Recent Developments in Random Forest and Practical Applications to Finance

6. Semester, Oecon

Aalborg University Business School

AALBORG UNIVERSITET

Samuel Hvidager - 20195146

Supervisor: Sebastian Valdecantos

June 7, 2022

Abstract

This thesis examines the recent advancements in improving bias correction for the machine learning algorithm, random forest. Random forest which builds upon the theory of decision trees is usually considered strongest when applied to nonparametric modeling. For this reason, this thesis evaluates the relative improvement when used in the practical application of predicting future valuations of publicly traded companies from financial statements. Specifically, we compare Breiman's orignal random forest algorithm with Athey, J. Tibshirani, and Wager's generalized random forest approach, and Ghosal and Hooker's one-step boosted forest.

The analysis was conducted with in-bag and out-of-bag test. The dataset worked with consisted of 5605 yearly financial reports from the period 2016 to 2021. A total 1213 different companies listed on the NASDAQ stock exchange were included. In the interest of model accuracy and due to resource constraints, the three models were built as supervised learning models. Our predicted criterion was the total market value of any given publicly traded company, one year after the reporting date. In total 23 distinct financial reporting metrics were selected as training parameters.

Our results for in-bag predictions showed that Breiman's original algorithm performed the best, with an R2 value of 0.969 and a mean prediction error of 31.7%. Ghosal and Hooker almost matched Breiman, with an R2 value of 0.901 and a mean prediction error of 30%. Athey et al. trailed behind the other two, with an R2 value of 0.773, and a mean prediction error of 66.3%. Athey et al. only improved when it came to runtime, where more than a fivefold increase in efficiency compared to Breiman's model, was seen for in bag results.

For out-of-bag evaluation, where we tested the models on unseen data, both Breiman and Athey et al. saw major decreases in performance. Breiman's R2 value dropped to 0.607, with the mean prediction error rising to 89.6%. For Athey et al. a similarly large drop in performance was seen, R2 value dropped to 0.581, with mean prediction error rising to 104.6%. Ghosal and Hooker's model in contrast outperformed expectations, with an R2 value of 0.833, with mean prediction error only rising to 36.9%.

That there is an overall drop in accuracy for all three models, when comparing In-bag and out-of-bag results, is regarded as a consequence of all three models initially being partially overfitted to the unique features of the dataset. This is in line with expectations, as Breiman's original algorithm didn't contain any improvements in bias correction, Breiman initially failed to generalize, and instead overfitted to the dataset, thereby giving a false impression of outperforming the other two models for in-bag results. We found that the reason for Athey et al. overall underperforming for both in-bag and out-of-bag stemmed from the fact that their bias correction was overly strict, as shown on global importance factors. As a result, too many parameters were underweighted, and their model ended up overgeneralizing. Finally, we contribute the improvement seen for Ghosal and Hooker's one-step boosted forest, to the fact that by training a second forest on the residuals of the first model, it could successfully detect which individual samples contributed to bias in the model. This conclusion is supported by the fact that for in-bag results Ghosal and Hooker didn't overfit to the same

degree as Breiman, due to a better ability to generalize from improved bias correction.

Overall we conclude that when a random forest is used for predicting future financial equity valuations, Ghosal and Hooker's one-step boosted forest offers a significant advantage over Breiman's original algorithm; while Athey et al. can not be considered an improvement in predictive capability.

Contents

List of Figures and Equations

List of Tables

1 Preface

This bachelor thesis was authored by Samuel Hvidager, with supervision by Sebastian Valdecantos. The following paper is written in English and there was given a waiver on the requirement to write in Danish. Chicago citation style is used. The character count with spacing is 53618, equivalent to 22.34 pages of 2400 characters. Table of contents, figures, pictures and appendix not accounted for.

2 Introduction

Random forest has seen a broad adoption use for a wide array of statistical analyses in the last two decades. The traditional random forest model uses a ensemble learning approach, where the given data is split into subsets to perform individual model training on it, the results of the sub samples are then used for bagging where the weight of the results are assigned. While random forest often outperforms a wide array of machine learning models and is generally considered robust compared to other nonparametric models Hamza and Larocque (2005), outlier bias can still be seen to significantly distort results for high dimensional data Tang, Garreau, and Luxburg (2018).

In recent years new additions to the field have made significant progress in correcting for the previously mentioned problem. This thesis investigates how well these new additions improve the overall accuracy and reliability when employed in practical applications. Respectively it was chosen to conduct an analysis of the causal relationships found in financial reporting data of publicly traded companies, as this is widely considered a prime example of non-parametric data Grant (2022).

The structural framework of this thesis consist of three primary parts. The first being section 4, where we give an theoretical overview of the three models, with the underpinnings in decision trees, and the key challenges encountered when working with random forest. The second part in section 5, comprise of model construction and evaluation. We describe how the models are build and, and what main considerations that there were taken when selecting features of the datasets. Afterwards we follow this up with a series of test to evaluate the performance of three models. The last primary part is section 6, which considers the main results from the evaluation phase, and incorporate this together with risk of bias, into a discussion about our findings.

In the next section we start out with defining our area of study for this thesis, as well as the central research methodology employed.

3 Area of Study

During recent developments in random forest analysis, there have been introduced multiple new novel solutions for tackling the deficiencies from Breiman's (2001) original paper when analyzing nonparametric high dimensional data. Specifically, this paper conducts an empirical comparison of Breinman's original model, with Athey, J. Tibshirani, and Wager's (2019) method from generalized random forests, and Ghosal and Hooker's (2021)one-step boosted forest. The analysis seeks to independently investigate the strength of using the mentioned models for identifying causal relationships between the fundamental financial factors of publicly traded companies and their equity market valuations.

The main question that this report attempts to answer can be summarised as follows:

How well do the recent developments in random forest modeling improve predictive capabilities compared to Breiman's original algorithm when employed as instrumental tools for performing out of sample predictive modeling of equity product valuation?

In this case "improvement" will primarily be measured by the marginal improvement in the predictive accuracy of the model. There are however several secondary factors that have to be taken into consideration in order to accept the improvements as satisfactory, they include but are not limited to:

1. Are the improved models more prone to overfitting?
2. Does the improvement come at the expense of a significant increase in computing?
3. Is there a significant variable bias present?
4. Are the improvements uniform across all sample sizes?

For further clarification about the selection of data for evaluation of Random Forest, see section 5.1.

3.1 Delimitation

As apparent when looking at our analysis, it can be seen that the dataset worked with are not time-series data. While Random forest isn't inherently an autoregressive model, in theory, we could have transformed a given time series, like the development of earnings over time, into some variable(s) representing this. A strong presence of nonlinear patterns could be possible when working with time-series data, as such if any of the models are comparatively better at identifying bias for nonlinear datasets, this feature might have been overlooked. That is not to imply that the findings are at question, just that it is unknown if the three models all comparatively rank in the same way for other predictive modeling problems.

4 Recent Developments in Random Forest Analysis

Random forest is a ensemble learning[1] algorithm for performing regression and classification. This paper is mainly interested with the use of random forest for regression. Random forest builds on the idea of decision trees and regression trees, where the dependant variables are predicted through the cataloguing of the independent variables relations with the predictor.

4.1 Decision and Regression Trees

This section was in large part based on chapter 9 from the book *The Elements of Statistical Learning* by Hastie, R. Tibshirani, and Friedman (2009a). The content was cross-referenced with the book *classification and regression trees*. Breiman et al. (2017). Unless otherwise is specifically stated, all proceeding writing is based on the first-mentioned source.

Decision tree is a term covering models where the data is split up into subgroups based on its characteristics, an illustrative model bears semblance to that of a tree, from which it takes its name. The general idea behind the decision tree is that instead of trying to fit one regression model to the entirety of the data set, the data is split up into smaller groups such that a categorical regression model can be fit to an individual data subset[2]. The advantage of using a regression model built on a decision tree is illustrated in figure 1

[1] Multi model approach or meta model

[2] In theory, there is not a limit to the number of times a decision tree can be split, but generally, we observe that the more times that data is split, the more overly specific the regression model will become fitted to arbitrary noise is the data set.

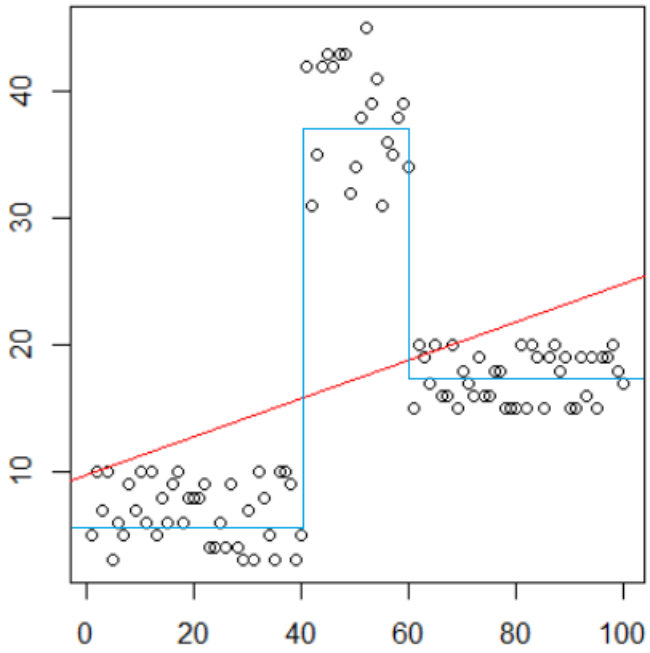

Figure 1: Illustration of a linear model vs a regression tree model[3].

.

A regression model fitted to a decision tree is referred to as a regression tree. An example of a decision tree is pictured on figure 2.

[3]This regression model is one where no localized regressions has been performed, see section 4.3 about smoothing for further clarification.

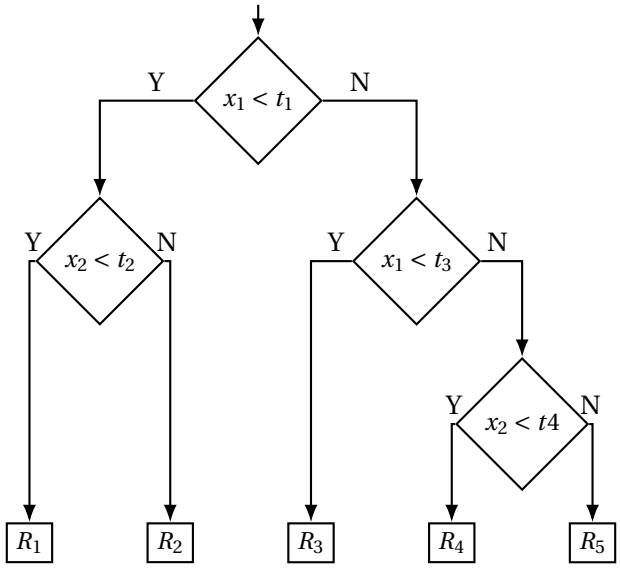

Figure 2: Model of a simple decision tree[4].

The various subgroups of the data $R_1, R_2, ..., R_5$ in figure 2 are referred to as leaves, with the points where the data is split into smaller sub-samples based on its characteristics $X_1 < t1, X_2 < t_2$ ect, referred to as features. Each node typically has varying depth and includes a varying amount of features; the ideal tree grows to a depth where it performs the best as a predictor. Assume a case where we wish to model the continuous response for a given variable called Y using the previously mentioned factors in figure 2. By performing a multitude of binary splits it can be found what individual subset of the data can best be used for predictive modeling. The regression model shown below can model Y with a responding criterion minimization for the sample variance set as the constant c_m in this case for the regions R_m[5]:

$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\{(X_1, X_2) \in R_m\}$$

The main challenge in growing a regression tree is finding the optimal binary partition of the dataset in regards to having the minimum sum of squares $\sum(y_i - f(x_i))^2$, i.e. the accuracy. In this case, the best $\hat{c}_m$ is given by the average of y_i for R_m:

$$\hat{c}_m = \bar{y}(y_i | x_i \in R_m)$$

[4]Data is split based on if the stated condition holds true or not, with Y indicating Yes and N indicating No. Code adopted in part from: Rendering forest diagram in LaTeX (2022)

[5]The five subgroups of the data

This however creates two unaccounted for problems, first, the computing need rises exponentially as each new entry increases the number of potential ways the data can be split. Second, while there in theory aren't a limit to the number of times a decision tree can be split, it is generally observed that the more times that data is split, the more overly specific the regression model will become to individual entries in the data set.

Due to the first problem being unsolvable, it is chosen to work around this fact by using a *greedy algorithm* (2022), where we attempt to make the approximately right splitting choice first time, i.e. the localized optimal choice is chosen at each stage. In a scenario where all the data is considered, v is defined as the splitting variable with the split point being s; a pair of the data split up into two groups can be defined as:

$$R_1(v,s) = \{X|X_v \leq s\} \text{ and } R_2(v,s) = \{X|X_v \geq s\}$$

The optimal splitting variable v and splitting point s are the ones that solves:

$$\min_{v,s}\left[\min_{c_1}\sum_{x_i\in R_1(v,s))}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(v,s))}(y_i-c_2)^2\right]$$

For all possible v and s combinations, the minimization can be solved with:

$$\hat{c}_1 = \bar{y}(y_i|x_i \in R_1(v,s)) \text{ and } \hat{c}_2 = \bar{y}(y_i|x_i \in R_2(v,s))$$

By using this greedy algorithm for splitting, all the potential splits will be considered at a single split point s, and can be done much quicker than if all potential lateral splits were considered. After having done the split, the above process is replicated at all regions.

For solving the secondary problem of deciding the depth of the tree, a proposed strategy could be only performing a split if there is a resulting decrease in the sum of squares. Though, this method, unfortunately, does not account for the possibility that a very good split could be created from previous splits that appeared bad. An alternative approach usually employed is to overgrow the tree, and then perform post pruning based on the cost complexity. That is to say that any subtree that has overly high complexity and, therefore potentially contributes to overfitting is removed. A subtree is defined as any T_0 on the overarching tree T that are not leaves, I.e. $T \subset T_0$. All leaves are indexed by m; any given leaf representing a region of data is defined as R_m. $|T|$ represents the total number of leaves in the overarching tree T. Breiman et al. (2017) The cost complexity criterion that is used to identify which trees to perform pruning on is given as:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$$

Where N_m is the substitution estimate for probability of misclassification given as:

$$N_m = \#\{x_i \in R_m\}$$

And Q_m is the resubstitution estimate of misclassification cost for the tree:

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

With the actual sample variance $\hat{c}_m$ defined as:

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$$

α then ends up as the tuning parameter that decides the size of the tree, with smaller values equating to larger trees, meaning $\alpha = 0$ results in the full tree T_0. For every α weakest link pruning is used to collapse internal nodes and find the subtree $T_\alpha \subseteq T_0$ that minimizes $C_\alpha(T)$. This can be done continuously until a single node tree remains. The final α is usually estimated manually through cross-validation of other decision trees.

4.2 Random Forest

Random forest builds on the theory of decision trees. The main idea behind random forest is intuitively simple; the way a pure random forest is built is by fitting a multitude of decision trees on the same dataset, and then averaging the results across the multiple trees Hastie, R. Tibshirani, and Friedman (2009b). While a single decision tree is usually referred to as a weak estimator, due to it being overly fit to the noise of the data, and thereby having a high amount of variance in the expected results, at the same time, it is still approximately unbiased. The law of large numbers implies that by taking the average of multiple decision trees, the unbiasedness of all the individual decision trees is kept, while eliminating the high degree of variance Breiman (2001). This method was originally pioneered by Tin Kam HoHo (1995), and a later addition was created by Leo Breiman that included Bootstrap aggregating (bagging).

Bootstrapping is the principal idea behind bagging, where a pure random forest would just create a series of decision trees with varying splits. Bootstrapping changes this by creating a series of sub-samples from the original dataset. Consider a dataset Z of a given size n, defined as $Z^{(0)}_{[n]} = (Z^{(0)}_1, (Z^{(0)}_2, ..., (Z^{(0)}_n)$, where $(Z^{(0)}_i = (Y_i, X_i)$ for $[n]$ is given as $\{1, ..., n\}$. A random forest is fitted to the dataset, with a chosen number of trees times equal to B from $k < n$ number of sub samples[6]. These are created with replacement[7]. A decision tree is then fitted

[6] Any given number of samples can be selected depending on needed parameters for the tree.

[7] Replacement is necessary or else there would be a limit on the number of sub-samples that could be created.

for each sample, and the total is then aggregated[8]. Random forest usually has considerably less variance than decision trees in its results, though as explained in section 4.4 the curse of dimensionality, there is still a tendency for bias in the feature selection to occur for high dimensional data. An example implementation of a simplified version of Breiman's original random forest algorithm is shown below, by Hastie, R. Tibshirani, and Friedman (2009b):

Algorithm 1 Breiman's original Random Forest algorithm

1: For $b = 1$ to B :
   (a) Draw a bootstrap sample k from the training data.
   (b) Grow a random-forest tree Tb to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i Select m variables at random from the p variables.
      ii Pick the best variable/split-point among the m.
      iii Split the node into two daughter nodes.
2: Output the ensemble of trees $\{T_b\}_1^B$ To make a prediction at a new point x :

$Regression: \hat{F}^{(0)}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x; Z^0_{I_b^{(0)}})$

4.3 Kernel Density Estimation

Due to the fact that random forest is fundamentally categorical in its predictive output, a pure Random Forest would for each new input produce a prediction that fitted exactly into one of the predetermined categories that were identified from the training on the original data set. It's often preferred when using random forest for producing predictive outputs that the results can be any set of continuous variables, so the results are not locked into any fixed position. While not expletively stated in Breiman's original paper, practical implementations of the random forest algorithm normally employs the use of localised neighborhood regression, usually using a kernel weighing function to perform localised regression; localised regression is also colloquially referred to as smoothing.

[8]While aggregating for classification problems is somewhat complex, in this case where random forest is simply used for regression, the procedure is just to average the output of all the individual trees.

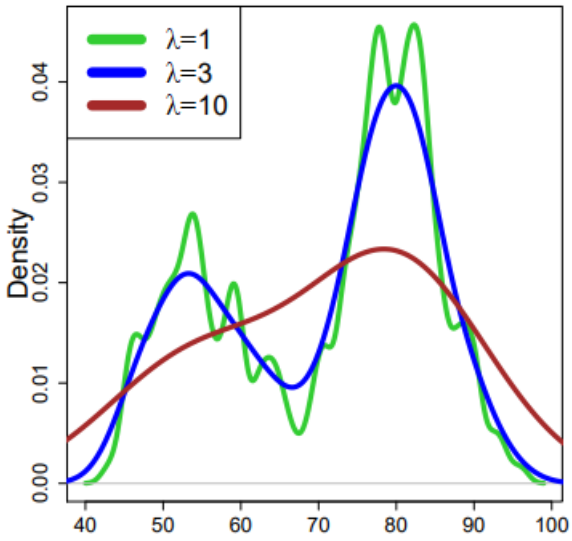

Figure 3: Plot of various kernel smoothing bandwidth's. Yen-Chi (2018)

The fundamental idea is to use a kernel weighting function for performing smoothing. Kernel weight is calculated such that nearby observations in the covariate space are weighted more when performing local maximum likelihood estimation Scornet (2015). For nonparametric regression problems, the conditional expectation is given as $f(x) = E(Y|X = x)$ where we have Y relative to X. The connotation is that Y is predicted at all points where the conditional mean is $X = x$. The kernel nearest neighbor average for that function can be roughly estimated using:

$$\hat{f}(x) = Ave(y_i|x_i N_k(x))$$

This method for estimating K-nearest neighbor however gives a uneven discontinuous $\hat{f}(x)$. A more precise kernel regression method was proposed by Nadaraya (1964) and Geoffrey (1964); it is given as:

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

Where K_λ is a given kernel with a bandwidth parameter λ that determines how large an area to take into consideration. As shown figure 3, kernel density is somewhat finicky, with varying amounts of adjustments to the bandwidth λ resulting in different levels of smoothness. The green line shows the case with an overly fit kernel estimation, where the bandwidth takes too much noise into consideration, The red line illustrates an scenario where the kernel estimation is overly smooth, the blue line captures the real underlying distribution present.

4.4 The Curse of Dimensionality

The curse of dimensionality refers to problems arising when working with data analysis containing a high number of dimensions for data *Dimensionality Increase - an overview | ScienceDirect Topics* (2022). Due to the fundamental nature of dimensionality the space encompassed by each additional dimension increases the total volume of the space exponentially. The volume of a ball in Euclidean space can be calculated as $(2r)^d$.

The curse of dimensionality presents two main challenges:

1. As the number of dimensions goes up, the computational power needed increases exponentially.

2. The level of bias in the model rises, and a larger sample is necessary to populate the new depth, due to the distance increasing between the data points.

Since computing power increases quickly, it becomes impractical to solve. Conventional machine learning models have to do some kind of dimensionality reduction, whereby the features of the high dimensional data are still being taken into consideration when estimating the model without suffering from problems of excessive computing needs. While there is a rise in computational power as the number of trees expand with larger datasets, generally, decision trees and as an extension random forest are relatively efficient compared to other machine learning models Nitze (2012). Due to the inherent properties of decision trees, where only one factor is considered at each splitting point, effectively this emulates dimensionality reduction, meaning relatively little computing power is needed.[9] Consequently, random forest along with boosted regression is widely accepted as well suited for non-parametric data.

The implication of the second challenge directly relates to kernel density estimation. As the number of dimensions increases, the space becomes more sparsely populated with data and the distance to the nearest neighbor increases. The consequence of this increase in distance is that the kernel density becomes overly fit to singular points of data. A few data points doesn't capture the overall dynamics and relationships at play, thus leading to the introduction of bias to the model. As the number of dimensions increases, the number of observations of the sample that can be considered outliers is generally expected to increase *Dimensionality Increase - an overview | ScienceDirect Topics* (2022). An illustration of how distance increases as dimensions are added is illustrated in figures 3, 4, and 5.

[9] for further reading into how the curse of dimensionality effects decision trees and see: Verleysen and François (2005) page 795

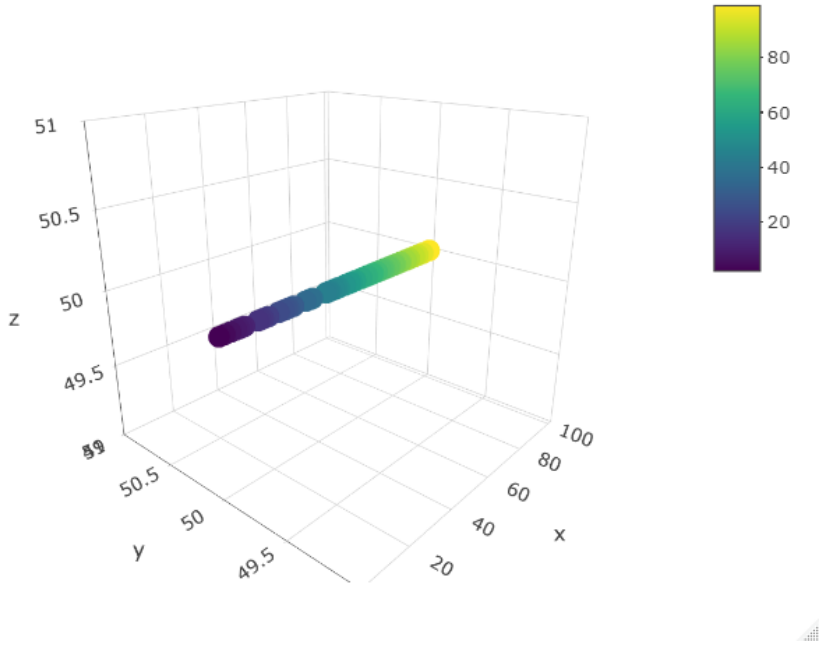

Figure 4: Illustration of distance for one-dimensional data.

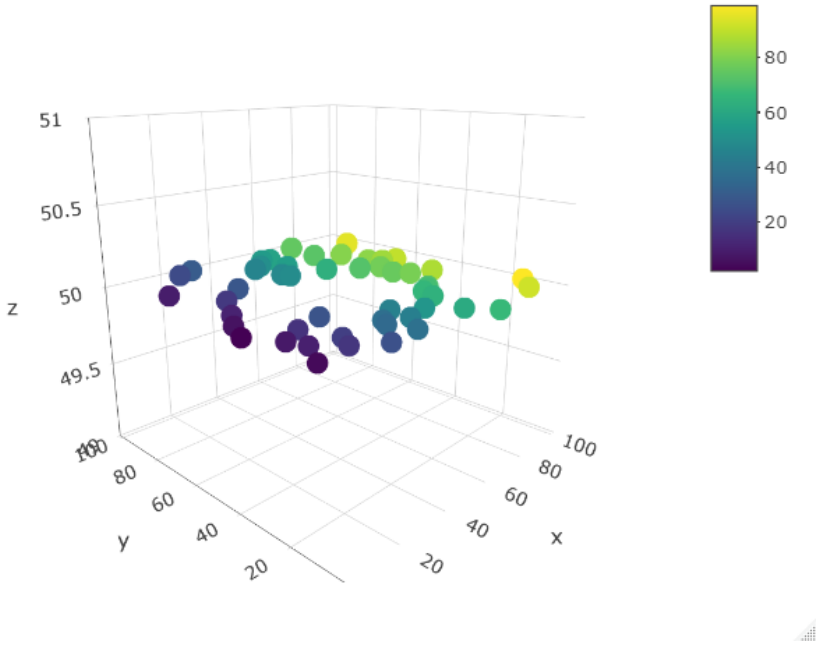

Figure 5: Illustration of distance for two-dimensional data.

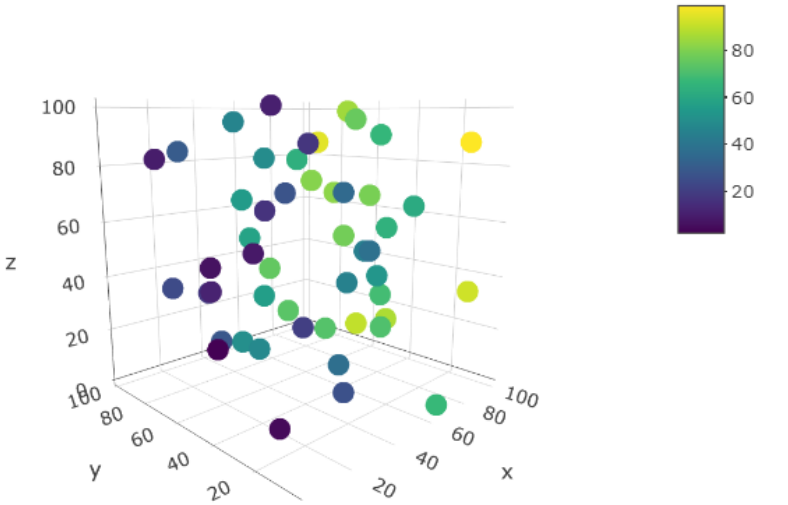

Figure 6: Illustration of distance for three-dimensional data.

4.5 Athey, Tibshirani, and Wager Generalized Random Forest

Athey, J. Tibshirani, and Wager's (2019) implementation of random forest mainly differs in its approach by implementing an alternative method for performing smoothing. Instead of using a kernel weighting function, their solution utilizes a dynamic adaptive weighing function to perform local maximum likelihood estimation. As outlined in 4.4, the main argument for why it could be disadvantageous to utilize a kernel weighting function is that as the number of dimension rises the covariant space also increases exponentially, sic "curse of dimensionality". This adaptive weighting function determines the weights by factoring in how often other points in covariant space appears on the same leaf as the chosen covariant vector of interest. Athey et al. argue that by using this alternative weighing method, there is significant reduction in bias for high dimensional data, while also at the same time reducing the computing need in such environments.

Athey et al. stipulate that given the data $(X_i, O_i) \in X \times O$, we can obtain the estimate for the local maximum likelihood parameter $\theta(x)$ and the nuisance parameter $\nu(x)$ that might be present in the data, through the equation shown below:

$$[H]\mathbb{E}\Big[\Psi\theta_{(x),\nu(x)}(O_i)|X_i = x\Big] = 0 \text{ for all } x \in X \tag{1}$$

Where $\psi(\cdot)$ is a given scoring function that indicates the sensitivity resulting from infinitesimal changes to model parameters. The shown formula is applicable to a diverse set of statistical problems.

Deriving forest-based local estimation, given that n number of independent and unbiased samples, are indexed as $i = 1, ..., n$. for each entry into the index we an observed value O_i that contains information useful for predicting $\theta(\in)$ as well as the covariates X_i. As in our case with non-parametric regression, each entry of O_i is just equal to its outcome, i.e. $O_i = Y_i$ for all cases where Y_i is a real number. The objective is to estimate solutions taking the form of equation 1. The method Athey et al. suggest for estimating the functions $\theta(x)$ is to start with defining a similarity weight $\alpha_I i(x)$ that describes how relevant each entry in the training dataset is when it comes to fitting $\theta(\in)$ at point x. After identifying relevancy, the chosen entry is fitted to an empirical version of equation 1, shown below:

$$[H]\left(\hat{\theta}(x), \hat{v}(x)\right) \in \text{atgmin}_{\theta,v}\left\{\left|\left|\sum_{i=1}^{n} {}_i(x)\psi_{\theta,v}(O_i)\right|\right|_2\right\} \tag{2}$$

When the equation shown above has a unique root, it is also the case that $\hat{\theta}(x), \hat{v}(x)$ solves $\sum_{i=1}^{n} {}_i(x)\psi_{\hat{\theta}(x),\hat{v}(x)}(O_i) = 0$ In a traditional random forest the weights $\alpha_I i(x)$ are obtained with a kernel weighing function. The adaptive weight is instead obtained by averaging neighborhoods as subgroups of trees. First, B number of trees are grown and indexed as $b = 1, ..., B$. The total number of datasets where the training examples overlap with x is given as $L_b(x)$. The function shown below obtain weights $\alpha_I i(x)$ based on I-th number of trees where the training example overlaps with x:

$$\alpha_{bi}(x) = \frac{1(\{X_i \in L_b(x)\})}{|L_b(x)|}, \alpha_i(x) = \frac{1}{B}\sum_{b=1}^{B} \alpha_{bi}(x)$$

The weight estimation equation shown above should add up to 1 and describe the forest adaptive neighborhood of the given example x. An illustration of weight allocated using this method is shown in figure 7. The lines indicate the points where the data is sectioned off into the parts that fall inside and outside the leaves. In this case, it means that the data parameters that most consistently end up in a given leaf are weighted the most; if the parameter falls outside the leaf, it's not weighted at all. As it can be seen in the bottom-most picture of figure 7, the aggregate is then found by adding up the weights of each decision tree. The result of using this adaptive weighing method is that the points that are strictly closer aren't necessarily the ones that are weighted the most.

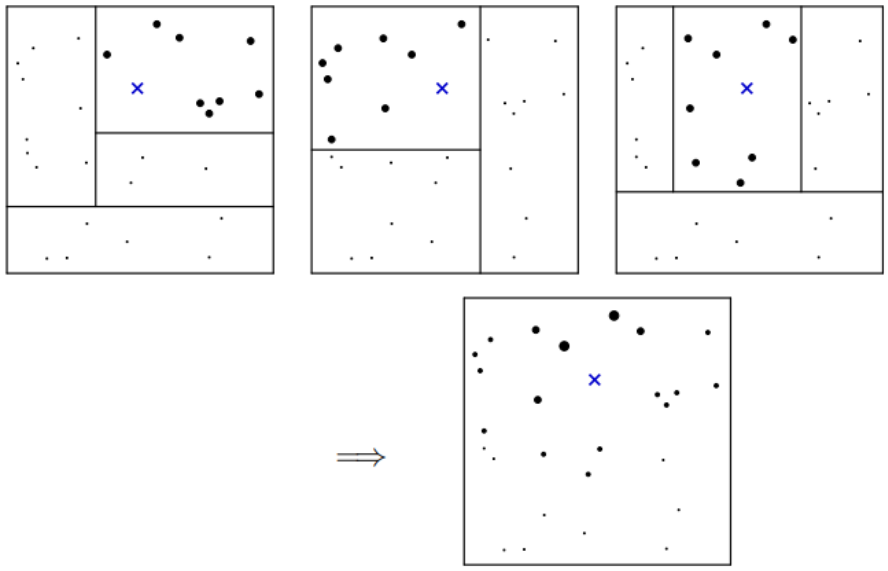

Figure 7: Resulting weight from adaptive weighting method, size of dots indicating allocated weight. Athey, J. Tibshirani, and Wager (2019)

4.6 Ghosal and Hooker Generalised Boosted Forests

Ghosal and Hooker's (2021) approach, like Breiman's and Athey et al, start with building a pure random forest, though they add another layer to the analysis by fitting an additional random forest to the residuals of the first random forest, supposedly creating a new meta-model that is better at reducing bias while keeping a good trade-off with variance.

The primary idea is performing gradient boosting to improve results, but instead of boosting a series of decision trees, the boosting is performed by training a new random forest on the residuals of a prebuilt random forest. By doing this the need for dataset shrinkage is avoided because results are not removed during the bias detection, and some samples might still be descriptive, even though they don't immediately appear that way.

Ghosal and Hookers addition builds on Breiman's version of random forest for regression shown below:

$$\hat{F}^{(0)}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x; Z^{0}_{I_b^{(0)}})$$

Like in Breimans research the equation works with a given dataset Z of size n, defined as $Z^{(0)}_{[n]} = (Z^{(0)}_1, (Z^{(0)}_2, ..., (Z^{(0)}_n)$ where $(Z^{(0)}_i = (Y_i, X_i)$ for $[n]$ is given as

$\{1, ..., n\}$. A random forest is fitted to the dataset, with a number of trees equal to B and $k < n$ number of sub-samples which are selected at random [10]. We let $T(x; Z_I^0)$ indicate the estimate of a given tree, where I is an individual subset of k.

Ghosal and Hooker modify Breiman's original by assigning weights $w_I^{(0)}$ at random to every $\binom{n}{k}$ subset. We define $w_I^{(0)}$ as taking a binary position; there is a probability of $B/\binom{n}{k}$ that it takes the value $(\binom{n}{k}/B)$ and a probability of $1 - B/\binom{n}{k}$ that it takes the value 0. Because the weights $w^{(0)_I}$ are distributed at random, they should be independent and identically distributed weights; the expected weight should be equal to 1 written as $\mathbb{E}(w_I^{(0)}) = 1$ and $Var(w_I^{(0)} = \binom{n}{k})B - 1$. Breiman's random forest estimation method can then be rewritten such that it takes the form shown below:

$$\hat{F}^{(0)}(x) = \frac{1}{\binom{n}{k}} \sum_{I \subseteq [n]:|I|=k} w_I^{(0)} T(x; Z_I^{(0)}$$

If we build the first random forest and obtain $\hat{F}^{(0)}$, residuals can be derived for $e_i = Y_i - \hat{F}^{(0)}(X_i)$. Then, as with the first random forest, we construct a new one with data constructed as $Z_{[n]}^{(1)} = (Z_1^{(1)}, (Z_2^{(1)}, ..., (Z_n^{(1)})$, where we replace Y_i with the residuals so we get $(Z_i^{(0)} = (e_i, X_i)$. The steps taken before are then repeated, and the second stage can be estimated as:

$$\hat{F}^{(1)}(x) = \frac{1}{\binom{k}{n}} \sum_{I \subseteq [n]:|I|=k} w_I^{(1)} T(x; Z_I^{(1)}$$

Simplifying, Ghosal and Hooker write that the one step boosted forest is a product of the first and second layer and can be written as:

$$\hat{F}(x) = \hat{F}^{(0)}(x) + \hat{F}^{1}(x)$$

Effectively, we end up with a random forest that can identify variable bias, being guided by training on the residuals of the first random forest.

[10] see 4.2 for more details on how the initial random forest is built

5 Model Construction and Evaluation

In the interest of reducing the scope of the following analysis, it was chosen to work with a limited subset of publicly traded companies. Due to constraints on data availability, this study is constrained to not include micro-cap stocks, micro-cap here are defined as companies with a market cap of less than 10 million dollars. Furthermore, only stocks listed on the American stock exchange NASDAQ are included. The data utilized consists of yearly financial reports from 2016 to 2021 if available[11]. 1213 companies were included, the dataset used for training consists of a total 5605 financial reports. Moreover, practical limitations on available compute power meant that unsupervised training was infeasible. 23 different metrics were chosen to be included in the final version of the model, see next section for selection criteria.

The data was retrieved using the IEX API. For further information about about data collection, please refer to the code repository available at the GitHub repository or see attached files.

5.1 Main Considerations and Selection Criteria

As described in the area of study, the stated goal is to evaluate the predictive capabilities of the three variations of the random forest models. In the case of "equity product valuation", specifically refers to the valuation of publicly traded companies. The future prices of a given stock won't be used as the predictor variable of value, because the number of shares floated varies drastically from company to company. The stock price doesn't on its own contain info about how much a given company is worth if we don't account for the number of outstanding shares. Therefore estimating the total market value or the total price of the entire company has to be considered, not just the price of an individual share. The total market value is defined as market capitalization (market cap) and is the best-suited criterion for measuring the valuation, as there isn't a need to adjust all the included metrics for outstanding float.

This analysis works with financial statements as the predictor variables; the given assumption is that the best long-term indicator of any given company's true value is the financial performance of that company. [12]

When selecting the market cap data for training and evaluation of random forest predictive capability, it was selected to work with the market cap one year from the financial reporting date. This time frame was chosen, as this was a considerably long enough time in the future that, in most cases, the market should have had time to adjust to a new price that reflected their intrinsic

[11] Because some of the included companies weren't publicly traded from 2016, not all included companies have financial statements going back that far.

[12] There are cases with outliers where this isn't the case. E.g. pharmaceutical companies that still have their product under development and have yet to report financial results. Though, in most cases, indicators like sales, income, expenses, etc. are the best available data for predicting the long-term value.

value. At the same time, one-year post reporting data is not so far out that data availability becomes problematic.

When constructing a supervised learning model, we need to consider what factors to include in the model. There are some main considerations for determining the relevancy of including a certain parameter.

- Is the selected parameter accurate and/or a relevant predictor?[13]
- Are the variables present across all or most of samples in the dataset?
- Are the included features already represented by some other variable?

For the first consideration about the relevancy of the included factor, two approaches can be considered. The first is to construct a simple regression model, where all factors are included, and then compare how much each factor weighs in the model. The challenge with this approach is that simple regression models, like linear regression, are not generally designed for non-parametric cases. There is a risk that deeper patterns and relations between variables in the data might be overlooked. A secondary approach could be to approach this from the theoretical point of view of an existing financial model. Existing financial models, such as the enterprise value model Fernando (2022) or discounted cash flow model Fernando (2021), give an insight into what the market broadly considers as impotent metrics for stock valuation.

Ultimately, it was chosen to take a purely practical approach to criterion selection. First, parameters were not taken into consideration if sufficient data wasn't present; if the reported factor wasn't universally reported by all companies, it was discarded[14]. Second, any parameter we didn't expect to have any direct impact on the financial performance was removed; examples of this are things like reporting date, financial ticker, fiscal year, etc. Finally, if a given variable is derived from some other metric or already represented by another more common financial performance metric, it's also discarded[15]. The final dataset contained a total of 23 different financial metrics. A full list of all the included metrics is listed in the code depository or can be viewed in figures 13, 14, and 15.

5.2 Model Training

The model's training consisted of two primary parts, in-bag, and out-of-bag analysis.

For the in-bag model, the three models are trained on the entirety of the dataset, with the market cap set as the dependent variable. After the models have been fitted to the training data, the same set of training data is fed back

[13] Do we expect there to be a causal or correlational relationship

[14] This cut the number of reported financial parameters down to 72, see dataset "financedata.csv"

[15] This removed things like EBITDA and net borrowings, as they are derived from other factors already included in the model

into the model, and the fitted models make predictions of market cap for each financial entry.

For out-of-bag, the model's training data is constricted and only exposed to half of the original dataset during training. Instead of feeding the models the same dataset during evaluation, as was the case with in-bag, the unseen out-of-bag data is used. Due to the fact that the models weren't trained on the out-of-bag data, it offers an advantage when evaluating machine learning models, such as random forest. Generally, it's preferred that the measured accuracy is a result of the identification of causal relationships and not due to the model being overly fit to the specific dataset. Using out-of-bag data for evaluation should give a deeper insight into how much of the accuracy can be contributed to generalization vs overfitting on the training data.

5.3 Model Evaluation Metrics

Model Evaluation Factors in-bag					
Models	RSMPE[16]	R^2[17]	PCC[18]	RRSE[19]	Runtime[20]
Breiman model	0.317	0.969	0.984	0.207	57.638
Athey et al. model	0.663	0.773	0.879	0.745	10.543
Ghosal and Hooker	0.300	0.901	0.949	0.374	NA

Table 1: Table of in-bag Performance

In table 1, evaluation metrics of model performance trained on the in-bag dataset can be seen for the three models. Runtime for the in-bag training is considerably higher because of the larger dataset[21]. Table 2 shown below displays out-of-bag performance, for the three tested models. It's noticeable how there is a visible decline in performance for all three models.

Model Evaluation Factors out-of-bag					
Models	RSMPE	R^2	PCC	RRSE	Runtime
Breiman model	0.896	0.607	0.779	1.494	14.556
Athey et al.	1.042	0.581	0.762	1.345	4.936
Ghosal and Hooker	0.369	0.833	0.902	1.024	NA

Table 2: Table of out-of-bag Performance

[16] Root mean square percentage error, see equation 17 in appendix.
[17] Coefficient of determination, see equation 18 in appendix.
[18] Pearson correlation coefficient, see equation 19 in appendix.
[19] Root Relative Squared Error, see equation 20 in appendix.
[20] Measured in seconds.
[21] Ghosal and Hooker's runtime is non-applicable because their implementation in R uses various methods for speeding up the runtime, like multi-thread handling.

5.4 Performance of Models In-Bag

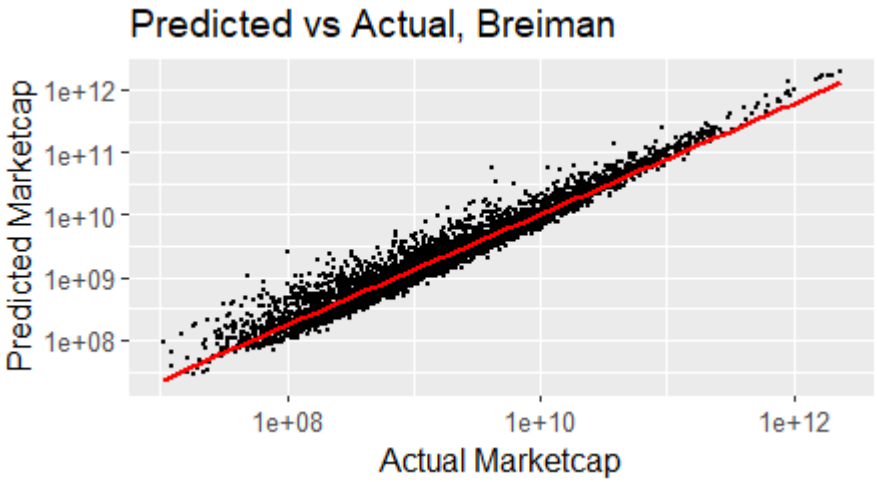

Figure 8: Comparison of predicted vs actual values for in-bag forecast, model trained using Breiman's orignal implementation of random forest.

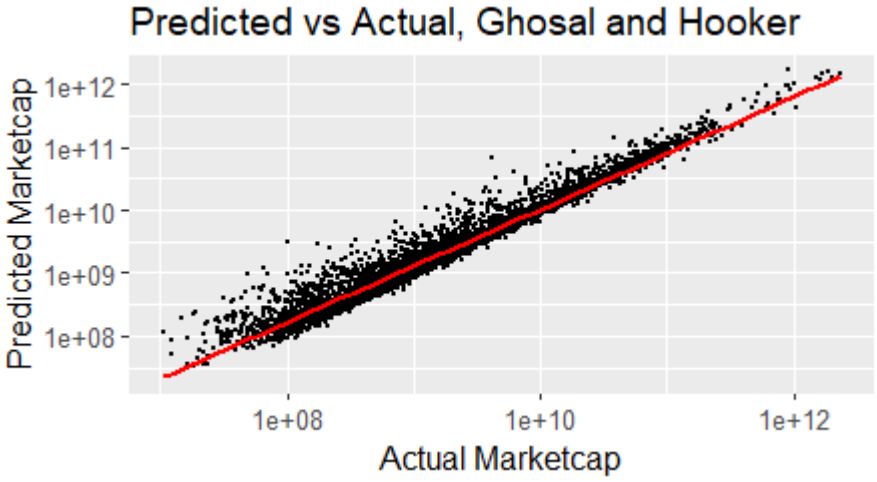

Figure 9: Comparison of predicted vs actual values for in-bag forecast, model trained using Ghosal and Hooker's random forest variant.

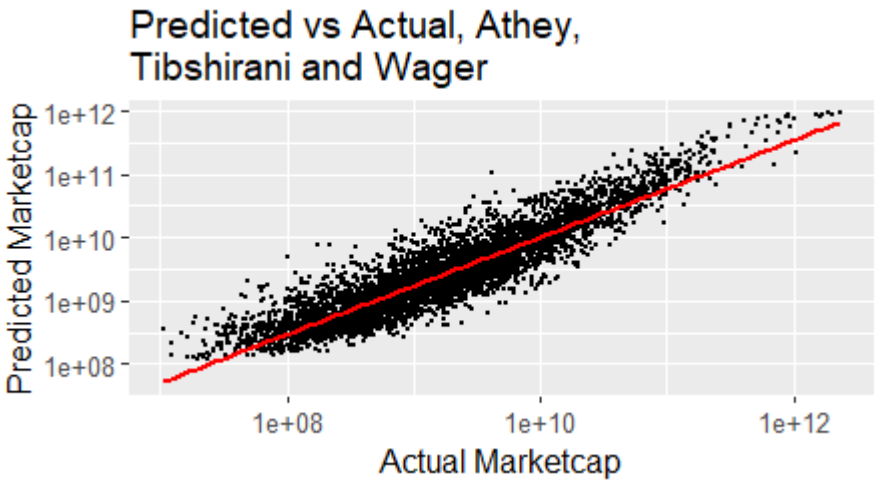

Figure 10: Comparison of predicted vs actual values for in-bag forecast, model trained using Athey, Tibshirani and Wager's random forest variant.

5.5 Performance of Models Out-Of-Bag

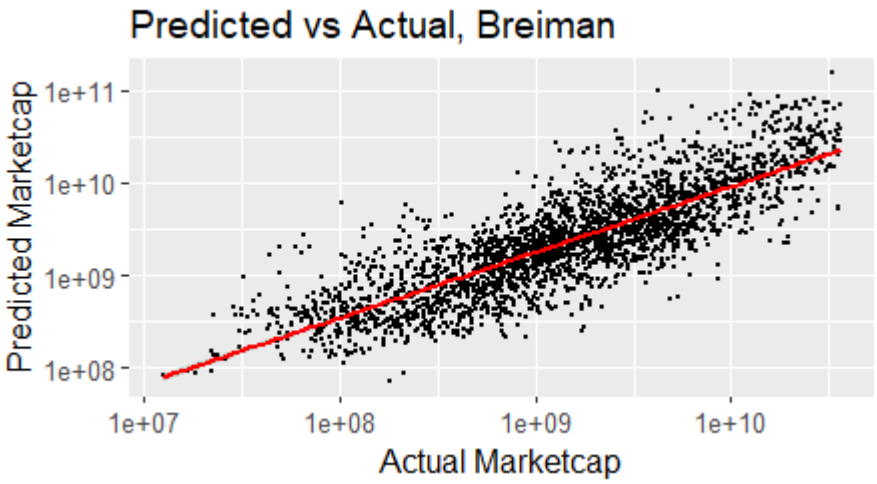

Figure 11: Comparison of predicted vs actual values for out-of-bag forecast, model trained using Breiman's orignal implementation of random forest.

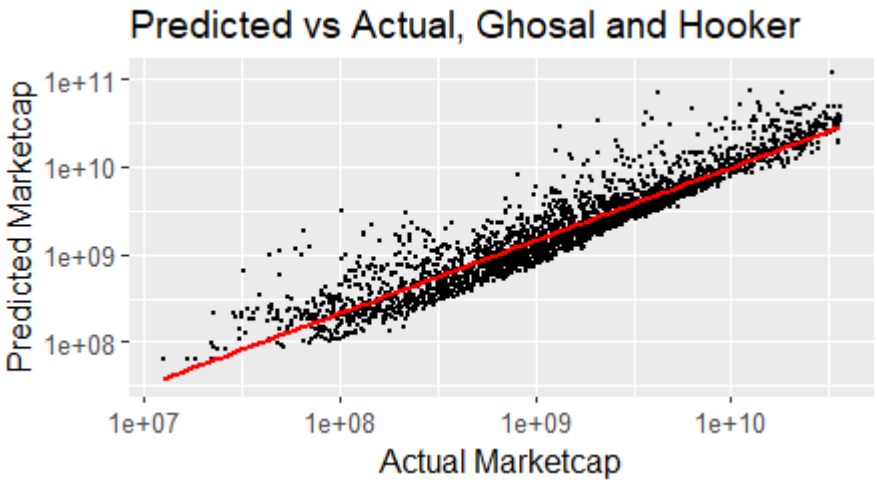

Figure 12: Comparison of predicted vs actual values for out-of-bag forecast, model trained using Ghosal and Hooker's random forest variant.

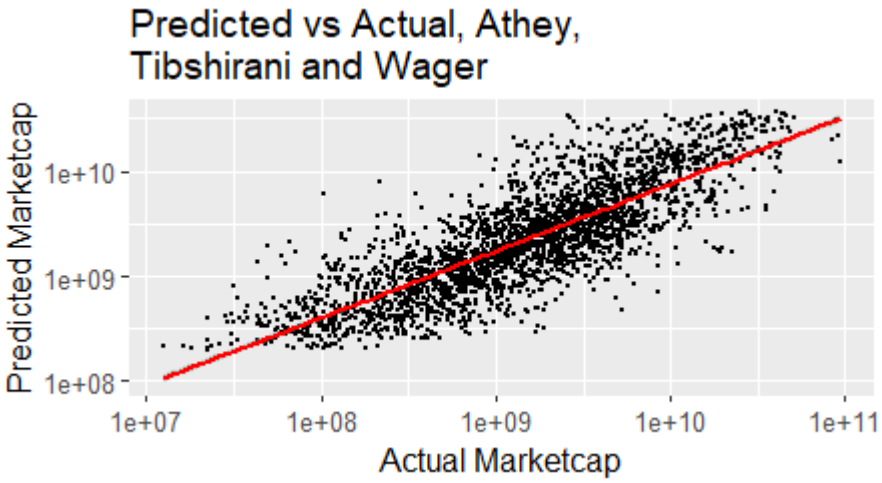

Figure 13: Comparison of predicted vs actual values for out-of-bag forecast, model trained using Athey, Tibshirani and Wager's random forest variant.

5.6 Global Importance Factors

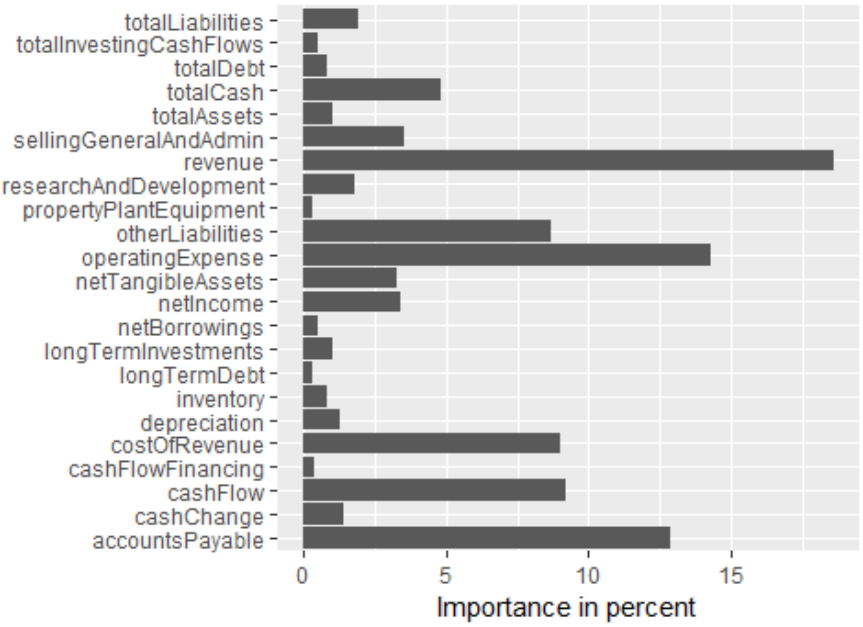

Figure 14: Global importance of factors for Breiman's model in percent, indicating how much the various parameters weigh on average.

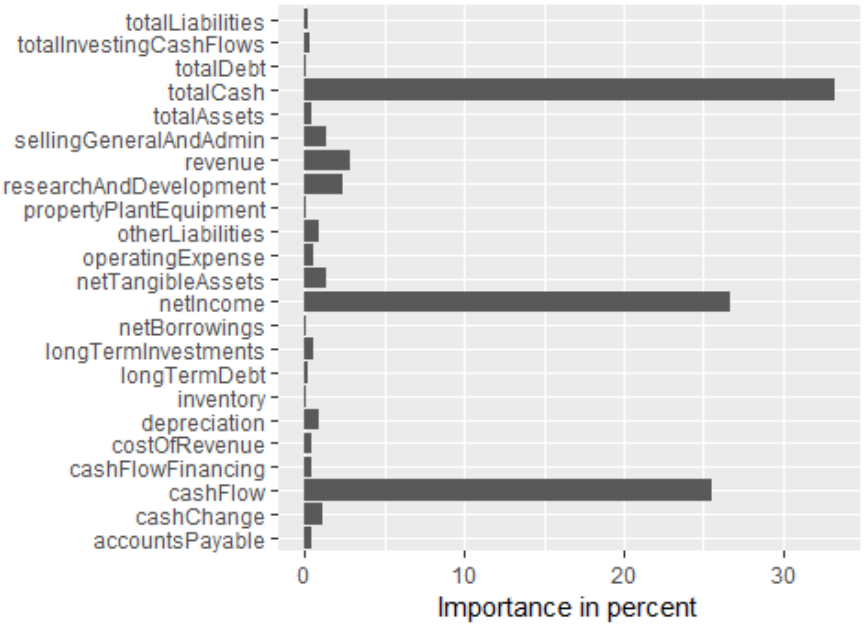

Figure 15: Global importance of factors for Athey, Tibshirani and Wager model in percent, indicating how much the various parameters weigh on average.

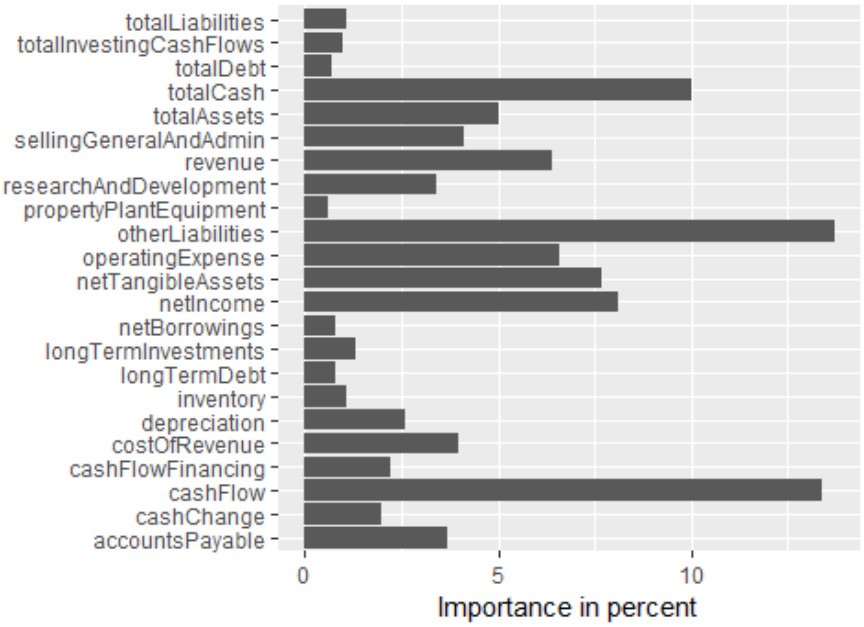

Figure 16: Global importance of factors for Ghosal and Hooker's model, indicating how much the various parameters weigh on average.

5.7 Results

Accuracy

As listed in table 1, all three variations of random forest generally performs well at conducting in-bag forecasting. Ghosal and Hooker achieves result accuracy comparable to Breiman's model, with Breiman's model having only slightly better R-squared values and a higher Pearson correlation coefficient, at 0.969 and 0.984 respectively, vs Ghosal and hooker's 0.901 and 0.949. Athey et al. model underperforms the other two models on almost all metrics. This is especially apparent for root square mean percentage error. It can be seen that Breiman and Ghosal and Hookers on average are within 30% of the actual values, while Athey et al. sees a significant rise to 66% error in prediction.

For out-of-bag, we see that all models drop in accuracy, though some more than others. The largest drops in precision can observed for Breiman and Athey et al. For Breiman RSMPE rise to 89.6% and Athey et al. rises to 104.2%. Even though Athey et al. performs worse than Breiman, the relative drop in performance is not as large; this might be due to the fact that during the initial training Athey et al. didn't overfit to the dataset, but nevertheless, Athey still underperforms for out-of-bag even if Breiman did overfit. Ghosal and Hooker surprisingly vastly outperforms the other two models on out-of-bag predictions, only experiencing a small marginal drop in performance, with RSMPE rising to

36.9%, the largest drop for Ghosal and hooker is seen in root relative squared error.

By comparing Figures 8, 9 and 10, that show prediction vs actual market cap for in-bag, with figures 11, 12 and 13, that show out-of-bag results. We can see clearly how the drop in prediction error, leads to much larger spread for the out-of-bag test.

Computation need

Though Athey et al. generally underperform on accuracy, there can be seen a significant improvement in compute time, with Athey et al. offering a reduction in computation need for training on both dataset sizes. For in-bag, the largest increase is seen where Athey et al. has a almost sixfold decrease in computation need compared to Breiman, 57.638 seconds vs 10.543 seconds. For out-of-bag, there is a seen a slightly smaller improvement, with Athey et al. only being three times as efficient, at 14.556 seconds vs 4.936 seconds. The fall between the two tests might be due to fact that the computation need doesn't rise linearly with dataset size. So it might be the case with larger datasets that Athey et al. sees even larger increases in runtime compared to Breiman. The run time for Ghosal and Hooker is non-applicable, as their implementation uses various methods to speed up the process like multi-threading.

Global Importance Factors

Global importance factors indicate how much the individual parameters weigh on average, across all trees. While this doesn't give an exact image of how the parameters interact with each other, it does give a general idea of how often the included parameters were considered important on average. There can be seen some recurring results, for example, in all three models, "Cashflow" score highly in importance. Cashflow denominates the total change in cash compared to the previous period. Cash flow is often used by investors as one of the prime indicators of a company's performance. It can be seen that, in general, the three models identify the same relationships, though there are major deviations, especially with Athey et al. That Athey's global importance factor weights deviate so much from the others probably is an indication of strong misallocation. Athey et al. global importance factors doesn't represent the real world relations between the different financial performance metrics, as such it's likely that a lot of minor relationships are overlooked or oversimplified.

Even so, it's debatable how much weight should be placed on these results, as the models are non-linear; there might be unique situations where the weighing locally is radically different. It's not apparent when taking the average weight, such as here, what is strictly indicative of the real relationships between the different factors.

6 Discussion of Results

While it's apparent from the previous evaluation phase, that Ghosal and Hooker outperforms their counterparts quite strongly. This chapter relates those results to the challenges first mentioned in the area of study. As such before any robust conclusions can be drawn, an examination of the risk of bias and overfitting has to be performed. We also have to take into consideration whether any flaws in the methodology are influencing our results significantly.

6.1 Reflection of Results - Risk of Misspecification and Bias

It's evident for both tests that Athey et al. were distinctly the worst performing models, we contribute this to the fact that replacing the kernel weighing function with the adaptive weighing function dislocating weight to a select few model parameters. As illustrated in figure 15, There was placed an overly high weight on the three parameters, "totalCash", "netincome" and "cashFlow". The reasons these three were excessively weighted was because they consistently appeared in the same leaf, i.e. Athey et al. successfully identified that the common denominator for large companies was that they had high income, large positive cash flow, and a large amount of cash on hand. While these three indicators are commonly regarded as good measures of a company's performance, and it could be argued that Athey et al. were better at identifying them than the other two models, the adaptive weighing function doubtlessly under-allocated weight to other factors. As further shown in figure 15 the model assigned almost no weight to aspects like total debt or inventory. We argue that in trying to achieve better bias reduction Athey et al. effectively over-corrected, to such a degree that the model could no longer be considered nonparametric.

As we talked about in section 5.7 Ghosal and Hooker slightly underperformed Breiman for in-bag results, though Ghosal and Hooker experienced almost no drop in performance for out-of-bag results. Recall that the main addition by Ghosal and Hooker was using the residuals from a random forest to build a secondary random forest, which estimates the biases present in the first given model. This is consistent with why Ghosal and hooker could appear to underperform Breiman for in-bag analysis, it's likely the case that the bias estimation from Ghosal and Hooker's model made it more robust against overfitting. That is to say, Breiman only shows better accuracy because of overfitting, while Ghosal and Hooker successfully generalized. Due to the fact that Breiman overfitted to the dataset there can be seen a significant drop in accuracy when comparing to out-of-bag predictions, conversely, Ghosal and Hooker only experience a slight drop in performance. However this slight drop between the in-bag and out-of-bag results, indicate that there is still some slight bias present. Nevertheless, our results indicate that Ghosal and Hooker's extension can be considered a very versatile improvement in random forest over Breiman's original algorithm.

6.2 Other Considerations

During conventional development and evaluation of machine learning models, such as with Random Forest, the model would continuously be modified throughout development as to best satisfy the intended goal. Though in this case where the goal in and of itself isn't to achieve a certain model accuracy but to compare three alternative models abilities. It wouldn't necessarily make sense to conduct conventional fine-tuning of the individual model's, for local optimization, as that would directly impact model performance.

With that taken into consideration, we still find that there is room for improvement in the methodology with which this study employed. As mentioned in the previous section we did not test on varying sample sizes with the exception of the in-bag and out-of-bag tests. It might have been the case that some of the models would have produced better results, relatively speaking, if we had worked with a larger dataset. Athey et al. evaluate their addition by conducting a random forest analysis in the same manner as Angrist and Evan's (1998) study on "The Effect of Child Rearing on Labor-Force Participation". Their dataset consisted of $n = 334,535$ mothers with two or more children, with 16 different variables included for each sample. In comparison, our study only consisted of $n = 5605$ financial reports, of 23 included predictor variables. Though it's debatable how much we should equate predictive failure to sample size, as Athey et al. state in the opening lines of their article:

> *"We propose generalized random forests, a method for nonparametric statistical estimation based on random forests (Breiman, 2001) that can be used to fit any quantity of interest identified as the solution to a set of local moment equations"*-Athey, J. Tibshirani, and Wager (2019)

A secondary problem is that it would also be difficult to obtain a comparatively large set of financial rapports to the one that they tested on. Even so, we consider it outside the possible scope of this study to expand the model so drastically, and as such we will consider the apparent outcome.

6.3 Verdict

In regards to answering the main question set out in the area of study,

> *"How well does the recent developments in Random Forest modeling improve predictive capabilities compared to Breimans original algorithm"*

We can summarize our findings as follows:

Breiman's original model initially appeared to be very accurate for in-bag predictions, performing better than the other two models. Though in reality, Breiman's results from out-of-bag predictions showcased that the model was strongly afflicted by variable bias, and as result, overfitted to the dataset.

Athey et al. addition of using an adaptive weighing function as an alternative to a kernel weighing largely doesn't improve the model's ability to consistently correct for bias and generally overcorrects when trying to reduce overfitting. On net Athey et al. method can not be considered an improvement over Breiman's when applied in the practical application of predicting financial valuations from financial reports. Although there might be alternative outcomes for different sample sizes, or in other applications.

Contrasting this was Ghosal and Hooker's addition, which showed a large step up in performance stemming from a refined ability to generalize the model, through a stronger aptness to identify variable bias.

Overall our findings are that Ghosal and Hooker can widely be considered an strong improvement over Breiman's original algorithm, while Athey, J. Tibshirani, and Wager's generalized random forests method weakens the model and can not be considered an improvement, at least in our case of financial valuation modeling.

7 Conclusion

This thesis carried out an empirical analysis, to investigate the relative performance of three variations of the machine learning model random forest. This was done by using using the models for predicting publicly-traded companies' future valuations, from their financial reports. The three selected models were Breiman's original Random forest variant, Athey, J. Tibshirani, and Wager's generalized random forest, and Ghosal and hookers one-step boosted forest. Athey et al. and Ghosal and Hooker's additions to the original random forest were both attempts at correcting for variable bias, which is often observed to subvert results for high dimensional data, leading to overfitting.

To ensure the robustness of the given results, the analysis was carried out on both in-bag and out-of-bag data. For in-bag the same data with which the models were trained, was used for evaluation, contrasting was out-of-bag where the models were evaluated using unseen data. In both cases, the goal of the models was to accurately approximate the market cap of a given company, based on its financial performance.

The main findings from the analysis showed that for the in-bag test Breiman's appeared to be the best performing model, almost matching in performance was Ghosal and Hooker's one-step boosted forest, and trialing behind in performance was Athey et al. For out-of-bag performance, where the models were evaluated using data not used during training, the results had changed such that Ghosal and Hooker' method was now the best performing, only seeing a slight decrease in performance compared to in-bag. Both Breiman and Athey et al. saw a drastic decline in performance, with the overall accuracy halving. Ghosal and Hooker saw a relatively low rate of prediction error, with the mean prediction only deviating around 36.9% from the actual value. With Breiman and Athey et al. seeing a mean prediction error of 89.6% and 104.2% respectively.

The implication of the overall decrease between in-bag and out-of-bag was attributed to initial overfitting of the models to the dataset. Breiman, which saw the biggest relative decrease between the two tests didn't contain any additional bias correction, as was the case with Athey et al. and Ghosal and Hooker, thereby leading to overfitting.

While Athey et al. applied their alternative adaptive weighing function, to better correct for bias when performing localized regression. It was found that this method lead it to being overly aggressive in excluding samples, thereby leading to overgeneralization. As a consequence Athey et al. was the worst performing model for both tests, with the exception of computing need, where a significant reduction was seen.

Ghosal and Hooker who saw the smallest decline in performance between in-bag and out-of-bag were considered the best model overall. This increase in performance was achieved by building an additional random forest using the residuals of the first. By being guided by the first random forest model Ghosal and Hooker's model could more consistently estimate what samples attributed

to bias, and thereby strongly improve its ability to generalize to the real features of the data.

Our final verdict is that Breiman's original random forest algorithm was generally poorly performing when trying to predict financial valuations, being prone to overfitting and strongly affected by variable bias. Athey, J. Tibshirani, and Wager's model proved a poor improvement over Breiman's model, going from overfitting to underfitting to the dataset. Lastly, Ghosal and Hooker showed a very significant improvement and can be considered the Superior model, when random forest is used for predicting valuations of publicly traded companies.

References

Athey, Susan, Julie Tibshirani, and Stefan Wager (Apr. 2019). „Generalized random forests“. In: *The Annals of Statistics* 47.2, pp. 1148–1178. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/18-AOS1709. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-47/issue-2/Generalized-random-forests/10.1214/18-AOS1709.full (visited on 03/15/2022).

Breiman, Leo (2001). „Random forests“. In: *Machine learning* 45.1, pp. 5–32.

Breiman, Leo et al. (Oct. 2017). *Classification And Regression Trees*. New York: Routledge. ISBN: 9781315139470. DOI: 10.1201/9781315139470.

Dimensionality Increase - an overview | ScienceDirect Topics (2022). URL: https://www.sciencedirect.com/topics/computer-science/dimensionality-increase (visited on 05/24/2022).

Fernando, Jason (Dec. 2021). *Discounted Cash Flow (DCF)*. en. URL: https://www.investopedia.com/terms/d/dcf.asp (visited on 05/30/2022).

— (Jan. 2022). *Understanding Enterprise Value (EV)*. en. URL: https://www.investopedia.com/terms/e/enterprisevalue.asp (visited on 05/30/2022).

Geoffrey, Watson (1964). „Smooth regression analysis“. In: *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372. URL: http://www.jstor.org/stable/25049340.

Ghosal, Indrayudh and Giles Hooker (Apr. 2021). „Boosting Random Forests to Reduce Bias; One-Step Boosted Forest and Its Variance Estimate“. In: *Journal of Computational and Graphical Statistics* 30.2, pp. 493–502. ISSN: 1061-8600. DOI: 10.1080/10618600.2020.1820345. URL: https://doi.org/10.1080/10618600.2020.1820345 (visited on 03/18/2022).

Grant, Mitchell (2022). *How Nonparametric Statistics Work*. en. URL: https://www.investopedia.com/terms/n/nonparametric-statistics.asp (visited on 04/06/2022).

greedy algorithm (2022). URL: https://xlinux.nist.gov/dads//HTML/greedyalgo.html (visited on 04/01/2022).

Hamza, Mounir and Denis Larocque (Aug. 2005). „An empirical comparison of ensemble methods based on classification trees“. In: *Journal of Statistical Computation and Simulation* 75.8, pp. 629–643. ISSN: 0094-9655. DOI: 10.1080/00949650410001729472. URL: https://doi.org/10.1080/00949650410001729472 (visited on 04/06/2022).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009a). „Additive Models, Trees, and Related Methods“. en. In: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Ed. by Trevor Hastie, Robert Tibshirani, and Jerome Friedman. New York, NY: Springer, pp. 295–336. ISBN: 9780387848587. DOI: 10.1007/978-0-387-84858-7_9. URL: https://doi.org/10.1007/978-0-387-84858-7_9 (visited on 03/23/2022).

— (2009b). „Random Forests“. en. In: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Ed. by Trevor Hastie, Robert Tibshirani, and Jerome Friedman. New York, NY: Springer, pp. 587–604. ISBN:

9780387848587. DOI: 10.1007/978-0-387-84858-7_15. URL: https://doi.org/10.1007/978-0-387-84858-7_15 (visited on 03/23/2022).

Ho, Tin Kam (Aug. 1995). „Random decision forests“. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 1, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

Nadaraya, E. A. (Jan. 1964). „On Estimating Regression“. In: *Theory of Probability & Its Applications* 9.1, pp. 141–142. ISSN: 0040-585X. DOI: 10.1137/1109020. URL: https://epubs.siam.org/doi/10.1137/1109020 (visited on 05/24/2022).

Rendering forest diagram in LaTeX, Rendering forest diagram in LaTeX (2022). *Rendering forest diagram in LaTeX*. URL: https://tex.stackexchange.com/questions/418745/issues-with-a-global-setting-for-forest (visited on 03/24/2022).

Scornet, Erwan (Sept. 2015). „Random forests and kernel methods“. In: arXiv: 1502.03836. URL: http://arxiv.org/abs/1502.03836 (visited on 05/24/2022).

Tang, Cheng, Damien Garreau, and Ulrike von Luxburg (2018). „When do random forests fail?“ In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc. URL: https://papers.nips.cc/paper/2018/hash/204da255aea2cd4a75ace6018fad6b4d-Abstract.html (visited on 03/18/2022).

Verleysen, Michel and Damien François (2005). „The Curse of Dimensionality in Data Mining and Time Series Prediction“. en. In: *Computational Intelligence and Bioinspired Systems*. Ed. by Joan Cabestany, Alberto Prieto, and Francisco Sandoval. Berlin, Heidelberg: Springer, pp. 758–770. ISBN: 9783540321064. DOI: 10.1007/11494669_93.

Yen-Chi, Chen (2018). „Density Estimation: Histogram and Kernel Density Estimator“. In: p. 4. URL: http://faculty.washington.edu/yenchic/18W_425/Lec6_hist_KDE.pdf.

8 Appendix

Due to space limitations it was unpractical to include all the code and data used for this thesis in the appendix. The code is available to view at the GitHub repository or alternatively see submitted files at digitaleksamen.aau.dk

8.1 Equations

$$RMSPE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \Delta X_{rel,i}^2 \cdot 100\%}$$

Figure 17: RMSPE

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2}$$

Figure 18: R^2 equation

$$\rho X, Y = \frac{cov(X, Y)}{\sigma X \sigma Y}$$

Figure 19: Pearson correlation

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}}$$

Figure 20: RRSE equation