Master Thesis

A Comprehensive Validation Procedure In Credit Scoring Model

Maastricht University School of Business and Economics

Place & Date: Maastricht, 22/01/2021

Name: Min Woo (Mike) Lee

ID Number: i6127619

Study: MSc. International Business

Specialized in Information Management and Business Intelligence

Course Code: EBS4011

Supervisor Name: Alexander Grigoriev

Abstract

Purpose

The paper studies the stability of a classifier to the change of dataset with resampling and noise in a specific case of the German Credit dataset. The purpose of this study is to include a robust check on top of predictive accuracy in order to propose a holistic validation procedure. Unstable classification and inaccurate credit scoring model prediction would potentially cost lenders a substantial cost due to poor estimation. To address this issue, this study aims to check the robustness of classification, given two machine learning algorithms such as Logistic regression and Support Vector Machine

Methodology

Given the same dataset and method that were used by Ala'Raj & Abbod (2015), the baseline model is constructed. Then, the original procedure is repeated to the resampled models for pairwise comparison. The entire process is executed with Python and Scikit-Learn to apply Machine Learning algorithms, statistical analysis, and data visualization.

Findings

The results showed that Logistic regression and Support Vector Machine were quite robust when the original training sample was perturbated with the bootstrap method and noise was added. However, these two classifiers' classification was not robust when the sample was randomly oversampled or under-sampled. Bootstrap models regardless of iterations showed a very similar proportion of disagreement against baseline classification. The robust result was also apparent when noise was added. Regarding model performance, the robustness did not necessarily improve predictions' accuracy compared to the baseline models. This was the case for bootstrap and noisy models. However, the accuracy of oversample and under-sample models did not deviate much from the baseline models.

Table of Contents

1 Introduction

In 2004, the Basel Committee introduced Basel II Accord to reform the existing Basel I Accord framework. The Basel II Accord comprises three main pillars: minimum capital requirements, supervisory review, and market discipline (Decamps et al., 2004). To fulfill these pillars especially minimum requirement and supervisory review, financial institutions have paid more attention to create an appropriate credit scoring model (Brown & Mues, 2012). Generally, a credit loaner from banks or financial institutions creates a credit scoring model as a decision-making tool when deciding to grant a credit to a potential borrower. With the help of a credit scoring model, it has been widely used in diverse fields such as consumer credit and corporate credit (Thomas, 2000). The purpose of a credit scoring model is to avoid the subjectivity of a credit analyst's decision and improve objectivity that solely uses information given by the borrower (Fensterstock, 2005). In terms of consumer credit, the model is usually developed with statistical analysis or machine learning algorithms based on the historical data of a borrower such as income, age, and existing debts to determine the level of credit risk in percentage or a binary form like good credit or bad credit (Thomas, 2000).

However, the established model does not perform as intended sometimes, due to changes in a data structure or failure of data integrity (Smith & Doyle, 1992). Chye Koh et al. (2006) pointed out that if the performance or classification of the model becomes inaccurate or non-robust, then some borrowers with a good credit status may lose an opportunity to receive an appropriate amount of credit or may not even get any credit in worst-case scenarios. On the other hand, borrowers with a low credit rating may still receive a credit if the model inaccurately classifies it. A potential problem would arise for the lenders, as the borrowers will have a higher chance not to make future payments on time. When the payments are continuously delayed from bad credit customers or ended up going default, then the lenders would face a substantial cost and an opportunity cost. The cause of this effect is due to the poor and non-robust credit scoring

model. To prevent these issues, the importance of the model validation procedure has risen to academics and managers. The following reason is that invalid results from the model would lead to an inaccurate or inappropriate indication to them (Robinson & Froese, 2004). As a result, validation is a necessary procedure to check the validity of the model and stability of output, given a method.

Robustness test and validation on unseen data are some of the validation process components (Heyden et al., 2001). Box (1953) introduced a term called "robustness test" to check the possibility of results when an underlying assumption of the test is violated. It also examines how stable the outputs are compared to the outputs from the correct assumption, given the following assumption is violated (Boneau, 1960). Heyden et al. (2001) described robustness as "the ability to reproduce the analytical method under different circumstances without the occurrence of unexpected differences in the obtained results." In other words, a method is robust when a classifier's stability remains similar to the change of dataset (Salciccioli et al., 2016). Another measurement of validation is that the developed model should be validated to answer the following question, "Can the model account for all of the previously observed input-out behavior?" (Smith & Doyle, 1992). In other words, this element of validation includes a verification of accuracy that measures both the number of correct and incorrect classifications (Salciccioli et al., 2016).

A validation procedure is a key activity for financial institutions' risk management in terms of business perspective. In recent years, European Central Bank (ECB) and the Basel Committee demand financial institutions for additional attention on the existing model validation framework on a regular basis to determine shortcomings of the model due to missing risk exposures or unsuitable coverage. Hence, establishing a robust model validation framework would improve the effectiveness of model predictions, avoid backlash from model results, capture an appropriate risk exposure, and identify the model's weakness. Besides, financial

institutions can also mitigate the significant risk with their decision making beforehand (Chandrawat, 2020). This phenomenon indicates that the validation process is highly demanding in financial institutions. Besides, an extensive validation process would provide more significant insights into the management of risk exposure. Hence, financial institutions can create relevant internal policies that cover model validation activities and responsibilities.

Furthermore, in terms of academic perspective, many previous studies are concerned with one part of the validation procedure, which is an accuracy performance metric. It typically considers with K-fold cross-validation or various ratio of hold-out samples (Ala'Raj & Abbod, 2015; Trivedi, 2020; West, 2000). However, not many papers focused on the component of robustness in the topic of credit scoring. Therefore, the validation procedure was not studied and a holistic manner, which remains a question the robustness of the proposed model results. To address the gap in current knowledge on a validation procedure particularly in the credit scoring model, this paper intends to benchmark the study by Ala'Raj & Abbod (2015). This study is selected as a benchmark paper because none of the studies currently have been investigated further validation on this specific paper. The authors also suggested that extra validation is recommended for further research. Although Ala'Raj & Abbod (2015) repeated their experiments 100 times with different hold-out samples to validate their models, the robustness of a method was not executed. Given the same dataset and method that were used by Ala'Raj & Abbod (2015), this paper aims to add an extra validation procedure for expanding the aspect of robustness. This study specifically studies the robustness of a given algorithm's output to the change of dataset by resampling and noise. Then, the perturbed sample predictions are tested on a hold-out sample for accuracy performance to provide an extensive validation procedure. The following problem statement is answered:

How robust are the outputs by a classifier to the change of dataset?

The remainder of the study structures with four sections. Firstly, section 2 is a literature review that discusses applications and usages of a credit scoring model, existing studies of classification algorithms, and validation techniques commonly used in credit scoring. After that, the following hypothesis is formulated based on the existing literature. Secondly, section 3 describes a methodology that describes data, overall pipeline, predictive models, resampling methods, noise, and evaluation metrics. Thirdly, section 4 compares the results between the baseline model and the resampled model. Lastly, section 5 discusses theoretical contributions, managerial implications, limitations, and suggestions for further research.

2 Literature Review

2.1 Credit scoring

In prior years, financial institutions have mainly adopted credit scoring as a decision-making tool for loan applications. The credit scoring model application did not only limit loans, like other fields requiring decision-making have also been recently started to implement credit scoring. It specifically drew the mortgage and insurance industry's attention to adopt credit scoring as well (Chye Koh et al., 2006). For the mortgage industry, credit scoring is used to quantify a mortgage applicant's probability of default, which follows a similar principle as loan applications (Wagner, 2004). For instance, Federal Home Loan Mortgage Corporation and GE Capital Mortgage Corporation have been used credit scoring to examine which mortgage applicants require additional look or check for credit risk adjustment (Mester, 1997). Chye Koh et al. (2006), however, pointed out that the level of credit scoring in other fields is not widely used as loan applications. The leading cause of this trend is that banking institutions put additional attention on their credit risk management than other fields as their loans business is their primary income source. Another factor is that demand for personal loans have been escalated due to economic pressures, which stimulated financial institutions to develop the best statistical credit scoring among competitors (Janeska et al., 2014). Therefore, the use of credit scoring is more concentrated in loan business than other fields.

Furthermore, the utilization of the credit scoring model for loan applications results in numerous advantages. Firstly, only a small number of lenders are required to process the large volume of loan applications, which minimizes the total cycle time of the application process and reduces operating costs (Ince & Aktan, 2009). Referring to Leonard (1995), Canadian banks spent nine days on average for credit approval before credit scoring was used. After the implementation, the approval time was drastically reduced to three days, which allowed them to allocate their

resource to other activities. However, a faster lending process indicates that the scoring model may have a greater chance of errors due to flawed interpretation or analysis by the model. The reason is that some of the unique features of a borrower may not be examined or weighted correctly as the objective credit scoring model is to make generalized predictions. This issue may not be significant when a loan officer evaluates the application manually because these unique features would be considered individually (Janeska et al., 2014). Hence, some applicants might be unintentionally negatively influenced by the credit scoring model, although it helps streamline the process efficiently.

Secondly, the model can prevent poor judgments by loan officers with limited experience. This can be a crucial factor for adopting a scorecard because the aftermath of the inadequate decision by loan officers would create additional bad debt, consequently influencing financial institutions' profitability (Ince & Aktan, 2009). Also, it provides a positive side to borrowers as well. Before the presence of credit scoring, borrowers especially small businesses struggled with credit availability due to information asymmetries between lenders. Hence, the borrowers could not obtain the appropriate amount in the credit market. With the emergence of credit scoring, information failure was significantly minimized between two parties, which created a greater lending opportunity for small business borrowers (Frame et al., 2001). On the other hand, statistical credit scoring's assumption can also contribute to information failure. The assumption is that a borrower's personal information like age, gender, type of work, and existing loans are linked to risk to a certain extent, but the question remains "what share of risk is linked with those factors that can be included in a scorecard" (Schreiner, 2004),

Lastly, Mester (1997) highlighted that the loan approval process becomes more objective when credit scoring is implemented. Elimination of subjectivity and personal discrimination help loan officers to make a non-biased decision as the identical underwriting criteria are applied to borrowers. However, the entire objective decision cannot be achieved by the model. In some

cases, some borrowers who had received bad credit due to recent economic difficulties from divorce, redundancy, or sudden job loss may have a lower chance to receive good credit status in the future. When a single condition is applied to these people and grouped them in a "default" group without applying different rules, the objective decision's benefit might be questionable. The reason is that when the model initially classifies a borrower as bad status, then the status may not be adjusted even after the overcome of economic difficulties (Hand, 2001).

2.2 Classification Algorithms

Many studies applied diverse statistical methods to make an accurate credit scoring model. Different algorithms are implemented based on the credit data type because some of the credit data's target variable comprises continuous values or binary form. The regression algorithms are proposed when a dependent variable has continuous values, while classification algorithms are adopted for binary numbers. As private credit data is usually not publicly available due to privacy concerns, most publicly available credit data mainly follow a binary form. Due to this reason, classification techniques are popular among credit scoring model research (Lessmann et al., 2015; Xia et al., 2018). In the same vein, Clancey (1984) defined the simple classification as "the simplest kind of classification problem is to identify some unknown object as a member of a known class of objects." To be specific, a classification problem assigns into a particular class, given an input or independent variable. In the example of classification problem in the credit risk domain, if the borrower is above 40 years old and earns more than 3000 euros per month, then the class is assigned as positive credit, otherwise negative credit. Hence, a borrower who follows with a prior condition will be assigned to Class 1, but in other cases, Class 0. To solve this classification problem in machine learning, classification methods or algorithms are used. However, the pitfall of the method is that the classification technique does not necessarily give a "right answer" all the time since the output is only based on "known solutions" (Clancey, 1984). This indicates that the solution of the classification algorithm may not always be robust.

Despite the potential pitfall of the classification algorithm, Dong et al. (2010) listed popular classification techniques into five groups mostly used in literature. “Statistical models, operational research methods, artificial intelligence techniques, hybrid approaches, and ensemble models” are prominent classification methods to construct a credit scoring model. Among these techniques, statistical models and artificial intelligence techniques are widely studied compared to methodological simplicity and proven performance. Logistic Regression (LR), Decision Tree (DT), and K-Nearest Neighbor (KNN) are the primary example of statistical models. At the same time, Support Vector Machine (SVM) and Neural Network (NN) algorithms belong under artificial intelligence algorithms.

Moreover, to explore which specific binary classification methods are extensively implemented to separate good and bad credit, Louzada et al. (2016) investigated a systematic literature review based on 187 studies that were published in scientific journals during 1992-2015. The literature review suggested that artificial intelligence techniques like NN and SVM are common methods by researchers in addition to hybrid and ensemble techniques. This result aligns with the five groups that are suggested by Dong et al. (2010). In contrast to Dong et al. (2010), statistical models, specifically LR and DT are used to compare prediction performance against newly developed techniques. This phenomenon explains that statistical models are generally inferior to the advanced techniques in performance, which encourages them to use them as a benchmark algorithm (Louzada et al., 2016). Nevertheless, these studies both confirmed that SVM and LR are qualified as an appropriate classification algorithm in the study of credit scoring as these algorithms consistently performed based on the existing literature.

LR and SVM algorithms have their benefits and drawbacks due to distinctive characteristics of algorithms. Logistic regression does not assume that the relationship between response and explanatory variables are linear, and thus variables do not have to be normally distributed. This unique feature allows LR to generate a “simple probability formula” for the classification

problem. However, non-linear cannot be solved a problem with LR, which is the downside of the algorithm (Hooman et al., 2013). On top of that, Feng et al., (2014) exclusively corrupted the single sample from negative to positive class to measure LR's sensitivity. The sample corruption influenced the impact of the regression curve, which showed an unstable output. Therefore, the authors stated that LR is quite sensitive when the sample is disturbed, and further research on this topic is suggested to gather more knowledge on this topic.

Regarding prediction performance, Baesens et al. (2003) studied various state-of-the-art classification algorithms and other advanced kernel-based classification algorithms to compare how various techniques perform when eight credit datasets are applied. Among classification algorithms, LR, Linear Discriminant Analysis (LDA), KNN, NN, DT, SVM, and least-square SVM (LS-SVMs) are tested for prediction performance and robustness of algorithms in terms of area under the receiver operating characteristic curve (ROC). Even though the result suggested that advanced kernel-based classification algorithms like LS-SVM and NN showed the highest performance, LR also achieved a comparable performance as the complex algorithms. The reason is that the credit datasets used for this study did not have a strong non-linear relationship. The result suggests that the LR assumption still holds in credit scoring applications. On top of that, Dong et al. (2010) highlighted that although NN and SVM may predict better than other simple classifiers like LR, these methods may not perform consistently when the population's characteristic is changed. However, this was not the case for LR, which is why LR was mainly used the credit scoring model.

In comparison to LR, SVM does not follow the assumption on the data structure. In other words, “continuity and normal distribution” are not considered in the case of nonparametric. This helps to optimize the machine learning process robustly as outliers are not significantly affected (Hooman et al., 2013). Besides, to be known as an efficient classification algorithm, it requires to be “immune to data uncertainties and perturbations.” The reason is that real-life problems

usually contain random data or a high level of noise. To prevent this, a robust algorithm is a crucial part of solving a classification problem. Therefore, the authors studied the robustness of SVM regards to the change of dataset. The results revealed that SVM remains robust even sample training is perturbated, which was investigated both in linearly separable and non-linearly separable datasets. The reason was that SVM uses "margin maximization," which gives an appropriate regularization to separate the class (Bennett & Campbell, 2000; Trafalis & Gilbert, 2007). To strengthen the study of SVM robustness, Chen et al., (2015) went even further to propose a robust classification framework. The authors applied the perturbed subspace method (PSM) with an extensive sample space for sample training purposes. As a result, the PSM approach suggested that SVM classification's robustness had been shown and allowed to reduce the total sample training.

In addition to the robustness, Schebesch & Sleeking (2005) used SVM and LR for building credit scoring models with real-life credit data and thus found that SVM was the best model for predictions. However, the accuracy of LR was not significantly different from SVM. Another benefit of SVM is that it has a better ability to minimize Type II error compared to Artificial Neural Network (ANN) due to its "fitness function" (Li & Zhong, 2012). Although SVM gives superior prediction accuracy and minimal Type II error, interpretability is more complicated than the LR model. SVM is characterized as a "black-box" because "the functional relationships between variables" are unnecessary. Due to the characteristic of "black-box," banks cannot explain why the SVM model classified this specific borrower as a bad credit (Dong et al., 2010).

2.3 Validation

Once a specific algorithm executes the classification problem, the validation analysis on the following outputs and model predictions is necessary to gain a full image of the influence under perturbation (Lam, 2016). To analyze this, resampling techniques such as bootstrapping can be

implemented to assess a classifier's stability. The bootstrap approach allows to draw numerous samples with replacement, as the original sample is treated as a population; thus, inferences from the bootstrap sample can be gathered. This resampling technique brings a significant benefit when the available dataset is limited because it simulates many resampling data to "estimate prediction error." Another advantage is that the bootstrap method does not require underlying population distribution (Campbell & Torgerson, 1999; Dupret & Koda, 2001).

Furthermore, Xu & Goodacre (2018) mentioned that model validation is necessary for supervised learning since the prediction performance analyzes the generalization error against the unseen data. Generally, the original dataset is split into two sets for training and testing purposes. When the dataset is large enough to split into three parts, the additional set is used to validate the trained models, which is done before testing on the hold-out test. Data splitting approaches include cross-validation, bootstrapping, and resampling (Xu & Goodacre, 2018). According to Paliwal & Kumar (2009), two types of validation methods are commonly used in a credit scoring application study. The literature review referred that the K-Fold cross-validation and Hold-out sample (train-test split) appeared 50 times in previous papers, while only one study used bootstrapping technique. As a result, further research is suggested to implement other validation techniques to extend the study of bootstrapping.

Moreover, since the resampling technique's principle is to create multiple training samples from the original sample iteratively, the repetition provides multiples cases of perturbated samples to compare robustness, given an algorithm (Bischl et al., 2012). In terms of performance validation, Brown & Mues (2012) applied a more comprehensive selection of classification algorithms on a combination of leading Benelux financial institutions' credit datasets and UCI public datasets. In this study, LR, DT, NN, LDA, SVM, Random Forests, and Gradient boosting are selected to measure the area under the curve (AUC). The results revealed that ensemble methods tend to achieve higher AUC scores than other classifiers in the class imbalance

environment. Regardless of the AUC score, LR and LDA performed consistently than other methods. Marqués et al. (2013) included oversampling and under-sampling on imbalanced data to extend the effect of resampling toward class imbalance. The authors explicitly investigated the aftermath of resampling techniques on five credit datasets with LR and SVM. Based on the area under curve (AUC) evaluation metric, over-sampling techniques generally performed better than the under-sampling method. In the aspect of stability of the model, the results showed that Logistic regression and Support vector machine is robust regardless of imbalanced class ratio and resampling techniques. This confirms that resampling methods allow measuring the quality of the model. As a result, the paper shares a similar conclusion as the previous study, even when an imbalanced class is resampled with oversampling and under-sampling.

Despite the resampling technique that can also be used in the model validation as model accuracy assessment and robust check, some techniques have their advantages and pitfalls. The K-fold cross-validation technique randomly splits training and testing sets into K-fold. For instance, 10-fold CV provides ten different training and test sets for the model. The pitfall of K-fold cross-validation is that a large variance of accuracy could occur as a small number of samples are used (Bischl et al., 2012). On the other hand, Efron (1983) highlighted that not all model validation techniques have the same characteristics. Compared to K-fold cross-validation, bootstrapping does not suffer from a large variance, but it is likely to be overfitted or optimistically biased (Bischl et al., 2012; Efron, 1983). Hence, the attention of a comprehensive validation techniques should be highlighted after the model construction to check the robustness of classification and accuracy of the proposed model for "minimizing the impact of data dependency" (West, 2000).

2.4 Hypothesis

As Clancey (1984) suggested, an algorithm may not give the "right answer" all the time in terms of classification. Logistic Regression was also sensitive when noise is added, while Support Vector Machines was robust even data is perturbated by resampling and noise (Bennett & Campbell, 2000; Feng et al., 2014; Trafalis & Gilbert, 2007). Based on the existing literature, the following research hypothesis is formulated:

When the original dataset is corrupted by noise or resampled with bootstrapping, oversampling, or under-sampling, then Logistic Regression would not give robust outputs unlike Support Vector Machine.

3 Methodology

3.1 Original Dataset

The original dataset called "German Credit Data" is retrieved from the University of California Irvine (UCI) Machine Learning Repository. It consists of 7 numerical attributes, 13 categorical attributes, and one target attribute. The target attribute "Classification" contains 1 and 2 to classify between Good and Bad Credit. In other words, this dataset concerns with a binary classification problem. The numerical variables list is displayed in Table 1, as these variables are explored further in the data exploration section. The overview of the entire attributes contained in the German Credit Dataset can be found in Appendix 1. The following dataset does not have any missing values.

Attribute Name	Attribute Description	Data Type
duration	Time Frame (Month)	Integer
creditamount	Credit Amount (DM)	Integer
Installmentrate	Installment Rate (%)	Integer
pre_residencesince	Duration in Current Residence (Year)	Integer
age	Age (Year)	Integer
existingcredits	Number of Credits at Bank	Integer
peopleliable	Number of People Liable	Integer

Table 1: The list of German Credit Data's numerical variables

3.2 Data Exploration

Before the dataset is applied to create predictive modeling in a classification problem, it is crucial to have a glimpse of the dataset, precisely the balance of target attribute. When a class balance is not evenly distributed, a given Machine Learning algorithm would have a higher chance of predicting the majority class most of the time, which may return a biased prediction performance. The class imbalance problem can be mitigated with a resampling technique, which will address during the validation process (Marqués et al., 2013). According to the German Credit dataset, there are 700 instances of Good Credit and only 300 instances of Bad Credit. Thus, the ratio of the class imbalance is mild in this specific dataset.

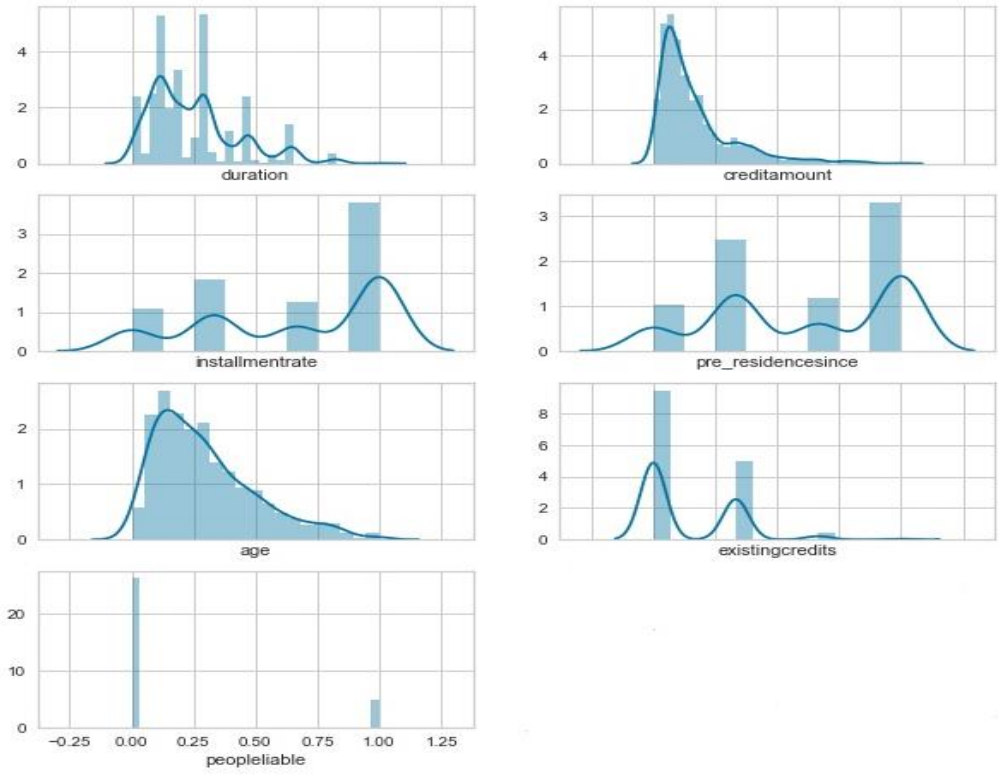

Figure 1: Original Dataset Distribution Plots

Moreover, to have a closer look at the distribution of the numerical attributes (Figure 1), each attribute's skewness is explored. The skewness of 'Installmentrate' is – 0.53 and -0.27 for 'pre_residencesince'. In other words, these variables are moderately negatively skewed. While 'duration' and 'age' features are moderately positively skewed, the skewness of these variables

is around 1. Lastly, ‘creditamount’, ‘peopleliable’, and ‘existingcredits’ variables are highly positively skewed as the coefficient of skewness is above 1.

After exploring numerical variables’ distribution, the box plot is applied to detect outliers of these variables. According to Figure 2, ‘duration’, ‘age’, and ‘existingcredits’ have several outliers. Most of the observations from ‘peopleliable’ feature have one people and very few observations have two people. Interestingly, many outliers in ‘creditamount’ are located between 7500-12,000. As this boxplot considers all the observations from both classes, it is not easy to detect which class belongs to these outliers.

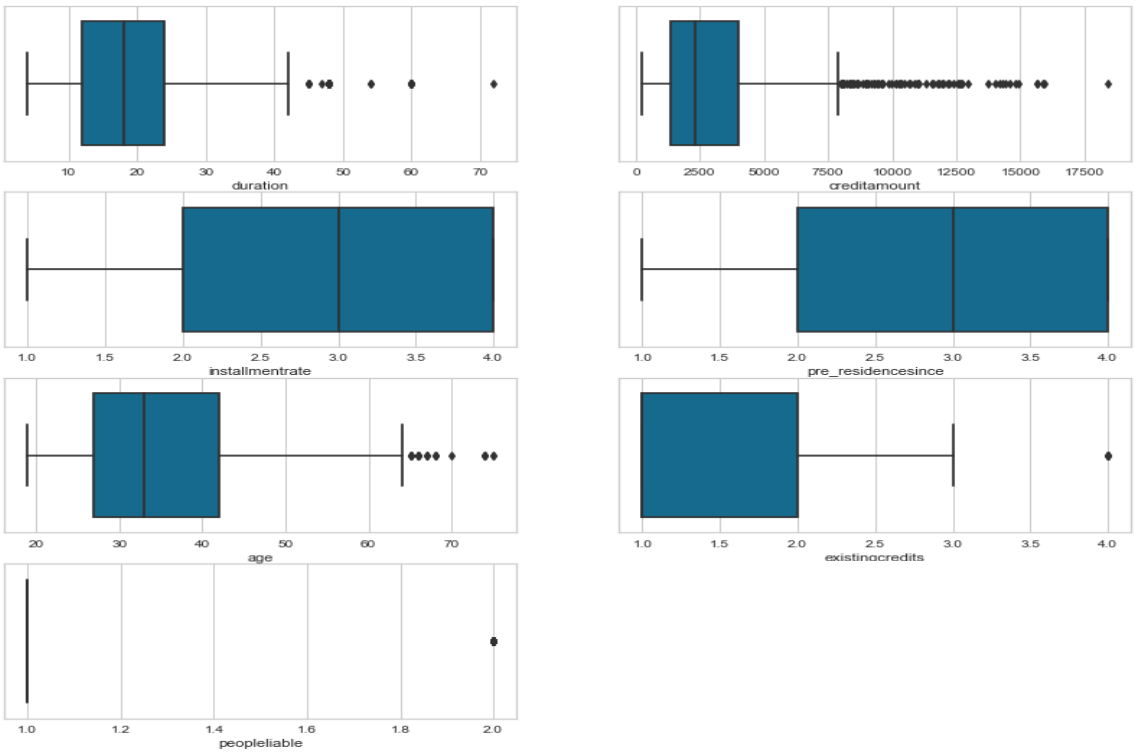

Figure 2: Original Dataset Box Plots

In Figure 3, the box plot now takes the class variable into account. The blue box plot represents a negative class (Good Credit), and the green box shows a positive class (Bad Credit). Only the negative class of ‘duration’ has several outliers in this case. Both ‘existingcredits’ and ‘age’ variables have a similar number of outliers. In terms of ‘creditamount’ variable, both classes

have a similar median. The vast number of outliers from Class 1 is located from 10,000 to 15,000, while Class 0 is located in the range of 7,000 to 14,000.

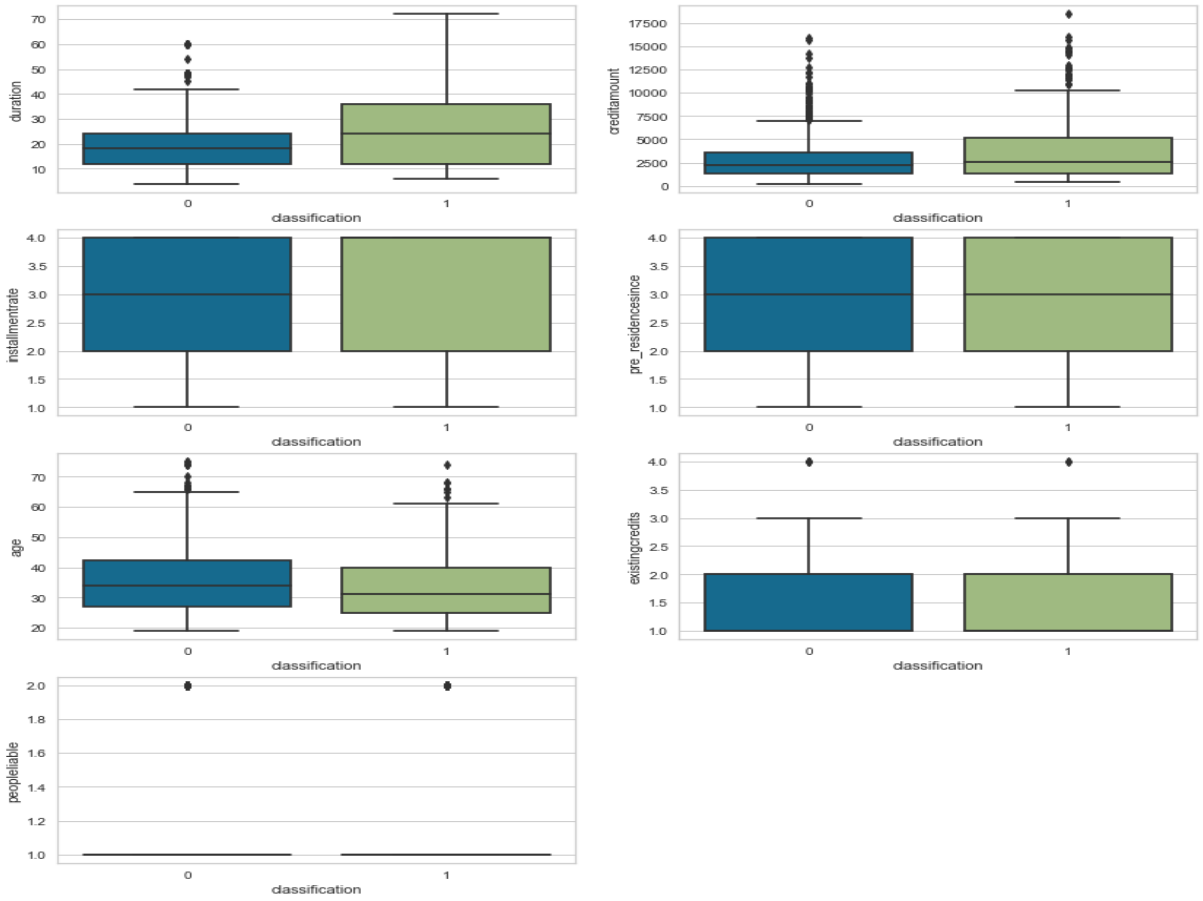

Figure 3: Box Plots Between Class 0 & Class 1

To sum it up, these continuous variables' skewness implies that these variables do not follow a normal distribution and contain several outliers. However, the outliers are not deleted or replaced with other statistical methods such as mean or median. The reason is that the benchmark study by Ala'Raj & Abbod (2015) did not explicitly handle outliers. As this study aims to follow the method to keep the consistency of the study, the outliers are not dealt with during the data cleaning process.

3.3 Data Pre-Processing

Before the original dataset is split into training and test for the predictive modeling and robust check, the categorical variables are transformed to numeric values. Currently, the categorical variables are defined as "Existing-Checking A11, A12, A13, and A14", so the ML algorithms would not be able to train these instances unless it transformed to a numerical value. The One Hot Encoding feature is executed on Python to convert categorical variables into numerical variables, and then each categorical variable is transformed into dummy variables. As this paper follows Ala'Raj & Abbod (2015) procedure, the dataset is split the Training set into 80%, and the Testing set into 20%. The train set has 562 negative class (Good Credit) and 238 of positive class (Bad Credit), while the test set has 138 of negative class and 62 positive class. After, Train-Test data split (X_train, X_test, y_train, y_test), X_train, and y_train are trained to fit Logistic Regression (LR) classifier and Support Vector Machine (SVM) classifier. The classification from this will be used for robustness check against resampled datasets. Moreover, the original dataset is scaled to the range 0-1 with normalization by taking each attribute's maximum value. Then, each value of the corresponding column is divided by the maximum value as these numerical variables do not follow the same scale.

3.4 Overall Pipeline

The study's primary goal is to check the robustness of a classifier; the baseline model's procedure is repeated to the resampled models. Referring to Figure 4, the original dataset is initially preprocessed with the transformation of categorical variables, data separation, and normalization. After that, the original training set is used to fit with Logistic regression and Support Vector Machine, then predicted on training sample to gather classification from these classifiers. In terms of the bootstrap approach, the train set is applied to draw observations with replacement, given many iterations. In this case, 100, 1000, 5000, and 10,000 iterations are

implemented. Afterward, the drawn samples are normalized again. These sampling data follow the same Machin Learning algorithms' configuration as the original method to gather classification. Unlike the original method, the sampled data is used to train with the following algorithms. The same logic applies to random oversampling and under-sampling. However, when a Gaussian Noise is added to the original train set, the noisy data is not normalized.

For robust check, the original class is compared with the resampled class to evaluate a classifier's stability. As an evaluation metric, the paper created a 2x2 matrix called "Disagreement confusion matrix' that follows a similar characteristic as the typical confusion matrix. In addition to the robust test, all the resampled models and baseline models are always tested on the original test set to check model accuracy. In terms of evaluation metrics, accuracy score and confusion matrix measure the total number of misclassifications on an unseen-data.

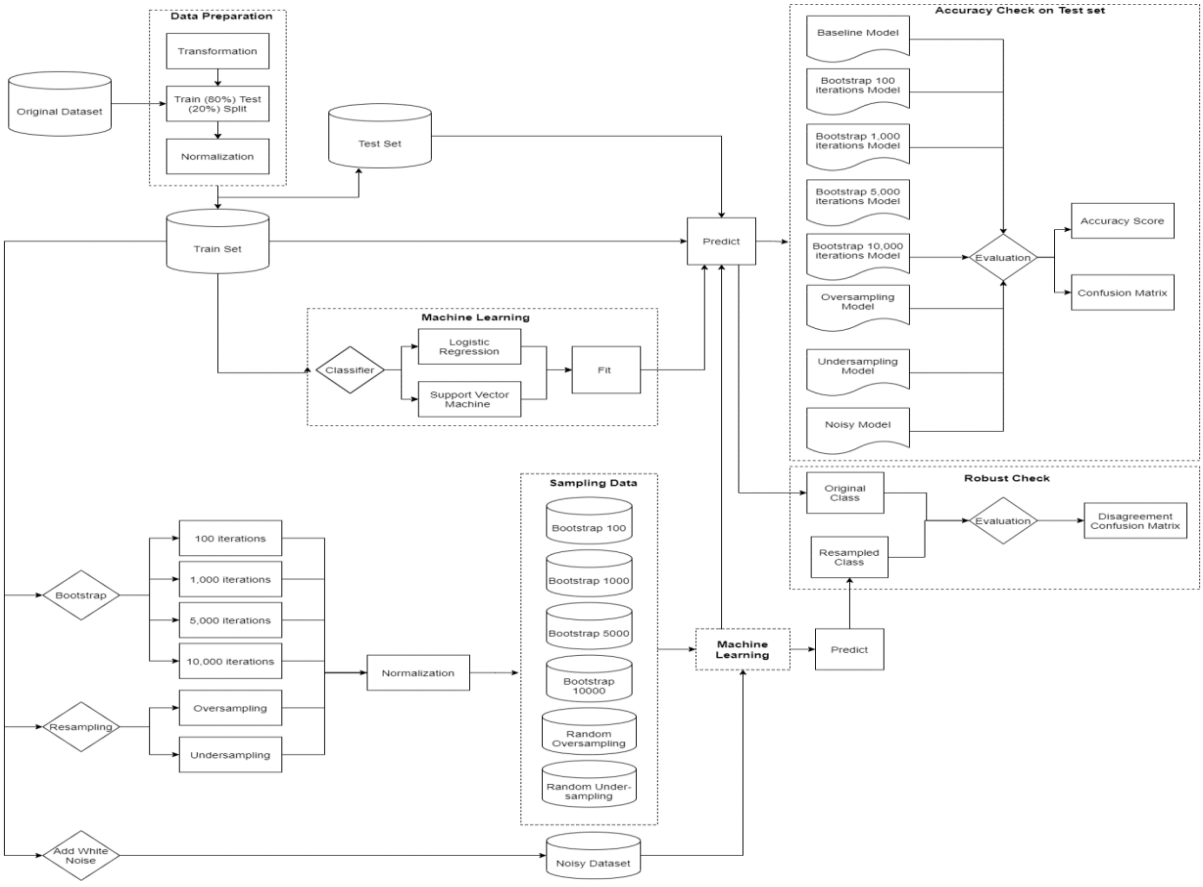

Figure 4: Overall Pipeline

3.5 Predictive Model

In the Machine Learning domain, supervised and unsupervised learning techniques are used based on the dataset's characteristics. The principle of the supervised learning method is to predict a target label from given inputs and outputs. Then, the predictive model is evaluated based on the unseen data. Within supervised learning approaches, classification and regression are widely used based on the types of target variables. If a target feature contains continuous outputs, then the regression is the appropriate approach to apply it.

In contrast, the classification task is more suitable with binary form. On the other hand, unsupervised learning is applied when the output does not exist or unknown. Thus the learning algorithm extracts insights only based on the given input (Müller & Guido, 2017). The paper previously mentions that the German credit dataset's target variable contains two class labels, suggesting that the binary classification algorithm is the most appropriate method.

3.5.1 Logistic Regression (LR)

Logistic regression (LR) is a supervised learning algorithm that uses to solve classification and regression tasks. It is a statistical modeling approach that transforms linear regression equation into the logistic function, which computes probability regarding a dichotomy or binary dependent variable (Kleinbaum & Klein, 2010). In the mathematical form, linear regression equation (1) specifies as z, where α = intercept, βi = slope, and Xi = independent variables. The sum of equation fits into logistic function $f(z)$ (2), where e = exponential function to draw $f(z)$ values between 0 and 1. Thus, if an instance has an estimated probability $f(z)$ that is smaller than 0.5, then the LR model classifies as negative class, otherwise it classifies as positive class if the outcome is higher than 0.5 (Géron, 2019).

$$z = \alpha + \beta 1X1 + \beta 2X2 + \cdots + \beta kXk \qquad (1)$$

$$f(z) = \frac{1}{1 + e^{-Z}} \qquad (2)$$

$$= \frac{1}{1 + e^{-(\alpha + \Sigma \beta i Xi)}}$$

Although the assumptions of LR is similar to linear regression, logistic regression has slightly different assumptions than linear regression. First, a target variable of binary logistic regression requires a binary class. Second, no requirement for a linear relationship between the response and explanatory variables. Third, observations cannot be repeated and need to be independent. Fourth, independent variables are not correlated to each other. Fifth, the association between independent variables and log odds requires linearity (Hooman et al., 2013; Schreiber-Gregory, 2018).

As Python supports the scikit-learn machine learning package, this paper uses logistic regression from the package. The LR package has several parameters such as C regularization, penalty, tolerance, class weight, and max iterations. Despite the fact that Ala'Raj & Abbod (2015) did not specify logistic regression parameters, this paper had to find relevant parameters that generate similar outcomes as Ala'Raj & Abbod (2015). After trial and errors, *class_weight* sets as *None* to avoid the algorithm automatically adjust the class balance ratio, *random_state* sets at 10 to reproduce the same result, and *max_iter = 1000* converges LR 1000 times. The given parameters returned 76% of the accuracy score close to the benchmark study (75.5%). Therefore, these parameters are explicitly used throughout the study.

Logistic Regression = LogisticRegression (class_weight=None, random_state=10, max_iter=1000)

3.5.2 Support Vector Machine (SVM)

Vapnik (1998) developed the Support Vector Machine (SVM) technique in 1995 to solve "function estimation problems" such as pattern classification, nonlinear classification, and regression. SVM is another type of supervised learning algorithm that works relatively well to classify two classes regardless of dataset's complexity. However, it performs relatively better with small-medium size datasets. On top of that, SVM can classify both linear and non-linear data into high dimensional spaces based on the kernel function. In other words, SVM can work with samples that contain many features and are not only limited to linear data (Géron, 2019). In terms of linear SVM, the idea is to create a linear discriminant line or a hyperplane to separate into two classes. SVM then aims to make the margin as wide as possible around the hyperplane.

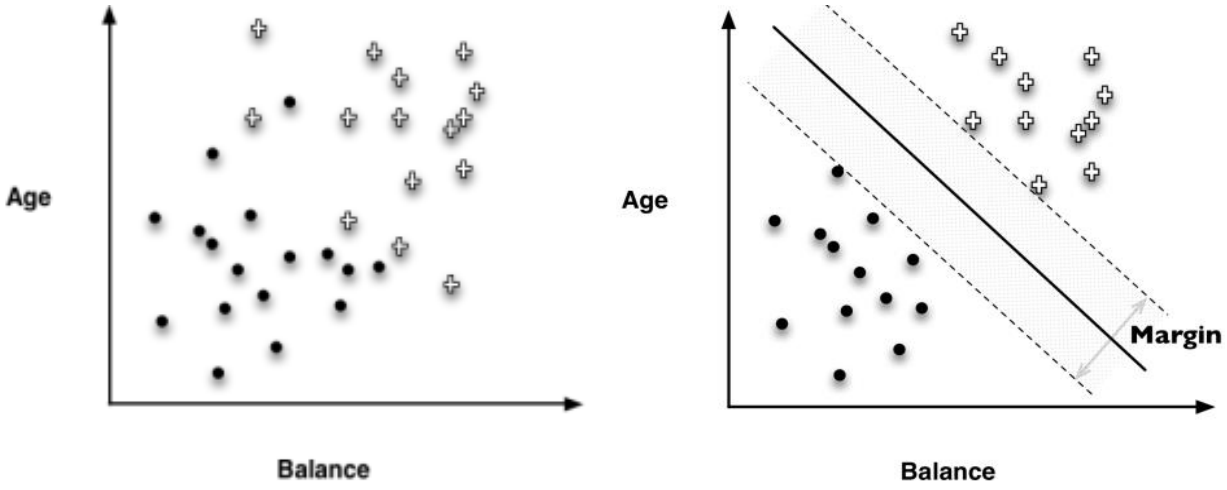

Figure 5: Support Vector Machine (Provost & Fawcett, 2013)

Referring to Figure 5, the figure's left-hand side shows all the instances of Age and Balance based on two classes. Few instances are located on the incorrect side of the decision boundary. To solve this, SVM maps into two-dimensional space with a hyperplane and margin, where the bold line refers to hyperplane and dashed lines that are parallel to the hyperplane indicate margin. Then, instances on the wrong side of the decision boundary and around the margin are removed to separate two classes linearly. When this logic applies in predictive modeling, SVM

fits a model with training data to create a similar distribution as testing data (Provost & Fawcett, 2013).

In this study, the SVM kernel is selected as 'linear' as Ala'Raj & Abbod (2015) explicitly stated that linear kernel was used. This is the list of configurations of the Support Vector Machine.

Support Vector Machine = SVC (kernel = 'linear')

3.6 Resampling Methods

3.6.1 Bootstrapping

Bootstrapping is a part of the resampling method, which is initially introduced by Efron (1983). It is a non-parametric method that repeatedly draws n-size of random samples from the original dataset. In other words, some of the observations are randomly selected multiple times in a resampled dataset. The idea of bootstrap is to estimate a sample distribution or gather other relevant inferences from the original sample instead of making incorrect assumptions of the population. In this case, the original sample is treated as the population. Referring to Figure 6, various resamples are created from the original sample, which means that multiple independent datasets are available after bootstrapping (Campbell & Torgerson, 1999).

Figure 6: Bootstrap Approach (Campbell & Torgerson, 1999)

In addition to multiple datasets, the first resample contains multiple 1 and 5s (Figure 6), while the second resample has various 2 and 3s. This example shows that some instances are included

several times, but not all the instances are drawn from the original sample (Campbell & Torgerson, 1999).

Singh & Sedory (2011) recommended that sufficient iteration is 500 to find a standard deviation confidence interval. However, this study uses from 100 iterations to 10,000 iterations to have complete knowledge of the effect bootstrap. In the example of 100 iterations, 100 independent resamples are drawn from the original sample. After that, the aggregation of 100 resamples is used to compute estimates of confidence intervals and standard errors, given a specific sample statistic such as mean. Thus, the standard error of the mean shows a bootstrapped sample statistic instead of the population statistic.

3.6.2 Random Oversampling & Under-Sampling

Another approach of data perturbation is with oversampling and under-sampling. The basic concept of oversampling is to resample the observations from the minority class until it matches the number of the positive class. Thus, the balance between the two classes is equally balanced. In respect of random oversampling, the following technique randomly replicates the minority class to eliminate the class imbalance problem. Therefore, the sum of oversampled observations becomes more extensive than the original dataset (Barandela et al., 2003). As the original training set contains 562 instances of the negative class and 238 instances of the positive class, the positive class's number increases to 562 by repeating some of 238 instances of the positive class. After the oversampling, the total number of instances become 1124 with 50:50 of negative and positive class ratio.

In contrast to oversampling, random under-sampling randomly downsizes some of the majority class instances to create a symmetric class distribution between the majority class and minority class. This decreases the total number of observations compared to the original dataset, which may face a loss of information. (Ghorbani & Ghousi, 2020). In this study, the under-sampled

dataset's total instances become 476 since only 238 minority instances are contained in the original training sample.

3.7 Gaussian Noise

In addition to the implementation of various resampling techniques to measure the baseline classifier's robustness, another technique to measure the robustness of the classification is by adding noise to the original training set. The creation of noise simulates a real-life scenario since the real-life datasets may contain unexplainable random instances due to human errors or machine errors. Therefore, the inheritance of noise might affect the classification and the accuracy score of the model (Singh & Sedory, 2011). One way to test this theory is by adding a Gaussian Noise, which transforms into a normal distribution. In the present study, the numerical variables are selected to change the data distribution into a normal distribution. First, each numerical variable's mean and standard deviation is gathered from the original training set. These values were then used to create 800 random values, given Gaussian distribution. After the random noise is created for each variable, it is then added to the training sample.

3.8 Evaluation Metrics

Various evaluation metrics are available to examine the quality of classification predictions. In this study, an accuracy score and confusion matrix are applied to evaluate a prediction model's generalization performance. The accuracy score measures the overall performance of the model. Given the total number of decisions, the proportion of correct decisions are calculated, in other words, the sum of True Positive (TP) and True Negative (TN) is divided by the sum of True Positive, True Negative, False Positive (FP), and False Negative (FN) (Provost & Fawcett, 2013).

$$Accuracy\ Score = TP+TN / TP+TN+FP+FN$$

In addition to the accuracy score, the confusion matrix evaluates how the actual class label differs from the predicted class label in a contingency table. The confusion matrix can be displayed in n x n matrix. Since the German Credit dataset's target variables have two classes, a 2 x 2 confusion matrix is used (Provost & Fawcett, 2013). Referring to Table 2, the total number of TP is calculated when the predicted positive class matches with the actual positive class and vice versa for TN. FP (Type I) occurs when the actual negative class is misclassified as a positive class. On the other hand, FN appears when the actual positive class is misclassified as a negative class (Zeng, 2020). Moreover, Table 3 shows the "Disagreement Confusion Matrix" that shares the same aspect of the confusion matrix. However, it instead focuses on the comparison of classification from the baseline method against the resampled method.

	Predicted Class 0	Predicted Class 1
Actual Class 0	TN	FP
Actual Class 1	FN	TP

Table 2: Confusion Matrix

	Resampled Classifier Class 0	Resampled Classifier Class 0
Baseline Classifier Class 0	TN	FP
Baseline Classifier Class 1	FN	TP

Table 3: Disagreement Confusion Matrix

4 Results

4.1 The Baseline Model

After the original training set is fitted with Logistic regression and Support Vector Machine, Baseline LR classified 626 instances of Class 0 and 174 observations for Class 1. A similar output was returned by SVM as well. In this case, 619 instances of Class 0 were assigned baseline SVM, while 181 observations for assigned for Class 1 (Table 4). Given the original training set containing 562 instances for Class 0 and 238 instances for Class 1, 64 observations were not correctly assigned by LR. On the other hand, SVM misassigned 57 instances. This classification's outputs will be used as a benchmark to conduct a pairwise comparison for the resampled datasets.

Classification	0	1
Baseline LR	626	174
Baseline SVM	619	181

Table 4: Baseline Model Classification

After the classification, the Baseline models were validated against the out-of-sample set. In Table 5, Logistic Regression scored 76%, and Support Vector Machine scored 76.5%, which indicates that the selection of an algorithm does not significantly influence the accuracy performance. The cause of this result is that SVM made less during classification, which potentially allowed to get a slightly higher accuracy score than LR.

Accuracy Score	LR	SVM
Original Dataset	76%	76.5%

Table 5: Accuracy Score of the Original Dataset

In terms of the confusion matrix (Table 6), LR correctly classified 34 instances for the positive class, while SVM accurately predicted 37 instances, which is 3 more instances than the LR. However, LR's TN has two extra instances than SVM's TN. In other words, LR was able to predict marginally better than SVM. Regarding Type I error (FP), LR made less error on predicting positive class than SVM did. Type I error occurred when the model predicted a bad credit although the person had good credit. Lastly, the LR model misclassified 3 additional instances towards the negative class (Type II error), even though the actual value was positive.

The proposed result will also be used as a performance comparison.

Confusion Matrix	TP	TN	FP	FN
Original Dataset LR	34	118	20	28
Original Dataset SVM	37	116	22	25

Table 6: Confusion Matrix of the Original Dataset

4.2 Bootstrapping

Referring to Table 7, even when the original training sample was bootstrapped with a different number of iterations, the average value of variables did not vary against the original sample. The standard deviation of the average values also did not deviate significantly within bootstrapped samples. Between 1000 – 10,000 iterations, the mean and standard deviation of the variables are very marginal, so the changes become significantly smaller as the number of iterations increases.

	Duration	Creditamount	Installmentrate	Pre_residencesince	Age	Existingcredits	Peopleliable
Original: Mean	0.292 (0.168)	0.179 (0.153)	0.742 (0.282)	0.712 (0.276)	0.473 (0.153)	0.352 (0.146)	0.578 (0.182)
Bootstrap 100: Mean	0.292 (0.006)	0.179 (0.006)	0.741 (0.011)	0.710 (0.009)	0.472 (0.006)	0.352 (0.005)	0.577 (0.006)
Bootstrap 1000: Mean	0.292 (0.026)	0.179 (0.006)	0.742 (0.010)	0.7111 (0.010)	0.472 (0.005)	0.352 (0.005)	0.578 (0.006)
Bootstrap 5000: Mean	0.292 (0.006)	0.179 (0.005)	0.742 (0.010)	0.7112 (0.010)	0.473 (0.005)	0.352 (0.005)	0.578 (0.006)
Bootstrap 10000: Mean	0.292 (0.006)	0.179 (0.005)	0.742 (0.010)	0.7112 (0.010)	0.473 (0.005)	0.352 (0.005)	0.578 (0.006)

Table 7: Mean of numerical variables (Original VS Bootstrap)

Moreover, the bootstrapped mean's confidence interval was constructed to observe how the mean of numerical variables from the bootstrapped samples would likely exist in the original sample. Since Bootstrap 1000 and Bootstrap 5000 have a similar distribution (Appendix 3) as the 10,000 iterated samples, only Bootstrap 100 and Bootstrap 10,000 samples are explored closely in this section. In Figure 7, the average value of numerical attributes is plotted in a histogram. Bootstrap 100 is illustrated on the left-hand side of Figure 7, while Bootstrap 10,000 is displayed on the right side. The two red lines indicate a 95% confidence interval range, the black line shows the average of the corresponding variables, and the purple line displays the mean of the original variable. When 100 iterations were implemented to draw samples from the original training sample, most of the variables are normally distributed to some extent except 'creditamount' (Figure 7 Left). In this case, 'creditamount' have a multimodal distribution.

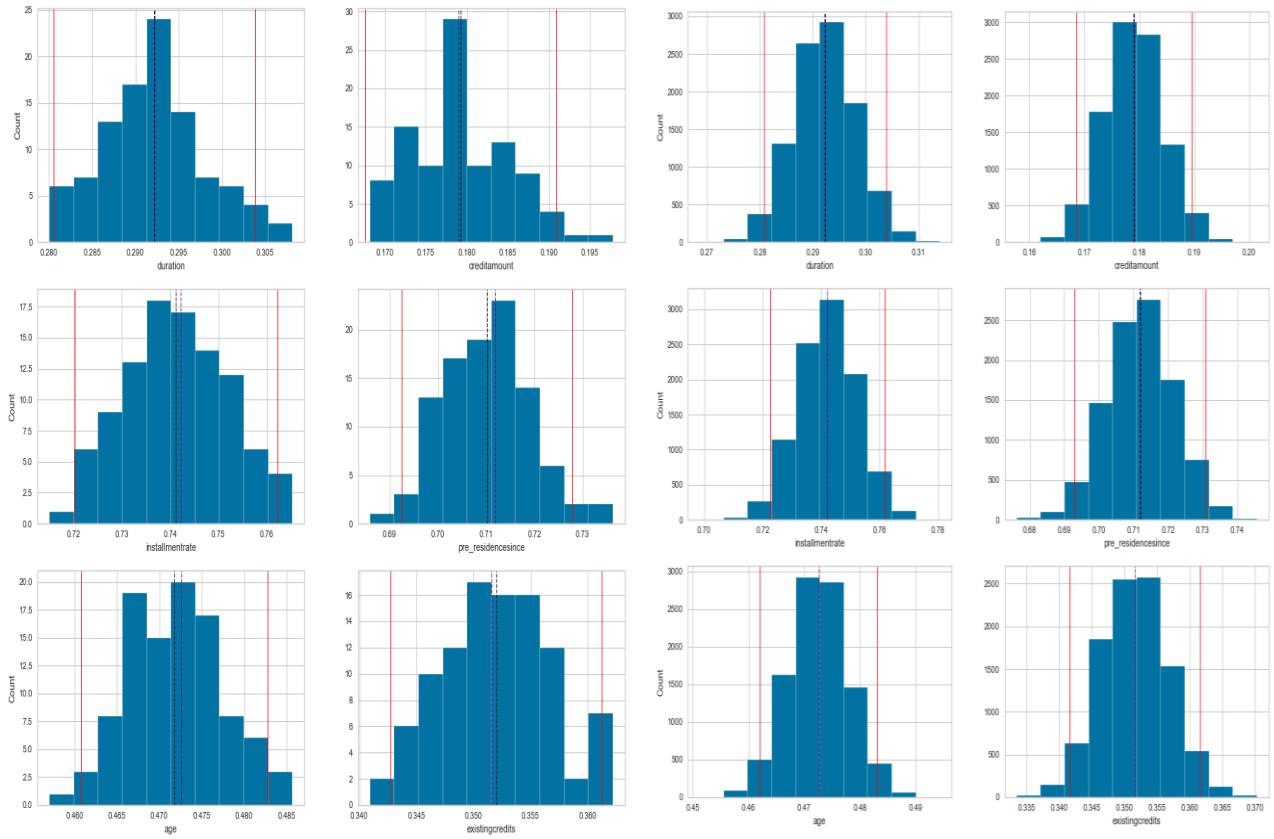

Figure 7: Histogram of Bootstrap 100 iterations (Left) & Histogram of Bootstrap 10,000 iterations (Right)

However, as the number of iterations increases, the histogram's shape became more symmetric and normally distributed (Figure 7 Right). The reason is that more observations were repeated as the iterations went up, hence it was likely to have a normal distribution. Besides, the original sample's mean also became very similar to the Bootstrap sample's mean as the frequency of iterations increased. The result suggests that the Bootstrap sample is sufficient to draw inferences about the original training sample.

A 95% Bootstrap confidence interval (CI) is calculated for each variable in addition to the histogram. For example, the lower boundary of 'duration' from Bootstrap 100 is calculated by taking the average of Bootstrap 100's average duration. Then the multiplication of 1.96 and the standard deviation of Bootstrap 100's average duration is subtracted. For the upper boundary, the multiplication is added instead. Thus, the lower boundary and the upper boundary are 0.281, 0.304, respectively. Given the boundary, the 95% confidence interval of Bootstrap 100 'duration' lies between 0.281 and 0.304. Hence, the original sample's average duration would lie somewhere between this range, which is visible in Figure 7 Left (Appendix 2). The same pattern also can be found for the rest of the variables.

B100_Duration_Lower Boundary:

*Mean (B100_Duration_Average) – 1.96 * Standard Deviation (B100_Duration_Average)*

B100_Duration_Upper Boundary:

*Mean (B100_Duration_Average) + 1.96 * Standard Deviation (B100_Duration_Average)*

4.2.1 Robust Check

	Bootstrap 100 LR 0	Bootstrap 100 LR 1
Baseline LR 0	193.95 (11.63)	36.78 (7.19)
Baseline LR 1	20.33 (5.03)	43.59 (8.53)

Table 8: Bootstrap 100 LR - Disagreement Confusion Matrix

	Bootstrap 1000 LR 0	Bootstrap 1000 LR 1
Baseline LR 0	193.99 (10.71)	36.48 (6.47)
Baseline LR 1	20.01 (4.92)	43.60 (7.18)

Table 9: Bootstrap 1000 LR - Disagreement Confusion Matrix

	Bootstrap 5000 LR 0	Bootstrap 5000 LR 1
Baseline LR 0	193.36 (10.98)	36.85 (6.67)
Baseline LR 1	20.27 (5.19)	43.79 (7.21)

Table 10: Bootstrap 5000 LR - Disagreement Confusion Matrix

	Bootstrap 10,000 LR 0	Bootstrap 10,000 LR 1
Baseline LR 0	193.39 (10.93)	36.75 (6.73)
Baseline LR 1	20.30 (5.07)	43.76 (7.17)

Table 11: Bootstrap 10,000 LR - Disagreement Confusion Matrix

After exploring the statistics of Bootstrap samples, each sample is trained and fitted in the same condition as the baseline models for the robust check. It was previously mentioned that Bootstrap repeats some of the observations from the original training sample; hence unique observations are considered in the disagreement confusion matrix. From Table 8 to Table 11, which shows a total classification of Logistic regression for each Bootstrap. In the result of Bootstrap 100 LR, a total of 57 instances were classified differently on average, which has about 7.1% (57/800) of disagreement between Baseline LR and Bootstrap 100 LR. The almost same result also occurred with other iterations as well, such as 1000, 5000, 10,000. As a result, Logistic regression is quite robust even the original training sample is perturbated by the bootstrap method.

	Bootstrap 100 SVM 0	Bootstrap 100 SVM 1
Baseline SVM 0	190.07 (11.27)	37.47 (6.96)
Baseline SVM 1	24.21 (5.24)	42.90 (8.37)

Table 12: Bootstrap 100 SVM: Disagreement Confusion Matrix

	Bootstrap 1000 SVM 0	Bootstrap 1000 SVM 1
Baseline SVM 0	190.33 (10.71)	37.73 (6.64)
Baseline SVM 1	23.67 (5.32)	42.35 (7.24)

Table 13: Bootstrap 1000 SVM: Disagreement Confusion Matrix

	Bootstrap 5000 SVM 0	Bootstrap 5000 SVM 1
Baseline SVM 0	189.66 (10.92)	37.88 (6.81)
Baseline SVM 1	23.98 (5.55)	42.75 (7.14)

Table 14: Bootstrap 5000 SVM: Disagreement Confusion Matrix

	Bootstrap 10,000 SVM 0	Bootstrap 10,000 SVM 1
Baseline SVM 0	189.74 (10.96)	37.84 (6.82)
Baseline SVM 1	23.96 (5.43)	42.67 (7.06)

Table 15: Bootstrap 10,000 SVM: Disagreement Confusion Matrix

Similar to LR, SVM was also not significantly influenced by the iterations of Bootstrap. In this scenario, only 7.6% (61/800) disagreed with Baseline SVM and Bootstrap SVM (Table 12 – Table 15). An additional 0.1% of disagreement has occurred under SVM. Baseline SVM and Bootstrap SVM's total agreement is about the same as Logistic regression in terms of mean and standard deviation. Therefore, the output of LR and SVM are relatively stable to the Bootstrap method.

4.2.2 Accuracy

Accuracy Score	LR	SVM
Baseline	76%	76.5%
Bootstrap 100	41% (0.001)	41% (0.000)
Bootstrap 1000	41% (0.001)	41% (0.000)
Bootstrap 5000	41% (0.001)	41% (0.000)
Bootstrap 10,000	41% (0.001)	41% (0.000)

Table 16: Accuracy Score (Original LR & SVM VS Bootstrap LR & SVM)

Despite LR and SVM are quite robust to Bootstrap, the model prediction on a test set tells a different story. Referring to Table 16, the accuracy score of Bootstrap LR and SVM decreased by 35% compared to the baseline models. In other words, given approximately 7% disagreement, the accuracy score went down by 35% against the baseline models. Regardless of the number of iterations, all the models achieved almost identical accuracy. The possible

explanation of these occurrences is that both LR and SVM previously had a very close range of disagreement for classification, which leads similar model accuracy score. Disagreed classification was also repeated in the Bootstrap sample, which affected the overall prediction performance.

Confusion Matrix	TP	TN	FP	FN
Baseline LR	34	118	20	28
Bootstrap 100 LR	10.01 (0.10)	72.01 (0.10)	65.99 (0.10)	51.99 (0.10)
Bootstrap 1000 LR	10.01 (0.10)	72.02 (0.10)	65.98 (0.10)	51.99 (0.10)
Bootstrap 5000 LR	10.00 (0.10)	72.01 (0.10)	66.00 (0.10)	52.00 (0.10)
Bootstrap 10,000 LR	10.00 (0.10)	72.00 (0.15)	65.99 (0.15)	51.99 (0.10)

Table 17: Confusion Matrix (Original LR VS Bootstrap LR)

Confusion Matrix	TP	TN	FP	FN
Baseline SVM	37	116	22	25
Bootstrap 100 SVM	10 (0)	72 (0)	66 (0)	52 (0)
Bootstrap 1000 SVM	10 (0)	72 (0)	66 (0)	52 (0)
Bootstrap 5000 SVM	10 (0)	72 (0)	66 (0)	52 (0)
Bootstrap 10,000 SVM	10 (0)	72 (0)	66 (0)	52 (0)

Table 18: Confusion Matrix (Original SVM VS Bootstrap SVM)

In terms of the confusion matrix, all Bootstrap LR models have a similar result. Table 17, the Bootstrap 100 LR model correctly predicted 10 instances of positive class, while 72 instances for negative class. The correct prediction gap between the Baseline LR model and Bootstrap LR 100 model is quite broad. On the other hand, the model made 45 more Type I error, and 23 for Type II errors than the Baseline model, which implies the change of dataset with Bootstrap significantly affects the level of misclassification. Moreover, all Bootstrap SVM models have an exact confusion matrix regardless of iteration (Table 18). The potential cause of this result would be due to a specific mechanism of SVM. However, further investigation on this odd pattern should be conducted in future research.

4.3 Oversampling

Concerning the skewness of the oversampled dataset (Figure 8), all the numerical variables have similar skewness as the original sample. The distribution of the original variables shows in green, while the oversampled variables display in blue. Previously, the original 'installmentrate' and 'pre_residencesince' were moderately negatively skewed, while 'creditamount', 'peopleliable', and 'existingcredits' variables were highly positively skewed. A similar pattern is also apparent in the oversampled dataset, as the duplication of the positive class instances did not necessarily shift the distribution against the minority class. The result indicates that the oversampled datasets are less sensitive in terms of distribution.

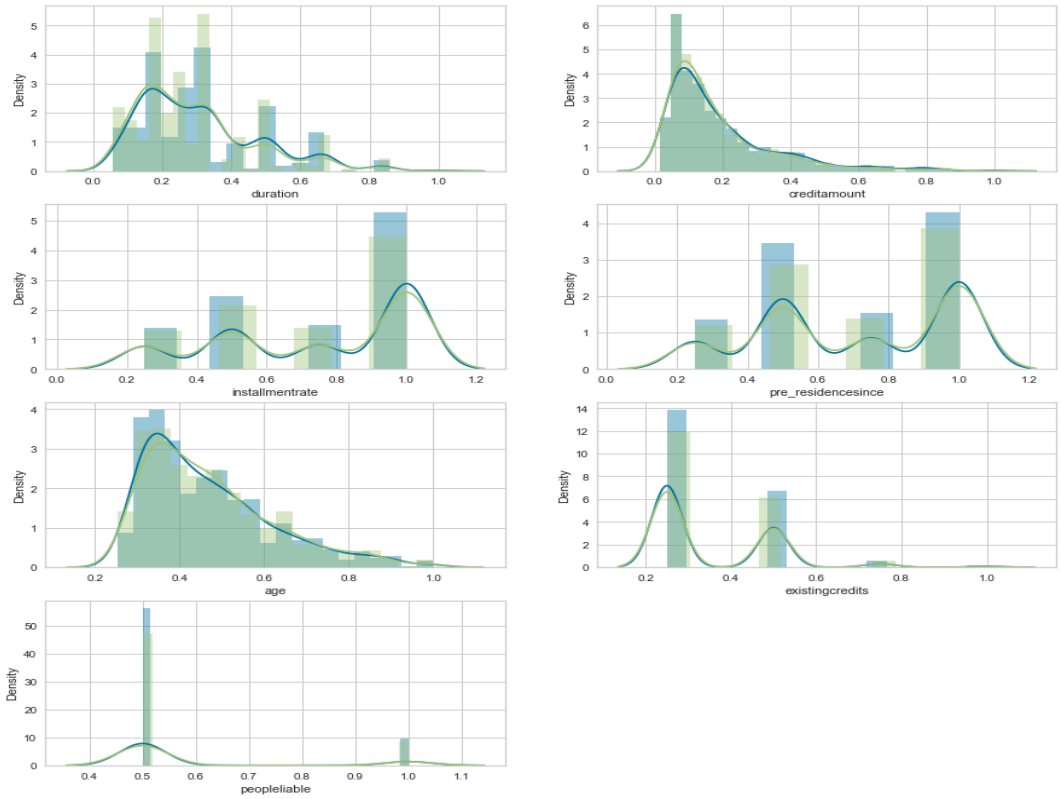

Figure 8:The distribution plots of the oversampled dataset

4.3.1 Robust Check

	Oversample LR 0	Oversample LR 1
Baseline LR 0	72	97
Baseline LR 1	0	173

Table 19: Oversample LR: Disagreement Confusion Matrix

	Oversample SVM 0	Oversample SVM 1
Baseline SVM 0	69	102
Baseline SVM 1	1	170

Table 20: Oversample SVM: Disagreement Confusion Matrix

During oversampling technique, some of the positive class instances are repeated to balance the class ratio between the minority class. Similar to Bootstrap, unique observations are recorded under the disagreement confusion matrix. In Table 19, Oversample LR had 12.1% (97/800) against Baseline LR. Interestingly, there was no disagreement between oversample LR 0 and Baseline LR 1, while 97 disagreements between oversample LR 1 and Baseline LR 1. The reason is that oversampled dataset had more instances of positive class compared to the original dataset, which led to a conflict between the original sample and oversample.

Moreover, Oversample SVM had 12.9% (103/800) instances of disagreement against Baseline SVM (Table 20). In comparison to LR, SVM had additional 6 instances of disagreement that includes 1 conflict between Baseline SVM 1 and Oversample 0. This means that Baseline SVM is assigned differently for Class 1 against Oversample SVM. Although these two algorithms had more disagreement under oversampled dataset than the Bootstrap sample, 12% still indicates moderate robustness.

4.3.2 Accuracy

Accuracy Score	LR	SVM
Baseline	76%	76.5%
Oversample	73.5%	75%

Table 21: Accuracy Score (Original LR & SVM VS Oversample LR & SVM)

Confusion Matrix	TP	TN	FP	FN
Baseline LR	34	118	20	28
Oversample LR	51	96	42	11
Baseline SVM	37	116	22	25
Oversample SVM	52	98	40	10

Table 22: Confusion Matrix (Original LR & SVM VS Oversample LR & SVM)

Given 12.1% disagreement between Baseline LR and oversample LR, the oversample LR model achieved a slightly lower accuracy score than the Baseline model. In this case, the model was able to predict better on positive class. According to Table 22, the number of TP increased by 17 as the model had more positive class observations. This means that the model made less generalization error on the positive class, but it also means that the model misclassified more instances for the negative class (FP). Moreover, the oversample SVM model's accuracy score also did not deviate much from the baseline model due to the same reason as previously mentioned. Overall, the accuracy score of the oversampled models was not notably affected by 12.1% of disagreement.

4.4 Under-sampling

The under-sampled variables' distribution is plotted to go through the same procedure as the oversampling technique (Appendix 4). Since the under-sampled distributions are also quite like the original sample, further exploration of the data distribution is not conducted.

4.4.1 Robust Check

	Under-sample LR 0	Under-sample LR 1
Baseline LR 0	238	92
Baseline LR 1	0	146

Table 23: Under-sampling LR: Disagreement Confusion Matrix

	Under-sample SVM 0	Under-sample SVM 1
Baseline LR 0	234	95
Baseline LR 1	4	143

Table 24: Under-sampling SVM: Disagreement Confusion Matrix

As the sample's total size decreased to 476 and none of the observations are repeated, it is more logical to compute a disagreement ratio on the available number of observations. Referring to Table 23, Under-sample LR has 19.33% (92/476) of disagreement against Baseline LR, which indicates that Logistic regression is not very stable when the original training sample is under-sampled. A similar outcome is also apparent under Under-sample SVM, which has a 20.80% conflict (Table 24). A possible explanation is that only a small sample size was available for these classifiers, which caused inconsistent outputs compared to the Baseline classifiers. Although the two classes were balanced, the impact of class balance was not significant enough to enhance the classifier’s stability. Therefore, the output of LR and SVM are very robust to the under-sample dataset.

4.4.2 Accuracy

Accuracy Score	LR	SVM
Baseline	76%	76.5%
Undersample	72%	69%

Table 25: Accuracy Score (Original LR & SVM VS Under-sample LR & SVM)

Confusion Matrix	TP	TN	FP	FN
Baseline LR	34	118	20	28
Undersample LR	50	94	44	12
Baseline SVM	37	116	22	25
Undersample SVM	47	91	47	15

Table 26: Confusion Matrix (Original LR & SVM VS Under-sample LR & SVM)

Although LR and SVM are not robust by under-sampled datasets, these classifiers interestingly did not poorly on predictions than the baseline models. One significant difference from the oversampled dataset is that the accuracy score decreased by 4% under the LR model and 7.5% with the SVM model (Table 25). The leading cause of this result is due to the characteristic of the under-sampling technique. Since most majority class instances are randomly decreased, the models had fewer values to capture the majority class's pattern.

Furthermore, the under-sampled models significantly misclassified the negative class compared to the original models. Referring to Table 26, the under-sampled LR model correctly classified 50 positive classes out of 62 on the original test set. However, 44 instances are misclassified as a positive class, although they are part of the negative class. On top of that, the model predicted only 94 negative class instances out of 138 instances, but the Type II error was reduced by 15 compared to the original LR model. This kind of results suggests that the loss of information on the negative class skeptically impacted the prediction of the negative class's ability. The comparable result also showed in the SVM model.

4.5 Gaussian Noise

4.5.1 Robust Check

	Gaussian Noise LR 0	Gaussian Noise LR 1
Baseline LR 0	603	23
Baseline LR 1	13	161

Table 27: Gaussian Noise LR: Disagreement Confusion Matrix

	Gaussian Noise SVM 0	Gaussian Noise SVM 1
Baseline LR 0	591	28
Baseline LR 1	20	161

Table 28: Gaussian Noise SVM: Disagreement Confusion Matrix

After the Gaussian noise is added to the numerical variables, the distribution of these variables changed to a normal distribution with a bell curve (Appendix 5). Even when Gaussian Noise only corrupted numerical variables, the outputs from LR and SVM were relatively stable. Only 4.5% (36/800) of the original training sample is classified differently with LR, while 6% (48/800) for SVM. This indicates that the following classifiers are not very sensitive to the noise.

4.5.2 Accuracy

Accuracy Score	LR	SVM
Baseline	76%	76.5 %
Gaussian Noise	69%	69%

Table 29: Accuracy Score (Original LR & SVM VS Gaussian Noise LR & SVM)

Confusion Matrix	TP	TN	FP	FN
Baseline LR	34	118	20	28
Gaussian Noise LR	0	138	0	62
Baseline SVM	37	116	22	25
Gaussian Noise SVM	0	138	0	62

Table 30: Confusion Matrix (Original LR & SVM VS Gaussian Noise LR & SVM)

Even though the noise was not significantly sensitive to classification, the test set's prediction performance was diminished by 7% compared to Baseline models (Table 29). Both the Gaussian LR model and SVM model surprisingly had an identical accuracy score and confusion matrix. These two models could not correctly predict negative class on the unseen dataset.

These outcomes suggest that the prediction performance is very sensitive to noise, even though classifiers' classification is quite robust. One might argue that different magnitudes of noise could affect the robustness and accuracy score. However, in this specific case, the noisy data did not largely affect the classifiers' stability.

5 Discussion & Conclusion

The paper examined how the change of dataset by resampling and noise would influence a classifier's stability in a specific case of the German Credit dataset. This study aimed to check the robustness of classification by two machine learning algorithms such as Logistic regression and Support Vector Machine. Previously Chandrawat (2020) addressed the importance of a robust model validation framework in credit risk management, emphasizing that the final models should be regularly validated and tested in diverse scenarios to prevent consequences of unwarranted results. Besides ECB and the Basel Committee's also implemented strict minimum standards in the credit lending business to promote a sustainable business environment.

The existing literature indicated that classification from Logistic regression and Support Vector Machine were robust regardless of imbalanced class ratio and resampling techniques. However, LR was sensitive when data was corrupted with noise. The hypothesis was formulated to test the following hypothesis and extend the current validation procedure in the credit scoring domain. The reason was that previous studies related to credit scoring did not exclusively study the aspect of robustness. Instead, many studies rather focused on making an accurate predictive model. As a result, a comprehensive validation procedure was not studied in the context of credit scoring. To extend the validation procedure, this research selected the study by Ala'Raj & Abbod (2015) as the authors also suggested doing extra validation on their method.

Furthermore, this study followed the same procedure as Ala'Raj & Abbod (2015), using the same dataset, data handling process, and similar algorithm configuration. Initially, the baseline model was created by LR and SVM, and then the same procedure was implemented to resampled models and gaussian noise models. The classification outputs, accuracy score, and confusion matrix were used as a benchmark for pairwise comparison. The results showed that Logistic regression and Support Vector Machine were quite robust when the original training sample was perturbated with the bootstrap method and noise was added. However, these two classifiers' classification was not robust when the sample was randomly oversampled or under-sampled. This finding contradicts the point suggested by Marqués et al. (2013) because resampling techniques influenced the classifiers' stability.

Moreover, bootstrap models, regardless of iterations showed a very similar proportion of disagreement against baseline classification. The robust result was also apparent when noise was added. Regarding model performance, the robustness did not necessarily improve predictions' accuracy compared to the baseline models. This was the case for bootstrap and noisy models. However, the accuracy of oversample and under-sample models did not deviate much from the baseline models.

5.1 Theoretical Contribution

The research included a robust check on top of the existing validation procedure to shed light on the importance of a classifier's stability. Given the results, this study proposes a holistic approach to the validation procedure. The construction of the validation procedure follows 10 steps. (1) define a business objective, (2) preprocess data, (3) split train-test set, (4) use an appropriate machine learning algorithm based on the defined business problem, (5) train a baseline model & gather classification, (6) predict on a test set, (7) evaluate model, (8) apply

any resampling techniques and repeat step (5-7), (9) apply noise and repeat step (5-7), and (10) conduct a pairwise comparison.

Step (4) is not mandatory to implement Logistic Regression or Support Vector Machine as a primary machine learning algorithm. The machine learning algorithm's performance varies based on the specific business problem because the characteristics and mechanism of machine learning algorithms are not the same. Moreover, suppose future research aims to measure the model's performance with additional evaluation metrics not used in this study, then other evaluation metrics such as F-measure, recall, and precision can also be used.

Step (8) and (9), the paper recommends using the bootstrap technique with various iterations. Given this study's findings, the mean and standard deviations of numerical variables did not vary after 100 iterations. However, this pattern may not be visible when different kind of data is used for other business problem. As a result, the framework suggests starting with a minimum of 100 iterations and a max of 10,000 iterations. In addition to bootstrap, it also advises implementing oversampling and under-sampling to measure how stable a classifier assigns a class when it is balanced.

5.2 Managerial Implication

The proposed validation procedure would provide a general guideline for a credit analyst or a loan officer to examine which customers are inconsistently assigned by a machine learning algorithm. These specific customers could be analyzed further to zoom in which characteristics are specifically influenced by the algorithm. After that, the final credit scoring model could be adjusted to provide a robust assessment. Moreover, a small bank with a limited size of customer data would be able to apply the suggested resampling techniques to simulate real-life scenarios. As a result, they would have various scenarios to make the model more stable when deployed in the physical environment.

5.3 Limitations & Further research

In this study, the only distinct observation was considered for "Disagreement Confusion Matrix.", which only presented a general aspect of robustness. The following result was sufficient to conclude whether the classification of the algorithms was stable to the change of dataset. However, this naïve robustness check could not analyze which specific items were always misassigned by a classifier when the dataset was perturbated. To go one step further for a robust check, it is recommended to zoom into specific items that are jumping from class to class to answer why these items show this kind of pattern and how these instances influence model predictions against an out-of-sample.

Furthermore, this study only applied three types of resampling techniques that still lack a comprehensive knowledge of the effect of different resampling techniques on classification stability. The potential issue of random oversampling and under-sampling technique is that only some observations are randomly repeated. Especially the instances that had an unstable class could affect the overall robustness of classification. As a result, it suggests applying other advanced resampling techniques such as Synthetic Minority Oversampling Technique (SMOTE) that creates synthetic minority class instances for in-depth analysis (Chawla et al., 2002).

The focus of this study was to add noise in order to examine the robustness. However, only one type of noise was added to the numerical variables. This method showed only one side of the story, so different results might be provided when another kind of noise is added. To address this gap, future research could add a different level of noise to the variables. Nevertheless, some of the categorical variables should be corrupted in some ways to create various situations.

Reference

Ala'Raj, M., & Abbod, M. (2015). A systematic credit scoring model based on heterogeneous classifier ensembles. *INISTA 2015 - 2015 International Symposium on Innovations in Intelligent SysTems and Applications, Proceedings, May*. https://doi.org/10.1109/INISTA.2015.7276736

Baesens, A. B., Gestel, T. Van, Viaene, S., Stepanova, M., Suykens, J., Baesensl, B., Gestel, T. Van, Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Linked references are available on JSTOR for this article : Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, *54*(6), 627–635.

Barandela, R., Sã Anchez B;, J. S., Garcã, V., & Rangel, E. (2003). Rapid and Brief Communication Strategies for learning in class imbalance problems. In *Pattern Recognition* (Vol. 36). www.elsevier.com/locate/patcog

Bennett, K. P., & Campbell, C. (2000). Support Vector Machines: Hype or Hallelujah? *SIGKDD Explor. Newsl.*, *2*(2), 1–13. https://doi.org/10.1145/380995.380999

Bischl, B., Mersmann, O., Trautmann, H., & Weihs, C. (2012). Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evolutionary Computation*, *20*(2), 249–275. https://doi.org/10.1162/EVCO_a_00069

Boneau, C. A. (1960). *The effects of violations of assumptions underlying the t test*. *37*(1). https://doi.org/10.1037/h0041412

Box, G. E. P. (1953). Non-Normality and Tests on Variances. *Biometrika*, *40*(3–4), 318–335. https://doi.org/10.1093/biomet/40.3-4.318

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, *39*(3), 3446–3453. https://doi.org/10.1016/j.eswa.2011.09.033

Campbell, M. K., & Torgerson, D. J. (1999). Bootstrapping: estimating confidence intervals for cost-effectiveness ratios. *QJM : Monthly Journal of the Association of Physicians*, *92*(3), 177–182. https://doi.org/10.1093/qjmed/92.3.177

Chandrawat, G. (2020). *Model Validation*. https://home.kpmg/xx/en/home/insights/2020/01/model-validation.html

Chen, J., Takiguchi, T., & Ariki, Y. (2015). A robust SVM classification framework using PSM for multi-class recognition. *EURASIP Journal on Image and Video Processing*, *2015*(1), 7. https://doi.org/10.1186/s13640-015-0061-x

Chye Koh, H., Chin, W., Statistical, T., & Manager, R. (2006). A Two-step Method to Construct Credit Scoring Models with Data Mining Techniques. In *International Journal of Business and Information* (Vol. 1, Issue 1).

Clancey, W. J. (1984). Classification Problem Solving. *Proceedings of the Fourth AAAI Conference on Artificial Intelligence*, 49–55.

Decamps, J. P., Rochet, J. C., & Roger, B. (2004). The three pillars of Basel II: Optimizing the mix. *Journal of Financial Intermediation*, *13*(2), 132–155. https://doi.org/10.1016/j.jfi.2003.06.003

Dong, G., Lai, K. K., & Yen, J. (2010). Modeling options markets by focusing on active traders. *Procedia Computer Science*, *1*(1), 2463–2468. https://doi.org/10.1016/j.procs.2010.04.278

Dupret, G., & Koda, M. (2001). Bootstrap re-sampling for unbalanced data in supervised learning. *European Journal of Operational Research*, *134*(1), 141–156. https://doi.org/10.1016/S0377-2217(00)00244-7

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, *78*(382), 316–331. https://doi.org/10.1080/01621459.1983.10477973

Feng, J., Xu, H., Mannor, S., & Yan, S. (2014). Robust Logistic Regression and Classification. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, 253–261.

Fensterstock, A. (2005). Credit Scoring And The Next Step. *Business Credit*, *107*(3), 46–49. http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=16336667&site=ehost-live

Frame, W., Srinivasan, A., & Woosley, L. (2001). The Effect of Credit Scoring on Small-Business Lending. *Journal of Money, Credit and Banking*, *33*, 813–825. https://doi.org/10.2307/2673896

Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems. In *O'Reilly Media*. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

Ghorbani, R., & Ghousi, R. (2020). Comparing Different Resampling Methods in Predicting Students ' Performance Using Machine Learning Techniques. *IEEE Access*, *8*, 67899–67911. https://doi.org/10.1109/ACCESS.2020.2986809

Hand, D. J. (2001). Modelling consumer credit risk. *IMA Journal of Management Mathematics*, *12*(2), 139–155. https://doi.org/10.1093/imaman/12.2.139

Heyden, Y. Vander, Nijhuis, A., & Smeyers-verbeke, J. (2001). *Guidance for robustness / ruggedness tests in method validation*. *24*, 723–753.

Hooman, A., Marthandan, G., & Karamizadeh, S. (2013). Statistical and Data Mining Methods in Credit Scoring. *SSRN Electronic Journal*, *50*(5), 371–381. https://doi.org/10.2139/ssrn.2312067

Ince, H., & Aktan, B. (2009). A comparison of data mining techniques for credit scoring in banking: A managerial perspective. *Journal of Business Economics and Management*, *10*(3), 233–240. https://doi.org/10.3846/1611-1699.2009.10.233-240

Janeska, M., Sotiroski, K., & Taleska, S. (2014). Application of the Scoring Model for Assessing the Credit Rating of Principals. *TEM Journal*, *3*(1), 50–54. www.temjournal.com

Kleinbaum, D. G., & Klein, M. (2010). Logistic Regression. In *Journal of the American Statistical Association* (3rd ed., Vol. 91, Issue 433). Springer New York. https://doi.org/10.1007/978-1-4419-1742-3

Lam, H. (2016). Advanced tutorial: Input uncertainty and robust analysis in stochastic simulation. *2016 Winter Simulation Conference (WSC)*, 178–192.

https://doi.org/10.1109/WSC.2016.7822088

Leonard, K. J. (1995). The development of credit scoring quality measures for consumer credit applications. *International Journal of Quality & Reliability Management*, *12*(4), 79–85. https://doi.org/10.1108/02656719510087346

Li, X.-L., & Zhong, Y. (2012). An Overview of Personal Credit Scoring: Techniques and Future Work. *International Journal of Intelligence Science*, *02*(04), 181–189. https://doi.org/10.4236/ijis.2012.224024

Louzada, F., Ara, A., & Fernandes, G. B. (2016). Classification methods applied to credit scoring: Systematic review and overall comparison. In *Surveys in Operations Research and Management Science* (Vol. 21, Issue 2, pp. 117–134). Elsevier Science B.V. https://doi.org/10.1016/j.sorms.2016.10.001

Marqués, A. I., García, V., & Sánchez, J. S. (2013). On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society*, *64*(7), 1060–1070. https://doi.org/10.1057/jors.2012.120

Mester, L. J. (1997). What's the Point of Credit Scoring ? *Business Review*, *3*(January), 3–16. https://www.phil.frb.org/research-and-data/publications/business-review/1997/september-october/brso97lm.pdf

Müller, A. C., & Guido, S. (2017). Introduction to with Python Learning Machine. In *Proceedings of the Speciality Conference on Infrastructure Condition Assessmenr: Art, Science, Practice*.

Paliwal, M., & Kumar, U. A. (2009). Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications*, *36*(1), 2–17. https://doi.org/10.1016/j.eswa.2007.10.005

Provost, F., & Fawcett, T. (2013). *Data Science for Business*. O'Reilly Media, Inc.

Robinson, A. P., & Froese, R. E. (2004). Model validation using equivalence tests. *Ecological Modelling*, *176*(3), 349–358. https://doi.org/https://doi.org/10.1016/j.ecolmodel.2004.01.013

Salciccioli, J. D., Crutain, Y., Komorowski, M., & Marshall, D. C. (2016). Sensitivity Analysis and Model Validation. In *Secondary Analysis of Electronic Health Records* (pp. 263–271). Springer International Publishing. https://doi.org/10.1007/978-3-319-43742-2_17

Schebesch, K. B., & Sleeking, R. (2005). Support vector machines for classifying and describing credit applicants: Detecting typical and critical regions. *Journal of the Operational Research Society*, *56*(9), 1082–1088. https://doi.org/10.1057/palgrave.jors.2602023

Schreiber-Gregory, D. (2018). *Logistic and Linear Regression Assumptions : Violation Recognition and Control*. 1–21.

Schreiner, M. (2004). Benefits and pitfalls of statistical credit scoring for microfinance. *Savings and Development*, *28*(1), 63–86.

Singh, S., & Sedory, S. A. (2011). Sufficient bootstrapping. *Computational Statistics & Data Analysis*, *55*(4), 1629–1637. https://doi.org/https://doi.org/10.1016/j.csda.2010.10.010

Smith, R. S., & Doyle, J. C. (1992). Model Validation : A Connection Between Robust

Control and Identification. *IEEE Transactions on Automatic Control*, *31*(7), 942–952. https://doi.org/10.1109/9.148346.

Thomas, L. C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. In *International Journal of Forecasting* (Vol. 16). www.elsevier.com/locate/ijforecast

Trafalis, T. B., & Gilbert, R. C. (2007). Robust support vector machines for classification and computational issues. *Optimization Methods and Software*, *22*(1), 187–198. https://doi.org/10.1080/10556780600883791

Vapnik, V. (1998). The Support Vector Method of Function Estimation. In J. A. K. Suykens & J. Vandewalle (Eds.), *Nonlinear Modeling: Advanced Black-Box Techniques* (pp. 55–85). Springer US. https://doi.org/10.1007/978-1-4615-5703-6_3

Wagner, H. (2004). The use of credit scoring in the mortgage industry. *Journal of Financial Services Marketing*, *9*(2), 179–183. https://doi.org/10.1057/palgrave.fsm.4770151

West, D. (2000). Neural network credit scoring models. *Computers and Operations Research*, *27*(11–12), 1131–1152. https://doi.org/10.1016/S0305-0548(99)00149-5

Xia, Y., Liu, C., Da, B., & Xie, F. (2018). A novel heterogeneous ensemble credit scoring model based on bstacking approach. *Expert Systems with Applications*, *93*, 182–199. https://doi.org/10.1016/j.eswa.2017.10.022

Xu, Y., & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, *2*(3), 249–262. https://doi.org/10.1007/s41664-018-0068-2

Zeng, G. (2020). On the confusion matrix in credit scoring and its analytical properties. *Communications in Statistics - Theory and Methods*, *49*(9), 2080–2093. https://doi.org/10.1080/03610926.2019.1568485

Appendices

Appendix 1: Variable Description

Attribute Name	Data Type
duration	Integer
creditamount	Integer
Installmentrate	Integer
pre_residencesince	Integer
age	Integer
existingcredits	Integer
peopleliable	Integer
existingchecking	Categorical
savings	Categorical
pre_employmentsince	Categorical
status_sex	Categorical
otherdebtors	Categorical
property	Categorical
otherinstallmentplans	Categorical
housing	Categorical
Job	Categorical
Telephone	Categorical
Foreignworker	Categorical
classification	Categorical

Table 31: Variable List

Appendix 2: Bootstrap Confidence Interval

	Lower Boundary	Upper Boundary
Duration	0.281	0.304
Credit Amount	0.168	0.191
Installment rate	0.720	0.762
Pre_residencesince	0.693	0.728
Age	0.461	0.483
Existing Credits	0.343	0.361

Table 32: Bootstrap 100 Confidence Interval 95%

	Lower Boundary	Upper Boundary
Duration	0.280	0.304
Credit Amount	0.168	0.190
Installment rate	0.723	0.762
Pre_residencesince	0.693	0.730
Age	0.462	0.483
Existing Credits	0.342	0.361

Table 33: Bootstrap 1000 Confidence Interval 95%

	Lower Boundary	Upper Boundary
Duration	0.281	0.304
Credit Amount	0.168	0.190
Installment rate	0.722	0.762
Pre_residencesince	0.693	0.731
Age	0.462	0.483
Existing Credits	0.342	0.362

Table 34: Bootstrap 5000 Confidence Interval 95%

	Lower Boundary	Upper Boundary
Duration	0.281	0.304
Credit Amount	0.169	0.190
Installment rate	0.723	0.762
Pre_residencesince	0.693	0.731
Age	0.462	0.483
Existing Credits	0.342	0.362

Table 35: Bootstrap 10,000 Confidence Interval 95%

Appendix 3: Bootstrap Histogram

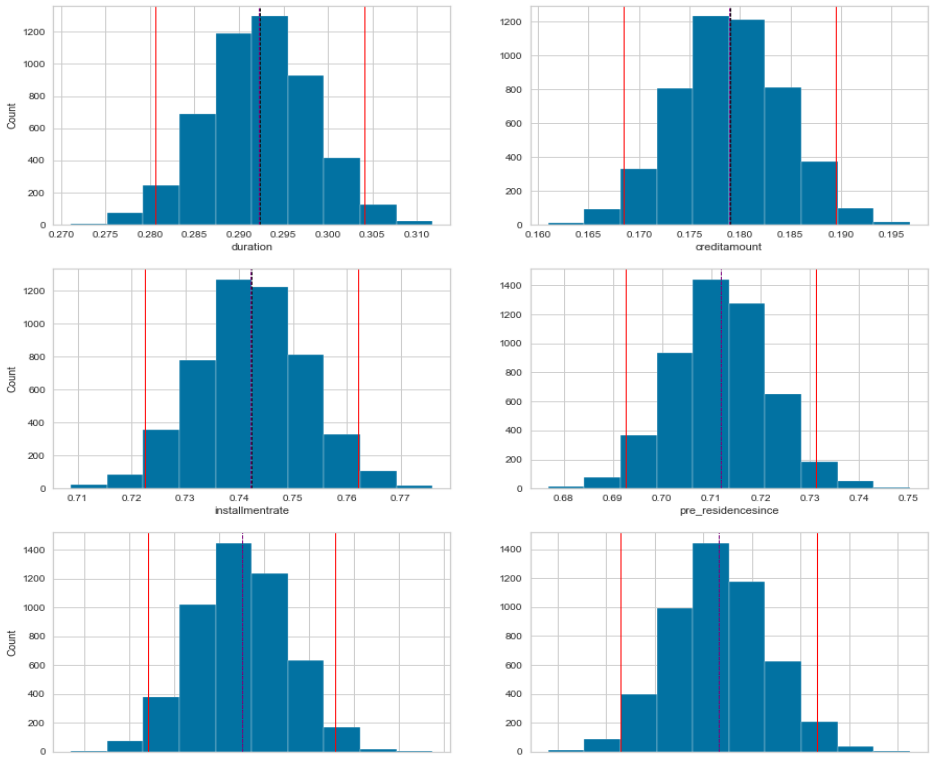

Figure 9: Histogram of Bootstrap 1000 iterations

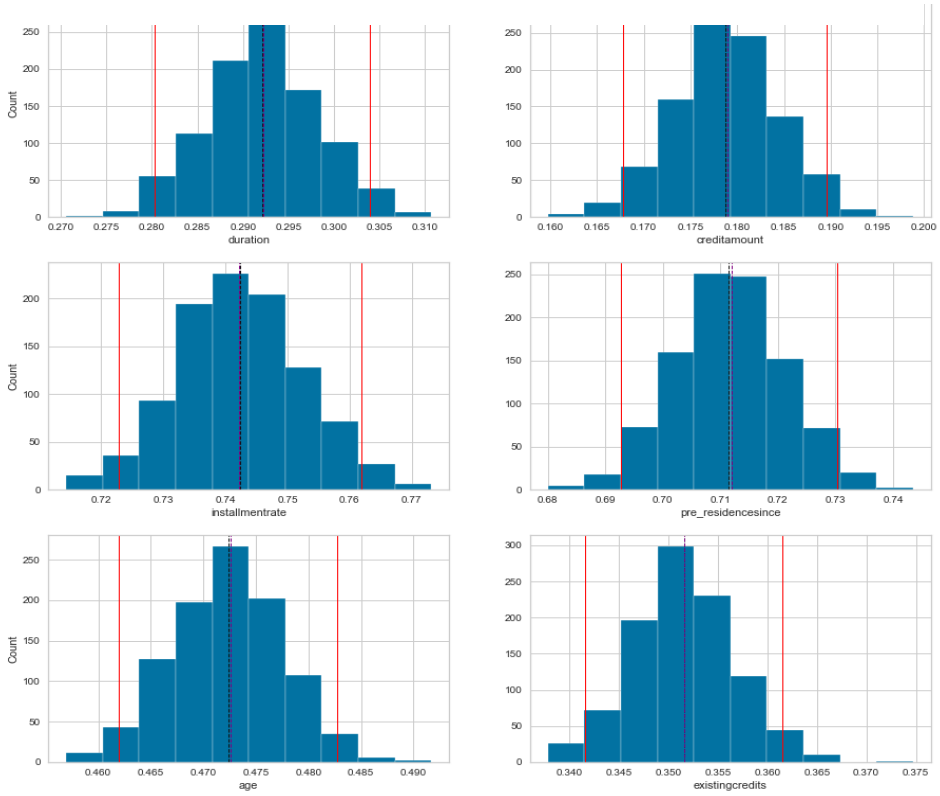

Figure 10: Histogram of Bootstrap 5000 iterations

Appendix 4: Distribution Plot of Undersample

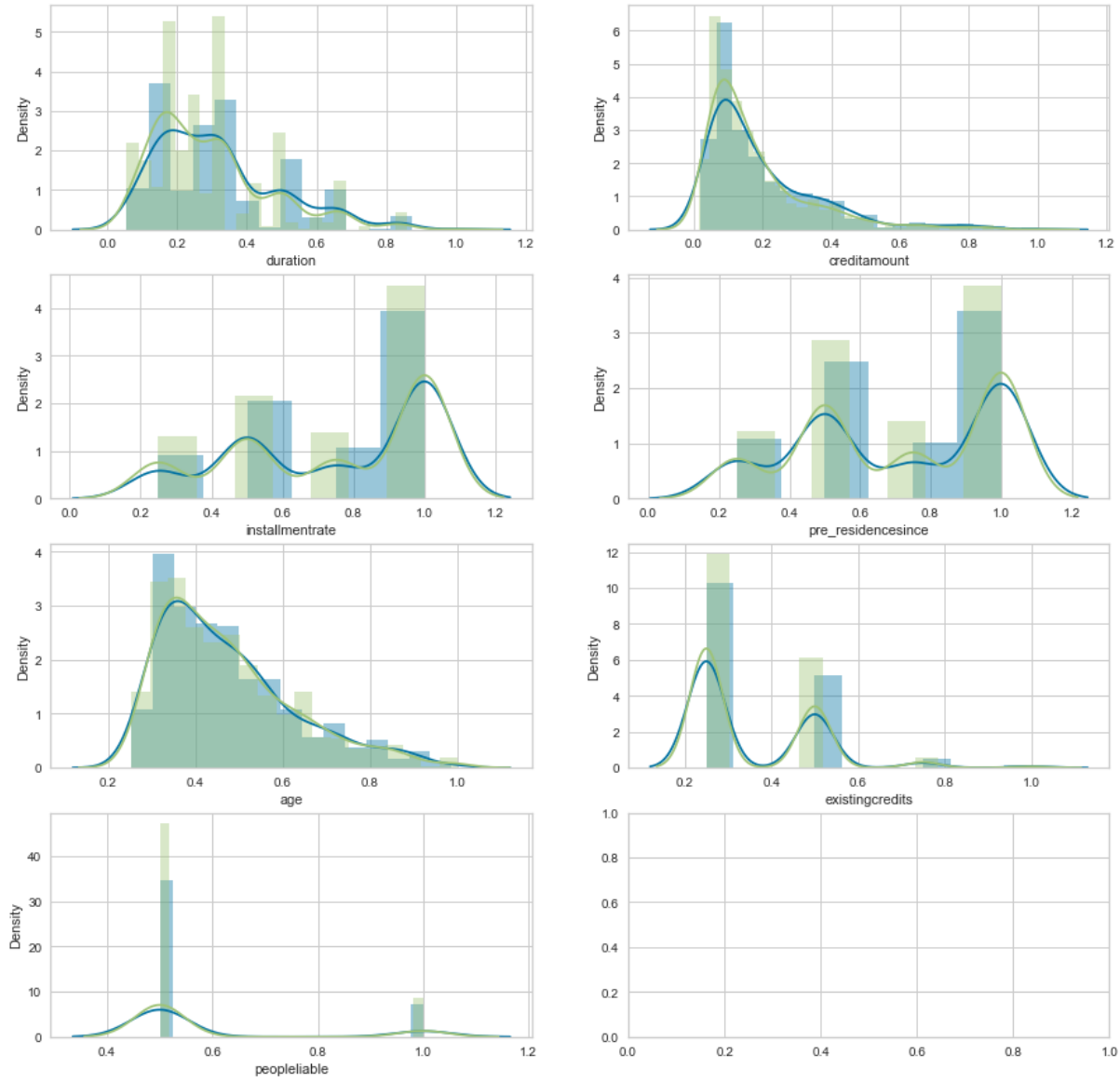

Figure 11: The distribution plots of the undersampled dataset

Appendix 5: Distribution Plot of Gaussian Noise Sample

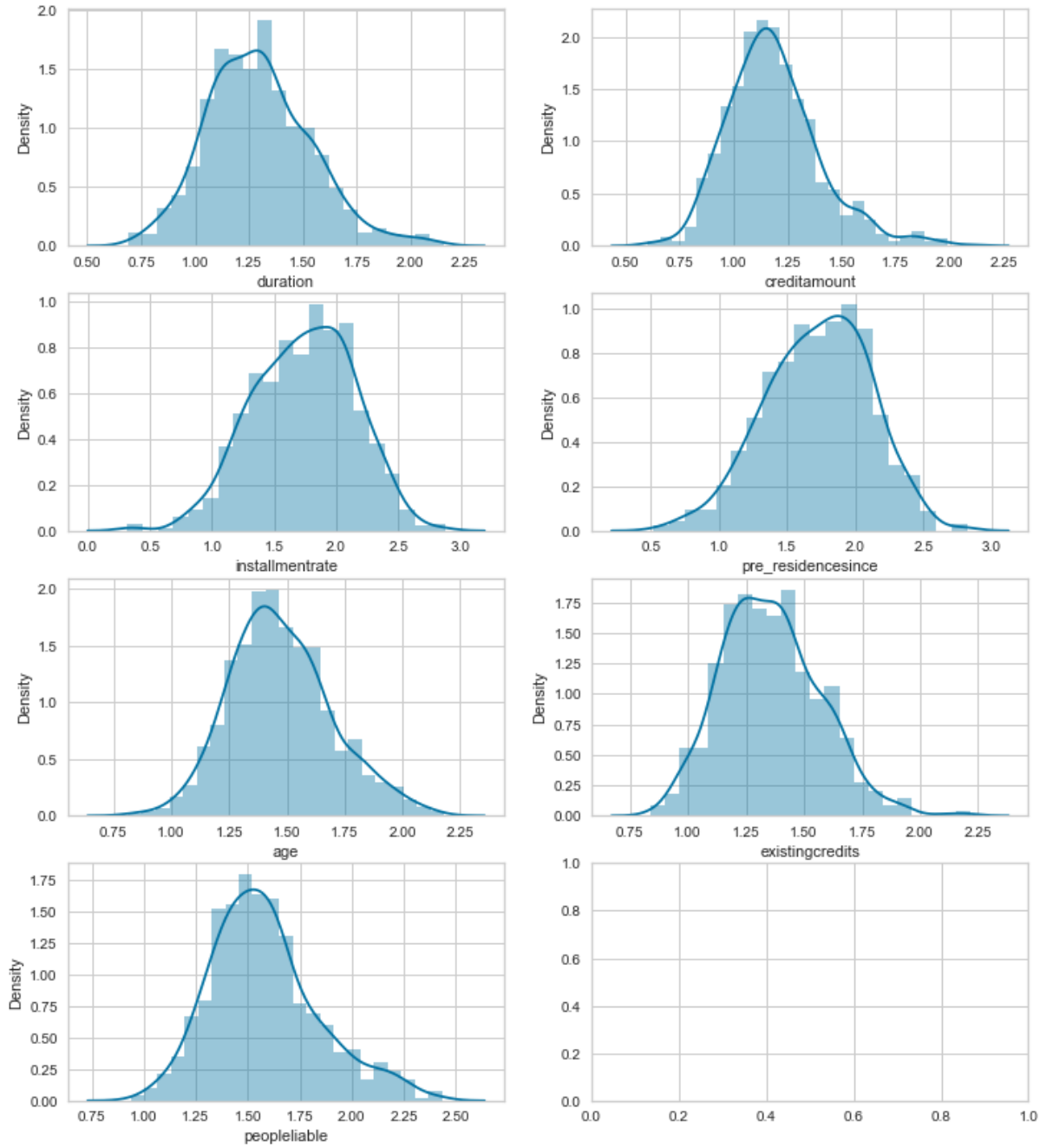

Figure 12: The distribution plots of the Gaussian Noise dataset

Appendix C: Declaration of Originality MSc Thesis *

By signing this statement, I hereby acknowledge the submitted MSc Thesis titled

A Comprehensive validation procedure in credit scoring model

to be produced independently by me, without external help.

Wherever I paraphrase or cite literally, a reference to the original source (journal, book, report, internet, etc.) is provided.

By signing this statement, I explicitly declare that I am aware of the fraud sanctions as stated in the Education and Examination Regulations (EERs) of SBE, Maastricht University.

Place: Maastricht

Date: 22/1/2021

First and last name: Min Woo Lee

Study programme: MSc IB Info mgmt & BI

ID number: i6127619

Signature: