Evaluation of Machine Learning in Finance

Ze Yu Zhong

Supervisor: David Frazier

Monash University

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.

Background

- Factors: a collection of regressors to be used in pricing returns that can be used to proxy for unknown underlying risk factors due to their correlation with cross sectional returns, (Harvey et al., 2016)
  - Violates strict view that risk factors should be variables that have unpredictable variation through time, and be able to explain cross sectional returns independently

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

What is Machine Learning?

"...a vast set of tools for understanding data."

- *An Introduction to Statistical Learning*, (Hastie et al., 2009)

We will define it as a diverse collection of:

- high dimensional models for statistical prediction,
- "regularization" methods for model selection and mitigation of overfitting in sample data
- efficient systematic methods for searching potential model specifications

Applications in the Literature

- Kozak et al. (2017), Rapach and Zhou (2013), Freyberger et al. (2017), and others apply shrinkage and selection methods to identify important factors
- 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?
- Explore this via simulation and popular machine learning models
- Models will also be evaluated again, but with more recent, representative financial data to explore robustness.

Model Overview

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 Functions
- Models/Algorithms considered

Expanding/Growing Window Approach

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

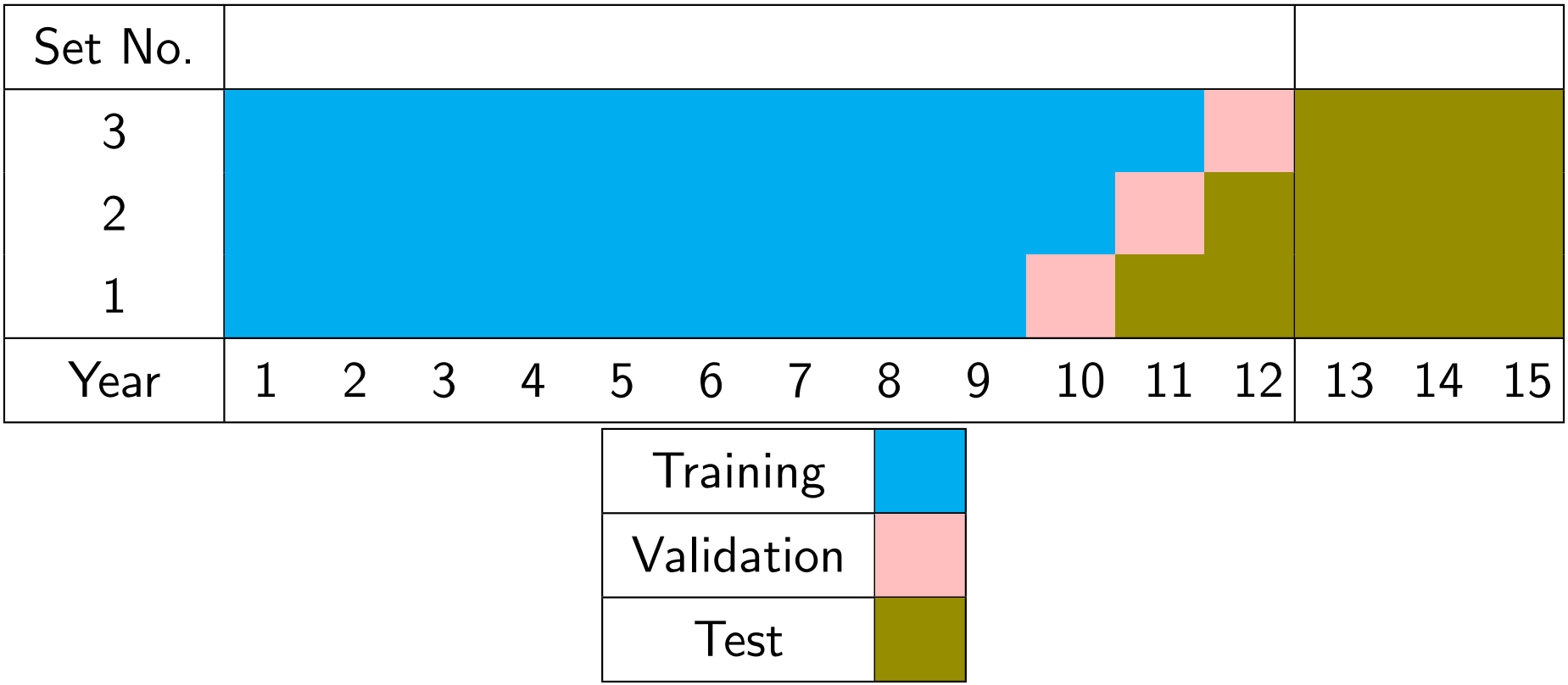

Figure 1: Sample Splitting Procedure

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)

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

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 produces 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

$$x_k^{(l)} = \alpha(x^{(l-1)'}\theta_k^{l-1}) \tag{8}$$

$$x_1^{(1)} = \alpha\left((x_0^{(0)}, x_1^{(0)}, x_2^{(0)}, x_3^{(0)})'\theta_1^0\right) \tag{9}$$

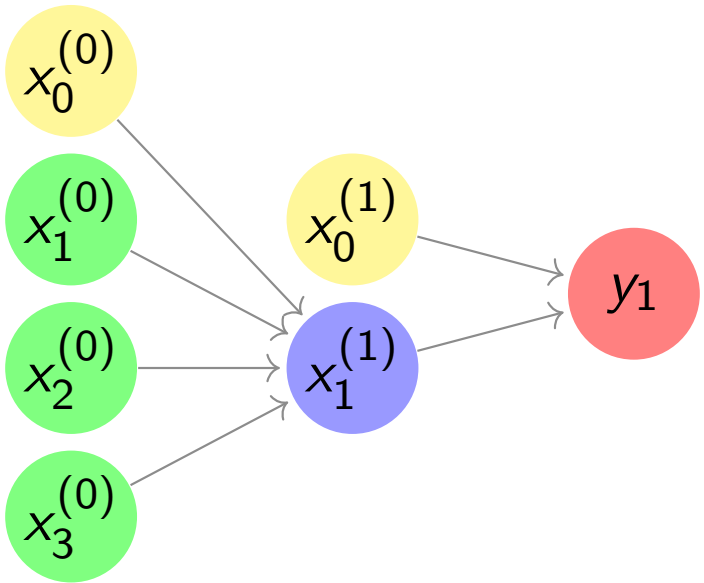

Figure 2: Sample Neural Network

Neural Network Specifications

- 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:

$$\text{ReLU}(x) = max(0, x) \tag{10}$$

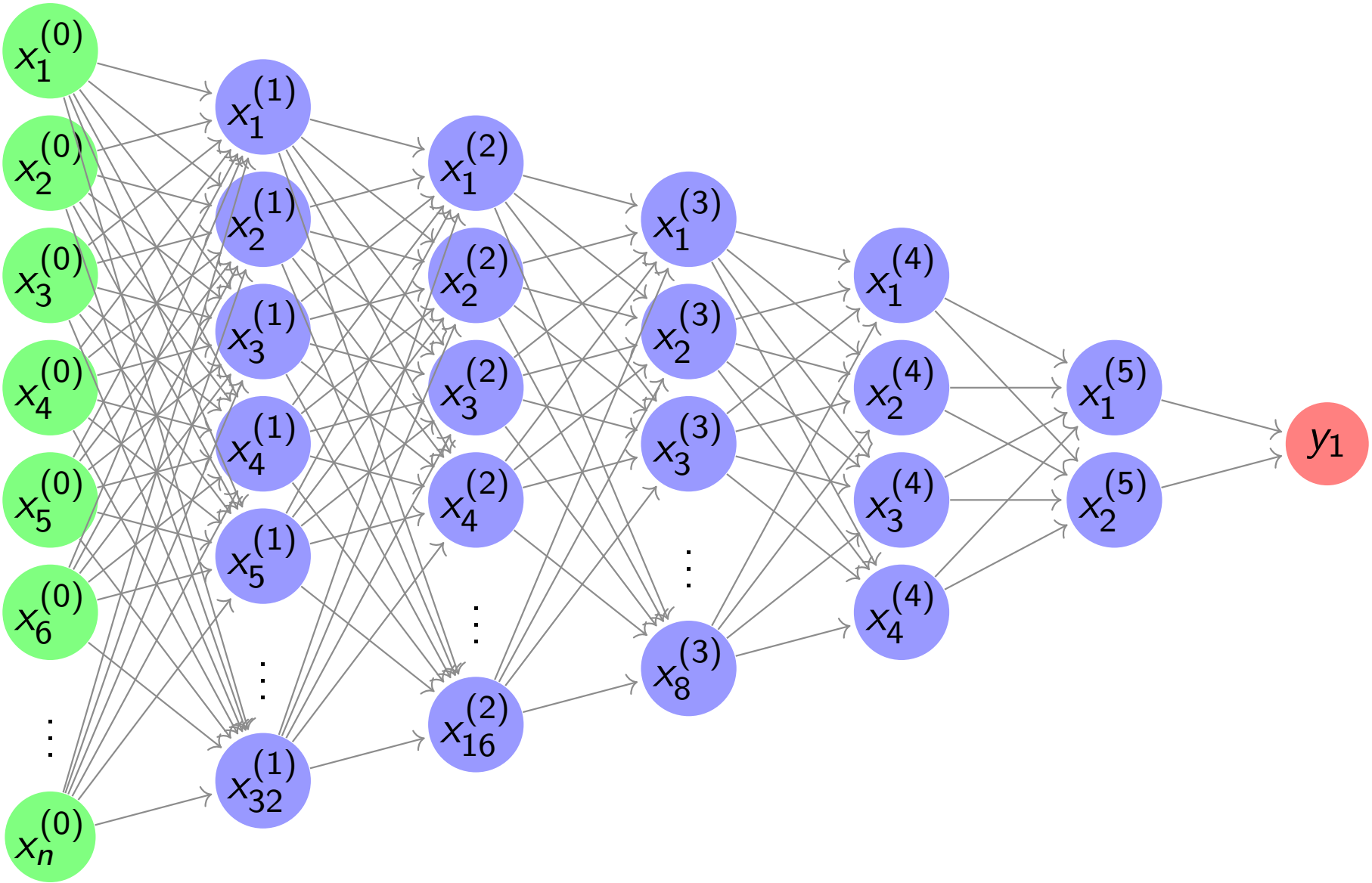

Figure 3: Neural Network 5 (most complex considered)

Overall Simulation Design

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

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

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

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

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

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.

Data Source

- CRSP/Compustat database for stock returns with stock level characteristics such as accounting ratios and macroeconomic factors will be queried.
- Only more recent data will be used, such as the period before and after 2008 GFC

Out of Sample R Squared

Overall predictive performance for individual excess stock returns were assessed using the out of sample R^2:

$$R^2_{OOS} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \widehat{r}_{i,t+1})}{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \bar{r}_{i,t+1})^2} \tag{20}$$

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

Diebold Mariano Tests for Predictive Accuracy

- Compares the forecast accuracy of two forecast methods, (Diebold and Mariano (2002) and Harvey et al. (1997))
- Tests whether or not the difference series ($d_t = e_{1t} - e_{2t}$) between two forecast methods' errors is different from zero
- e_{1t} and e_{2t} represent the average forecast errors for each model

Variable Importance

- The importance of each predictor j is denoted as VI_j
- Defined as the reduction in predictive R-Squared from setting all values of predictor j to 0, while holding the remaining model estimates fixed
- Will allow us to see what factors the models have determined to be important

Results

Work is currently being done on trying to tune the R-Squared values of the simulated datasets

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.

Diebold, F. X., Mariano, R. S., 2002. Comparing predictive accuracy. Journal of Business & economic statistics 20, 134–144.

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

Questions and Answers