MAFS6010Z Artificial Intelligence in Fintech
Project 2

TANG Tsz Hong 20735194

LAM Chung Wai 20430732

CHAN Koon Lam 20748995

1 Paper replication: Empirical Asset Pricing via Machine Learning

In this paper, we try to replicate the paper "Empirical Asset Pricing via Machine Learning" by Shihao Gu, Bryan Kelly and Dacheng Xiuare. Using recursive evaluation method, OLS, OLS-3, PLS, PCR, ENet, RF, GBRT are investigated. The evaluation between models is made by comparing their out-of-sample performance and variable importance.

2 Data Preprocessing

2.1 Data Description and Collection

The paper investigates the monthly total individual equity returns for all firms listed in NYSE, AMEX, NASDAQ. 60 years of financial data is collected by the authors from 1957 to 2016.

In the dataset, there are 4345508 rows and 101 columns, which contain about 30,000 stocks in 60 years' time. The 101 columns contains 94 firm characteristics predictors, and 7 other information describing the stock data, they are:

- '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 (74 industry sector dummy variables in total)

Meanwhile, there are 8 macroeconomic predictors that are important interacting features for the model. The original idea and concept is from the paper "A Comprehensive Look at the Empirical Performance of Equity Premium Prediction" by Amit Goval & Ivo Welch. Some of them are not provided in the dataset, but we can find them in Amit Goyal's website. Below shows the short form and its corresponding calculation of the 8 predictors.

column name	name	note
dp	Dividend price ratio	column D12
ep	Earnings price ratio	column E12
bm	Book to market ratio	column b/m
ntis	Net equity expansion	/
tbl	Treasury-bill rate	/
tms	Term spread	column lty - column tbl
dfy	Default spread	column BAA - column AAA
svar	Stock variance	/

Notice that in the main data set, there is also Dividend price ratio (no.30, 'dy'), Earnings price ratio (no.33, 'ep'), Book to market ratio (no.9, 'bm'). They have the same name but in the main data the ratio refers to ratio of stock, and here, the macroeconomic one refers to ratio of S&P 500 index.

2.2 Data Cleaning

Dealing with NaN values

In the dataset, 100 out of 104 features (including dependent and target variables) contain missing values. The top 10 proportion of missing values for various features is shown below:

Feature	% of NaN
realestate	75.666619
rd_sale	67.382087
rd_mve	66.823971
secured	63.707949
stdcf	63.682888
rstdacc	63.682888
roavol	53.106357
orgcap	52.140371
grltnoa	52.030096
pchsaleinv	50.961476

As in the paper said, we can fill the missing values by replacing it by the cross-sectional median at each month for each stock, respectively. i.e replace by the median for the data with same i. One important thing to mention is that it is incorrect to replace NA value first and then split into train, valid and test. Because the training set should not rely on the future data in any way.

3 Models

We are going to investigate the OLS with all 920 predictors, OLS-3 (which only chooses size, book-to-market, and momentum as the predictors), partial least squares (PLS), principal component regression (PCR), elastic net (ENet), random forest (RF), gradient boosted regression trees (GBRT). For OLS, ENet, GLM, and GBRT, we introduce Huber loss, which will make the model more robust and perform better.

Due to computational constraint, we sacrifice the implementation of the interaction terms, as $94 \times (8 + 1) + 74 = 920$ covariates are too much for the Recursive Evaluation Method. We fit original 94 covariates and the 8 macroeconomic factors separately for the below analysis. Our objective in this

project is to understand the mathematical concept behind the model. For coding part, we try not to reinvent the wheel. We use the existing package developed by the scholars.

Huber loss

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

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

When ξ tend to 0, huber loss tends to MAE; when ξ tend to ∞, huber loss tends to MSE. It is an approach that let the loss more robust to the outlier compare to the MSE.

Hyperparameters Turning

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.

3.1 OLS, OLS-3

For OLS, OLS-3, we have a slightly different setup compare to the paper. For OLS, because of computation time, we are using the original 94 covariates instead of interacted 920 covariates. Also, instead of using Accelerated Proximal Gradient Method suggested in Appendix B1 (due to unknown reason it is not converging when we try it), we use Stochastic Gradient Descent to solve the OLS with Huber loss and l2 regularization.

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

where $H(x;\xi)$ is the huber loss function.

3.2 PLS

Partial least squares can help reduce the number of predictors. It is useful when the predictors are highly collinear or when the number of predictors is more than the number of observations. PLS performs dimension reduction. The idea is maximizing the covariance of different Zw's, where

$$R = (Z\Omega_K)\theta_K + E$$

where Ω_K is the weight vector contain $w_1, ..., w_K$

3.3 PCR

In the paper, they used the SIMPLS algorithm from "SIMPLS : an alternative approach squares regression to partial least squares regression" by Sijmen tie Jong. We try to do the principal components analysis from python directly, and use a simple linear regression without any regularization term, so that the significance of the PCR can be spot out.

3.4 ENet

Elastic Net is combining ridge and lasso regularization with a mixing parameter α. The penalty function $\phi(\theta;\cdot)$ is

$$\phi(\theta;\lambda,\rho) = \lambda(1-\rho)\sum_{j=1}^{P}|\theta_j| + \frac{1}{2}\lambda\rho\sum_{j=1}^{P}\theta_j^2$$

Clearly from its formation, when $\rho = 0$, it becomes the l_1 penalization (Lasso). When $\rho = 1$, it becomes the l_2 penalization (Ridge).

3.5 RF

Random forest is applying the technique of bagging to tree learners. The main idea is to train multiple decision trees and get the majority voting or averaging to achieve the final prediction. In the paper, they use 300 trees to form Random forest, but it is too costly. We decided to sacrifice some accuracy, reduced to 30 trees. As it is averaging the predictions, the final prediction should not be affected by a lot by reducing the sample. Loss function/ impurity function:

$$H(\theta, C) = \frac{1}{|C|} \sum_{z,t \in C} (r_{i,t+1} - \theta)^2)$$

3.6 GBRT

Gradient boosting regression tree is introduced by Friedman. It is also called gradient boosted machine. GBRT uses gradient boosting. Here we introduce Huber loss into GBRT like to paper did. The idea is to build weak classifier and the next classifier is trained to improve the combination of the weak classifier built before. At last, combine many weak learners become one strong learner.

4 Evaluations

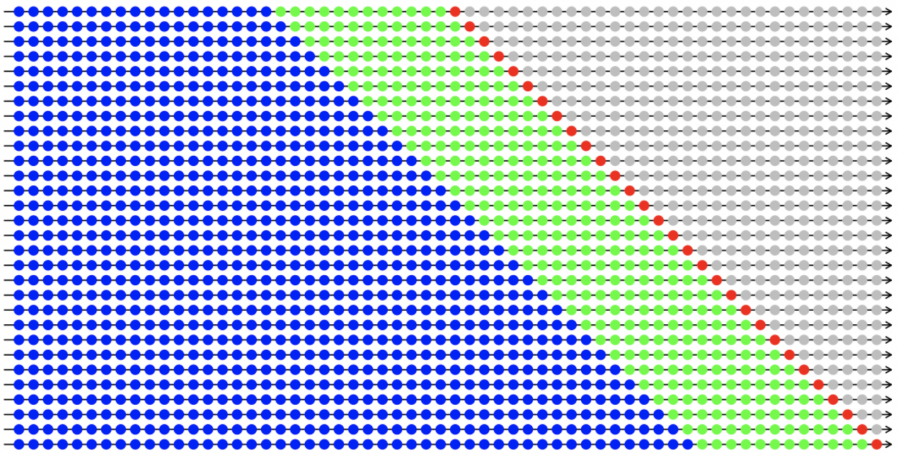
4.1 Methodology

4.1.1 A Recursive Evaluation Method

For the below analysis, we tried to implement the Recursive Evaluation Method. Divide the 60 years of data For example,

- 1st Evaluation:
  - -18 years of training sample (1957-1974)
  - -12 years of validation sample (1975-1986)
  - -1 year of out-of-sample testing (1987)
- 2nd Evaluation:
  - -19 years of training sample (1957-1975)
  - -12 years of validation sample (1976-1987)
  - -1 year of out-of-sample testing (1988)
- ....etc with the last evaluation having 47 years in train set ,12 years in valid set,1 year in test set

With the visualization constructed by the TA shown below, the recursive evaluation method can be easily understood. Blue, green, red dots represent training set, validation set, test set respectively.



4.2 Out-of-sample performance

We use R^2_{oos} compare the out-of-sample stock-level prediction performance. One important point to mention is this out of sample R^2 is different from the usual R^2 that we have learnt before. The R^2_{oos} has its denominator being the sum of squared excess returns without demeaning. This approach is avoiding that the noisiness of historical mean stock return worsen the forecasting performance. The formula is as follows

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

where $\mathcal{T}_\ni$ means the data assessed in the test set.

Because of huge computation time, here we decided to evaluate the final test period only, instead of testing all 30 rolling periods. So there will be 47 year in training set, 12 year in validation set, and 1 year for the test test. The result are shown below.

Table 1: Monthly Out-of-sample Stock-level Prediction Performance (Percentage R^2_{oos})

	OLS +H	OLS-3 +H	PLS	PCR	ENet +H	RF	GBRT +H
All	-0.078	0.7	0.888	0.684	0.569	0.662	0.156
Top1000	-1.641	3.814	4.851	0.663	4.154	0.509	3.169
Bottom1000	0.010	0.183	0.762	-0.12	0.845	0.285	0.164

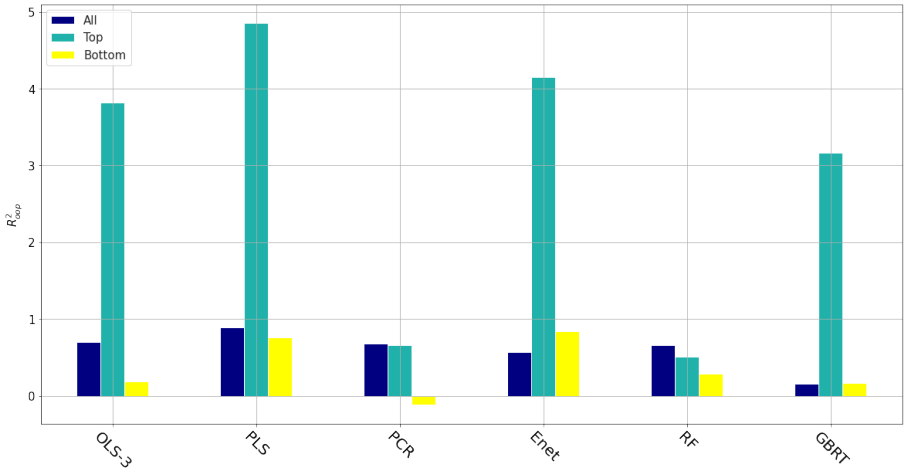


Figure 1: Percentage of NaN values in application train

Our findings are similar to the paper that OLS performs poorly, with negative or R^2 close to zero. Also, it is the same that The models usually predicts the Top 1000 stock data very well.

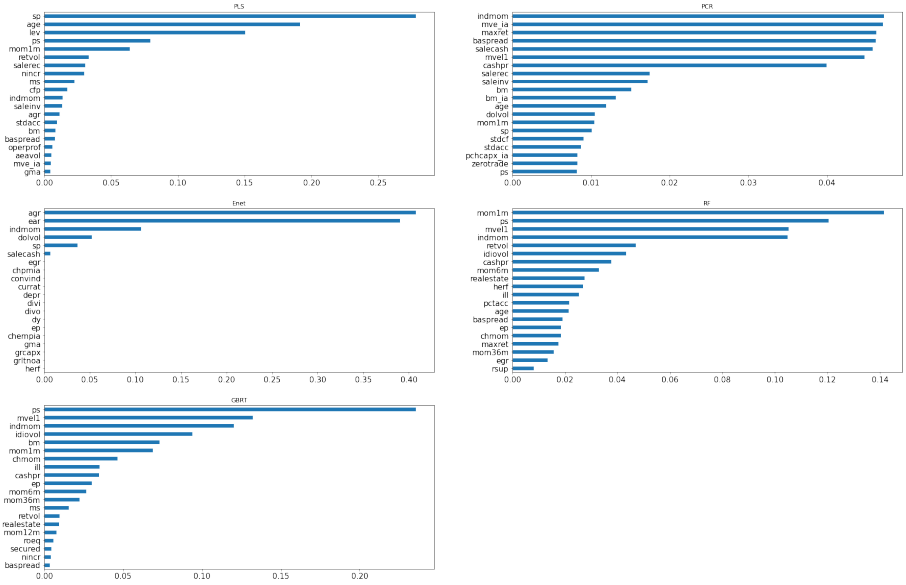
4.3 Variable importance

One of our goal is to analyse the feature importance of different models, we have to understand how to find the feature importance. A common way is to use sum of squared partial derivatives (SSD),

$$SSD_j = \sum_{i,t\in\mathcal{T}_1} \left(\frac{\partial g(z;\theta)}{\partial z_j}\bigg|_{z=zi_t} \right)$$

We analyse the feature importance by model, and by macroeconomic predictors

Figure 2: Variable Importance By Model



In terms of model, mvel1, retvol, and momentum related predictors like mom1m, mom6m are importance features in our setup. Enet is having few predictors because of its penalty. It is a normal case.

Table 2: Variable Importance for Macroeconomic Predictors

	PLS	PCR	ENet+H	RF	GBRT+H
dp	0.003	0.103	0.002	0.055	0.013
ep	0.05	0.102	0.0	0.027	0.006
bm	0.058	0.109	0.047	0.088	0.065
tbl	0.174	0.113	0.265	0.046	0.004
ntis	0.143	0.155	0.162	0.153	0.163
svar	0.285	0.144	0.256	0.393	0.558
tms	0.058	0.137	0.011	0.064	0.01
dfy	0.229	0.137	0.258	0.173	0.182

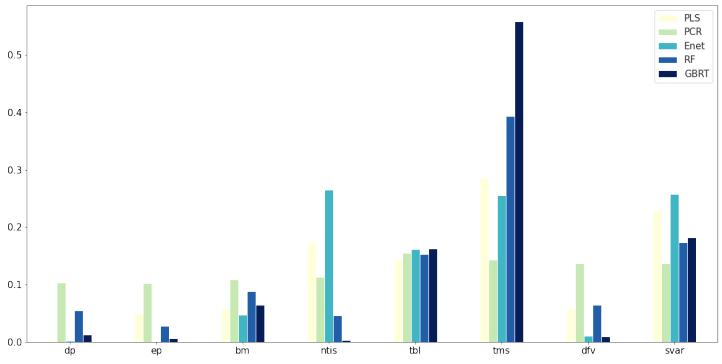


Figure 3: Variable Importance for Macroeconomic Predictors

For macroeconomic predictors, here we fit the 8 macroeconomic predictors into different models to predict the return. Term spread (tms) is the most useful predictor in our setup.

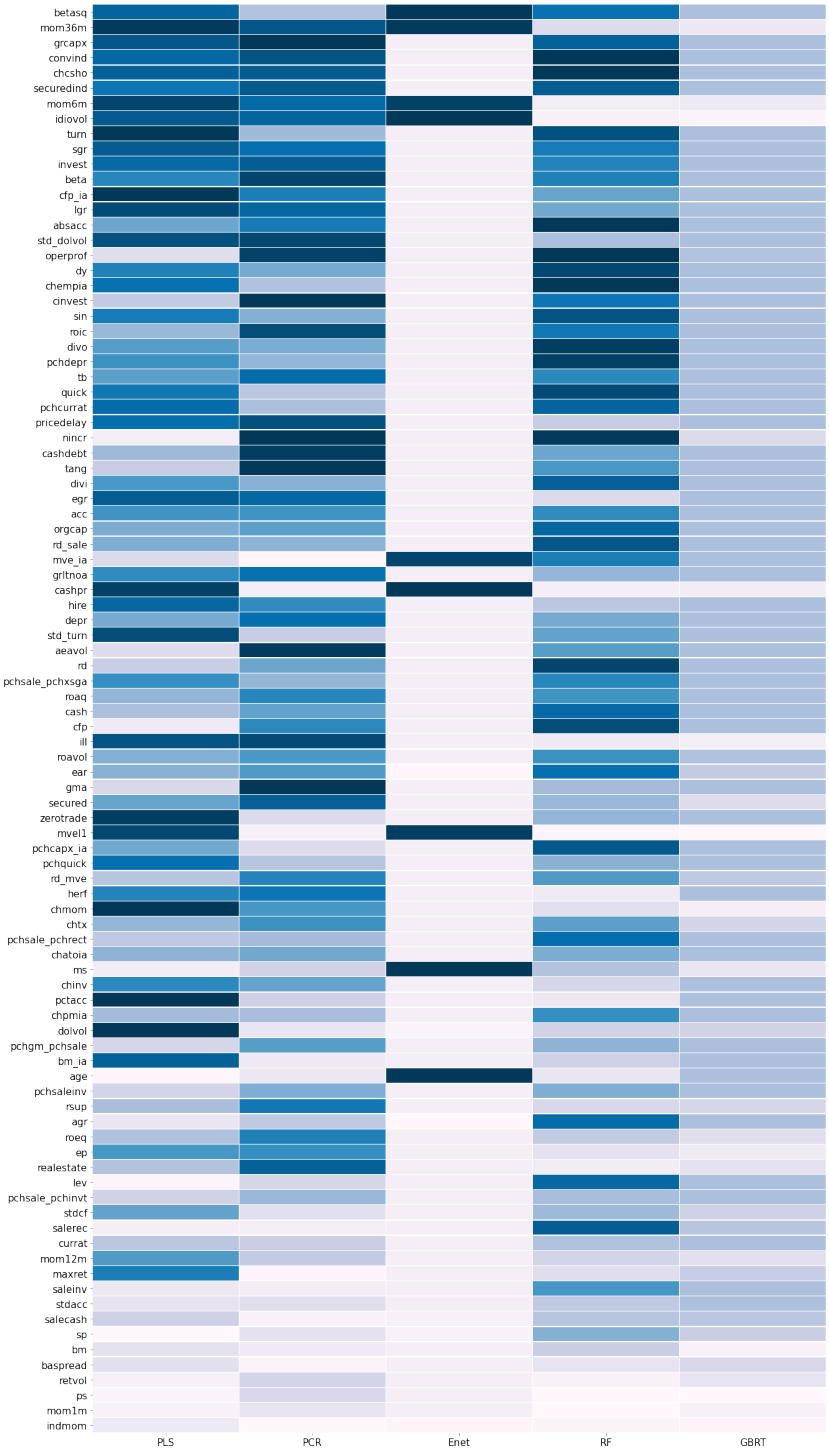


Figure 4: Characteristic Importance

The characteristic importance of different models is constructed. Darker colour represent more influential the feature is. It is sorted by the sum of their ranks over all models. Top 5 characteristic are betasq (Beta squared), mom36m (36-month momentum), grcapx (Growth in capital expenditures), convind (Convertible debt indicator) and chcsho (Change in shares outstanding).

5 Conclusion

One important insight from the paper is it is not true that deeper the tree or deeper the neural network is, the better the model. We agree this statement as when we manually tune the hyperparameter, after some threshold, increase the number of tree do not improve the model.

In terms of replicating the paper, there are still many concept and code we haven't try, like the portfolio forecast in section 3.4. We can dig into them in the future. The field of machine learning is huge. The paper would be a nice piece to start explore.

References

Diebold, Francis X., and Roberto S. Mariano, 1995, Comparing predictive accuracy, Journal of Business & Economic Statistics 13, 134–144.

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.

Welch, Ivo, and Amit Goyal, 2008, A Comprehensive Look at The Empirical Performance of Equity Premium Prediction, Review of Financial Studies 21, 1455–1508.

Appendix

Appendix I : Individual's responsibilities

Table 3: Contribution of groupmate

Name	Student ID	Contribution	Duty
TANG Tsz Hong	20735194	33.33%	Data preparation, Modelling, Writing report
LAM Chung Wai	20430732	33.33%	Data preparation, Modelling, Writing report
CHAN Koon Lam	20748995	33.33%	Data preparation, Modelling, Writing report

Appendix II : A quick summary of the models

column name	name	hyperparameters	amendments we did to reduce the computation time
OLS	huber loss (ξ = 99.9% quantile)	/	we use the 94 covariate only.
OLS-3	huber loss (ξ = 99.9% quantile)	/	we use the 94 covariate only.
PLS	/	K	/
PCR	/	K	/
Enet	huber loss (ξ = 99.9% quantile)	ρ =0.5, $\lambda \in (10^{-4}, 10^{-1})$	/
RF		depth = $1 \sim 6$, trees = 300 feature in each spilt $\in \{3, 5, 10, 20, 30, 50...\}$	no of trees reduced to 30
GBRT	huber loss (ξ = 99.9% quantile)	depth = $1 \sim 2$, trees = $1 \sim 1000$ LR $\in (10^{-4}, 10^{-1})$	/