Evaluation of Machine Learning in Finance

Ze Yu Zhong

Supervisor: David Frazier

Monash University

Overview

Main Motivation

To evaluate the application of machine learning to predicting financial asset returns, with specific regard to how they deal with the unique challenges present in financial data via simulation.

Motivation

Empirical finance is typically concerned with two main goals:

- Prediction accuracy - want to predict future returns!
- Statistical causal inference - want to understand *what* drives returns

Motivation

Recognise that these goals are very difficult, especially for traditional regression!

The underlying data generating process for financial data is unknown - returns are estimated by *risk factors*, defined by Harvey et al. (2016) as a collection of regressors that can proxy for underlying risk factors

Factors are often unsuitable for regression:

- Persistent
- Cross-sectionally correlated (multicollinearity)
- Non-stationary
- Pre-known, and therefore little time series variation

Motivation

Consequences when included in traditional regression well documented:

- Biased t-stats
- High variance for coefficient estimates
- Unstable coefficient estimates

These can adversely affect statistical inference.

The estimated coefficients, due to being imprecise, will also result in poor out of sample prediction, which are a function of coefficients. This is particularly so if the multicollinearity between regressors changes over time, which is likely in financial data

Background - Dividend Ratio Example

- Included due to good in sample performance in the 1990s (Goyal and Welch, 2003)
- *Persistent* (Goetzmann and Jorion (1993), Ang and Bekaert (2006))
  - Correlated with lagged dependent variables on the right hand side of the regression equation.
  - Violates assumptions of independent regressors of OLS: t stats are biased upwards due to autocorrelated errors
  - GMM and NW errors corrections are also biased, (Goetzmann and Jorion, 1993)
- Not robust and have poor out of sample performance since 2000s (Goyal and Welch (2003), Lettau and Ludvigson (2001), Schwert (2003))

Dividend Ratio Example

- Factors such as dividend ratios, earnings price ratio, interest and inflation etc. were "widely accepted" able to predict excess returns, (Lettau and Ludvigson, 2001)
- Welch and Goyal (2008) conclude that not a single variable had any statistical forecasting power, and the significance values of some factors change with the choice of sample periods.

Background

- More factors produced by literature: currently over 600 documented (Harvey and Liu, 2019)
  - False discovery problem, (Harvey et al., 2016)
  - Factors are cross sectionally correlated - inefficient covariances, factors may be subsumed within others, (Feng et al., 2019)
  - Number of factors may be more than sample size, making regression impossible

Why apply Machine Learning in Finance?

Machine learning methods have been used within the literature and appear to be well suited:

- High dimensional - more flexible than traditional regression models, which make strong functional form assumptions and are sensitive to outliers, (Freyberger et al., 2017)
- Explicit "regularization" methods for guarding against overfitting
- Methods to produce an optimal model from all possible at manageable computation cost

More able to manage the explosion in the number of factors suggested by the literature!

Applications in the Literature

Causal Analysis:

- Kozak et al. (2017), Rapach and Zhou (2013), Freyberger et al. (2017), and others apply shrinkage and selection methods to identify important factors

Prediction Performance:

- Gu et al. (2018), Feng et al. (2018), construct machine learning portfolios that historically outperform traditional portfolios in terms of prediction error and predictive R^2
- Attribute their success to machine learning's ability to find non-linear interactions

Motivations

However, little work has been done on how machine learning actually recognises and deals with the challenges in financial data.

- Feng et al. (2018) cross validates their training set, destroying temporal aspect of data, and only explore a handful of factors
- Gu et al. (2018) only use data up until the 1970s to produce predictions in the last 30 years
- Gu et al. (2018)'s models do not have consistent importance metrics - only their tree based methods recognise dividend yield as important

Motivations

Can machine learning deal with the challenges in financial data?

- Persistent Regressors?
- Identify true factors from a high dimensional, cross sectionally correlated panel?
- Is regularization enough to handle non-robustness?
- Are their conclusions consistent?
- Do they perform better than traditional methods?

Motivations

We explored this via two studies, focusing on the prediction performance and factor selection performance of some common machine learning models:

Simulation study

-

Empirical study

-

Model Specification

Returns are modelled as an additive error model

$$r_{i,t+1} = E(r_{i,t+1}|\mathcal{F}_t) + \epsilon_{i,t+1} \tag{1}$$

where

$$E(r_{i,t+1}|\mathcal{F}_t) = g^*(z_{i,t}) \tag{2}$$

Stocks are indexed as $i = 1, \ldots, N$ and months by $t = 1, \ldots, T$.

$g^*(z_{i,t})$ represents the model approximation using the P dimensional predictor set $z_{i,t}$.

Overview

Machine Learning Methodology consists of 3 overall components:

- Sample Splitting
- Loss Function(s)
- Models/Algorithms considered

Sample Splitting - Expanding Window Scheme

An expanding/growing window approach was used to

- Allow models to incorporate more data over time
- Preserve temporal ordering of data (compared to cross-validation)
- Allows the model to use the most recent data, in some sense

The training/validation split was chosen such that the size of the training set was 1.5 times the length of the validation set to begin with, consistent with Gu et al. (2018). Split specification is ultimately subjective

Sample Splitting

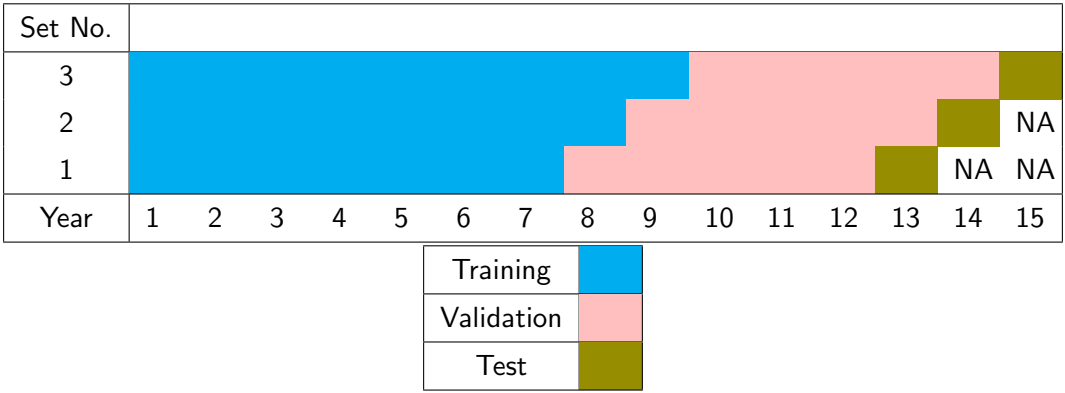


Figure 1: Sample Splitting Procedure

Loss Functions

Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{j=i}^{n}|y_j - \hat{y}_j| \tag{3}$$

Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n}\sum_{j=i}^{n}(y_j - \hat{y}_j)^2 \tag{4}$$

Models

We focus on 4 common machine learning models:

- Linear Models
- Penalized Linear Models
- Random Forests
- Neural Networks

Linear Models

Linear Models assume that the underlying conditional expectation $g^*(z_{i,t})$ can be modelled as a linear function of the predictors and the parameter vector θ:

$$g(z_{i,t};\theta) = z_{i,t}'\theta \tag{5}$$

Optimizing θ w.r.t. MSE yields the Pooled OLS estimator

Penalized Linear Models

Linear Models + Penalty term (Elastic Net by Zou and Hastie (2005) shown):

$$\mathcal{L}(\theta; .) = \underbrace{\mathcal{L}(\theta)}_{\text{Loss Function}} + \underbrace{\phi(\theta; .)}_{\text{Penalty Term}} \tag{6}$$

$$\phi(\theta; \lambda, \rho) = \lambda(1-\rho)\sum_{j=1}^{P}|\theta_j| + \frac{1}{2}\lambda\rho\sum_{j=1}^{P}\theta_j^2 \tag{7}$$

Elastic Net penalty aims to produce efficient and parsimonious via shrinkage and selection

Regression Trees & Random Forests

- Fully non-parametric models that can capture complex multi-way interactions.
- A tree "grows" in a series of iterations:
  1. Make a split ("branch") along one predictor, such that it is the best split available at that stage with respect to minimizing the loss function
  2. Repeat until each observation is its own node, or until the stopping criterion is met
- Slices the predictor space into rectangular partitions, and predicts the unknown function $g^*(z_{i,t})$ with the "average" value of the outcome variable in each partition to minimize the loss function

Random Forests

Trees have very low bias and high variance

They are very prone to overfitting and non-robust

Random Forests were proposed by Breiman (2001) to address this

- Create B bootstrap samples
- Grow a highly overfit tree to each, but only using m random subset of all predictors for each
- Average the output from all trees as an ensemble model

Neural Networks - Overview

Most complex model considered

Able to capture non-linearities systematically

In theory able to approximate any function - Universal Approximation Theorem

Neural Network Specification

- Neural networks with up to 5 hidden layers were considered.
- The number of neurons is each layer determined by geometric pyramid rule (Masters, 1993)
- All units are fully connected

ReLU activation function was chosen for all hidden layers for computational speed, and hence popularity in literature:

$$\text{ReLU}(x) = max(0, x) \tag{8}$$

Neural Network Tuning

Neural Networks have the largest number of hyperparameters to tune - computationally infeasible to conduct comprehensive grid search.

A conservative gridsearch was conducted, but it was observed that in general, default values provided the best performance.

Observed that neural networks are high sensitive to hyperparameters - not easy to tune!

Hyperparameter Tuning

Hyperparameters gridsearched:

- Optimizer
- Learning Rate
- Batch Size
- l1 penalty
- Activation function
- Momentum

Neural Network - Activation Function

Gu et al. (2018) specified using the ReLU activation function for all layers

We found that this specification does not work due to the "dying ReLU" problem"

Alternatives such as Leaky ReLU also did not give very good performance

A specification of the tanh activation function was observed to give the best results

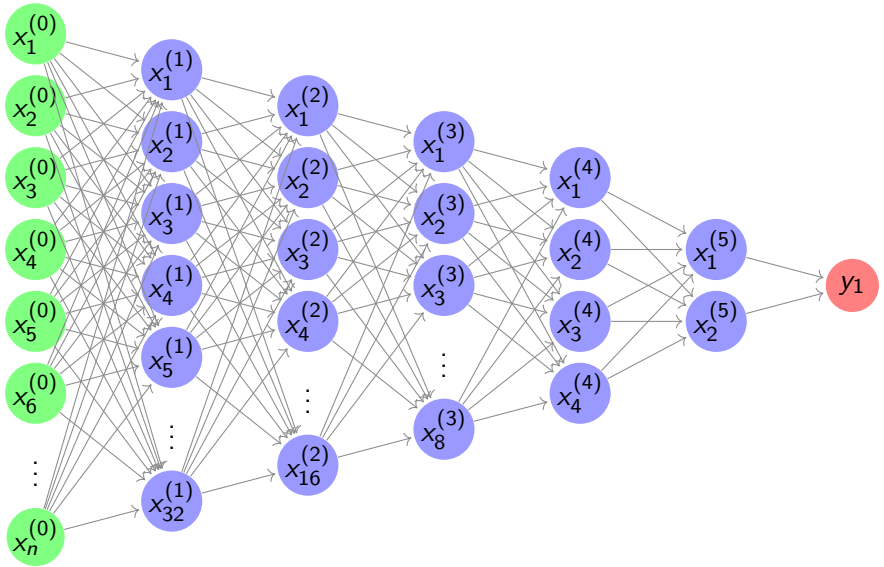


Figure 2: Neural Network 5 (most complex considered)

Loss Metrics

Overall predictive performance for individual excess stock returns were assessed using the following loss metrics:

Mean Squared Error

Mean Absolute Error

Out of sample predictive R

Out of Sample R squared

An out of sample R squared metric was also reported, as is popular in the financial literature. There is no consensus as to how this metric is to be calculated, and we use the following formulation:

$$R^2_{OOS} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \widehat{r}_{i,t+1})}{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \bar{r}_{i,t+1})^2} \tag{9}$$

where $\mathcal{T}_3$ indicates that the fits are only assessed on the test subsample

Interpretation of this is be cautioned, as R squared was originally designed for use in assessing in sample fit for linear models.

Variable Importance

- Defined as the reduction in predictive R-Squared from setting all values of predictor j to 0, while holding the remaining model estimates fixed
- Note some minor transformations applied to this metric for numerical reasons

$$VI_{j,norm} = \frac{VI_j + \min(VI_j) + o}{\Sigma VI_j + \min(VI_j) + o} \quad ; \quad o = 10^{-100} \tag{10}$$

Simulation Study

Simulation study conducted see how well machine learning performs in a controlled environment which we understand, as opposed to empirical data which we do not understand
Characteristics of financial data to be captured

- Stochastic volatility in errors
- Low signal to noise ratio
- Persistence across time in regressors
- Cross sectionally correlated (collinear) regressors
- High number of regressors

Gu et al. (2018)'s Simulation Design

We replicate Gu et al. (2018)'s simulation design as a starting point Several Problems:

- White noises, constant volatility specification
- No cross sectional correlation

Proposed Simulation Design

Latent factor model with stochastic volatility for excess return, r_{t+1}, for $t = 1, \ldots, T$:

$$r_{i,t+1} = g\left(z_{i,t}\right) + \beta_{i,t+1} v_{t+1} + e_{i,t+1}; \tag{11}$$

$$z_{i,t} = \left(1, x_t\right)' \otimes c_{i,t}; \quad \beta_{i,t} = \left(c_{i1,t}, c_{i2,t}, c_{i3,t}\right); \tag{12}$$

$$e_{i,t+1} = \exp\left(\frac{\sigma^2_{i,t+1}}{2}\right) \varepsilon_{i,t+1}; \tag{13}$$

$$\sigma^2_{i,t+1} = \omega + \gamma_i \sigma^2_{t,i} + w_{i,t+1} \tag{14}$$

v_{t+1} is a 3×1 vector of errors, $w_{i,t+1}, \varepsilon_{i,t+1}$ are scalar error terms. Variances tuned such that the R squared for each individual return series was 50% and annualized volatility 30%.

Simulating Characteristics

Matrix C_t is an $N \times P_c$ vector of latent factors.

x_t is a 3×1 multivariate time series

ε_{t+1} is a $N \times 1$ vector of idiosyncratic errors.

Simulation mechanism for C_t that gives correlation across the factors & time

Draw normal random numbers for each $1 \leq i \leq N$ and $1 \leq j \leq P_c$, according to

$$\overline{c}_{ij,t} = \rho_j \overline{c}_{ij,t-1} + \epsilon_{ij,t}; \quad \rho_j \sim \mathcal{U}\left(\frac{1}{2}, 1\right) \tag{15}$$

Simulating Characteristics

Then, define the matrix

$$B := \Lambda\Lambda' + \frac{1}{10}\mathbb{I}_n, \quad \Lambda_i = (\lambda_{i1}, \ldots, \lambda_{i4}), \quad \lambda_{ik} \sim N(0,1),\ k = 1, \ldots, 4 \tag{16}$$

Transform this into a correlation matrix W via

$$W = (\text{diag}(B))^{\frac{-1}{2}} (B) (\text{diag}(B))^{\frac{-1}{2}} \tag{17}$$

Use W to build in cross sectional correlation for $N \times P_c$ matrix $\bar{C}_t$:

$$\widehat{C}_t = W\overline{C}_t \tag{18}$$

Simulating Characteristics

Finally, the "observed" characteristics for each $1 \leq i \leq N$ and for $j = 1, \ldots, P_c$ are constructed according to:

$$c_{ij,t} = \frac{2}{n+1} \operatorname{rank}\left(\hat{c}_{ij,t}\right) - 1. \tag{19}$$

with the rank transformation normalizing all predictors to be within $[-1, 1]$

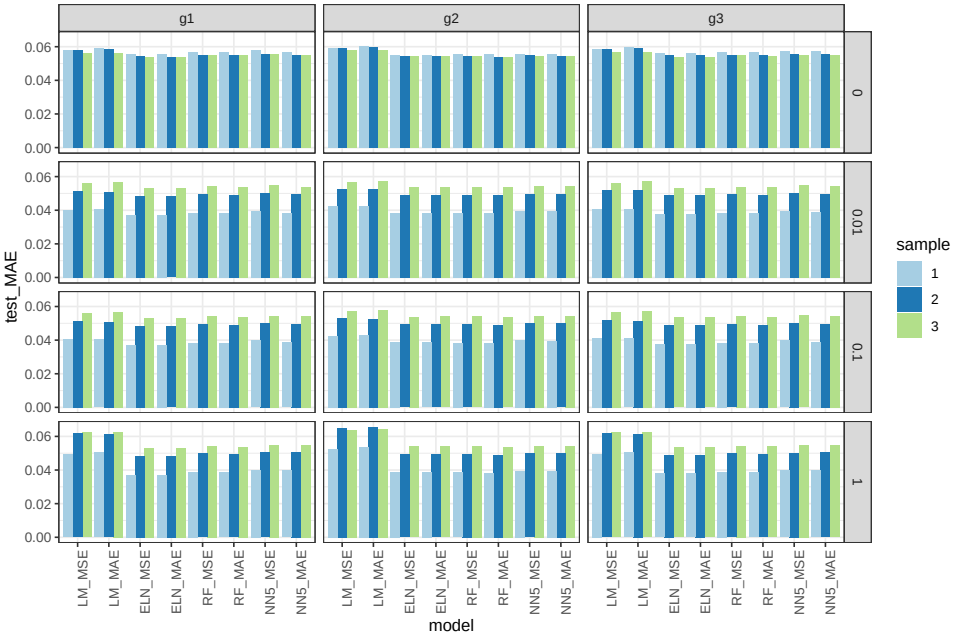
Simulating Return Series

We will consider four different functions $g(\cdot)$:

$$(1)\ g_1(z_{i,t}) = \left(c_{i1,t}, c_{i2,t}, c_{i3,t} \times x'_{3,t}\right)\theta_0$$
$$(2)\ g_2(z_{i,t}) = \left(c^2_{i1,t}, c_{i1,t} \times c_{i2,t}, \text{sgn}\left(c_{i3,t} \times x'_{3,t}\right)\right)\theta_0$$
$$(3)\ g_3(z_{i,t}) = \left(1[c_{i3,t} > 0], c^3_{i2,t}, c_{i1,t} \times c_{i2,t} \times 1[c_{i3,t} > 0], \text{logit}\left(c_{i3,t}\right)\right)\theta_0$$

Tune θ^0 s.t. cross sectional R^2 is 25%, and predictive R^2 is 5%.

The simulation design results in $3 \times 4 = 12$ different simulation designs, with $N = 200$ stocks, $T = 180$ periods and $P_c = 100$ characteristics. Each design will be simulated 50 times to assess the robustness of machine learning algorithms.


g1
g2
g3
0
0.01
0.1
1
test_MAE
0.00
0.02
0.04
0.06
LM_MSE
LM_MAE
ELN_MSE
ELN_MAE
RF_MSE
RF_MAE
NN5_MSE
NN5_MAE
model
sample
1
2
3


g1
g2
g3
0
0.01
0.1
1
test_MSE
0.0075
0.0050
0.0025
0.0000
LM_MSE
LM_MAE
ELN_MSE
ELN_MAE
RF_MSE
RF_MAE
NN5_MSE
NN5_MAE
model
sample
1
2
3

Data Source

Gu et al. (2018)'s dataset of individual factors available from their website:

- March 1957 - December 2016
- 61 annual factors
- 13 quarterly factors
- 20 monthly factors
- Industry dummy with 74 levels

Data Cleaning Procedure

Reducing dataset size:

- Only include NASDAQ stocks
- Filter out penny stocks (microcaps)
- Keep only instruments with share code of 10, 11 (filtering out REITs, etc)
- Convert to a monthly format
- Industry dummy was dropped due to inaccuracy and high dimensionality

Data Cleaning Procedure

Missing Data

- Significant increase in data quality and availability after 1993 Q3
- Factors with more than 20% missing data were removed
- Remaining missing factors were imputed with cross sectional medians

Macroeconomic Factors

Table 1: Macroeconomic Factors, (Welch and Goyal (2008))

No.	Acronym	Macroeconomic Factor
1	macro_dp	Dividend Price Ratio
2	macro_ep	Earnings Price Ratio
3	macro_bm	Book to Market Ratio
4	macro_ntis	Net Equity Expansion
5	macro_tbl	Treasury Bill Rate
6	macro_tms	Term Spread
7	macro_dfy	Default Spread
8	macro_svar	Stock Variance

Cleaned Dataset

Baseline set of covariates defined as:

$$z_{i,t} - (1, x_t)' \otimes c_{i,t} \tag{20}$$

where $c_{i,t}$ is a P_c matrix of characteristics for each stock i, and $(1, x_t)'$ is a $P_x \times 1$ vector of macroeconomic predictors.

Number of covariates in this baseline set is $61 \times (8 + 1) = 549$.

Sample Splitting

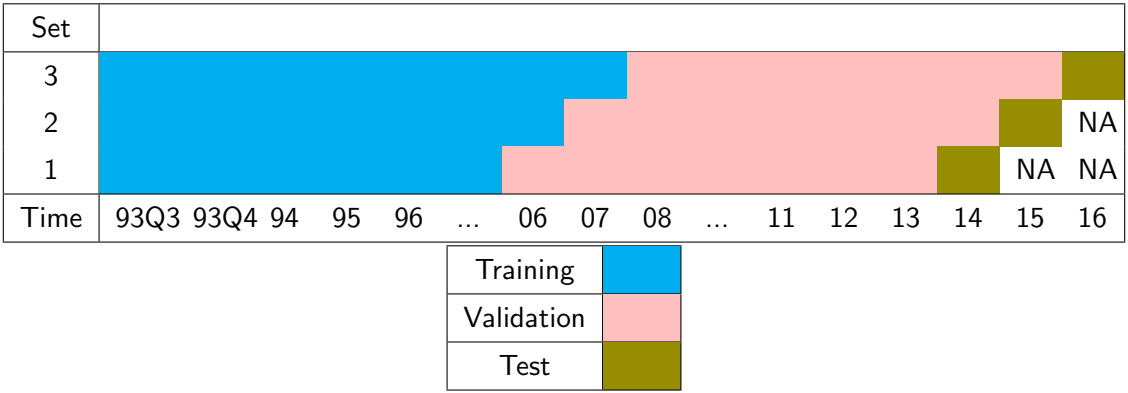


Figure 5: Empirical Data Sample Splitting Procedure

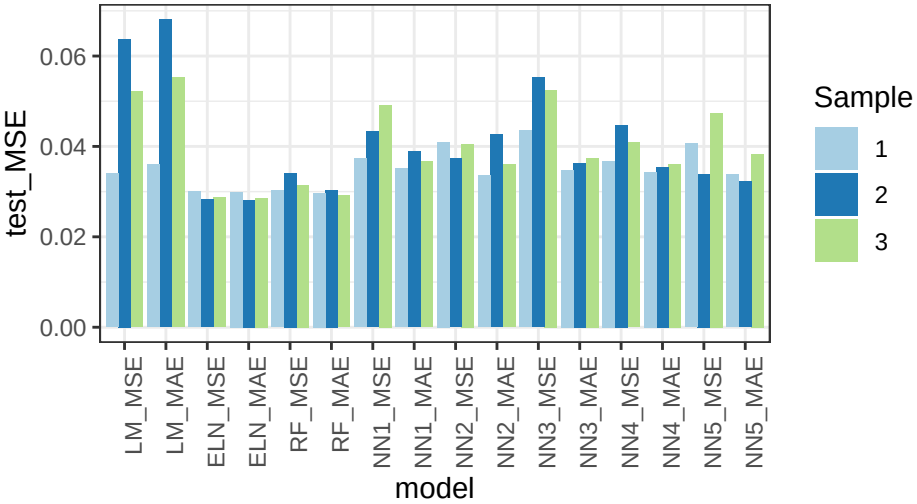
Overview

In general, results from the simulation study were repeated.

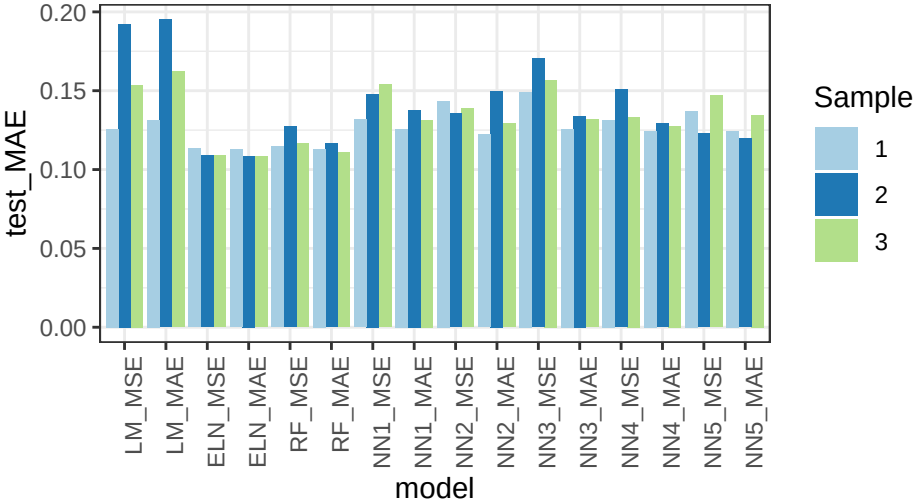
- Suggest that simulation study was accurate to some degree
- Support for same conclusions to be made

Again see that results are different from Gu et al. (2018). Neural networks do not vastly outperform other models!

Prediction Accuracy - MSE



Prediction Accuracy - MAE

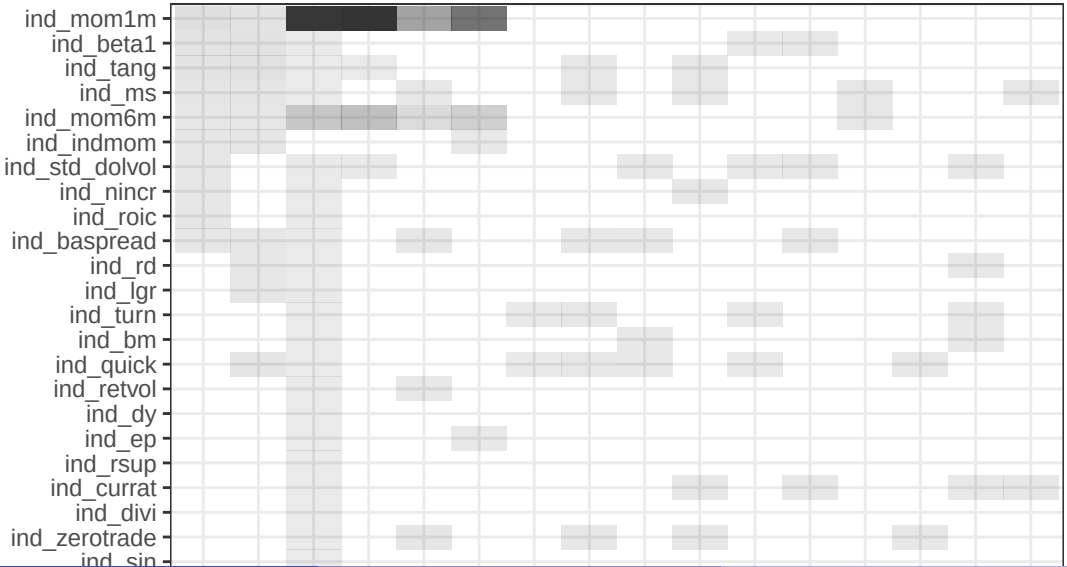


Prediction Accuracy - Neural Networks

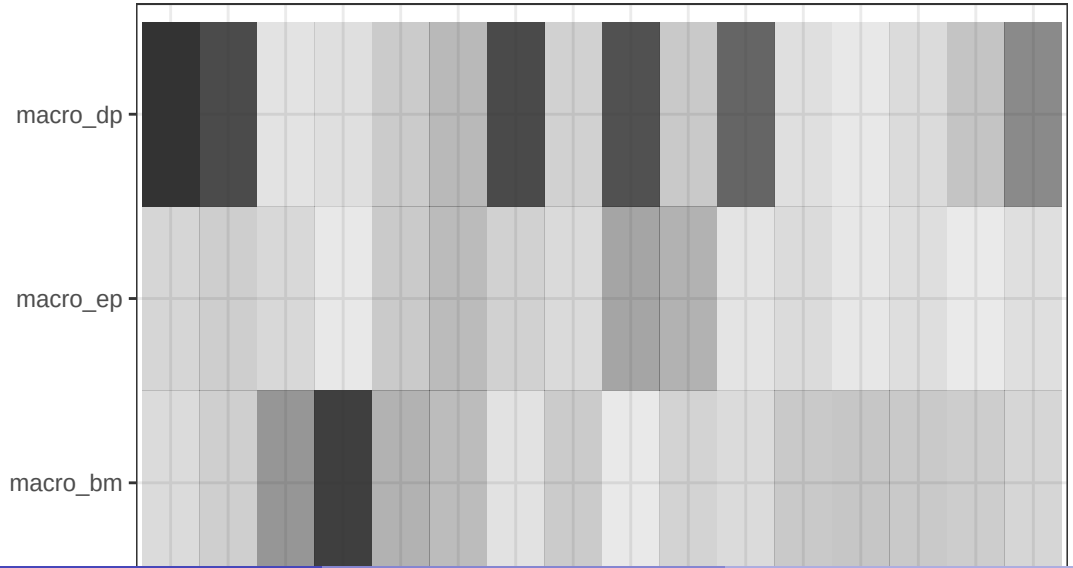
Similarly, some weak evidence that neurla networks with more hidden layers perform better

Neural Networks are still unstable

Variable Importance - Individual Factors



Variable Importance - Macroeconomic Factors



Variable Importance

Important factors in penalized linear models and random forests:

- 1 and 6 month momentum factors

Important factors in neural networks:

- Market Value (though inconsistent)

Conclusion

Machine learning offers tools to improve stock return prediction and identification of true underlying regressors.

Penalized linear models, and to a lesser extent, random forests are the best performing models.

Feed-forward neural network architectures considered fail in the context of stock return prediction and variable importance analysis. The result is consistent across simulated data and empirical data. Importantly, this is in direct contradiction to previous literature, which concluded that neural networks vastly outperform.

Top performing models - penalized linear and random forests, tend to agree and correctly identify causal regressors in simulated contexts, and same subset in empirical contexts. Neural networks only agree in simulated contexts.

Penalized linear models are most consistent at correctly identify causal regressors, though this result is not always reliable, particularly when non-linearities are introduced.

References

Ang, A., Bekaert, G., 2006. Stock return predictability: Is it there? The Review of Financial Studies 20, 651–707.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

Feng, G., Giglio, S., Xiu, D., 2019. Taming the Factor Zoo: A Test of New Factors. Tech. Rep. w25481, National Bureau of Economic Research, Cambridge, MA.

Feng, G., He, J., Polson, N. G., 2018. Deep Learning for Predicting Asset Returns. arXiv:1804.09314 [cs, econ, stat] ArXiv: 1804.09314.

Freyberger, J., Neuhierl, A., Weber, M., 2017. Dissecting characteristics nonparametrically. Tech. rep., National Bureau of Economic Research.

Goetzmann, W. N., Jorion, P., 1993. Testing the predictive power of dividend yields. The Journal of Finance 48, 663–679.

Goyal, A., Welch, I., 2003. Predicting the equity premium with dividend ratios. Management

Questions and Answers