MAFS 6010Z Artificial Intelligence in AI : Project 2 – Empirical Asset Pricing

Wong Hoi Ming (20641276) / WONG Sik Tsun (20038819)

1. Introduction

In this project, we attempted to study how different machine learning methods can be applied to asset pricing defined by stock risk premium. We followed the similar approach by Dacheng Xiu[1] (2020) and performed a comparative analysis of 6 methods, namely PCR, PLS, elastic nets, random forests, gradient boosting trees and neural networks.

2. Data Processing & Modelling Details

What we did and followed Xiu's paper

- **Constructing the predictors**: we constructed the same predictors by selecting the same stock-level predictive characteristics, creating the industry dummies as well as merging the macroeconomic predictors as the paper suggested.
- **Avoiding forward-looking bias**: we adopted the same approach to rearrange the predictors by assuming monthly, quarterly and annual data are delayed by at most 1 month, 4 months and 6 months respectively.
- **Handling missing values**: we replace missing values by the cross-section median. For any cross-sections where the values of a predictor are missing for all stocks, we fill those with zeros.
- **Cross-section demeaning, scaling and batch normalization** for neural networks (NNs): This is to restore the representation power of the unit and potentially give a faster convergence in training[2]
- **Regularization & algorithm specification**: we adopted the same regularization and algorithms in training models, such as the learning rates, L1/L2 penalty and Huber losses for elastic nets and "Adam" for NNs
- **Hyperparameters tuning**: we performed hyperparameters tuning for some of the models based on the potential values suggested in the paper as well as the recursive performance evaluation scheme.

2. Data Processing & Modelling Details (Continued)

What we did and followed Xiu's paper (continued)

- **Recursive performance evaluation scheme**: we adopted the same evaluation method, which retains the entire history in the training windows and gradually include more recent observations.

How we differed from Xiu's paper and why

- **Hyperparameters tuning**: For some machine learning methods, we omitted hyperparameter tuning (ie. Elastic Nets, NNs) when we enlarge the training sample, due to limited computing power. In such cases, we stick to the hyperparameters that performed reasonably well in the first few training samples and assume they will be steady over time
- **Omitted ensembling for NNs** due to limited computing power: The resultant NNs trained may vary on different attempts with different initial points on SGD algorithms
- **Omitted explicit polynomial interaction terms** owing to insufficient memory space. However, the interactions terms have been implicitly included in the tree-based methods and neural networks.

3. Results & Analysis (Detailed Charts & Tables in Appendix)

30 training samples – Models with hyperparameter tuning (except ELN)

R squared	Elastic Net	PCR	PLS	RF	GBT
Mean	-0.11970	-0.02327	-0.00062	0.03419	0.02385
Median	-0.02773	-0.01349	-0.00221	0.04112	0.00847
Max	0.17426	0.09755	0.05401	0.16573	0.16743

First 10 training samples – Models without hyperparameter tuning

R squared	NN1	NN2	NN3	NN4	NN5
Mean	0.02021	0.03631	-0.03688	-0.03417	-0.01776
Median	0.00263	0.04551	0.00628	0.00568	0.01020
Max	0.14373	0.14403	0.05975	0.06414	0.15894

3. Results & Analysis (Continued)

- **Non-linear models outperformed**: The table shows the out-of-sample R-squared in predicting monthly stock risk premiums using our models. RF, GBT and NN tended to outperform given their higher flexibility.
- **R squared**: The R squared figures are low for dimension reduction or linear models (lower than the paper's), probably due to lack of interaction terms, while our tree-based methods worked better.
- **Hyperparameter tuning is pivotal**: Predictive power of NNs deteriorate as we move along the timeline, while other non-linear models RF and GBT remain largely steady. This highlighted the importance of hyperparameter as financial markets could evolve, so the degree of model flexibility required will vary. Earlier training samples indicate overall better performance than linear models. (See appendix)
- **Feature importance**: Similar to the paper's, the distribution of the feature importance is highly skewed towards a few predictors for dimension reduction models, while RF and NN are more democratic, drawing predictive information. The predictors of high importance are different in our results. For example, share price momentum plays a less important role in ours.
- **In NNs, shallow learning seemed to outperform deeper learning**. Neural network performance peaks at three hidden layers in the paper, and similarly ours peaks at two layers.

4. Discussions on ways for improvement

- **Incorporate interaction terms:** While tree-based methods and networks have implicitly considered the interactions among the covariates, this is lacking in linear models. To balance the consumption of memory usage and model accuracy, it is possible to do feature selection before computing interaction terms.
- **Hyperparameters tuning in ELN, NNs** by using the R-squared in the validation set: This can be done in every training sample, or every 5 years' forward step.
- **Fitting models with recent data while dropping the old ones** to ensure time relevance of financial data.

5. References & Citations (Please refer to appendix)

Appendix

All 30 testing samples – Models with hyperparameter tuning (*except for Elastic Net)

R-squared	Elastic Net*	PCR	PLS	RF	GBT
Mean	-0.11970	-0.02327	-0.00062	0.03419	0.02385
Median	-0.02773	-0.01349	-0.00221	0.04112	0.00847
Max	0.17426	0.09755	0.05401	0.16573	0.16743

First 10 testing samples – Without hyperparameter tuning

R-squared	NN1	NN2	NN3	NN4	NN5
Mean	0.02021	0.03631	-0.03688	-0.03417	0.01776
Median	0.00263	0.04551	0.00628	0.00568	0.01020
Max	0.14373	0.14403	0.05975	0.06414	0.15894

All 30 testing samples – Without hyperparameter tuning

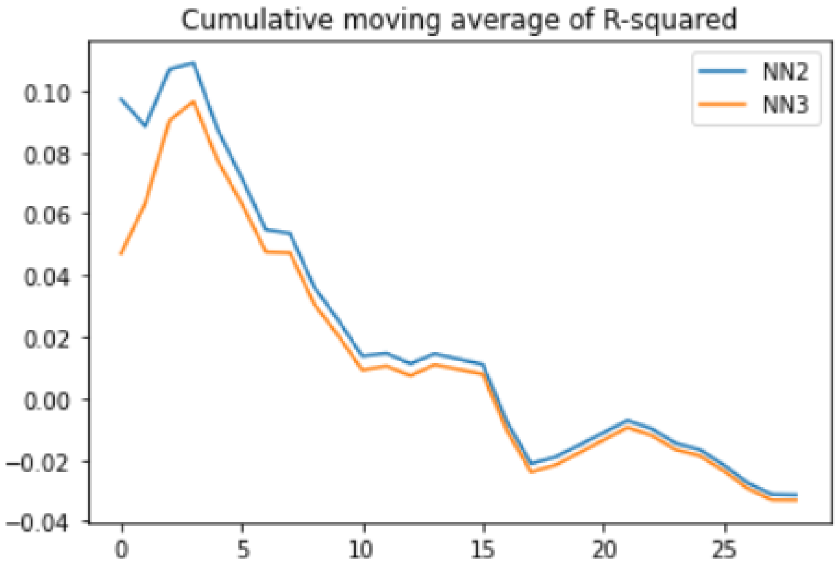
R square	NN1	NN2	NN3	NN4	NN5
Mean	-0.02696	-0.03139	-0.05861	-0.05410	-0.03039
Median	-0.02004	-0.01459	-0.02608	-0.02777	-0.00569
Max	0.14373	0.14403	0.07488	0.10840	0.15894

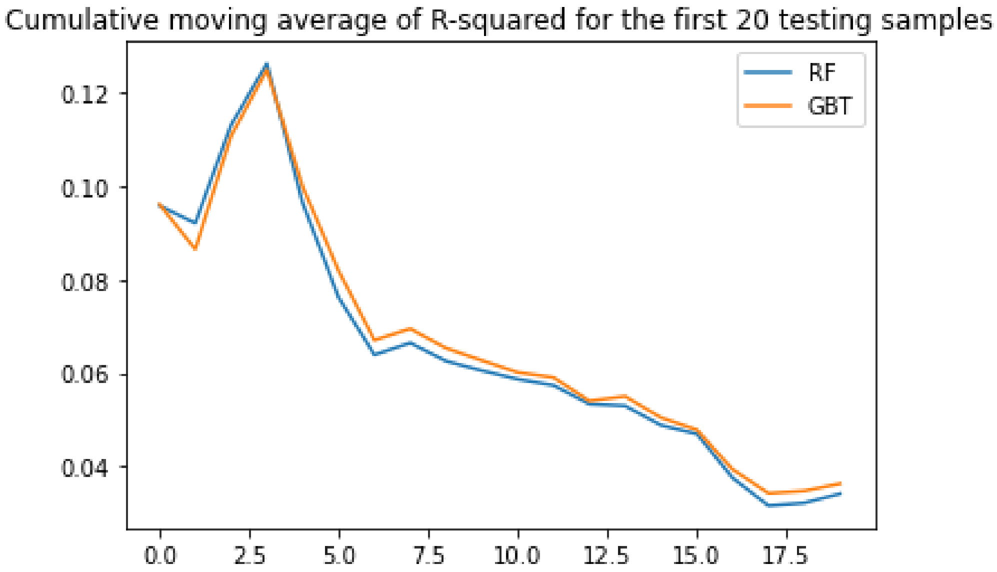
5. References & Citations

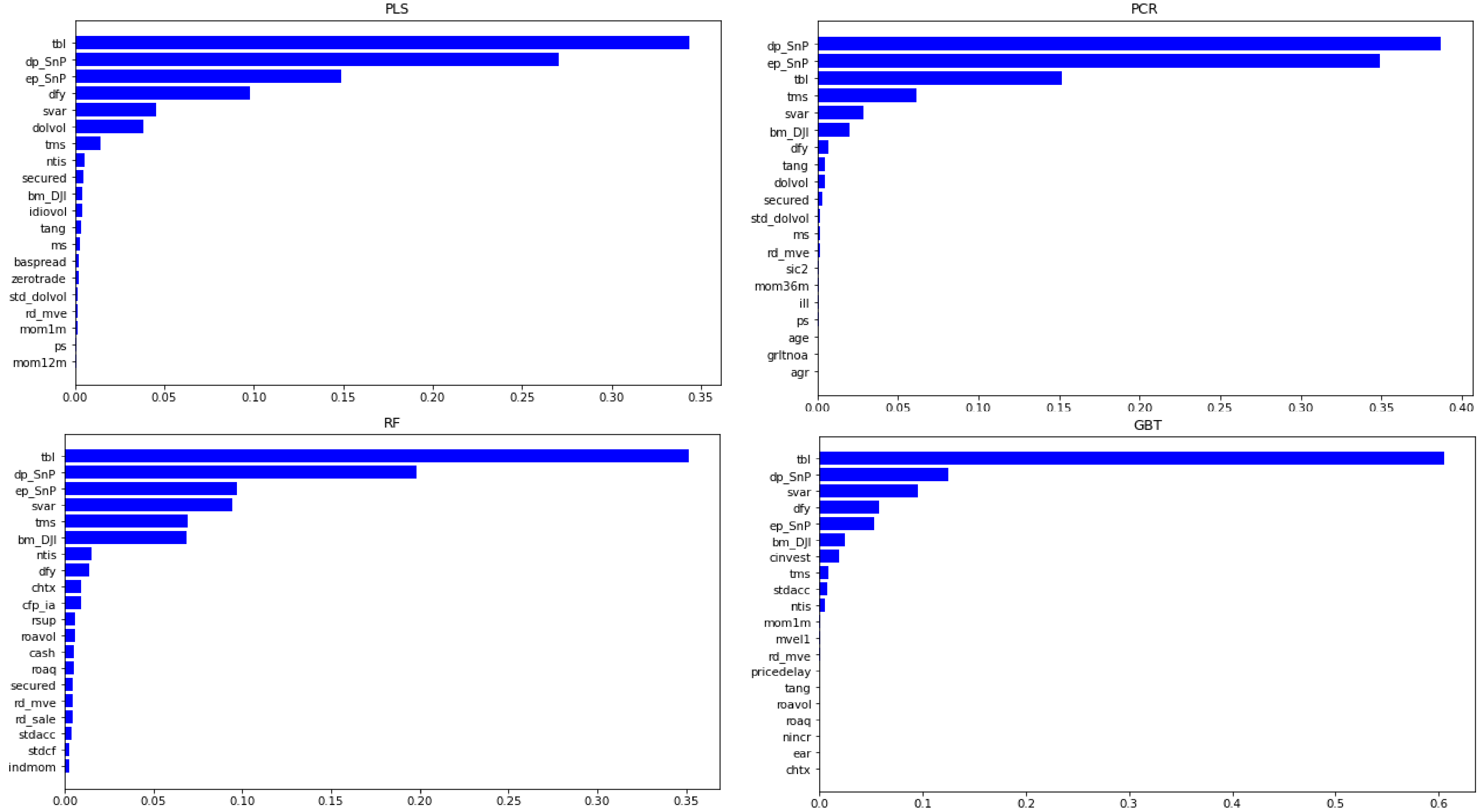
1. **Shihao Gu, Bryan Kelly and Dacheng Xiu**
"Empirical Asset Pricing via Machine Learning", Review of Financial Studies, Vol. 33, Issue 5, (2020), 2223-2273
https://dachxiu.chicagobooth.edu/download/ML.pdf

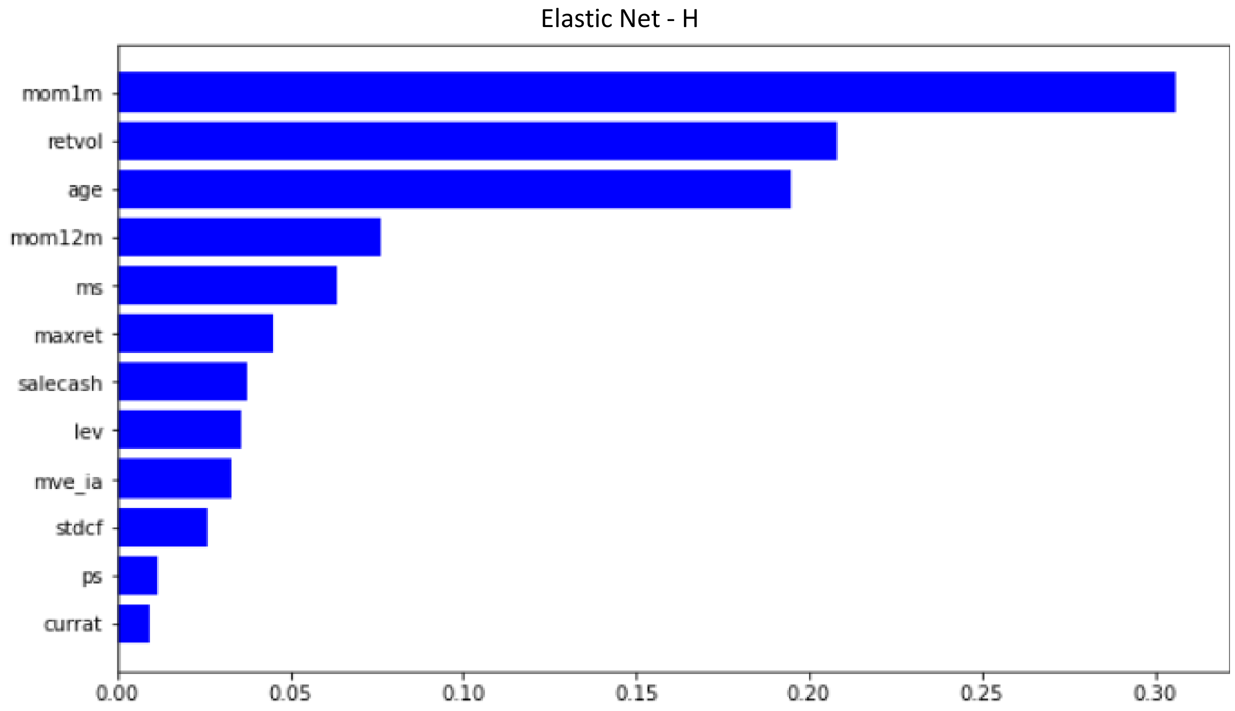
2. **Sergey Ioffe and Christian Szegedy**
"Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", (2015)
https://arxiv.org/abs/1502.03167

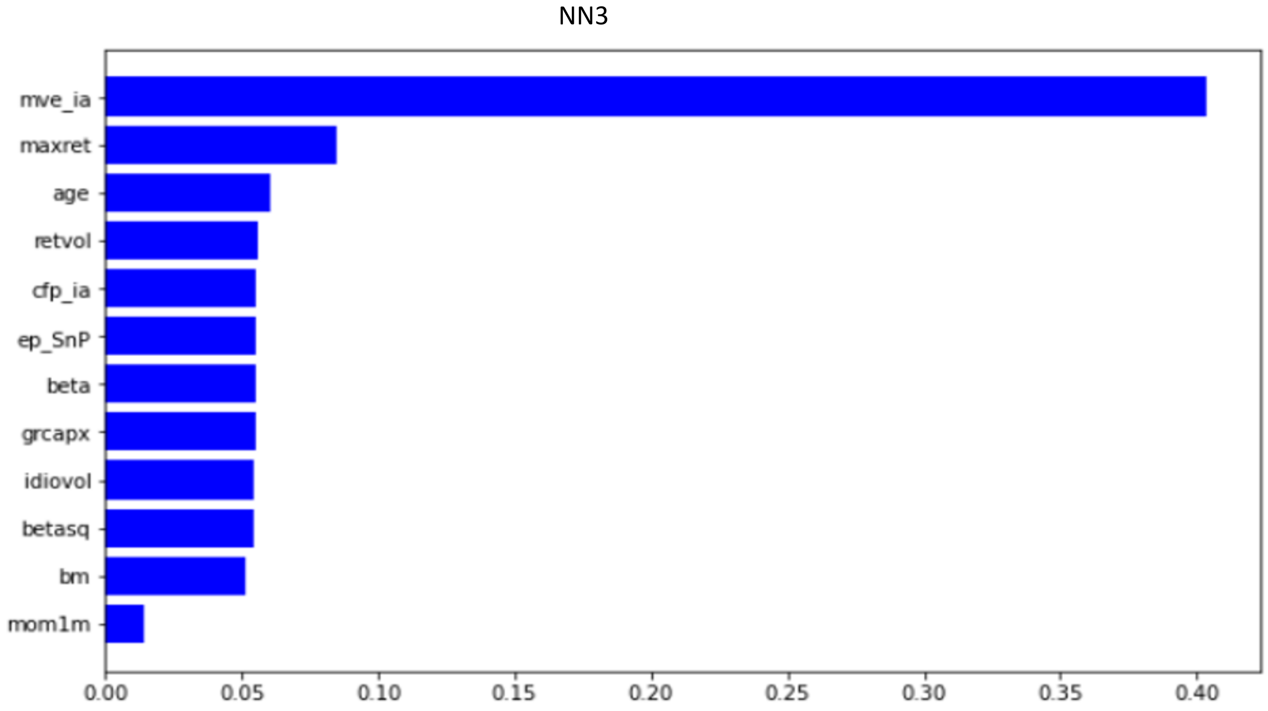
3. **Ivo Welch and Amit Goyal**
"A Comprehensive Look at The Empirical Performance of Equity Premium Prediction" (2008)
https://drive.google.com/file/d/1uvjBJ9D09T0_sp7kQppWpD-xelJ0KQhc/view






PLS
tbl
dp_SnP
ep_SnP
dfy
svar
dolvol
tms
ntis
secured
bm_DJI
idiovol
tang
ms
baspread
zerotrade
std_dolvol
rd_mve
mom1m
ps
mom12m
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35
PCR
dp_SnP
ep_SnP
tbl
tms
svar
bm_DJI
dfy
tang
dolvol
secured
std_dolvol
ms
rd_mve
sic2
mom36m
ill
ps
age
grltnoa
agr
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40
RF
tbl
dp_SnP
ep_SnP
svar
tms
bm_DJI
ntis
dfy
chtx
cfp_ia
rsup
roavol
cash
roaq
secured
rd_mve
rd_sale
stdacc
stdcf
indmom
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35
GBT
tbl
dp_SnP
svar
dfy
ep_SnP
bm_DJI
cinvest
tms
stdacc
ntis
mom1m
mvel1
rd_mve
pricedelay
tang
roavol
roaq
nincr
ear
chtx
0.0 0.1 0.2 0.3 0.4 0.5 0.6


Elastic Net - H
mom1m
retvol
age
mom12m
ms
maxret
salecash
lev
mve_ia
stdcf
ps
currat
0.00
0.05
0.10
0.15
0.20
0.25
0.30


NN3
mve_ia
maxret
age
retvol
cfp_ia
ep_SnP
beta
grcapx
idiovol
betasq
bm
mom1m
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40