Master Thesis

Reliability of Selective Credit Risk Models

Identifying and mitigating bias in PD model statistics after model selection and cross validation.

Author:	W. van den Brink
Date:	25 August 2022
Supervisors:	dr.ir. W.J.A. van Heeswijk
	dr. B. Roorda
	ir. J. Muis
	dr. V.L. Tchistiakov

Colophon

Title	Reliability of Selective Credit Risk Models
Master Thesis	In fulfilment of a Master of Science Degree in Industrial Engineering and Management
Program	Industrial Engineering and Management
Specialization	Financial Engineering and Management
Additional specialization	Healthcare Engineering

Author	W. van den Brink
Contact	w.vandenbrink-1@student.utwente.nl
Company	Coöperatieve Rabobank UA

Exam committee	dr.ir. W.J.A. van Heeswijk (University of Twente) dr. B. Roorda (University of Twente) ir. J. Muis (Rabobank) dr. V.L. Tchistiakov (Rabobank)

Definitions

Term	Definition
Credit risk	*The risk of a borrower not paying its obligations completely.*
Default	*A borrower not paying its obligations completely.*
Regressor x	*Predictor variable, referred to as x.*
Regressor parameter β	*Weight of a regressor, referred to as β.*
Output variable y	*The response variable PD, can be referred to as y.*
Intercept β_0	*If all (weighted) regressors equal zero, this is the output of a regression model. Referred to as β_0.*
Linear model	$y = \beta_0 + \sum_i \beta_i x_i + \epsilon$.
Logistic model	$ln\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{j=1}^{n} \beta_j x_j$.
t-value	*A test statistic that can be compared to a tail-value of the student's T distributions to conclude on the power of a regressor.*
Null Hypothesis H_0	*Commonly, a null hypothesis may be $H_0 : \beta_{x_1} = 0$. This hypothesis can be rejected or accepted with a certain confidence or power.*
Power	*The degree to which a regressor can be used to predict PD.*
p-value	*The probability of observing a more extreme t-value under H_0 given an observed t-value.*
Type-I error	*Rejecting H_0 while it is true.*
Inference	*In this research, drawing inference is the science of establishing a relationship between a variable of interest y and a regressor x within a certain model.*
Prediction	*A regression model can be suitable for making predictions. That is, observed data $\boldsymbol{x}$ is placed in a regression model to estimate e.g. a PD.*
Biased model statistics	*See biased inference and biased prediction.*
Biased inference	*Post model selection β and t-value sample distributions can be different from their theoretical distributions, possibly resulting in wrong estimates of the true value of βs and wrong conclusions regarding the power of regressors.*
Biased Prediction	*The performance of a model may be artificially inflated by certain model selection and performance evaluation techniques.*
Bootstrapping	*Drawing samples from a finite data set.*
Ordinary Least Squares (OLS)	$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$
Lasso	$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \lvert\beta_j\rvert$.
Log Likelihood	$\mathcal{L}(\boldsymbol{\beta}) = \sum_i y_i (\ln \hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$
Lasso Logit	$\min_{\boldsymbol{\beta}} -\mathcal{L}(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \lvert\beta_j\rvert$

Abbreviations

Abbreviation	Definition
PD	*Probability of default.*
LGD	*Loss Given Default.*
EAD	*Exposure at Default.*
EL	*Expected Losses.*
RR	*Recovery Rate.*
CSMU	*Credit Model Strategy, Methodology and Monitoring Unit.*
RA	*Risk Analytics.*
OLS	*Ordinary Least Squares.*
MSE	*Mean Squared Error.* $$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predicted} - y_i^{true}\right)^2.$$
MAE	*Mean Absolute Error.* $$MAE = \frac{1}{n}\sum_{i=1}^{n}\left\lvert y_i^{predicted} - y_i^{true}\right\rvert.$$
AIC	*Akaike Information Criterion.* $$AIC = 2k - \ln(\hat{\mathcal{L}})$$
BIC	*Bayesian Information Criterion.* $$BIC = k\ln(n) - 2\ln(\hat{\mathcal{L}})$$
LLF ($\hat{\mathcal{L}}$)	*Log-likelihood function* $$LLF = \sum_i y_i(\ln \hat{y}_i) + (1 - y_i)\ln(1 - \hat{y}_i)$$

Symbols

Symbol	Definition
β	Regressor parameter.
$\boldsymbol{\beta}$	A vector containing regressor parameters, e.g. $\boldsymbol{\beta} = [\beta_1 \quad \dots \quad \beta_p]$
β_0	Intercept.
y	Outcome variable, variable of interest.
$\hat{y}$	Prediction of a regression model.
x	A regressor.
$\boldsymbol{x}$	A set of regressors.
$\boldsymbol{x}^j$	A vector containing data $(x_{j1}, \dots, x_{jn})^T$ with n observations, labelled j with , $j = 1,2, \dots, p$.
p	Depending on the context, p can be a probability or the number of regressors in a data set.
λ	A penalty constant in the lasso objective.
$\mathcal{L}$	Log-likelihood of a logistic regression model.
N	Data size in terms of number of observations.
n	Sample size with which we draw from N.

Preface

Dear reader,

With this thesis I conclude my student life at the University of Twente, upon which I happily reflect. I am grateful for the opportunity to finish my master's with research commissioned by Rabobank, which would certainly not have been possible without the help of the people that this preface is dedicated to.

Thanks to Viktor Tchistiakov for his patience and helpful insights. I thank Joost Muis for his help in getting acquainted with the software and computer programming at Rabobank. I further thank CSMU for making my time at Rabobank enjoyable. Thanks to Wouter van Heeswijk and Berend Roorda, the UT supervisors, for their helpful feedback. Lastly, thanks to the reader for taking the time to read this thesis, hopefully it will aid you in improving your own research.

Wessel van den Brink
Zwolle, August 2022

Table of Contents

Management Summary

This management summary introduces the problem of biased probability of default (PD) model statistics and then discusses an approach to quantifying this bias using computer simulations. Thereafter, results of the simulations are discussed, based on which conclusions and recommendations are presented.

Problem

This research is commissioned by Coöperatieve Rabobank UA, where probability of default (PD) models are used to estimate the probability a counterparty will default on a loan. These PD models are engineered using model selection algorithms. PD model statistics (regressor parameters β and t-values) after model selection, however, are not always aligned with reality, possibly making regressors appear to have stronger power than they actually have. This phenomenon is referred to as biased inference. This makes it difficult for modellers to conclude on what the true model responsible for generating the variable of interest (PD) is, if such a model even exists. Furthermore, inflated model performance as a result of k-fold cross validation is discussed, potentially making it more difficult for modellers to estimate the true performance of a model.

Reliable models are crucial for banking, hence we study how model statistics can be biased after model selection and how to reduce or avoid this as best as possible. From this objective we define the following research question: *In what way, if at all, are the regressor-parameters and model performance of credit risk models at Rabobank biased, and how can Rabobank adjust for this bias both statistically and managerially?*

Approach

According to literature, we can opt for a Monte Carlo simulation style approach to recognize biased model statistics. To investigate **biased inference**, we use a model with known parameters to generate defaults. Specifying a known model allows for a reference point to which results can be compared. Data samples are generated from this known model, on which different combinations of model selection algorithms and data strategies can be tested. In order to adjust for biased inference, we found that according to the literature, modellers can make use of data splitting, data carving, or adding noise to the data during the model selection stage. These data strategies are tested on relative performance to each other and to using the same data sample for both model selection and model fitting. Furthermore, we found in the literature that using Lasso as a model selection algorithm may reduce biased inference, hence it is tested as well.

To investigate **biased model performance**, a similar Monte Carlo simulation can be utilized, but the data must be such that we have a controlled true metric of how well the data can be used for predictive modelling. In order to achieve this we use a method described by Moshontz et al. (2020). This method entails generating random data and assigning defaults to this random data with a certain probability. So, generated defaults are completely unlinked to the data. This means that the true model performance is controlled when we express performance by means of a metric that measures the probability of ranking a random positive case higher than a randomly selected negative case (the area under the curve (AUC) of the receiver operating characteristic curve), as it can be argued that this metric should equal 0.5. On this random data different combinations of model selection algorithms and data strategies can be tested. Biased model performance can be quantified by observing how the reported AUC deviates from 0.5.

Results

Biased inference can occur when using the same data sample for both model selection and fitting the selected model. This bias is expressed in terms of irrelevant regressors that are not actual risk drivers of PD getting selected in a model, as well as being accepted after fitting the selected model. A promising solution, based on the results of the experiments, is to **split** a data sample in a model selection part and a model fitting part. The model selection part should have a size of 25% of the data sample, and the model selection algorithm should be Forward stepwise. The remainder (75%) of the data sample should be used to fit the selected model. Compared to the baseline of using the same data sample for Forward stepwise model selection and subsequent fitting, the proposed solution reduces the probability of selecting irrelevant regressors in a model by 15% on average. After fitting the selected model, the probability of accepting irrelevant regressors in a model is reduced by 86% on average.

Model performance appears to be more inflated when increasing the number of regressors in the data set, and/or reducing the size of the data set. We observe that the average model performance over k-folds when using cross validation is least affected by inflated performance, compared to a separate test-set outside of the k-folds. The risk of inflated model performance diminishes no matter the model selection algorithm or data strategy when enough data is available. Modellers can make use of the tables in this research to determine if the risk of inflated model performance is too high. These tables show empirical probabilities of model performance being inflated, given the model selection algorithm, data strategy, data size, and reported model performances.

Conclusions

To answer the main research question: statistics, t-values as well as βs, of regressors post model selection can be biased such that they inaccurately reflect their true value. As a result, significant risk of irrelevant regressors being added to a final model is present when using the same data sample for both stepwise model selection and subsequent fitting. Model performance can be inflated as a result of model selection, but with sufficient data this problem is insignificant. From the experiments and the analysis of the data, the following recommendations on improving the PD model selection methodology of Rabobank may be formulated:

- To, first and foremost, be aware of biased inference with PD model selection and;
- To, when developing models, use Forward stepwise in combination with data splitting, where 25% of a data sample is used for model selection and the remaining 75% for model fitting and;
- To make use of bootstrapping in practice, as from the results of Forward stepwise combined with data splitting we are able to recognize the true model responsible for generating defaults when bootstrapping via reported t-values after model fitting and;
- To, given enough data, use Forward stepwise combined with data splitting and 5-fold cross validation to obtain an estimate of how well the data at hand can be used for predictive modelling; a modeller can judge the risk of inflated model performance using the tables in this research.

1. Introduction

This research is about biased statistics in regression models as a result of model selection, and how Rabobank can be subject to different types of bias as a result of using model selection algorithms for constructing probability of default (PD) models. This chapter is structured such that we first establish the necessary background information on theory of regression, inference, and model selection (section 1.1) and on Rabobank (section 1.2) to finally introduce the problem by means of a casus-description (section 1.3).

1.1 Introductory Background Theory

In this section we first give an introduction to linear (section 1.1.1) and logistic (section 1.1.2) regression models. Thereafter (section 1.1.3) the idea of statistical inference is discussed as well as model selection. Lastly (section 1.1.4) an experiment is conducted to illustrate how establishing statistical inference after model selection can be problematic.

Rabobank, as part of a set of financial services, lends money to clients ranging from other financial institutions and corporate businesses to retail clients. Needless to say, money-lending involves the risk that the counterparty goes into default, thereby potentially leaving Rabobank with losses. This risk is defined as credit risk. Quantifying credit risk can be done with credit risk models, which are, for example, used in estimating the probability of default (PD): the probability that the counterparty does not pay back the loan. Credit risk models may be regression models such as linear and logistic regression models for drawing inference and estimating default probabilities. Needless to say, for both the bank and its customers it is crucial that these models are reliable.

1.1.1 Linear Regression

The idea of simple linear regression is to estimate a value y given an input x. The general form of a simple linear regression formula $y = \beta_0 + \beta x + \epsilon$ shows this relationship clearly: imagine a two dimensional graph (Figure 1) with a horizontal x-axis and a vertical y-axis where observations in terms of (x, y) coordinates are plotted, a line passes through these coordinates with regressor parameters $\boldsymbol{\beta}$ such that, e.g., the sum of the squared distances between the line and the observations, called residuals, is minimized by e.g. linear algebra. In the general form ϵ is a random error term that is irreducible.

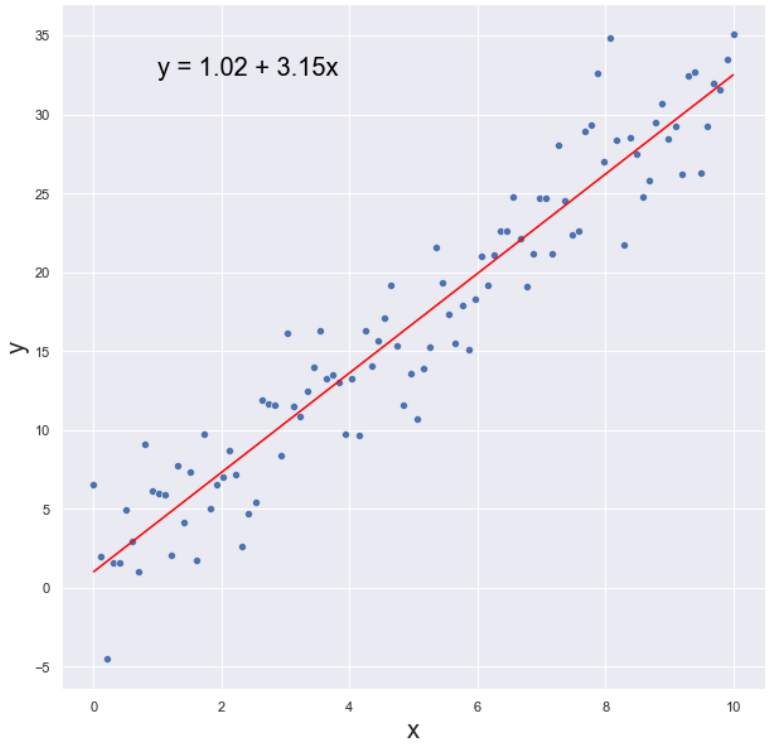

Figure 1: An example of a linear model (the red line) based on natural observations of $(x,\ y)$ coordinates (the blue dots).

The output of a linear regression model may be seen as an estimate for y conditional on the value of regressor(s) $\boldsymbol{x}$. If multiple regressors $\boldsymbol{x}$ seem to predict the variable of interest y the general form, called multiple linear regression, looks like: $y = \beta_0 + \sum_i \beta_i x_i + \epsilon$.

1.1.2 Logistic Regression

A logistic regression model can be used is to estimate the probability that an observation belongs to the positive outcome, and can therefore be used in binary classification problems. Predicting defaults can be a binary classification problem, as a client defaults on a loan or does not.

When PD is defined as a default probability (PD), $\boldsymbol{\beta}$ as the regressor parameters, β_0 as the y-axis intercept, and $\boldsymbol{x}$ as the regressors x_i, $i = 0, \dots, p$. Then the general form of a logistic regression model is

$$\ln\left(\frac{PD}{1-PD}\right) = \beta_0 + \sum_{i=1}^{p} \beta_j x_j \,.$$

Notice the right-hand side is equivalent to that of linear regression but the left-hand side is a transformation of the PD. Obtaining PD values is performed via the following transformation: $PD = \frac{1}{1+e^{-\left(\beta_0+\sum_{i=1}^{p}\beta_j x_j\right)}}$. Details of logistic regression are discussed in the next chapter.

1.1.3 Inference & Model Selection

1.1.3.1 Statistical Inference and t-Values to Measure Power

Deciding which regressors to include in a linear or logistic model can be done in a number of ways. First, there is the classical way where we hypothesize a relationship between the outcome variable y and a regressor x_1. This hypothesized relationship can, e.g., be linear or logistic. To test if regressor

x_1 explains the outcome variable y a null hypothesis H_0 is defined as $H_0 : \beta_{x_1} = 0$. Notice that if H_0 is not rejected then regressor x_1 is insignificant in explaining the outcome variable y.

For illustrative purpose, assume a linear model is developed and β_{x_1} is estimated using data. With n data points, we transform this estimate of β_{x_1}, denoted as $\widehat{\beta_{x_1}}$, in a t-value as follows:

$$t_{\beta_{x_1}} = \frac{|\widehat{\beta_{x_1}}| - H_0(\beta_{x_1})}{SE(\widehat{\beta_{x_1}})} = \frac{|\widehat{\beta_{x_1}}|}{\sqrt{\frac{1}{n-2} * \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (x_i - \bar{x})^2}}}$$

where $H_0(\beta_{x_1})$ is the value of β_{x_1} under H_0 and $SE(\widehat{\beta_{x_1}})$ and the standard error of the estimate $\widehat{\beta_{x_1}}$ (Zach, 2021). Based on $t_{\beta_{x_1}}$ the significance of the regressor parameter estimate $\widehat{\beta_{x_1}}$ can be determined by comparing it to the $1 - \frac{\alpha}{2}$ percentile tail value of a continuous distribution. In the case of linear relationships this is the t distribution (see e.g. Federighi, 1959), with degrees of freedom df equal to the number of observations minus two, and where, commonly, significance level $\alpha = 5\%$. If $t_{\beta_{x_1}} \geq t_{\alpha=5\%,df=n-2}$ then H_0 can be rejected and we can conclude that regressor x_1 explains the outcome variable y. Intuitively, if $t_{\beta_{x_1}}$ is much larger than the threshold of acceptance $t_{\alpha=5\%,df=n-2}$ then we speak of a regressor that has strong **power**. Generally, the smaller the value of α exceeded the stronger the inference, because α measures the maximum probability of the type-I error: the probability of H_0 being correct given the data; also referred to as a p-value. In the case of testing logistic relationships, the distribution the test statistic is compared to is the standard normal curve and is sometimes referred to as a z-value.

It must be noted that the standard error, and thus the t-values, of regressors in multiple linear regression (MLR) analyses depend on the inclusion of other regressors in the analysis (see e.g. The Pennsylvania State University, n.d.). Formulae for the standard error with multiple linear regression are implemented in statistical software such as the *statsmodels*[1] Python package, and for the sake of brevity only the concept of standard error must suffice: the estimate $\widehat{\beta_{x_1}}$ is sample-dependent, meaning that if we perform the above statistical test on another data-sample the estimate $\widehat{\beta_{x_1}}$ will be different, $SE(\widehat{\beta_{x_1}})$ describes this uncertainty and adjusts t-values accordingly. This concept holds no matter the type of regression, simple, multiple or logistic.

Concluding, the phenomenon of regressor x_1 explaining the outcome variable y significantly given a linear or logistic model is referred to as statistical inference. The above procedure can be performed to verify hypothesized relations between regressors and outcome variables for both linear and logistic relations.

1.1.3.2 Model Selection

When dealing with many regressors in a dataset statistical algorithms can be used to find hypothesized relationships for us. That is, statistical algorithms can select the models for us. This common practice is named model selection.

Suppose we have data with p regressors and an outcome variable y. Here, p can be a very large number. One model selection algorithm could be to generate and compare all the possible different models, this is called **best-subset selection**. This method of model selection may take an unreasonable amount of time (discussed in section 2.2.2.3), so another method could be to start

[1] Statsmodels v0.13.2: https://www.statsmodels.org/stable/index.html

with a model that has one regressor, and keep adding regressors until no improvements can be made, this is called **forward-stepwise** model selection. Numerous model selection algorithms exist, the relevant of which will be discussed in the next chapter.

1.1.3.3 Model Performance

The definition of a best model that is the result of a model selection algorithm depends on the type of problem and thus on the type of regression. For linear regression, an example of a common performance metric is the mean squared error (MSE) which squares and summates every prediction error and is calculated by: $MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predicted} - y_i^{true}\right)^2$.

An example of a common performance metric for logistic regression models, when dealing with binary classification problems, is the area under the receiver operating characteristic curve (AUC), which is a value between 0 and 1 that is an estimate of how well a model can distinguish between binary outcomes as the AUC is an estimate of the probability of a model predicting a higher score for an arbitrarily selected positive case than for an arbitrarily selected negative case (Fawcett, 2006; Moshontz, Fronk, Sant'Ana & Curtin, 2020).

1.1.4 Biased Inference After Model Selection

One can imagine searching for models, iteratively comparing the generated models based on suitable performance metrics, and selecting the best one. However, when researching which regressors explain the variable of interest best using a model selection algorithm, the search (the model selection algorithm) that such a conclusion is the result of must be taken into consideration before concluding on the power of a regressor, as we will see in this section. Recall that power is defined as how certain we are of the relationship between a regressor and the output variable, and a measure of power is a t-value (section 1.1.3.1).

Conditioning on the fact that the observed power of regressors depend on the model selection algorithm can be referred to as the science of establishing **statistical inference after model selection**. Berk, Brown & Zhao (2010) show that when model-selection methodologies, such as the forward-stepwise algorithm, where regressors are iteratively added to a model if the performance increases significantly, are used for building linear models strangely shaped mixed-sampling distributions of regressor parameters and corresponding t-values can arise. This occurs when the selected model is fitted using the same data it was found with. As a result, regressors may show inflated power.

To illustrate the phenomenon of biased regressor parameters, observe Figure 2 that is the result of an experiment, designed after Berk, Brown & Zhao (2010), using a forward stepwise algorithm. Pseudo code for the experiment is given in Table 1. The experiment can be summarized as follows: data from a known model is sampled a thousand times and on each sample a forward stepwise model selection algorithm is used, the outcome (i.e. the selected model) of this algorithm is then fitted to find final regressor parameters and corresponding t-values of a particular regressor if it is present in the set of selected regressors.

Table 1: Model selection algorithm on data from a known model.

Pseudo-code for generating a t-value sample distribution (after Berk, Brown & Zhao, 2010)

#A linear model is defined
$y = \beta_0 + \beta_1 w + \beta_2 x + \beta_3 z + \epsilon$
#Parameters are set (intercept, w, x, z)
$\beta_0 = 3,\ \beta_1 = 0, \beta_2 = 1\ ,\ \beta_3 = 2$
#Experiment parameters are defined
$samples = 1000$
$sampleSize = 200$
#The multivariate normal data are distributed as follows:
$mean = [0, 0, 0]$,$cov = [[5,4,5][4,6,5][5,5,7]]$
#The error term ϵ is defined as
$\epsilon \sim Normal(mean = 0, standardDeviation = 10)$

#In a for loop over *samples*, data is sampled from the multivariate normal distribution. The forward model selection algorithm is used on this data sample. Regressor parameters are found using ordinary least squares regression provided by the *statsmodels* library.

For i in $range(samples)$:
    #Draw a sample, the result is a matrix with 4 columns representing regressors w, x, z, and a constant β_0.
    $X = np.random.multivariate\_normal(mean, cov, size = sampleSize)$
    $X['constant'] = 1$
    #Generate error array
    $\epsilon = np.random.normal(mean = 0, standardDeviation = 10, size = sampleSize)$
    #Compute y vector from X and ϵ
    $for\ i\ in\ range(sampleSize)$:
        $y.append(\beta_0 + \beta_1 w[i] + \beta_2 x[i] + \beta_3 z[i] + \epsilon[i])$
    #Use the model selection algorithm to find a model
    $forwardModel = Forward(X, y, criterion = MSE).run()$
    #Store set of regressors selected by the model selection algorithm
    $variables = forwardModel.report['Variables']$
    #Fit a linear model to the sample using the selected regressors. Store *t*-values of regressor *x*
    $LinearModel = LinearRegression(y, X['Variables']).fit()$
    $tValues.append(linearModel.tvalues['x'])$
#Plot the *t*-values in a histogram
$PlotHistogram(tValues)$

The same procedure (Table 1) without forward model selection is performed using the same method of data generation, with an equal number of samples and an equal sample size. A linear model is fitted using the known model (i.e. regressor w is left out of the data used to fit a linear model, because its β equals 0) and the t-values for regressor x reported by the *statsmodels* software are stored and plotted.

A thousand samples were drawn and hence a thousand models were selected (Table 1), regressor x was part in 581 of the thousand selected models, the result is visualized in Figure 2. Notice that the t-value sample distribution of the forward stepwise model selection based on the MSE of models is exhibiting symptoms of bias. That is, the whole t-value histogram is moved to the right and is bimodal, it contains more extreme t-values in contrast to the true t-values.

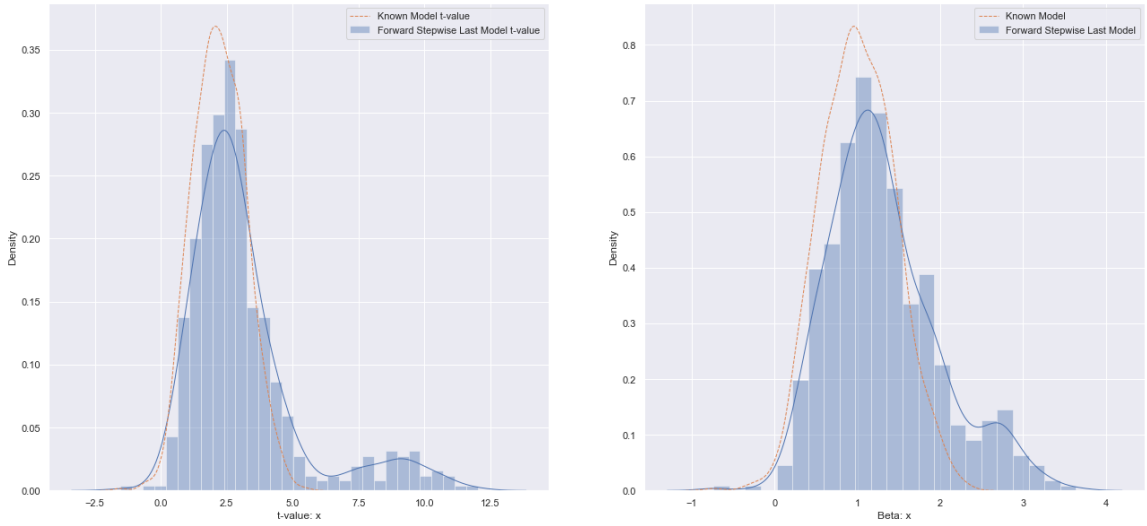

Figure 2: Output of an experiment to visualize bias after Berk, Brown & Zhao (2010). The histograms in blue are the reported values after model selection and fitting, the density functions in orange are the results on fitting the known model. On the left a bimodal t-value sample distribution can be observed, with false power as a consequence. On the right, a skewed sample distribution of β can be observed, resulting in a wrong estimate of the true β.

Ultimately, a p-value that describes the probability that a regressor is insignificant belongs to a t-value. As discussed, a common threshold is a maximum **type-I error** of 0.05, this is also called a significance level of 0.05. Based on the experiment of Table 1, conditional on Rabobank forward-stepwise MSE accepting x in a model, the empirical probability of accepting regressor x as significant is 0.70. The empirical probability of accepting regressor x as significant when the model is known is 0.58. Now the idea of **biased inference after model selection** becomes clear.

Berk, Brown & Zhao (2010) state that the shape of these sampling-distributions determine the severity of potential consequences in power, and are influenced by (i.) the model selection directly, (ii.) the in- and exclusion of other regressors in the linear model, (iii.) the dispersion of the regression parameter itself, and (iv.) the interactions of these three mechanisms.

Concluding, drawing inference from regressors in models must, somehow, take into consideration the search that is model selection, in the sense that models have a certain probability of being selected. This is difficult to achieve because the value of regressor parameters $\boldsymbol{\beta}$, and thus of their corresponding t-values, depend on complex interactions of the discussed mechanisms.

1.2 Scope & The Context of CSMU

The Credit Model Strategy, Methodology and Monitoring Unit (CSMU) is part of the Risk Analytics (RA) department of Rabobank. The topic of this research fits in the CSMU mission, which is *to maintain trustworthy and valuable models for the bank's decision making of Rabobank's customers*. Identifying and correcting for statistical bias in regression models, which are used for decision making, aligns with this mission.

In more detail, CSMU delivers three services:

- Model monitoring and back-testing, which involves assessing and reporting model performance and;
- Model strategy and control, which involves making decisions regarding the future model landscape and;
- Model methodology, which involves developing methodologies, policies and procedures related to, for example, credit risk modeling.

Credit risk management should take into consideration the (possibility of) bias in models. The task of identifying and correcting for bias is an example of model methodology, and may have implications for model strategy. There are procedures available within the bank that elaborately describe how credit risk models should be developed. From a CSMU perspective, it is important to understand if the policies that are in place result in biased credit risk models. Hence this thesis focusses on identifying this bias and how to correct for this when developing a credit risk model instead of model development itself.

1.3 Casus Description

Banking is largely about modelling, profitable banking is about understanding the unreliability of models. At Rabobank this is no different, and to ensure a tailored solution we must establish, from the current situation, a problem. From this problem follows a research goal. Based on this research goal follows a plan of approach, containing research questions.

1.3.1 Situation and Problem

When a client, being a financial institution, a corporate client, a retail client, or any other client, applies for a loan, the applicants data is input to a credit risk model. This credit risk model estimates the probability that the client will default on the loan applied for. At Rabobank, these PD models can be in the form of logistic regression models, which contain specific (classified) regressors. These regressors are selected using a combination of linear and logistic regression techniques after which inference can be established from t-values.

After regressors are selected, for example, we want to estimate a PD based on the educational attainment of the applicant, weights must be established in order for the model to be useful for drawing inference or prediction. These weights are the result of complex interactions of (i.) the model selection directly, (ii.) the in- and exclusion of other regressors in the linear model, (iii.) the dispersion of the regression parameter itself, (iv.) and interactions of these three mechanisms (Berk, Brown & Zhao ,2010)

The idea of biased inference after model selection is as follows: a model is constructed using a model-selection methodology and regressor parameters are subsequently found, but these parameters are dependent on, at least, the data-sample and, as we have demonstrated in section 1.1.4, on the model the regressor is present in. This makes it unclear what the true value of a

regressor parameter should be, if such a value exists, from which inference is drawn and models are constructed.

Then, when a model is the correct one, assuming a correct model exists that accurately describes how the real-world data is actually generated, going with the average results of the sample distribution can result in systemic bias (Berk, Brown & Zhao ,2010). Such bias, as discussed, is difficult to identify due to the shape of the distribution being the result of interactions of multiple mechanisms. Banks have to cope with this phenomenon somehow otherwise mismanagement may occur by relying too much on models, and a logical starting-point is to identify, and later correct for, this bias.

1.3.2 Research Goal

The goal of this research is to identify and correct for bias in model statistics that may occur with credit risk model selection and, based on this bias, give Rabobank advice on credit risk management improvements.

1.3.3 Plan of Approach and Research Questions

In order to reach the research goal (section 1.3.2) a plan containing deliverables and research questions is made. Here, the research is split into different parts. First, we define the main research question: *In what way, if at all, are the regressor-parameters and model performance of credit risk models at Rabobank biased, and how can Rabobank adjust for this bias both statistically and managerially?*

In the second chapter, a literature review is conducted where the goal is to answer the following sub-questions:

- According to the literature, how to identify bias in model statistics for linear regression?
- According to the literature, how to identify bias in model statistics for logistic regression?
- According to the literature, how may Rabobank correct for bias in model statistics for linear – and logistic regression models?

In the third chapter, we synthesize the data found in chapter two such that a method to identify bias in (selected) credit risk models for Rabobank can be constructed. In other words, this is a paper-model or methodology. The research question can then be:

- What does a conceptual model to find bias in selected Rabobank credit risk models look like, and which techniques that are expected to reduce this bias should be added to this conceptual model?

In the fourth chapter, we conduct experiments using the model from chapter three using Rabobank data. Eventually, these experiments yield the results based on which conclusions to the main research question are formulated. A suitable research question for this chapter:

- Based on the conceptual model of chapter three, how are regressor parameters and model performance of credit risk models biased as a result of Rabobank model selection algorithms and how are these model statistics biased as a result of the hypothesized solutions?

In the fifth chapter we analyse the results of these experiments, and based on these results, in chapter seven, we draw conclusions and give recommendations such that the main research question, defined above, is answered. The chapter in-between, chapter six, discusses the methods used and stress tests the solution to see how well generalized it is. An overview of this plan is given in Table 2 below.

Table 2: Plan of Approach Overview

Ch.	Title	Deliverable and/or Questions	Method
1	Introduction	Plan of approach, research goal, research questions.	Background → Problem → Goal → Approach
2	Literature	- According to the literature, how to identify bias in model statistics for linear and logistic regression? - According to the literature, how may Rabobank correct for bias in model statistics for linear and logistic regression models? This chapter, furthermore, is structured such that it places the problem and theory in context of Rabobank.	Literature review.
3	Synthesis	A method to identify bias in (selected) credit risk models for Rabobank. - What does a conceptual model to find bias in selected Rabobank credit risk models look like, and which techniques that are expected to reduce this bias should be added to this conceptual model?	Data-analysis from chapter 2.
4	Experiments	An experiment in e.g. Python that identifies bias in selected credit risk models. I.e. executing the chapter 3 plan. - Based on the conceptual model of chapter three, how are regressor parameters and model performance of credit risk models biased as a result of Rabobank model selection algorithms and how are these model statistics biased as a result of the hypothesized solutions?	Implement the model from chapter 3 in Python.
5	Analysis of Results	A written chapter on the outcome of the experiments in the context of improving credit management practices by reducing bias.	N.A.
6	Validation of solution	To address the shortcomings of the method used, and propose topics for future research that may solve these problems. Furthermore, we test the solution to see how generalized it is.	Simulation model and data analysis.
7	Conclusions & Recommendations	Based on the results, we formulate conclusions and recommendations that are expected to improve the credit risk management accuracy at Rabobank. An answer to the main question - *In what way, if at all, are the regressor-parameters of credit risk models at Rabobank biased, and how can Rabobank adjust for this bias both statistically and managerially?* is given.	N.A.

2. Theory

In this chapter we lay the foundation of this research by conducting a literature study. Different facets and parts of the problem of identifying and correcting for statistical bias are discussed. Furthermore, an attempt is made at structuring this chapter such that it gives the problem context by linking theoretical concepts to real-world practice at Rabobank.

We do this by first studying how different types of regression may be used in relevant credit-risk models (2.1). Based on the identified relationship between credit risk management and regression models, we study linear regression theory (2.2) and logistic regression theory (2.3). Lastly, the chapter is concluded by answering the research questions of chapter 2 given in Table 2.

2.1 Quantitative Credit Risk Management

The goal of this research is to identify, and correct for, bias in regressor parameters that may occur with model selection, and, based on this bias, give Rabobank advice on credit risk management improvements. The core of credit risk management are credit risk models. Such models, as we will see in coming sections, may depend on regression techniques. In this section we identify different credit risk models that may be relevant at Rabobank.

2.1.1 Credit Risk Models

Being creditworthy is defined by Merriam-Webster[2] as the extension of credit being justified by being financially sound enough. In other words, the creditor is financially healthy and is expected to service its debt with interest without problems. Credit risk models aim at estimating the creditworthiness of Rabobank's borrowers. There are different ways to do so.

In a book on credit risk by the Oesterreichische Nationalbank (OeNB) and the Financial Market Authority (FMA) (2004), models are divided into (i.) **heuristic models,** (ii.) **statistical models,** and (iii.) **causal models**. Heuristic models are based on practical experience, and often quite fuzzy due to their subjective nature. Statistical models aim at verifying hypotheses (e.g. a hypothesized relationship) on the basis of empirical data, using statistical techniques. Regression models fit in this realm. Lastly, causal models use financial theory, such as option pricing models, to establish creditworthiness. It must be noted that combinations of the different types exist, e.g. the Z-score developed by Altman (1968) uses statistical techniques and financial theory to develop a simple linear formula with different financial ratios as input to compute a number that is a measure of creditworthiness.

The OeNB & FMA (2004) discuss the use of logistic regression models in predicting binary outcomes of creditors being solvent enough to pay their obligations, which is referred to as a probability of default (PD) model. PD models are a subset of credit risk models, and are discussed in the next subsection. Other credit risk models are discussed in subsections thereafter.

2.1.1.1 PD Models

A rating of creditworthiness, according to the OeNB & FMA (2004), must be expressed as a default probability. PD models are designed to estimate the probability that a borrower will default on the loan, and there are different techniques present in the literature.

[2] https://www.merriam-webster.com/dictionary/creditworthy

Logistic Regression Models

Examples of logistic regression to estimate PDs are present in literature. Examples include predicting defaults of consumers by Costa e Silva, Lopes, Correia & Faria (2020), predicting defaults of Norwegian corporates based on microeconomic data by Westgaard & van der Wijst (2001), and using a version of Lasso (Tibshirani, 1996), which penalizes the presence of regressor parameters in a logistic regression model, suitable for logistic regression to find variables that are predictive of default probabilities by Blaskó (2019). Wang, Xu & Zhou (2015) show that the lasso for logistic regression also works when placed in a more advanced statistical algorithm that solves the problem of having unbalanced data (i.e. many negative outcomes and few positive).

Linear Regression Models

Linear regression techniques can be used in model selection by analysing how regressors explain an outcome variable. That is to say, linear regressor techniques can be used for establishing statistical inference. Although inference does not necessarily yield default probabilities, the variables with strong power, i.e. low p-values, may be selected to model PDs using other models. Hence, from an holistic point-of-view, linear regression models may be considered when studying credit risk management.

Machine Learning Models

Machine learning (ML) models such as random forest algorithms and neural networks can be useful for predictive modelling. When used for finding inference, by hard-coding an ML model we know how relevant regressors can get selected. These techniques, however, are generally more complex than linear and logistic regression. Identifying bias in regressor parameters, or bias in any aspect of the outcome function, for ML models is therefore left out of scope: we lay a foundation by analysing linear and logistic regression models in this research because of their relatively simple structure, which may later be built upon further by analysing bias in more complex ML models.

Causal Models

Default probabilities for corporates may also be retrieved by means of causal models, examples of which are given in Hull (2018, pp. 431-455). One method Hull describes is based on hazard rates, the probability of default within a relatively short period of time, which are determined from historical cumulative default probabilities, published by e.g. Moody's (Figure 3).

Time (years)	1	2	3	4	5	7	10	15	20
Aaa	0.000	0.011	0.011	0.031	0.085	0.195	0.386	0.705	0.824
Aa	0.021	0.060	0.110	0.192	0.298	0.525	0.778	1.336	2.151
A	0.055	0.165	0.345	0.536	0.766	1.297	2.224	3.876	5.793
Baa	0.177	0.461	0.804	1.216	1.628	2.472	3.925	7.006	10.236
Ba	0.945	2.583	4.492	6.518	8.392	11.667	16.283	23.576	29.733
B	3.573	8.436	13.377	17.828	21.908	28.857	36.177	43.658	48.644
Caa-C	10.624	18.670	25.443	30.974	35.543	42.132	50.258	53.377	53.930

Source: Moody's.

Figure 3: An example of Moody's Average Cumulative Issuer-Weighted Default Rates. Taken from Hull (2018, p.434).

Hazard rates may also be calculated from credit spreads (e.g. credit default swaps spreads, bond yield spreads, or asset-swap spreads). Credit spreads are excess returns on a corporate bond as compared to a similar risk-free bond (Hull, 2018, p.442), and example of which could be to subtract the premium paid on a credit default swap from a corporate bond because such a structure would be approximately risk-free. Note that the recovery rate (RR) of the corporate bond must be known

or estimated for such methods to be theoretically sound, as an estimate for the hazard rate $\bar{\lambda}$, given a credit spread $s(T)$ where T is the maturity (end-date) of the bond, is defined as $\bar{\lambda} = \frac{s(T)}{1-RR}$.

One option to estimate RRs, because they are a percentage value of the face-value of the bond, could be (logistic) regression models that are based on historical RRs. If one were to estimate RRs using regression techniques then a model needs to be selected and thus the RRs may be subject to bias because the regressor parameters may be subject to bias.

Lastly, PDs can theoretically be estimated by modelling a company's equity as an option on the assets of the company (Merton, 1974); doing this requires estimates of the value of equity and assets at different times, as well as the volatility of these values. And thus, regression *may* indirectly affect the outcome of this model if regression models are used in estimating these values and their volatilities.

2.1.1.2 From EAD and LGD to Expect Losses

In managing a portfolio of loans it may be useful to know the Exposure At Default (EAD). The exposure at default is typically expressed as an amount of money (GARP, Apostolik & Donohue, 2015), and represents the exposure of the lender when the counterparty defaults. An example of EAD modelling is k-factor modelling, where the exposure at the time of default may be estimated by e.g. using regression techniques.

Loss Given Default (LGD) may be expressed as a percentage and can be modelled as $1 - RR$. As discussed in section 2.1.1.1, the RR is the expected fraction of the face value of e.g. a bond that the lender can still make a claim on and receive. So, the LGD is the fractional expected losses incurred conditional on a default. Ultimately, in portfolio management, the expected losses (EL) may be calculated based on the LGD (and thus the RR), the EAD, and the PD as follows: $EL = PD * LGD * EAD$. We established the relationship between PD, EAD, and LGD models and what they can be ultimately used for: calculating expected losses.

It may be interesting to see how RRs can be modelled by means of regression techniques, but these are out of the scope of this research and hence will not be further researched. To maintain a manageable research scope, EAD models will not be further researched too.

2.1.2 Applications of General Regression Models in Credit Risk Management

In section 2.1 we studied different quantitative credit risk management models. For this research the relevance lies in how these models may dependent on regression techniques. In the domain of PD models these dependencies are mostly present. We identified that logistic regression models are commonly used in estimating PDs. Furthermore, linear regression models may be used to draw inferences on factors that influence PDs. Causal models such as predicting PDs from credit spreads or equity prices may indirectly be affected by said regression techniques when assumptions are avoided, and statistical models to estimate e.g. recovery rates, asset-values or volatilities are preferred.

Concluding, statistical regression models can, and are, used in (quantitative) credit risk management to make predictions and draw inferences; thus we study these models, how they work, and how biases arise in the next sections.

2.2 Linear Regression

Within Rabobank linear regression techniques can be used for determining regressors in PD models. In this subsection we first explain how linear regression techniques can be used for model selection. Then, we discuss the model selection algorithms used by Rabobank. Based on this we discuss how

bias in regressor parameters may occur and be recognized. Lastly, we formulate potential solutions to this phenomenon found in literature.

Within Rabobank linear regression techniques can be used for determining regressors that may be used in PD models. That is, from a long list of potential regressors a few can be selected that seem to be related to PD rates. In this section we discuss model selection algorithms suitable in combination with linear regression (section 2.2.1), cross validation for model fitting (section 2.2.2), how bias in linear regression model statistics can be recognized (section 2.2.3), and what potential solutions may be in resolving this bias (section 2.2.4).

2.2.1 Linear Regression Techniques for Model Selection

Recall, from section 1.1.1, the general form of a multiple linear regression model: $y = \beta_0 + \sum_i \beta_i x_i + \epsilon$. Researchers may formulate a question and then develop a linear model to test for inference. This, in the context of predicting defaults, could have been, e.g., to study the relationship between PD and educational attainment. The researcher may then set the null-hypothesis H_0: $\beta = 0$ and compute p-values, based upon which H_0 can be rejected with a particular significance α of, say, 5%. This is one method of model selection, where the relationship of one regressor to PD is tested, after which it the regressor may be used in an actual PD model.

Storing data and measurements have become easier, and statistical algorithms are used to find hypothesized relationships for us. That is, statistical algorithms select the models for us. This common practice is named model selection, which we will now discuss.

2.2.2 Model Selection Algorithms

Commonly, model selection algorithms iteratively generate and compare models. Comparisons are based on model performance statistics such as, as discussed, the MSE. Different performance statistics for model comparisons exist, as well as different model selection algorithms.

2.2.2.1 Linear Model Performance Statistics

Mean Squared Error & Mean Absolute Error

We start with a performance statistics that we discussed earlier: the mean squared error (MSE) of a linear model. When a linear model is established it can be used to make predictions. When we think of the error in MSE as the difference between the predicted value of the outcome variable y and its observed actual value, the MSE becomes clear from its formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predicted} - y_i^{true}\right)^2.$$

It can be seen from the formula above that the error for each observation is squared and averaged. Another similar idea is not to square the error but to take the absolute value of it. This is referred to as the mean absolute error (MAE) and can be calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^{predicted} - y_i^{true}\right|.$$

Akaike Information Criterion & Bayesian Information Criterion

MSE and MAE do not account for overfitting of the model, that is to say in short that, given some data sample, the MSE and MAE are likely to decrease in value if more regressors are added to the model. Then, when such an overfitted model is presented new data the predictions are highly inaccurate because the model is fitted too heavily on observed data.

Penalizing the number of regressors in a model while rewarding how well a model predicts the observed value of the outcome variable may be preferred. The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) can do that. The AIC and BIC are computed by

$$AIC = 2k - 2\ln(\hat{\mathcal{L}})$$

and

$$BIC = k\ln(n) - 2\ln(\hat{\mathcal{L}})$$

respectively, where k is the number of regressors in a model, n is the total number of data points that are used for developing a model and $\ln(\hat{\mathcal{L}})$ is the natural logarithm of the maximum likelihood, which is a measure of how well the model is fitted to the n data points. In the next section (2.3) we elaborate on likelihood functions in more detail.

An illustration of the idea that the MSE can keep decreasing when adding more variables to a model, and that, on the contrary, this is not the case for the AIC metric is given in Figure 4.

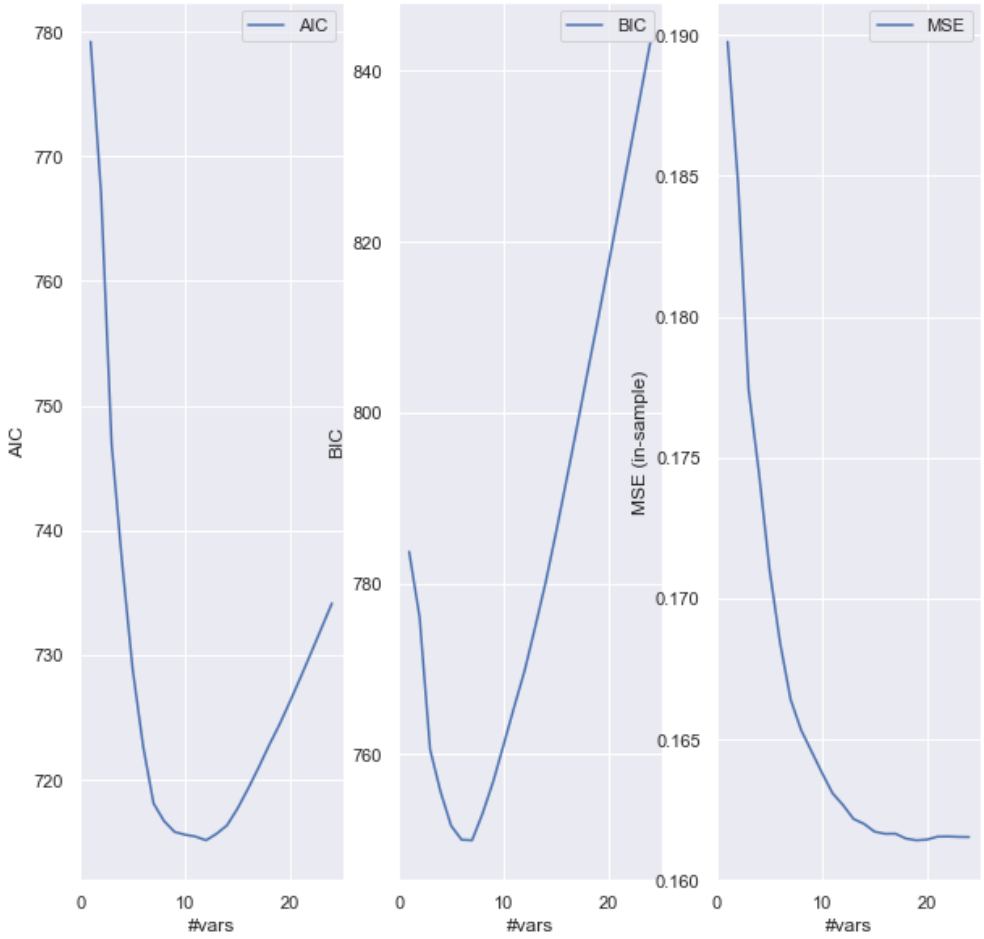

Figure 4: An example of a backwards stepwise model selection procedure. Every model is evaluated on its AIC, which penalizes the number of regressors in a model, and its test MSE, which does not penalize the number of regressors. At some point, the AIC and BIC increase while the MSE appears not to do so in the same significant way.

Note that AIC and BIC do not tell us explicitly, like MSE or MAE do, how well the model can be used to estimate the outcome variable y. Hence, AIC and BIC are model performance statistics that indicate how well models perform relative to other models.

F-test p-values

An F-test (see e.g. Brockhoff, 2013) p-value is a relative performance statistics that compares a simpler model M_1 with a more complicated model M_2 that has one additional regressor in it. A null hypothesis can then be defined: the regressor parameter of the added regressor in the new model equals zero. With n data points the test statistic F based on which to reject or accept the null hypothesis is

$$F = \frac{\frac{SSR_1 - SSR_2}{k_1 - k_2}}{\frac{SSR_2}{n - k_2}}$$

(Brockhoff, 2013) where SSR is the sum of squared residuals, also referred to as the sum of squared errors and k_i the number of regressors in model $i,\ i = 1,2$. Test statistic F is compared to a critical value with significance α and degrees of freedom $df = (k_2 - k_1, n - k_2)$. Rejecting H_0 implies model 2 is better than model 1. Test statistic F can also be converted to a p-value that can be interpreted as the probability that the new more complex model is not preferred over the old simpler model.

2.2.2.2 Stepwise Algorithms

Iteratively adding or removing a regressor from a model is the idea of stepwise model selection. At Rabobank, different stepwise algorithms can be used.

Forward Stepwise

With forward stepwise, we start with a model without regressors in it. Then, if no regressors are forced in the model (i.e. they must be in the model and are therefore selected unconditionally), regressors are iteratively added to the model by adding the regressor that contributes most to the performance statistic that is set as the model-selection criterion until no improvements are made.

Backward Stepwise

With backward stepwise the first model is a model that contains all available regressors. The regressor that improves the selection criterion (i.e. a performance statistic such as BIC) is removed. Without constraints on the minimum number of regressors in a model, regressors being forced in a model, or strict F-test p-value acceptance thresholds, the algorithm will continue until the defined performance metric cannot statistically improve.

Bidirectional Stepwise

With bidirectional stepwise we start with a model that contains one regressor, add another regressor that improves a set performance statistic the most, and subsequently test if removing any of the current two regressor in the model improves the set performance statistic. This continues until no improvements can be made, or terminating constraints such as maximum number of regressors in a model are reached.

2.2.2.3 Subset Algorithms

Suppose we have a data set with outcome variable y and a total of N regressors. Then, when choosing k regressors, the total number of combinations of regressors in a model is $\binom{N}{k}$. With all-subsets model selection, the total number of different models would be $\sum_{k=0}^{N}\binom{N}{k}$. All models are tested on a set performance statistic and the best one is selected. This practice may be inefficient because with, say, $N = 15$ the total number of models, if we allow $k = 0$, is 32768. Hence, a case can be made for restricting the total number of regressors in a model, this is referred to as modified subset model selection by Rabobank.

2.2.3 Cross Validation for Model Selection

Simple k-fold cross validation (CV) is a technique that splits the data up in k equal size parts, referred to as folds, such that model parameters $\boldsymbol{\beta}$ can be found using $k-1$ folds and the final model can be evaluated on the 1 fold that was left out. When we, after defining k folds, iteratively use a different fold for evaluation and use the remaining folds for parameter estimation, we have generated k different models. Different types of k-fold CV exist, three are discussed below.

2.2.3.1 Simple K-Fold Cross Validation

With k-fold CV, data is partitioned in k different parts. The procedure is to use a subset of the k folds for model selection and parameter estimation, and the remaining folds for assessing the performance, called validation, of this model. The result of this procedure can be k different models with k different performance measures. The model with the best (average) performance may be selected to be put into practice.

2.2.3.2 K-Fold Cross Validation and a Test Set

With simple k-fold CV we can be biased when selecting a model because the selected model can perform well just by chance alone. Introducing a test set of data before splitting the remainder into k folds, with which simple k-fold CV is performed, may reduce the probability of selecting a model that performs well just by chance. That is, we perform k-fold CV to obtain k different fitted models. The model that has the average best performance on the validation data set is selected. The performance of the selected model is evaluated using a test data set, and this performance is what we expect when the model is deployed.

2.2.3.3 Nested K-Fold Cross Validation

The characteristics of the test data set may influence the selected model performance due to chance alone. We can split the data into a train and a test set, using outer folds. We leave one outer fold out, and perform k-fold CV on the other sets. A model is selected and its expected performance is estimated using the test set. We loop over the outer folds, using a different fold to test the selected model from the simple k-fold CV. The result are a number (equal to the number of outer folds) of models and performance estimates.

2.2.3.4 Concluding CV

Performance of models can be estimated using k-fold CV. Robust analyses consider the variability of the data across these folds, an example of which is making use of a test set that is used to estimate the performance that is to be expected when the model is used in practice. The variability of this test set can be taken into consideration by using nested k-fold CV that loops over the outer folds and uses one as a test set each time. A schematic overview is given in Figure 5. In the next section we discuss how different types of CV can result in biased model performance.

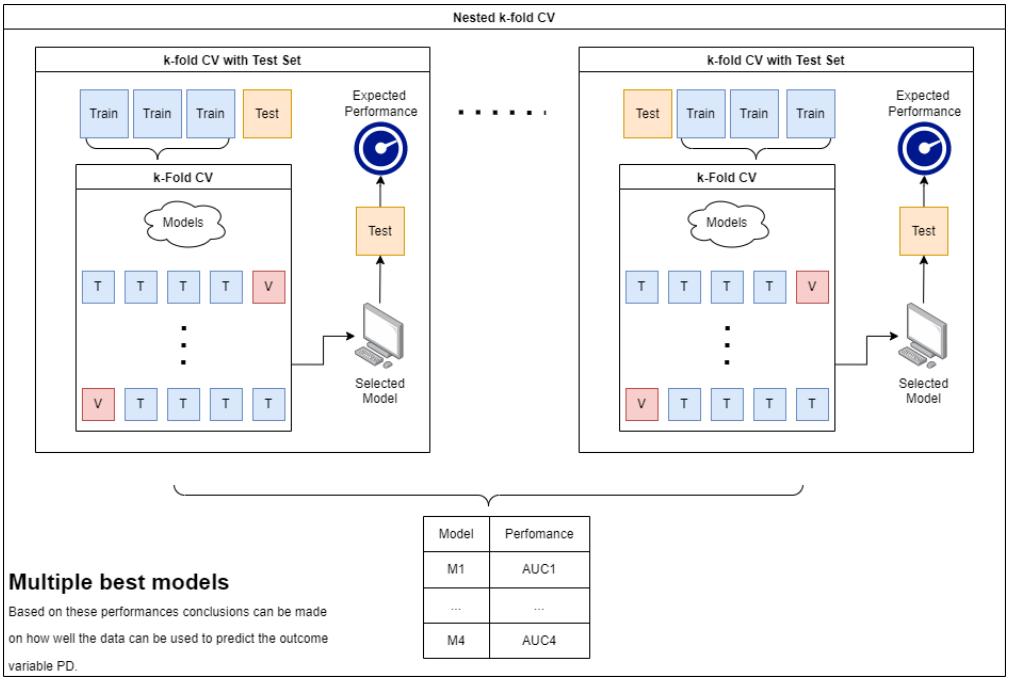

Figure 5: Different types of CV (after Feldman, 2019). In k-fold CV, the letter T and V stand for training and validation respectively. The 'Models' cloud is a (set of) model(s) after model selection, that do not have fitted parameters yet.

2.2.4 Recognizing Biased Parameters in Linear Models

Model selection algorithms are used by Rabobank to find promising models. While such algorithms can help, the power of the regressors is questionable because the algorithms may over-exploit the data by fitting many models, and select a promising one that can predict noise (Loftus, 2019). Hence, valid inference after model-selection, which is the main topic of this research, can be a problem for Rabobank. In the following sub-sections we discuss techniques that can be used to establish whether or not the credit risk models at Rabobank are subject to biased parameters.

2.2.4.1 Regressor Parameters Sample Distributions

The result of experiment from section 1.1.4 is a step in the direction of visualizing bias in regressor parameters that is the result of a model selection algorithm. We can control the way regressor data is generated, as well as the true value of the regressor parameters that are used to calculate an outcome variable. Furthermore, noise can be controlled.

Samples are drawn, model selection is performed on each sample, and parameters are found using that same sample. By measuring the mean and standard deviation of the sample distribution of a regressor parameter and/or its respective t-value, bias can be recognized. Measuring the number of instances in which a regressor is selected, conditional on its true β, the severity of the bias can further be commented on. That is, if, for a regressor, its parameter $\beta = 0$, and a stepwise algorithm selects this regressor many times out of the number of samples, there may be a problem.

2.2.4.2 Model Uncertainty in Stepwise Algorithms

Stepwise algorithms generally yield a final model that cannot be improved upon. We are, however, uncertain about this model being the optimal one because after every iteration there is a probability that the new model is not better than the previous model.

For the sake of practicality, let us assume that the F-test p-value (section 2.2.2.1) is a useful estimate of, after an iteration, the probability that the previous model is better. Then, a nested for loop can

be constructed where model selection is performed on a sample with the result being a list of all iterations made including the respective F-test p-values. The outer for loop samples data, the inner for loop samples from the iterations-list as shown in the example Table 3.

Table 3: Example of selecting a model from a bidirectional stepwise model selection list.

Iteration	Model	Direction	F-test p-value	P(selected)
0	{const}	Forward	-	0,03
1	{ const, x1}	Forward	0,03	0,001
2	{ const, x1, x6}	Forward	0,001	0,004
3	{ const, x1, x6, x7}	Forward	0,004	0,05
4	{ const, x1, x6, x7, x2}	Forward	0,05	0,07
5	{ const, x6, x7, x2}	Backward	0,07	0,845
		SUM	0,155	1

A number of draws can be made from a table similar to Table 3, after which the selected model is fitted and its statistics such as β and t-value are stored. These statistics can be compared to the case of always selecting the final model: the more uncertain we are in accepting a new model at an iteration, the worse the bias is likely to be in terms of strangely shaped sampling distributions.

2.2.4.3 Biased Performance Statistics after Cross Validation

Instead of searching for bias in regressor parameters, which may be considered parts of models, we can also model potential bias of the complete PD model. Moshontz, Fronk, Sant'Ana & Curtin (2020) identify optimization bias of cross-validation bias in a simulation-based way. Optimization bias is defined as model performance statistics being overly positive. Here, different types of cross-validation (CV) are seen as the model selection algorithm.

Moshontz, Fronk, Sant'Ana & Curtin (2020) developed a methodology for quantifying optimization bias, which may aid in deciding which form of k-fold CV to use, simple, with a test data set, or nested. The methodology is as follows: first, regressor data $\boldsymbol{X}$ is randomly generated from a multivariate normal distribution with zero covariance between regressors. Secondly, the outcome variable can be binomially generated with a certain probability of being positive. Third, a form of k-fold CV can be applied to fit and validate models, for which eventually one is selected based on its performance.

Performance is defined as the area under the receiver operating characteristic curve (AUC), which is a value between 0 and 1 that is an estimate of how well a model can distinguish between binary outcomes as the AUC is an estimate of the probability of a model predicting a higher score for a randomly selected positive case than a randomly selected negative case (Fawcett, 2006; Moshontz, Fronk, Sant'Ana & Curtin, 2020).

A number of simulations are performed by Moshontz, Fronk, Sant'Ana & Curtin (2020) using different settings regarding the sample size of the data used in the CV algorithms, the dichotomy of the outcomes of the variable to be predicted y, the number of potential regressors in a model, the value of k in the k-fold CV algorithms, and the type of k-fold CV used for selecting a final model for which the performance (AUC) is reported. Bias is reported to be quantified by observing how different the AUC is from the true value of 0.5 (recall, all data including the y variables are randomly generated from a known model). The study does not show these quantitative results but suggests

that (i.) K-fold CV gives overly optimistic model evaluation, (ii.) smaller sample sizes of the k folds increase optimization bias, and (iii.) unbalanced, i.e. un-dichotomous, outcomes increase optimization bias.

2.2.4.4 Practical Considerations in Searching for Bias

We searched the literature and found that model selection algorithms may influence the t-values of regressor parameters and k-fold cross validation techniques may introduce optimization bias in terms of model performance statistics. Monte-Carlo style simulation techniques can be helpful in giving quantified estimates of these biases.

The characteristics of biased model statistics may depend on (i.) the model selection algorithm directly, (ii.) the in- and exclusion of other regressors in the model, (iii.) the dispersion of the regression parameter itself, (iv.) the sample size, (v.) the size of the folds in k-fold CV methods, and (vi.) unbalanced outcomes in the data-set. Hence, further practical implications may be to assess different combinations of model selection algorithms, types of CV, number of folds k in k-fold CV, and balance and size of the data-set.

Lastly, we must consider in what way the bias should be presented to Rabobank: regressor parameters β, t-values, or model performance statistics such as the AUC. These considerations are elaborated upon in the next chapter.

2.2.5 Statistical Solutions to Biased Parameters in Linear Models

Additionally to identification, we should know about potential solutions such that we can add these to the technical solution, because it may be that the solution lies in a different or modified model selection algorithm, and by finding the bias of this technique it can be compared to the techniques used by Rabobank.

2.2.5.1 Lasso and Adjusting p-values using the Polyhedral Lemma

Tibshirani (2018) discusses the of inference after model selection, and mentions false power may occur with the Lasso model-selection algorithm. Suppose we have some data $(\boldsymbol{x}^j, y_i), j = 1,2, \dots, p$ where $\boldsymbol{x}^j = (x_{j1}, \dots, x_{jn})^T$ and we introduce a constraint on the sum of the absolute values of $\boldsymbol{\beta}$ in terms of a maximum value s. Then the Lasso algorithm has the following objective:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \; subject\ to \sum_{j=1}^{p} |\beta_j| \leq s.$$

Equivalently, when we define a penalty parameter λ that penalizes $\sum_{j=1}^{p} |\beta_j|$ the Lasso algorithm has the following objective:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

By penalizing $\sum_{j=1}^{p} |\beta_j|$ the final model is expected to become sparser when increasing λ. The important part is that, according to Tibshirani (2018), a mathematical property[3], exists such that p-values of regressors after model selection can be adjusted such that they are conditional on the

[3] This mathematical property is called the polyhedral lemma by Tibshirani (2018), and is not elaborated upon in this research because this lemma cannot be used in the case of logistic regression (Tibshirani, 2018).

model-selection algorithm. Doing so is based on defining a range for which lasso always selects the same model. Therefore we may consider modelling the bias of parameters in models selected by lasso after adjustment using this analytical approach, and compare this bias to that of other model selection algorithms.

2.2.5.2 Data Splitting, Data Carving, and Adding Noise

Controlling the type-I error is not the only thing we should be interested in because by conditioning the p-values on the model selection algorithm the confidence interval of the regressor parameters (the range the true value of a parameter is in with a certain probability) may become larger, thereby possibly making inference uncertain instead of biased. A common approach is data splitting where one sample is used to find a model and another sample to find $\boldsymbol{\beta}$. But here we lose half of the data, and thus we may lose power: the confidence intervals for the parameters become wider.

More sophisticated methods for drawing inference, discussed by Tibshirani (2018), based on when and how to use the data, exist. The first example is data carving, which involves removing a small portion of the data in model selection and then using all for tuning parameters. The second example is to add noise to the data by means of, e.g., adding a small random number to every observation, in the model selection stage, and then removing this noise when tuning parameters. Tibshirani (2018) shows us that data splitting is giving far worse results than the other two examples, in terms of the final model being able to distinguish between positive and negative outcomes.

2.2.5.3 Potential Solutions to Biased Model Parameters

Concluding, when modelling bias, in terms of regressor parameters and model performance, of techniques that are expected to reduce said bias, the following should be considered:

- Model selection could be done with lasso, which penalizes the sum of absolute values of all β, after which confidence intervals for the final $\boldsymbol{\beta}$ can be adjusted analytically. However, implementing this is out of scope for this research due it not being applicable to logistic regression models. Lasso may be considered as a model selection method just because it may select the same model consistently across data samples.
- When model selection is performed, data carving and adding noise to the data should be considered because these techniques yield unbiased estimates of confidence intervals for $\boldsymbol{\beta}$. Data splitting may be experimented with as well.

2.3 Logistic Regression

Rabobank uses logistic regression models to predict if a borrower is a $good$ or a bad client. This is determined based on the PD of the (potential) borrower, where an applicant is bad is the PD is above a certain threshold. Regressors that seem to say something about going into default based on linear regression techniques are subsequently placed in a logistic regression model selection procedure. The final logistic regression model estimates the PDs.

Logistic regression models are used to predict binary outcomes based on a probability of an observation being positive. In the case of Rabobank that means that a client applies for a loan and the model, based on characteristics of the client, estimates a probability of default. How a logistic regressions is built is discussed in the next sub-section.

2.3.1 The Mechanism of Logistic Regression

A logistic regressions model looks like an s-shaped curve that flows from zero to one on the y-axis. Its use is to estimate the probability that an observation belongs to the positive outcome, and is therefore used in classification problems. Predicting defaults can be a classification problem.

When p is defined as, in this case, a default probability, β_0 as the y-axis intercept, $\boldsymbol{\beta}$ as the regressor parameters and $\boldsymbol{x}$ as the regressors $j = 0, \dots, n$. Then the general form of a logistic regression model is

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{j=1}^{n} \beta_j x_j \,.$$

Notice the right-hand side is similar to that of linear regression but the left-hand side is a transformation of the PD, denoted with p; which we will discuss now.

2.3.1.1 Transformation

The first step in constructing a simple logistic regression model, conceptually, is to place the (historical) observations on a probability-graph. This graph has a y-axis with values between zero and one, and all observations are discrete zero or one. Then a **logit transformation** is performed on the probability-values using $y := \ln(\frac{p}{1-p})$ to get the log-odds. These log-odds can be graphed, and because all the known outcomes are either zero or one the new y-values will be either ∞ or $-\infty$ after the logit transformation. On this log-odds graph an initial linear model is defined, and the ∞

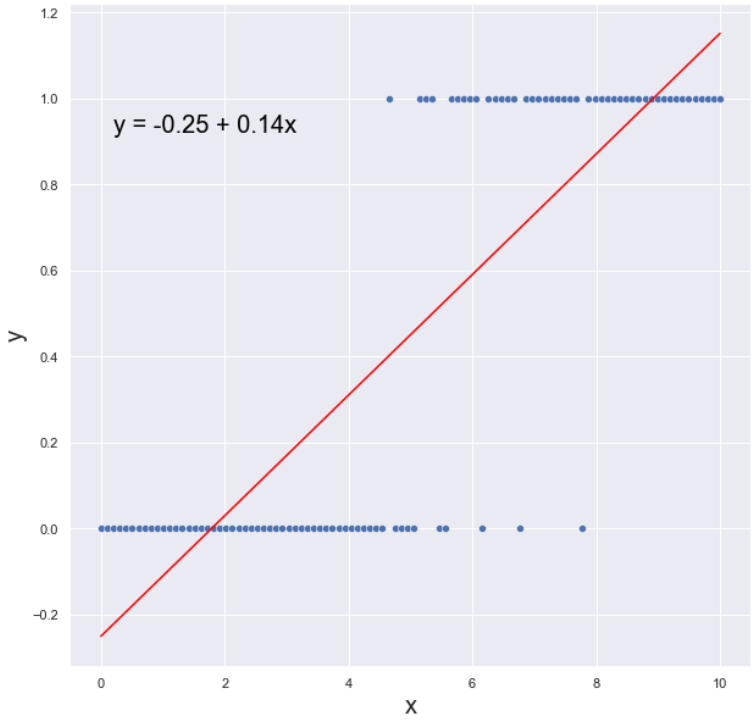

Figure 6: An example of fitting a linear model (the red line) to binary observations (the blue dots) in a probability graph. The corresponding log-odds graph has the same shape, but the blue dots ara in the limit of ∞ or $-\infty$. After StatQuest (2018).

and $-\infty$ are mapped onto this linear function. The goodness of this linear model cannot be expressed in e.g. MSE of MAE since all observations are now either ∞ or $-\infty$. So, these candidate log-odds are transformed back to an s-shape curve resembling the candidate probabilities as follows: $p = \frac{1}{1+e^{-y}}$.

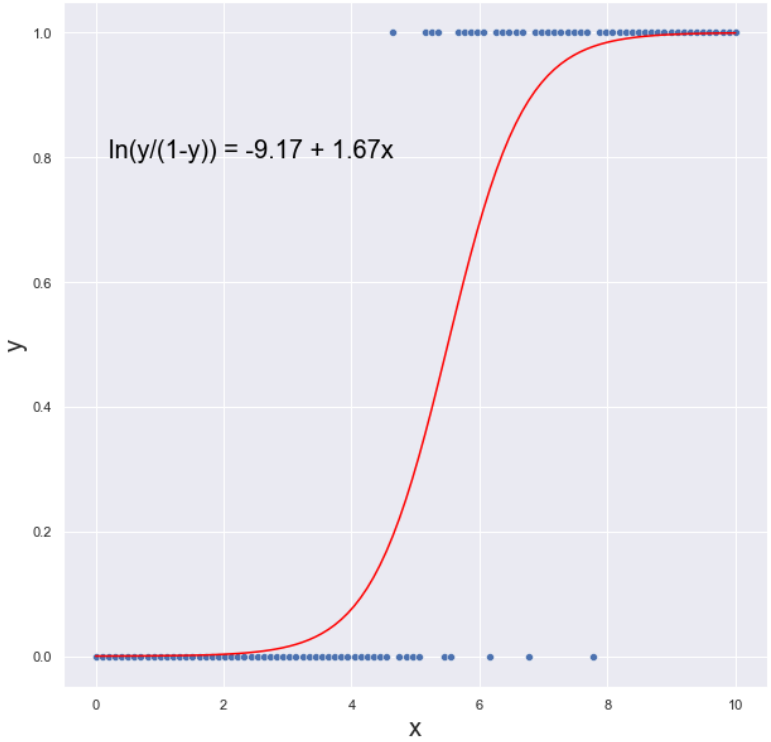

Figure 7: An example of a logistic regression model. After mapping the observations on a linear model, the scores are transformed back to probabilities and the likelihood of the model can be computed. After Statquest (2018).

A good model is an s-curve that can predict defaults well and has a relatively high likelihood, discussed in the next section.

2.3.1.2 Likelihood

The s-shaped curve of a logistic regression model gives a probability that a borrower will default: the likelihood $y_i$that a bad borrower will default, and the likelihood that a $good$ borrower will **not** default is $1 - y_i$, where i is an observed borrower. Given the model (the s-shaped curve) and the model outcomes $\hat{y}_i$ for each observation i, $i = 1, \dots, n$, we can calculate the **log-likelihood** ($\mathcal{L}$) as follows:

$$\mathcal{L} = \sum_i y_i(\ln \hat{y}_i) + (1 - y_i)\ln(1 - \hat{y}_i).$$

For a mathematical derivation see e.g. Faraway (2005, pp.224-228).The log-likelihood is iteratively calculated and compared, the s-shaped curve with the maximum log-likelihood is selected as the best model. Selecting the 'best model' in this case is adjusting $\boldsymbol{\beta}$ until the log-likelihood cannot increase, but from previous sections we know that, in the case of linear regression at least, this vector is dependent on the regressors already present in the regression model, which may be determined by model selection algorithms. Model-selection algorithms that may be used for logistic regression are therefore discussed in the next section.

2.3.1.3 Model-Selection in Logistic Regression

Recall the general form of a logistic regression model: $\ln\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{j=1}^{n} \beta_j x_j$. Here, $\boldsymbol{\beta}$ is found such that the log-likelihood, as discussed in the previous sub section, is maximized. Intuitively, the s-shaped curve is not the direct result of a particular model but rather of a transformation of the log odds of the PD values p. Therefore, the same model selection algorithms as discussed in the linear

regression section may be applied. These are: forward- and backward stepwise, a combination of forward and backward stepwise, best-subset selection, and a modified subset where constraints are added in order to reduce the maximum number of models tested. Furthermore, we may use lasso as a model selection algorithm.

2.3.2 Recognizing Biased Parameters in Logistic Models

The simulation-based methods to find bias, discussed in the previous section on linear regression, are expected to be useful for logistic regression as well, because the engine behind logistic regression is, in the first place, a linear regression. The only aspect of logistic regression models to keep in mind is that the regressor parameters are in terms of the log odds of the PDs. CV methods discussed in the section on linear regression can be used for performance evaluation of logistic regression as well. Concluding, no additional difficulties in recognizing bias in regressor parameters or performance statistics for logistic regression models are expected.

2.3.3 Statistical Solutions to Biased Parameters in Logistic Models

Recall the polyhedral lemma (section 2.2.5.1) where confidence intervals for the true value, if it exists, of $\boldsymbol{\beta}$ can be analytically constructed if model selection is done based on the lasso algorithm. This does not work for the logistic regression case of adjusting for bias (Tibshirani, 2018). Lasso regularization for logistic regression models differs from the linear regression case in having the following objective function[4]:

$$\min_{\boldsymbol{\beta}} -\mathcal{L}(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|,$$

where $-\mathcal{L}(\boldsymbol{\beta})$ is the negative log likelihood of the model, λ is a penalty constant, $\boldsymbol{\beta}$ is the vector of regressor parameters for all p regressors. This does not mean lasso, or any other model selection algorithm, should be excluded from the conceptual model in the next chapter because, similar to the linear regression case, the model selection algorithm may select the same model consistently.

Data carving, adding noise, and possibly data splitting (discussed in section 2.2.5.2), may be solutions to bias in regressor parameters in logistic regression models as these methods are not based on the mechanisms of regression directly; i.e. in these cases we seek solutions in the data, not in the models.

2.4 Concluding Chapter 2

In this chapter attempts were made at understanding credit risk models and how they may depend on linear or logistic regression techniques. This link was studied because Rabobank uses these regression techniques to ultimately estimate PDs. Depending on the nature of the model, heuristic, statistical or causal, credit risk models may depend heavily on these regression techniques.

The results of a literature review on identifying and correcting for biased regressor parameters show that, for both linear and logistic regression techniques, biased model statistics may arise with model selection algorithms and versions of k-fold CV. We now answer the following research question:

- According to the literature, how to identify bias in model statistics for linear and logistic regression?

[4] Since the statsmodels package is used in this research, the objective function of $logit.fit\_regularized()$ applies. Maximizing the log-likelihood is similar to minimizing the negative log-likelihood.

No matter if we consider linear or logistic regression, identifying biased regressor parameters as a result of model selection algorithms can be done by assessing the shape of the distribution of β or the respective t-values and observing how these distributions deviate from a normal bell-shaped curve. Furthermore, data can be controlled and model selection algorithms can be used to estimate known parameter values, after which comparisons are possible.

Optimization bias, which is an artificially inflated model performance, can be identified, in the case of logistic regression, by performing k-fold CV model building and performance evaluation techniques on data for which the model responsible is known such that the true $AUC = 0.5$. The response variable y should be binomially distributed.

- According to the literature, how may researchers correct for bias in model statistics for linear – and logistic regression models?

When using the lasso algorithm to select models, p-values based on which to reject null hypotheses can be analytically adjusted using a mathematical property (the polyhedral lemma). This analytical property, however, is discarded because it is not applicable to logistic regression. The lasso algorithm itself does not have to be discarded because it may select the same model consistently across data samples.

There are no model selection algorithms that should be discarded, because correcting for bias may also be done by utilizing models that seem to perform well in terms of generating bell-shaped t-value sampling distributions that are similar to the sampling distributions of regressor parameters when the correct model is known. Rabobank may then effectively reduce bias by adopting certain model selection techniques.

Solutions can also be sought for outside of model selection algorithms. Data splitting, data carving and adding noise to the data, in combination with a model selection algorithm, are techniques that may be applied in order to get valid inference after model selection without greatly increasing confidence interval widths of $\boldsymbol{\beta}$ for carving and adding noise.

Outcomes always depend on the data sample. Therefore, the sample should be used such that it resembles the (unknown) population as much as possible when building credit risk models. K-fold CV techniques can be used for this, but exploiting the data in this way is not without the theoretical cost of optimization bias. The severity of this bias may depend on the dichotomy of the outcomes of the variable to be predicted y and the number and size of folds k. The above mentioned techniques and aspects may be considered when building a technical solution to bias, which is the topic of the next chapter.

3. Methodology to Quantify Bias

In this chapter we answer the research question *'what does a conceptual model to find bias in selected Rabobank credit risk models look like, and which techniques that are expected to reduce this bias should be added to this conceptual model?'* by using the theory gathered in chapter 2, and apply this theory to Rabobank specifically.

3.1 Quantifying Bias

In chapter 2 we identified two different types of potentially problematic bias: biased inference after model selection and biased performance statistics as a result of k-fold cross validation and, possibly, model selection. Hence, we split the bias up in two parts: biased inference and biased model performance.

3.1.1 Biased Inference

Recall that statistical inference can be referred to as the science of using data to verify hypotheses. These hypotheses describe a relationship between, in this case, probability of default and any regressor we may want to analyse. We have seen that when model selection algorithms are used to define hypotheses for us, subsequent inference can be problematic. That is, for example, one model selection algorithm may be given a data sample and the result can be a different model every time. And since the model one regressor is placed in affects the power of that regressor, drawing inference may be problematic.

Furthermore, we may make use of multiple and different model selection algorithms (e.g. forward, backward, and bidirectional stepwise, or the lasso algorithm), and all these algorithms can have different parameters regarding the maximum or minimum number of variables, the p-value threshold of adding or removing a regressor to or from a model, etcetera.

The above greatly increases the number of possible models that can be outcomes to a model selection procedure, which in turn can make drawing inference problematic because the power of regressors can be conditional on the model these are present in.

3.1.1.1 A Method for Quantifying Biased Inference

To quantify biased inference we propose, based on several insights of chapter 2, the following methodology:

1. Sample data $\boldsymbol{X}$ that is representative for Rabobank, i.e. the data looks like data available at Rabobank for building a PD model. This may be uniformly distributed data.
2. Transform $\boldsymbol{X}$ such that each regressor has the same order: a value between zero and one. In the case of categorical regressors this can be done by means of, e.g., one-hot encoding, and in the case of continuous regressors this transformation can be e.g. a logit transformation with empirically established transformation function parameters. Or, $\boldsymbol{X}$ is already normalized by means of using a suitable, e.g. uniform, random number generator.
3. Generate a PD vector y using known regressor parameters $\boldsymbol{\beta}$ combined with $\boldsymbol{X}$.
4. Perform a (set of) simplified full model selection pipeline that may be regarded to as the current practice at Rabobank, and store regressor parameters β and t-values of the regressors in the selected model.
5. Perform a (set of) proposed solutions from Chapter 2 (i.e. Lasso, data carving, adding noise, and data splitting), and store regressor parameters β and t-values of the regressors in the selected model.

6. Compare the results of steps 3 and 4 to the known true values of $\boldsymbol{\beta}$, and compare the empirical acceptance probabilities (from t-values) of each regressor between the methods performed in step 3 and 4.
7. The methods where the results deviate the least from the known model parameters may be preferred.

The above steps will be further elaborated on in the subsections below.

3.1.1.2 Data with a Known Model

The reason for defining a known model to generate defaults in credit default data is twofold: firstly, it is necessary because with natural credit default data we do not know the true model responsible for the observed defaults, and, secondly, it allows for publication of this thesis without compromising the obligation of Rabobank to keep any client data confidential.

To later test if the proposed solution works on real data as well, in this thesis, we make use of the publicly available **German Credit Risk**[5] data set. We normalize this data set using techniques described in section 3.1.1.3 and remove its default column. This is expected to yield a dataset that is indistinguishable from real, yet its usefulness is now zero as the structure between regressors is removed: i.e. the relationships between regressors and the binary default outcome variable is removed. By using known regressor parameters $\boldsymbol{\beta}$ to, via a logit transformation, generate a binary outcome vector y, we define our own structure that we can control, and compare to the structures found by the different model selection algorithms.

3.1.1.3 Logistic and Linear Transformations

Transformations for continuous regressors are useful in regression analyses because they transform the values of each regressor to a standard range, e.g. $[0, 1]$, which, generally, allows for fair comparison across regressors. That is, regressors are not given relatively more power simply because of the order of their values. An observation, or natural value, x_i is transformed, for $i = 1, \dots, n$, with the **logistic transformation** function as follows:

$$T_{logistic}(x_i) = \frac{1}{1 + e^{slope*(midpoint-x)}},$$

where $midpoint = \frac{1}{2}(x^{5th} + x^{95th})$ and $slope = \frac{2.944}{x^{95th} - midpoint}$, where $x^{p^{th}}$ is the p-th percentile of the empirical values of x and 2.944 is found by solving $T(x^{95th})$.

In the case of categorical regressors, the observed default rate (ODR) for each category is computed, which is used in computing a $score$ as follows:

$$ODR = \frac{1}{1 + e^{score}} \rightarrow score = \ln\left(\frac{1}{ODR} - 1\right).$$

These $score$ values are not values between zero and one, so a **linear transformation** is performed:

$$T_{linear}(x_i) = \frac{x_i - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})},$$

where x_i is a value from the vector of regressor vector $\boldsymbol{x}$ for $i = 1, \dots, n$.

[5] The German Credit Risk data set is available at Kaggle (link: https://www.kaggle.com/datasets/uciml/german-credit).

3.1.1.4 Rabobank Model Selection algorithms

Currently Rabobank uses several model selection algorithms: **backward** stepwise, **forward** stepwise, and **bidirectional** stepwise, the details of which are discussed in section 2.2.2.2. These algorithms will be used where the model performance will be measured in terms of the **AIC** (see section 2.2.2.1), because this performance statistic considers the complexity of the model and penalizes every additional regressor added. Furthermore, model uncertainty in terms of F-test p-value selection will not be modelled as initial experiments show that the impact of this uncertainty is small, while the additional computational time greatly increases. It is crucial to state that currently all data from a sample is used for both model selection and parameter estimation, without splitting.

3.1.1.5 Proposed Model Selection Algorithms

We identified, in chapter 2, several potential solutions to biased inference. The first solution is to use the **Lasso** objective when selecting models, and every regressor that is selected in the Lasso, i.e. every regressor for which its parameter is not shrunken to zero, is subsequently used in a model for which the parameters are estimated again such that the log-likelihood is maximized.

The second solution is found by manipulating the data during the model selection stage by **adding noise** to the data, and removing this noise when estimating parameters after having selected a model. Thirdly, removing a fraction of the data sample during the model selection stage and adding it back when estimating parameters can be a solution, this is called **data carving**. Lastly, **data splitting**, i.e. using half of the data for model selection and the other half for parameter estimation, does yield valid confidence intervals for beta, but these confidence intervals might be too wide to be useful: that is, the final model is expected to be less capable in distinguishing between positive (bad customers) and negative (good customers) outcomes. Techniques involving the manipulation or strategic withdrawal of data can be combined with any model selection algorithm.

3.1.1.6 Outcome Data-frames

The relevant outcome of the experiments can be visualized as a framework, visualized in Figure 8 below. For each model selection algorithm, and each data-strategy, statistics for each regressor are computed. These statistics are the mean and standard deviation of the sampling distribution of regressor parameter β_i, $i = 1, \dots, n$.

	Model Selection Algorithm:	**Forward Stepwise**				...	**Lasso**			
Regressor	strategy:	None	Data Splitting	Data Carving	Adding Noise		None	Data Splitting	Data Carving	Adding Noise
x_1	*mean beta*									
	std beta									
	P(accept) from t-values									
x_2	*mean beta*									
	std beta									
	P(accept) from t-values									
...	...									
x_n	*mean beta*									
	std beta									
	P(accept) from t-values									

Figure 8: Framework for Analysis of Outcomes in Identifying Biased Inference

The reported mean beta for the different combinations of model selection algorithms and data strategies will be compared to the known true value of beta, and the standard deviation may be used as a metric of stability of the fitted model. The probability of accepting a regressor as significant, after model selection, can be estimated from the array of t-values for a regressor, transforming it to a p-value, setting a threshold p-value, and counting every p-value smaller than threshold.

Lastly, the selected model must be stored each time such that comparisons can be made in terms of stability of the model selection algorithm, combined with a data strategy, because when selecting

the same model consistently across samples the probability of strangely shaped distributions may decrease.

3.1.2 Biased Model Performance

While inference can be useful in verifying hypothesized relations between, in this case, PDs and (a set of) regressor(s), the final model is used for **prediction**. That is, a fitted logistic regression model can be used to estimate the PD of new loan applicants.

Hence, valid inference is necessary otherwise a model is selected based on the wrong reason simply because the model selection algorithm gave that regressor false power. Secondly, when explicitly searching for regressors that are linked to PDs, we may end up with **a model that predicts only noise**. That is, the outcome variable was naturally distributed as noise: there are no real patterns in the data.

In chapter 2 we saw that k-fold CV may yield an inflated ROC AUC. We expect model selection, after which the selected model is fitted using k-fold CV, may add to this **optimization bias**. This can be problematic when a final model is accepted based on its performance (e.g. accept the model if $AUC \geq 0.7$), while the model predicts noise. Hence, an useful metric of this optimization bias is the probability that a model that is fitted to noise is accepted to be put into production. This probability can be estimated as follows:

(i.) Drawing $n = 1000$ data samples where the outcome variable y is not linked to the regressor matrix $\boldsymbol{X}$, but is simply a Bernoulli trial with a known probability of success p, i.e. the true AUC equals 0.5.
(ii.) Performing model selection on this data sample.
(iii.) Fitting the model using a form of k-fold CV.
(iv.) Dividing the number of instances where the average AUC across k folds, or for/across test sets, is greater than a fixed threshold of, say, 0.7, by the total number of samples drawn.

The above can be performed for different combinations of sample size n and number of folds k, as well as for different model selection algorithms and data strategies. The results may be presented in a table, where, for the different (n, k) combinations and $AUC\ Thresholds$, the estimated probability of accepting a model that predicts noise is given.

3.2 Overview for Quantifying Bias

We split bias in two parts: inference and prediction. For inference, we are interested in the value of regressor parameters that the model selection procedure found, and compare these to the known true value. Furthermore, the reported t-values are of interest such that we can comment on how well model selection procedures can recognize power.

For the prediction part we perform a simplified complete model selection procedure. That is, we select a model, fit it, and test its performance. The important part here is that the data is random, such that the true performance is $AUC = 0.5$. By completing a complete model selection procedure we quantify the total bias resulting from, possibly, false power by model selection and optimization bias from k-fold CV and model selection.

An overview of an experiment, for any model selection algorithm combined with a data strategy, is given in Figure 9. The analysis of results, for inference and prediction, are along the lines of Figure 8 and the table described in section 3.1.2 respectively.

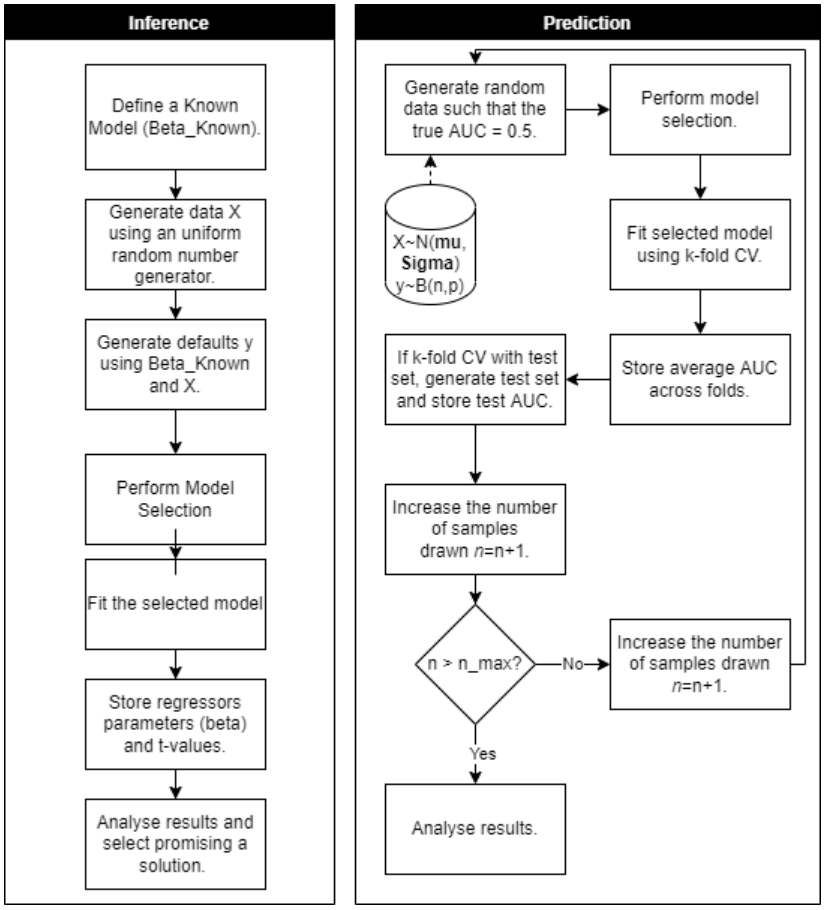

Figure 9: Conceptual model for identifying biased inference and inflated performance.

3.3 Expected Practical Implications

The reason for not working with real PD data in the experiments from 3.2 is that by using artificial data we can define a known model for inference, and a known performance for prediction. Would we use natural data, the results would mean nothing, as we have nothing to compare them to. The results of these controlled-data experiments may show us which algorithms and techniques identify relevant regressors, give good estimates of true regressor parameters, and have a low probability of accepting a model that predicts noise. Then, if any particular model selection algorithms combined with certain data strategies, which are currently not in use by Rabobank, are performing well in the three criteria above, Rabobank may consider using these algorithms and strategies: so, the potential implications of this research are not abstract.

3.4 Concluding Chapter 3

Now we answer the research question *'What does a conceptual model to find bias in selected Rabobank credit risk models look like, and which techniques that are expected to reduce this bias should be added to this conceptual model?'*

We developed a conceptual model that can be used to find biased inference and performance statistics for any combination of model selection algorithm and data strategy (Figure 9), as well as overviews for data analysis that can quantify and compare the bias across model selection algorithms and data strategies.

To answer the second part of the research question, the techniques that are expected to reduce biased inference, and possibly optimization bias, are to use Lasso as a model selection algorithm, and to strategically use the data by performing either data-carving or adding noise.

We now continue to the next chapter, where the experiments will be formally defined. That is, all experiments will be defined in terms of parameters, such as sample size, number of samples, etc.

4. Experiments

In the fourth chapter, we conduct experiments using the model from chapter three. First biased inference is addressed (Section 4.1), then biased model performance is addressed (section 4.2), and lastly we conclude chapter four (section 4.3).

4.1 Biased Inference

In this section we define the parameters of the experiments necessary to run in order to understand what, empirically, drives biased regressor parameters, and false power of regressors (section 4.1.1). Thereafter we discuss and present the general results (section 4.1.2) of these experiments, and elaborate on well performing strategies in terms of selecting the correct regressors and β estimates (section 4.1.3). Lastly, we discuss the Lasso results (section 4.1.4).

The regressor data used for these experiments is randomly generated uniformly distributed data, where defaults are randomly generated with a PD obtained via a known model (Table 4). This data combined with the specified known model and an intercept $\beta_{Intercept} = -1.5e^2$, yields an average default rate of 5%. In the experiments we generate samples with sample size $n = 700$.

Table 4: The value of the regressors parameters used in generating scores from artificial data, after which the score is transformed into a PD via a logit transformation. With this PD a default is binomially generated.

$\beta_0 = 0$	$\beta_1 = 5$	$\beta_2 = 2$
$\beta_3 = 0$	$\beta_4 = 0$	$\beta_5 = 5$
$\beta_6 = 0$	$\beta_7 = 0$	$\beta_8 = 1$

4.1.1 Strategies Simulated

A strategy is a combination of a model selection algorithm and a method of using the data (e.g. data splitting or data carving). In all cases, each strategy is performed a thousand times by means of generating a data sample $(n = 700)$ using the known model, and subsequently executing the strategy on the generated sample.

4.1.1.1 Model Selection Algorithms

Three stepwise model selection algorithms are currently in use at Rabobank: forward, backward, and bidirectional stepwise. These stepwise algorithms are configured by means of parameters, given and explained in the table below (Table 5).

Table 5: Stepwise Model Selection Algorithm Parameters and their Values.

Parameter	**Explanation**	**Value in Stepwise Experiments**
Endog	The endogenous data vector y.	The y vector.
Exog	The exogenous data matrix X.	The regressor matrix or data frame.
Criterion	Model performance evaluation criterion.	AIC.
Report_Card	A Boolean that must be set to true if a printable report card must be generated.	False.
Forced	A set of regressors that is unconditionally added to the solution.	Empty.
Max_var	The maximum number of variables present in a model.	The total number of regressors present in the data set, which is 9.
Max_vif	A constraint that tries to find a solution with a variance inflation factor below a certain threshold, if this is impossible ignore the constraint.	2

Family	The type of Generalized Linear Model (GLM). See *statsmodels*[6].	sm.families.Binomial().
Positive_beta	A Boolean that is set to true if regressor parameters can only take positive values.	False.
Second_layer_ criterions.FTest	An additional constraint on whether to accept a new model as statistically better based on p-values of an F-test.	P_value_enter = 0.05. P_value_leave = 0.2.

Additionally, we use Lasso as a model selection algorithm by accepting and subsequently fitting regressors that, after a solution to the Lasso optimization function is found, are not equal to zero. Recall (section 2.3.3) that the Lasso optimization objective contains a constant λ which is a multiplication constant for the sum of the absolute values of regressor parameters in a model, which is added to the negative log likelihood $-\mathcal{L}$. In our Lasso experiments we make use of the One Standard Error (1SE) heuristic, discussed by e.g. Chen & Yang (2021), to find a λ that prefers a more sparse model if the error, in our method this is the absolute value of the reported log likelihood, of that model is within $1SE$ of the global minimum observed. In the experiments this means that we generate data, find a λ using all data, and then perform model selection using specifically that λ.

4.1.1.2 Data Strategies

Using our data in particular ways may impact the severity of the post model selection biased inference. We found that data splitting may give us valid inference at the cost of increased widths of confidence intervals. Furthermore we found that data carving and adding noise may be well performing substitutes for data splitting that may be preferred in terms of the performance of a final model being better with these strategies compared to data splitting. An overview of these data strategies is given below (Figure 10).

[6] Statsmodels GLM: https://www.statsmodels.org/stable/glm.html.

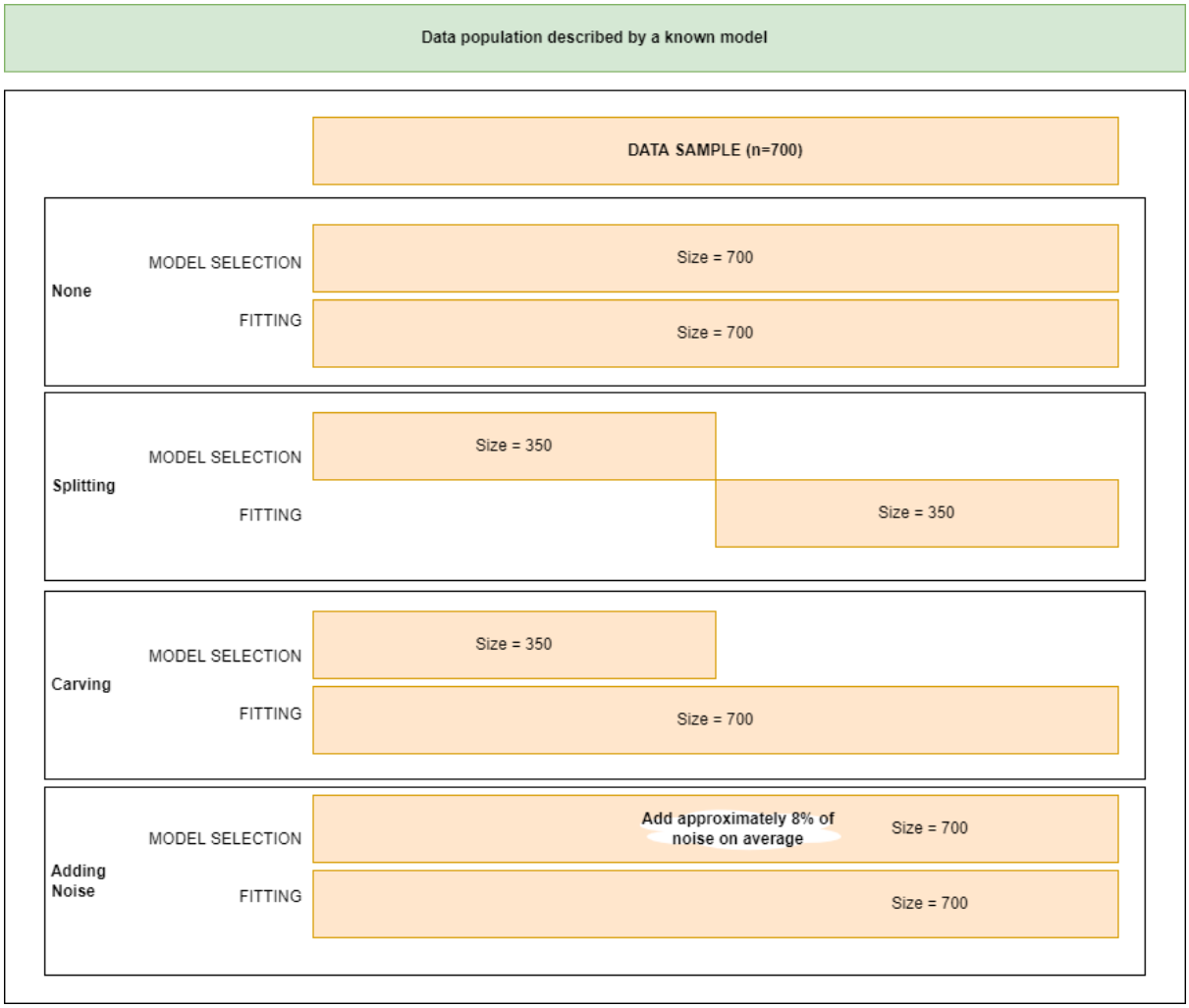

Figure 10: An overview of the simulated data strategies. After Tibshirani (2018).

In all cases data samples are randomly generated using a known model. In the case of data splitting, the drawn sample is randomly split in equal sizes. In the case of data carving, the model selection sample n_{MS} is randomly sampled from the drawn data sample where $n_{MS} = 0.5n$. In the case of adding noise, the noise added to the regressor data sample is generated from a multivariate normal distribution with mean zero and variance 1, multiplied by a constant 0.05.

4.1.2 General Results

We now present the results (Figure 11) of the inference experiments in tabular form according to the framework discussed in Figure 8. These results are analysed in the next chapter, after aspects of well performing strategies are defined.

	Model Selection Algorithm:	Defined	None	Forward Stepwise			Backward Stepwise			Bidirectional Stepwise			Lasso (lambda 1SE)		
Regressor	*strategy:*	*Beta Vector*	*Direct Fit*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*
x0	mean beta	0		0,033	0,014	0,047	-0,058	0,185	0,003	0,059	0,001	0,064	-0,002	0,083	0,073	-0,127	-0,01	-0,18	-0,077
	std beta			1,406	1,167	1,073	1,337	1,085	1,175	0,938	1,064	1,366	1,034	1,046	1,336	0,843	1,12	0,836	0,905
	P(accept) from t-values			0,35	0,06	0,19	0,3	0,13	0,05	0,11	0,1	0,3	0,03	0,18	0,28	0,05	0,04	0,09	0,09
	n			161	167	181	169	421	436	425	390	154	190	149	170	705	492	485	574
x1	mean beta	5	5,26	5,28	5,508	5,232	5,393	5,326	5,53	5,275	5,301	5,25	5,497	5,196	5,3	5,31	5,583	5,266	5,28
	std beta		1,063	1,113	1,789	1,071	1,144	1,153	1,693	1,102	1,073	1,056	1,926	1,022	1,094	1,066	1,67	1,091	1,096
	P(accept) from t-values		1	1	0,99	1	1	1	0,99	1	1	1	0,99	1	1	1	1	1	1
	n		1000	1000	999	997	1000	1000	973	978	992	1000	997	999	1000	1000	998	1000	1000
x2	mean beta	2	2,087	2,232	2,213	2,329	2,235	2,159	2,191	2,205	2,169	2,209	2,242	2,262	2,22	2,154	2,189	2,223	2,163
	std beta		0,796	0,742	1,169	0,748	0,704	0,729	1,263	0,759	0,713	0,705	1,278	0,68	0,699	0,744	1,189	0,738	0,757
	P(accept) from t-values		0,82	0,88	0,54	0,9	0,9	0,85	0,5	0,87	0,86	0,89	0,54	0,91	0,89	0,85	0,5	0,9	0,83
	n		1000	924	742	737	923	988	874	870	966	931	725	709	935	983	807	798	979
x3	mean beta	0		-0,051	0,027	0,011	0,203	0,025	-0,002	0,064	-0,056	-0,062	0,072	0,082	0,033	-0,063	-0,098	-0,176	-0,074
	std beta			1,367	1,149	1,121	1,432	1,093	1,204	0,893	1,061	1,382	1,201	1,071	1,387	0,859	1,174	0,798	0,885
	P(accept) from t-values			0,26	0,08	0,18	0,38	0,16	0,04	0,11	0,12	0,3	0,07	0,18	0,37	0,07	0,05	0,08	0,07
	n			153	173	182	168	416	436	427	412	165	181	184	161	707	485	490	606
x4	mean beta	0		0,103	-0,003	0,05	0,184	0,085	-0,118	-0,036	-0,071	-0,197	-0,18	-0,121	0,248	-0,114	0,01	-0,118	-0,082
	std beta			1,383	1,133	1,126	1,312	1,104	1,101	0,901	1,077	1,425	1,233	1,014	1,369	0,853	1,079	0,839	0,862
	P(accept) from t-values			0,33	0,07	0,24	0,27	0,14	0,05	0,11	0,15	0,4	0,04	0,19	0,31	0,06	0,04	0,08	0,07
	n			181	149	178	155	403	411	422	383	172	180	174	156	684	512	482	592
x5	mean beta	5	5,222	5,351	5,466	5,255	5,259	5,328	5,567	5,252	5,303	5,291	5,553	5,197	5,316	5,369	5,69	5,13	5,333
	std beta		1,114	1,134	1,761	1,045	1,12	1,112	1,67	1,118	1,103	1,092	1,861	1,046	1,152	1,079	1,848	1,05	1,135
	P(accept) from t-values		1	1	0,99	1	1	1	0,99	1	1	1	1	1	1	1	0,99	1	1
	n		1000	1000	999	996	1000	1000	973	987	992	1000	998	999	1000	1000	997	994	1000
x6	mean beta	0		0,002	0,014	0,136	-0,139	0,049	0,02	-0,047	-0,031	0,11	0,013	-0,118	-0,188	-0,046	0,02	-0,134	-0,113
	std beta			1,403	1,176	1,086	1,422	1,087	1,136	0,898	1,088	1,399	1,111	1,02	1,349	0,871	1,186	0,816	0,883
	P(accept) from t-values			0,32	0,07	0,19	0,36	0,14	0,04	0,11	0,12	0,32	0,05	0,15	0,32	0,08	0,05	0,09	0,08
	n			170	161	191	156	411	408	401	400	165	198	178	160	703	480	509	582
x7	mean beta	0		0,026	0,014	-0,001	-0,242	0,04	0,051	0,071	0,066	-0,052	-0,067	0,155	0,03	-0,042	-0,02	-0,143	-0,042
	std beta			1,387	1,178	1,07	1,348	1,083	1,113	0,928	1,075	1,478	1,036	1,113	1,33	0,866	1,216	0,846	0,913
	P(accept) from t-values			0,31	0,05	0,18	0,35	0,13	0,05	0,1	0,11	0,38	0,03	0,22	0,31	0,07	0,05	0,09	0,08
	n			175	157	174	155	419	419	411	399	178	200	177	170	684	498	482	616
x8	mean beta	1	1,04	1,642	-0,178	1,538	1,612	1,348	1,175	1,299	1,409	1,593	1,026	1,512	1,613	1,233	1,138	1,265	1,288
	std beta		0,732	0,533	1,032	0,718	0,523	0,623	1,221	0,728	0,644	0,54	1,139	0,702	0,52	0,723	1,244	0,772	0,75
	P(accept) from t-values		0,3	0,63	0,18	0,63	0,63	0,39	0,18	0,43	0,46	0,54	0,13	0,57	0,63	0,37	0,17	0,45	0,41
	n		1000	552	382	381	510	748	606	606	723	524	352	369	518	852	537	535	768

Figure 11: Results of the inference experiments, in tabular form.

4.1.3 Stepwise and Data Splitting

It can be seen from Figure 11 that data splitting increases the standard deviation of β estimates compared to using the same generated sample for both model selection and fitting on average. However, data splitting reduces the probability of accepting irrelevant regressors in a model significantly. The t-value sample distributions for Forward stepwise and data splitting (Figure 12) are not bimodal, whereas the same distributions as a result of not using a data strategy are (Figure 13).

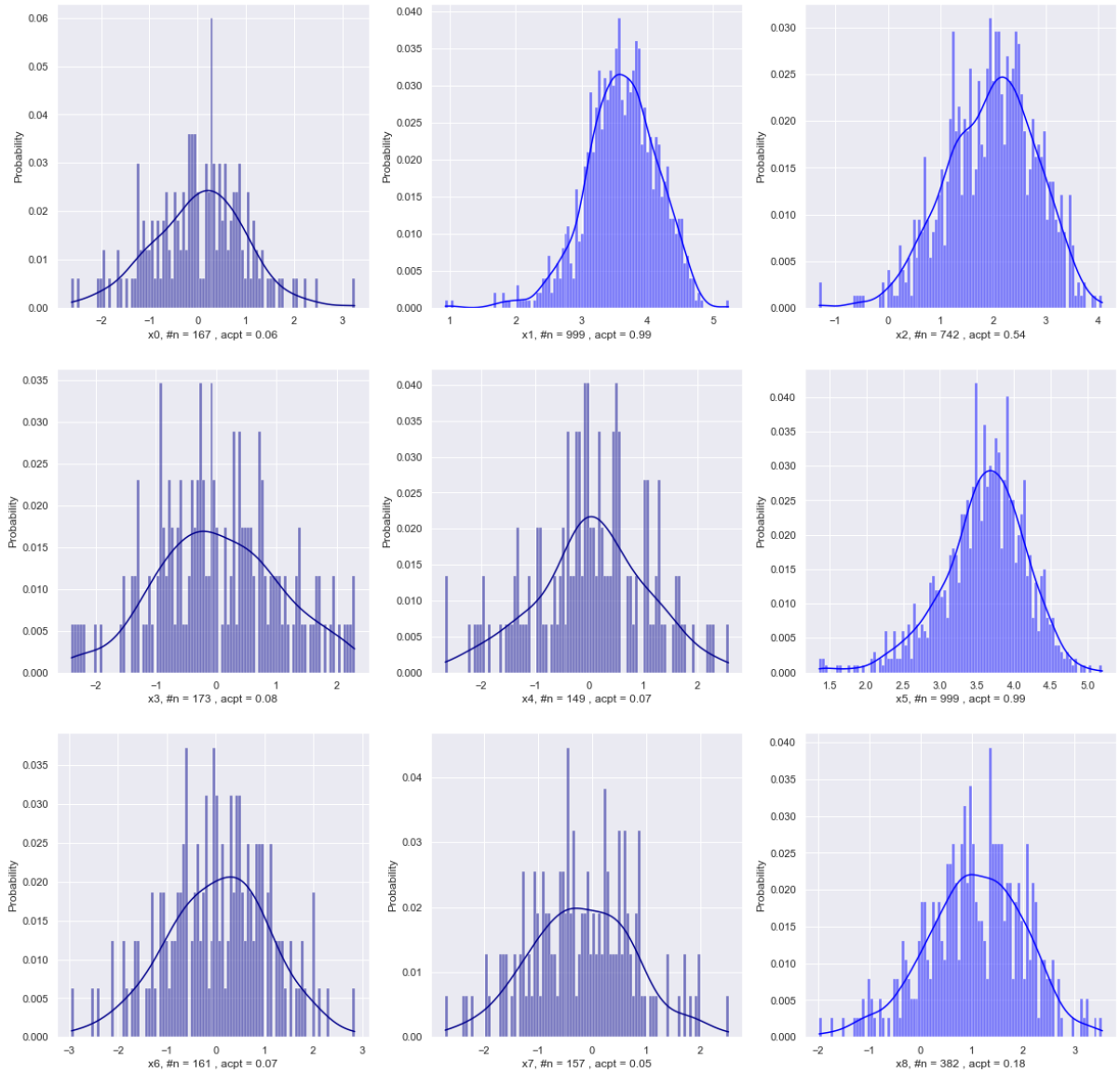

*Figure 12: t-value sample distributions as a result of **Splitting** the generated sample. The one half is used for **Forward** stepwise model selection, the other half for fitting the selected model. Observe how regressors $x2$ and $x1$ are selected and accepted less frequently compared to the other relevant regressors x_1 and x_8, which may be a result of their true betas being smaller.*

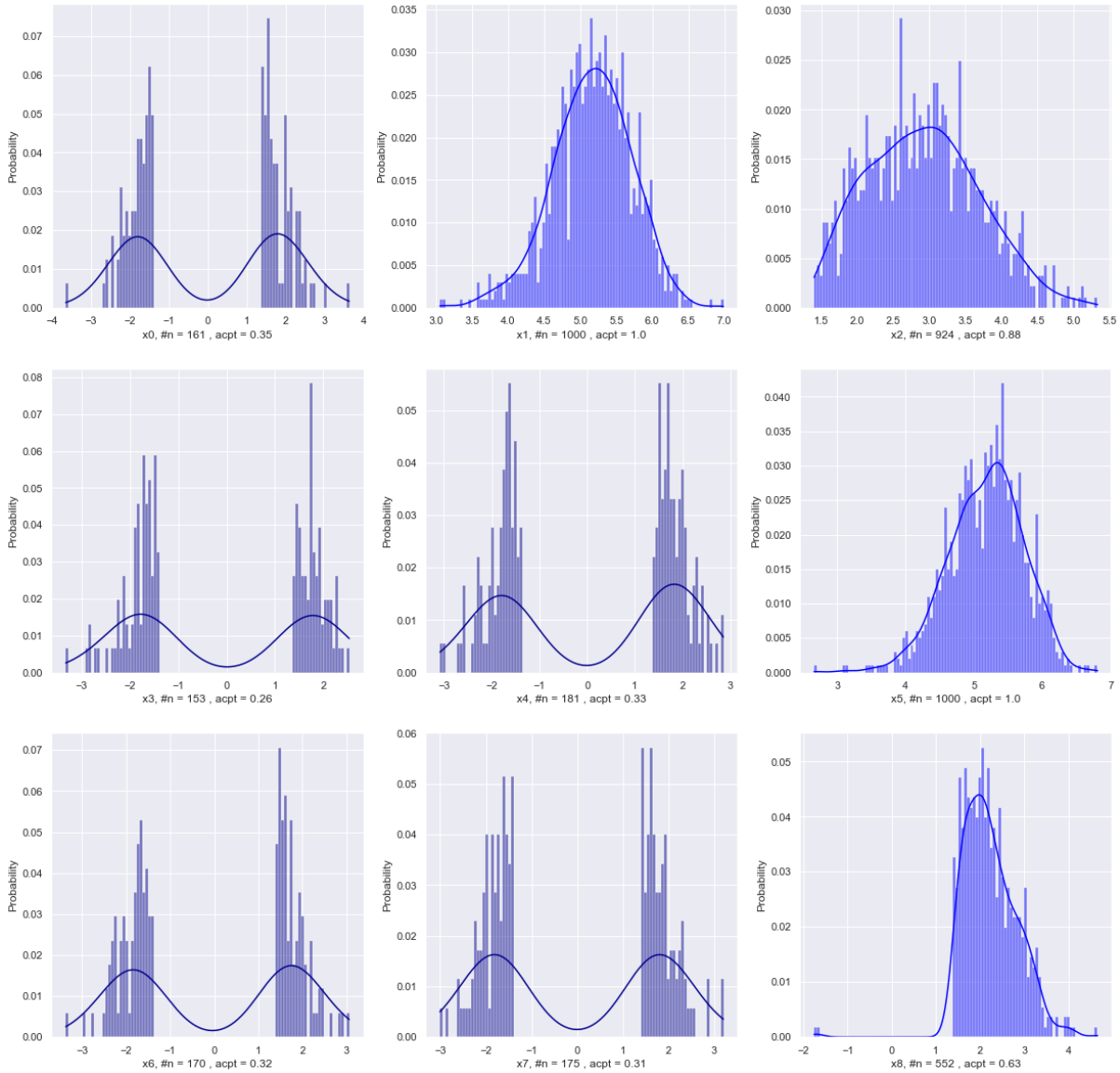

*Figure 13: t-value sample distributions as a result of using the generated data sample for both **Forward** stepwise model selection and subsequent fitting of the selected model. Observe how the t-value sample distributions of irrelevant regressors are strongly bimodal. Furthermore, their acceptance rate may be seen as problematic as well. That is, a modeller can observe with a high probability that a noise regressor is significant in predicting credit risk after model selection.*

4.1.4 Lasso 1SE Rule

It is interesting to see (Figure 11) that lasso selects most regressors in a model most of the trials, but, when not using a data strategy, these regressors do not get accepted in a subsequent fit. Using any other data strategy turns this around: irrelevant regressors get selected less often, but accepted after fitting the selected model more often.

4.2 Biased Model Performance

In this section we define the hyperparameters of the experiments necessary to run in order to understand what, empirically, drives biased performance of logistic regression models (4.2.1). Thereafter we discuss and present the general results (section 4.2.2) of these experiments, and elaborate on well performing strategies in terms of having a low risk of artificially inflating performance (section 4.2.3). Lastly, we discuss the Lasso results (section 4.2.4).

The data used for these experiments is randomly generated data using a multivariate normal model with a mean vector of zero, and a covariance matrix that is an identity matrix times ten. Defaults are randomly generated with $PD = 0.15$ via a binomial model. The covariance matrix, with number of rows and number of columns equal to the number of regressors in the generated data, is defined as follows:

$$Covariance\ Matrix = 10\ \cdot Identity\ Matrix = 10 \cdot \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} 10 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 10 \end{bmatrix}.$$

Several parameters during model selection and model fitting may impact the final model performance that is reported. Amongst others, the sample size n_{sample} and the number of regressors in the dataset $numvar$ were identified in the literature study of Chapter 2. For every model selection algorithm combined with a data strategy, n_{sample} is evaluated for $[500, 1000, 2000]$ and $numvar$ is evaluated for $[5, 10, 15]$ a number of times $n_{experiments} = 1000$. The test AUC, which is needed to evaluate the performance of k-fold CV with a separate test set, is computed based on X_{test} and y_{test} both of size $n_{test} = 0.25 n_{sample}$ generated via the same multivariate normal and binomial distribution as the training sets on which k-fold CV is applied.

4.2.1 Strategies Simulated

Every stepwise model selection algorithm (forward, backward, bidirectional) discussed in section 3.1.1.4 as well as Lasso with a fixed $\lambda = 0.1$ are simulated, all in combination with the data strategies *None, Splitting,* and *Carving* visualized in Figure 10. The data strategy of *Adding Noise* is not evaluated because it would be nonsensical to add noise to noise. Not using a data strategy, i.e. using the drawn sample for both model selection and model fitting, is evaluated to have a baseline to which data carving and data splitting can be compared. For each experiment the average AUC across 5 folds in k-fold CV is stored, as well as the maximum AUC from each 5 folds. The maximum AUC is the result of a fitted model, and this fitted model is tested on X_{test} and y_{test} from which a 'test AUC' is computed.

4.2.2 General Results

The results of the simulations to quantify optimization bias are given in Figure 14. Its structure is along the model discussed in section 3.1.2, where the output is the fraction of times out of the total times the experiments converged ($n_{experiments} = 1000$) the reported AUC was above a certain threshold $AUC_T = [0.60, 0.61, \ldots, 0.70]$. For each data strategy, results are shown for each model selection algorithm in terms of the test AUC, the average AUC across 5 folds, and the maximum AUC present in 5 folds.

Figure 14: The prediction results for the data strategies None, Splitting, and Carving. For each of these data strategies results are given for the model selection algorithms Forward, Backward, and Bidirectional stepwise as well as Lasso with $\lambda = 0.1$. Red cells indicate a fraction, given an AUC_T, greater than 1%.

4.2.3 Data Carving

The inference results presented in section 4.1.2 indicated that data carving appears not to yield more valid inference than the other strategies and the prediction results presented in Figure 14 in section 4.2.2 visibly have fewer red cells than the other data strategies. This means that, on average, data carving seems to yield more valid performance estimations judging based on the average AUC across 5 folds. A more elaborate analysis is part of the next chapter.

4.2.4 Lasso Fixed Lambda

In terms of preventing an inflated AUC, lasso appears not to be significantly better or worse than other model selection strategies. We note that Lasso did only result in a solution, i.e. a fitted selected model with measurable performance, at most 763 times out of 1000 trials for $n_{sample} = 1000$ and 475 times out of 1000 trials for $n_{sample} = 500$ when no data strategy is utilized. This can make comparisons to results of other model algorithms problematic, which is discussed in chapter 5.

4.3 Concluding Chapter 4

We defined experiments such that we can formulate an answer to the research question of this chapter: *Based on the conceptual model of chapter three, how are regressor parameters and model performance of credit risk models biased as a result of Rabobank model selection algorithms and how are these model statistics biased as a result of the hypothesized solutions?*

The conceptual model (Figure 9) presented in chapter 3 was put into practice. That is, we performed the experiments such that biased inference and biased model performance can be quantified. Different model selection algorithms in combination with different data strategies were simulated, using data and hyperparameters resembling practice at Rabobank.

The model selection algorithms currently in use at Rabobank show symptoms of biased regressor parameters, mainly in terms of consistently selecting irrelevant regressors in a model. Data splitting and data carving, from the results given in Figure 11, can help in improving the stepwise model selection algorithms. However, it appears that it comes at the cost of more volatile model selection. Irrelevant regressors are added to a model more often, but they are not accepted after a subsequent fit more often. This trade-off is further elaborated on in the next chapter.

Using Lasso as a model selection algorithm by accepting regressors in a model when their corresponding parameter does not equal zero is a difficult to evaluate case. From the results in Figure 11 it can be seen that, regarding biased inference, the typical problem of the stepwise algorithms without a data strategy are solved: irrelevant regressors are accepted in a model less frequently after fitting the selected model. However, all regressors get selected most of the time, which is not what model selection is trying to achieve. This may have to do with the $1SE$ method of selecting an optimal λ value, which is discussed in the next chapter.

Lastly, from the results presented in Figure 14, upwards biased model performance appears mainly to be a problem based on the performance evaluation method used. That is, having a separate test set to evaluate the best fit out of k-folds of a selected model performs significantly worse than evaluating the performance of averaging model performance over these k-folds. The obvious drawback is that this is the average performance of k different fitted models, which introduces the problem of selecting the best model. This is further elaborated on in chapter 5.

5. Analysis of Results

In this chapter we analyse the results presented in chapter 4, and finally, based on this analysis, formulate an answer to the main research question: *In what way, if at all, are the regressor-parameters and model performance of credit risk models at Rabobank biased, and how can Rabobank adjust for this bias both statistically and managerially?*

We first recall and summarize aspect of good performing strategies for unbiased model statistics post model selection (section 5.1). Thereafter, we analyse the results of chapter 4 specifically on these aspects (sections 5.2, 5.3, and 5.4). The inference results are mostly presented in terms of histograms: each bar representing the fraction of times a regressor was given a β estimate in that interval, with a red vertical line representing its true value. Lastly we conclude this chapter (section 5.5).

5.1 Aspects of a Well Performing Strategy

In chapter 1 and chapter 2 we discussed what constitutes bias. One type of bias are skewed, bimodal, or worse regressor parameter β sample distributions resulting in unclarity on what the true value of a regressor parameter should be, after model selection. This bias can also occur for t-value sample distributions, potentially resulting in false power of a regressor after it is accepted in a model. Finally, the performance of a model may be inflated by unknowingly accepting noise regressors in a model. This can be problematic for Rabobank when, e.g., a true model with p regressors and a performance of, e.g., AUC_1, whereas the selected model with $p + 2$ regressors, the two additional being noise, has an expected performance of AUC_2 that is much greater than AUC_1.

From the above we can define three concrete aspects of good performance: (i.) accurate mean and standard deviation of β compared to the true values thereof, (ii.) accurate rates of accepting regressors as significant based on t-values compared to the true significance of regressors, and (iii.) a relatively low risk of inflated performance statistics after model selection and k-fold CV. We analyse the results from chapter 4 based on these three aspects of good performance in the next sections.

5.2 Beta Accuracy

Accurate regressor parameter estimations after model selection are assessed on, first, the shape of the β sample distributions. These sample distributions are easy to interpret and from them we can quickly see if the model selection algorithm, combined with a data strategy, is accurately estimating regressor parameters. In the next section we analyse the beta accuracy of the different data strategies: no data strategy (section 5.2.1), data splitting (section 5.2.2), data carving (5.2.3), and adding noise (section 5.2.4).

5.2.1 No Data Strategy

The forward (Figure 15) and backward (Figure 16) stepwise model selection algorithms without a data strategy show a major problem: irrelevant regressors for which $\beta_{true} = 0$ are selected quite often, and their regressor parameter β sample distributions are bimodally distributed. This can be problematic for modellers because they can observe one mode after regression, which, no matter which mode is observed, makes their estimate unaligned with the true value of these irrelevant regressors. Bidirectional stepwise without a data strategy shows very similar results. With Lasso, however, irrelevant regressors are more often selected in a model and also bimodally distributed, but both modes are positioned closer to the true value of zero. This reduces the standard deviations of the beta estimates, increasing the certainty of the modeller in choosing a β value.

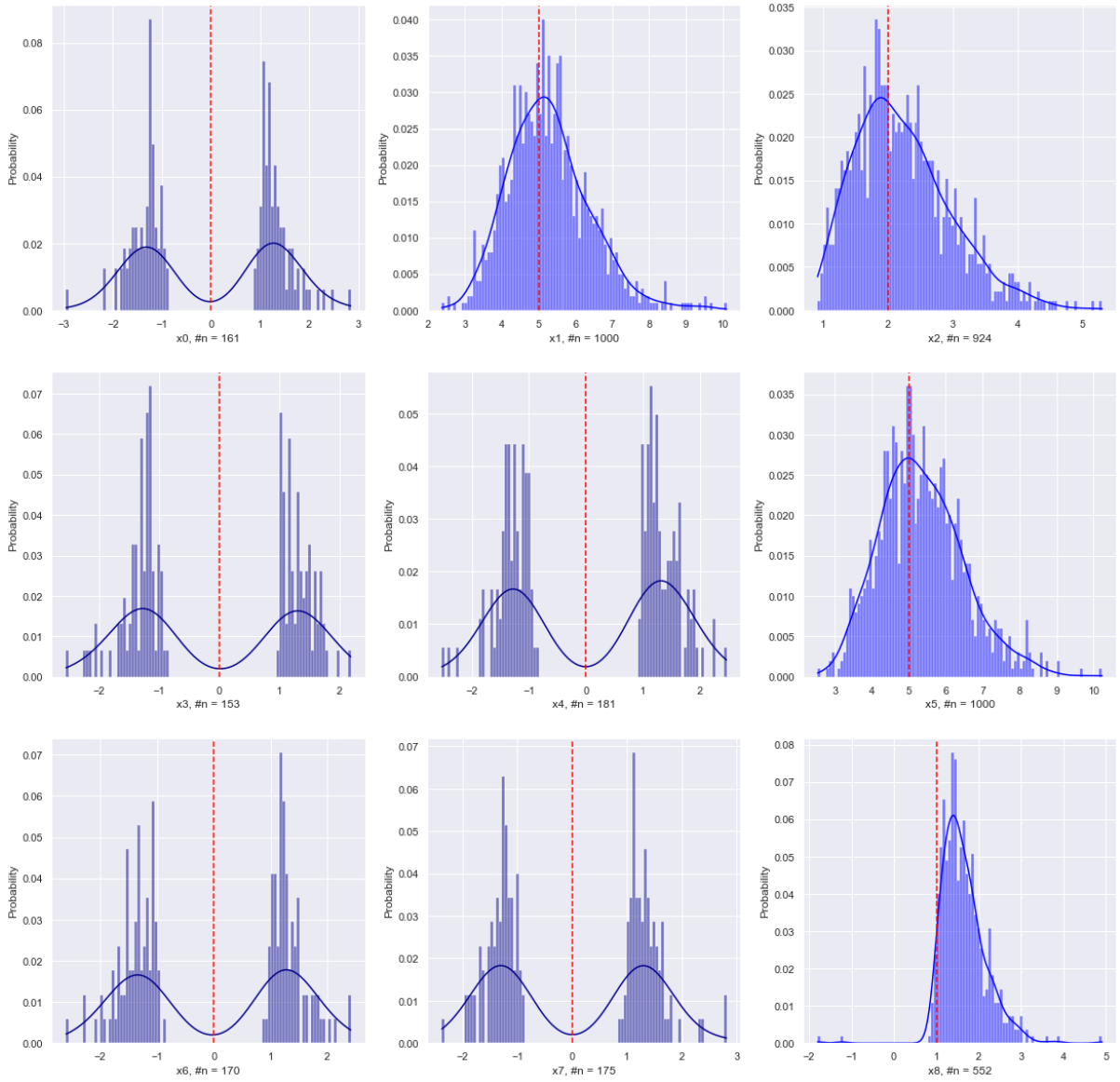

*Figure 15: Regressor parameter β sample distributions as a result of using a generated data sample from a known model for both **Forward** stepwise model selection and subsequent fitting. Observe how irrelevant regressors are bimodally distributed. This, for modellers, increases the uncertainty of the true value of β for these regressors and, furthermore, if they base their regression on any mode they will always be wrong as both modes are not aligned with the true value for these βs (the red vertical dotted lines).*

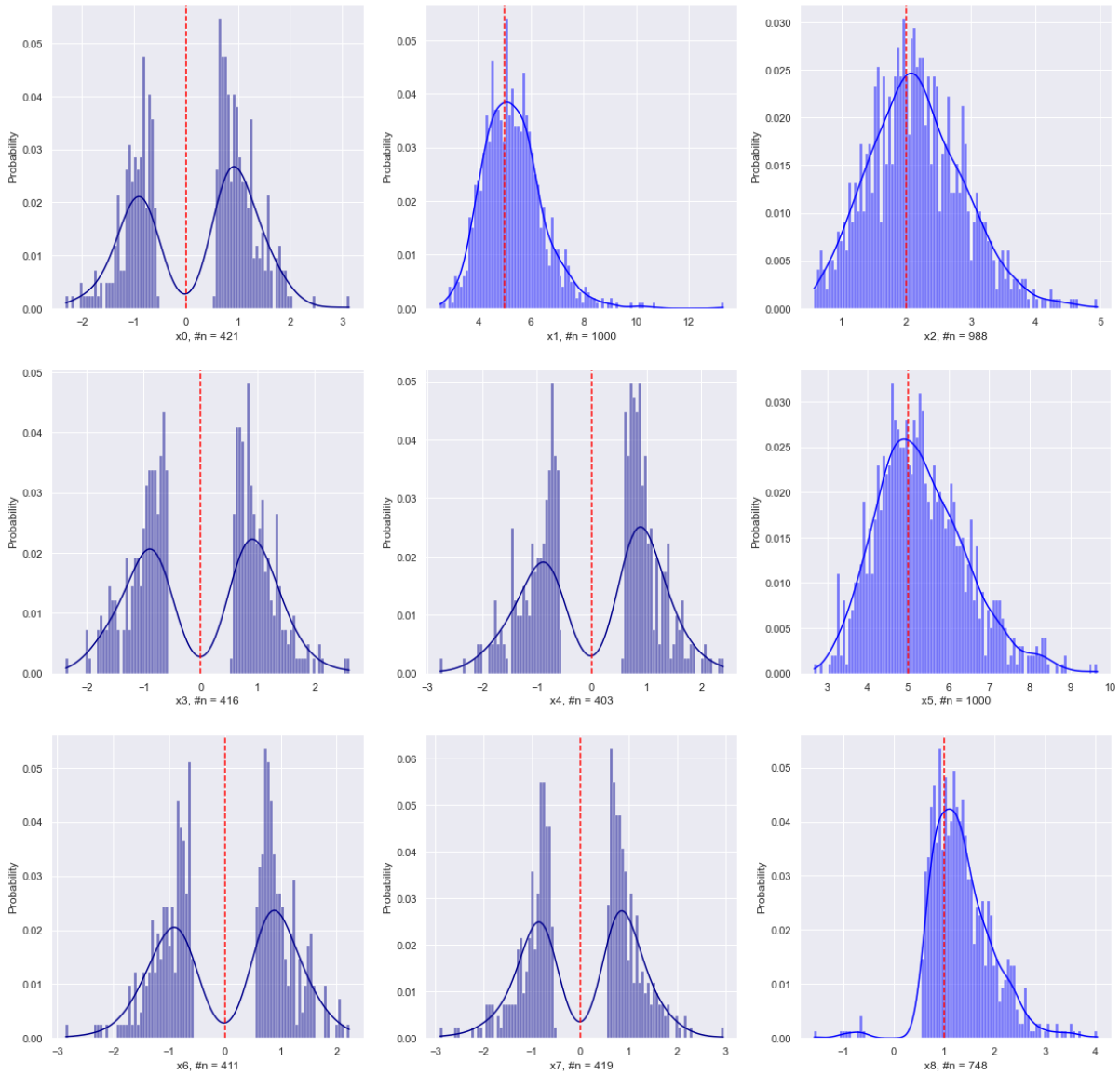

*Figure 16: Regressor parameter β sample distributions as a result of using a generated data sample from a known model for both **Backward** stepwise model selection and subsequent fitting. Observe how, in similar fashion to Forward stepwise (Figure 15) irrelevant regressors are bimodally distributed.*

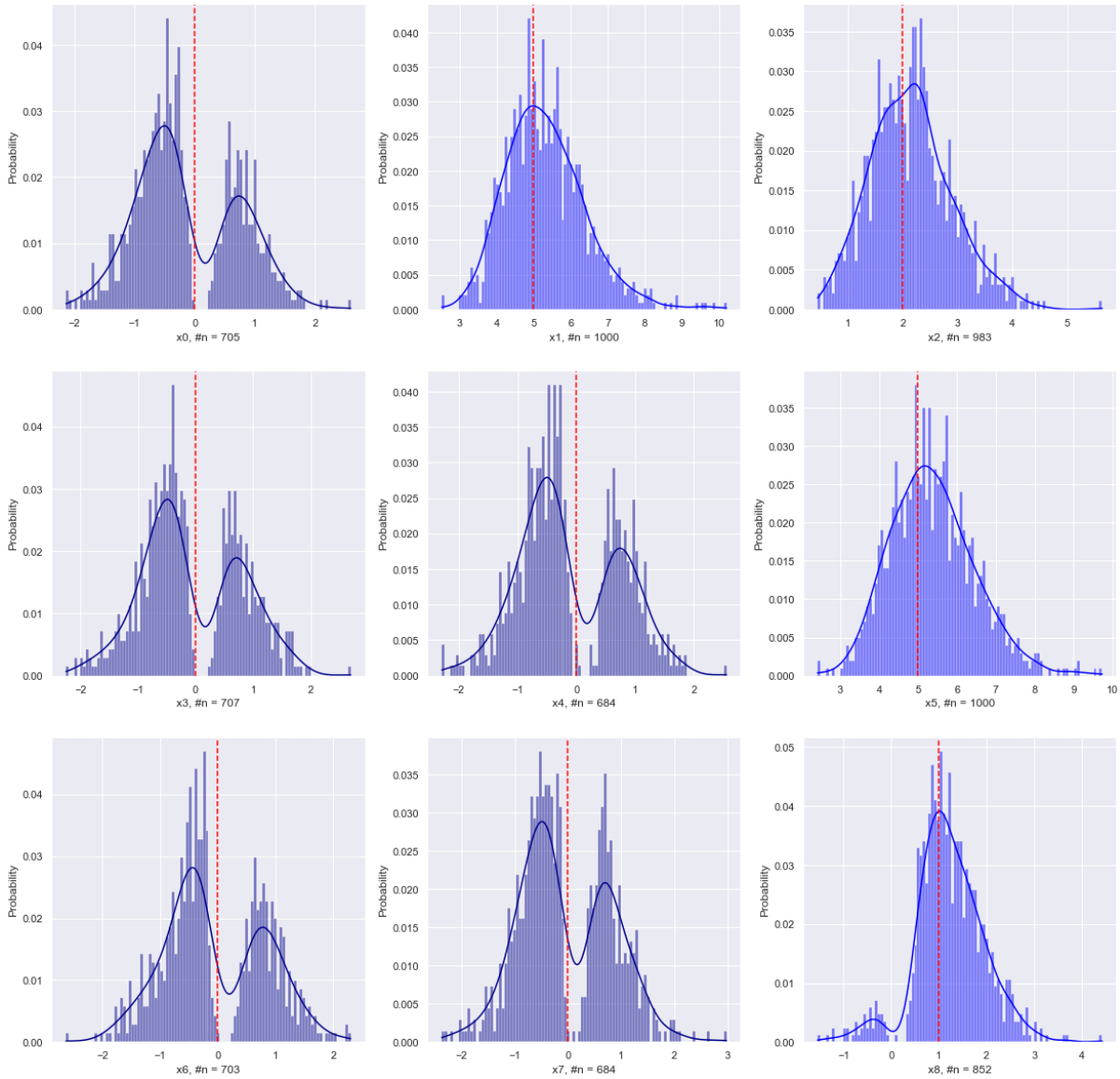

*Figure 17: Regressor parameter β sample distributions as a result of using a generated data sample from a known model for model selection using **Lasso** and subsequent fitting. Observe how, in similar fashion to Forward stepwise (Figure 15) and Backward stepwise (Figure 16) irrelevant regressors are bimodally distributed. It must however be noted that the bimodally distributed irrelevant regressors are closer to the true value of zero, and that these irrelevant regressors get selected more often than with stepwise model selection algorithms. This is discussed in the next section on t-value acceptance rates.*

5.2.2 Data Splitting

Data splitting, it seems, may be an effective cure against bimodal β sample distributions. Observe from the results for Forward stepwise and data splitting (Figure 19) that the irrelevant regressors, compared to not using a data strategy (Figure 15), have disappeared. Similar results can be observed for Bidirectional stepwise. For Backward stepwise (Figure 20), irrelevant regressors do not have bimodal sample distributions anymore, but irrelevant regressors are selected in a model more often than for Forward stepwise.

Furthermore, it can be seen from Figure 18 that for Forward and Bidirectional stepwise the standard deviations of β estimates increase for relevant regressors when using data splitting, whereas they decrease for irrelevant regressors; both compared to the case of not using a data strategy. This is due to the bimodal distributions, which increase standard deviations significantly, not being present in our results for data splitting. For the one mode that exists for each regressors now, however, its dispersion has increased naturally due to using less data to obtain the β estimates. This does not have to be a problem, as each mode is centred around its true β value for both relevant and irrelevant regressors.

This phenomenon is not visible in the Lasso and Backward stepwise results, however, where data splitting increases the standard deviations for all β estimates despite no bimodal or multimodal distributions being present in the results. A logical cause can be that the decrease in standard deviations due to sample distributions being unimodal is less than the increase in standard deviations from using less data yields.

				Deviation from direct fit															
	Model Selection Algorithm:	**Defined**	**None**	**Forward Stepwise**				**Backward Stepwise**				**Bidirectional Stepwise**				**Lasso (lambda 1SE)**			
Regressor	*strategy:*	*Beta Vector*	*Direct Fit*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*
x0	mean beta	0																	
	std beta			1,406	1,167	1,073	1,337	1,085	1,175	0,938	1,064	1,366	1,034	1,046	1,336	0,843	1,12	0,836	0,905
	P(accept) from t-values																		
	n																		
x1	mean beta	5	5,26																
	std beta		1,063	1,113	1,789	1,071	1,144	1,153	1,693	1,102	1,073	1,056	1,926	1,022	1,094	1,066	1,67	1,091	1,096
	P(accept) from t-values		1																
			1000																
x2	mean beta	2	2,087																
	std beta		0,796	0,742	1,169	0,748	0,704	0,729	1,263	0,759	0,713	0,705	1,278	0,68	0,699	0,744	1,189	0,738	0,757
	P(accept) from t-values		0,82																
	n		1000																
x3	mean beta	0																	
	std beta			1,367	1,149	1,121	1,432	1,093	1,204	0,893	1,061	1,382	1,201	1,071	1,387	0,859	1,174	0,798	0,885
	P(accept) from t-values																		
	n																		
x4	mean beta	0																	
	std beta			1,383	1,133	1,126	1,312	1,104	1,101	0,901	1,077	1,425	1,233	1,014	1,369	0,853	1,079	0,839	0,862
	P(accept) from t-values																		
	n																		
x5	mean beta	5	5,222																
	std beta		1,114	1,134	1,761	1,045	1,12	1,112	1,67	1,118	1,103	1,092	1,861	1,046	1,152	1,079	1,848	1,05	1,135
	P(accept) from t-values		1																
	n		1000																
x6	mean beta	0																	
	std beta			1,403	1,176	1,086	1,422	1,087	1,136	0,898	1,088	1,399	1,111	1,02	1,349	0,871	1,186	0,816	0,883
	P(accept) from t-values																		
	n																		
x7	mean beta	0																	
	std beta			1,387	1,178	1,07	1,348	1,083	1,113	0,928	1,075	1,478	1,036	1,113	1,33	0,866	1,216	0,846	0,913
	P(accept) from t-values																		
	n																		
x8	mean beta	1	1,04																
	std beta		0,732	0,533	1,032	0,718	0,523	0,623	1,221	0,728	0,644	0,54	1,139	0,702	0,52	0,723	1,244	0,772	0,75
	P(accept) from t-values		0,3																
	n		1000																

Figure 18: Colour coded standard deviations of the sample distributions of regressor parameters. It can be seen that data splitting increases the standard deviations of the beta estimates for relevant regressors. Because data splitting removes the bimodal distributions, compared to no data strategy, for irrelevant regressors the standard deviations decrease for Forward and Bidirectional stepwise.

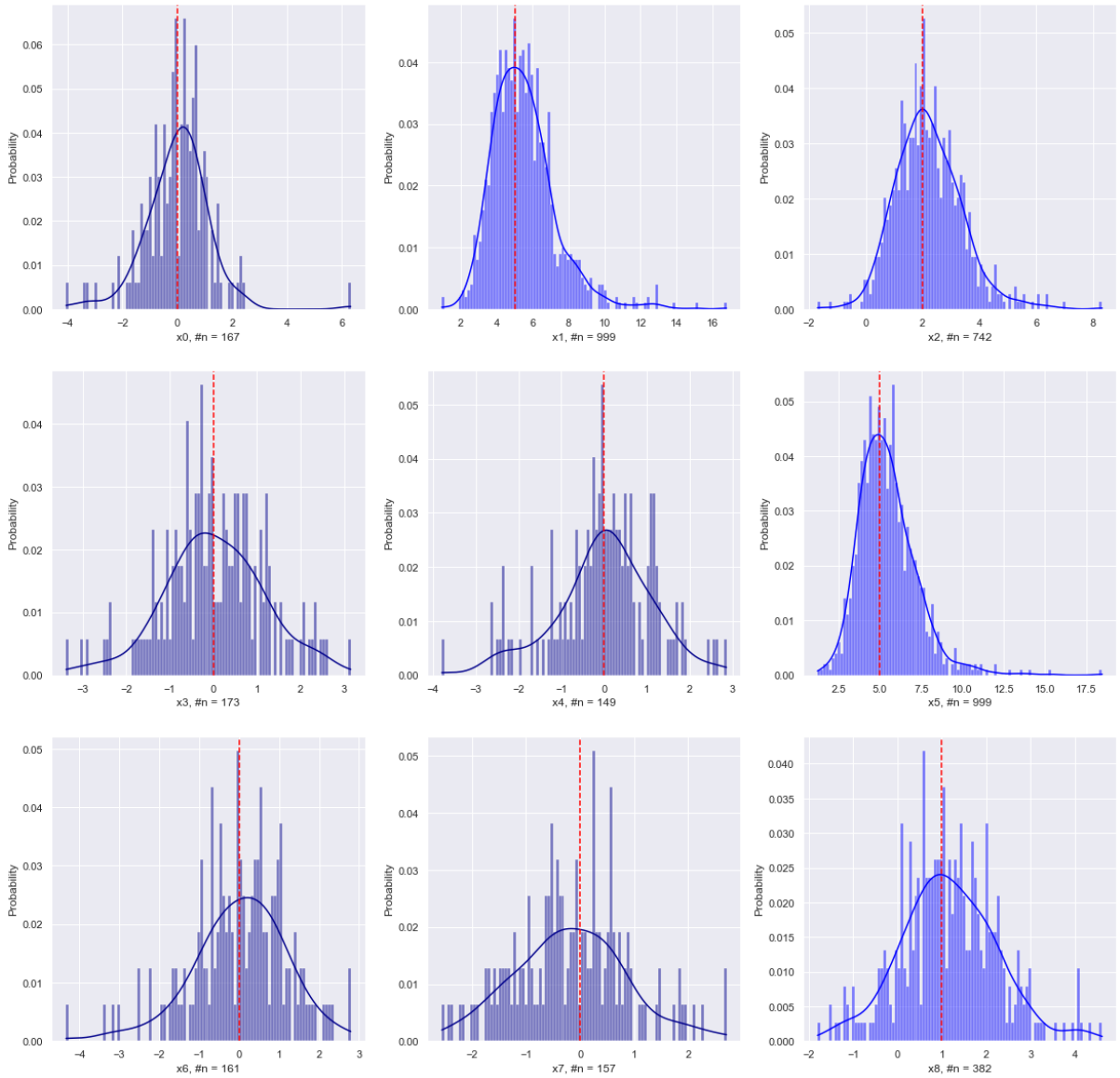

*Figure 19: Regressor parameter β sample distributions as a result of **splitting** a generated data sample from a known model to use one half for **Forward** stepwise model selection and the other half subsequent fitting. Observe how irrelevant regressors are **not** bimodally distributed. Observe furthermore that relevant regressors who's true betas are relatively small $(\beta_{x_2} = 2 \; and \; \beta_{x_8} = 1)$ are selected in a model less often, more accurately reflecting their relative impact on driving PDs.*

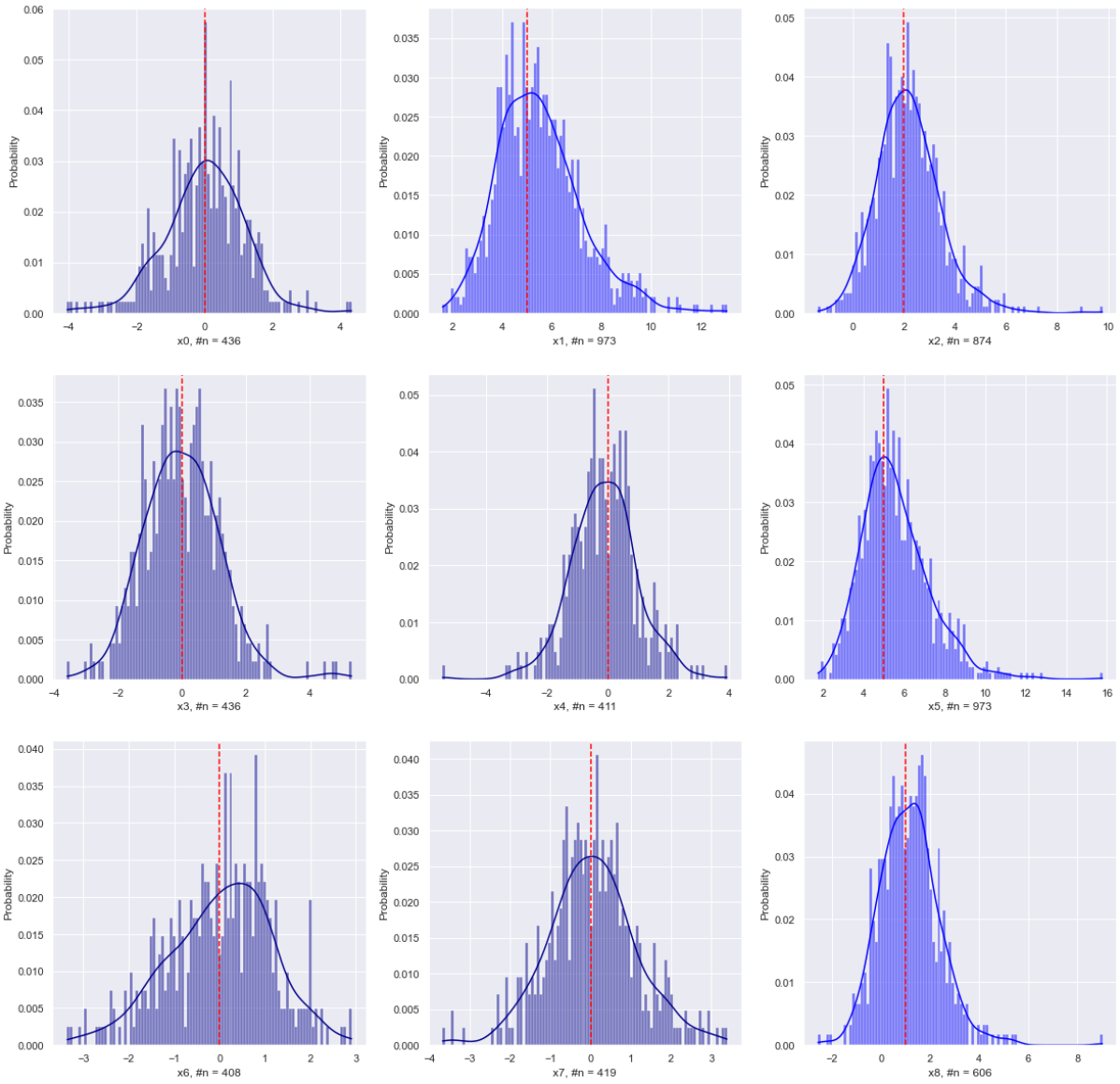

*Figure 20: Regressor parameter β sample distributions as a result of **splitting** a generated data sample from a known model to use one half for **Backward** stepwise model selection and the other half subsequent fitting. Observe how irrelevant regressors are **not** bimodally distributed, but are, compared to Forward stepwise, included in a model more often.*

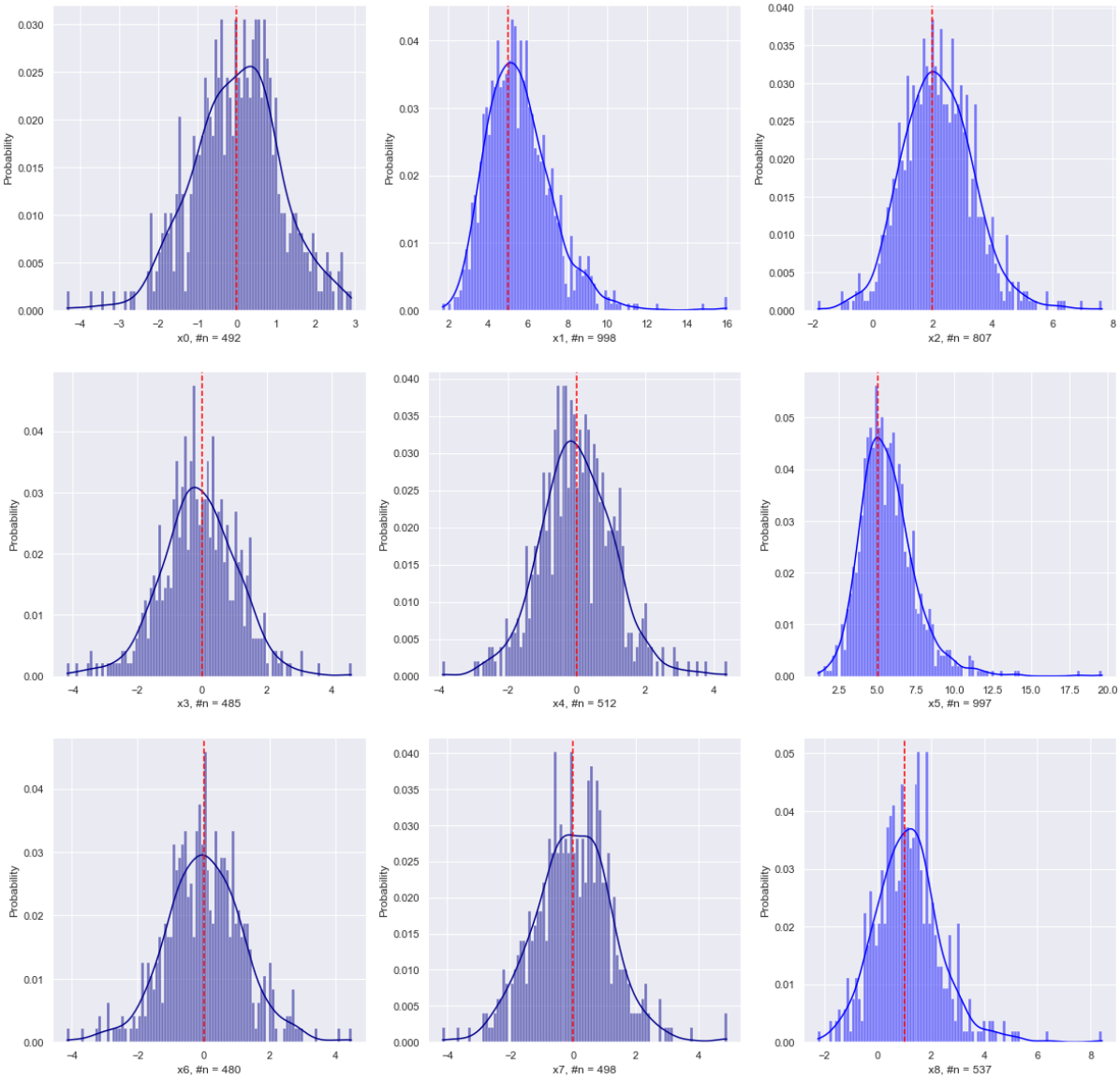

*Figure 21: Results of **splitting** the data sample in combination with using **Lasso** as a model selection algorithm. Similar to Backward stepwise, but different from Forward stepwise, irrelevant regressors are selected relatively often; with the mode of these sample distributions centred around the true value of* 0.

5.2.3 Data Carving

Data carving, on average, reduces the standard deviations of β estimates compared to both data splitting and not using a data strategy. Furthermore, the irrelevant regressors have bimodal distributions to a lesser degree for the Forward (Figure 23) and Bidirectional stepwise results. Note that for the Backward stepwise results (Figure 24), the bimodal structure is further reduced at the cost of more frequently selecting irrelevant regressors in a model. This is contrary to the essence of model selection, and thus t-value acceptance rates must be studied in a future section. For now, we take these results as improvements. Lasso combined with data carving shows comparable results to Backward stepwise, and thus must be further investigated on its t-value sample distributions.

				Deviation from defined model															
	Model Selection Algorithm:	**Defined**	**None**	**Forward Stepwise**				**Backward Stepwise**				**Bidirectional Stepwise**				**Lasso (lambda 1SE)**			
Regressor	*strategy:*	*Beta Vector*	*Direct Fit*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*
x0	mean beta	0		0,033	0,014	0,047	0,058	0,185	0,003	0,059	0,001	0,064	0,002	0,083	0,073	0,127	0,01	0,18	0,077
	std beta																		
	P(accept) from t-values																		
	n																		
x1	mean beta	5	5,26	0,28	0,508	0,232	0,393	0,326	0,53	0,275	0,301	0,25	0,497	0,196	0,3	0,31	0,583	0,266	0,28
	std beta		1,063																
	P(accept) from t-values		1																
			1000																
x2	mean beta	2	2,087	0,232	0,213	0,329	0,235	0,159	0,191	0,205	0,169	0,209	0,242	0,262	0,22	0,154	0,189	0,223	0,163
	std beta		0,796																
	P(accept) from t-values		0,82																
	n		1000																
x3	mean beta	0		0,051	0,027	0,011	0,203	0,025	0,002	0,064	0,056	0,062	0,072	0,082	0,033	0,063	0,098	0,176	0,074
	std beta																		
	P(accept) from t-values																		
	n																		
x4	mean beta	0		0,103	0,003	0,05	0,184	0,085	0,118	0,036	0,071	0,197	0,18	0,121	0,248	0,114	0,01	0,118	0,082
	std beta																		
	P(accept) from t-values																		
	n																		
x5	mean beta	5	5,222	0,351	0,466	0,255	0,259	0,328	0,567	0,252	0,303	0,291	0,553	0,197	0,316	0,369	0,69	0,13	0,333
	std beta		1,114																
	P(accept) from t-values		1																
	n		1000																
x6	mean beta	0		0,002	0,014	0,136	0,139	0,049	0,02	0,047	0,031	0,11	0,013	0,118	0,188	0,046	0,02	0,134	0,113
	std beta																		
	P(accept) from t-values																		
	n																		
x7	mean beta	0		0,026	0,014	0,001	0,242	0,04	0,051	0,071	0,066	0,052	0,067	0,155	0,03	0,042	0,02	0,143	0,042
	std beta																		
	P(accept) from t-values																		
	n																		
x8	mean beta	1	1,04	0,642	0,822	0,538	0,612	0,348	0,175	0,299	0,409	0,593	0,026	0,512	0,613	0,233	0,138	0,265	0,288
	std beta		0,732																
	P(accept) from t-values		0,3																
	n		1000																

Figure 22: Colour coded absolute deviances of the β estimates from the defined true βs. For example, the mean of the sample distribution for the regressor Duration ($\beta_{x_8} = 1$) when using Forward Stepwise and data carving is $\beta_{Duration}^{Fwd,Carving} = 1.538$. Then, the value in this figure equals $\left|\beta_{Duration} - \beta_{Duration}^{Fwd,Carving}\right| = 0.538$. Lower values equal better results in terms of the average accuracy of the fit. It can be seen that Data Carving is most accurate on average for Forward and Backward stepwise; whereas data splitting is on average most accurate when using Bidirectional stepwise. When using Lasso as a model selection algorithm, all data strategies except for adding noise decrease the accuracy of β estimates.

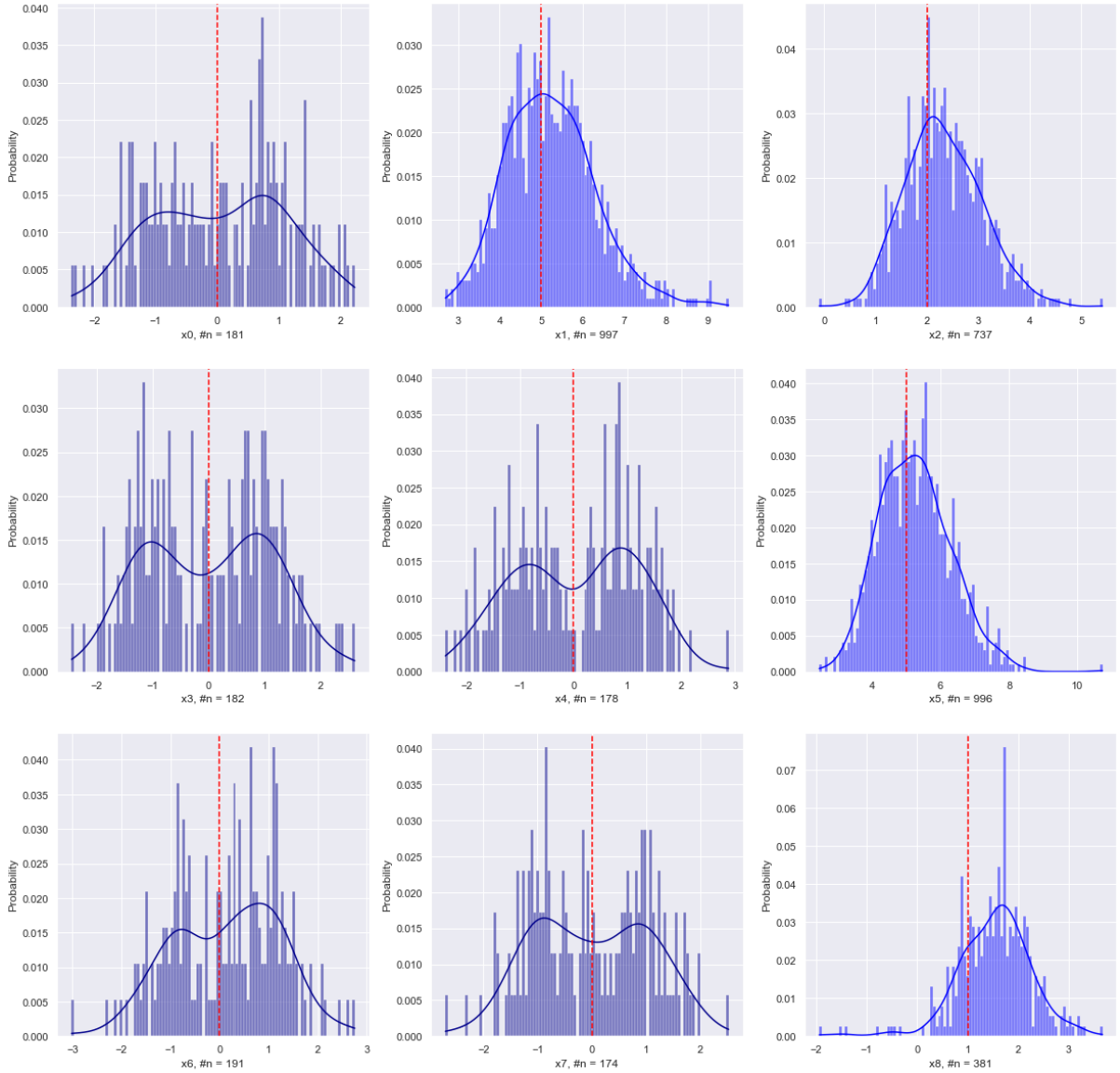

*Figure 23: Regressor parameter β sample distributions as a result of **carving** a generated data sample from a known model to use one half for **Forward** stepwise model selection and the complete sample for subsequent fitting. Observe how irrelevant regressors are still bimodally distributed, but to a lesser degree. Observe furthermore that relevant regressors who's true betas are relatively small $(\beta_{x_2} = 2 \; and \; \beta_{x_8} = 1)$ are selected in a model less often, more accurately reflecting their relative impact on driving PDs.*

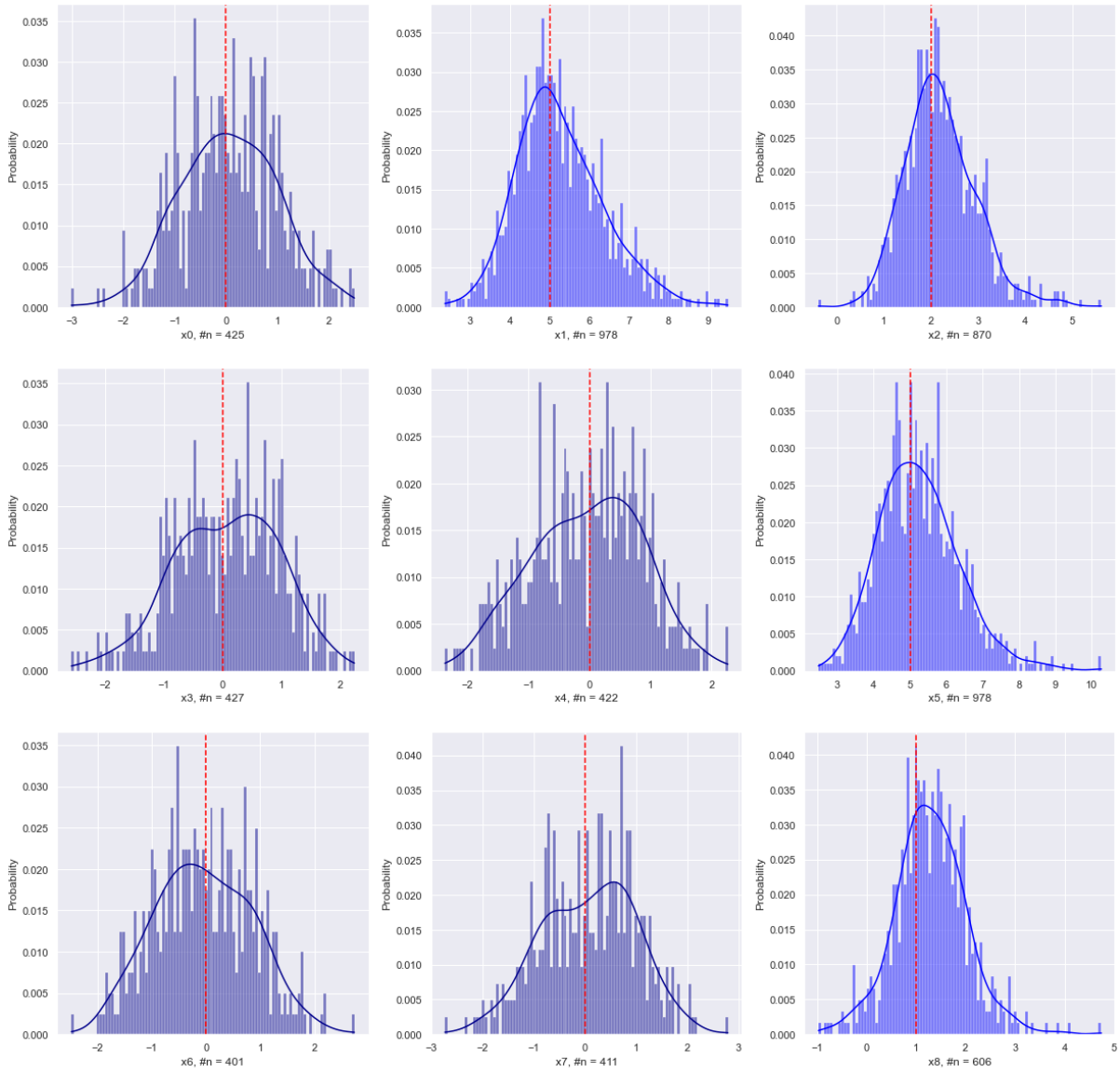

*Figure 24: Regressor parameter β sample distributions as a result of **carving** a generated data sample from a known model to use one half for **Backward** stepwise model selection and the complete sample for subsequent fitting. Observe how, similar to Forward stepwise, irrelevant regressors are still bimodally distributed, but to a significantly lesser degree. Furthermore, it must be noted that irrelevant regressors get selected in a model more often, which may increase the probability of accepting them in a final model.*

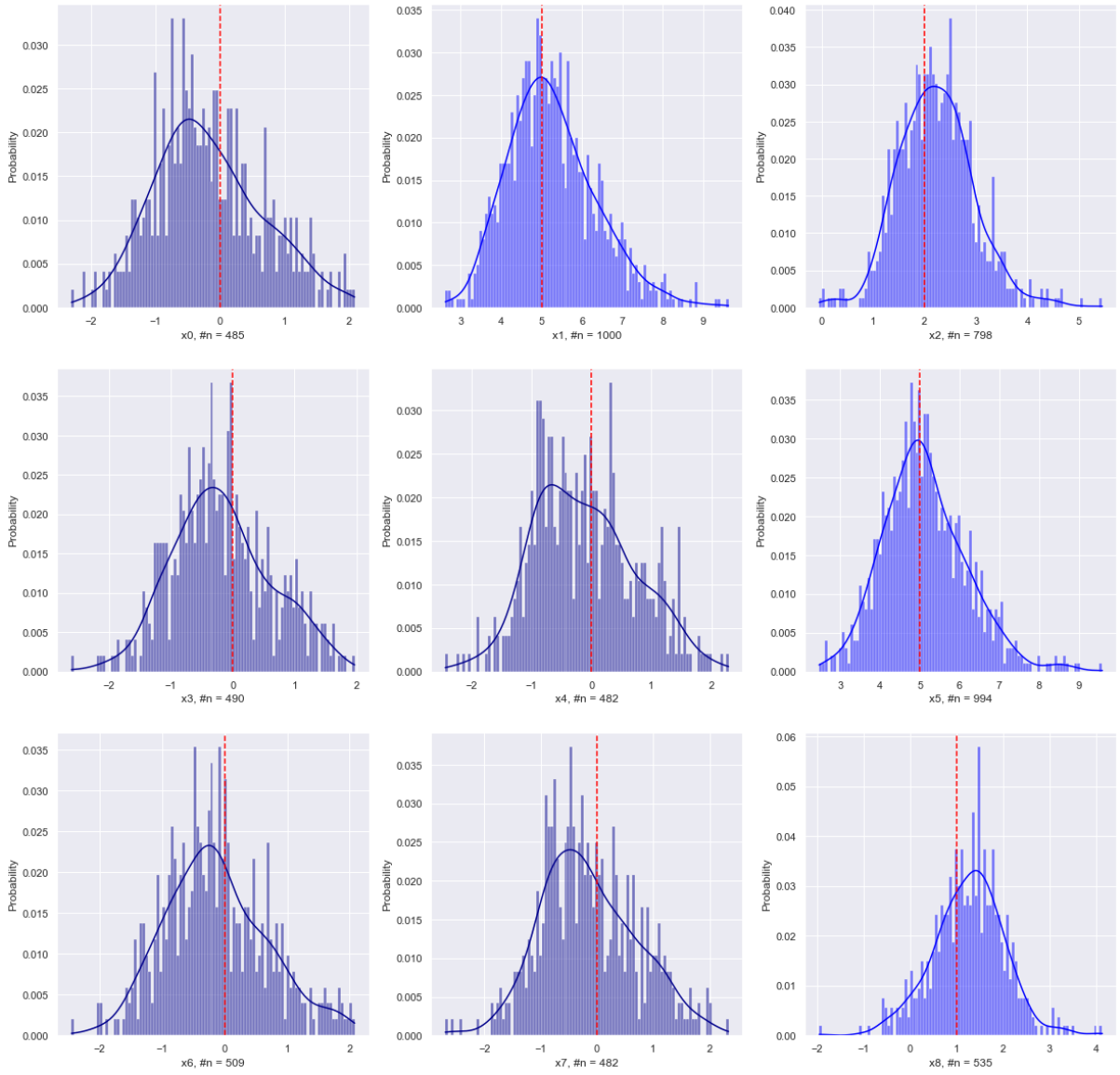

*Figure 25: Results of **carving** the data sample in combination with using **Lasso** as a model selection algorithm. Similar to Backward stepwise, but different from Forward stepwise, irrelevant regressors are selected relatively often; with the mode of these sample distributions centred around the true value of* 0.

5.2.4 Adding Noise

Literature analysed in chapter 2 showed us that adding noise is expected to have comparable results to data carving in terms of a final (fitted) model being able to distinguish between true and false instances. Our inference results (histograms), however, show that when adding approximately 8% of noise to the generated data during the model selection stage all stepwise results are highly comparable to the stepwise results of using no data strategy. So, the irrelevant regressors still have bimodal *t*-value sample distributions and a relatively high risk of getting accepted after fitting the selected model.

5.3 t-Value Acceptance Rate

Figure 26 shows how the results of different model selection algorithms and data strategies deviate from directly fitting the known model to data. It can be seen that, for stepwise algorithms, data splitting most accurately, on average, estimates the power of regressors based on empirical p-value acceptance rates followed by data carving. This comes at the cost of these strategies selecting relevant regressors in a model less often. This trade-off, for modellers, means that the probability of selecting the correct model is linked to less sparse model selection results.

So, a modeller that uses data splitting or data carving has a lower probability of selecting the complete correct model, but the selected regressors have accurate t-values. Or, a modeller that does not use data splitting has a higher probability of selecting the correct model, but has a greater probability of adding irrelevant noise-regressors to the model.

To better illustrate this phenomenon, observe the t-value sample distributions as a result of Bidirectional stepwise without a data strategy (Figure 27) and combined with data splitting (Figure 28). When we split the data, irrelevant regressors are included in a model 13.8% more often on average, but they are accepted based on their p-values -87.1% less often on average. Relevant regressors, on the contrary, are included in a model -11.1% less often on average when splitting the data, and their average acceptance rate decreased with -22.5%. Similar results can be achieved with Forward stepwise, where the only notable difference is that irrelevant regressors are slightly less frequently included in a model when data splitting compared to no data strategy.

Decreasing the probability of selecting and accepting relevant regressors is not per se problematic, as long as the true power of regressors is accurately reflected in the results, e.g. without model selection the relevant regressor $x8$ is only accepted after a fit 30% of the time. Data carving combined with any model selection algorithm selects, similar to data splitting, relevant regressors less often in a model but accepts these after a subsequent fit more often. This may be due to data carving using the complete sample during fitting.

Regarding Lasso, it seems to perform better without a data strategy in terms of both accurate betas and acceptance rates. That may be due to Lasso being a penalty in itself, resulting in too much restrictive use of the data when combined with any data strategy. Lasso in itself, however, results in relatively un-sparse model selection, which may be due to how the value of λ is determined, which is discussed in the next chapter.

				Deviation from direct fit															
	Model Selection Algorithm:	**Defined**	**None**	**Forward Stepwise**				**Backward Stepwise**				**Bidirectional Stepwise**				**Lasso (lambda 1SE)**			
Regressor	*strategy:*	*Beta Vector*	*Direct Fit*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*	*None*	*Splitting*	*Carving*	*Noise*
x0	mean beta	0																	
	std beta																		
	P(accept) from t-values			0,35	0,06	0,19	0,3	0,13	0,05	0,11	0,1	0,3	0,03	0,18	0,28	0,05	0,04	0,09	0,09
	n			161	167	181	169	421	436	425	390	154	190	149	170	705	492	485	574
x1	mean beta	5	5,26																
	std beta		1,063																
	P(accept) from t-values		1	0	0,01	0	0	0	0,01	0	0	0	0,01	0	0	0	0	0	0
	n		1000	0	1	3	0	0	27	22	8	0	3	1	0	0	2	0	0
x2	mean beta	2	2,087																
	std beta		0,796																
	P(accept) from t-values		0,82	0,06	0,28	0,08	0,08	0,03	0,32	0,05	0,04	0,07	0,28	0,09	0,07	0,03	0,32	0,08	0,01
	n		1000	76	258	263	77	12	126	130	34	69	275	291	65	17	193	202	21
x3	mean beta	0																	
	std beta																		
	P(accept) from t-values			0,26	0,08	0,18	0,38	0,16	0,04	0,11	0,12	0,3	0,07	0,18	0,37	0,07	0,05	0,08	0,07
	n			153	173	182	168	416	436	427	412	165	181	184	161	707	485	490	606
x4	mean beta	0																	
	std beta																		
	P(accept) from t-values			0,33	0,07	0,24	0,27	0,14	0,05	0,11	0,15	0,4	0,04	0,19	0,31	0,06	0,04	0,08	0,07
	n			181	149	178	155	403	411	422	383	172	180	174	156	684	512	482	592
x5	mean beta	5	5,222																
	std beta		1,114																
	P(accept) from t-values		1	0	0,01	0	0	0	0,01	0	0	0	0	0	0	0	0,01	0	0
	n		1000	0	1	4	0	0	27	13	8	0	2	1	0	0	3	6	0
x6	mean beta	0																	
	std beta																		
	P(accept) from t-values			0,32	0,07	0,19	0,36	0,14	0,04	0,11	0,12	0,32	0,05	0,15	0,32	0,08	0,05	0,09	0,08
	n			170	161	191	156	411	408	401	400	165	198	178	160	703	480	509	582
x7	mean beta	0																	
	std beta																		
	P(accept) from t-values			0,31	0,05	0,18	0,35	0,13	0,05	0,1	0,11	0,38	0,03	0,22	0,31	0,07	0,05	0,09	0,08
	n			175	157	174	155	419	419	411	399	178	200	177	170	684	498	482	616
x8	mean beta	1	1,04																
	std beta		0,732																
	P(accept) from t-values		0,3	0,33	0,12	0,33	0,33	0,09	0,12	0,13	0,16	0,24	0,17	0,27	0,33	0,07	0,13	0,15	0,11
	n		1000	448	618	619	490	252	394	394	277	476	648	631	482	148	463	465	232

Figure 26: Colour-coded absolute deviances of the number of times a regressor is present in a model and the acceptance rate compared to the sample distribution of a direct fit of the known model. For example, fitting the known model on 1000 samples of size 700 resulted in the regressor $x8$ with $p_{Duration} \leq 0.05$ a fraction 0.3 of the time. When using Forward Stepwise with Data Splitting, this fraction was 0.18 after fitting. The value in the table hence equals $|0.3 - 0.18| = 0.12$.

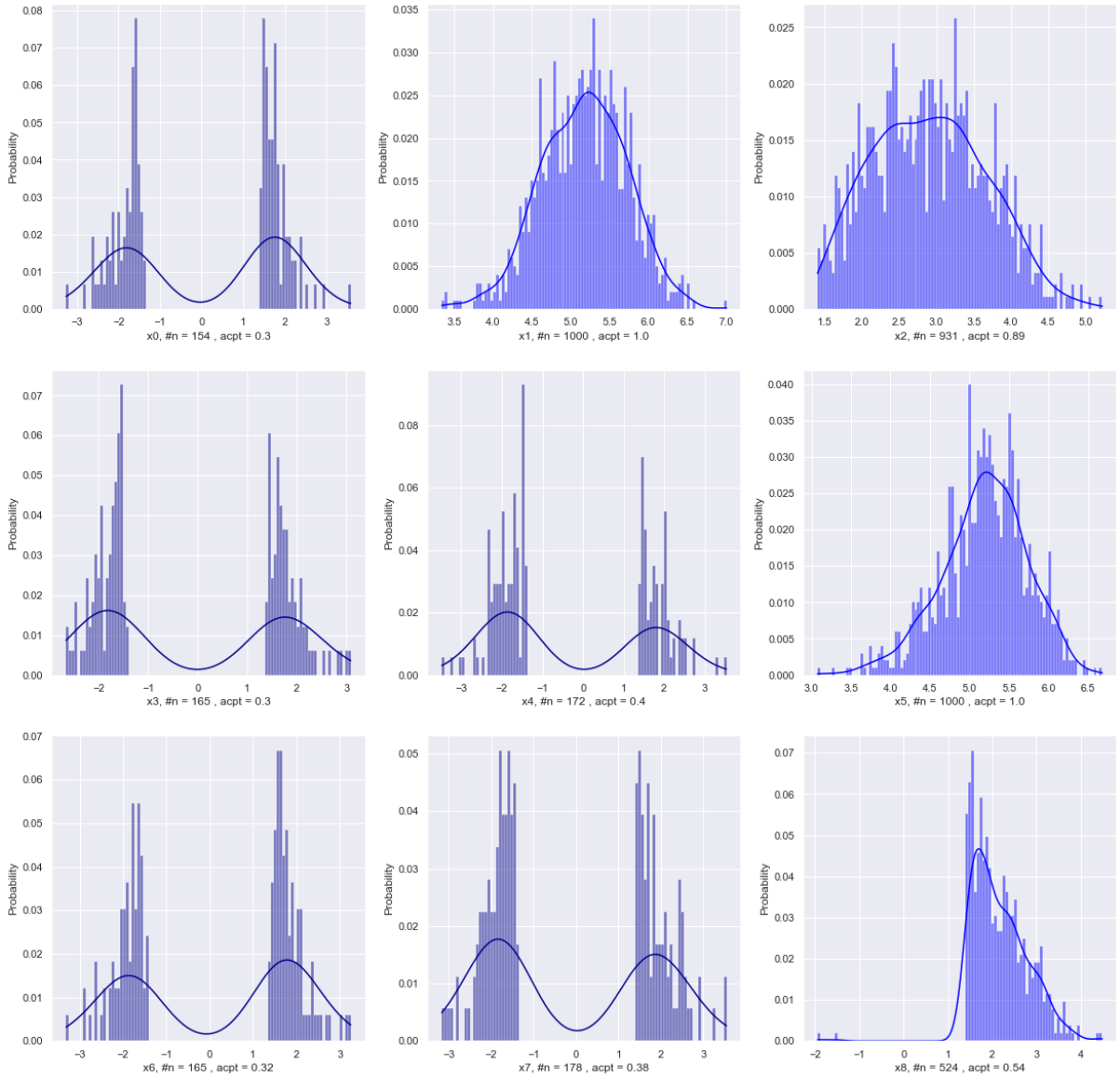

*Figure 27: t-Value sample distributions as a result of using a generated data sample for both **Bidirectional** stepwise model selection and subsequent fitting. Observe how irrelevant regressors $(x_0, x_3, x_4, x_6, x_7)$ get selected in a model, and subsequently accepted relatively often.*

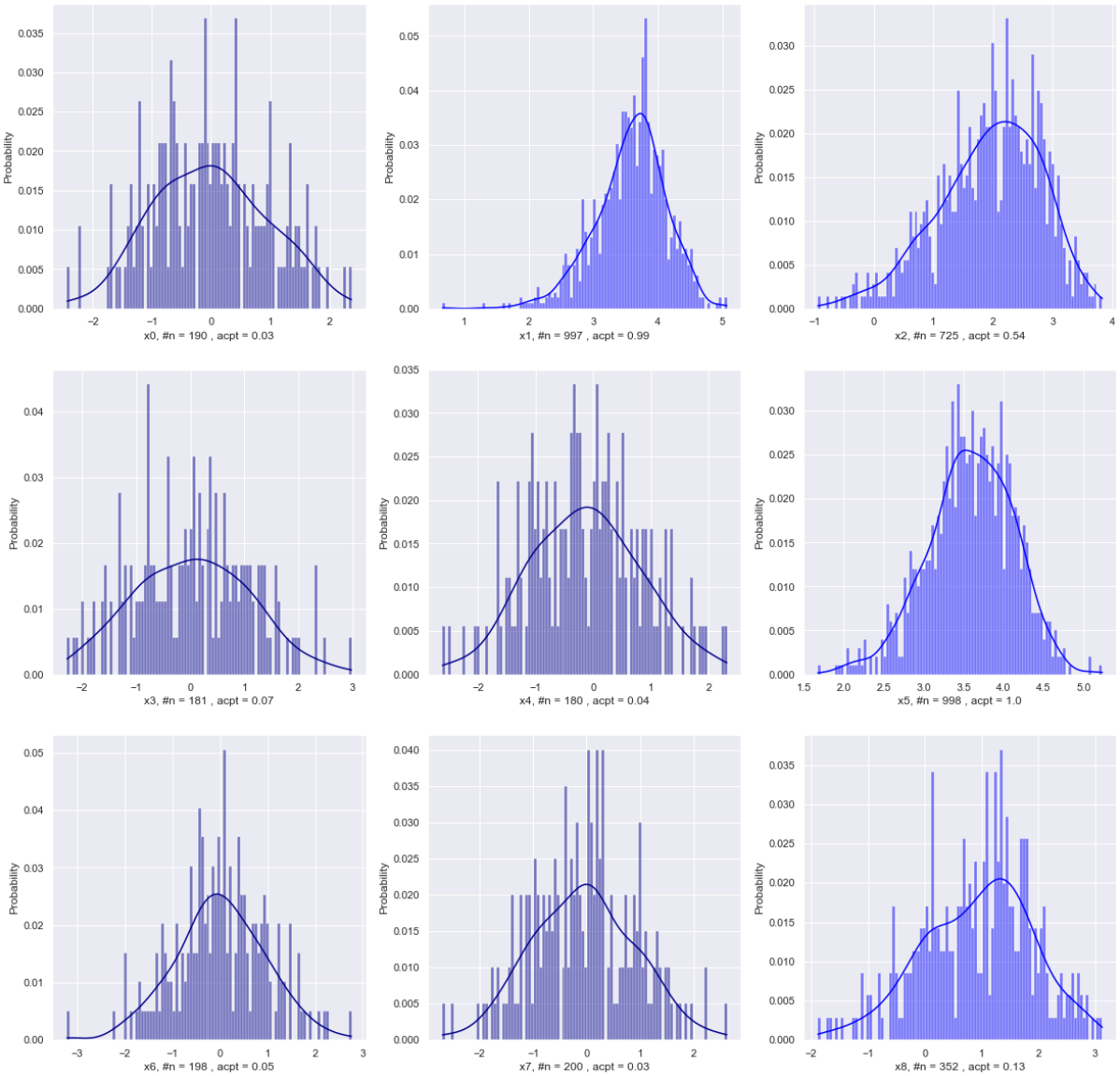

*Figure 28: t-Value sample distributions as a result of **splitting** a generated data sample and use one half for **Bidirectional** stepwise model selection and the other for subsequent fitting. Observe how irrelevant regressors $(x_0, x_3, x_4, x_6, x_7)$ get selected in a model more often than without a data strategy, but get accepted after the secondary fit -87.06% less frequently.*

5.4 Valid Performance

We performed experiments where random data X was generated with number of rows $N \in [500, 1000, 2000]$ and the number of regressors $k \in [5,10,15]$. Forward, bidirectional, and backward stepwise as well as lasso with fixed λ ($\lambda = 0.1$) model selection algorithms were performed in combination with no data strategy, data splitting, and data carving. During the fitting stage each fitted model is tested on a left out fold, and this test AUC is stored 5 times, as we perform 5-fold CV, and the average is stored. The highest AUC is the result of one particular fit, this maximum AUC is stored as well and the corresponding fit is tested on new data with size $0.25N$. This test AUC is also stored. Results were presented in the previous chapter, which we now analyse.

We use the fraction of computed AUC values greater than a set threshold as a metric for bias. For example, random data $N = 1000$ with $k = 10$ is generated one thousand times, and the same number of times model selection and model fitting according to some algorithm and data strategy are performed. Average AUC over 5 folds is hence computed a thousand times, and the metric for bias is the number of times this average $AUC \geq Threshold$ divided by one thousand.

5.4.1 Test AUC versus Average AUC

The results (Figure 14) indicate that using a separate test set to compute the test AUC of the best fit that occurred during model fitting is generally a riskier method of assessing true performance of a model compared to drawing conclusions based on the average AUC over the folds during model fitting. However, Rabobank must settle on a final model with fixed regressor parameters when a model is to be deployed, and, clearly, the average performance of five different fits is not consistent with one model whereas selecting and testing the best fit is.

Therefore a case can be made for using average AUC over the cross validation folds to get an idea of how well the data, in general, can be used to distinguish between good and bad borrowers. Data **carving** has the lowest risk, and any stepwise algorithm can be used to assess this as differences in inflated performance across stepwise algorithms are negligible. Then, when it is concluded that the data is suitable for predictive modelling, a final model may be selected using data splitting combined with a stepwise model selection algorithm, because the inference results show that data splitting yields the most valid model.

5.4.2 Data Size and Number of Regressors

The size of the data set and the number of regressors present in the data set increase the risk of inflated performance. From Figure 14 it can be seen from the red cells, red cells indicate a risk value of greater than or equal to 1%, that, no matter the type of AUC metric, the risk of inflated model performance decreases as the size of the data set increases. As for the number of regressors in X, the average of all output for 5 regressors equals 0.054, for 10 regressors it equals 0.058 which is an increase of 6.86%, and for 15 regressors the average output equals 0.060 which is a 3.57% and 10.67% increase compared to 10 and 5 regressor respectively. Concluding, increasing the number of regressors in a data set increases the probability of inflated model performance on average.

5.4.3 Lasso

The most notable results from using lasso as a model selection algorithm on data that is just noise, is that for data carving and not using a data strategy the lasso solver did not find a solution. That is, a solution to the objective $\min_{\boldsymbol{\beta}} -\mathcal{L}(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|$ was not found. Most likely this is due to the data being noise: no patterns could be recognized. In terms of reducing the risk of adding noise to a model to inflate its performance, the lasso test AUC is consistently higher than the stepwise

algorithms for each data strategy. It is up for discussion on whether lasso and stepwise are comparable due to the fewer solutions lasso resulted in.

5.4.4 Data Splitting

Data splitting, contrary to data carving, has the highest inflated average AUC for each data strategy modelled. Therefore it can be argued that data splitting should not be used when modellers want to gain understanding in how well the data may be used for predictive modelling. The test AUC results (Figure 14), however, do not appear to significantly deviate from using no data strategy or data carving: meaning that any data strategy does not magically increases a model's performance on new data. Taking into consideration the inference results, where both data splitting and data carving performed well, it is unclear which strategy to use when assessing performance; and further research is performed in the next chapter (Chapter 6) to optimize hyperparameters on which fraction of the data to use for model selection when either splitting or carving the data.

5.5 Concluding Chapter 5

In this chapter we analysed the result of the experiments that were conducted to quantify bias. Different combinations of model selection algorithms and data strategies were simulated and analysed with the goal of identifying and quantifying biased PD model statistics.

We identified that, as literature indicated, biased inference can occur as a result of using the same data sample for both model selection and model fitting. The way this bias manifests in the results is via bimodal sample distributions of both regressor parameters β and t-values, resulting in irrelevant regressors that do not contribute to PD getting selected and accepted in a model. Data splitting can be, combined with any stepwise model selection algorithm, a good solution in reducing the probability of accepting irrelevant regressors in a model.

Data splitting, however, decreases the prevalence of relevant regressors in selected models, and reduces their acceptance rates after fitting. This does not have to be a problem, as, naturally, regressors can have less power and should in that case be accepted less frequently. When the goal is to most accurately select the correct model one may opt for using data carving, as acceptance ratios of relevant regressors are not significantly decreasing compared to not using a data strategy. This can be a result of data carving using all data during model fitting, which increases the probability of relevant regressors being identified. All taken together, further analysis is required in the next chapter where we compare data splitting with data carving for different configurations regarding hyperparameters.

From the results of the prediction experiments (Figure 14), inflated model performance can be a problem depending on the size of the data and on the number of regressors present in the data. It is therefore advised to gain insight into the usefulness of the data for predictive modelling by analysing the data by building and fitting models using either data splitting or data carving and 5-fold CV. Thereafter a final fitted model may be selected but, considering the inference results (Figure 11), it is unclear by which methods: either data splitting or data carving. To conclude on this matter requires further analysis in the next chapter. For now, if modellers deem the probability of inflated performance, listed in Figure 14, too high then modellers may retrieve more data or reduce the number of regressors in the data.

Concluding, it is advised to make use of data carving or data splitting when evaluating the quality of the data for predictive modelling. Actions against inflated performance are unnecessary when sufficient data is available, exact thresholds may be retrieved from Figure 14. Regarding inference, in

the next chapter both carving and splitting will be tuned on hyperparameters and compared on performance to see which one is best.

6. Validating and Tuning the Solutions

In this chapter we discuss well performing strategies that were identified in the analysis of chapter 5, to subsequently fine tune and stress test a final one. First, we recall the well performing strategies (section 6.1), then we see if these strategies are performing a certain way because of the data (section 6.2). Thereafter, we try to conclude on what fraction of a data sample to use for model selection (section 6.3), and if changing the way the penalty constant λ is determined yields better results regarding inference when using Lasso as a model selection algorithm (section 6.4). We thereafter use the German credit risk dataset where a known model is applied to its normalized data to see if with using real data distributions we still observe improvements when using well performing strategies; also, the natural risk of the German credit risk data set is used to see if the proposed solution yields a different model on average (section 6.5). Finally, we conclude this chapter by summarizing it (section 6.6).

6.1 Well Performing Strategies

In chapter 5 we have seen that the probability of accepting irrelevant noise-regressors in a model can be significantly reduced by using data splitting. For example, Forward stepwise combined with data splitting selects irrelevant regressors in a model 3.93% less often and accepts these in a model after a subsequent fit on new data 78.98% less often, compared to not using a data strategy. The general trend appears to be that data splitting has a similar probability of selecting irrelevant regressors in a model, but, because bimodal t-value sample distributions do not occur, accepts these after fitting the selected model much less frequently.

Data carving can, when using stepwise model selection algorithms, also decrease the probability of accepting irrelevant regressors in a model, but not as significantly as data splitting. This may be a result of carving using a fraction of the same data used for model selection in model fitting, in similar fashion to not using a data strategy. When using Lasso, data carving yields worse results both in terms of accurate βs and acceptance rates.

Backward stepwise selects irrelevant regressors in a model more frequently than Forward and Bidirectional stepwise. Combined with data splitting or data carving the probability of accepting these irrelevant regressors decreases. So, when sparse model selection is preferred, Backward stepwise should not be preferred, just as Lasso, compared to Forward or Bidirectional stepwise.

Concluding, the well-performing strategies of combining Forward or Bidirectional stepwise with either data splitting or data carving should be further investigated in the next section (section 6.2) on how to optimally split the data into a model selection (MS) fraction and a fitting fraction.

6.2 Optimal Splitting

We evaluated Forward and Bidirectional stepwise combined with the data strategies data splitting and data carving, where the model selection part was set to 25%, 50%, and 75%. From the results of these experiments it visually became clear that reducing the model selection fraction, i.e. from 50% to 25% yields better results than increasing the fraction. Hence, we study data splitting and data carving for both Forward and Bidirectional stepwise in more detail.

	Model Selection Algorithm:	**Defined**	**None**	**Forward**			**Bidirectional**		
Regressor	*strategy:*	*Beta Vector*	*Direct Fit*	*None*	*Splitting 25MS*	*Carving 25MS*	*None*	*Splitting 25MS*	*Carving 25MS*
x0	mean beta	0		0,033	0,044	0,064	0,064	0,126	-0,053
	std beta			1,406	0,836	0,863	1,366	0,855	0,697
	P(accept) from t-values			0,35	0,05	0,1	0,3	0,04	0,05
	n			161	154	156	154	136	146
x1	mean beta	5	5,26	5,28	5,141	5,153	5,25	5,182	5,044
	std beta		1,063	1,113	1,172	1,014	1,056	1,211	0,953
	P(accept) from t-values		1	1	1	1	1	1	1
	n		1000	1000	817	813	1000	814	815
x2	mean beta	2	2,087	2,232	2,02	2,336	2,209	2,092	2,303
	std beta		0,796	0,742	0,926	0,718	0,705	0,873	0,736
	P(accept) from t-values		0,82	0,88	0,67	0,91	0,89	0,7	0,92
	n		1000	924	437	420	931	431	436
x3	mean beta	0		-0,051	0,029	0,003	-0,062	0,037	-0,046
	std beta			1,367	0,822	0,913	1,382	0,816	0,849
	P(accept) from t-values			0,26	0,04	0,1	0,3	0,05	0,1
	n			153	155	152	165	169	129
x4	mean beta	0		0,103	0,072	0,045	-0,197	0,009	-0,16
	std beta			1,383	0,882	0,902	1,425	0,891	0,851
	P(accept) from t-values			0,33	0,08	0,11	0,4	0,06	0,12
	n			181	129	152	172	156	149
x5	mean beta	5	5,222	5,351	5,104	5,083	5,291	5,177	5,061
	std beta		1,114	1,134	1,165	1,012	1,092	1,238	0,995
	P(accept) from t-values		1	1	1	1	1	1	1
	n		1000	1000	811	814	1000	812	808
x6	mean beta	0		0,002	0,041	0,154	0,11	-0,02	-0,03
	std beta			1,403	0,851	0,875	1,399	0,813	0,822
	P(accept) from t-values			0,32	0,05	0,1	0,32	0,02	0,06
	n			170	145	168	165	150	160
x7	mean beta	0		0,026	-0,017	0,076	-0,052	0,036	0,069
	std beta			1,387	0,749	0,908	1,478	0,753	0,819
	P(accept) from t-values			0,31	0,03	0,13	0,38	0,05	0,1
	n			175	135	142	178	152	155
x8	mean beta	1	1,04	1,642	1,015	1,387	1,593	0,889	1,368
	std beta		0,732	0,533	0,961	0,687	0,54	0,926	0,739
	P(accept) from t-values		0,3	0,63	0,26	0,53	0,54	0,19	0,5
	n		1000	552	227	225	524	209	223

Figure 29: Numerical results of tuning the fraction of a data sample that should ideally be used for model selection. It can be seen that Forward stepwise combined with data splitting $25\% MS$ is optimal in terms of average β accuracy and accurately, on average, assessing the true power of regressors after fitting the selected model. Standard deviations of β estimates can be slightly reduced when using Bidirectional stepwise and data carving, but at the cost of accepting irrelevant regressors in a model significantly more often.

Figure 29 shows the results for Forward and Bidirectional stepwise combined with data splitting and data carving. The best results in terms of accurate average β estimates and accurate assessment of a regressor's true power are achieved when using Forward stepwise combined with data splitting where 25% of a data sample is used for model selection, and 75% for fitting the selected model. From Figure 30, sample distributions of t-values for each regressor, we can confirm these results: irrelevant regressors get selected and accepted in a model significantly less often compared to not using a data strategy. Furthermore, the natural power of relevant regressors is accurately described by the t-value histograms: we observe different results across relevant regressors depending on the value used for the true βs.

In terms of absolute improvements compared to the baseline strategy of using a data sample for both Forward stepwise model selection and model fitting, Forward stepwise combined with data splitting $(25\% \, MS)$ selects irrelevant regressors 15% less often in a model and subsequently accepts these 86% less often after fitting the selected model. This comes at the cost of relevant regressors getting selected 34% less often and accepting these 17% less often; it must, however, be taken into consideration that this cost is not necessarily bad, but rather the result of a more accurate power estimate as not every regressor contributes to model performance equally.

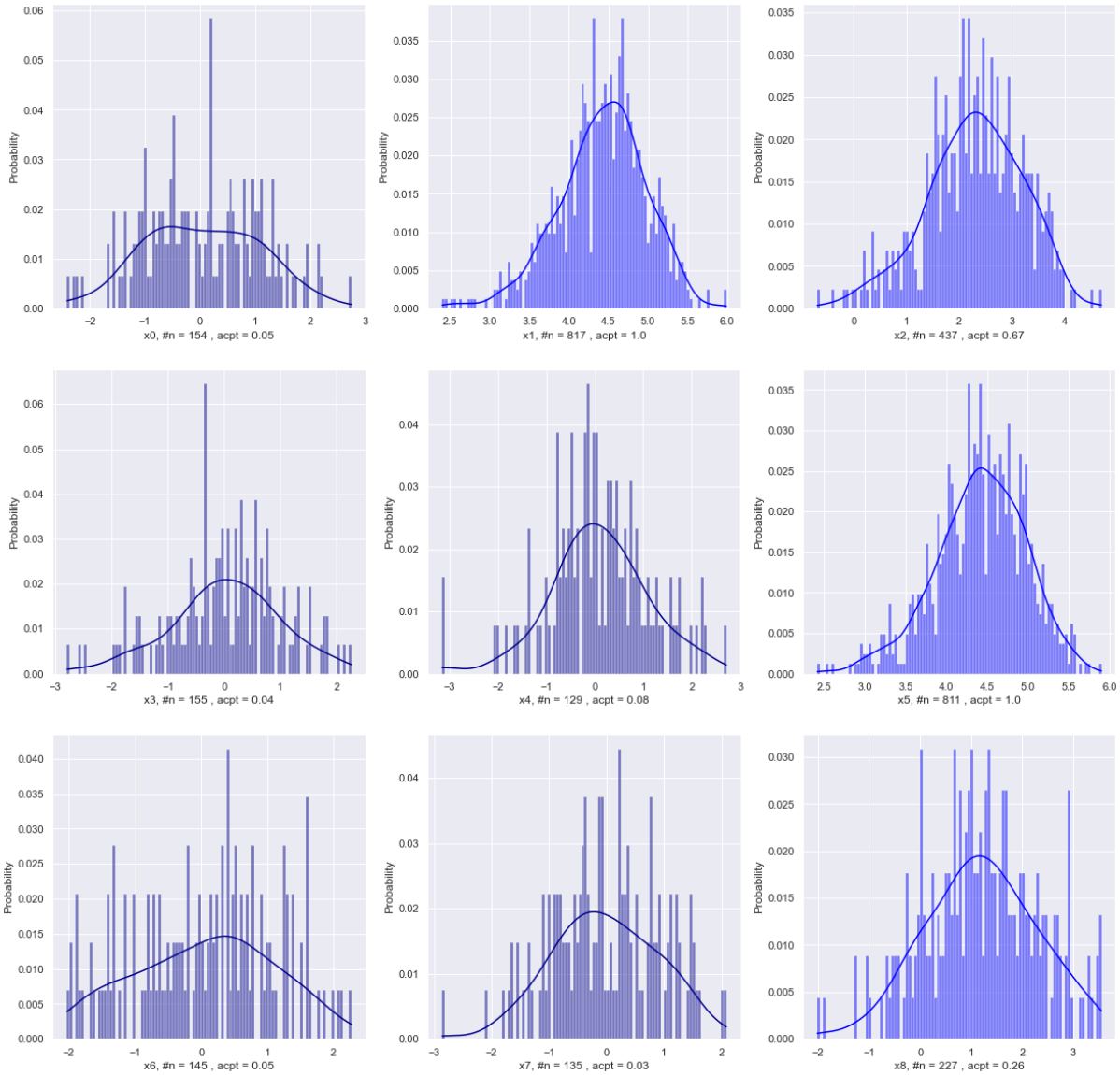

Figure 30: t-value sample distributions as a result of using Forward stepwise as a model selection algorithm where generated data samples are split into a model selection (MS) fraction with a size of 25% *of the generated data sample, and the remainder used for fitting the selected model. Compared to not using a data strategy, all irrelevant regressors (**grey-blue**) are selected in a model* 15% *less often and accepted after fitting the selected model* 86% *less often. The relevant regressors (**blue**) are selected in a model* 34% *less often and accepted after fitting* 17% *less often; this is not per se a bad result, as the natural power of regressors in a known model logically depends on the value of their corresponding* β*. I.e., relevant regressors with relatively smaller true* β*s should get selected and accepted in a model less often.*

6.3 The Data

To see if the proposed solution of using Forward stepwise combined with data splitting $25\%MS$ not only performs well on average on different samples, but also on a random sample generated from a known true model, we generate a sample from the known model and bootstrap from it. The results can also be used to see how well the proposed solution can recognize a true model using bootstrapping. On the bootstrapped sample model selection and fitting using the promising solution of Forward stepwise combined with $25\%MS$ data splitting is used.

Figure 31 shows t-value sample distributions of the proposed solution, and Figure 32 of the baseline of using the same sample for both Forward stepwise model selection and subsequent fitting. It can be seen that the proposed solution of Forward stepwise and data splitting $25\%MS$ has a significantly lower risk of selecting and accepting irrelevant regressors in a model. This, however, does happen at the cost of less often selecting and accepting relevant regressors in a model. This cost, as discussed, is not necessarily problematic as the prevalence and acceptance rates of regressors should describe the true power of regressors, which may vary between relevant regressors depending on relative β sizes.

Concluding, the improvements as a result of using the proposed solution of Forward stepwise $25\%MS$ are likely to not be on average, but are expected to be prevalent across samples. Furthermore, bootstrapping combined with the proposed solution can more accurately identify the correct model responsible for generating the data sample, compared to the baseline of using the same data for model selection and fitting.

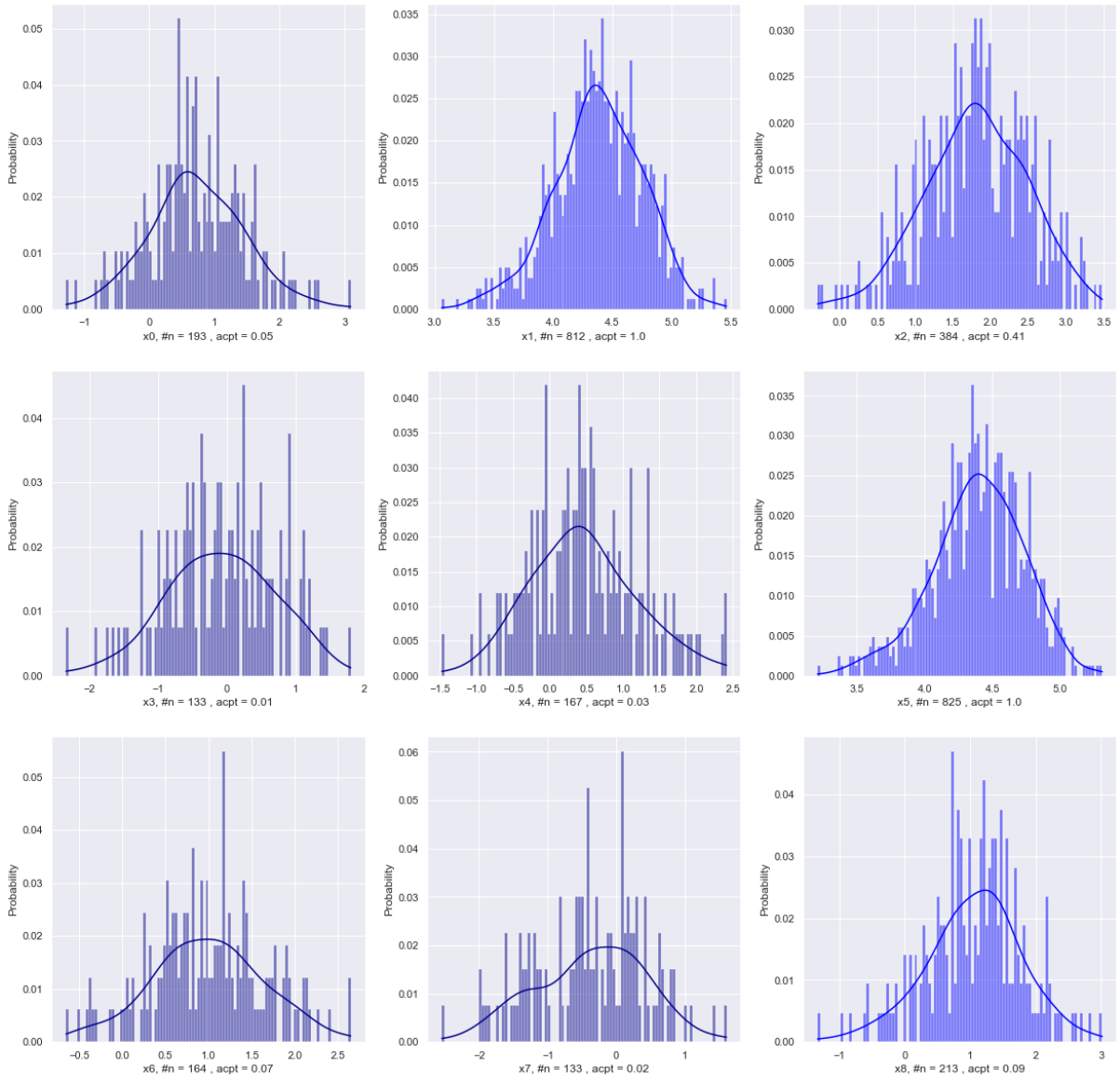

*Figure 31: t-value sample distributions as a result of using the proposed solution (**Forward** stepwise combined with data **splitting** $25\%MS$). Observe how the risk of selecting and accepting irrelevant regressors is still relatively low. Relevant regressor $x8$ with a true parameter of $\beta_{x_8} = 1$, however, is significantly less often accepted after fitting. Compared to the baseline of using identical samples for both Forward stepwise model selection and subsequent fitting (Figure 31), splitting yields significant improvements.*

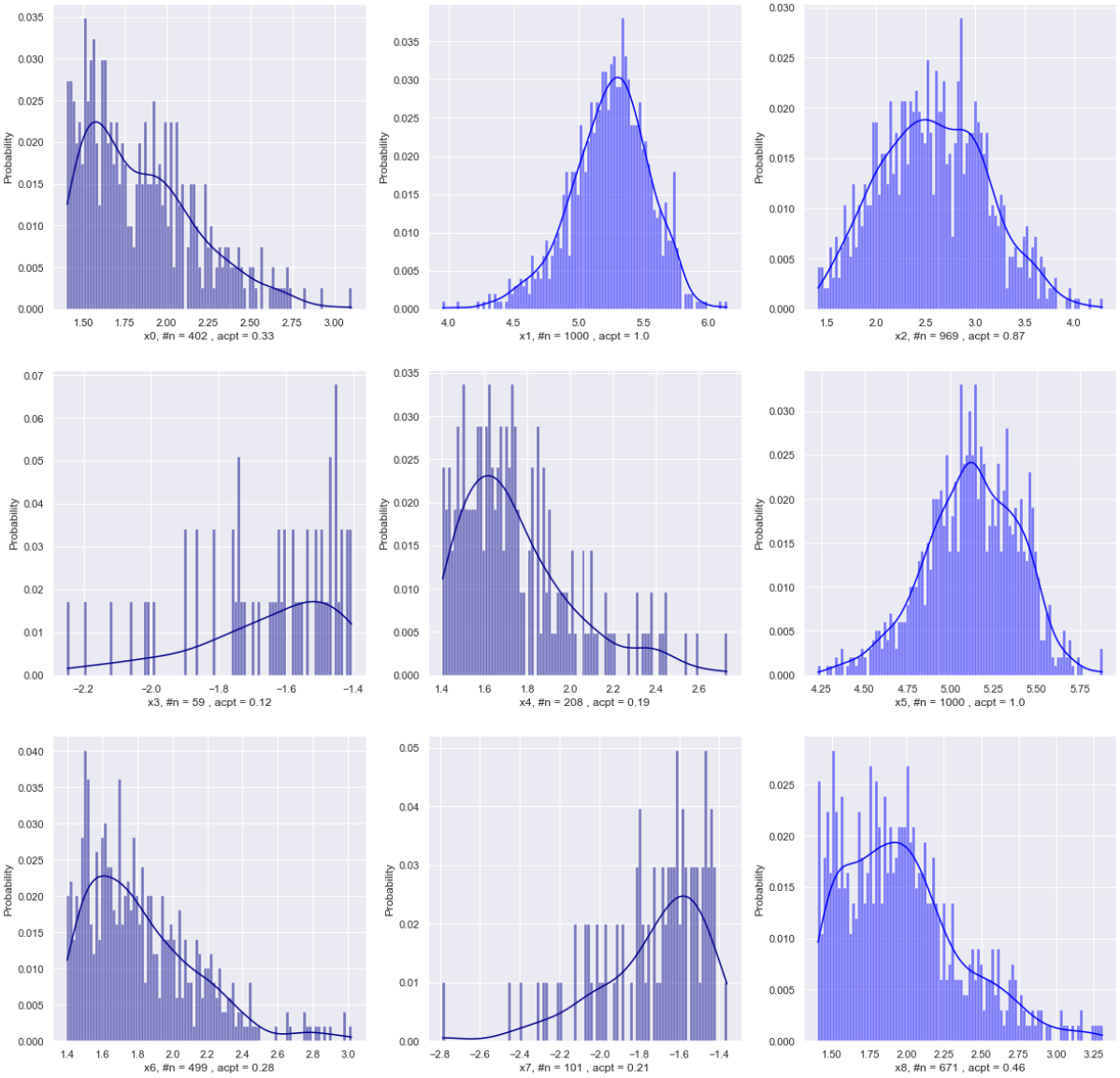

*Figure 32: Baseline results of using the same bootstrapped sample for **Forward** stepwise model selection and subsequent fitting to compare with the proposed solution in Figure 31. Observe how there is significant risk of selecting and accepting several irrelevant regressors in a model.*

6.4 Optimal Lambda using AUC

The Lasso inference results presented in chapter 4 and chapter 5 are not sparse enough. That is, irrelevant regressors get selected in a model rather often. It is expected that increasing the penalty constant λ will yield more sparse results. Initially, we used the log likelihood function value in a cross validation experiment to determine the optimal penalty constant λ on every generated data sample. Now, we will find the optimal λ value using out-of-sample AUC performance.

The experiment is set up as follows:

- A data sample is generated from the known model;
- Using this data sample, an optimal penalty constant λ^*_{1SE} is computed in the range $[2, 2.1, \dots, 4]$ using the **1 Standard Error Heuristic** (1SE) (Figure 33);
- Using λ^*_{1SE}, a Lasso regularized logit fit is performed as a model selection algorithm;
- All regressors for which their regressor parameter β is not equal to zero are selected in the model, which is subsequently fit.

The 1SE heuristic works by first splitting the generated data sample in 5 equally sized folds. Secondly, given a λ, a model is fitted on 4 folds and tested on the left-out fold in terms of AUC performance. This can be done five times, as there are five folds that can be left out. The average performance is stored, as well as the standard deviations between the five performances. Then, λ is increased and the process is repeated. If the average performance corresponding to a greater λ is better than that of a smaller λ, the greater λ is accepted as a (temporary) global optimum. At some point it is expected that the out-of-sample average performance is worse than the global optimum, but a greater lambda is nonetheless accepted if its performance is within one standard error (1SE) of the global optimum. An example of how λ^*_{1SE} can be seen in Figure 33. For a discussion on this heuristic, see e.g. Chen & Yang (2021).

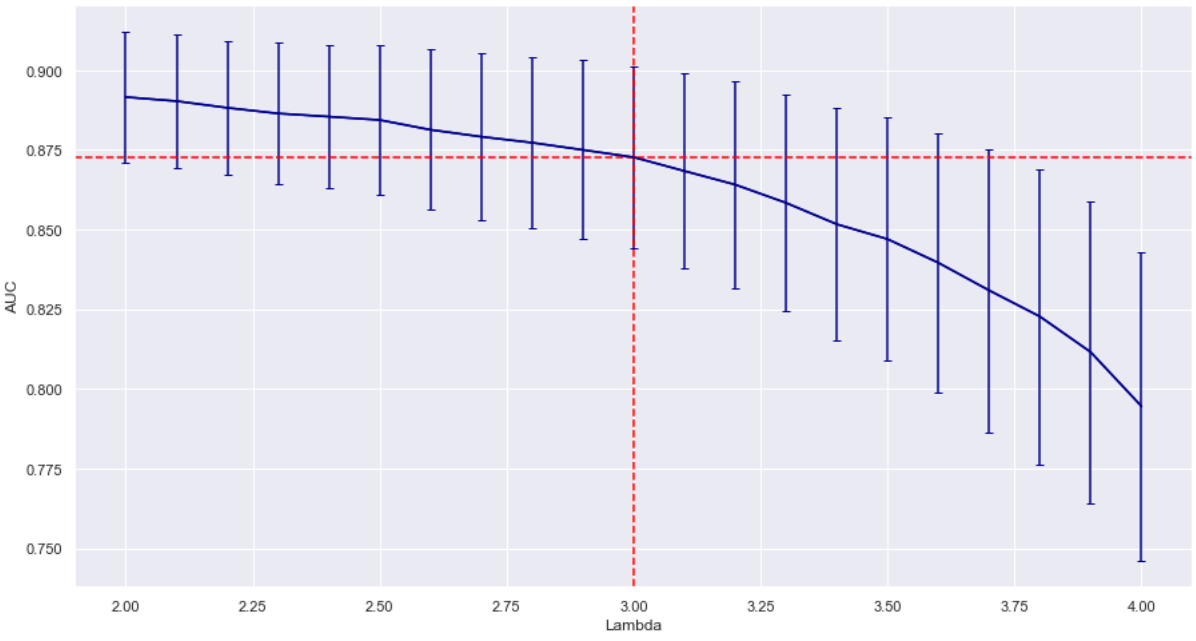

*Figure 33: An example of using the **1SE heuristic** to find an optimal penalty constant λ^*_{1SE}. Notice how the out-of-sample average AUC over five folds is greatest for $\lambda = 2$, but the, following the 1SE heuristic rules, we allow for increasing λ to 3 as its corresponding performance is within $1SE$ of the optimum at $\lambda = 2$.*

Compared to the original results (Figure 26), the new results of searching for an optimal λ where greater values are allowed indeed indicate more sparse model selection (Figure 34). Irrelevant regressors, however, have a relatively high acceptance rate after fitting the selected model in which they may be present. Compared to the current proposed solution of using Forward stepwise combined with 25% MS data splitting, the $1SE$ Lasso results are significantly worse. From these results we must conclude that using Lasso as a model selection algorithm is either not preferred for PD model selection, or requires more research on λ-tuning before put into practice.

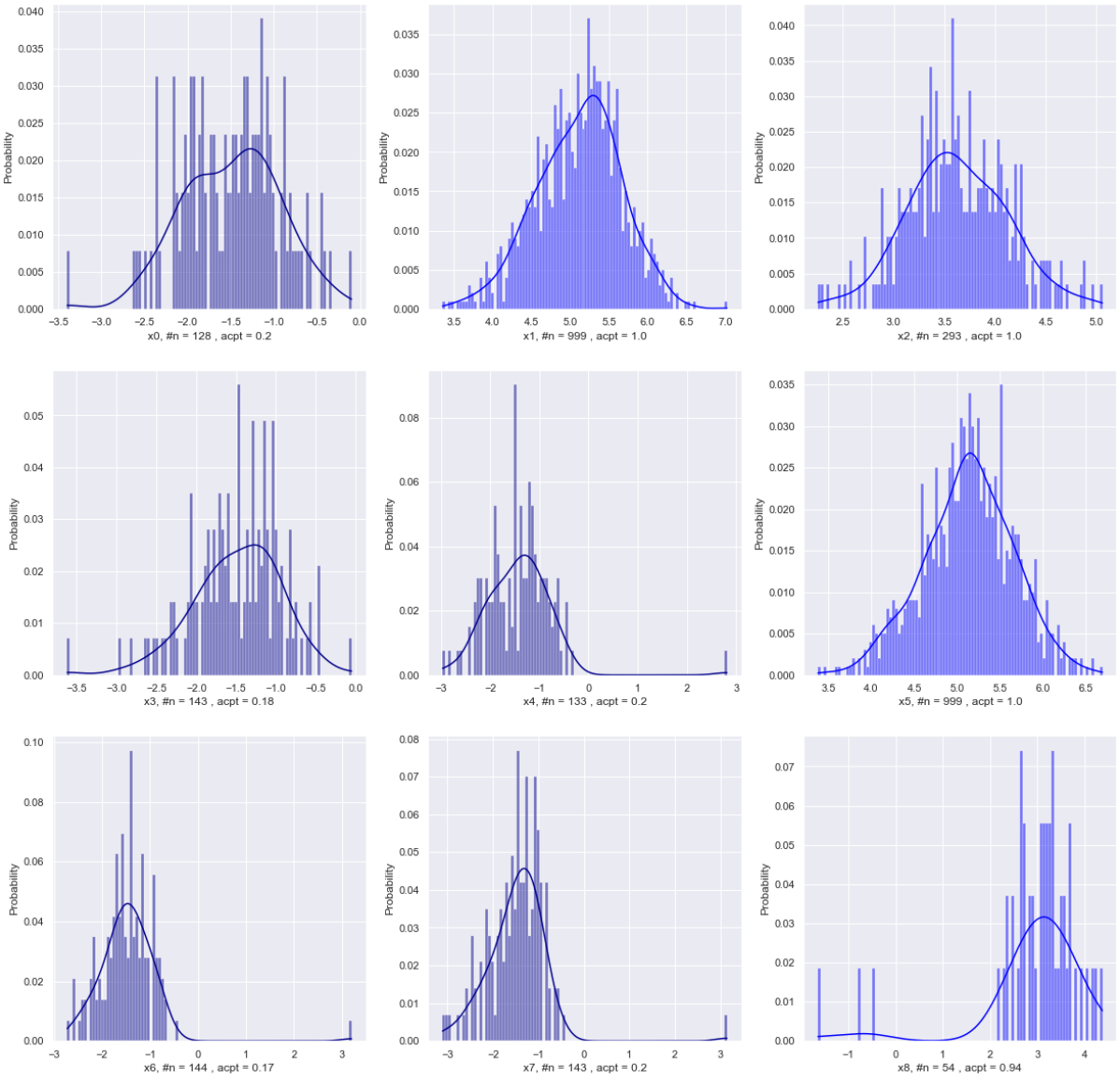

Figure 34: t-value sample distributions of using Lasso as a model selection algorithm. Irrelevant regressors, plotted in the darker shade of blue, are accepted after being selected in a model often and hence Lasso should not be preferred as a model selection algorithm over Forward stepwise.

6.5 German Credit Risk

In this section we compare results of using Forward stepwise without a data strategy to the thus-far proposed solution of using Forward stepwise combined with data splitting $25\%MS$ (section 6.5.1). Thereafter we discuss if the characteristics of these results are due to correlations within the data (section 6.5.2), value inflation factor values (section 6.5.3), or relative beta sizes of the known model (section 6.5.4). Lastly, we bootstrap from the real German credit risk data, i.e. the real defaults in this data, to see if, on average, the proposed solution selects a different model than the baseline of not using a data strategy.

6.5.1 Known Model

In the real world, we cannot choose the distribution of our data. That is, a regressors can be distributed following some empirical shape, this shape is beyond our control. To see if data splitting is as robust as to show improved results on natural data, we normalize the publicly available German credit risk[7] containing 1000 observations and apply a known model (Table 4) to this data to generate, in this case, 150 defaults. From this known-model German credit risk data we draw 1000 samples ($n_{sampleSize} = 700$) on which Forward stepwise without a data strategy as well as combined with data splitting is performed.

From the baseline results (Figure 35) it can be seen that irrelevant regressors are selected to a model infrequently, except for $CheckingAccount$. Furthermore, irrelevant regressor $CheckingAccount$ is accepted 22% of the selected cases after the subsequent fit. The proposed solution (Figure 36) more accurately describes the true power of irrelevant regressors. That is, there is significantly less risk, on average, of including irrelevant regressors in a model when using Forward stepwise combined with data splitting 25% MS. Concluding, Forward stepwise combined with data splitting 25% MS appears to yield better performance than not using a data strategy.

There are several factors that could affect the outcomes when using the natural distribution of regressors as input data to model selection, a few of which are: correlations within the data, the value inflation factor of regressors, and, in the case of using a known model, the size of the true βs used in that known model. These will be discussed in the next sections.

[7] https://www.kaggle.com/datasets/uciml/german-credit

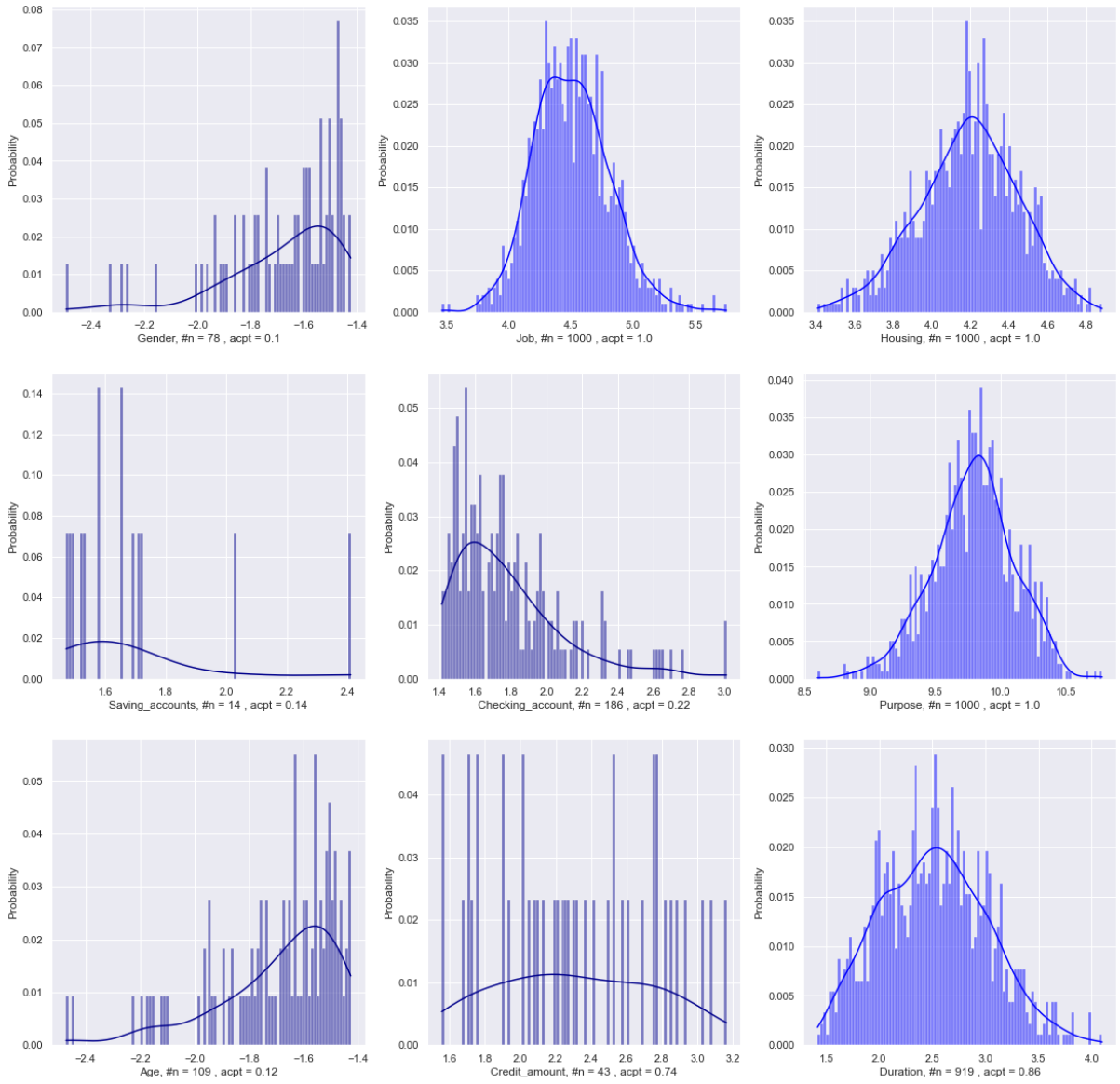

*Figure 35: t-value sample distributions as a result of using the same data sample, bootstrapped from normalized known-model German credit risk data, for **Forward** stepwise model selection and subsequent fitting.*

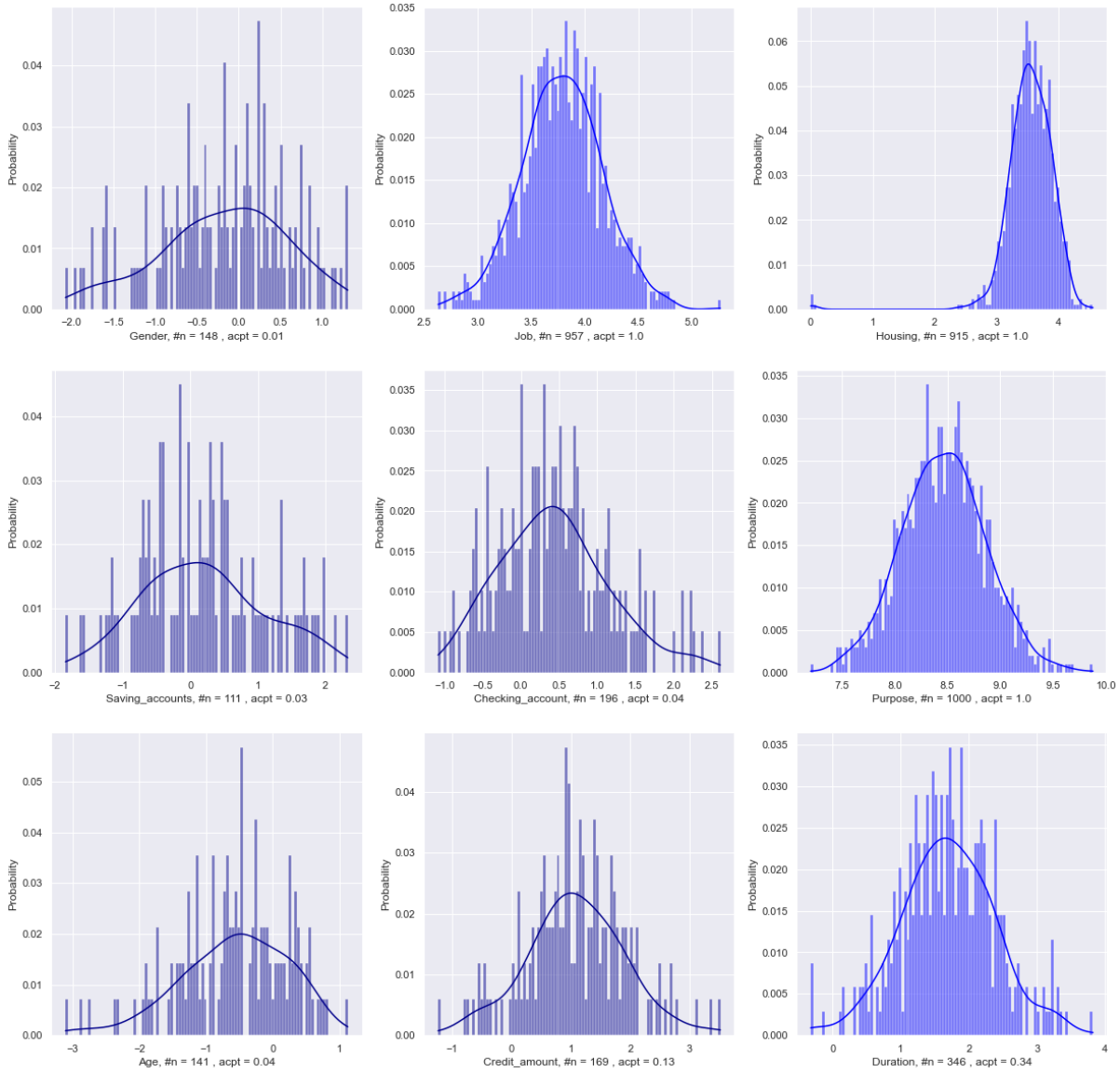

*Figure 36: t-value as a result of **splitting** a bootstrapped data sample in a model selection (MS) part* (25%) *and a model fitting part* (75%) *which are used for **Forward** stepwise model selection and subsequent fitting respectively. Observe how, compared to not using a data strategy (), regressors get selected more often but accepted significantly less often.*

6.5.2 Correlation

One reason for relevant regressors not getting selected in a model is that they may be heavily correlated to one or more irrelevant regressors. Then, instead of selecting the regressor that is responsible for generating defaults, regressors that are correlated to that relevant regressor may get selected. An example is given on correlated data (Figure 37). When a column containing relevant regressor data, in a particular order, is strongly correlated to a column containing data from an irrelevant regressor, the absence of the relevant regressor is less problematic when the strongly correlated irrelevant regressor has taken its place.

Figure 37: An example of correlated data. On the left two series of uncorrelated data. On the right the generated data series are strongly positively correlated.

The correlation matrix of the German credit risk data is given in Figure 38. It can be seen that the largest correlation is the correlation ($\rho = 0.63$) between Duration, a relevant regressor, and Credit Amount, an irrelevant regressor. When we look at the t-value inference results of Forward stepwise (Figure 36) combined with data splitting we can observe that $Duration$ is selection 346 times and $CreditAmount$ is selected 1000 times. The other correlations are negligible, and their corresponding regressors are selected infrequently. This leads us to conclude that correlation structure of the data used is most likely not problematic, i.e. it does not invalidate the inference

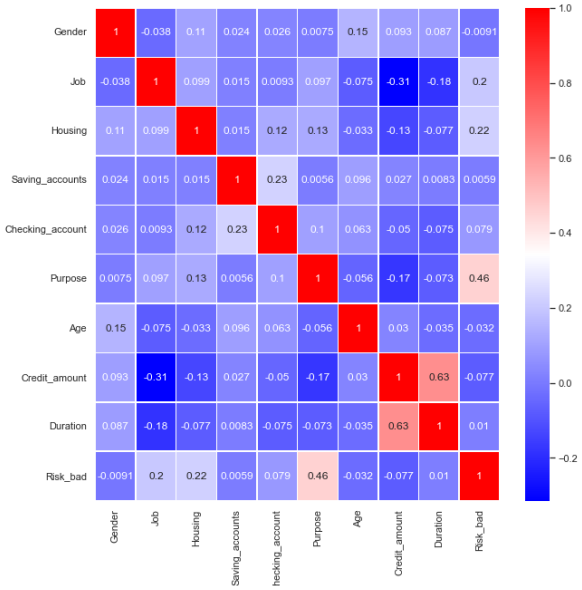

Figure 38: The correlation matrix of the normalized German credit risk data.

results.

6.5.3 Variance Inflation Factor

The variance inflation factor (VIF) of a regressor is another metric of how correlated a regressor is to all other regressors in the data set; when a VIF equals one the regressor is uncorrelated to the other regressors, as one observes greater VIF values the more correlated the regressor is to the other regressors, and for values greater than 4 it may be wise to investigate the data (The Pennsylvania State University, n.d). When a regressor has a low VIF, it is not easily substituted for by one or more other regressors when performing model selection.

We calculated the VIF of each regressor in the normalized German credit data set, and obtained the results given in F. The new model with which defaults are generated is given in T, the true intercept β_0 is adjusted from $-1.5e^2$ to $-1.1e^2$ such that a comparable number of defaults (145) is present in the new data.

	feature	VIF
0	Gender	3.230556
1	Job	4.458260
2	Housing	3.853087
3	Saving_accounts	1.545170
4	Checking_account	2.389636
5	Purpose	4.327547
6	Age	2.273277
7	Credit_amount	3.292005
8	Duration	3.964613

Figure 39: : Regressors and their VIFs.

*Table 6: The value of the regressors parameters used in generating scores from normalized German credit risk data, after which the score is transformed into a PD via a logit transformation. With this PD a default is binomially generated. Note that the four regressors corresponding to the lowest four VIFs have regressor parameters **not** equal to zero.*

$\beta_{Gender} = 1$	$\beta_{Job} = 0$	$\beta_{Housing} = 0$
$\beta_{SavingsAccount} = 5$	$\beta_{CheckingAccount} = 2$	$\beta_{Purpose} = 0$
$\beta_{Age} = 5$	$\beta_{CreditAmount} = 0$	$\beta_{Duration} = 0$

The problem of selecting and accepting irrelevant regressors that was visible in e.g. using Forward stepwise without a data strategy on the original data, is still visible in the 'low-VIF data' results when not making use of a data strategy (Figure 40). The proposed solution of using Forward stepwise combined with data splitting $25\%MS$ (Figure 41) significantly reduced the acceptance rates of irrelevant regressors. This means that the solution is not model and data dependent in even when using the best possible model in terms of regressors being relatively un-substitutable.

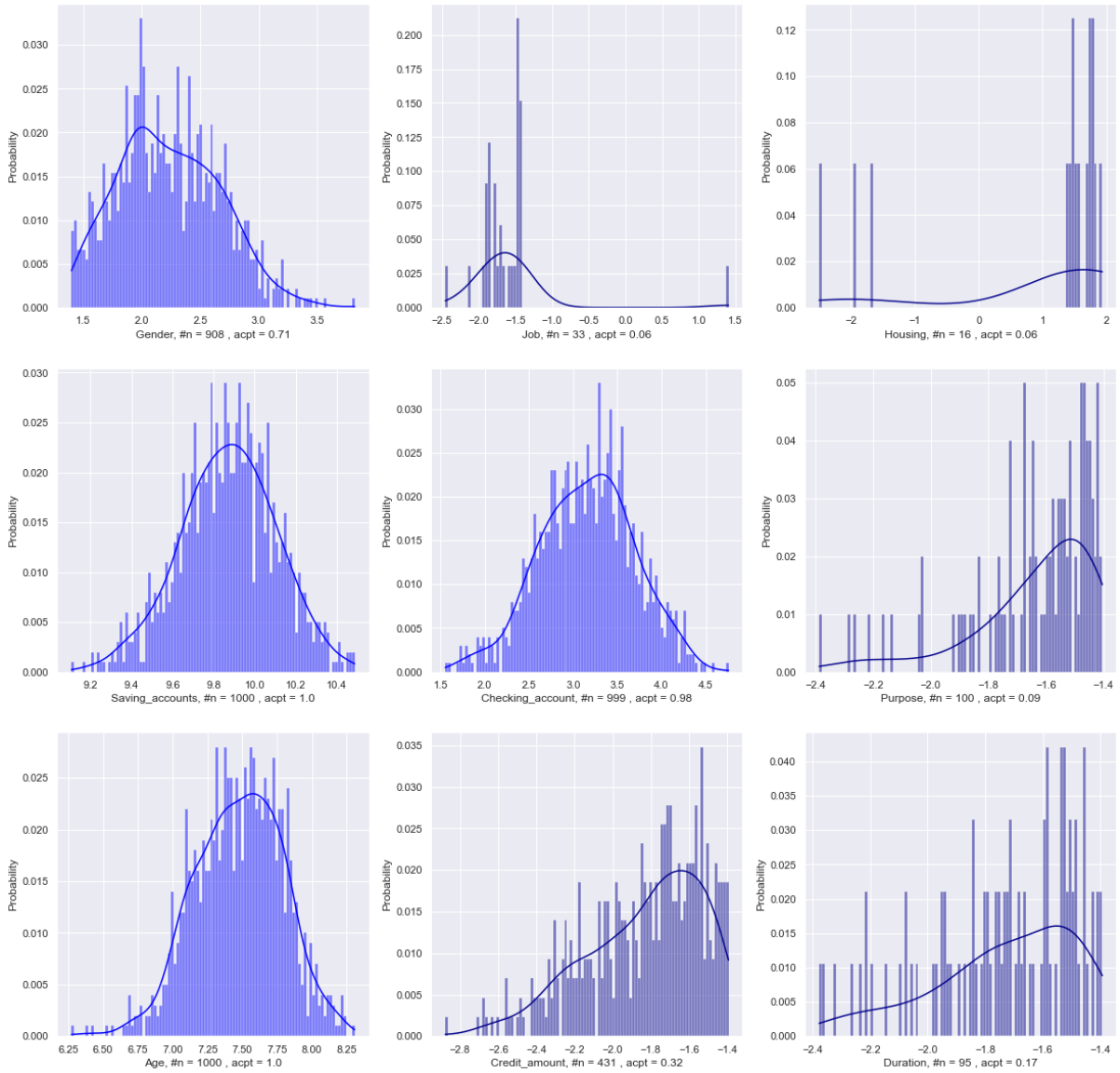

*Figure 40: : t-Value sample distributions as a result of using a **low VIF** data sample for both **Forward** model selection and subsequent fitting of the selected model. In the lighter blue, the relevant regressors. Observe that irrelevant regressor $CreditAmount$ gets selected and accepted often.*

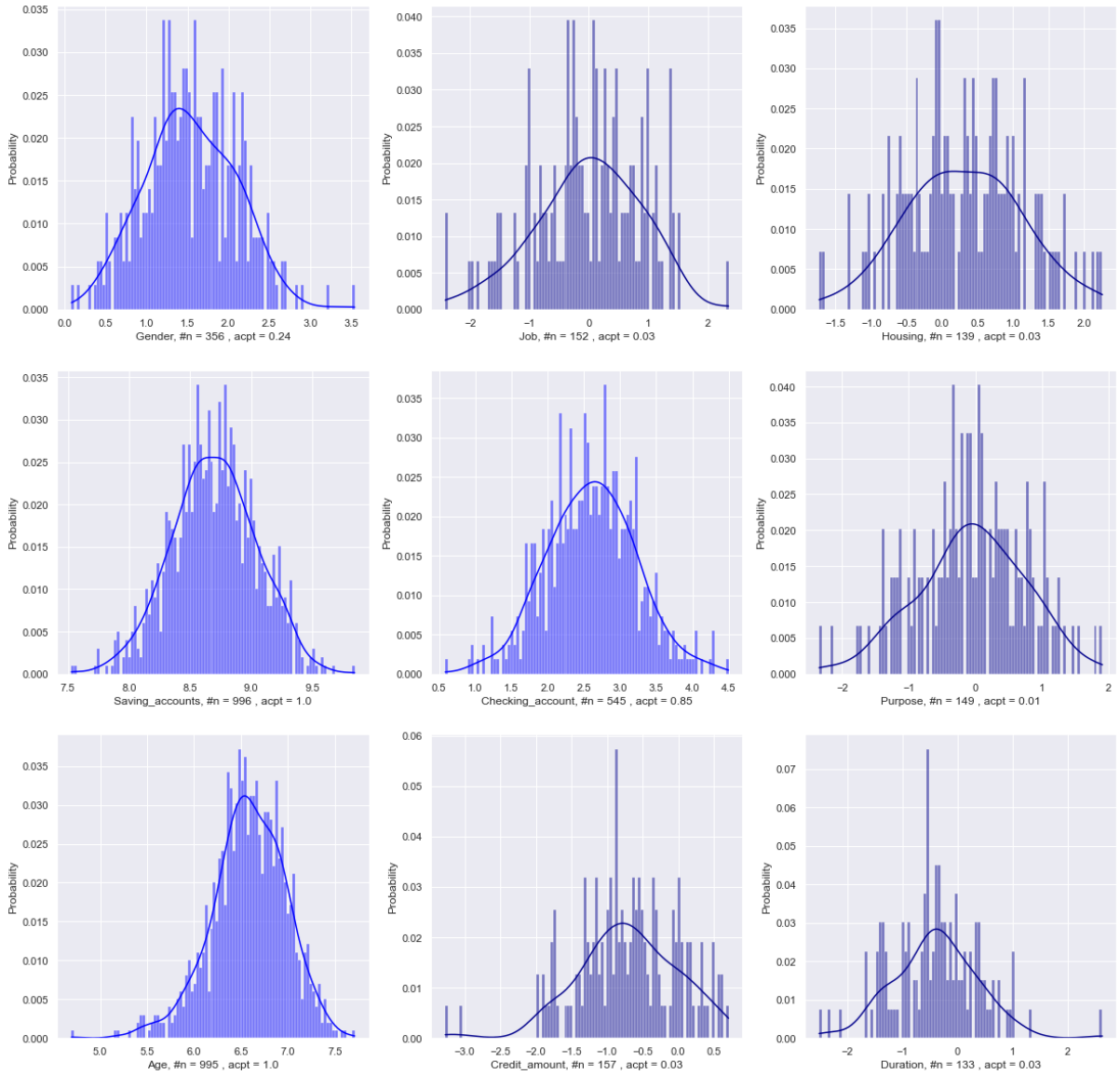

*Figure 41: t-Value sample distributions as a result of **splitting** the **low VIF** data, using 25% for **Forward** stepwise model selection and the other 75% for subsequent fitting. Irrelevant regressors get selected in a model a reasonable number of times, but do not pass the subsequent fit.*

6.5.4 Relative Beta Sizes

From the analysis of results in chapter 5 it can be seen that relevant regressors with the greatest true parameters are selected significantly more often than the relevant regressors that have lower true parameters. Needless to say, the larger the parameter the greater that regressor can drive probability of default. We study what happens when all relevant regressors are given a true regressor parameter β that equals 5. The intercept β_0 is adjusted to such that the number of defaults in the generated data is comparable to the number of defaults in the original experiment (156).

When using Forward stepwise without a data strategy, the results (Figure 42) show that there is significant risk of adding irrelevant regressors to a model. It must be noted that now all relevant regressors have equally sized regressor parameters, they do get selected and accepted an equal number of times. This suggests that the relative size of the true regressor parameters affects model selection output. The results of the proposed solution (Figure 43) show that the risk of adding irrelevant regressors to a model is significantly decreased compared to the baseline. Furthermore, when using the solution, relevant regressors are selected and accepted according to their true power; so no trade-off is visible.

Concluding, the true value of the relative regressor parameter β sizes do not reduce the probability of selecting irrelevant regressors to a model when not using the proposed data strategy. Making use of Forward stepwise combined with data splitting 25% MS is still preferred over the using the same sample for both model selection and model fitting, because the true power of both relevant and irrelevant regressors are accurately assessed.

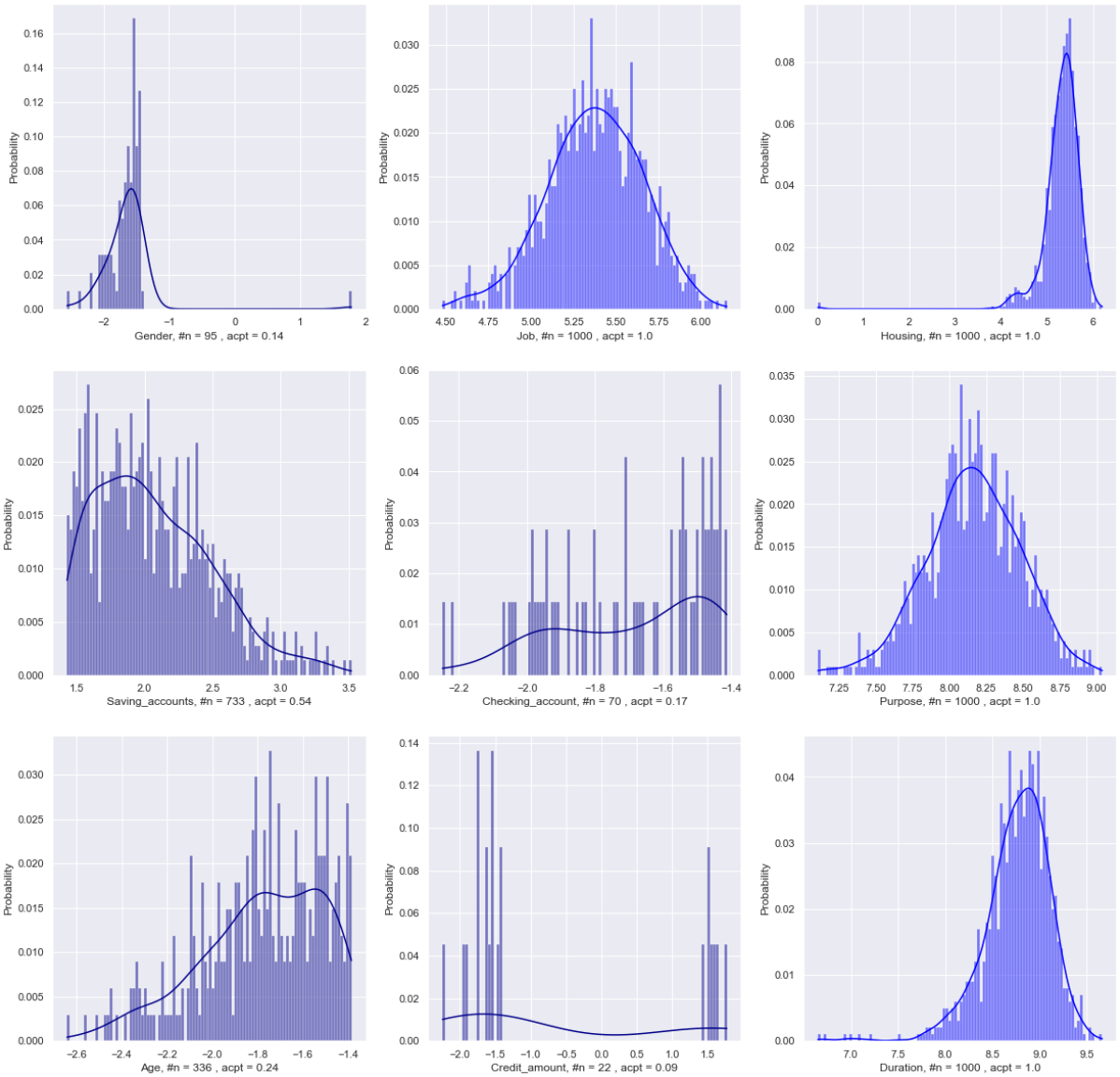

*Figure 42: t-value sample distributions as a result of using an 'equal-beta data sample' for both **Forward** stepwise model selection and subsequent fitting. Observe how irrelevant regressors (e.g. $SavingAccounts$ and Age) are selected and accepted rather often. This means that even when relevant regressors are easily recognizable, there is still a high risk of selecting and accepting irrelevant regressors in a model.*

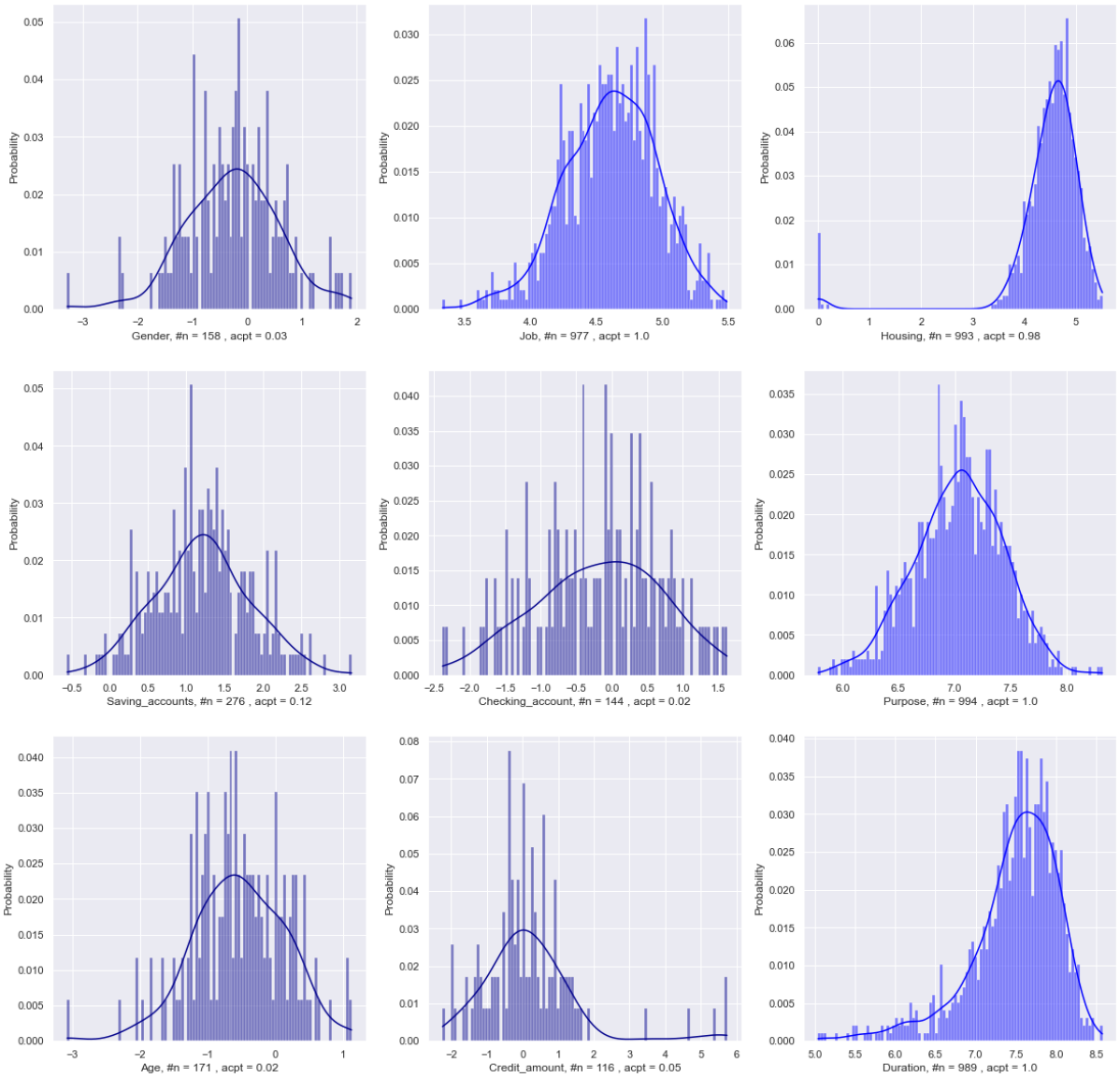

*Figure 43: t-value sample distributions as a result of **splitting** an 'equal-beta data sample' where 25% is used for **Forward** stepwise model selection and the other 75% for subsequent fitting. Observe how the relevant regressors in the lighter shade of blue are recognized according to their true power. That is, prevalence in selected models and acceptance rates after fitting are equal across all the relevant regressors. This means that, when using the proposed solution, the true power of both irrelevant and relevant regressors are accurately assessed.*

6.5.5 Real Data

Data splitting is inherent to model selection uncertainty because the selected model is dependent on the data used. That is, one random split yields a model but another random split yields a different model. So, to make the proposed solution of Forward stepwise and data splitting 25% MS useful for Rabobank, a method should be engineered that removes the uncertainty in which model to select when using data splitting. This method can be based on bootstrapping as follows:

- A sample is drawn from an available data set at Rabobank;
- This sample is split into a model selection part (25%) and a model fitting part (75%);
- Forward stepwise is used to select a model;
- The selected model is fitted and the regressor parameters β as well as the reported t-values and p-values are stored;
- The above four steps are repeated a number of times;
- Based on the reported p-values, a human decides on which regressors to add in a model. The average βs reported for these selected regressors can be used as final regressors parameter values.

The above procedure is performed using Forward stepwise without data splitting first (Figure 44), and thereafter with data splitting (the proposed solution) (Figure 45). It can be seen that the t-value sample distributions when using the proposed solutions are highly different than from using the baseline of no data strategy. From the perspective of the author, the regressors that should at least be selected based on the results of not using a data strategy are $Gender$, $Housing$, $SavingAccounts$, $CheckingAccount$, $Purpose$, and $Duration$.

When using the proposed solution of Forward stepwise combined with data splitting 25% MS, the regressors that should at least be selected to a model are $SavingAccounts$, $CheckingAccount$, and $Duration$. From a comparison of fitting these models to all of the data, it can be seen that performance in terms of $Log\ Likelihood$, AIC, and BIC are relatively comparable (Table 7). It is notable that, even though the proposed solution yields a model with only 3 regressors, the BIC is slightly lower than the baseline-model. This is a good result, because it means that the model that is selected using the proposed solution explains the data better than a baseline model with twice as many regressors in it.

Table 7: A comparison of the selected models using the 'old way' and the improved new way.

Method:	**Forward Stepwise – No Data Strategy**	**Forward Stepwise – Data Splitting 25% MS**
Selected Model	$Gender$, $Housing$, $SavingAccounts$, $CheckingAccount$, $Purpose$, and $Duration$	$SavingAccounts$, $CheckingAccount$, and $Duration$.
LLF	-510	-519
AIC	1034	1046
BIC	1069	1066

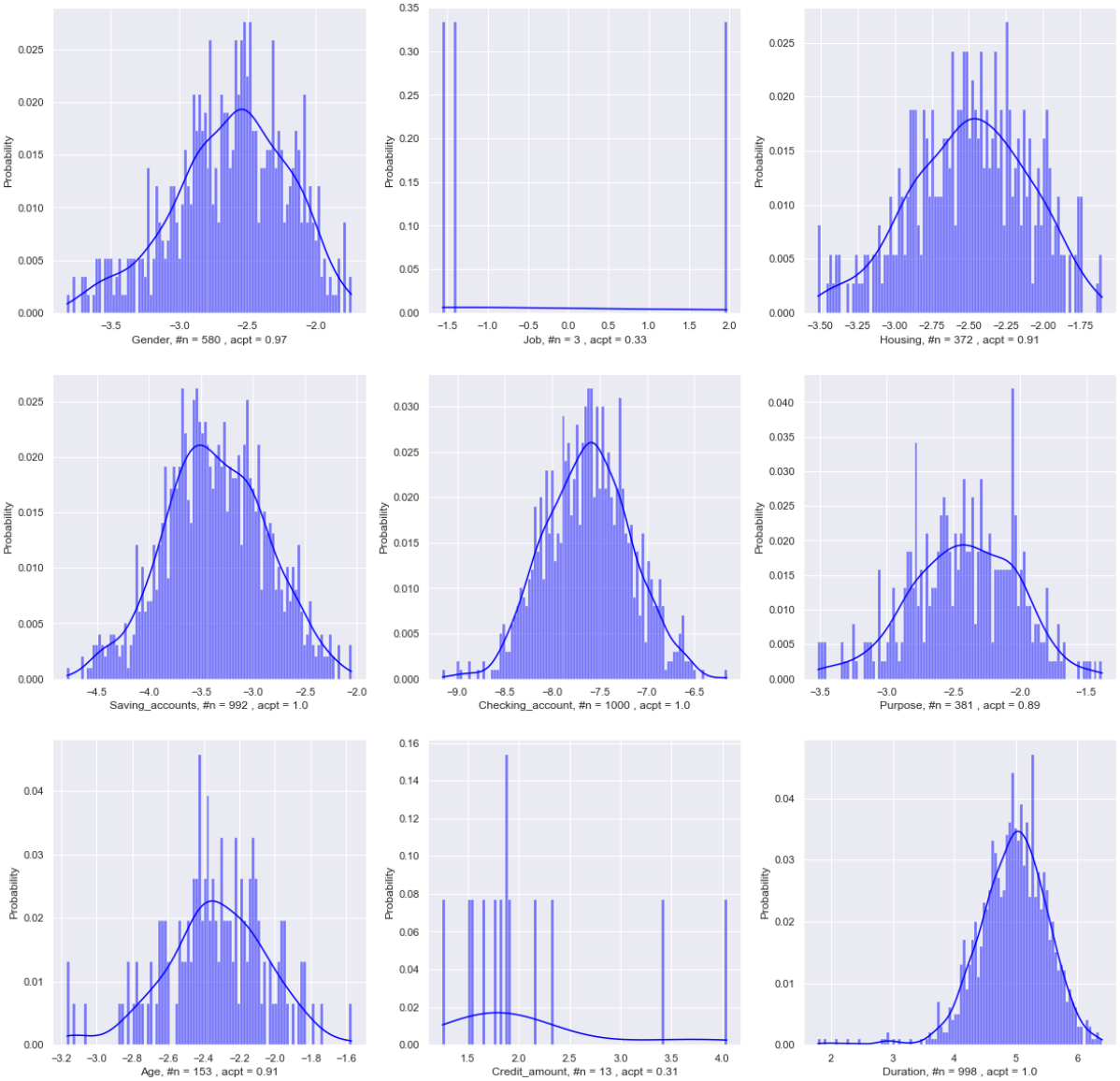

Figure 44: t-value sample distributions as a result of using a data sample, drawn from the original German credit risk, for both Forward stepwise model selection and subsequent fitting. From the judgement of the author, based on these results, a modeller may select the regressors $Gender$, $Housing$, $SavingAccounts$, $CheckingAccount$, $Purpose$, and $Duration$ in a model because these have high prevalence in selected model and high acceptance rates after fitting.

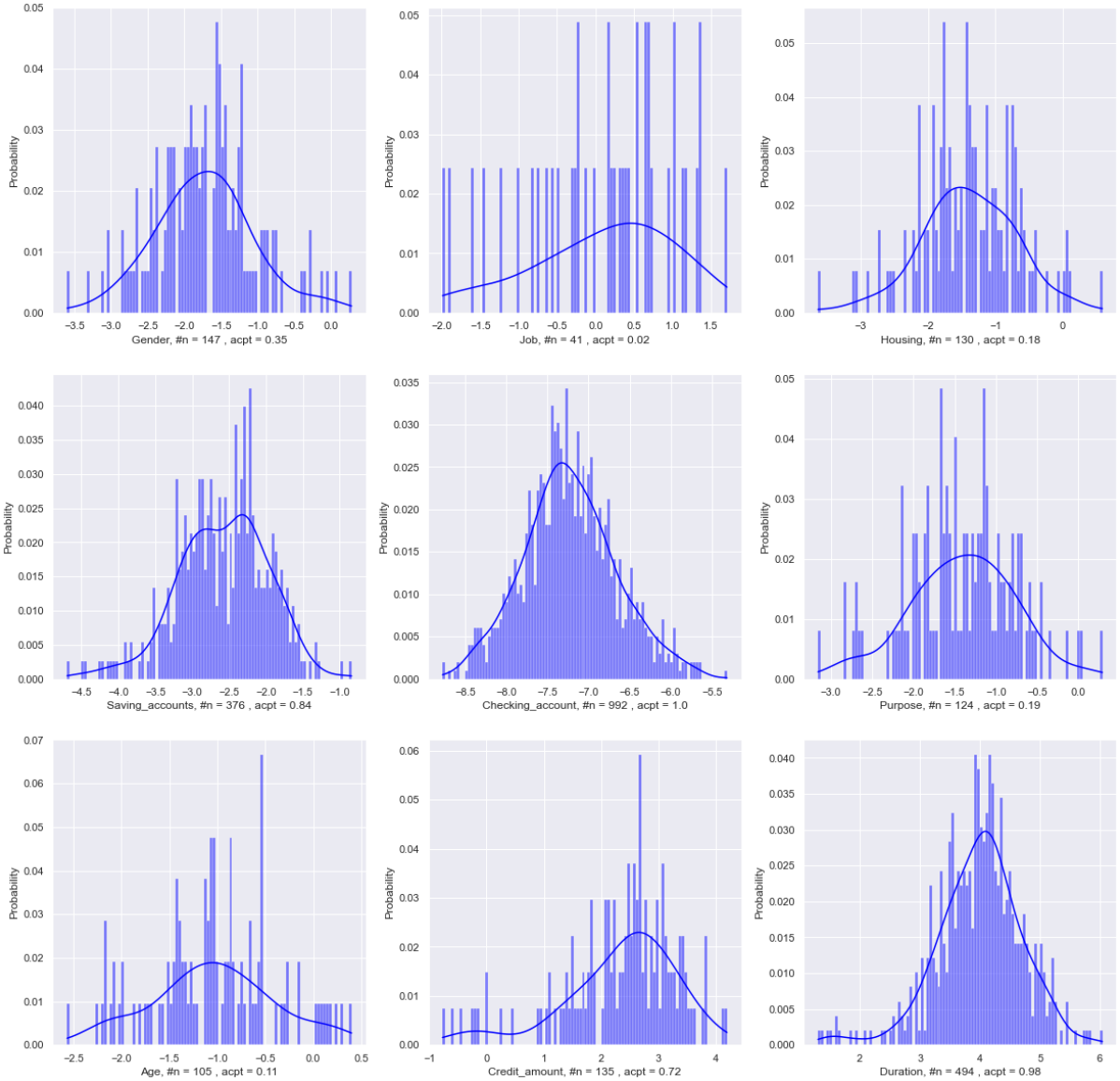

*Figure 45: t-value sample distributions as a result of **splitting** a data sample, drawn from the original German credit risk, in a **Forward stepwise** model selection part (25%) and fitting part (75%). From the judgement of the author, based on these results, a modeller may select the regressors $SavingAccounts$, $CheckingAccount$, and $Duration$ in a model because these have high prevalence in selected model and high acceptance rates after fitting.*

6.6 Concluding Chapter 6

In this chapter we decided on which model selection algorithm in combination with which data strategy to use: forward stepwise combined with data splitting. Then, we determined the optimal splitting strategy: to use 25% of a data sample for model selection, and 75% for model fitting the selected model. Because the selected model is dependent on the data sample given to the model selection algorithm, we studied if via this proposed solution we can recognize the model responsible for generating defaults accurately. We recognize the known true model by bootstrapping, where we draw samples from a generated data sample. The probability of selecting and accepting irrelevant regressors is significantly lower compared to not using a data strategy when bootstrapping.

The solution of using forward stepwise combined with data splitting is then tested on real data, because in the real world we cannot determine how regressors are distributed. First, we use German credit risk data from Kaggle by applying a known model to it. This known model generates defaults, and from this generated data we bootstrap. Our solution performs better than the baseline in terms of not selecting irrelevant regressors to a model. Furthermore, these results are stress tested, so to speak, by studying correlation in the data set, having a different model generate defaults with regressors that are statistically difficult to substitute for others, and let another model generate defaults where all relevant regressors have equal β sizes. In all these scenarios, the solution (forward stepwise and data splitting $25\%MS$) performs better than the baseline of forward stepwise and not using a data strategy, and correlation within the data is found to not be problematic.

After this stress test, we use our proposed solution on real data to see if it will result in selecting a different model, compared to using only forward stepwise without a data strategy. We use the original German credit risk. We bootstrap samples from this data, select a model and fit this model using the proposed solution one thousand times. The results are t-value sample distributions (histograms), which we can use to decide on which model to select by looking at the number of times a regressor is selected and the acceptance rate of regressors after fitting. While this is subjective, by selecting regressors with relatively strong prevalence in selected models and high acceptance rates after fitting, we are able to reduce the number of regressors in a selected model from 6 to 3 while the performance in terms of AIC, BIC and LLF is very comparable and even slightly better for BIC. This means that with our solution, modellers can achieve better performance with simpler models.

Lastly, to obtain more sparse results with Lasso model selection, we used a different performance metric to determine an optimal penalty constant λ^*_{1SE} via the one standard error $(1SE)$ rule. Previously we determined λ^*_{1SE} via the in-sample log-likelihood value with 5-fold CV, now we base it on the out of sample (OOS) AUC. An optimal penalty constant λ^*_{1SE} is computed in the range $[2, 2.1, \dots, 4]$, and the model selection results are indeed more sparse. Irrelevant regressors are selected in a model a comparative number of times to Forward stepwise and data splitting. However, with Lasso these irrelevant regressors are accepted after fitting the selected model significantly more often. This makes Lasso unpreferred compared to the proposed solution of using forward stepwise and data splitting $(25\%\ MS)$, and hence it is not advised to use Lasso as a model selection algorithm.

7. Conclusions and Recommendations

In this chapter we answer the main research question, and based on this formulate recommendations for Rabobank. These recommendations are such that the current PD modelling methodology can, without much effort, be adjusted. First, recall the mean research question: *In what way, if at all, are the regressor-parameters and model performance of credit risk models at Rabobank biased, and how can Rabobank adjust for this bias both statistically and managerially?*

After a summary of the business case (section 7.1), we conclude this research by first concluding on how the current model selection algorithms at Rabobank can yield biased model statistics (section 7.2), then we discuss solutions (section 7.3), and formulate concrete recommendations on how to modify the model selection methodology (section 7.4). Finally, we discuss ideas for future research (section 7.5).

7.1 The Business Case of Biased PD Model Statistics

In the banking industry, probability of default (PD) models are used in estimating the probability of a client not paying back a loan. Needless to say, accurate PD estimates are crucial to responsible and sustainable banking. Building PD models, however, is complex and when an algorithm is used to build these models, there is a risk of biased model statistics. Literature (Berk, Brown & Zhao, 2010) on this bias indicates that regressor parameter β and t-value estimates can not only be inaccurate, but these estimates are conditional on the model selection algorithm used, other regressors present in a model, and the standard error of the regressor parameter given the data. Furthermore, model performance can be inflated as a result of k-fold cross validation (Moshontz, Fronk, Sant'Ana & Curtin, 2020) and possibly model selection algorithms. This research, therefore, aims at quantifying bias in PD model statistics to improve the PD model development methodology at Rabobank.

7.2 Identified Current Problems

This research was conducted to investigate if the stepwise model selection algorithms resulted in biased PD model statistics: PD model performance, regressor t-values, and regressor parameters β. Biased PD model statistics can be problematic when model performance is inflated, or when the model contains regressors that actually do not contribute to the PD of a customer. We identified, from literature, how to identify this bias and how to potentially correct for it as well. We used known models to artificially generate defaults. That is, the regressors weights $\boldsymbol{\beta}$ with which defaults are generated are known.

To identify biased inference, we generate samples from a known model and we use that generated sample for model selection and subsequent model fitting. We simulated the stepwise model selection algorithms currently available at Rabobank, as well as Lasso. These model selection algorithms can be combined with data splitting, data carving, adding noise, or doing nothing. Furthermore, we determined that aspects of well performing strategies yield accurate β estimates with low standard deviations and ideally select and accept only the regressors for which their true β is not equal to zero. We found that the main problem of the current PD model selection methodology in use at Rabobank is the risk of adding irrelevant regressors to a model. This risk is greatest when using the same data sample for both model selection and fitting the selected model.

We furthermore studied inflated model performance as a result of model selection algorithms and k-fold CV. We generated random data, and randomly assigned defaults to this data such that it can be argued the true area under the receiver operating characteristic curve (AUC) of a model that is selected based on this data must equal 0.5. We found that increasing the number of regressors and decreasing the size of the data set increases the risk of the average AUC over 5 folds, in 5-fold CV,

being inflated: i.e. being significantly above 0.5. It would be wise for modellers to at least be aware of this risk and use the tables generated in this research to determine if, given the importance of a model, the risk of inflated performance is too high.

7.3 Promising Solutions

From the experiments described in section 7.2, it can be concluded that the risk of selecting and subsequently accepting irrelevant regressors to a model is lowest with Forward stepwise in combination with data splitting. From analysing different configurations, data splitting should be such that 25% of the data sample is used for model selection, and the other 75% for fitting the selected model.

This solution performs well in terms of both accurate regressor parameters, as well as acceptance rates of both irrelevant and relevant regressors. Compared to the baseline of using the same data sample for Forward stepwise model selection and subsequent fitting, the proposed solution reduced the risk of selecting irrelevant regressors in a model by 15% on average. After fitting the selected model, the risk of accepting irrelevant regressors in a model is reduced by 86%.

To reduce the volatility of the model selection algorithm when using data splitting, Rabobank may make use of bootstrapping (Figure 46). With bootstrapping, samples are drawn from a data set (a population) and this drawn sample is split in a model selection part (25%) and a model fitting part (75%). The regressor statistics, in terms of parameters β, t-values and p-values, are stored for every fit. Finally, after all bootstrapping iterations are completed, sample distributions can be assessed to, via human expert-based intuition, select regressors. Here, it must be considered, Rabobank should be in favour of adding those regressors to a final model that are often present in selected models, and have high acceptance rates after fitting the selected model.

7.4 Recommendations for Improving the Model Selection Methodology

To answer the main research question: statistics, t-values as well as βs, of regressors post model selection can be biased such that they inaccurately reflect their true value. As a result, significant risk of irrelevant regressors being added to a final model is present when using the same data sample for both stepwise model selection and subsequent fitting. Model performance can be inflated as a result of model selection, but with enough data this problem is insignificant. From the experiments and the analysis of the data may be formulated the following recommendations on improving the PD model selection methodology of Rabobank:

- To, first and foremost, be aware of biased inference with PD model selection and;
- To, when developing models, use Forward stepwise in combination with data splitting, where 25% of a data sample is used for model selection and the remaining 75% for model fitting and;
- To make use of bootstrapping in practice, as from the results of Forward stepwise combined with data splitting we are able to recognize the true model responsible for generating defaults when bootstrapping via reported t-values after model fitting and;
- To, given enough data, use Forward stepwise combined with data splitting and 5-fold cross validation to obtain an estimate of how well the data at hand can be used for predictive modelling; a modeller can judge the risk of inflated model performance using the tables in this research.

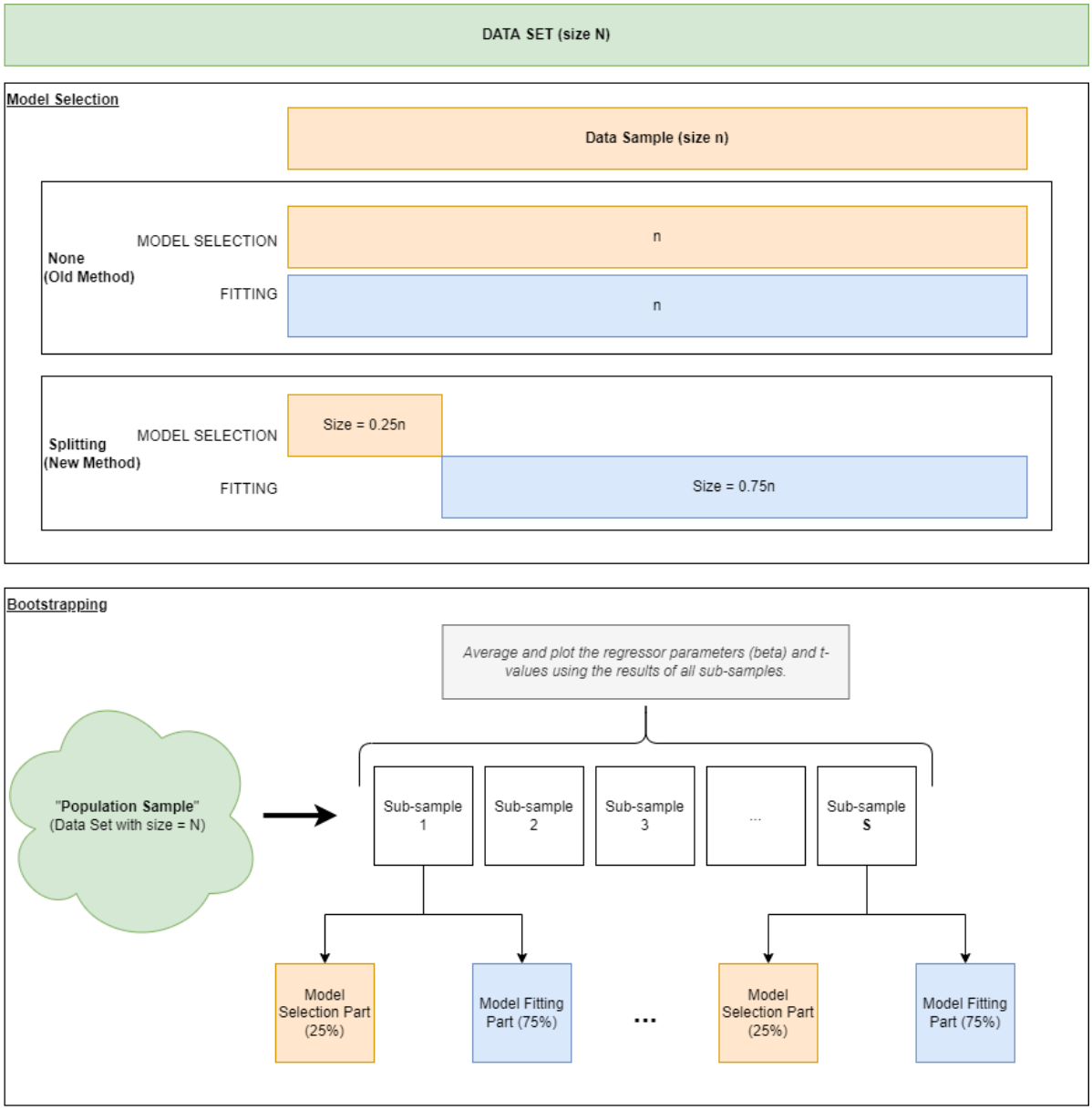

Figure 46: An overview of the old method of model selection, where a data sample is used for both model selection and model fitting. The new method, data splitting, is visualized as well. Bootstrapping to counter-act the variability in model selection when data splitting is visualized in the second block. Here, the idea is to sample 'sub-samples' from a data set (bootstrapping), which are used for model selection and model fitting. The β and t-value statistics should be stored for each sub-sample, which can then be aggregated and analysed as we have done in this research.

7.5 Future Research

The fact that irrelevant regressors can be added to a model and subsequently accepted with a high probability when using a data sample for both model selection and model fitting brings with it, even after this research, questions regarding hyperparameter tuning and selective use of data. That is, model selection pipelines can have several hyperparameters that influence the outcome, i.e. the model, significantly. Here, an example of a model selection pipeline can be selecting and subsequently fitting a model using some strategy. There is reason for future research to investigate which parameters of the model selection pipeline to tune, and whether or not to use unique data to do this tuning because we have seen that using the same data for both model selection and model fitting can result in bias. Examples of hyperparameters to tune are:

- The fraction of data to use for model selection and the fraction of data to use for model fitting, given a data set and;
- The number of samples to draw when using bootstrapping as a means to counter-act the volatility of the model selection algorithm when data splitting, to be time-efficient as bootstrapping many samples may take a significant amount of time.
- The penalty constant λ when researching Lasso as a model selection algorithm;
- Whether or not to use unique data to tune this λ constant;

A logical place to start is to make use of a known model that is responsible for generating defaults, because with such an experimental setup one is not limited to a dataset, minimizing the risk of results being inherent to that particular data set. Furthermore, this research, based on our results, being in favour of data splitting is contrary to modern literature on the topic of inference post model selection. Hence, future research may focus on hyperparameter tuning the type and amount of noise to add during the model selection stage or the specifics of data carving, when making use of the adding-noise strategy or the data carving strategy respectively.

Finally, even though our solution of combining forward stepwise model selection with data splitting is (stress-) tested using German credit risk data from Kaggle, an important comment to make is that we can only generalize the results of this research to a certain degree because we made use of only one particular known model. Therefore, future research may focus on performing different experiments using different known models depending on the context, or adopt a more formal mathematical approach with the goal of proving minimal bias occurs with some combination of a model selection algorithm and a data strategy.

References

1. Berk, R. A., Brown, L. D., & Zhao, L. (2010). *Statistical Inference After Model Selection. Journal of Quantitative Criminology, 26 (2), 217-236.* http://dx.doi.org/10.1007/s10940-009-9077-7
2. Federighi, E. T. (1959). *Extended Tables of the Percentage Points of Student'st-Distribution. In Journal of the American Statistical Association (Vol. 54, Issue 287, pp. 683–688). Informa UK Limited. https://doi.org/10.1080/01621459.1959.10501529*
3. The Pennsylvania State University. (n.d.). *5.7 - MLR Parameter Tests. Online.Stat.Psu.Edu.* Retrieved March 9, 2022, from https://online.stat.psu.edu/stat462/node/137/
4. Altman, E. I. (1968). *Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. The Journal of Finance, 23(4), 589.* doi:10.2307/2978933
5. Oesterreichische Nationalbank, Guidelines on credit risk management. Rating models and validation. Vienna, OeNB Printing Office, 2004.
6. Costa e Silva, E., Lopes, I. C., Correia, A., & Faria, S. (2020). *A logistic regression model for consumer default risk. Journal of Applied Statistics, 47(13-15), 2879–2894.* doi:10.1080/02664763.2020.1759030
7. Westgaard, S., & van der Wijst, N. (2001). *Default probabilities in a corporate bank portfolio: A logistic model approach. European Journal of Operational Research, 135(2), 338–349.* doi:10.1016/s0377-2217(01)00045-5
8. Blaskó, P. (2019). *Identification of credit default drivers via lasso estimation in the logistic regression model* (Doctoral dissertation, Wien). https://repositum.tuwien.at/handle/20.500.12708/15075
9. Tibshirani, R. (1996). *Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288.* doi:10.1111/j.2517-6161.1996.tb02080.x
10. Wang, H., Xu, Q., & Zhou, L. (2015). *Large Unbalanced Credit Scoring Using Lasso-Logistic Regression Ensemble. PLOS ONE, 10(2), e0117844.* doi:10.1371/journal.pone.0117844
11. Hull, J. C. (2018). *Risk Management and Financial Institutions (5th ed.). John Wiley & Sons.*
12. Merton, R.C. (1974), ON THE PRICING OF CORPORATE DEBT: THE RISK STRUCTURE OF INTEREST RATES*. The Journal of Finance, 29: 449-470. https://doi.org/10.1111/j.1540-6261.1974.tb03058.x
13. GARP (Global Association of Risk Professionals), Apostolik, R., & Donohue, C. (2015). Foundations of financial risk: An overview of financial risk and risk-based financial regulation (2nd ed.). Standards Information Network.
14. Loftus, J. Data Council. (2019, November 26). Valid Inference after Model Selection and the selectiveInference Package | NYU [Video file]. Retrieved from https://www.youtube.com/watch?v=bQhEALoxoGE
15. Moshontz, H., Fronk, G., Sant'Ana, S. J. K., & Curtin, J. J. (2020, September 14). Quantifying Optimization Bias in Model Evaluation when using Cross-Validation in Psychological Science: A Monte Carlo Simulation Study. https://doi.org/10.31234/osf.io/ns9mj
16. Tibshirani, R. Microsoft Research. (2018, July 8). *Invited Talk: Post-selection Inference for Forward Stepwise Regression, Lasso and other procedures [Video file].* Retrieved from https://www.youtube.com/watch?v=RKQJEvc02hc
17. Feldman, S. PyData Miami 2019. (2019, June 17). *You Should Probably Be Doing Nested Cross-Validation [Video File].* Retrieved from https://www.youtube.com/watch?v=DuDtXtKNpZs

18. Fawcett, T. (2006). *An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874.* https://doi.org/10.1016/j.patrec.2005.10.010
19. Faraway, J. J. (2005). *Extending the linear model with R: Generalized linear, mixed effects and nonparametric regression models. CRC Press*.
20. Zach. (2021, September 30). *Understanding the standard error of a regression slope*. Statology. Retrieved March 24, 2022, from https://www.statology.org/standard-error-of-regression-slope/
21. Starmer, S. StatQuest. (2018, June 11). *Logistic Regression Details Pt 2: Maximum Likelihood [Video File].* Retrieved from https://www.youtube.com/watch?v=BfKanl1aSG0
22. Brockhoff, B. DTUdk. (2013, December 15). *Lect.12D: F-Test, F-Distribution Leacture 12 [Video File].* Retrieved from https://www.youtube.com/watch?v=4Gaj-IQgptI
23. The Pennsylvania State University. (n.d). 10.7 - *Detecting Multicollinearity Using Variance Inflation Factors. Online.Stat.Psu.Edu.* Retrieved June 2nd, 2022, from https://online.stat.psu.edu/stat462/node/180/
24. Chen, Y., & Yang, Y. (2021). *The One Standard Error Rule for Model Selection: Does It Work?* In Stats (Vol. 4, Issue 4, pp. 868–892). MDPI AG. https://doi.org/10.3390/stats4040051