PAPER REPLICATION: EMPIRICAL ASSET PRICING VIA MACHINE LEARNING

A PREPRINT

SHANG Zhenhang
Department of Mechanical and Aerospace Engineering
Hong Kong University of Science and Technology
Hong Kong, Clear Water Bay, China
zshangab@connect.ust.hk

SUN Lei
Department of Mathematics
Hong Kong University of Science and Technology
Hong Kong, Clear Water Bay, China
lsunak@connect.ust.hk

QUAN Xueyang
Department of Mathematics
Hong Kong University of Science and Technology
Hong Kong, Clear Water Bay, China
xquan@connect.ust.hk

https://github.com/zhshang1221/math5470_Quan_Sun_Shang

ABSTRACT

Measuring the asset premium is a classical empirical asset pricing problem, and the authors conduct a series of analyses of machine learning algorithms on it. For instance, simple linear model, penalized linear model, generalized linear model, dimension reduction techniques, and boosted regression trees. By comparison, the authors also point out the excellent methods (trees), and track the predictive gains to nonlinear predictor interactions. Overall, the machine learning methods improve the evaluation of the risk premia, and their forecasts show more economic gains.

Keywords Machine Learning · Return Prediction · Big Data · Gradient Boosting

1 Introduction

Empirical asset pricing research is gradually becoming a data-driven topic. To better forecast conditional expected stock returns (risk premia), one can combine empirical asset pricing together with machine learning. In the field of empirical asset pricing, it has the following three features. First of all, there are two themes in modern empirical asset pricing research, one is understanding the differences of expected return among various assets, the other is concerning the dynamics of the overall equity risk premium. Secondly, for the risk premium, the set of available conditioning variables is quite large. The third one is the uncertainty of the functional forms from high-dimensional predictors entering the risk premium.

Fortunately, the machine learning is well suited for the above three problem. For risk premium measurement, actually, the risk premium is the conditional expectation of the excess return realized in the future. For complicated prediction problems, dimension reduction techniques reduce the degree of freedom among predictors. For uncertain functional forms, the features of machine learning, for instance, diversity, nonlinear association approximations and parameter penalties, can help handle.

There are various models, in this report, we select six models from the original research paper: simple linear model, penalized linear model, generalized linear model, data reduction methods via principal components regression (PCR) and partial least squares (PLS), boosted trees in regression trees.

This report is organized as the following sections: section 2 introduces the detailed information of the dataset. In the section 3, we describe the fundamental knowledge of six machine learning methods. Then, we will give the results and discussions in section 4. In last section, overall conclusions are stated clearly. All codes in this report are implemented by Python & R.

2 Datasets

The original paper uses the monthly individual equity returns data of US stocks from March 1957 to December 2016 from CRSP, with nearly 30,000 stock samples, an average of 6,200 stocks per month, 94 stock company characteristic factors, and 74 industry dummies (SIC classification standard) and 8 macroeconomic variables.

The dataset is divided into 3 non-adjacent time intervals: from 1957 to 1974, training set (which is used for training data, estimate model); from 1975 to 1986, validation set (which is used to select hyperparameters); from 1987 to 2016, testing set (which is used to evaluate the predictive performance of the model).

Refit the model on a year basis by a rolling window. Which means, after estimating the model by training and validation samples, make the predictions and evaluation on testing dataset in the following year. Then, increase the training samples by one year, and the validation sample window move back one year and maintains the length of 12 years. So back and forth.

3 Methodology

Generally, the excess return of asset can be described as an additive prediction error model:

$$r_{i,t+1} = \mathrm{E}_t\left(r_{i,t+1}\right) + \epsilon_{i,t+1} \tag{1}$$

where

$$\mathrm{E}_t\left(r_{i,t+1}\right) = g^\star\left(z_{i,t}\right) \tag{2}$$

Stocks can be indexed as $i = 1, \cdots, N_t$ and months are indexed by $t = 1, \cdots, N$ (each month has different number of stocks). Our goal is isolating a representation form of $E_t\left(r_{i,t+1}\right)$, which is a function of predictor functions, and has the largest out-of-sample explanatory power for $r_{i,t+1}$. The predictors are P-dim vectors $z_{i,t}$, and g^* is the conditional expected return, which is assumed to be independent of both i and t.

Throughout the whole analysis, define the baseline set of stock-level covariates $z_{i,t}$:

$$z_{i,t} = x_t \otimes c_{i,t} \tag{3}$$

where $z_{i,t}$ represents the features which predict individual stock returns with size $P \times 1$ ($P = P_x P_c$). x_t represents macroeconomic predictors with size $P_x \times 1$, and $c_{i,t}$ represents the characteristics for stock i with size $P_c \times 1$. And the number of covariates is $94 \times (8+1) + 74 = 920$.

3.1 Simple Linear

The simple linear predictive model via ordinary least squares (OLS) is a relative simple method, due to its poor performance, here we list out this, which is just used to compare with other effective methods.

This linear model allows the linearity among raw predictor variables. Then, use this linear relation, together with some parameter vector θ, to approximate the conditional expected return g^*, i.e.,

$$g(z_{i,t};\theta) = z_{i,t}'\theta. \tag{4}$$

By minimizing the following sum of squares of error (called the objective function), we obtain the pooled OLS estimator:

$$\mathcal{L}(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(r_{i,t+1} - g\left(z_{i,t};\theta\right)\right)^2 \tag{5}$$

Since the squares of the error is a quadratic function, when the error is large, due to the convexity of objective 4, $\mathcal{L}(\theta)$ will be abnormally large, and will affect the stability of OLS-based predictions. To alleviate this problem, we use Huber

objective (showm as following), that is, when the error is greater than a threshold, it is changed from square to linear form:

$$\mathcal{L}_H(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} H\left(r_{i,t+1} - g\left(z_{i,t};\theta\right),\xi\right), \tag{6}$$

where

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

3.2 Penalized Linear

When occurring too many predictors, the traditional linear model will overfit noise, and the signal-to-noise ratio of the return prediction is quite low, which leads to poor performance. Under this circumstance, overfitting problem can be avoided by adding a penalty term $\phi(\theta;\cdot)$. Different forms of penalty terms can be added to the linear model 4, for instance, LASSO (when $\rho = 0$), Ridge (when $\rho = 1$), etc. The original paper uses the combination of LASSO and Ridge, that is, the "elastic net" penalty. So the new objective becomes:

$$\mathcal{L}(\theta;\cdot) = \mathcal{L}(\theta) + \phi(\theta;\cdot) \tag{7}$$

$$\phi(\theta;\lambda,\rho) = \lambda(1-\rho)\sum_{j=1}^{P}|\theta_j| + \frac{1}{2}\lambda\rho\sum_{j=1}^{P}\theta_j^2 \tag{8}$$

Then, optimize λ and ρ (both are non-negative hyperparemeters) using validation set. And use the accelerated proximal gradient algorithm to optimize the penalized regression.

3.3 Dimension Reduction via PCR and PLS

Besides the above models, the paper also introduces two dimension reduction models: Principal Component Regression (PCR) and Partial Least Squares (PLS).

For PCR, it firstly uses some low dimensional components to represent high dimensional predictors without loosing much original information, and the obtained first several components are called the principal components. Then, use these leading components in the standard predictive regression. However, PCR is just a transformation in the perspective of predictors, it does not consider the forecasting returns when reducing the dimension.

For PLS, it considers the predictive ability of the predictors. Firstly, PLS estimates the univariate return prediction coefficient (φ_j) using OLS for each predictor j. Then it uses the obtained coefficients as weights to perform a weighted average of all the predictors to obtain the first predictive component. Then, the predictors and target are orthogonalized to the previously obtained components, and repeat this procedure on the orthgonalized set until getting the desired number of PLS components.

Both PCR and PLS considers the vectorized form of linear models (see equations from (1) to (4)). That is rewrite $r_{i,t+1} = z_{i,t}'\theta + \epsilon_{i,t+1}$ as:

$$R = Z\theta + E, \tag{9}$$

where R is a vector of $r_{i,t+1}$ with size $NT \times 1$, Z is a matrix of predictors $z_{i,t}$ with size $NT \times P$, and E is a vector of residuals $\epsilon_{i,t+1}$ with size $NT \times 1$. And the forecasting model is:

$$R = (Z\Omega_K)\theta_K + \tilde{E}, \tag{10}$$

where $Z\Omega_K$ is set which is obtained by reducing the dimension of the original predictor set. If each w_j (with size $P \times 1$) represents the weights to form j^{th} predictive components, Ω_K is a matrix with columns $w_1, w_2, \cdots, w_K$.

For PCR, it chooses the Ω_K recursively. To get the j^{th} component, one solves:

$$w_j = \arg\max_{w} \mathrm{Var}(Zw), \quad \text{s.t.} \quad w'w = 1, \quad \mathrm{Cov}\left(Zw, Zw_l\right) = 0, \quad l = 1, 2, \ldots, j-1 \tag{11}$$

By calculating the singular value decomposition, one can solve the above problem, and the get Ω_K.

For PLS, it searches K linear combinations of Z which has the largest predictive related to the predicted target. To get the j^{th} PLS component, one solves:

$$w_j = \arg\max_{w} \mathrm{Cov}^2(R, Zw), \quad \text{s.t.} \quad w'w = 1, \quad \mathrm{Cov}\left(Zw, Zw_l\right) = 0, \quad l = 1, 2, \ldots, j-1 \tag{12}$$

To solve above, the most famous way is the SIMPLS algorithm of de Jong (1993).

3.4 Generalized Linear

The models introduced above are all parametric models, and the following two are some non-parametric models.A generalized linear model is a spline function that expands the predictors:

$$g(z; \theta, p(\cdot)) = \sum_{j=1}^{P} p\left(z_j\right)' \theta_j \tag{13}$$

where $p(\cdot) = (p_1(\cdot), p_2(\cdot), \ldots, p_K(\cdot))'$ is a vectorized basis functions, and the $K \times N$ matrix $\theta = (\theta_1, \theta_2, \ldots, \theta_N)$ represents the parameters. The paper chooses a spline series of order two: $\left(1, z, (z-c_1)^2, (z-c_2)^2, \ldots, (z-c_{K-2})^2\right)$, with knots: $c_1, c_2, \ldots c_{K-2}$.

This paper uses the objective function in the form of least squares, and tries with the Huber robust correction, and chooses the form of group LASSO for the penalty function:

$$\phi(\theta; \lambda, K) = \lambda \sum_{j=1}^{P} \left(\sum_{k=1}^{K} \theta_{j,k}^2 \right)^{1/2} \tag{14}$$

Embedding equation (14) in equation (7), then use the accelerated proximal gradient descent algorithm to solve this optimization problem. λ and K are two tuning parameters.

3.5 Boosted Regression Trees

To add the interactions between predictors, one way is expanding the generalized model to include multivariate functions of predictors, however, the generalized linear model will be computationally infeasible. Alternatively, regression trees can achieve this. The key idea in the tree model is continuously doing the branch according to (the similarity of) the data, and at each step, a new branch classifies the remaining data from the previous step into bins based on one of the predictors.

The prediction of a tree, $\mathcal{T}$, with K leaves, and L depth is:

$$g\left(z_{i,t}; \theta, K, L\right) = \sum_{k=1}^{K} \theta_k \mathbf{1}_{\{z_{i,t} \in C_k(L)\}} \tag{15}$$

where $C_k(L)$ denotes one of the K partitions of the data. And for every partition, it is a product which has maximal L indicator functions. Within the partition, the constant (θ_k) is sample average outcomes.

The paper adopts the algorithm raised by Breiman et al.(1984) which shows quick convergence on approximately optimal trees. When branching continuously, one needs to improve the "purity" of each branch as much as possible. And the popular l_2 "impurity" (the loss related to the forecast error of a branch C) for branch of the tree is:

$$H(\theta, C) = \frac{1}{|C|} \sum_{z_{i,t} \in C} \left(r_{i,t+1} - \theta\right)^2 \tag{16}$$

where $|C|$ means the number of elements in set C. If C is given, the optimal solution of θ is: $\theta = \frac{1}{|C|} \sum_{z_{i,t} \in C} r_{i,t+1}$. Stop branching if the number of leaves or the depth of the tree achieve the pre-specified threshold. However, trees are one of the prediction methods that have a large tendency to overfit, the "ensemble" method can help improve the robustness. The original paper introduces two regularizers: boosting and bagging. We consider the boosting. It combines the forecasts from over-simplified trees. And the boosting procedure refers to gradient boosted regression trees (GBRT). Three tuing parameters will occur: (L, μ, B), which can be obtained in the validation step.

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

Model	OLS	OLS-3	LASSO	PLS	PCR	GL	Boost Tree
All	-0.12	0.23	0.31	0.31	0.29	0.15	0.44
Top 1000	-1.7	0.4	0.34	-0.12	0.04	0.13	0.55
Bottom 1000	-0.7	0.11	0.23	0.5	0.56	0.3	0.42

4 Results and Discussions

4.1 Performance Evaluation

To evaluate the individual excess return forecasts' predictive performance, one calculates out-of-sample R^2:

$$R^2_{\text{oos}} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3} \left(r_{i,t+1} - \widehat{r}_{i,t+1}\right)^2}{\sum_{(i,t)\in\mathcal{T}_3} r^2_{i,t+1}} \tag{17}$$

where $\mathcal{T}_3$ indicates the out-of-sample testing set, and the subscript i refers to the i^{th} stock. Besides, the paper modifies the Diebold-Mariano (1995) test and gets the revised DM test statistic.

4.2 The Cross Section of Individual Stocks

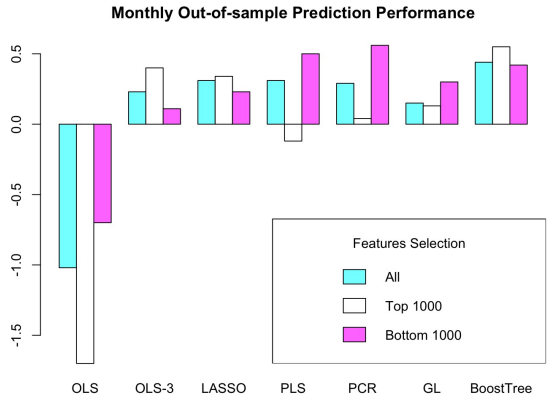


Figure 1: Monthly with OLS

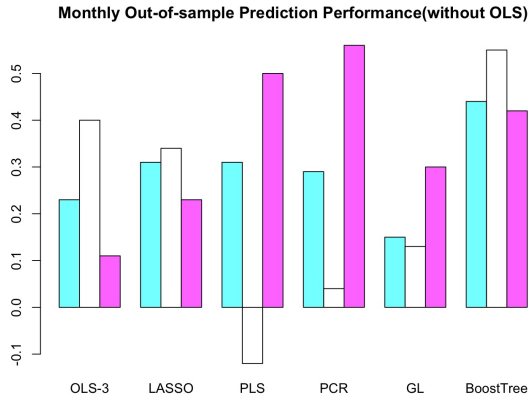


Figure 2: Monthly without OLS

Table 1 compares the machine learning models using the monthly out-of-sample predictive R^2. And in this report, we choose six models to compare: OLS with all covariates, OLS-3 (with only three covariates, including pre-selected size, book-to-market and momentum), penalized linear model (with LASSO penalty), PLS, PCR, generalized linear model with group lasso (GL), and gradient boosted regression trees (BoostTree). Particularly, we use Huber loss for OLS, GL, and BoostTree, which is more robust.

We analyze the first row in Table 1 first, that is the "all" features represented by blue box in figure 1 and 2. This shows the entire sample's R^2_{oos}. The OLS with all covariates gives an R^2_{oos} of -1.02%, which means the OLS has high sensitivity to overfit. In contrast, OLS-3 only uses three covariates, and it improves the R^2_{oos} to 0.23%.

Besides, some penalties can be added to alleviate such problem, for instance, the LASSO has R^2_{oos} 0.31%. From the observation, data reduction via PCR and PLS provide better predictions. PCR has R^2_{oos} 0.29%, and PLS has 0.31%. This phenomena indicates that some characteristics are redundant and probably are noise rather than signals. As for the generalized linear model with group LASSO, it performs even worse than previous purely linear models, GL has R^2_{oos} of 0.15%. The kind of terrible performance is due to the lack of interaction among features, and we can tell this from the high R^2_{oos} 0.44% of boosting tree.

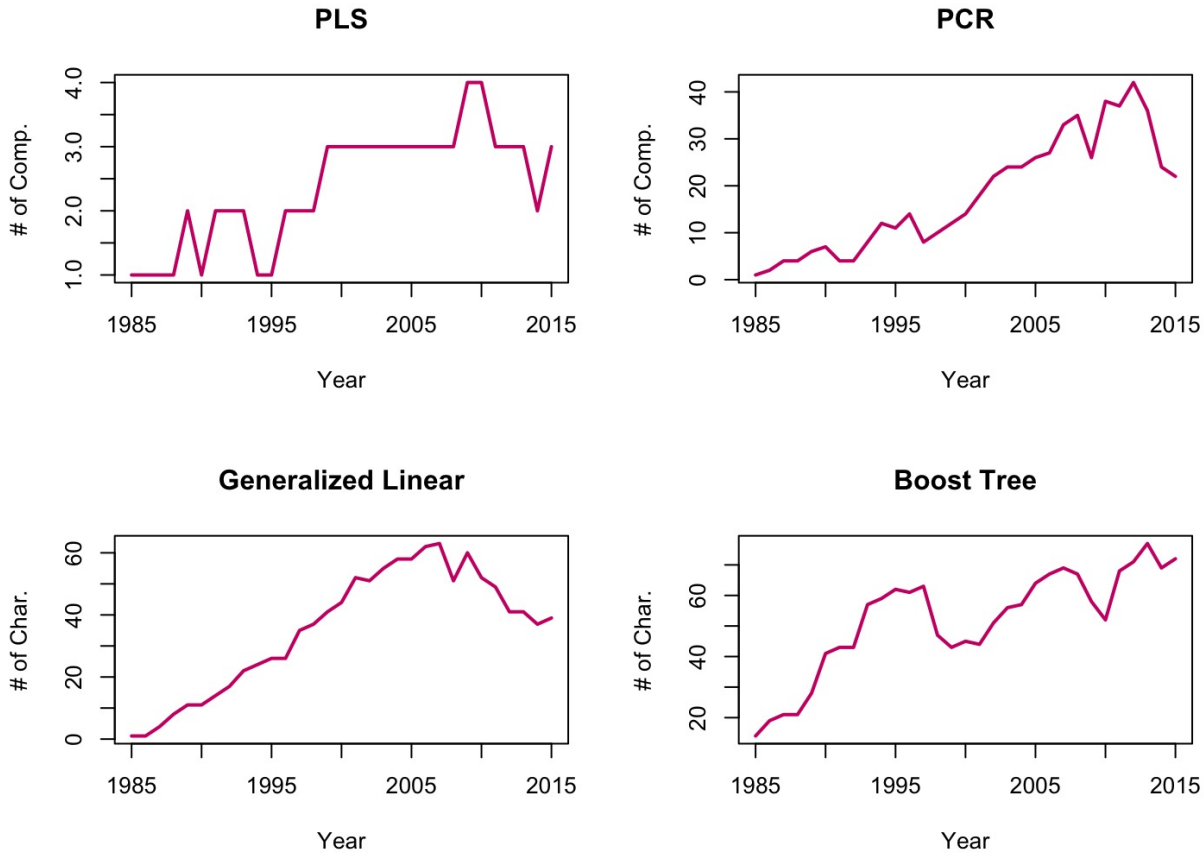


Figure 3: Time-varying Model Complexity

For the second and third row in Table 1, they discribe the predictions of large (the top 1,000 stocks by market equity monthly) and small (the bottom 1,000 monthly) stocks. ("Top 1000" and "Bottom 1000" features represented by white and pink boxes in figure 4 and 5.) As observed, among large stocks, the OLS preforms poorly, the regularized linear models improves a lot from purely linear (R^2_{oos} of approximately 0.1%), and the tree methods are quite successful (R^2_{oos} of 0.55%).

The figure 3 shows the time-varying model complexity for PLS, PCR, generalized linear and boost tree at each re-estimation date. From this figure, we can tell that PLS doesn't find useful components in the early stage, but afterward, it searches up to four compnents. PCR roughly use 20 to 40 components in the prediction. The generalized linear model has the tendecy of selecting more features, however, it doesn't improve the performance (see this from R^2_{OOS} value). As for the boost tree, at the re-estimation point, we count the number of features. Initially, the boost tree uses less than 10 features, afterwards, it needs more than 60 features.

Table 2 describes the annually out-of-sample prediction performance. And the performances among different models are kind of similar to the monthly one, however, the annually R^2_{oos} has quite larger value than monthly one. From this table, it also indicates that the prediction of annual returns are also successful. This concludes that the machine learning method are capable of isolating the risk premia.

4.3 Variable Importance

Previously, in linear regression, the factor analysis can be used to intuitively explain the importance of factors. For machine learning algorithms, there are also some analysis ideas for the importance of factors. The paper introduces two

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

Model	OLS	OLS-3	LASSO	PLS	PCR	GL	Boost Tree
All	-12.07	2.5	2.7	2.93	3.08	2.4	3.61
Top 1000	-26.9	2.44	1.91	1.78	1.55	1.67	4.23
Bottom 1000	-7.55	4.8	4.99	5.27	5.44	4.2	4.61

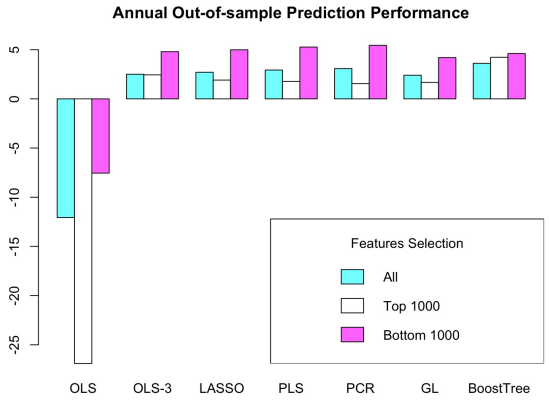


Figure 4: Annually with OLS

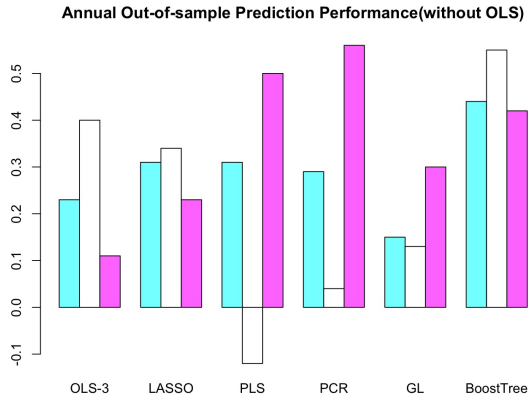


Figure 5: Annually without OLS

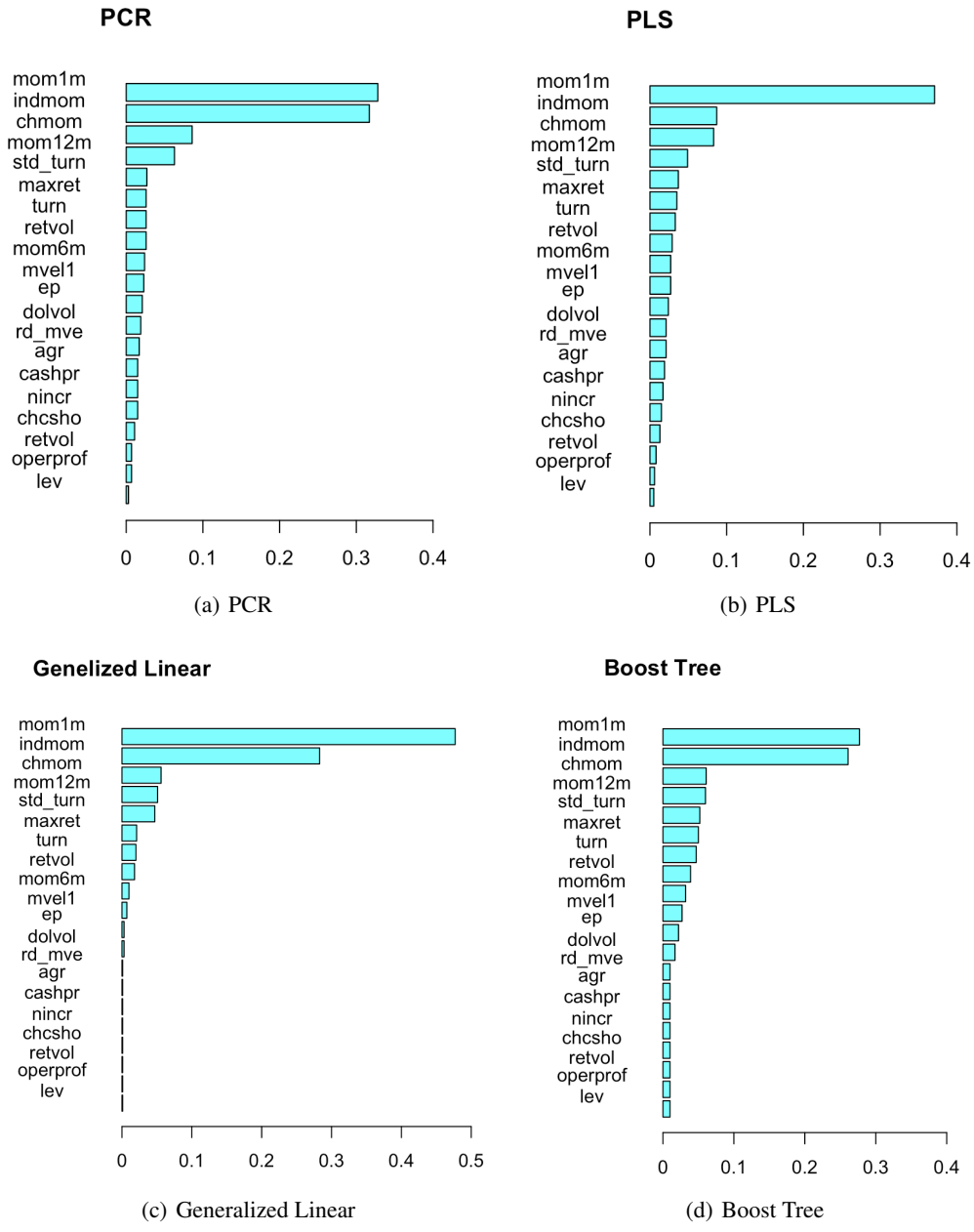
ways: (the variable importance, denoted as VI_j for j^{th} input value)
(1) When keeping all other variables unchanged and setting all values of the variable j to 0, observe the reduction of the panel predictive R^2;
(2) Calculate the sum of squared partial derivatives (SSD) with respect to an input variable j (Dimopoulos, Bourret, and Lek (1995)). Define the j^{th} variable importance in SSD as: $SSD_j = \sum_{i,t\in\mathcal{T}_1} \left(\frac{\partial g(z;\theta)}{\partial z_j}\Big|_{z=z_{i,t}} \right)^2$, where $\mathcal{T}_1$ is the training set, z_j is the j^{th} element of input variable vector. Note that since the tree-based models are not differentiable, the impurity mean decrease is used.

In this section, we use the reduction in R^2 to study the relative importance of variables in each model. Figure 6 shows the order of importance of top 20 stock-level characteristics. All variable importance sum up to one within every model. By comparing the variable importance of four models in figure 6(a), 6(b), 6(c) and 6(d), we can give a rough ranking for the importance, which consists of four categories:
(1) Recent price trends. Like, short-term reversal (mom1m), industry momentum (indmom), stock momentum (mom12m), momentum change (chmom), and recent maximum return (maxret).
(2) Liquidity. For example, turnover & turnover volatility (turn, std turn), and log market equity (mvel1), dollar volume (dolvol).
(3) Risk measures. For instance, the total and idiosyncratic return volatility (retvol, idiovol).
(4) Valuation ratios and fundamental signals. Including earnings-to-price (ep), sales-toprice (sp), and asset growth (agr).

Besides, we also find that for dimension reduction models, they prefer to include the momentum and reversal features, but for boost tree, it is more comprehensive, which includes a wider set of characteristics.
(Of course we also can use SSD to measure the variable importance.)

4.4 Conclusion

In this project, our aim is to replicate the paper "Empirical Asset Pricing via Machine Learning", and we successfully done it. In original paper, it has more than 10 models are used to compare with each other, we select six of them in this report, which are : simple linear model (OLS, OLS-3), penalized linear model (LASSO), generalized linear model (GL), data reduction methods via principal components regression (PCR) and partial least squares (PLS), and boosted trees.



(a) PCR

(b) PLS

(c) Generalized Linear

(d) Boost Tree

Figure 6: Variable Importance By Model

After the comparison, we find that to some extent, the tree models provide the best prediction performance. OLS-3 term and GL considering factor interaction performs better than traditional OLS, while nonlinear model is better than linear model, particularly the boost tree model. And we draw almost the same conclusion from our experiments as the original paper: the most powerful predictors are related to the price trends including reversal and momentum, while the fewer important predictors are liquidity, etc. Finally, by replicating the paper, we also understand the importance of machine learning in the field of fintech industry.

4.5 Contribution

- SHANG Zhenhang
  - Code in python for PLS, PCR, Generalized Linear replication and visualization.
  - Write PPT
  - Presentation
- SUN Lei
  - Code in python for OLS, OLS-3, Penalized Linear and Boost Tree replications and the integrate all the replicated model.
  - Write PPT
  - Presentation
- QUAN Xueyang
  - Write report
  - Write PPT
  - Presentation

References

[1] Shihao Gu, Bryan Kelly and Dacheng Xiu (2020). Empirical Asset Pricing via Machine Learning, *The Review of Financial Studies*, vol.33, no.5, 2223-2273.

[2] de Jong, Sijmen (1993). Simpls: An Alternative Approach to Partial Least Squares Regression, *Chemometrics and Intelligent Laboratory Systems* 18, 251-263.

[3] Diebold, Francis X., and Roberto S. Mariano (1955). Comparing Predictive Accuracy, *Journal of Business & Economic Statistics* 13, 134-144.

[4] Breima Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen (1984). *Classification and regression trees*, (CRC press).

[5] B¨uhlmann, Peter, and Bin Yu (2003). Boosting with the l2 loss, *Journal of the American Statistical Association* 98, 324-339.

[6] Fama, Eugene F, and Kenneth R French (1993). Common risk factors in the returns on stocks and bonds, *Journal of financial economics* 33, 3-56.