**Final Project. Factors War and Machine Learning**

**Factors War**

A fundamental question in asset pricing is to find factors to predict and interpret stock returns. In the 1960s, the seminal CAPM model points out that the market return explains individual stock returns:

Fama-French (1992) finds firm size (market equity) and book-to-market ratio also predicts stock return. Correspondingly, they add the Small Minus Big (SMB) factor and High Minus Low (HML) factor to the model:

Specifically, they chop the firms by their market capitalization, and independently sort the firms by their book-to-market ratio to form three groups (Low, Medium, and High) based on the 30% and 70% break-points. SMB is the return difference between the simple average of 3 small portfolios and 3 large portfolios, and HML is difference between the 2 value portfolios and 2 growth portfolios.

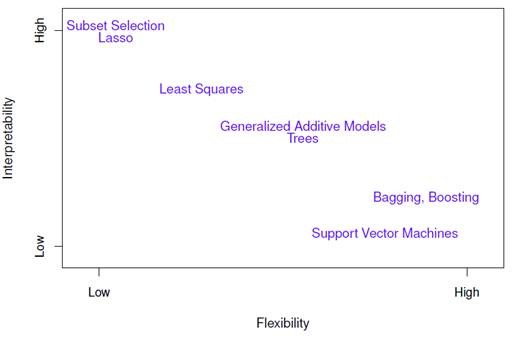
Hundreds of pricing factors have been proposed since then. However, Hou, Xue, and Zhang (2018) replicated more than 300 factors and find most of them are not robust under various types of portfolio construction setups. On the other hand, Harvey, Liu, and Zhu (2016), questioned the “data mining” practices and asks for higher t-statistics for future discovered factors.[[1]](#footnote-1) Furthermore, the commonality (similarity) of the factors is also a serious problem in considering the economic meaning of the factors.

Our goal is to use machine learning techniques to extract the interaction term of factors, as well as find higher order conjunctions of the existing factors.[[2]](#footnote-2)

**The LASSO**

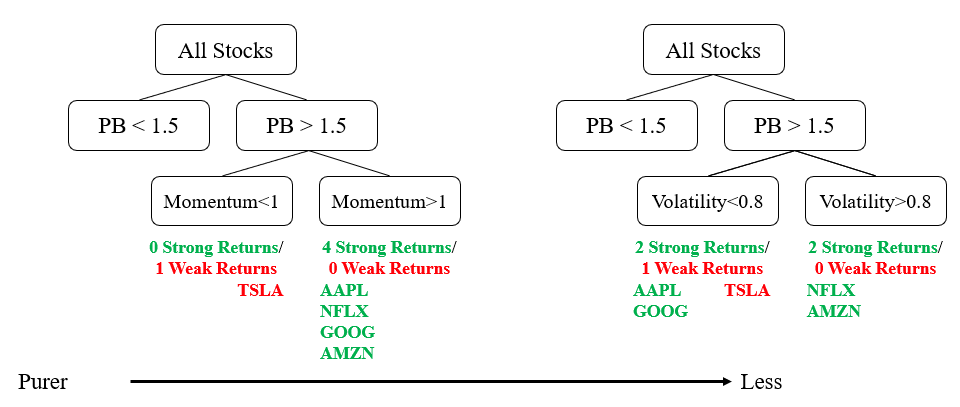
As the baseline machine learning algorithm, the LASSO (least absolute shrinkage and selection operator) revises OLS in adding a penalty term (L1-norm) to eliminate unimportant/redundant predictors:

The regularization can alleviate overfitting and, more importantly, perform *variable selection*. The larger , the heavier penalty on non-zero . Thus, LASSO will select most important (significant) variables to explain the left-hand side variable, which is the source of its interpretability.



**Decision Tree/Bagging/Random Forest**

Decision Trees focuses on the interaction terms of the covariates by construction:



Bagging is simply use subsets of data to run decision trees and average all results when predicting, in order to alleviate the overfitting problem. Random forest, different with the bagging, subsets both data and features. This avoids a small subset of strong predictors dominates all trees. In practice, random forest usually has the best performance.

**Instructions**

We have generated a fully COMPUSTAT/CRSP monthly merged table for all US stocks that have ever listed on NYSE, NASDAQ, or AMEX in 1967-2018. The columns include all Fama-French firm characteristics (size, B/M ratio, etc) and other ~50 factors. Fred and I calculated those factors following Hou, Xue, and Zhang (2018)’s Appendix A.[[3]](#footnote-3) We followed the naming rule of the characteristics/columns, so you can map the variables with the paper correspondingly. The last columns are realized returns of stocks of the record month. lead(return) is the left-hand side variable for the models to fit to, i.e. the variable that we are predicting. DO NOT use them as model input.

The data is very large, and our model is also computation intense. Therefore, supercomputer is highly recommended. You are free to choose the parallelization methods we covered in the project 1. Please use your pylon5 folder as the working directory, because /home/ is small.

(3 points) Understand and clean the data.

The training sample is available in [dropbox](https://www.dropbox.com/s/cqtdwk2uui3i5qi/features_training.Rdata?dl=0) and supercomputer /pylon5/se5phip/slid/project1/data/ folder. It has 1.5 million firm-months of records from 1972 to 2002, and 55 columns of basic information and regressors. Columns 1-16 are fundamental information, and 17-20 are price, volume, shares outstanding, and lagged market equity. Column 21-24 are Fama-French’s Book-to-Market ratio, Asness-Frazinni’s Book-to-Market ratio, Fama-French (2018) added momentum factor, and Market Equity of last December, respectively. Column 25-53 are our replicated factors following Hou, Xue, and Zhang (2018)’s Appendix A. All of these numbers have been properly lagged, i.e. was available at the *end* of the yyyymm.[[4]](#footnote-4) Therefore, when performing a back test, it is safe to say, “according to these numbers, I want to buy the stocks at the beginning of next month.” However, column 55 is the stock return of *this* month (not lagged).

For every single stock (group by PERMNO), the first thing you need to do is sort it by yyyymm. Then you’ll need to lead the $RET by one period. lead(return) is the object to fit on.

(2 points) First run a simple regression using FF factors only. Do they predict stock returns?

(8 points) Construct models to optimize in-sample-performance given all the data. For every single month, sort the characteristics and construct large, medium, and small dummies. Construct interaction terms of these dummies. Apply the LASSO, decision trees, or even more complicated models on the firm characteristic dummies and their interaction terms. Computation difficulty could be a serious concern, so you may want to train your model in a small subset of the data, e.g. random pick some PERMNO. The better your model perform in the sample, the higher your score in this part. Using future data will lead to significant penalty, though.

(6 points) Out of sample test

Your final model must be a function that takes the first 54 columns of one line of data, and outputs its predicted return. Directly send the code to me, because I will test it in the truncated sample 2003-2018. Also, try to make some economic argument about why your model makes sense. DO NOT USE FUTURE DATA.[[5]](#footnote-5) The better your model performs, the higher score from this part.

Your submission will consist of:

1. A report of your simple regression results, explanation of your full model, and your full model’s predicting power, i.e. calculate .
2. A code file (just a function takes 54 variables in one line of data except the $RET, and outputs a number). You should test the function on the training sample to make sure it works.

(5 points) Grade your peers. Send an individual email of the grade you give to your peers to Fred.

1. In short, if one performs a simple test and get t=1.96, the probability of type I error is 0.05. When one tries, say, 100 possible models, the joint probability of type I error is significantly higher. If we want to keep the overall p-value at the 0.05 level, we need to set the threshold of significance at p=0.05/100=0.0005, i.e. t=3.48. This adjustment is called Bonferroni correction. [↑](#footnote-ref-1)
2. In this document, we use the following terms interchangeably: independent/right-hand side variable, feature, covariate, predictor, and “x”. All of them means the factors that predict return. [↑](#footnote-ref-2)
3. We spent weeks in constructing a unified framework to merge, clean, and compute the characteristics. However, errors and mistakes can’t be avoided. Please do not hesitate to email me at [sidali3@illinois.edu](mailto:sidali3@illinois.edu) if you have any question or find anything inconsistent. [↑](#footnote-ref-3)
4. Though different paper uses different methods to lag the variables, we universally follow the “best available data” rule, that we lag all earning/fundamental data by 4 months, and use the last month’s close price to calculate market equity. [↑](#footnote-ref-4)
5. However, we do not forbid data-mining at this point, as some seemingly data mining behaviors may have deep underlying economic mechanism. A famous critique of the asset pricing factors is the construction of “A-minus-Z”, i.e. stock names start with “A” outperforms those with “Z”. You can google this case for more details. Of course, you should not say things like “buy if (TICKER==AMZN)” or “sell if (yyyymm==200701)” in your code. I will read the code and penalize time travelers. [↑](#footnote-ref-5)