Stockholms universitet

Predictive Power of Logistic Regression versus Random Forest: A simulation study

Amanda Möller

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2019:7
Matematisk statistik
Juni 2019

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Bachelor Thesis **2019:7**
http://www.math.su.se

Predictive Power of Logistic Regression versus Random Forest: A simulation study

Amanda Möller*

June 2019

Abstract

In statistical and machine learning, there are many powerful techniques that can be used to make predictions using big data. A central problem in statistical learning involves choosing the best method for a given application. In this thesis two classifiers are examined, logistic regression and random forest, where these two are investigated and compared to each other in order to find the best classifier. The analysis is done both practically and theoretically and is focused on binary classification where the response variable is categorical and divided in classes. The simulation studies are performed for three different models where the number of predictors and the size of the data set vary and in each study are 25 data sets generated. AUC, F-score and misclassification rate are the measures used to analyze and evaluate the predictive power for the classifiers. It is concluded that the predictive power of logistic regression is better than random forest when the two classes are linearly separable and random forest predicts better than logistic regression when the two classes are non-linearly separable. Both classifiers perform better when the correlation between the explanatory variables increases.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: amanda.moller95@gmail.com. Supervisor: Ola Hössjer.

Acknowledgement

I would like to express my sincere thanks to my supervisor Ola Hössjer for the valuable guidance throughout the writing of this bachelor thesis. I would also like to thank my supervisor group for valuable advice to my work. I am very grateful to my family and friends for the excellent support.

Contents

1 Introduction

In *statistical and machine learning*, there are many powerful techniques that can be used to understand and to interpret the results from big data [9]. A central problem in statistical learning involves choosing the best method for a given application. Logistic regression and random forest are two approaches for *supervised learning*, where the statistical models are adapted to predict or estimate an outcome based on one or more input variables. [11]

Logistic regression was introduced in the 1940's by various authors and is one of the first explored statistical models when the response variable is binary. Logistic regression is a *parametric method*, which implies that an assumption about the function has to be made. Hence, the parameters in the model are estimated by the maximum likelihood method [11].

Random forest, on the other hand, is a *non-parametric method* where no explicit assumption about the function has to be made [11]. Random forest was generally introduced in 1995 by Ho, but the properties of random forest were first explored by Breiman in 2001. The idea of the method is that it should improve a decision tree's ability to predict the response variable [3].

The aim of this thesis is to evaluate the predictive power of these two different statistical methods, logistic regression and random forest, in order to find the best classifier. By using simulated data sets, we will examine the performance of each classifier when different types of data sets are used and then compare the results to each other.

First, in Section 2 the theoretical framework is provided, where the structure of logistic regression and random forest is explained. In order to understand the results, different methods used to evaluate the predictive power are also explained in this section. The creation of the simulated data and the different simulation studies are explained in Section 3. The results are then presented in Section 4. Finally, a discussion and the conclusions will be presented in Section 5 and Section 6.

2 Theory

This section presents the theory that will be used in this thesis. The explanation of the different models, classification algorithms and measures of model accuracy are made in order to understand the background of the simulation study.

2.1 Introduce to Machine Learning

The area of machine learning makes use of data where the algorithms iteratively learn from data. This means that the models of the algorithms are created using sample data and not by explicit programming [9]. The data set $\mathcal{D} = \{(x_i, y_i), i = 1, 2, \ldots, n\}$ used to create and analyze the algorithms contains n observations where each observation consists of predictors and a response. The predictors in the ith observation are given by $x_i = (x_{i1}, x_{i2}, \ldots x_{ip})'$ and the corresponding response variable is given by y_i [14].

The *training set* denoted by $\mathcal{Z} = \{(x_i, y_i), i = 1, 2, \ldots, n_{train}\}$ in our investigation is generated by randomly selecting n_{train} observations from the data set $\mathcal{D}$. These observations are used to create the function f such that $f : X \to Y$, in other words a function that maps the set of predictors $x = (x_1, x_2, \ldots, x_p)'$ into some label y. In statistics, this function is called a *classifier* and the relationship between the inputs and the outcome can be denoted as $f(x) = y$. The function represents information that the predictors provide about the response variable and explain the variations in the response variable related to changes in the predictors. The algorithms of machine learning aim to find the best classifier [14].

The remaining observation in $\mathcal{D}$ are used as a *test set* or *validation set* denoted by $\mathcal{V} = \{(x_i, y_i), i = n_{train} + 1, n_{train} + 2, \ldots, n_{train} + n_{test}\}$ where $n = n_{train} + n_{test}$. These observations are used to predict the outcomes by applying the inputs to the estimated function for f. Thus, the predicted values of the response variable are given by,

$$\hat{f}(x) = \hat{y}.$$

Since the information of the true value of y can be obtained in the validation data, a comparison between the predicted outcome $\hat{y}$ and the true outcome can be made in order to investigate the predictive power of the classifiers [14].

2.1.1 Classification Problem

In machine learning classification and mainly *binary classification* is one of the most usually studied problems when the response variable is qualitative or categorical. In other words, the response variable belongs to a particular category or class [21]. The process of predicting the class of a given observation is called

classification and *classifying* an observation means that one observation is predicting to have a certain qualitative response. Binary classification means that there are only two possible classes for the response variable. Usually dummy variables are used to denote these outcomes were 0 corresponds to one class and 1 corresponds to the other class [11].

2.2 Multiple Logistic Regression

Logistic regression is a regression model suitable for classification problems. The model explains the relationship between a response variable, Y, and one or more explanatory variables $X = (X_1, X_2, \ldots, X_p)'$. This method is the most commonly used to predict the response variable when Y is categorical and the values of the explanatory variables is given by the vector $x = (x_1, x_2, \ldots, x_p)'$, where we have p predictors [8].

Assume that Y is a binary response variable. The logistic regression model estimates the conditional probability that $Y = 1$ given the explanatory variables and it is denoted by $p(x)$. The probability that Y belongs to class 0 is then given by $1 - p(x)$. To estimate the probability, a function that generates an output between 0 and 1 is used. This function is called logistic function and it is given by,

$$p(x) = Pr(Y = 1|X = x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}, \tag{1}$$

where β_0 is the intercept and $\beta = (\beta_1, \ldots, \beta_p)'$ contains the effect parameters of all predictors. The parameter β_j explains the relationship between the probability that Y belongs to class 1 and the explanatory variable X_j. If β_j is greater than zero, then increase in x_j is associated with increasing $p(x)$. The opposite is true if β_j is negative [11].

By rearranging Equation (1), we receive an alternative representation of the relation between the Y and the predictors called the *log-odds* or *logit* transformation of $p(x)$,

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p. \tag{2}$$

On the log odds scale there is a linear relation between the response variable and the predictors [1].

2.2.1 Predictions and Estimating the Coefficients

The intercept β_0 and the effect parameters $\beta = (\beta_1, \ldots, \beta_p)'$ are unknown parameters and *maximum likelihood method* is used to estimate this parameters

using the training data set, where $i = 1, 2, \ldots, n_{train}$ corresponds to the observations. Since the binary response variable is a Bernoulli random variable, the maximum likelihood function is given by [19],

$$\begin{aligned} L(\beta_0, \beta) &= \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)) \\ &= \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i} \end{aligned} \tag{3}$$

where y_i attains the value 1 or 0. The log-likelihood function is obtained by taking the logarithm of the Equation (3),

$$\begin{aligned} \log L(\beta_0, \beta) &= \sum_{i=1}^{n} y_i \log p(x_i) + \sum_{i=1}^{n} (1 - y_i) \log(1 - p(x_i)) \\ &= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i \log\left(\frac{p(x_i)}{1 - p(x_i)}\right) \\ &= \sum_{i=1}^{n} -\log(1 + e^{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}}) \\ &+ \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}). \end{aligned} \tag{4}$$

To find the maximum likelihood estimate of the intercept $\hat{\beta}_0$ and the effect parameters, $\hat{\beta}$ we differentiate the log likelihood given in Equation (4) with respect to the parameters, set the derivatives equal to zero and solve the resulting system of equations [19]. The estimated parameters and Equation (1) are then used to make predictions about the response variable, Y. By applying the observation from the test set to the logistic function consisting of the estimated parameters, we receive the estimated conditional probability that Y belongs to class 1 given the predictors [11].

2.3 Random Forest

Random forest is an ensemble learning algorithm where many *decisions trees* are used to predicate an outcome [23]. Since the algorithm can both be used for classification and regression problems, decision trees are also known as *CART* which is an acronym for *classification and regression trees*. In general, CART is based on yes/no or true/false questions which means that each internal node has exactly two outgoing branches and then classifies based on the answers [20].

Bagging is a resampling version of decision trees and Random forest is an improvement of bagging. To apply and understand the algorithm of random forest, knowledge of both decision trees and bagging must be provided, which will be presented in this section. We will delimit ourselves to explaining the theory for classification trees because later in our simulations we have a qualitative response variable. The theory in the following sections is from *An Introduction to Statistical Learning* James, G. et al. (2017) [11] unless otherwise stated.

2.3.1 Classification Trees

Classification trees are used to predict a qualitative response variable and it is a non-parametric statistical method. Let Y be a binary response variable with outcomes 0 or 1 and $X = (X_1, X_2, \dots X_p)'$ be the p predictors. The predictor space, $\mathbb{R}^p$, in the classification tree methodology is segmented into a number of distinct and non-overlapping regions [10]. That is, each region is viewed as homogeneous for the purpose of predicting Y, denoted by $R_1, R_2, \dots, R_T$.

The trees are created with the training set $\mathcal{Z}$ through a *top-down* approach, which means that it starts at the top of the tree, followed by a series of splitting rules and ends with the *leaves* at the bottom of the tree. The starting point, called *root node*, consists the entire training set and it carries out the first split of the predictor space. This kind of split is called a binary split, which implies that the condition is either satisfied or not satisfied. Along the tree, more splits of the predictor space can be made and these are referred to as *internal nodes*. The results of each split depend on the previous splits, since there are just a subset of observations in each internal node. The connection between two nodes is called *branches*. The left-hand branch corresponds to all observations that satisfy the conditions and the right-hand branch corresponds to all observations who do not. What we have mentioned previously as leaves or regions is also known as *terminal nodes*. They constitute the end of the tree where the nodes do not split anymore. All observations in the same region are classified to the same outcome according to the *most commonly occurring class* based on the training model.

To better understand the process of a classification tree, a simple example is illustrated in Figure 1 and Figure 2, where we have a two-dimensional classification tree involving the predictors, X_1 and X_2. The outcome of the response variable Y can either be 0 or 1. In Figure 2, we can see the proportion of observations between the different regions, when fitting the model with the training set. We can also see that the observations in the regions are classified to the most commonly occurring class. By applying the classification tree to new observations, we can predict the outcome of these. In the root node, where the first split is made, the observations with a value of X_2 less than or equal to θ_1 fall down in the left-hand branch and a new decision is made. Of these, the observations with a value of $X_1 \leq \theta_2$ fall down to the region R_1 and are predicted as $Y = 0$. The observations that do not satisfy this condition fall down to the

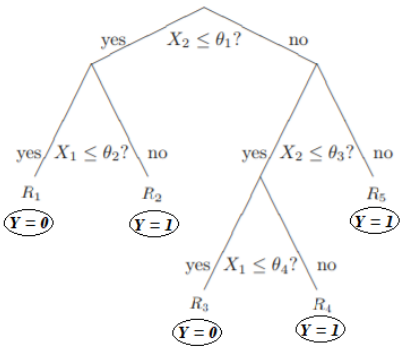

Figure 1: *A two-dimensional classification tree with four splits and five terminal nodes*

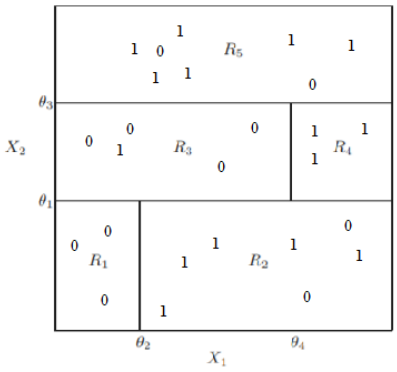

Figure 2: *A graph of the partition of the two-dimensional predictor space*

region R_2 and the response variable is predicted as $Y = 1$. The observations in the root node with a value of X_2 bigger than θ_1 instead fall to the right-hand branch. As we can see in Figure 1, more decisions are made in each split before the observations reach the regions and are classified. [10],[11].

2.3.2 Splitting Strategies

In order to grow a classification tree, there are methods used for the trees to predict as well as possible. Classification tree is a *greedy* approach which implies that we want to choose the "best" split in each node, rather than choosing a split that leads to a better tree in any future step. In order to choose such splits, *recursive binary splitting* is used, such that each possible predictor variable, X_j $j = 1, \ldots, p$ are tested against different cutpoints θ, in order to minimize a *cost function*. The procedure of testing the predictor variables works differently depending on whether the variable is numerical or categorical. Let n be the number of unique values for a numerical variable, then there are $n - 1$ possible cutpoints to be tried in order to find the best split. Instead, if the variable is an K-categorical variable with categories $l_1, \ldots, l_K$, there are $2^{K-1} - 1$ possible cutpoints to be tried [10]. The selected splitting criterion that minimizes the cost function and splits the predictor space in two is defined by,

$$R_L(j, \theta) = \{X | X_j \leq \theta\} \text{ and } R_R(j, \theta) = \{X | X_j > \theta\}, \tag{5}$$

where R_L corresponding to the left-hand branch and R_R to the right-hand branch. Splitting continues until the regions are too small or some stopping criterion is activated. Some problems with classification trees is that the tree is often too complex and overfits the data. An approach used to reduce the

variance and increase the interpretation is called *pruning trees*. This method is not used in random forest and will therefore not be explained in such detail in this thesis. In random forest the trees are full-grown [7] which implies that the minimum size of terminal nodes is set to one [13].

For a categorical response variable with K classes, the selected cost function that is later used in our simulations is called *Gini index*, defined by

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}). \tag{6}$$

Here $\hat{p}_{mk}$ denotes the proportion of observation in the mth internal node belonging to class k and it is given by

$$\hat{p}_{mk} = \frac{1}{n_m} \sum_{x \in R_m} I(y_i = k), \tag{7}$$

where n_m indicates the number of observations in node m and $I(y_i = k)$ is an indicator returning 1 if the response variable of observation i belongs to class k, otherwise it will return 0. When the response variable is categorical with value 0 or 1, Equation (6) can be expressed as

$$G = 2\hat{p}_m(1 - \hat{p}_m), \tag{8}$$

where $\hat{p}_m$ denotes the proportion of observations in the mth node belonging to class 1. Other common measures of the cost function are *classification error rate* and *entropy*. The prediction model for an input x in classification trees can be formulated as [11]:

$$f(x) = \sum_{t=1}^{T} c_t I(x \in R_t), \tag{9}$$

where $c_t = \arg\max_k \{\hat{p}_{tk}\}$ denotes the predicted class label k based on the training set for the tth region and $I(x \in R_t)$ returns 1 if the input belongs to R_t [23].

2.3.3 Bagging

Bagging, an acronym for *bootstrap aggregating*, is an approach that mainly improve the classification trees' ability of predicting the outcome. Some trees may have high variance, that is, the trees can differ quite a lot if we randomly split the training data into two parts and fit a tree to each of them. The idea

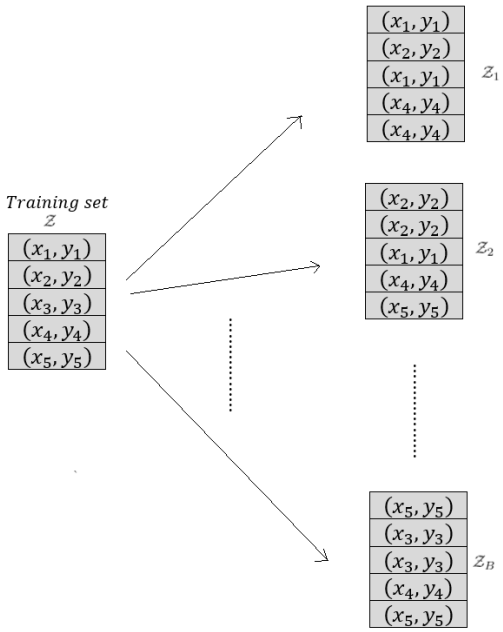

Figure 3: *Illustrated example of B bootstrapped data set generated from the training set, where the training set contains five observations*

of bagging is based on generating B different *bootstrapped* training data sets denoted by $\{\mathcal{Z}_b; b = 1, \ldots, B\}$, where the observations are drawn randomly *but with replacement* from the training set. Hence, some observations may appear several times or alternatively not at all in any particular $\mathcal{Z}_b$. Each bootstrapped set has observations with the same variance, σ^2 and are of the same size as the training set. Assume that the training set contains n_{train} observations, then the variance of the mean $\bar{\mathcal{Z}}$ is given by σ^2/n_{train}, if the observations are independent. A simple example of bootstrapped data is illustrated in Figure 3, where the training set only contain five observations.

Bagging is applied to the trees when we want to reduce the variance of the trees but at the same time increase the prediction accuracy. The trees are created by using the B bootstrapped data sets and fit a classification tree to each of them. [2] The idea of bagging is to combine all the predictions of an observation from all B trees into a single classifier, and thereby reduce the variance of the trees. To classify a test observation, the observation is applied to each one of the B trees and the classifier is recorded for each one of them. The overall prediction is decided by a majority vote, which implies that each observation is classified to the most commonly occurring class among the B predictors [11]. This can be expressed by using Equation (9) and calculating the predicted outcome of an input vector x for each bootstrapped set $\hat{f}^1(x), \hat{f}^2(x) \ldots \hat{f}^B(x)$ and the single

classifier is determined by [7]:

$$\hat{f}(x) = \arg\max_k \sum_{b=1}^{B} I(\hat{f}^b(x) = k). \tag{10}$$

The bagging technique has some weaknesses since the different trees may be highly correlated to each other. This is because strong predictors in the data set entails that the first splits will be similar in many of the trees.

2.3.4 Random Forest

Random forest is a modification of bagging, which aims to reduce the variance and minimize the correlation between the trees. The approach is similar to bagging where we fit a tree for each bootstrapped training set and combine the predicted outcome from each of them to a single predictor. However, the difference between the two approaches is that random forest only uses a random subset of m predictors from the full set of p predictors as split candidates in each node. Typically, m is determined by $m \approx \sqrt{p}$. Notice, when $m = p$, that all the predictors can be chosen as split candidates, then pure bagging is obtained again.

The random forest algorithm can be summarized as follows. In the first step, B bootstrapped training sets are created as mentioned in Section 2.3.3. For each $\mathcal{Z}_b, b = 1, \ldots, B$, a tree is created by using recursive binary splitting described in Section 2.3.2, but for each split, only a random subset of selected predictors can be used as split candidates in order to reduce the correlation between the trees. For a given test observation x, we use Equation (10) to combine $\hat{f}^1(x), \ldots \hat{f}^B(x)$. The single predicted outcome for the observation is determined by a majority vote. That is, the most commonly occurring outcome among all the trees.

2.4 Logistic Regression vs Random Forest

A big difference between logistic regression and random forest is that logistics regression is a parametric method while random forest is a non-parametric method [11]. For a binary response variable that attains the value 1 or 0, this implies that logistic regression estimates a fixed set of parameters and, as we will see in Section 2.5, predictor space is divided by a linear hyperplane defined as:

$$g(x) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = 0, \tag{11}$$

where observations on one side of the hyperplane will be classified as 1 and the observations on the other side will be classified as 0 [14].

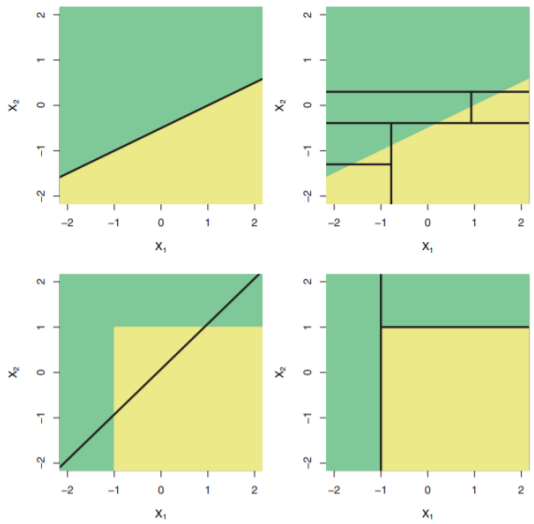

Figure 4: Top row: *The true decision boundary of the classes is linear where logistic regression (left) outperform a classification tree (right) that splits the predictor space parallel to the axes* Bottom row: *The true decision boundary of the classes is non-linear where a classification tree (right) outperform logistic regression (left) The figure is from [11] pp.315*

Random forest, on the other hand, is a non-parametric method that is not characterized by a finite set of parameters. That is, all observations in the training set are used to fit the model [17] in such a way that the observations in the test set should be predicted as accurately as possible without the predictive function being too wiggly or rough. [14].

If the estimated parameters in logistic regression are close to the true form of the function f, logistic regression will outperform the non-parametric approach. However, random forest will outperform logistic regression if the relationship between the response variable and the predictors is highly non-linear and complex [11].

An illustrated example for one classification tree and logistic regression for the two-dimensional predictor space is presented in Figure 4. The green area indicates one class and the yellow area indicates the other class. The two plots at the top shows the case where the predictor space is separated by a linear hyperplane. Logistic regression (top left) predicts the classes perfectly, for a very large data set, and random forest (top left) has some struggle and missclassifies some observations. The two plots in the bottom shows the case where the classes are not separated linearly. In this case, random forest predicts the classes perfectly and logistic regression has some struggle and missclassifies some observations.

2.5 Model Accuracy

There are various measures for evaluating the ability of random forest and logistic regression to predict the outcome. The measurements that will be used in this study are described in the following sections. For logistic regression, the classification of an observation is determined by a given threshold value, c. By default the threshold is set to 0.5, which implies that Y belongs to class 1 if $P(Y = 1|X) > 0.5$ or class 0 if $P(Y = 1|X) < 0.5$ [11]. For random forest, we recall that Equation (10) is used for classification, with the argmax ranging over two indices $k = 0, 1$.

2.5.1 Misclassification Rate

Misclassification rate is also called *test error* or *training error* and is the most common approach to investigate the accuracy of the statistical model when the response variable is qualitative. Misclassification rate indicates the proportion of incorrectly classified observations. Let $\hat{y}_i$ be the predicted class label for the ith observation. The misclassification rate is then given by,

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i) \tag{12}$$

where $I(y_i \neq \hat{y}_i)$ is an indicator variable returning 1 if the observation is predicted to the wrong class and 0 if the observation is classified correctly. The misclassification rate that occurs when we use the same observations to predict the class label as we used to estimate the statistical model, is called training error. Instead, if the observations from the test data are applied to the model in order to predict the outcome, the proportion of incorrectly classified observations is called test error. A low value of the test error indicates that the statistical model is a good classifier [11].

2.5.2 Confusion Matrix

Table 1: A confusion matrix

		True Condition	
		0	1
Predicted	0	TP	FP
Condition	1	FN	TN
	Total	P	N

We start by labeling our outcomes as positive and negative. Let class 0 be defined as a positive outcome and class 1 as a negative outcome. When we have

a binary classification problem, there are two types of errors that can occur. The first error, called *false positive (FP)*, implies that an observation belonging to class 1 is predicted incorrectly. The second error, called *false negative (FN),* implies that an observation belonging to class 0 is predicted incorrectly. This can be represented in a *confusion matrix* displayed in Table 1. The confusion matrix also shows the observations that have been correctly predicted. These observations are called *true postive (TP)* and *true negative (TN)* [11]. The true positive rate and the false positive rate is defined by,

$$\textit{True positive rate} = \frac{TP}{TP + FN} \tag{13}$$

$$\textit{False positive rate} = \frac{FP}{FP + TN}. \tag{14}$$

It is also possible to determine the misclassification error by using the confusing matrix,

$$\textit{Misclassification rate} = \frac{FP + FN}{FP + FN + TP + TN}. \tag{15}$$

2.5.3 F-Score

F-score, also known as *F_1-score* or *F-measure*, is another method for measuring performance to predict the response variable of different statistical models based on the confusion matrix. To compute the F-score, both *recall* and *precision* are included in the model. Recall is the same as the true positive rate defined in Equation (13) and precision is the fraction between the true positives and all the positively predicted outcomes given by,

$$\textit{Precision} = \frac{TP}{TP + FP}. \tag{16}$$

A high recall score indicates that the model does well to predict the response variable relative to all the true positive observations and a high precision score indicates that the model does well relative to all the predicted positive outcomes. Since both measures are often equally important, a single measure called F-score was developed for combining these two,

$$F\textit{-score} = 2 \cdot \frac{\textit{Precision} \cdot \textit{Recall}}{\textit{Precision} + \textit{Recall}}, \tag{17}$$

and it attains a value between 0 and 1. A value close to 1 indicates that the statistical model predicts well and a value close to 0 indicates that it does not

perform well [5]. F-score is derived from the harmonic mean between precision and recall [18],

$$Harmonic\ mean = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2}{\frac{P+R}{P \cdot R}} = 2 \cdot \frac{P \cdot R}{P + R}$$

2.5.4 ROC

Receiver operating characteristics (ROC) graph is also a commonly used method to evaluate the accuracy of a statistical method using the information from confusion matrices based on several thresholds. ROC is presented in a two-dimensional graph, where the true positive rate is plotted against the false positive rate for all possible thresholds. The coordinates (fpr,tpr) correspond to a single point in ROC space, obtained from the confusion matrix that each threshold produces. An example of a ROC curve is shown in Figure 5. The diagonal line shows where the true positive rate is equal to the false negative rate. Any points at that line means that the proportion of correctly classified positives is the same as the proportion of incorrectly classified positives. The diagonal is also referred as the the result when the classifier randomly guesses a class. The upper right point (1,1) represents the case when the threshold is so large that every observation is classified as a positive. The opposite scenario is presented in the lower left point (0,0), where the threshold is low enough so that all observations are classified as negatives. The ideal classifier corresponds to the upper left point (0,1), were the fitted model makes perfect classification or near that [6].

To use ROC as a single measure of the accuracy of a statistical model, the *area under the ROC curve* (AUC) is used. The value of AUC is obtained by calculating the area under the ROC curve and it is a number between 0 and 1. A value of AUC close to one indicates that the method classifies almost perfectly and a value close to or less than 0.5 implies that the classifier performs poorly [6].

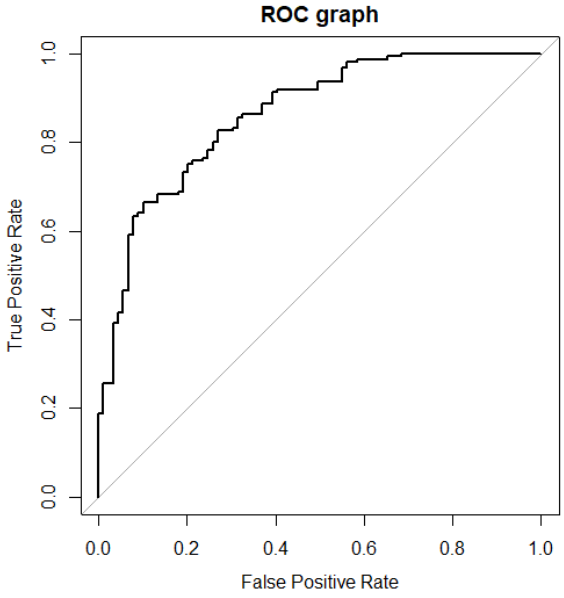

Figure 5: *An illustrated example of a ROC-curve where the true positive rate is plotted against the false positive rate for all possible thresholds.*

3 Simulation

The purpose of this simulation study is to investigate the predictive power of the two statistical models, logistic regression and random forest, for different data sets when the number of observation and predictors vary. This section explains how the simulated data sets are generated and the various simulation scenarios. The entire simulation study has been performed with the statistical software R and the development of the simulated data set is inspired by a previous thesis written by A.Nöu [15].

3.1 Simulation Data

To create our simulated data set, we start by generating the values of the predictors, $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})'$ for each observation $i = 1, \ldots, n$, by sampling from a multivariate normal distribution with the density function

$$f_{\mathbf{X}}(\mathbf{x}) = \left(\frac{1}{2\pi}\right)^{p/2} \frac{1}{\sqrt{\det \boldsymbol{\Sigma}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}, \tag{18}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_p)'$ is the mean vector and $\boldsymbol{\Sigma}$ denotes the covariance matrix. In our studies, we let the expected values, μ_i, and the variance, $Var(X_i)$

be equal to 0 respectively 1 for all X. The form of the covariance matrix is a *Toeplitz matrix* where each descending diagonal from left to right have the same value and is expressed as follows

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{p-1} \\ \rho & 1 & \rho & \cdots & \rho^{p-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{p-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{p-1} & \rho^{p-2} & \rho^{p-3} & \cdots & 1 \end{bmatrix} \tag{19}$$

where ρ attains values between 0 and 1. To generate the values of the response variable, y_i, to our observations, we use a sigmoid function given by,

$$\boldsymbol{\sigma}(x) = \frac{1}{1 + e^{-g(x)}} \tag{20}$$

where $g(x)$ is a given real-valued function that can attain both positive and negative values. The function is based on given values of the intercept β_0, the effect parameters $\beta = (\beta_1, \ldots, \beta_p)'$ and the predictors. Then, the values of the response variable is sampled from a Bernoulli distribution where $P(Y = 1|X = x)$ is equal to $\boldsymbol{\sigma}(x)$.

The simulated observations are then divided into a training set used to train the models and a test set used to investigate the predictive power of the model. The training set contains 75% of the observations and the test set 25%. The same training and test set are used when the two methods are trained and used for prediction.

3.2 Simulation Studies

As mentioned in the previous section we will use a real-valued function in our sigmoid function to generate the probability that y_i belongs to class 1 given x_i. The two different functions we will use in our simulation studies are the following:

$$g(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \tag{21}$$

$$g(x) = \beta_0 + \beta_1 x_1^2 + \cdots + \beta_p x_p^2 \tag{22}$$

where β_0 is used to balance the outcomes so we get approximately the same number of observations belonging to class 0 and class 1.

3.2.1 Linearly Separable Classes with Uncorrelated Predictors

Simulation Study 1

In Simulation study 1 are we going to investigate the predictive power of the methods when the two classes are linearly separable and the predictors are independent ($\rho = 0$). The covariance matrix is then given by the identity matrix. Furthermore, to get a balanced data set, we set the intercept β_0 equal to zero and the other parameters are set to

$$\beta_1 = \beta_2 = \cdots = \beta_p = 1.$$

To get the two classes linearly separable, we insert Function 21 into the sigmoid function when we generate the response variables. The probability that y belongs to class 1 is then given by,

$$\boldsymbol{\sigma}(x) = \frac{1}{1 + e^{-(x_1 + x_2 + \cdots + x_p)}}. \tag{23}$$

This function is also known as the logistic function as we explained in Section 2.2.

3.2.2 Linearly Separable Classes with Correlated Predictors

Simulation Study 2

Simulation study 2 is almost similar to the previous one but in this study the predictors are correlated. The intercept β_0, the effect parameters $\boldsymbol{\beta}$ and the sigmoid function are determined in the same way as in Section 3.2.1. Since the variances of the predictors are set to one, we get a correlation between two predictors X_i and X_j that equals $\rho^{|i-j|}$, as seen from the covariance matrix. Since $0 \leq \rho \leq 1$, predictors close to each other are highly correlated when ρ is close to 1 and predictors far from each other are less correlated. For instance, the correlation between X_1 and X_2 is given by ρ and the correlation between X_1 and X_3 is given by ρ^2. In this study we are going to investigate the predictive power of logistic regression and random forest when ρ is set to 0.2 and 0.8.

3.2.3 Non-linearly Separable Classes

Simulation Study 3

Unlike the previous studies, we want to investigate the predictive power when we have two non-linearly separable classes and uncorrelated predictors. Let $\boldsymbol{\beta}$ be chosen as follows,

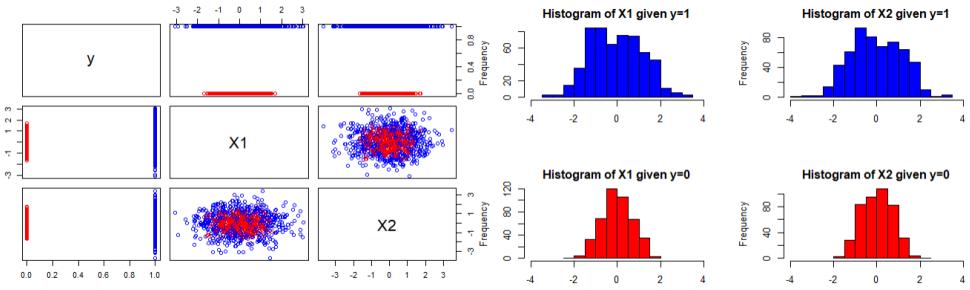

Figure 6: *(Left) Example of a simulated data set in the two-dimensional case where the two classes are non-linearly separable. (Right) The distribution of the observations for X1 and X2 when the response variable is either 0 or 1. In the case where y equals to 1 the observations for X1 are distributed between -3 and 3 and for X2 the observations are distributed between -4 and 4. When y equals 0 the observations are distributed between -2 and 2 for X1 and between -2 and 2 for X2.*

$$\beta_1 = \beta_2 = \cdots = \beta_p = 1.$$

We insert Function (22) into the sigmoid function when we generate the response variable. The probability that y belongs to class 1 is then given by,

$$\boldsymbol{\sigma}(x) = \frac{1}{1 + e^{-(-\beta_0 + x_1^2 + x_2^2 + \cdots + x_p^2)}}. \tag{24}$$

Since a sum of squared independent standard normal random variables have a χ^2-distribution with p degrees of freedom [16], we let β_0 equal the median of the $\chi^2(p)$-distribution in order to get a balanced data set. In this case, class 0 gets trapped by class 1. This is demonstrated in Figure 6 where the observations belonging to class 0 are in the inner circle and the observations belonging to class 1 are in the outer circle. The two classes are to a large extent separated by a circle.

3.3 Package in R

The models of logistic regression and random forest have been created in R Statistical Software with the packages described in this section.

3.3.1 Logistic Regression

The *glm* function in the *stats* [22] package has been used to fit the model of logistic regression where all the predictors are included in the model.

3.3.2 Random Forest

The package *randomForest* [13] for *R* is used to fit a model to the simulated data set and it implements Breiman's random forest algorithm, based on Breiman and Cutler's original Fortran code [4]. The number of predictors m sampled as candidates at each node are set to $m = \sqrt{p}$, where p is the total number of predictors in the data set. The cost function used to fit the model of random forest is the *Gini index*, as explained in Section 2.3.2. The number of trees created in each simulation are set by default to 500 trees and the size of the tree is determined by a stop criterion where the size of the terminal nodes has meaning. By default, the trees are full-grown and stop to split when the minimum size of the terminal node is equal to one. The sizes of the trees for each cases of the simulation studies are presented in the appendix (Figures 10-13).

4 Results

The results obtained from our simulations and a brief explanation of how the results were generated will be presented in this section. The simulation of the three different cases was repeated 25 times and both models were fitted with the same training set and used to predict the same test set. In each simulation the misclassification rate, F-score and AUC are calculated both for logistic regression and random forest. In Figures 7-9 the results are represented for each simulation when $n_{test} = 500$ and $p = 50$. In Figure 8 the results are represented for $\rho = 0.8$.

In order to get a single value for these measurements, we take the average of the outcomes in each simulation and examine the standard deviations. These results are given in Table 2 - 4 when the number of observations in the test set are 1000, 500 or 100 and the number of predictors are 100, 50 or 10. The differences between the mean AUC, ME and F-scores are also presented in these tables. For mean AUC and F-score, the differences are calculated by taking the mean values of logistic regression minus those for random forest. For mean ME, the differences are calculated by taking the mean values of random forests minus those for logistic regression. A positive difference implies that logistic regression predicts better than random forest and a negative value indicates that random forest predicts better than logistic regression.

4.1 Results from Linearly Separable Classes with Uncorrelated Predictors

The results in this section are based on Simulation study 1 presented in Section 3.2.1

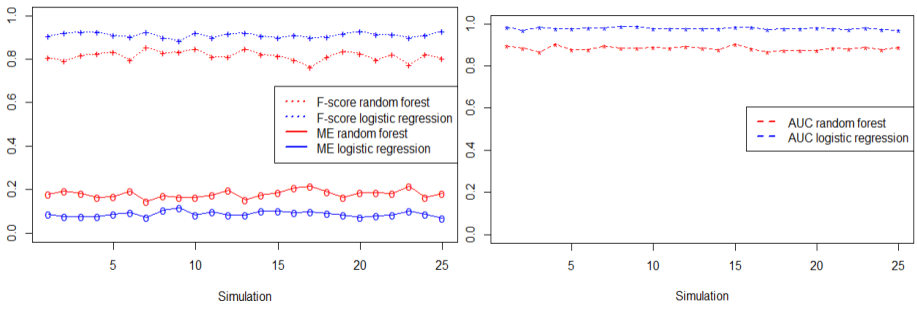

Figure 7: *The values of the misclassification rate, F-score and AUC in each simulation for logistic regression and random forest when the two classes are linearly separable but uncorrelated and* $n_{test} = 500$ *and* $p = 50$

Table 2: Results from Simulation study 1

	$\mathbf{n}_{test}$	p	AUC			F-score			ME		
			Ave.	Diff.	Sd.	Ave.	Diff.	Sd.	Ave.	Diff.	Sd.
LR	1000	100	0.9862	0.12	0.0032	0.9359	0.13	0.0091	0.0637	0.13	0.0084
RF			0.8623		0.0079	0.8012		0.0214	0.1966		0.0158
LR	1000	50	0.9792	0.08	0.0027	0.9205	0.09	0.0084	0.0795	0.08	0.0080
RF			0.9027		0.0037	0.8353		0.0123	0.1642		0.0120
LR	1000	10	0.9244	0.03	0.0086	0.8396	0.03	0.0139	0.1594	0.03	0.0129
RF			0.8943		0.0069	0.8123		0.0124	0.1861		0.0116
LR	500	50	0.9784	0.10	0.0050	0.9166	0.10	0.0139	0.0809	0.10	0.0109
RF			0.8832		0.0099	0.8172		0.0202	0.1778		0.0180
LR	500	10	0.9227	0.03	0.0125	0.8380	0.04	0.0177	0.1621	0.04	0.0155
RF			0.8879		0.0095	0.8017		0.0212	0.1979		0.0195
LR	100	10	0.9236	0.07	0.0321	0.8477	0.06	0.0466	0.1540	0.06	0.0440
RF			0.8498		0.0244	0.7896		0.0557	0.2160		0.0502

The standard deviations and the average values of misclassification rate, F-score and AUC for logistic regression and random forest when the classes are linearly separable but uncorrelated and the number of observations in the test set and the number of predictors vary. The size of the training data set is $n_{train} = 3n_{test}$.

4.2 Results from Linearly Separable Classes with Correlated Predictors

The results in this section are based on Simulation study 2 presented in Section 3.2.2

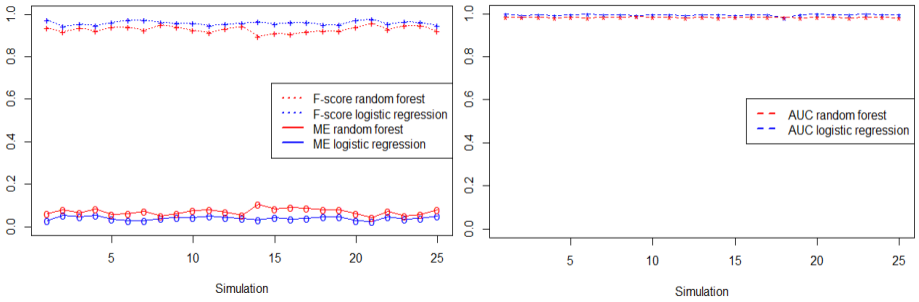

Figure 8: *The values of the misclassification rate, F-score and AUC in each simulation for logistic regression and random forest when the two classes are linearly separable and correlated with* $\rho = 0.8$ *and* $n_{test} = 500$ *and* $p = 50$

Table 3: Results from Simulation study 2

		$\mathbf{n_{test}}$	**p**	**AUC**			F**-score**			**ME**		
				Ave.	Diff.	Sd.	Ave.	Diff.	Sd.	Ave.	Diff.	Sd.
$\rho = 0.2$	LR	1000	100	0.9766	0.09	0.0044	0.9122	0.10	0.0117	0.0879	0.09	0.0116
	RF			0.8853		0.0086	0.8162		0.022	0.1810		0.0174
	LR	1000	10	0.9424	0.02	0.0064	0.8628	0.02	0.0107	0.1366	0.02	0.0110
	RF			0.9229		0.0042	0.8402		0.0151	0.1590		0.0151
	LR	500	50	0.9835	0.06	0.0036	0.9300	0.02	0.0123	0.0694	0.08	0.0116
	RF			0.9205		0.0061	0.8460		0.0185	0.1530		0.0146
	LR	500	10	0.9416	0.02	0.0112	0.8660	0.03	0.0140	0.1357	0.03	0.0140
	RF			0.9167		0.0072	0.8378		0.0185	0.1646		0.0205
	LR	100	10	0.8881	0.08	0.0293	0.8277	0.09	0.0332	0.1764	0.08	0.0341
	RF			0.8080		0.0242	0.7351		0.0621	0.2592		0.0476
$\rho = 0.8$	LR	1000	100	0.9837	0.003	0.0136	0.9697	0.04	0.0054	0.0301	0.04	0.0052
	RF			0.9804		0.0022	0.9304		0.0087	0.0690		0.0083
	LR	1000	10	0.9827	0.005	0.0023	0.9274	0.006	0.0098	0.0723	0.006	0.0089
	RF			0.9779		0.0025	0.9211		0.0088	0.0786		0.0079
	LR	500	50	0.9955	0.01	0.0016	0.9636	0.03	0.0082	0.0355	0.03	0.0079
	RF			0.9828		0.0022	0.9288		0.0136	0.0697		0.0132
	LR	500	10	0.9815	0.005	0.0036	0.9249	0.006	0.0118	0.0752	0.006	0.012
	RF			0.9765		0.0031	0.9194		0.0115	0.0807		0.0119
	LR	100	10	0.9095	-0.05	0.0744	0.8694	-0.006	0.0735	0.1312	-0.01	0.0678
	RF			0.9592		0.0234	0.8758		0.0770	0.1184		0.0583

The standard deviations and the average values of misclassification rate, F-score and AUC for logistic regression and random forest when the classes are linearly separable and correlated with $\rho = 0.2$ and $\rho = 0.8$ and the number of observations in the test set and the number of predictors vary. The size of the training data set is $n_{train} = 3n_{test}$.

4.3 Results from Non-linearly Separable Classes

The results in this section are based on Simulation study 3 presented in Section 3.2.3

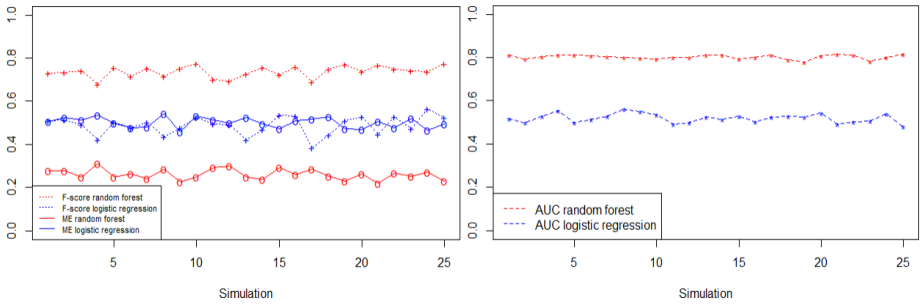

Figure 9: *The values of the misclassification rate, F-score and AUC in each simulation for logistic regression and random forest when the two classes are non-linearly separable but uncorrelated and* $n_{test} = 500$ *and* $p = 50$

Table 4: Results from Simulation study 3

	$\mathbf{n}_{test}$	p	AUC			*F*-score			ME		
			Ave.	Diff.	Sd.	Ave.	Diff.	Sd.	Ave.	Diff.	Sd.
LR	1000	100	0.5126	-0.26	0.0118	0.4970	-0.23	0.0235	0.5008	-0.23	0.0161
RF			0.7757		0.0097	0.7267		0.0191	0.2756		0.0151
LR	1000	10	0.5126	-0.38	0.0099	0.5379	-0.29	0.0392	0.5138	-0.33	0.0203
RF			0.8974		0.0136	0.8315		0.0151	0.1808		0.0037
LR	500	50	0.5156	-0.29	0.0177	0.4989	-0.23	0.0276	0.4998	-0.23	0.0227
RF			0.8055		0.0106	0.7328		0.0276	0.2681		0.0244
LR	500	10	0.5176	-0.36	0.0132	0.5146	-0.31	0.0458	0.5168	-0.33	0.0237
RF			0.8801		0.0070	0.8243		0.0234	0.1875		0.0193
LR	100	10	0.5483	-0.28	0.0297	0.4909	-0.27	0.0777	0.4900	-0.25	0.0458
RF			0.8249		0.0244	0.7592		0.0466	0.2400		0.0375

The standard deviations and the average values of misclassification rate, F-score and AUC for logistic regression and random forest when the classes are non-linearly separable but uncorrelated and the number of observations in the test set and the number of predictors vary.The size of the training data set is $n_{train} = 3n_{test}$.

5 Discussion

In this section, the results as we obtained in Section 4 will be explained and commented on. A discussion of improvements in the study will also be presented in this section.

5.1 Investigate the Predictive Accuracy

In Figure 7, we see the results in each simulation for Simulation study 1 when the number of observations in the test set is $n_{test} = 500$ and the number of predictors is $p = 50$. In this case, logistic regression performs better than random forest since both F-score and AUC are higher for logistic regression at each simulation. We can also see that ME is lower for logistic regression in comparison with random forest in each simulation in this case. Table 2 summarizes all results when the number of observations and the number of predictors vary. As expected, logistic regression predicts better than random forest in all this cases since the data set is simulated from the logistic regression model and as mentioned in Section 2.4, a parametric method outperform a non-parametric method if the estimated function f is similar to the true model as it is in this case. The biggest difference in the predictive power between logistics regression and random forest occurs when $n_{test} = 1000$ and $p = 100$, where logistic regression has a higher mean AUC by 12 percentage points, higher mean F-score by 13 percentage points and a lower mean ME by 13 percentage points.

We also obtain from the results in Simulation study 1, when the number of observations are fixed, the differences in mean ME, AUC and F-score between logistic regression and random forest increases as the number of predictors increases. The predictive power of logistic regression increases while the predictive power of random forest decreases. The reason to the increases in the predictive power of logistic regression depends on the fact that we generate our data set from the logistic regression model and let $\beta = \beta_1 = \beta_2 = \cdots = \beta_p = 1$. This makes the two classes more separated when the number of predictors p increases and this effect is in our case stronger than the one who tends to impair the ability of predict. A more detail explanation is represented in the appendix. For random forest, the reduction of predictive power is probably due to the classification trees becoming more complex as the number of predictors increases. These two effects together result in the increases of the differences in the mean ME, AUC and F-score between logistic regression and random forest.

Figure 8 shows the case for Simulation study 2, when $n_{test} = 500$, $p = 50$ and $\rho = 0.8$. Compare to Figure 7, the differences in mean ME, AUC and F-score between logistic regression and random forest seems to be reduced when the predictors are correlated but still, logistic regression preform slightly better than random forest in each simulation for this case. Table 3 indicates that the

differences for mean AUC, F-score and ME are quite small between logistic regression and random forest when $\rho = 0.8$. Logistic regression is somewhat better than random forest in the cases where n_{test} and p vary, except that case where $n_{test} = 100$ and $p = 10$. In this case, random forest preforms slightly better than logistic regression. The case where the biggest differences between the predictive power of logistic regression and random forest are obtained is when $n_{test} = 500$ och $p = 50$ which are represented in Figure 8. The cases where the predictors are less correlated ($\rho = 0.2$), is also presented in Table 3. Even here, logistic regression preforms better than random forest in all cases.

By examining Table 2 and Table 3, logistic regression predicts better than random forest in almost every case when the two classes are linearly separable but the differences between logistic regression and random forest decrease for all measures as the correlation between the predictors increases and n_{test} and p are fixed, except when $n_{test} = 100$, $p = 10$ and ρ increases from 0 to 0.2. Also the predictive power increases for both logistic regression and random forest as the correlation increases. The reason to why the predictive power of logistic regression increases is due to an increase in variance $V(\rho)$ for the projection of X onto to the direction v of the gradient of our sigmoid function $\boldsymbol{\sigma}(x)$ in the Bernoulli distribution. As X becomes more scattered along the gradient, the larger ρ is, the more separated two classes are. Even here, a more detail explanation is presented in the appendix. As mentioned, also random forest increases as the the correlation increase. An increased ρ means that fewer principal components are necessary to explain all variations in the data [12] which then makes it easier for random forest to perform well, apart from the fact that the two classes get more separated when ρ increases.

Lastly, the results from Simulation study 3 are presented in Table 4 and Figure 9. In this case, the two classes are non-linearly separable, since the observations from class 1 and class 0 can be separated by a circle. By examining the graphs in Figure 9, we can deduce that the difference between the measurements of predictive power differs markedly between random forest and logistic regression in the case where $n_{test} = 500$ and $p = 50$. Both F-score and AUC are much higher and ME is much smaller for random forest in each simulation of this study. In Table 4 the results are summarized for all cases when the number of observation and predictors vary for non-linearly separable classes. Since the differences have relatively large negative values for all the measures in all cases, we conclude, as expected based on the theory in Section 2.4, that the predictive power of random forest is better than for logistic regression since the relationship between the response variable and the predictors is highly non-linear. By examining mean ME, it can be seen that logistic regression predicts about half of the observations correctly and attains a mean AUC around 0.51. Based on the theory in Section 2.5.4, it implies that logistic regression performs as well as the classifier which randomly guesses a class to the observation and this in turn indicates that the model is a bad classifier when the classes are non-linearly separable as in Simulation study 3.

5.2 Further Investigation of the Work

Due to limited time, some delimitations have been made in this thesis which may be used for further investigations. Since the class of an observation was generated by the logistic regression model in our simulations for linearly separable classes, it resulted in that the true model and the estimated model for logistic regression were almost identical. As mentioned i Section 5.1, this had a major impact on the predictive power of logistic regression in those cases where the classes was linearly separable and it would be interesting to analyze the predictive power when the data set is simulated in a different way and not through the logistic regression model. The results for logistic regression in Simulation study 1 were affected by the fact that we chose $\beta = \beta_1 = \cdots = \beta_p = 1$, which resulted in the classes becoming more separate as p increased. Some improvements to the already existing simulation method that can counteract this effect are choosing $\beta_1 = \cdots = \beta_p = \frac{1}{\sqrt{p}}$. Then we expect that the predictive power of logistic regression may decrease slightly with increasing p. The two classes will then be equally separable for all p, but there are more parameters to estimate for a larger p.

We also mentioned in Section 5.1 that the results for logistic regression in Simulation study 2 are affected by the fact that we generate our data through the logistic regression model. The variance $V(\rho)$ of X along it first principal component increases when the correlation ρ increases and X becomes more scattered along the direction v which separates the two classes. This resulted in the classes becoming more separable. This effect can be avoided by normalizing the covariance matrix Σ of X, so that $V(\rho) = 1$ for all values of ρ.

An improvement in Simulation study 3 can be made by expanding the logistic regression model by adding a non-linear function of the predictors, such as $x = (x_1^2, x_2^2, \ldots, x_p^2)$. In this case, logistic regression will perform better and the estimated model will almost perfectly fit the data set. The problem with this is that one has to use the data to find out which non-linear functions of x_i that will be included as explanatory variables. In other cases, when the true boundary is unknown, it can be difficult and problematic to find an appropriate form of the logistic regression predictors when the classes are non-linear separable.

Improvement in random forest can also be made by using hyper-parameter tuning for random forest, which means that we for example can use cross validation to find the optimal number m of predictors used as split candidates in each split. In our case, we have used the recommended $m = \sqrt{p}$. It would also be possible to limit the number of trees that are constructed in random forest and examine different stop criteria that can optimize the random forest algorithm.

6 Conclusion

In this thesis, two different methods have been introduced, logistic regression a parametric method and random forest a non-parametric method. The purpose of the study was to compare and investigate the predictive power of them both and find the best classifier. The analysis was made both practically and theoretically. We conclude, based on the results from the simulation studies that logistic regression predicts better than random forest when the two classes of the data set were linearly separable and random forest predicts better when the classes was non-linearly separable. In the case were the classes could be separated by a circle , we found that logistic regression was an inappropriate method to use since the mean ME and AUC were around 0.5 which indicates that the classifier predicts about half of the observations correctly. We also discovered that two of the simulation studies were to the advantage of the logistic regression method since the data was generated through the logistic regression model. This resulted in logistic regression predicting very well when the two classes were linearly separable. Some suggestions for improvements were mentioned in view of reducing this impact. Some improvements of the logistic regression classifier were also proposed for non-linearly separated classes.

Appendix

Size of Trees

The mean size of the trees in each simulation study for each case is presented in Figures 10-13, where the size presented is calculated by taking the average of the 500 trees that random forest generate in one simulation.

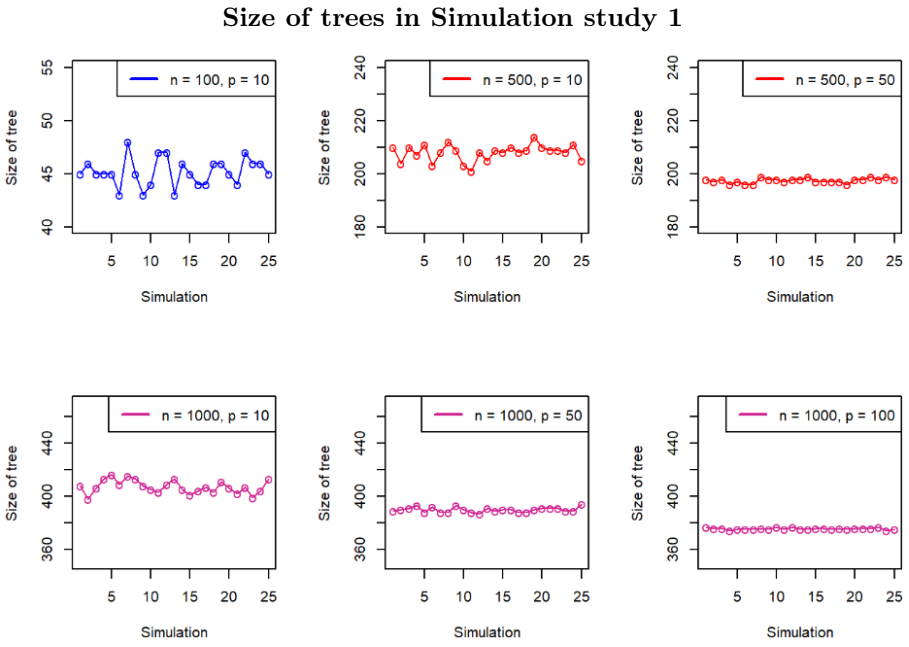

Figure 10: *The mean size of the trees for each simulation in Simulation study 1 where* $n = n_{test}$

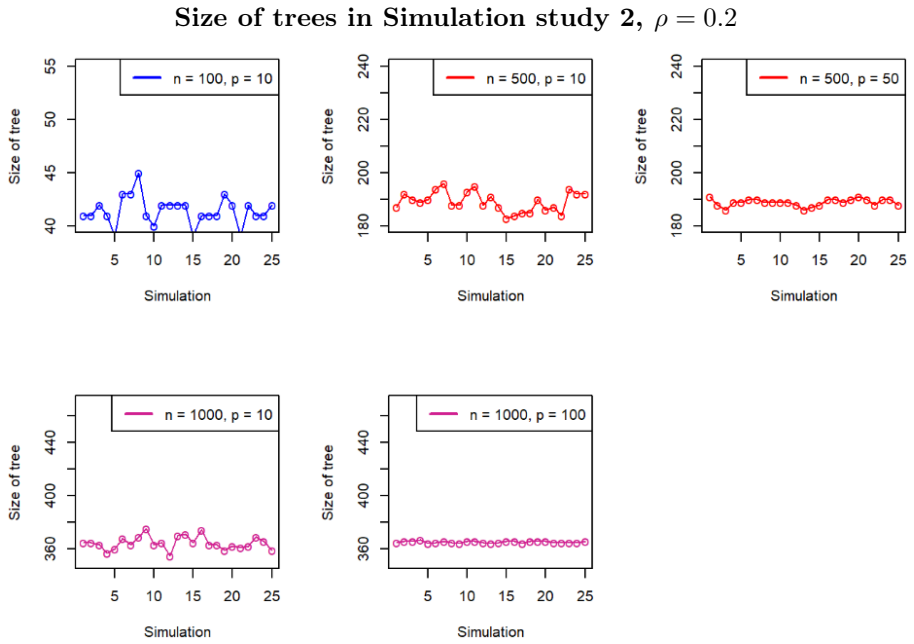

Figure 11: *The mean size of the trees for each simulation in Simulation study 2 when* $\rho = 0.2$ *and* $n = n_{test}$

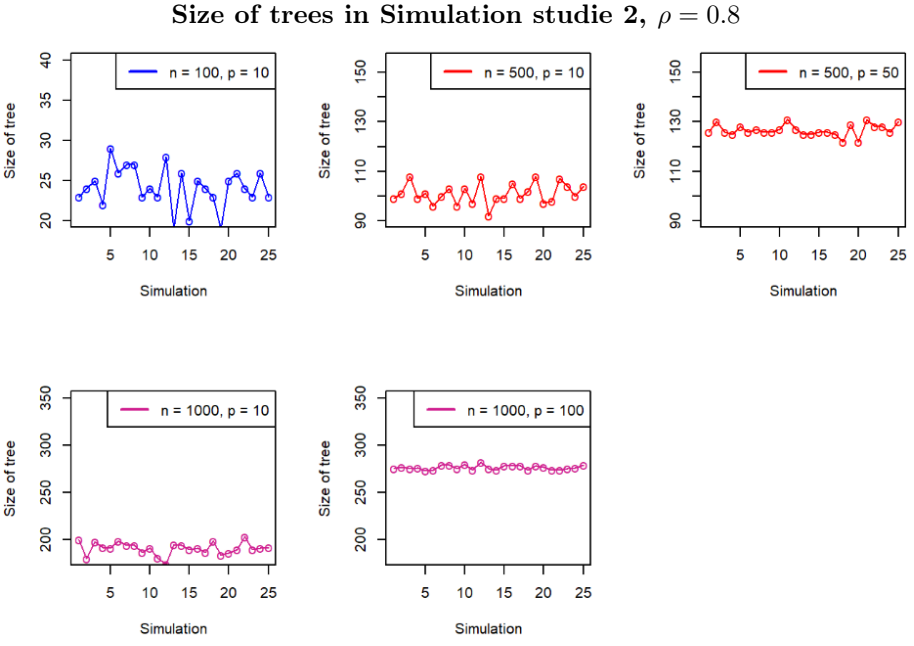

Figure 12: *The mean size of the trees for each simulation in Simulation study 2 when* $\rho = 0.8$ *and* $n = n_{test}$

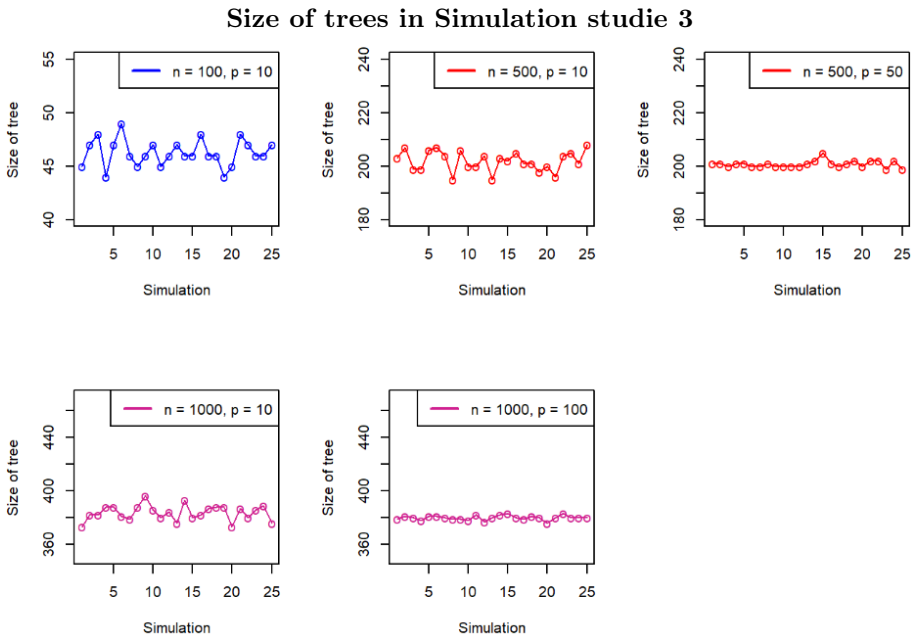

Figure 13: *The mean size of the trees for each simulation in Simulation study 3 where* $n = n_{test}$

Explanation of the Results

As discussed in Section 5.1, the results for logistic regression in the cases where the two classes are linearly separable are affected by the data being generated though the logistic regression model. A detailed explanation to why the predictive power of logistic regression increases when the number of predictors p and the correlation ρ increases will be present here.

Predictive power of logistic regression increases as the number of predictors increases

The reason why the separation between the two classes increases as the number of predictors p increases is due to the fact that our sigmoid function $\boldsymbol{\sigma}(x)$ in the Bernoulli distribution grows fastest (has its gradient) along the unit vector $v = (1, 1, \cdots, 1)/\sqrt{p}$ because we let $\beta_1 = \cdots = \beta_p = 1$. In Simulation study 1, we generate the predictors X from a standard normal distribution in p dimensions. This means that the projection $Z = v \cdot X$ along the gradient of $\boldsymbol{\sigma}(x)$ has a standard normal distribution $N(0, 1)$ in one dimension. The derivative of $g(x) = x_1 + \cdots + x_p$ in direction v is equal to $\sqrt{p}$, that is, it grows when p increases. This means that even $\boldsymbol{\sigma}(x)$ grows in the direction of v faster the larger p becomes. At the same time, the variance for X is kept constant in this direction and is not affected by the fact that p increases. This in turn means that the classes become more and more separated when the number of predictors p increases.

Predictive power of logistic regression increases as the correlation increases

Even in Simulation study 2, the sigmoid function $\boldsymbol{\sigma}(x)$ grows fastest along the direction v which implies that the optimum classifier consists of the hyperplane $g(x) = 0$ which is orthogonal to v. The reason to why the predictive power of logistic regression increases as ρ increases depends on the projection of X onto the direction v of the gradient. Since X is normally distributed N(0,Σ) then $Z = v \cdot X$ is normally distributed with variance $V(\rho) = v \cdot \Sigma \cdot v'$ where $V(\rho)$ increases as ρ increases. X is then more scattered along v the larger ρ is, which results in the classes becoming more separated.

References

[1] AGRESTI, A. (2012). *Categorical Data Analysis*. 3nd ed. Hoboken, New Jersey: John Wiley Sons, Inc.

[2] BREIMAN, L. (1996). Bagging Predictors *Machine Learning*, 24(2), 123-140. Boston: Kluwer Academic Publishers.

[3] BREIMAN, L. (2001). Random Forests *Machine Learning*, 45(1), 5–32. Boston: Kluwer Academic Publishers.

[4] BREIMAN, L. (2002). Manual On Setting Up, Using, And Understanding Random Forests V3.1. `https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf`

[5] CHINCHOR, N. (1992, June 16 - 18). MUC-4 Evaluation Metrics. `https://www.aclweb.org/anthology/M92-1002`

[6] FAWCETT, T.(2006). An introduction to ROC analysis *Pattern Recognition Letters*. 27(8), 861-874. Elsevier.

[7] HASTIE, T., TIBSHIRANI, R. & FRIEDMAN, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.

[8] HOSMER, D.W. & LEMESHOW, S. (2000). *Applied Logistic Regression*. 2nd ed. New York: John Wiley Sons, Inc.

[9] HURWITZ, J. & KIRSCH, D. (2018). *Machine Learning For Dummies*. Hoboken, New Jersey: John Wiley Sons, Inc.

[10] IZENMAN, A.J. (2013). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. 2nd ed. New York: Springer. E-book.

[11] JAMES, G., WRITTEN, D., HASTIE, T. & TIBSHIRANI, R. (2017). *An Introduction to Statistical Learning: with Applications in R*. 8th ed. New York: Springer. E-book.

[12] JOLLIFFE, I.T. (2002) *Principal Component Analysis*. 2nd ed. New York: Springer.

[13] LIAW, A. & WIENER, M. (2018). Breiman and Cutler's Random Forests for Classification and Regression. `https://cran.r-project.org/web/packages/randomForest/randomForest.pdf`

[14] MELLO, R.F.D. & PONTI, M.A. (2018). *Machine Learning: A Practical Approach on the Statistical Learning Theory*. Cham, Switzerland: Springer. E-book.

[15] NÖU, A. (2017, June). *Logistic Regression versus Support Vector Machines*. Stockholm University. https://www.math.su.se/publikationer/uppsatsarkiv/.

[16] ROSS, S.M. (2010). *Introduction to Probability Models*. 10th ed. Amsterdam: Academic Press Inc.

[17] RUSSELL, S. & NORVING, P. (2010). *Artificial Intelligence: A Modern Approach*. 3rd ed. Harlow, England: Pearson Education, Inc.

[18] SASAKI, Y. (2007, October 26). *The truth of the F-measure*. https://www.researchgate.net/publication/268185911_The_truth_of_the_F-measure

[19] SHALIZI, C.R. (2019, April 1). *Advanced Data Analysis from an Elementary Point of View*. https://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/ADAfaEPoV.pdf.

[20] SINGH, S. & GUPTA, P. (2014). Comparative Study ID3, CART and C4.5 Decision Tree Algorithm: A Survey. *International Journal of Advanced Information Science and Technology*. 27(27), 861-874.

[21] SMOLA, A. & VISHWANATHAN, S.V.N. (2008). *Introduction to Machine Learning*. Cambridge, United Kingdom: Cambridge University Press.

[22] TEAM, R. C. (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.

[23] ZHANG, C. & MA, Y. (2012). *Ensemble Machine Learning: Methods and Applications*. Boston: Springer.