circ g \circ h)(x)$$

We firstly use a zero vector as a test set and determine the output as a basic score of the NN-3 regression model. Then we use the unit vector of each predictors to get the different output and minus the basic score to get the marginal change of score of each predictors respectively. And we define the

Table 2: NN Arguments

	Hidden Layer	Feature Count	L1 λ	Learning Rate	Learning Rate Decay
NN1	32	98	1e-5	0.001	1e-7
NN2	(32, 16)	98	1e-5	0.001	1e-7
NN3	(32, 16, 8)	98	1e-5	0.001	1e-7
NN4	(32, 16, 8, 4)	98	1e-5	0.01	1e-7
NN5	(32, 16, 8, 4, 2)	98	1e-5	0.01	1e-6

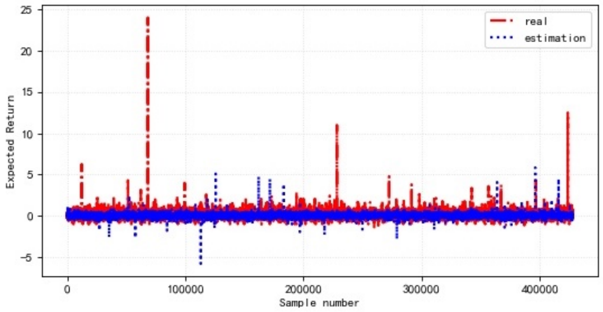

Figure 4: NN Prediction Performance

change of the output as the variable importance for each predictors.

3.10 Recursive Performance Evaluation Scheme

As told in Gu et al. (2020), we should use a ’recursive performance evaluation scheme’ to evaluate the result. Similar to the regression example statement before, the data set would be divided into 18 years of training sample, 12 years of validation sample, and remaining 30 years for out-of-sample testing. We refit the model once every year and increase the training sample by 1 year keeping the length of validation rolling window unchanged. And for those machine learning methods which do not need validation sample, we ignore the validation period and directly use the test set starts from the 31st year.

4 Result

Using the previous methods, we get the model score as well as the variable importance of predictors for each regression methods every recursive times for 94 stock-level characteristic variables and 8 macroeconomic predictors respectively. Then in each model, we rank the importance level for all predictors in every recursive times. As shown in the Figure 6-5, the variables importance are ordered based on the sum of their rank over each models respectively. And in both figures, for each individual model, the variable which is more influential shows with more dark blue, while the less influential one shows with lighter colour.
Similar to the result in Gu et al. (2020), from the Figure 6, dfy (default spread) is a very important variables in most of the models among all macroeconomic predictors. While about other predictors, the result shows quite different performance with different models. And about the Characteristic Importance in Figure 5, the mvel1 and the mom series of variables do show highly influential to the prediction, which is also similar to Gu et al. (2020).
However, since we have simplified the construction of the features, we cannot particularly analyze the interaction between the characteristic predictors and the macroeconomic variables like what Gu et al. (2020) did in the paper.

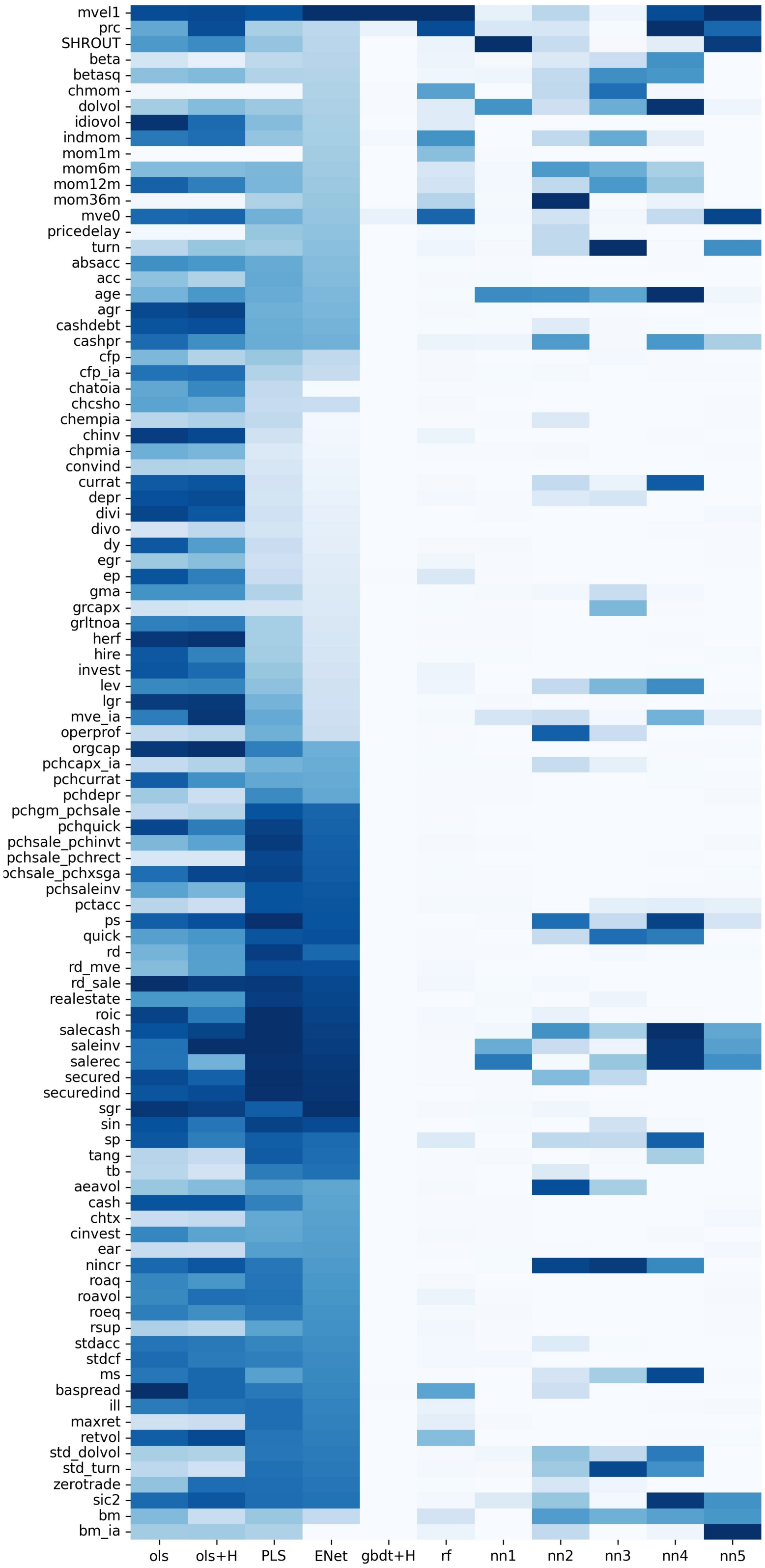

Figure 5: Characteristic Importance

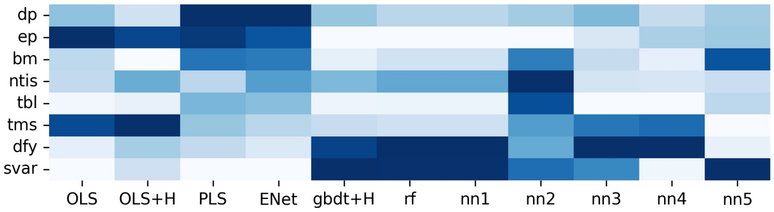

Figure 6: Variable Importance for Macroeconomic Predictors

5 Contribution

Python Coding: ZHAO JUNDA, HUANG WENJIN
Report Writing: LI MINGLUO, HE HAOKAI

References

Gu S, Kelly B, Xiu D. Empirical Asset Pricing via Machine Learning[J/OL]. The Review of Financial Studies, 2020, 33(5):2223-2273. https://doi.org/10.1093/rfs/hhaa009

Goyal A. [EB/OL]. https://sites.google.com/view/agoyal145/

Welch I, Goyal A. A Comprehensive Look at The Empirical Performance of Equity Premium Prediction[J/OL]. The Review of Financial Studies, 2007, 21(4):1455-1508. https://doi.org/10.1093/rfs/hhm014