Paper Replication: Empirical Asset Pricing via Machine Learning

**MATH 5470: Statistical Machine Learning Final Project Report  
Spring 2022**

**Group member:**

**Zhefeng CAO**

The Department of Civil and Environmental Engineering

**Mengyuan CHU**

The Division of Environment and Sustainability

**Contribution:**

Data Pre-processing: both

Zhefeng CAO: Results about the eNet, PLS, NN, DM test, and finalize the code

Mengyuan CHU: Results about the OLS , PCR and RF

# Overview

In this project, we are trying to replicate the paper “Empirical Asset Pricing via Machine Learning”. This paper predicts the expected return (‘RET’ in the dataset) and identifies informative predictor variables via machine learning methods, facilitates more reliable investigation into economic mechanisms of asset pricing. We are required to replicate section 2.2 and variable importance (in section 2.3) of the paper.

The reasons we chose this project including:

1) We are both beginners of Machine Learning (ML) and have very limited experiences. This paper was very detailed and comprehensive which may help us to put hands on more than one type of ML models;  
2) This paper is a comparative analysis, showing us the detailed process of identifying the best-performing methods among the most common models in real world data. As beginners, this is quite important as we both want to apply ML in our future research in different area.   
3) This research question of the paper is using ML for time series forecasting which is also what we are trying to do in our future research. Replicate this paper may help us avoiding some of the common pitfalls.

## Data

The data including the monthly total individual equity returns (‘RET’) from CRSP for all firms listed in the NYSE, AMEX, and NASDAQ. The data begins in January 1926 and ends in December 2016, but only used the data from 1957, totaling 60 years. The number of stocks in the dataset is almost 30,000, with the average number of stocks per month exceeding 6,200.28

The panel data has 4345508 rows and 101columns, includes 94 characteristics for each stock (61 of which are updated annually, 13 are updated quarterly, and 20 are updated monthly). The first two digits of the Standard Industrial Classification code(‘sic2’). And other variables that didn’t included in the train-validation-test process like the price of the stock(‘prc’), market value(‘’) for plotting, ‘permno’ CRSP Permanent Company Number.

To better understand the data, we firstly conducted a basic exploratory data analysis. There is no doubt that the most important column in our project is “RET” – the monthly total individual equity returns. It is lognormally distributed even with a logarithmic y-axis, it is highly skewed towards the left. (Figure 1). ML algorithms are sensitive to the range and distribution of attribute values. Such outliers can spoil the training process resulting in longer training times, less accurate models, and ultimately poorer results.

图表, 直方图

描述已自动生成 图表

描述已自动生成

Figure 1. Distribution of the monthly total individual equity returns (Left) and time-series plot of one stock (permno=17830)

Due to the imitated memory of our available devices and the huge data size of the study, we conducted the similar but smaller train-validation-test scheme, focusing mainly in understanding the whole progress from the simplest model the simple linear predictive regression model (OLS) to arguably the most powerful modeling neural network (NN). We didn’t include the interactions between stock-level characteristics and macroeconomic state variables for the same reason. Therefore, the figures and tables in this report are quite different from what in the Gu et al. (2020). More detailed description can be found in the following section.

## Data pre-pressing (both)

The pro-pressing including mainly 3 parts: generating of the 8 macroeconomic parameters, dummy variable creation and missing data. We didn’t include interactions between stock-level characteristics and macroeconomic state variables as this enlarge the data even 8 times.

Eight macroeconomic predictors were constructed followed the variable definitions detailed in Welch and Goyal (2008) with data downloaded from Amit Goyal's [webpage](http://www.hec.unil.ch/agoyal/). After creation, the 8 parameters were merged with the raw data. Then we merged the 2 datasets by time. Followed by replacing missing data with the cross-sectional median at each month for each stock respectively. 74 dummy variable was created from column “sic2”.

Below are some of the time series plots of the 8 macroeconomic parameters:

图表, 折线图

描述已自动生成图表

描述已自动生成图表

描述已自动生成图表

描述已自动生成图表, 直方图

描述已自动生成图表

描述已自动生成

Figure 2. Generated 8 macroeconomic parameters

# Models

We selected the following 6 methods: OLS with all variables (OLS), OLS using only size, book-to-market, and momentum (OLS-3), PCR, PLS, elastic net (ENet), random forest (RF), and neural networks.

## OLS (Mengyuan CHU)

Same as the paper, the OLS was selected as it is the least complex method, used as a reference. Starting with an interpretable model makes it start easier. As illustrated in the exploratory data analysis, to avoid the outliner impact, the paper conducted a modified least squares objective functions that tend to produce more stable forecasts than OLS in the presence of extreme observations. In statistics, Huber loss is a particular loss function that is used widely for robust regression problems where outliers are present that can degrade the performance and accuracy of least-squared-loss error-based regression.  
We done this with the help of the HuberRegressor from sklearn. And there was no tuning process for OLS and OLS-3.

## PCR (Mengyuan CHU)

PCA is extremely valuable for high dimension data, as it helps us to reduce the number of variables that are effectively used to describe the data like in this study. PCR including two-steps:

1. Run PCA on the data to decompose the independent variables into the ‘principal components’, corresponding to removing correlated components. In this step, we used PCA from sklearn, decomposition, and standardize features by removing the mean and scaling to unit variance with the help of sklearn, preprocessing StandardScaler.
2. Select a subset of the principal components and run a regression against the calibration values. Hyperpememter tuning are also needed in this step.

The metrics we use are the mean squared error (MSE) as stated in the Gu et al. (2020). The validation data are then used as an independent set to verify the predictive value of our model and selected the best K (the number of components). Below is an example for tuning in the 1st split.

图表, 折线图

描述已自动生成

Figure 3. Validation for up to 50 principal components (1st Split)

We follow the procedure of the original article, divide our sample into 3 disjoint time periods same as but shorter than the paper. The first training set was used to estimate the model subject to a specific set of tuning parameter values. The second validation sample was used for tuning the hyperparameters.

We did forecasts for data points in the validation sample based on the estimated model from the training data and calculate the MSE based on forecast value from the validation sample, and iteratively search for hyperparameters that optimize the MSE. (K = 0~100 for PCR). Lastly, the testing data was used for evaluating the method’s predictive performance which is, in this paper, to calculate the out of sample R2, Diebold and Mariano test and variable importance.

All the following algorithms below conducted the same tuning scheme.

## PLS (Zhefeng CAO)

The Partial Least Squares regression (PLS) is another method that reduces the variables to a smaller set of predictors. The components obtained from the PLS regression are based on covariance, they explain as well as possible Y. while the components of the PCR are built to describe X as well as possible. This explains why the PLS regression outperforms PCR when the target is strongly correlated with a direction in the data that have a low variance.

For PLS in this study, we tuned the number of components used. In this example, 1 to 10 components.

## eNet (Zhefeng CAO)

Elastic Net combines L1 and L2 (Lasso and Ridge) approaches. As a result, it performs a more efficient smoothing process. We adaptively optimize the tuning parameters, λ and ρ, using the validation sample. In this project, we tune the λ between (1e-4 , 1e-1), and set ρ as 0.5.

## RF (Mengyuan CHU)

Increasing the Depth of individual trees increases the possible number of feature/value combinations that are taken into account. The deeper the tree, the more splits it has and the more information about the data it takes into account. Depth L (1~6) of the trees, are the tuning parameters optimized via validation. Below is an example of tunning for the 1st split and when the feature number = 10.

图表, 折线图

描述已自动生成

## NN (Zhefeng CAO)

The final nonlinear method that we analyze is the artificial neural network. Arguably the most powerful modeling device in machine learning. In our project, we set up a 3 hidden layer neural network, which has three hidden layers with 32, 16, and 8 neurons respectively. As for each node, we use the nonlinear activation function rectified linear unit (ReLU) for faster derivative evaluation.

As for the loss function, we chose L1 loss, and we use Adam SGD method to train this network. In order to speed up the convergence process, we also use the Batch normalization, and set the batch size as 10000 each time.

# ﻿Replication Results

Below please see the figures and tables we replicated with smaller dataset:

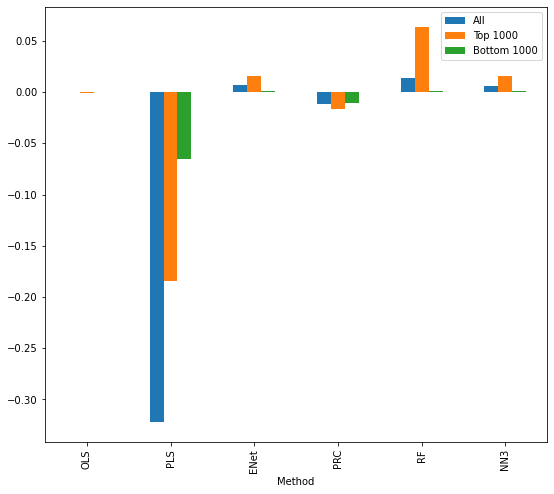


Table 1. Monthly out-of-sample stock-level prediction performance (percentage R2oos)

In this figure, we report monthly R2oos for the entire panel of stocks using OLS with all variables (OLS), PLS, PCR, elastic net (ENet), and neural networks with 3 layers. Because of the device limit, we only use 5-year dataset for training process, so the result is not that good as the original paper represents. But from this figure we could still find that the nonlinear model, like random forest and neural networks model are better than these linear models.

图表, 折线图

描述已自动生成 图表, 折线图

描述已自动生成

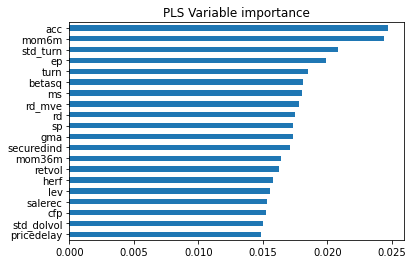
Figure 4. Time-varying model complexity

We did the Time-varying model complexity for PCR and RF from the 1988-1996, RF’s tree depth didn’t change a lot as we only used 5 years of data to train and validate.

Table 3. Comparison of monthly out-of-sample prediction using Diebold-Mariano tests

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | OLS3 | PCR | PLS | ENet | RF | NN3 |
| OLS3 |  | -0.455 | -1.158 | 0.202 | 0.207 | -0.272 |
| PCR |  |  | -1.120 | 1.089 | 0.316 | 0.213 |
| PLS |  |  |  | 1.209 | 1.174 | 1.094 |
| ENet |  |  |  |  | 0.077 | -0.386 |
| RF |  |  |  |  |  | -0.261 |
| NN3 |  |  |  |  |  |  |

This table reports pairwise Diebold-Mariano test statistics comparing the out-of-sample stock-level prediction performance among thirteen models. Positive numbers indicate the column model outperforms the row model. From this table we can see that the nonlinear method RF and NN3 generally outperform the linear models.

图表

描述已自动生成

Figure 5. Variable importance by model

As for evaluating the importance of each characteristic, here we calculated the reduction in R 2 from setting all values of a given predictor to zero within each training sample, and average these into a single importance measure for each predictor. Figure 5 represent the resultant importance of the top-20 stock-level characteristics for PLS model. This result is similar as the original paper. Because of the time limit and for each model we need to train 94 times, so here we only choose PLS model as an example for this part.

# Conclusion& Reflection

In this project, we tried to replicate the Gu et al., 2020, it's much more complicated than we expected. Due to the memory limitation, we only use a subset of the original data and didn’t reproduce the outputs reported in the paper, the process has help us improved the coding ability, especially get some hands-on experience dealing with larger data. We both got better understanding of the whole process of machine learning procedures, also how to interpreter research results. When doing the replication, we have met lots of difficulties including:

1) README files are incomplete or missing, there are some columns that not included in the appendix or README files, we must turn to the CRSP’s data descriptions guide to find the meaning of the data.

2) We can hardly understand the splitting scheme the paper explained, but thanks to the help from Xuantong, a TA of Prof.Yao, we finally got it.

3) To fully replicate the study, large memory is needed, we tried to borrow the sever from other labs but the waiting time was too long to catch the deadline, therefore we used only 5 years of the data for train and validation, and for RF , smaller number of tree numbers to shorten the running time to push ahead the whole project.

4) It is important to read all footnotes, notes under the tables and figures, and appendices which may help us to save more time.

5) We should have an estimation of the total computation power required and availability.

# 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
2. Welch, Ivo, and Amit Goyal. "A comprehensive look at the empirical performance of equity premium prediction." The Review of Financial Studies 21, no. 4 (2008): 1455-1508.