University of Bonn

Master Programme in Economics

Random Forest for Classification Problems

Submitted by

Burak Balaban

Arkadiusz Modzelewski

Raphael Redmer

Supervisor: Prof. Dr. Dominik Liebl

January 5, 2020

CONTENTS

ABSTRACT

As a non-parametric estimation tool, decision trees attract attention in the economics literature. Yet, decision trees suffer from high variance and, for prediction purposes higher variance seems to be a crucial problem, thus, several improvements were proposed such as bootstrap aggregation, boosting and most importantly random forests. In this project, while the main focus is being on the random forest. The elements of statistical learning by [6] [14] [11], [9] and as expected [4] are the main literature that will be utilized in this project.

To explain the concept of random forests in full extent, primarily decision trees should be discussed. Exploiting the main idea and struggles with bias-variance trade-off, random forests' importance can be emphasized as a more stable prediction tool [11]. Conceptual comparison of random forests with bagging and boosting can deliver a better understanding of its unique features as [8] shows in a similar fashion. To get a further understanding, random forests' estimation process can be mathematical explained [1] and likewise, examining the consistency of estimator and showing the properties can be included [3], [5]. Also, variable importance in the tree growing process is another area that needs to be delved into [7] and [10].

1

INTRODUCTION

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2

DECISION TREE

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

2.1 MAIN IDEA

The Decision Tree is a non-parametric supervised learning method used for classification and regression. It predicts the response with a set of if-then-else decision rules derived from the data. The deeper the tree, the more complex the decision rules and the closer the model fits the data. The decision tree builds classification or regression models in the form of a tree structure. Each node in the tree further partions the feature space into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and terminal nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor is called the root node. Decision trees can handle both categorical and numerical data.

An example of such a tree is depicted below in figure 1.

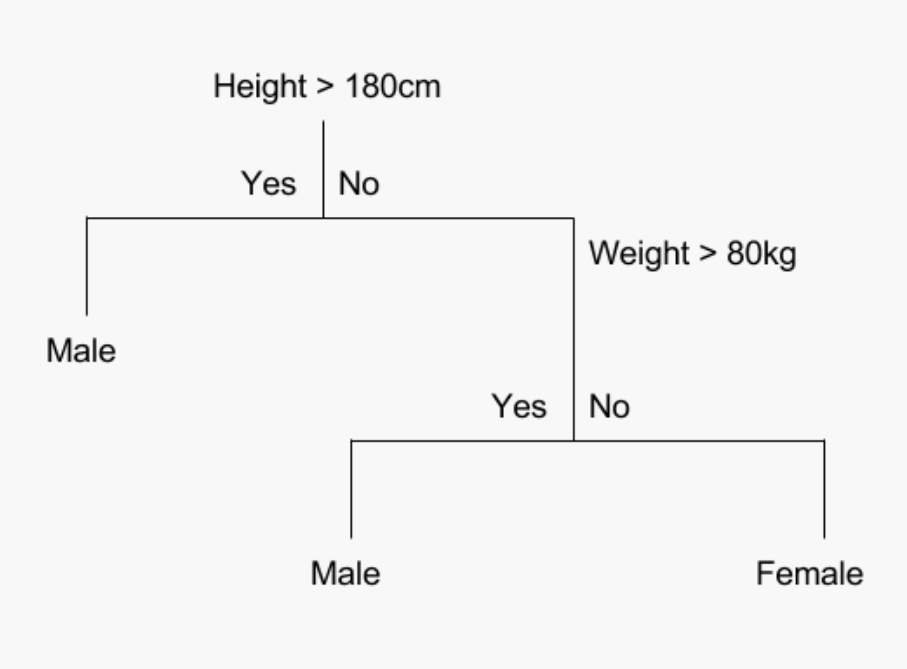

Figure 1: *Given a data set with two features height and weight, and gender as the target variable, this example tree stratisfies the two-dimensional feature space into three distinct subset each represented by the terminal nodes at the bottom. The stratification occurs at the two deciding nodes depending either on whether its height is above 180 cm and or its weight is above 80kg.*

2.2 TREE BUILDING PROCESS

This chapter describes the CART algorithm for tree building as specified in [2]. The basic idea of tree growing is to choose a split among all the possible splits at each node so that the resulting child nodes are the "purest". In this algorithm, only univariate splits are considered. That is, each split depends on the value of only one predictor variable. All possible splits consist of possible splits of each predictor.

A tree is grown starting from the root node by repeatedly using the following steps on each node (also called binary splitting)

(i) **Find best split s for each feature X_m:** For each feature X_m, there exist $K-1$-many potiential splits whereas K is the number of different values for the respective feature. Evaluate each value $X_{m,i}$ at the current node t as a candidate split point (for $x \in X_m$, if $x \leq X_{m,i} = s$, then x goes to left child node t_L else to right child node t_R). The best split point is the one that maximize the splitting criterion $\Delta i(s,t)$ the most when the node is split according to it. The different splitting criteria will be covered in the next chapter.

(ii) **Find the node's best split:** Among the best splits for each feature from Step (i) find the one s^*, which maximizes the splitting criterion $\Delta i(s,t)$.

(iii) **Satisfy stopping criterion:** Split the node t using best node split s^* from Step (ii) and repeat from Step (i) until stopping criterion is satisfied.

2.2.1 *Splitting criteria*

Since we are only concerned with classification, Y is categorical. The original CART algorithm uses Gini and Twoing as purity measures for the splitting criterion. However, implementations of the algorithm such as Python's sklearn package also contain entropy and misclassification rate as measures of impurity.

For a give learning sample L for a J class problem, let N_j be the number of instances $\{x,y\}$ belonging to in class j.

In node t, let $N(t)$ be the total number of instances with $\{x,y\} \in t$ and $N_j(t)$ the number of class j cases in t. The proportion of the class j instances in the sample L falling into t is $N_j(t)/N_j$. For a given set of priors, $\pi(j)$ is interpreted as the probability that an instance belongs to class j.

At node t let the probabilities $p(j,t)$, $p(t)$ and $p(j|t)$ be estimated by using Thus, let

$$p(j,t) = \frac{\pi(j)N_j(t)}{N_j} \tag{1}$$

be the estimate for the probability that na instance will both be in class j and fall into node t. Therefore, the estimate for the probability that any instance falls into node t is defined by

$$p(t) = \sum_j p(j,t), \tag{2}$$

The estimate $p(t)$ for the probability that an instance belongs to class j given that it falls into node t is defined by

$$p(j|t) = \frac{p(j,t)}{p(t)} = \frac{p(j,t)}{\sum_j p(j,t)}. \tag{3}$$

It holds that the conditional probability $p(j|t)$ must satisfy

$$\sum_j p(j|t) = 1 \tag{4}$$

Let $i(t)$ be an impurity measure evaluated at note t. Then, the decrease of impurity (i.e. the splitting criterion) is defined as

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R), \tag{5}$$

where p_L and p_R are probabilities of sending a case to the left child node t_L and to the right child node t_R respectively. They are estimated as $p_L = p(t_L)/p(t)$ and $p_R = p(t_R)/p(t)$.

As already stated abobe, the goal is to maximize $\Delta i(s,t)$ In the following, different measures for impurity will be presented.

Gini Measure

The Gini impurity measure is defined as

$$i(t) = \sum_j p(j|t)(1 - p(j|t)) = 1 - \sum_{j=1}^{J} p_j^2 \tag{6}$$

The intuition behind this measure is to assign nodes for which its probabilities are more skewed towards a particular group a higher value. Conversely, if a node has more balanced distribution, then $i(t)$ will turn out to be lower. For example, in the case of $J = 2$, $i(t)$ will be maximized with $p(j|t) = 0.5$ for $j = 1, 2$.

Information Entropy

The Entropy measure from information theory is defined as

$$i(t) = \sum_j p(j|t) log(p(j|t)) \tag{7}$$

which measures the average rate at which information is produced by a stochastic source of data . Thus, it can also be used for measuring impurity.

Rate of Misclassification

The rate of misclassification is defined as

$$i(t) = 1 - \max_{0<j\leq J} p(j|t). \tag{8}$$

which measure the proportion of instances node t not belonging to the dominant group in t.

2.3 BIAS-VARIANCE TRADE-OFF

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.3.1 *Bagging and boosting*

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3

RANDOM FOREST

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.1 MAIN IDEA AND ILLUSTRATION

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.2 MATHEMATICAL EXPLANATION AND CONSISTENCY

Let $D = (x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$ be the set, we want to train our model on and $T_{D,\theta}$ decision tree produced by using the set D and parameters θ. We assume that D is countable which normally is the case especially for Y values although replacing sums with integrals can extent the analysis and provide results for uncountable sets as well (Kohavi, 1996). As mentioned earlier, Random Forest classifier selects a bootstrapped subset of observations denoted as $\boldsymbol{D'}$ and grow the decision tree with only a subset of regressors. Repeating this tree growing process B times gives a Random Forest Estimator. Assume x^* is the value that we want to predict its class, there are two rules that can be used to get the prediction; majority voting and soft voting (Louppe, 2014; Zhou,2012).

In majority voting, after getting every trees prediction denoted as $\hat{T}(x^*)$ final prediction is the class that gets most votes from trees;

$$\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x^*) = \underset{c \in Y}{argmax} \sum_{b=1}^{B} 1(\hat{T}_b(x^*) = c) \tag{9}$$

In soft voting, probability estimates of a tree denoted as $\hat{p}_{D,\theta_b}(Y = c|X = x^*)$ is estimated and after all are averaged and most likely class is predicted;

$$\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x^*) = \underset{c \in Y}{argmax} \frac{1}{B} \sum_{b=1}^{B} \hat{p}_{D,\theta_b}(Y = c|X = x^*) \tag{10}$$

As mentioned in Breiman(1994), aformentioned two voting procedures provides similar results, yet using soft voting can provide smoother class probability estimates and be exploited in a deeper analysis setting such as certainty estimates investigation (Louppe, 2014).

3.2.1 *Properties*

The generalization error which also called test error or the expected prediction error of $T_{D,\theta}$ is;

$$\boldsymbol{Err}(T_{D,\theta}) = \mathbb{E}_{X,Y}\{L(Y, T_{D,\theta}(X))\} \tag{11}$$

where L is the loss function measuring the difference between its two arguments. Since we focus on classification setting, the zero-one loss function is our interest, however, to get a better understanding, widely used in regression type predictions, the squared loss function will be examined first. The bias-variance decomposition of both functions are similar and follow the same dynamics(Domingos, 2000). The squared loss function can be defined as

$$L(Y, T_{D,\theta}(x)) = (Y - T_{D,\theta}(x))^2 \tag{12}$$

while the zero-one loss function is

$$L(Y, T_{D,\theta}(X)) = 1(Y \neq T_{D,\theta}(X)) \tag{13}$$

The expected prediction error for both functions

$$\boldsymbol{Err}(T_{D,\theta}) = \mathbb{E}_{X,Y}\{(Y - T_{D,\theta}(x))^2\} \tag{14}$$

$$\boldsymbol{Err}(T_{D,\theta}) = \mathbb{E}_{X,Y}\{1(Y \neq T_{D,\theta}(X))\} = P(Y \neq T_{D,\theta}(X)) \tag{15}$$

where the last expression in the second equation is the probability of misclassification of the tree.

Given the probability distribution of P(X,Y), there exists a model ϕ_β that minimizes the expected prediction error and can be derived analytically independent or learning set D (Louppe, 2014). Conditioning on X gives;

$$\mathbb{E}_{X,Y}\{L(Y,\phi_\beta(X))\} = \mathbb{E}_X\{\mathbb{E}_{Y|X}\{L(Y,\phi_\beta(X))\}\} \tag{16}$$

Minimizing the term with conditional expectation with respect to Y;

$$\phi_\beta = \underset{c\in Y}{argmin}\mathbb{E}_{Y|X=x}\{L(Y,c)\} \tag{17}$$

ϕ_β is defined as Bayes model and $\boldsymbol{Err}(\phi_\beta)$ is the residual error, the minimum obtainable error with any model, which is considered as the irreducible error due to random deviations in the data(Louppe, 2014). We will exploit the irreducible concept when examining the dynamics of random forests and decision trees. In that sense, with couple of manipulations, Bayes Model for squared loss is

$$\begin{aligned}\phi_\beta &= \underset{c\in Y}{argmin}\ \mathbb{E}_{Y|X=x}\{(Y-c)^2\}\\ &= \mathbb{E}_{Y|X=x}\{Y\}\end{aligned} \tag{18}$$

For squared loss function, bayes model predicts the average value of Y at X=x. In zero-one loss function case Bayes Model is

$$\begin{aligned}\phi_\beta &= \underset{c\in Y}{argmin}\ \mathbb{E}_{Y|X=x}\{L(Y,c)\}\\ &= \underset{c\in Y}{argmin}\ P(Y\neq c\,|X=x)\\ &= \underset{c\in Y}{argmax}\ P(Y=c\,|X=x)\end{aligned} \tag{19}$$

The most likely class in the set Y is chosen by Bayes Model when using the zero-one loss function. Aformentioned residual error can be computed for both functions;

$$\boldsymbol{Err}(\phi_\beta) = \mathbb{E}_{Y|X=x}\{(Y-\phi_\beta(x))^2\} \tag{20}$$

$$\boldsymbol{Err}(\phi_\beta) = P(Y\neq\phi_\beta(x)) \tag{21}$$

With using the squared loss function, $\boldsymbol{Err}(T_{D,\theta}(x))$ can be written as

$$\begin{aligned}
\boldsymbol{Err}(T_{D,\theta}(x)) &= \mathbb{E}_{Y|X=x}\{(Y - T_{D,\theta}(x))^2\} \\
&= \mathbb{E}_{Y|X=x}\{(Y - \phi_\beta(x) + \phi_\beta(x) - T_{D,\theta}(x))^2\} \\
&= \mathbb{E}_{Y|X=x}\{(Y - \phi_\beta(x))^2\} + \mathbb{E}_{Y|X=x}\{(\phi_\beta(x) - T_{D,\theta}(x))^2\} \\
&+ \underbrace{\mathbb{E}_{Y|X=x}\{2(Y - \phi_\beta(x))(\phi_\beta(x) - T_{D,\theta}(x))\}}_{= 0 \text{ since } \mathbb{E}_{Y|X=x}(Y - \phi_\beta(x)) = 0 \text{ from (18)}} \\
&= \underbrace{\mathbb{E}_{Y|X=x}\{(Y - \phi_\beta(x))^2\}}_{\text{from (20) equals to } \boldsymbol{Err}(\phi_\beta(x))} + \mathbb{E}_{Y|X=x}\{(\phi_\beta(x) - T_{D,\theta}(x))^2\} \\
&= \boldsymbol{Err}(\phi_\beta(x)) + (\phi_\beta(x) - T_{D,\theta}(x))^2 \qquad (22)
\end{aligned}$$

As mentioned above, first term in the last equation corresponds the irreducible error and the second term is due to the prediction differences between Bayes Model and our decision tree estimation. The expected prediction error increases with an increase in that difference. Since the result does not depend on the Y-values, it can be also expressed without conditional expectation. Decision trees in random forest classifier uses a bootstrapped dataset and D is a random variable, thus, if we further examine the second term with taking expectation over D, it decomposes as

$$\begin{aligned}
&\mathbb{E}_D\{(\phi_\beta(x) - T_{D,\theta}(x))^2\} \\
&= \mathbb{E}_D\{(\phi_\beta(x) - \mathbb{E}_D\{T_{D,\theta}(x)\} + \mathbb{E}_D\{T_{D,\theta}(x)\} - T_{D,\theta}(x))^2\} \\
&= \mathbb{E}_D\{(\phi_\beta(x) - \mathbb{E}_D\{T_{D,\theta}(x))^2\} + \mathbb{E}_D\{(\mathbb{E}_D\{T_{D,\theta}(x)\} - T_{D,\theta}(x))^2\} \\
&\quad + \mathbb{E}_D\{2(\phi_\beta(x) - \mathbb{E}_D\{T_{D,\theta}(x)\})(\mathbb{E}_D\{T_{D,\theta}(x)\} - T_{D,\theta}(x))\} \\
&\text{since } \mathbb{E}_D\{\mathbb{E}_D\{T_{D,\theta}(x)\} - T_{D,\theta}(x)\} = \mathbb{E}_D\{T_{D,\theta}(x)\} - \mathbb{E}_D\{T_{D,\theta}(x)\} = 0 \\
&= \mathbb{E}_D\{(\phi_\beta(x) - \mathbb{E}_D\{T_{D,\theta}(x))^2\} + \mathbb{E}_D\{(\mathbb{E}_D\{T_{D,\theta}(x)\} - T_{D,\theta}(x))^2\} \\
&= (\phi_\beta(x) - \mathbb{E}_D\{T_{D,\theta}(x))^2 + \mathbb{E}_D\{(\mathbb{E}_D\{T_{D,\theta}(x)\} - T_{D,\theta}(x))^2\} \qquad (23)
\end{aligned}$$

In the equation (23), the first term shows how the expected prediction of our decision tree differs from Bayes Model also called squared bias and the latter term is the variance of our estimator. Therefore, we can define $\boldsymbol{Err}(T_{D,\theta})$ as follows

$$\boldsymbol{Err}(T_{D,\theta}) = noise(x) + bias^2(x) + var(x) \qquad (24)$$

where

$$\begin{aligned}
noise(x) &= \boldsymbol{Err}(\phi_\beta) \\
bias^2(x) &= (\phi_\beta(x) - \mathbb{E}_D\{T_{D,\theta}(x)\})^2 \\
var(x) &= \mathbb{E}_D\{(\mathbb{E}_D(T_{D,\theta}(x)) - T_{D,\theta}(x))^2\}
\end{aligned}$$

The same decomposition can be conducted for the zero-one loss function and as mentioned in (Louppe,2014), (Domingos,2000), (James,2003) ,(Friedman,1997) both zero-one and squared loss functions can be decomposed similarly. However, since the distribution of D is unknown, bias-variance decompostion cannot be solved explicitly as done for squared loss(Louppe, 2014). (Kohavi,1996) introduces another decomposition for zero-one loss function, but, still it remains to be unexplanatory compared to squared loss, thus, we explain the dynamics with using squared loss although the main focus of the paper remains to be on classification setting.

In regression setting, random forest classifier shares the same idea with classification prediction with soft voting. Random forest classifier for regression can be written as

$$\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x) = \frac{1}{B}\sum_{b=1}^{B}(T_{D,\theta_b}(x)) \tag{25}$$

When we take the average prediction in this case equals to expectation in terms of training set, we get

$$\begin{aligned}\mathbb{E}_{D,\theta_1,\theta_2,\ldots,\theta_B}\{\boldsymbol{RF}_{D,\theta_1,\theta_2,\ldots,\theta_B}(x)\} &= \mathbb{E}_{D,\theta_1,\theta_2,\ldots,\theta_B}\{\frac{1}{B}\sum_{b=1}^{B}(T_{D,\theta_b}(x))\} \\ &= \frac{1}{M}\sum_{b=1}^{B}\mathbb{E}_{D,\theta_b}\{T_{D,\theta b}(x)\} \\ &= \mu_{D,\theta}(x)\end{aligned} \tag{26}$$

where $\mu_{D,\theta}(x)$ is the average prediction of all ensembled trees as a random forest since θ's are random, independent and have the same distribution(Louppe, 2014). When we extend this finding bias of a random forest we can state that

$$bias^2(x) = (\phi_\beta(x) - \mu_{D,\theta}(x))^2 \tag{27}$$

meaning that squared bias cannot be decreased and will be same for any randomized models. So far regarding $noise(x)$ and $bias^2$, random forest does not propose any structure to decrease the prediction error, as the last remaining part of the prediction error, we can continue our exploration with variance of random forest. For any two trees $T_{D,\theta'}$ and $T_{D,\theta''}$ trained with the same training data and different growing parameters θ' and θ'', we can define the correlation coefficient as follows

$$\begin{aligned}\rho(x) &= \frac{\mathbb{E}_{D,\theta',\theta''}\{(T_{D,\theta'}(x) - \mu_{D,\theta'}(x))(T_{D,\theta''}(x) - \mu_{D,\theta''}(x))\}}{\sigma_{D,\theta'}(x)\sigma_{D,\theta'}(x)} \\ &= \frac{\mathbb{E}_{D,\theta',\theta''}\{T_{D,\theta'}(x)T_{D,\theta''}(x) - T_{D,\theta'}(x)\mu_{D,\theta''}(x) - T_{D,\theta''}(x)\mu_{D,\theta'}(x) + \mu_{D,\theta'}(x)\mu_{D,\theta''}(x))\}}{\sigma^2_{D,\theta}(x)} \\ &= \frac{\mathbb{E}_{D,\theta',\theta''}\{T_{D,\theta'}(x)T_{D,\theta''}(x)\} - \mu^2_{D,\theta}(x)}{\sigma^2_{D,\theta}(x)}\end{aligned} \tag{28}$$

3.3 INTERPRETATION

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.3.1 *Variable importance*

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4

APPLICATION AND COMPARISON

This chapter covers the application of random forest regression and the evaluation of its performance.

In subsection 4.1.1, we apply the random forest on simulated data, and show how its performance develops over increasing sample sizes.

Then in subsection 4.1.2, we apply the random forest on the real Titanic data set [13] and evaluate its performance.

In section 4.2 and section 4.2, we apply AdaBoost and Gradient Boosting respectively on the Titanic data set and compare their performance with that of the random forest.

4.1 APPLICATION OF RANDOM FOREST

4.1.1 *Simulated data*

In the simulation, we use a linear and a non-linear data generating process (DGP) for random forest regression. The linear DGP generates the data tuples (y, x_1, x_2, x_3) as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon, \tag{29}$$

whereas $(\beta_0, \beta_1, \beta_2, \beta_3) = (0.3, 5, 10, 15)$, $x_1, x_2, x_3 \sim \mathcal{N}(0, 3)$, and $\epsilon \sim \mathcal{N}(0, 1)$.

The performance of the Random Forest over an increasing sample is illustrated below in Figure 3 and Figure 3. For each sample drawn from the linear DGP, a set of parameters were optimized via cross validation. Then, the residual sum of squares (RSS) gets calculated based on the holdout set of 100 instances.

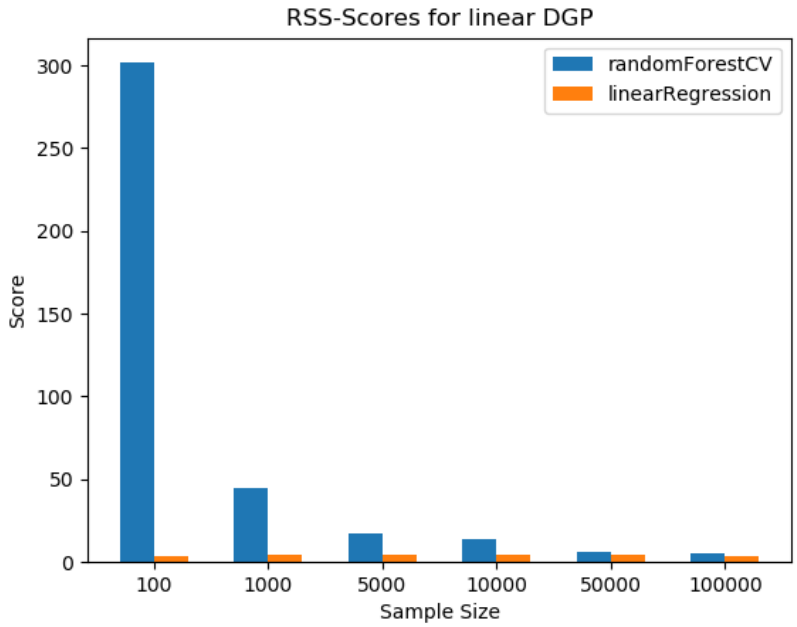

Figure 2: *This plot illustrates the RSS for different training sample sizes for Random Forest and OLS. These samples where drawn from a linear DGP in accordance to Equation 29. The holdout set for calculating the RSS were drawn again for each training sample from the same DGP. It always contained 100 observations. In case of the Random Forest, for each sample the parameters got optimized again via cross validation.*

As one can see in Figure 2 above, the RSS of the Random Forest converges for the linear DGP to that of the OLS for increasing sample sizes.

The non-linear DGP generates the data tuples (y, x_1, x_2) as follows:

$$y = \beta_0 + \beta_1 I(x_1 >= 0, x_2 >= 0) + \beta_2 I(x_1 >= 0, x_2 < 0) + \beta_3 I(x_1 < 0) + \epsilon, \tag{30}$$

whereas $(\beta_0, \beta_1, \beta_2, \beta_3)$, x_1, x_2 and ϵ are the same in the previous DGP.

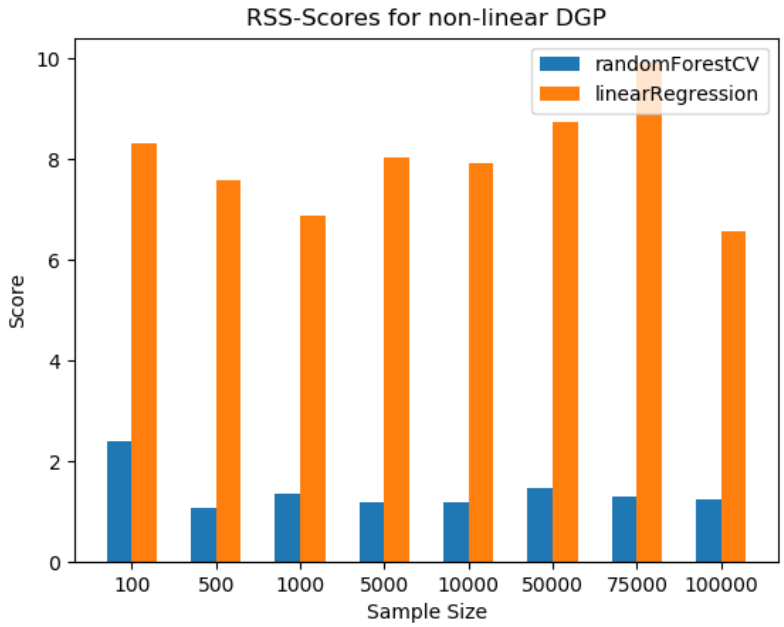

Figure 3: *This plot illustrates the RSS for different training sample sizes for Random Forest and OLS. These samples where drawn from a non-linear DGP in accordance to Equation 30. The holdout set for calculating the RSS were drawn again for each training sample from the same DGP. It always contained 100 observations. In case of the Random Forest, for each sample the parameters got optimized again via cross validation.*

As one can see above in Figure 3, the Random Forest performs strictly better than the OLS for any sample size. Due to this DGP resembling a stratification similar to that of a Descision Tree, the RSS of the Random Forest converges relatively quickly while that of the OLS remains unstable and high.

4.1.2 *Real data example*

As previously mentioned, we applied the Random Forest on the Titanic data set [13] in order to determine the survival of the passengers based on reported attributes like name title or booked cabin. In order to use the data to its fullest extent, we conducted additional feature engineering. Without that, many features remain unusable for our methods, because they contain missing values or values that are formatted as text. For the implementation of the feature engineering and the classification, one can consult our code repository [12]. The Random Forest managed to achieve a total classification accuracy of 84,32% on the holdout set. The holdout set consists of 15% of the total data. According to the confusion matrix in Figure 4, for passengers that died, the accuracy was slightly higher compared to those that survived. This is to be expected, since deaths outnumber survivals considerably.

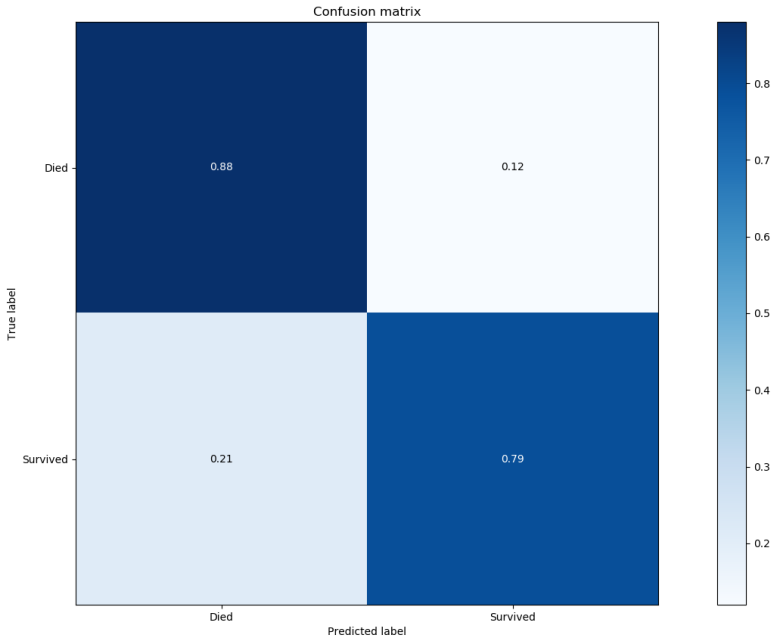

Figure 4: *This plot illustrates the accuracy of the Random Forest's prediction on the Titanic data set.*

4.2 GRADIENT BOOSTING CLASSIFIER

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4.3 ADABOOST CLASSIFIER

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5

CONCLUSION AND OUTLOOK

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

BIBLIOGRAPHY

[1] GÃŠrard Biau. "Analysis of a random forests model". In: *Journal of Machine Learning Research* 13.Apr (2012), pp. 1063–1095.

[2] L. Breiman et al. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. ISBN: 9780412048418. URL: https://books.google.de/books?id=JwQx-WOmSyQC.

[3] Leo Breiman. "Consistency for a simple model of random forests". In: (2004).

[4] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[5] Misha Denil, David Matheson, and Nando De Freitas. "Narrowing the gap: Random forests in theory and in practice". In: *International conference on machine learning*. 2014, pp. 665–673.

[6] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[7] Hemant Ishwaran et al. "Variable importance in binary regression trees and forests". In: *Electronic Journal of Statistics* 1 (2007), pp. 519–537.

[8] Tae-Hwy Lee, Aman Ullah, Ran Wang, et al. *Bootstrap Aggregating and Random Forest*. Tech. rep. 2019.

[9] Gilles Louppe. "Understanding random forests". In: *University of Liège* (2014).

[10] Gilles Louppe et al. "Understanding variable importances in forests of randomized trees". In: *Advances in neural information processing systems*. 2013, pp. 431–439.

[11] Oded Maimon and Lior Rokach. "Data mining and knowledge discovery handbook". In: (2005).

[12] *Research Module Application*. URL: https://github.com/RaRedmer/Research_Module_Application.

[13] *Titanic: Machine Learning from Disaster*. URL: https://www.kaggle.com/c/titanic/data.

[14] Hal R Varian. "Big data: New tricks for econometrics". In: *Journal of Economic Perspectives* 28.2 (2014), pp. 3–28.

DECLARATION

I hereby certify that this material is my own work, that I used only those sources and resources referred to in the thesis, and that I have identified citations as such.

Bonn, January 5, 2020