ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Combining Statistical and Machine Learning Techniques in Credit Scoring

Master Thesis - MSc Quantitative Finance

Supervisors
dr. Annika Camehl
Elaine Longden
Gerrit van Eck
Second assessor
Sebastiaan Vermeulen

Author
Jeroen van den Boogaard
435322

Abstract

The credit industry is in fast growth which yields reasoning to develop new models to predict credit scores more accurately. The logistic regression is the industry workhorse because of its ease of interpretation, which is required by regulators and credit risk managers in credit scoring applications. State-of-the-art machine learning models such as random forest and XGBoost are known to outperform the traditional statistical models, but lack interpretability. As such, this research investigates whether the standard framework of the logistic regression can be improved by extracting information from random forest and XGBoost, in terms of single-variable transformations and generated interaction terms. We assess the performance using the Gini Coefficient, the Brier Score and the percentage of correctly classified observations. The study is conducted on a data set containing the characteristics and transactional behavior of 35,660 self-employed persons and small business owners with an active bank account at Knab, a Dutch online bank. It is found that transforming the original set of predictors enhanced with generated interaction terms, results in a model that outperforms the standard logistic regression substantially. Moreover, this revised logistic regression performs competitively to random forest, while preserving its simple interpretation.

Keywords: credit scoring, machine learning, classification, logistic regression, interpretability

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Contents

1 Introduction **1**

2 Data **7**

2.1 Features . . . 7

2.2 Target . . . 8

2.3 Structure . . . 10

3 Methodology **14**

3.1 Logistic Regression . . . 15

3.2 Random forest . . . 15

3.3 XGBoost . . . 17

3.4 SAFE ML . . . 19

3.5 Hybrid Approach . . . 21

3.6 Performance Measures . . . 22

4 Results **24**

4.1 Benchmark model . . . 24

4.2 Competing Models . . . 25

4.2.1 Performance Comparison . . . 25

4.2.2 Interpretability characteristic . . . 28

4.2.3 Sensitivity analysis . . . 30

5 Conclusion **33**

A Appendix **39**

A.1 Additional Feature Information . . . 39

A.1.1 Feature Descriptions . . . 39

A.1.2 Descriptive Statistics: Characteristics . . . 40

A.1.3 Descriptive Statistics: Transactional Behavior . . . 40

A.1.4 Industry SBI Codes . . . 41

A.1.5 Partial Autocorrelation Functions . . . 42

A.2 Classifier Performance . . . 45

A.3 Performance Hybrid Approach . . . 46

A.4 Programming Codes . . . 47

A.4.1 Main Code . . . 47

A.4.2 Performance Measures . . . 47

1 Introduction

There is a fast growth in the credit industry, leading to extensive use of credit scoring models for the credit admission evaluation (Huang et al. 2007). Banking institutions apply credit scoring to classify both retail and corporate applicants as either accepted or rejected based on their creditworthiness. Borrowers with a high credit score are classified as acceptable applicants which get access to credit, while borrowers with a low score are classified as unacceptable applicants and are rejected or get access with less favorable conditions.

In the credit industry, credit risk managers try to increase the credit volume while minimizing their exposure to default. To do so, there is a need for new models and techniques to predict credit scores more accurately. As stated by Huang et al. 2007, more accurate credit scoring has several benefits: (1) reduce the credit analysis cost, (2) decide faster on credit admission, (3) control existing accounts more closely, and (4) prioritize credit collections. Additionally, Henley et al. 1997 has shown that improving the accuracy even by a fraction of a per cent could translate into significant future savings.

There is a wide range of credit scoring techniques available, consisting of both statistical and machine learning techniques. One of the most frequently used, and therefore the industry workhorse, is the logistic regression (Crook et al. 2007). The main reason for the popularity of the traditional logistic regression is its ease of interpretation. The Basel capital accord does not only encourage banks to develop an accurate credit scoring model, but also requires it to be easily interpretable (Van Gestel et al. 2005). There are many studies that prove that the traditional statistical model can be outperformed by machine learning techniques like random forest, neural networks and support vector machines in credit scorecard applications (Dumitrescu et al. 2018, Crook et al. 2007, Kumar and Ravi 2007, Luo et al. 2017 among many others). However, while these ensemble or deep learning models benefit from big data and achieve high accuracy, they end up in a black box, indicating that these models are unable to capture and present the relationship between the predictors and the target variable. This makes these machine learning methods undesirable for regulators and credit risk managers. Ideally, a bank would use a credit scoring model that performs as accurate as these complex black box models, with the ease of interpretation as obtained with the logistic regression. Consequently, the objective of this research is to improve the framework of the logistic regression in credit scoring applications by using information obtained from complex machine learning techniques, while preserving the simple interpretation. In addition, we investigate whether the revised logistic regression can perform competitively to the stand-alone machine learning models.

Blöchlinger and Leippold 2006 investigate the economic benefits from using credit scoring models and conclude that even small differences in discriminatory power lead to economically significant variation in the profitability of a bank's credit business. Moreover, their study shows that an increase in the performance of a bank's credit scoring model negatively affects the market share and profit of other banks. Berger et al. 2005 analyze the effects of credit scoring on the quantity, price and risk of small business credit. They find that this technique causes an increase in the availability of small business credit for higher average prices, in particular for enterprises with relatively higher risk levels. Since small and medium sized entities are major contributors to the strength of local economies (Dumitrescu et al. 2018), these findings yield economic relevance to the use and accuracy of credit scoring models.

In addition to reasons of profitability, accurate credit scoring is of importance for regulators since the Basel capital accord requires banks to hold an adequate financial buffer to ensure that they are sufficiently resilient to withstand losses in times of stress and as a consequence of counterparty credit risk[1]. This required financial buffer depends on the total credit risk that banks face, which can be both overstated and understated if the bank uses an inaccurate credit scoring model (Dumitrescu et al. 2018).

According to Crook et al. 2007 and Dumitrescu et al. 2018 the main reason that machine learning techniques such as support vector machines, neural networks and tree-ensemble methods outperform traditional statistical models is due the fact that the latter fails to fit non-linear effects. In addition to this, there is empirical and theoretical evidence that homogeneous ensemble classifiers (such as random forest and gradient boosting machines) improve the predictive accuracy of credit scoring models by pooling the predictions of various base models (Paleologo et al. 2010, Finlay 2011, Chen and Guestrin 2016). The reasoning behind this is the Strong Law of Large Numbers, which ensures that adding more trees does not cause overfitting of the model, but produces a limiting value of the generalization error (Breiman 2001). The two most well-known algorithms in this classifier family are the bagging algorithm and the boosting algorithm. These algorithms average the predictions of multiple base models in case of a numerical response, and apply the majority rule in case of class labels. The bagging algorithms use bootstrapping to create an ensemble of independent base models (Breiman 1996), whereas the boosting algorithms iteratively create base models that depend on the errors made in the previous model (Freund, Schapire, et al. 1996). We will apply both algorithms in this research as stand-alone machine learning techniques, but also as underlying models to extract information from.

Baesens et al. 2003 benchmarked state-of-the-art classification algorithms on eight credit scoring data sets. They found that least-squares support vector machines and neural network

[1] https://www.bis.org/publ/bcbs189.htm

perform very well, albeit not significantly better than the traditional statistical methods, for the majority of the data sets. The updated version of this paper that was published in 2015 (Lessmann et al. 2015) performs the same research in which several advancements are added to the study, among which novel classification algorithms. In contrast to Baesens et al. 2003, they find that the neural network classifier performs only mediocre, and again not significantly better than logistic regression. Another important finding is that the best-performing homogeneous ensemble method is the random forest, outperforming the support vector machines, neural networks, boosting algorithms and also the logistic regression.

Brown and Mues 2012 perform an experimental comparison on several classifiers for credit scoring data sets with large class imbalance. A common characteristic of credit scoring data sets is high class imbalance since the number of non-defaulting customers is generally much higher than the number of defaults in a credit portfolio. They find that the random forest and gradient boosting classifiers are the best-performing models while handling the data imbalance issue. This yields extra support that bagging and boosting algorithms are adequate models in this credit scoring research.

In order to improve the logistic regression by means of machine learning techniques, we propose the following methodology. First, we apply two tree-ensemble classifiers to the data set, namely random forest (bagging algorithm) and XGBoost (boosting algorithm), and their predictive performance is compared to that of the logistic regression. In order to obtain a fair benchmark model, stepwise selection (both forward and backward) and LASSO are applied to the logistic regression with the original set of predictors, of which the best-performing model is used as a benchmark. In a next step, generated and transformed features such as higher order interactions and discretized variables are extracted from the aforementioned machine learning methods to incorporate in the logistic regression. Particularly, the predictors in the logistic regression are chosen by means of the Surrogate Assisted Feature Extraction for Model Learning (SAFE ML) method and the Hybrid Statistical Inference Approach (Hybrid Approach).

The SAFE ML method, proposed by Gosiewska et al. 2019, transforms the input variables based on the response of the complex black box model, referred to as the surrogate model. The transformation consists of discretizing continuous variables and clustering categories of categorical variables based on the model's output conditional on the respective variable. With this approach the variables are transformed to new features, after which they are extracted from the complex model and incorporated in the logistic regression. With this technique Gosiewska et al. 2019 investigate whether a transparent *glass box* model (i.e. revised logistic regression) can be created by extracting information from the black box model, with a performance close to the black box model. They find that the revised logistic regression outperforms the standard logistic regression and even outperforms the black box model for several (artificial) data sets. We apply the SAFE ML method in order to construct a revised logistic regression with the same goal, but contribute to the existing literature in three different ways. Firstly, Gosiewska et al. 2019 only use XGBoost as the surrogate model, whereas we also use random forest for this purpose. Secondly, we apply the methodology for the first time to a credit scoring data set. Thirdly, the SAFE ML method will be applied to the original set of input variables to construct one model, but also to the set of variables that is augmented with new features that are created by the Hybrid Approach to construct a second model.

The Hybrid Approach, proposed by Levy and O'Malley 2020, captures interactions between variables from tree-ensemble methods and extracts these as newly generated features to include in the original set of predictors. In this way, new relationships between second-order interactions and the target variable can be captured where the logistic regression is unable to. The constructed interactions are based on SHAP interaction values (Lundberg, Erion, et al. 2018), which are on their turn based on the SHAP values. The SHapley Additive exPlanation (SHAP) values, as introduced by Lundberg and Lee 2017, represent the attribution of each single feature on the model's output. Lundberg and Lee 2017 state that this is the only consistent feature attribution method. The SHAP values are extended to the SHAP interaction values to consider the impact of pairwise interactions on a given model prediction. The interaction terms with the highest SHAP interaction values, and thus the most promising interactions, are included as predictors in the logistic regression. Levy and O'Malley 2020 perform a benchmark study on 556 data sets and find that the random forest as a stand-alone machine learning technique significantly outperforms the logistic regression. A more interesting finding is that also the Hybrid Approach significantly outperforms the standard logistic regression. We apply the same method but deviate in two aspects from what is done by Levy and O'Malley 2020: (1) we perform the Hybrid Approach not only in combination with random forest, but also in combination with XGBoost, and (2) the most promising second-order interaction terms will be added to the original set of predictors to construct the Hybrid model, and subsequently this new set will be transformed by means of the SAFE ML method to construct a second model. In this manner, the SAFE ML method and the Hybrid Approach generate separate models, but are also entwined in order to combine the best data representation of single features with new relationships between interactions and the target. This is an extention of Levy and O'Malley 2020 and Gosiewska et al. 2019 who apply both techniques separately.

As mentioned before, the stand-alone machine learning models end up in a black box. This means that it is impossible to find the drivers in the model, let alone their exact relationship with the the probability of default. By applying SAFE ML and the Hybrid Approach, it is possible to extract all the relevant information from the machine learning models and include these transformed and generated features in the logistic regression. For example, if the Hybrid Approach captures {`Number of Transactions` · `Percentage of Transaction Type 1`} as a promising interaction term, then this predictor can be interpreted as the number of type 1 transactions. Consequently, the logistic regression reveals the relative contribution of this driver, and all the other drivers, to the probability of default. In this way we gain new insights and interpretability compared to using solely machine learning techniques.

The different models will be evaluated and compared by means of five different performance measures, covering the three different facets of performance in classification scorecards: the discriminatory ability of the models (as measured by the Gini coefficient), the accuracy of probability predictions (as measured by the Brier Score), and the correctness of categorical predictions (as measured by the overall percentage of correctly classified observations and per class). By doing so, we follow both benchmark studies Baesens et al. 2003 and Lessmann et al. 2015 in which they argue that relying on various metrics that embody the different notions of performance creates a robust and complete assessment.

Including the generated and transformed features in the logistic regression as described above allows the proposed credit scoring model to enjoy the exploratory and predictive abilities of machine learning techniques in the high dimensional domain, together with the ability of the logistic regression to reveal how the set of features vary with each other and the target variable. As opposed to statistical modeling in combination with traditional variable selection techniques, the proposed methodology exploits second-order interactions and non-linear transformations that might have significant explanatory power which otherwise would be missed. This paper shows that there is much to gain from the marriage between traditional statistical modeling and machine learning techniques, where the best attributes of each are combined.

The focus in this paper is on self-employed persons and small business owners, which implies corporate credit scoring. The data that is used to develop the credit scoring model consists of characteristics of the companies and their transactional behavior. The characteristics are related to the industry, geographical and general information, whereas the transactional behavior is reflected by all the transactions that are made by the company. For corporate credit scoring models the data generally consists of balance sheet items, profit-and-loss-account items and macro-economic indicators (Lessmann et al. 2015). Hence, the data in this research deviates from the variables that are generally used for corporate credit scoring. Another aspect in which the data is different from credit scoring data is that it does not contain loans and consequently no actual default data. Therefore insolvency of a customer is used as an indicator for default. Insolvency is determined by comparing the balance on the customer's bank account against a specific threshold. Additionally, we perform a sensitivity analysis by means of a dynamic threshold that affects the number of defaults in the data set, such that the robustness of the model with respect to increasing class imbalance is investigated. The data contains 35,660 customers with an active bank account at Knab, a Dutch online bank, in the period January 2018 to December 2020.

The results show that the benchmark performance is substantially improved upon by the stand-alone machine learning models. This confirms that the chosen models are adequate competing classifiers in this research. Extracting information from these complex machine learning models by means of the SAFE ML method, to incorporate in the logistic regression also outperforms the standard logistic regression in terms of the majority of the performance metrics. Extracting interaction terms from the random forest and XGBoost models by means of the Hybrid Approach leads to an underperformance compared to the benchmark. However, combining both methodologies, such that also the identified interaction terms are transformed by the SAFE ML method, yields a revised logistic regression model that substantially outperforms the benchmark. In particular, there is an improvement of the Gini coefficient of more than 10% and an increase in the PCC for the default class of 9.15%. Moreover, when this technique is applied in combination with random forest, we find the best performance among the interpretable models, performing competitively to the stand-alone machine learning models.

In addition to improving the performance of the standard framework of the logistic regression, we also gain more insights into the drivers in the model. Namely, the interpretability characteristic of the logistic regression allows us calculate the marginal effect of each predictor on the probability of default. Applying this characteristic to the new features constructed by the SAFE ML method and the Hybrid Approach, allows us to gain interpretability in two ways. Firstly, we calculate the coefficient per single-variable for different intervals. For example, the variable `Days without Tax Payments` is transformed into four binary features for the intervals [-∞, 13.48], [13.48, 22.54], [22.54, 28.06], and [28.06, ∞]. These obtain a different estimated coefficient representing the marginal effect, leading to more detailed insights. Secondly, we calculate the coefficients for interpretable interaction terms. For example, the interaction term {`Number of Transactions · Percentage of Transaction Type 3`} is generated and transformed into binary features for different intervals. This leads to new insights arising from the newly constructed interaction term, and more detailed insights as a consequence of the discretization.

From the results can be concluded that the logistic regression is an adequate model for credit scoring applications since it performs well on the credit scoring data set and outperforms two of the proposed models. Nonetheless, enhancing the logistic regression with information extracted from complex machine learning techniques in terms of single-variable transformation and pairwise interactions, outperforms the industry workhorse substantially. This implies that the standard framework of the traditional logistic regression can be improved by means of machine learning techniques, while preserving its simple interpretation.

The rest of the paper is organized as follows. Section 2 provides a description of the data that is used to develop the models. Section 3 describes the methodology of the proposed techniques and summarizes the classification performance criteria. The results are discussed in Section 4. Finally, in Section 5 the conclusions are drawn from this research together with a discussion of the limitations and interesting further research.

2 Data

2.1 Features

The data that is used for this research consists of self-employed persons and small business owners (referred to as customers) that have an active bank account at Knab. The data set contains all the transactions of 35,660 customers over the period January 2018 to December 2020 ($\sim$ 9.6 million transactions). For each transaction various specifications are available such as the type of the transaction, the amount, a tax indicator and more. Using these specifications we have constructed the following variables: `Net Inflow`, `Tax`, `Crypto Trader`, `Money Service`, `Fraud Victim`, `High Risk Country`, `Medium Risk Country`, `Number of Transactions`, `Percentage of Type 1 Transactions`, `Percentage of Type 2 Transactions`, `Percentage of Type 3 Transactions`, `Percentage of Negative Transactions` and the `Days without Tax Payments`. We pool the specifications of the transactions per month for the corresponding customer to be used as a predictor. As an example, let customer X have 200 transactions in a certain month. Then for the feature `Net Inflow` the amount of each transaction in the respective month is summed over the 200 transactions, yielding the `Net Inflow` for customer X as a predictor. Next to this, the data contains characteristics of the customers such as the industry (`SBI Sector`), geographical information (`Province`) and general information such as the number of `Existing Days`, the number of `Employees` and the number of `Registered Days` at Knab. Table 1 shows some descriptive statistics of the continuous and binary variables corresponding to the transactional behavior and the characteristics of the customers. Refer to Tables 11 - 14 in the Appendix for more descriptive statistics and a list with the description of all features.

Table 1: Descriptive statistics of the continuous and binary variables corresponding to the transactional behavior and characteristics of the customers. Refer to Table 11 in the Appendix for the descriptions of the variables

Features	Type	Mean	Stand. Dev.	Min	Max
`Net Inflow`	Transactional behavior	390.6020	20220.1900	-4192296	1793273
`Tax`	Transactional behavior	-1037.1693	4056.0458	-291407	214899
`Number of Transactions`	Transactional behavior	23.2934	32.0348	0	2007
`Days without Tax Payments`	Transactional behavior	26.9099	4.8144	1	30
`Crypto Trader`	Transactional behavior	0.0010	0.0316	0	1
`Money Service`	Transactional behavior	0.0508	0.2196	0	1
`Fraud Victim`	Transactional behavior	0.0112	0.1052	0	1
`High Risk Country`	Transactional behavior	0.0005	0.0225	0	1
`Medium Risk Country`	Transactional behavior	0.0131	0.1139	0	1
`Percentage of Type 1 Transactions`	Transactional behavior	0.5478	0.2984	0	1
`Percentage of Type 2 Transactions`	Transactional behavior	0.2329	0.2257	0	1
`Percentage of Type 3 Transactions`	Transactional behavior	0.2131	0.2883	0	1
`Percentage of Negative Transactions`	Transactional behavior	0.7052	0.2160	0	1
`Employees`	Characteristic	0.9597	1.9610	0	200
`Registered Days`	Characteristic	740.2282	408.6101	24	1976
`Existing Days`	Characteristic	2196.8935	3111.8139	24	43812

2.2 Target

Since the data does not contain loans and consequently no actual default data, we use insolvency of a customer as an indicator for default in case that the customer would actually have a loan. The insolvency is calculated as follows. When a customer has a lower balance on their bank account than a certain threshold for 90 consecutive days in a time-frame of one year ahead, this customer is classified as insolvent. This is calculated for each customer where insolvency ($\sim$ default) is indicated with a one and solvency ($\sim$ no default) with a zero and used as the target variable for the credit scoring model. For the time period to determine the (in)solvency, a forward looking rolling window of one year is used, such that the model predicts the probability that the customer will default in the coming year. The choice for the rolling window of one year is based on the default rate curve which is shown in Figure 1. For the purposes of application scorecards there is no strict requirement for the forecast period. We want to model the number of expected defaults for the customers that are scored and by selecting the cut-off where the curve flattens, the number of months of data required is reduced, without missing material defaults after that point. This gives a good estimate of what can be expected in terms of the total defaults.

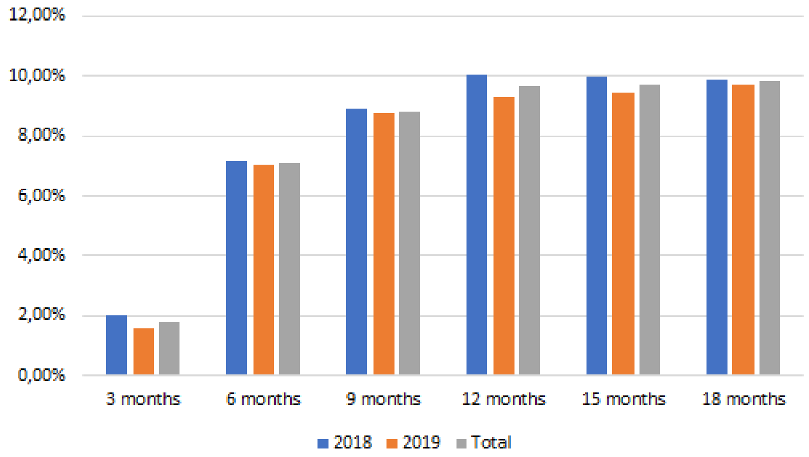


Figure 1: Default rate curve: showing the default rate for the different number of months for the forward looking rolling window.

In addition, we perform a sensitivity analysis of the model by means of a dynamic threshold. For this analysis the model is developed for different values of the threshold, which affects the number of defaults among the customers. This allows to investigate the robustness in model performance for increasing class imbalance and to check whether the same set of features is of importance for the different values of the threshold. These values are determined by three different principal loan amounts of €5000, €7500 and €10000, where the threshold is set equal to the monthly obligation for the respective loan including the interest of 4%[2] based on a 10-year duration. The principal loan amounts are chosen in accordance with the default rate that follows from the corresponding thresholds, which is based on the average default rate among SME loans at Knab over the years 2018-2020. Table 2 shows an overview of the default rate across all customers over 2018 and 2019 for the different values of the threshold. In the remainder of the paper we will refer to the corresponding three different vectors with the target variable `Defaults` and the resulting models as Low Default Rate (6.78%), Medium Default Rate (8.35%) and High Default Rate (9.64%).

Table 2: The default rate for the different values of the threshold.

	Threshold 1: €50.62	Threshold 2: €75.93	Threshold 3: €101.25
Default rate	6.78%	8.35%	9.64%

[2] This percentage is based on the average interest rate on loans over the years 2018-2020.

2.3 Structure

In order to maximally profit from the extensive information in the data set regarding the transactional behavior, the data is structured with monthly observations. This implies that each distinct customer will appear for each month as a separate observation. For example, if customer X has an active bank account in 2018 and 2019, this customer will appear in the data set as a separate observation for each month: X-18-Jan,.., X-18-Dec, X-19-Jan,...,X-19-Dec. This means that the characteristics of the customers are the same for all the months, but the features corresponding to the transactional behavior differ because they are pooled per month. These observations are considered as separate independent customers in the data set. Hence, no time subscript is added, such that this does not imply a panel data set. The choice to regard every month as a separate customer in the model is also desired by Knab for practical reasons. Namely, in this way the model can be recalibrated every month, which is very convenient for a big portfolio in which during every month customers enter and leave. In order to validate this approach and motivate it from an academic perspective, we investigate whether there is an autoregressive pattern present in the data. We establish an intuitive understanding of the data by plotting the time series for all the independent variables in three different ways. Firstly, we investigate the time series for all customers by means of a scatter plot. Secondly, we plot the mean over all customers over time. Thirdly, we randomly select 100 customers and examine the time series plot. By performing this analysis, we get an idea whether an autoregressive pattern exists for the whole cross-section or on individual level. Figure 2 presents the time series scatter plot for the variable `Net Inflow` for all 35,660 customers over the time period January 2018 to December 2019. The graph shows that the `Net Inflow` does not contain any particular pattern, but seems to fluctuate around its mean. The same inference can be drawn from Figure 3 and 4, showing the mean of `Net Inflow` over time and the time series for a random sample of 100 customers, respectively. This yields the intuitive belief that the observations for this variable are independent of each other, both on individual level and cross sectionally. The same analysis is performed for the other variables reflecting the transactional behavior, from which the same conclusion arises.

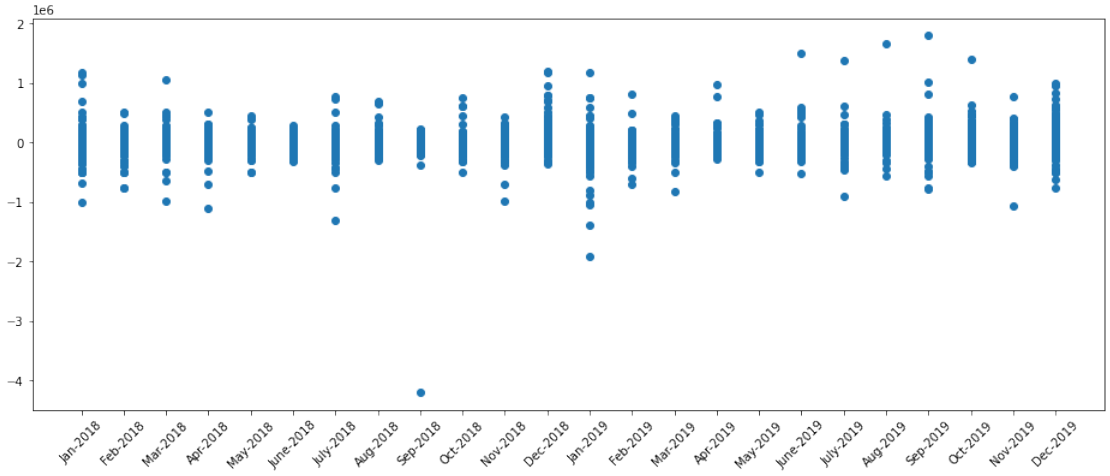


Figure 2: Time series scatter plot of `Net Inflow` over the time period January 2018 to December 2019.

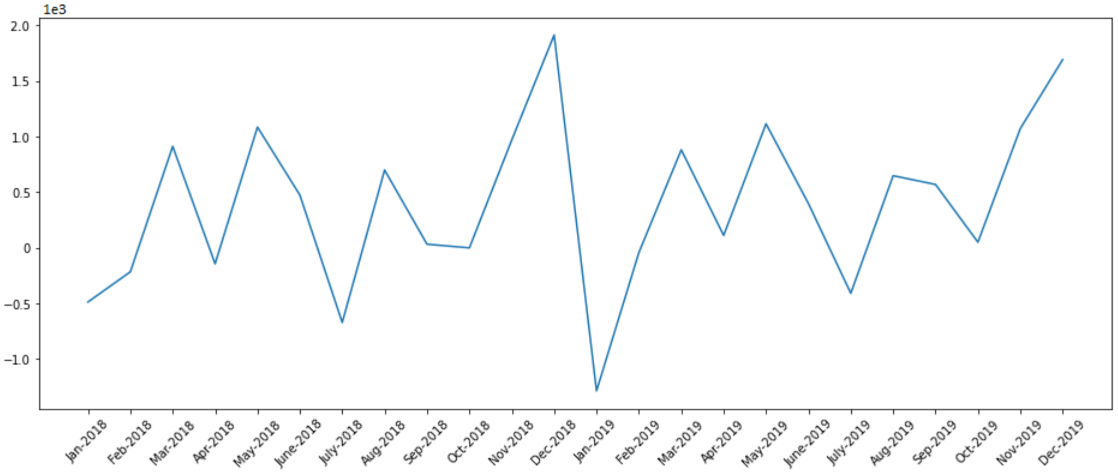


Figure 3: Mean over all customers of `Net Inflow` over the time period January 2018 to December 2019.

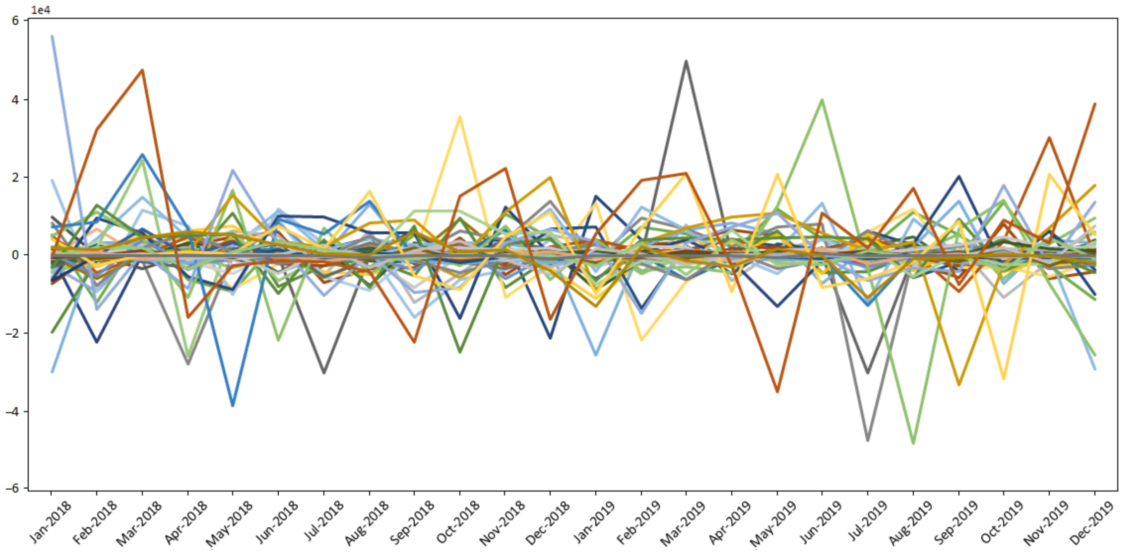
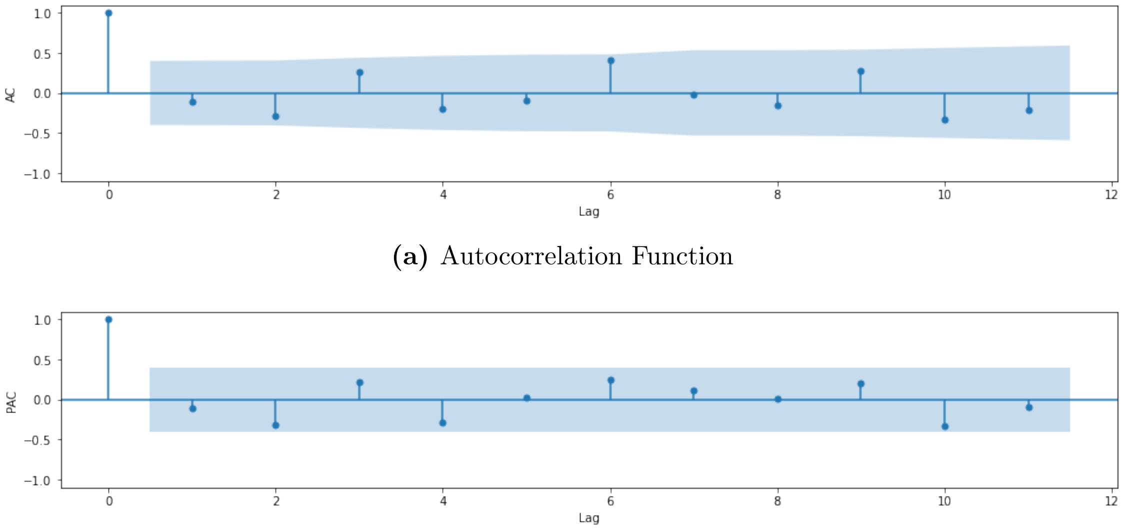


Figure 4: `Net Inflow` over the time period January 2018 to December 2019 for a random sample of 100 customers.

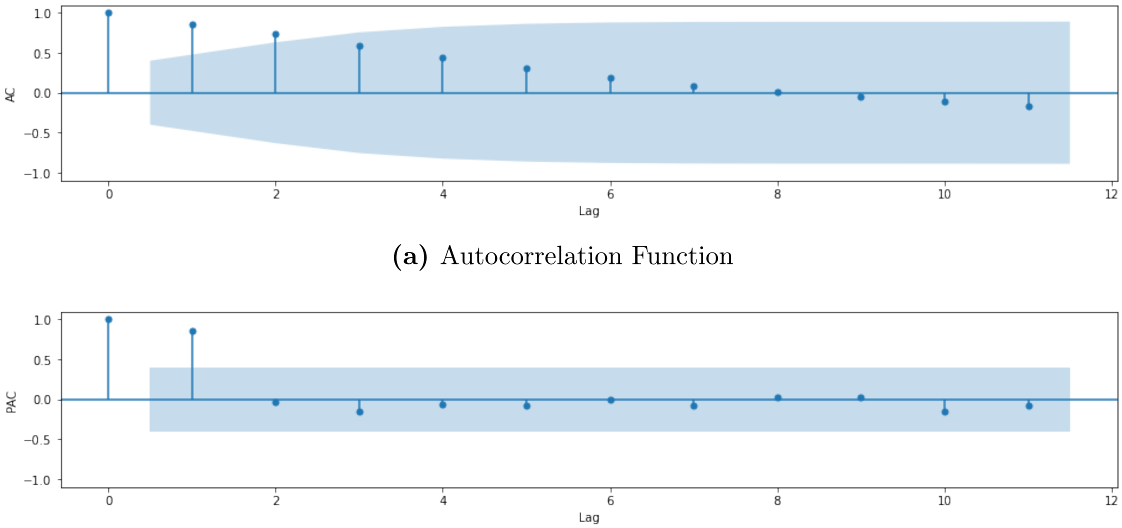
To statistically test for autocorrelation we plot the autocorrelation function (ACF) and partial autocorrelation function (PACF) together with the corresponding confidence levels for a significance level of 5%. We have chosen for this analysis instead of the rather standard tests on autocorrelation such as the Durbin-Watson or Ljung-Box test statistics. The reason for this is that these standard tests are applied to the residuals of a fitted linear regression model. However, since we are applying a logistic regression, the preference is for the ACF and PACF that do not require a pre-specified model. We plot the functions for the target variable and all the variables reflecting the transactional behavior. In order to deal with the cross-sectional dimension, the average over all customers is taken. Figure 5 and 6 show the ACF and PACF for `Net Inflow` and the target variable, respectively, for which the lags up to one year are included.



(a) Autocorrelation Function

(b) Partial Autcorrelation Function

Figure 5: ACF **(a)** and PACF **(b)** of `Net Inflow` with corresponding confidence interval.



(a) Autocorrelation Function

(b) Partial Autcorrelation Function

Figure 6: ACF **(a)** and PACF **(b)** of the `Defaults` with corresponding confidence interval. This is done for the Medium Default Rate.

Both the ACF and PACF for `Net Inflow` in Figure 5 show that there is evidence against autocorrelation for lags up to one year. Figure 6a shows that there is statistically significant autocorrelation in the defaults for lags up to two months. However, the PACF as presented in Figure 6b controls for autocorrelation that is present between observations for a shorter lag length. In this plot can be seen that there is only significant autocorrelation in the defaults for lag of one month, and very small and insignificant autocorrelation for the longer lag length. Table 3 presents the values of the PACF for the target variable and all the variables reflecting the transactional behavior. The corresponding graphs can be found Figure 10 in the Appendix. The bold faced values in Table 3 indicate autocorrelation that is significantly different from zero. It can be seen that except for the lag of one or two months for several variables, there is no significant autocorrelation in the data set. Hence, because there is almost no statistically significant autocorrelation present among the dependent and independent variables, there is no autoregressive pattern to be captured by using panel data. These results support the choice to use the monthly observations of each customer as separate customers in the model. We split the data in training and test set in such a way that each customer only occurs in either the training set or the test set in order to avoid a biased result in performance.

Table 3: Partial autocorrelation values for the target variable (`Defaults` for the Medium Default Rate) and the variables reflecting the transactional behavior, for lags up to one year. Bold faced values indicate significant autocorrelation for the respective lag.

	Lag										
	1	2	3	4	5	6	7	8	9	10	11
Defaults	**0.861**	-0.033	-0.152	-0.068	-0.079	-0.011	-0.075	0.017	0.031	-0.146	-0.083
Net Inflow	-0.114	-0.310	0.212	-0.282	0.028	0.252	0.110	0.005	0.203	-0.338	-0.099
Tax	-0.396	**-0.719**	0.360	-0.129	0.006	-0.186	0.185	-0.197	0.050	-0.277	-0.084
Number of Transactions	**0.463**	0.131	0.310	-0.140	0.062	0.251	-0.042	-0.358	-0.023	-0.155	0.093
Days without Tax Payments	**-0.455**	-0.352	0.259	0.058	-0.057	-0.141	-0.037	-0.336	-0.091	-0.116	-0.377
Crypto Trader	0.367	0.166	0.022	0.040	-0.040	-0.197	-0.142	-0.105	-0.204	-0.013	-0.172
Money Service	**0.826**	0.027	-0.115	0.035	0.072	-0.018	-0.071	-0.302	-0.002	-0.084	-0.106
Fraud Victim	-0.326	**-0.528**	0.111	-0.201	-0.071	-0.377	-0.015	-0.031	0.020	-0.011	-0.033
High Risk Country	0.335	-0.003	0.175	0.056	0.033	0.038	-0.060	-0.102	0.086	-0.208	0.055
Medium Risk Country	**0.530**	0.433	0.182	-0.127	0.034	-0.345	0.001	-0.098	0.042	-0.236	-0.095
Percentage of Type 1 Transactions	0.295	0.206	0.251	-0.379	0.052	-0.101	-0.038	-0.209	-0.077	-0.154	-0.103
Percentage of Type 2 Transactions	-0.242	-0.335	0.103	0.123	-0.054	0.099	-0.062	-0.083	0.110	-0.135	-0.042
Percentage of Type 3 Transactions	0.185	0.117	0.336	-0.146	-0.315	-0.090	-0.139	-0.099	-0.057	-0.224	-0.100
Percentage of Negative Transactions	-0.399	-0.165	0.315	0.296	0.046	0.005	-0.167	-0.164	0.023	0.027	0.020

3 Methodology

This section provides a description of the models and techniques that will be used to develop the credit scoring model. Firstly, the base model of this research, the logistic regression, is briefly explained. Next, two machine learning techniques, random forest and XGBoost, are described which are known to perform well in credit scoring applications and other classification problems (Dumitrescu et al. 2018, Lessmann et al. 2015, Gosiewska et al. 2019). Lastly, we introduce the two main methods of this research, which will entwine the framework of logistic regression with the aforementioned machine learning techniques.

3.1 Logistic Regression

Let (x_i, y_i) for $i = 1, ..., n$ be a sample of size n where $x_i \in \mathbb{R}^M$ is an M-dimensional vector of variables and $y_i \in \{0, 1\}$ is the binary target variable indicating the default of an obligor with a one and non-default with a zero. The aim of the logistic regression in this context is to calculate the probability that an obligor i defaults in the following year conditional on his characteristics $x_{i,j}$ for $j = 1, ..., M$, i.e. $\Pr(y_i = 1|x_i)$. This follows from

$$\log\left(\frac{\Pr(y_i = 1|x_i)}{1 - \Pr(y_i = 1|x_i)}\right) = \beta_0 + \sum_{j=1}^{M} \beta_j x_{i,j} = x_i'\beta, \tag{1}$$

which can be rewritten to

$$\Pr(y_i = 1|x_i) = \frac{1}{1 + \exp(-x_i'\beta)}, \tag{2}$$

where β is the vector of size $M + 1$ with coefficients of the predictors and a constant term. The main reason for the popularity of the logistic regression is its ease to interpret the results and its main drivers. Namely, the marginal effect of each feature to the target variable can easily be calculated as follows

$$\frac{\partial \Pr(y_i = 1|x_i)}{\partial x_{i,j}} = \hat{\beta}_j \frac{\exp(x_i'\hat{\beta})}{1 + \exp(x_i'\hat{\beta})}, \tag{3}$$

where $\hat{\beta}$ are the estimates of β.

In order to create a fair benchmark model, we apply stepwise selection, both forward and backward, and LASSO regularization to the standard logistic regression. The best-performing model resulting out of these three variable selection methods will be used as the benchmark model. Stepwise selection is performed based on a significance level of 5% for the Akaike information criterion. The penalty term for LASSO controls the regularization strength. When this is set equal to zero, we obtain the standard logistic regression. When we increase the penalty term, the amount of regularization is amplified. Hence, we determine the penalty term based on a 5-fold cross validated search over a wide range of values.

3.2 Random forest

The first machine learning model that will be applied is random forest. The choice for this machine learning technique is based on its performance in credit scoring applications in existing literature, where it outperforms several classifier families among which support vector machines and neural networks (Lessmann et al. 2015). The random forest classifier is a non-parametric tree-based classifier. It consists of an ensemble (also referred to as *forest*) of individual decision trees consisting of non-linear *if-then-else* rules. Each decision tree divides the feature space into a number of distinct and non-overlapping regions by splitting features for different thresholds. Following the notation of Dumitrescu et al. 2018, this can be formulated as follows. Let $D_{m,l}$ be the data at a specific leaf m for tree l, and $\theta_{m,l} = (j_{m,l}, t_{m,l,j})$ be the candidate for splitting where $j_{m,l} = 1, ..., M$ refers to a predictor $x_{i,j}$, and $t_{m,l,j}$ refers to the threshold to which the value of the predictor is compared. Each split divides the data for a given leaf into two new distinct subsets, $D_{m,l,1}(\theta_{m,l})$ and $D_{m,l,2}(\theta_{m,l})$, where

$$D_{m,l,1}(\theta_{m,l}) = (x_i, y_i)|x_{i,j} \leq t_{m,l,j}, \tag{4}$$

$$D_{m,l,2}(\theta_{m,l}) = (x_i, y_i)|x_{i,j} > t_{m,l,j}. \tag{5}$$

The optimal combination of the variable to split $j_{m,l}$ and the corresponding threshold $t_{m,l,j}$ can be estimated as

$$\hat{\theta}_{m,l} = (\hat{j}_{m,l}, \hat{t}_{m,l,j}) = \underset{\theta_{m,l}}{\arg\max} \Big\{ \mathrm{G}(D_{m,l}) - \mathrm{G}\Big(D_{m,l,1}(\theta_{m,l}), D_{m,l,2}(\theta_{m,l}) \Big) \Big\}, \tag{6}$$

where $G(\cdot)$ is an error measure which is defined as

$$\mathrm{G} = \sum_{k=1}^{K} \hat{p}_{m,k}(1 - \hat{p}_{m,k}), \tag{7}$$

where $\hat{p}_{m,k}$ is the proportion of class-k observations in the m-th subset, also referred to as the purity of each subset. In this research the classes are 1 ($\sim$ default) and 0 ($\sim$ non-default), such that $K = 2$. There are several error measures available, from which we adopt the most frequently used which is the Gini Index (Pal 2005) as denoted by Equation 7, such that $\mathrm{G}(D_{m,l})$ is the error measure for sample $D_{m,l}$ and $\mathrm{G}(D_{m,l,1}(\theta_{m,l}), D_{m,l,2}(\theta_{m,l}))$ is the average error over the two subsets $D_{m,l,1}$ and $D_{m,l,2}$. Consequently, the optimal estimate $\hat{\theta}_{m,l}$ as calculated by Equation 6 equals the candidate split that reduces the error measure the most.

Random forest applies smart-bagging, in which each decision tree is constructed independently by using a random subset of the features as candidates for splitting. As opposed to bootstrapping, this technique is applied in order to obtain an ensemble of trees that are not highly correlated. This bagging approach pools various decision trees in order to reduce instability. The final classification is done by means of majority voting, where each decision tree has a unit vote for the class that occurs most according to the values of the features.

The random forest has many hyperparameters which can be tuned in order to fit the model to the data set. However, to prevent the time-consuming parameter tuning we focus on some of the hyperparameters that have the biggest impact on model performance according to Breiman 2001, namely: the number of decision trees in the forest, the maximum number of features for splitting, the maximum depth of the individual trees, the minimum number of observations to split on at a node and the minimum number of observations for a leaf node. We perform a two-step approach in order to find the optimal hyperparameters, whilst minimizing the training time. The first step in this approach is to perform a random grid search on a wide range of parameters settings, in which not all parameters are tried out but only a number of random combinations is taken from the specified range. The second step consists of a non-random grid search that is performed on a narrow range around the optimal set of hyperparameters that was found in step one, in which all the parameter settings are fitted. Both grid searches are performed based on 5-fold cross validation.

3.3 XGBoost

The second machine learning technique that we apply to the data set is XGBoost, a scalable implementation of the gradient boosting framework introduced by Friedman 2001. This is a gradient tree boosting machine which generates the decision trees sequentially in which every tree learns from the previous errors in an iterative fashion. XGBoost gained in popularity due to its performance in the 2015 Kaggle competition[3], in which 17 out of the 29 winning solutions of classification problems XGBoost was applied, outperforming among others deep neural networks.

Since XGBoost uses the decision tree as base model and the target variable is binary, the model prediction is made by majority voting in a collection $\mathcal{F}$ of $\mathcal{A}$ additive tree functions:

$$\hat{y}_i = \phi(x_i) = \sum_{\alpha=1}^{\mathcal{A}} f_\alpha(x_i), \;\; f_\alpha \in \mathcal{F}, \tag{8}$$

where $\mathcal{F} = \left\{ f(x) = w_{q(x)} \right\} \left(q : \mathbb{R}^M \to T, w \in \mathbb{R}^T \right)$. Here q indicates a mapping function that assigns each observation to the corresponding leaf node, where T is the total number of leaves in each tree and w are the weights representing the scores of the leaves. Each tree function denoted by f_α has its own set of aforementioned parameters. A prediction $\hat{y}_i$ is made based on the sum of the scores of the leaves which the respective observation is classified into.

[3] https://www.linkedin.com/pulse/present-future-kdd-cup-competition-outsiders-ron-bekkerman/

The regularized objective function as defined by Chen and Guestrin 2016 is as follows

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{\alpha=1}^{\mathcal{A}} \Omega(f_\alpha) \quad \text{with} \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2, \tag{9}$$

where ℓ is the convex training loss function and Ω the regularization function to penalize the model complexity and avoid overfitting of the model. Since Equation 9 consists of estimation functions as parameters ($\hat{y}_i$), traditional optimization techniques are incapable of optimizing this objective function. This leads to a numerical approach in which the optimization is done in an iterative and additive fashion. Particularly, in each iteration t the tree function f_t that is found to have the biggest improvement of the model according to the objective function in Equation 9 is greedily added to the previous prediction:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \ell\big(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\big) + \Omega(f_t), \tag{10}$$

where $\hat{y}_i^{(t)}$ is the estimation of observation i for the t-th iteration. If one would be interested in the mathematical derivation of Equation 10 to arrive at the analytical solution of both the optimal weights and the corresponding optimal value please refer to Chen and Guestrin 2016.

Also XGBoost requires tuning of hyperparameters to improve and fully leverage its advantages over other machine learning models. The parameters can be divided into three types[4]: general parameters, booster parameters and task parameters. The (learning) task parameters decide on the learning scenario and thus consist of the hyperparameters to be tuned while training the model. Compared to tuning random forest, a slightly different approach is used to tune the XGBoost model. Namely, we perform a three-step approach in which first the optimal number of trees ($\mathcal{A}$) is determined, then tree-specific hyperparameters are tuned (such as the maximum depth of each individual tree and the minimum sum of weights needed in a node for splitting), and finally the regularization parameters are tuned to reduce model complexity. Once again, a random grid search on a wide range of parameter settings is followed by a non-random grid search on a narrow range around the previously found set of parameters.

[4] https://xgboost.readthedocs.io/en/latest/parameter.html

3.4 SAFE ML

The SAFE ML method, introduced by Gosiewska et al. 2019, transfers knowledge about relationships from a complex surrogate model to a simple model. This is done by transforming single features based on the expected output of the surrogate model, and subsequently extracting these transformed features to incorporate in a simple model, such as the logistic regression.

The first step is to train a complex surrogate model. The SAFE ML method allows for any class of models, which in this research will be random forest and XGBoost. After the surrogate model is trained, we calculate the partial dependence profile. The transformations of the features are based on the partial dependence function (Friedman 2001) which calculates the expected output of the model conditional on a specific feature. This is denoted as follows

$$f_j(x_j) = \mathbb{E}_{x-j}[F(x_j, x_{-j})], \tag{11}$$

which is approximated by

$$\hat{f}_j(x_j) = \frac{1}{n}\sum_{i=1}^{n} F(x_{i,j}, x_{i,-j}) \tag{12}$$

where x_{-j} denotes the subset of all features except x_j and F denotes the surrogate model. Calculating how the expected output of a model changes for varying values of the respective variable helps to understand how the model depends on each of them.

Next, to determine the transformation of continuous and categorical variables based on the model response the change point detection method and hierarchical clustering are used, respectively. The change point detection method (Truong et al. 2018) identifies homogeneous regimes in the model's output for the range of values of the respective variable. Subsequently, the variable is discretized based on the changing points and the regularization penalty γ, indicating when the amplitude of the changing point is of sufficient size. The higher the value of the penalty the smaller the number of levels of transformed variables will be created. We perform a grid search to find the optimal value for the penalty term. This practice transforms the original set of variables to new binary features. Figure 7 presents an illustration of the change point detection method in a partial dependence plot for the feature `Days without Tax Payments` where the changing points are indicated by the red vertical lines.

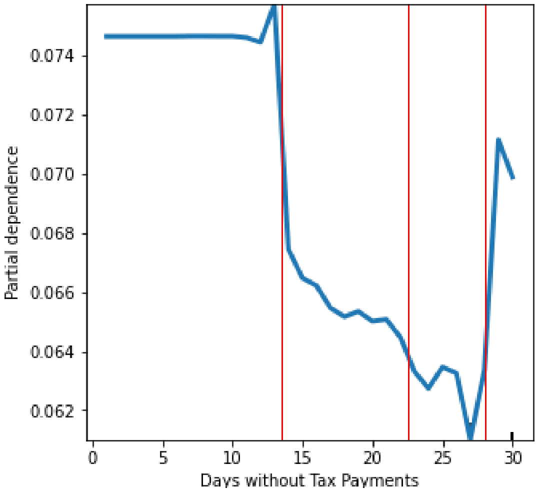


Figure 7: Partial dependence plot for the feature `Days without Tax Payments`. Red vertical lines indicate the changing points.

Based on the change point detection method, the variable `Days without Tax Payments` is transformed into four binary features corresponding to the four different intervals as indicated by the red vertical lines in Figure 7. Hence, the following features are constructed: (1) `Days without Tax Payments` $\left[\texttt{-}\infty\texttt{, 13.48}\right]$, (2) `Days without Tax Payments` $\left[\texttt{13.48, 22.54}\right]$, (3) `Days without Tax Payments` $\left[\texttt{22.54, 28.06}\right]$, and (4) `Days without Tax Payments` $\left[\texttt{28.06, }\infty\right]$. We exclude the first binary feature from the set of input variables to prevent multicollinearity.

In case of categorical variables we use hierarchical clustering (Rokach and Maimon 2005). In particular, divisive hierarchical clustering is performed in which the categories initially belong to a single cluster and are iteratively splitted into individual subclusters with the largest between-group dissimilarity, based on the model response. This results in a dendrogram which is cut at the desired level of (dis)similarity based on the same penalty term γ. A visualization of this approach is shown in Figure 8.

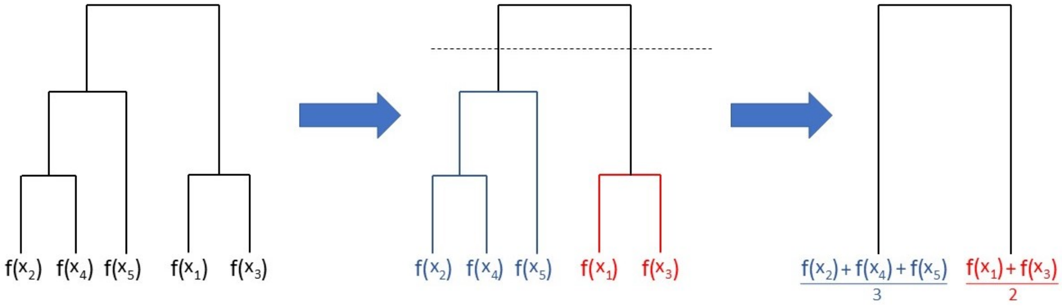


Figure 8: Divise hierarchical clustering: categories that initially belong to a single cluster are iteratively splitted into individual subclusters based on the model response and cut at the desired level of similarity.

The new binary variables that are created by means of the change point detection method and the new clusters of categorical variables that are created by hierarchical clustering are the predictors that will be included in the simple logistic regression model. The transformation of the variables changes the representation of the data such that the logistic regression model is trained with features that better capture the true relationships.

3.5 Hybrid Approach

Where the SAFE ML method transforms single features, the Hybrid Approach as proposed by Levy and O'Malley 2020 extents this by capturing pairwise interactions between variables to extract the most promising as new features. This technique allows statistical models to include non-linear predictors found by tree-ensemble methods in the high-dimensional feature space. The (non-linear) interaction terms to include are based on the SHAP interaction values. The SHAP interaction values represent the attribution of each pairwise interaction term on a given model prediction. They are an extension of the SHAP (SHapley Additive exPlanation) values which represent the attribution of each single feature on the model's output. Lundberg, Erion, et al. 2018 show that the SHAP values are the only consistent feature attribution method, indicating that features with a larger attribution value are always more important than features with a lower attribution. The function of the additive feature attribution method is defined as follows

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j, \tag{13}$$

where z'_j is a binary variable indicating the presence of feature j, M is the number of input features and $\phi_j \in \mathbb{R}$ is the SHAP attribution value for feature j, which is defined as

$$\phi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(|M| - |S| - 1)!}{M!} [f_x(S \cup \{j\}) - f_x(S)], \tag{14}$$

where $f_x(S) = \mathbb{E}(f(x)|x_S)$ with f as the original model, N as the set of all input features and S being the subset of the features represented by the non-zero indices in z'. The property of local accuracy ensures that the sum of the attribution values ϕ_j in Equation 13 equals the output of the original model. Lundberg, Erion, et al. 2018 use the more modern Shapley interaction index, as introduced by Fujimoto et al. 2006, as an extension to the SHAP interaction values as follows

$$\Phi_{h,j} = \sum_{S \subseteq N \setminus \{h,j\}} \frac{|S|!(M - |S| - 2)!}{2(M-1)!} \nabla_{h,j}(S), \tag{15}$$

for feature h and j when $h \neq j$, and

$$\nabla_{h,j}(S) \;=\; f_x(S \cup \{h,j\}) - f_x(S \cup \{h\}) - f_x(S \cup \{j\}) + f_x(S). \tag{16}$$

The SHAP interaction value is constructed such that it is equally divided among the two respective features h and j, yielding $\Phi_{h,j} = \Phi_{j,h}$ such that we obtain the total attribution of the interaction term by summing the respective interaction indices $\Phi_{h,j} + \Phi_{j,h}$. The presence of both the SHAP value and the SHAP interaction values allow to obtain the main impact of a feature on a given model prediction as follows

$$\Phi_{h,h} = \phi_h - \sum_{j \neq h} \Phi_{h,j}, \tag{17}$$

where $\Phi_{h,h}$ indicates the main effect of feature h, which is for notational convention denoted as the interaction between feature h and h. This isolates the interaction effect $\Phi_{h,j}$ from the main effect $\Phi_{h,h}$ and exposes whether the random forest and XGBoost have captured relevant relationships where the statistical model was unable to. Subsequently, the generated features with the highest SHAP interaction values are extracted and will be included in the logistic regression.

3.6 Performance Measures

To assess and compare the performance of the different classifiers, we apply five performance measures covering the three different facets of performance in classification scorecards. These three facets are the discriminatory power of the models, the accuracy of probability predictions and the correctness of categorical predictions (Lessmann et al. 2015). By applying multiple metrics and covering the different notions of performance, a robust and thorough comparison is made between the models.

We measure the discriminatory power of the models with the Gini coefficient, which is one of the most commonly used for this purpose (Bijak and Thomas 2012) and is unaffected by class imbalance (Fawcett 2006). The Receiver Operating Characteristic (ROC) curve has to constructed, which consists of the cumulative distribution functions of prediction scores for both the default and non-default observations. The sensitivity, which is the fraction of default observations that are predicted as default, is presented on the y-axis. The specificity, which is the fraction of non-default observations that are predicted as non-default, is presented on the x-axis. Subsequently, we calculate the area under the ROC curve (AUC), which refers to the probability that a classifier gives a higher score to a randomly selected default observation than a randomly selected non-default observation (Baesens et al. 2003). The Gini coefficient is related to the AUC in the following way

$$\text{Gini} = 2 \cdot \text{AUC} - 1. \tag{18}$$

To measure the accuracy of probability predictions we apply the Brier Score, as introduced by Brier 1950. This score calculates the mean squared error between the predicted default probability $\hat{y}_i = \Pr(y_i = 1|x_i)$ and the observed value $y_i \in \{0, 1\}$ for $i = 1, ...n$, formally denoted as

$$\text{Brier Score} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2. \tag{19}$$

The outcome is a value between zero and one, for which a lower value means higher accuracy.

The last facet to be covered is the correctness of categorical predictions. For this we use the percentage of correctly classified observations (PCC), which refers to the fraction of observations that are correctly classified by the model. This is denoted as follows

$$\text{PCC} = \frac{\sum_{i=1}^{n} \text{I}(\hat{y}_i = y_i)}{n}, \tag{20}$$

where $\text{I}(\hat{y}_i = y_i)$ is an indicator function that is equal to one if the class into which observation i is classified is equal to the observed class. Besides the overall PCC we also consider the PCC for the default class and the non-default class separately. The motivation for this is that credit risk managers can gain better insights in the corresponding costs of wrongly classifying customers. Since the classifiers in this study produce probability predictions, we have to transform these to discrete classes to be able to evaluate the correctness of categorical predictions. In order to do so, the predicted probability $\Pr(y_i = 1|x_i)$ is compared to a threshold τ in the following way

$$\begin{aligned} \hat{y}_i = 1 \quad &\text{if} \quad \Pr(y_i = 1|x_i) > \tau, \\ \hat{y}_i = 0 \quad &\text{if} \quad \Pr(y_i = 1|x_i) < \tau. \end{aligned} \tag{21}$$

Practically, the threshold value would be chosen based on the costs that are associated with accepting *bad* customers and rejecting *good* customers (Hand 2005). However, since this information is not available, we set the threshold such that the fraction of default observations is equal to the prior default rate in the training set (Lessmann et al. 2015).

4 Results

This section consist of four parts in which the performance of the different classifiers are evaluated and compared. First, the choice of the benchmark model is determined based on the performance of the logistic regression in combination with the best-performing variable selection technique. Secondly, the benchmark model is compared to all the competing classifiers for the Low Default Rate. Thirdly, we discuss new insights that result from using the interpretability characteristic of the revised logistic regression. Lastly, the sensivitity analysis is performed on the best-performing classifier at varying degrees of class imbalance corresponding to the different default rates. All models are evaluated based on the five performance measures.

4.1 Benchmark model

Firstly, we select the benchmark model based on the evaluation of the performance of the logistic regression (LR) in combination with backward elimination, forward selection and LASSO regularization on the validation set. Tabel 4 presents the results of the base model with the different traditional variable selection techniques.

Table 4: Performance measures of the logistic regression in combination with traditional variable selection techniques for the different default rates. Bold face indicates the best model per metric for the respective default rate.

			High Default Rate		
	Gini	BS	PCC	PCC defaults	PCC non-defaults
LR with backward elimination	68.81%	**0.0719**	88.44%	40.04%	93.60%
LR with forward selection	68.81%	**0.0719**	88.44%	40.04%	93.60%
LR with LASSO	**68.83%**	0.0723	**88.53%**	**40.10%**	**93.81%**
			Medium Default Rate		
	Gini	BS	PCC	PCC defaults	PCC non-defaults
LR with backward elimination	69.20%	**0.0640**	89.74%	**38.58%**	94.40%
LR with forward selection	69.20%	**0.0640**	89.74%	38.55%	94.40%
LR with LASSO	**69.23%**	0.0642	**89.78%**	**38.58%**	**94.53%**
			Low Default Rate		
	Gini	BS	PCC	PCC defaults	PCC non-defaults
LR with backward elimination	69.98%	**0.0536**	**91.36%**	**36.27%**	**95.37%**
LR with forward selection	69.96%	**0.0536**	**91.36%**	36.25%	95.36%
LR with LASSO	**70.03%**	**0.0536**	**91.36%**	**36.27%**	**95.37%**

As can be seen in Table 4, the performances between the models differ just slightly, but for all three default rates the logistic regression regularized by LASSO performs the best. Hence, we adopt this model as the benchmark in this study. This finding indicates that most of the included predictors have significant explanatory power, such that the variable selection methods end up with a very similar set of features, and thus also a very similar model performance.

4.2 Competing Models

4.2.1 Performance Comparison

Table 5 presents the performance of the benchmark model, the stand-alone machine learning models and the competing revised logistic regression models in terms of the five performance measures on the test set. Bold faced values indicate the best performance per metric over all the models, where both bold faced and underlined values indicate the best performance among the interpretable models. In order to create a clear comparison between the models Table 5 only shows the results for the Low Default Rate in the target variable and thus the highest class imbalance. Performances of the best-performing models at varying degrees of class imbalance are considered in the sensitivity analysis in the following section. If one is interested in the full overview of the performance of all models for the different default rates, please refer to Table 15 in the Appendix.

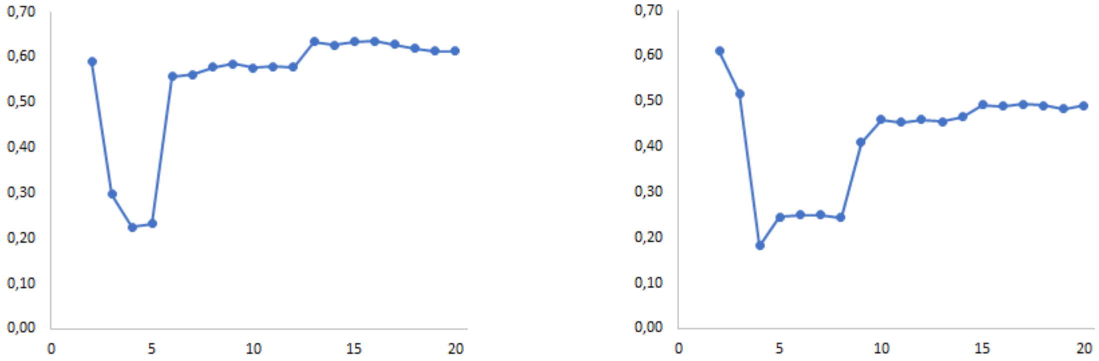
Table 5: Performance measures of the different models for the Low Default Rate. Bold face indicates the best performance per metric over all the models. Bold face and underlined indicates the best performance among the interpretable models.

	Low Default Rate				
	Gini	BS	PCC	PCC defaults	PCC non-defaults
LR with LASSO	69.15%	0.0542	91.20%	35.10%	95.28%
Random forest	81.62%	0.0440	93.39%	51.27%	96.46%
XGBoost	**84.39%**	**0.0435**	**93.57%**	**52.54%**	**96.55%**
SAFE ML with RF	74.08%	0.0522	91.59%	39.08%	95.41%
SAFE ML with XGB	72.43%	0.0519	91.10%	**45.11%**	94.44%
Hybrid with RF	63.03%	0.0696	89.64%	23.56%	94.44%
Hybrid with XGB	61.19%	0.0588	90.05%	27.35%	94.68%
Hybrid & SAFE with RF	**79.41%**	**0.0492**	**92.41%**	44.25%	**95.91%**
Hybrid & SAFE with XGB	71.70%	0.0525	91.57%	37.93%	94.47%

Several findings emerge from Table 5. Firstly, both complex machine learning models, random forest and XGBoost, outperform the benchmark on all performance measures, in particular a substantial improvement regarding the Gini coefficient is observed. This confirms the expectation as laid down in this paper, and the earlier findings from existing literature (e.g. Dumitrescu et al. 2018 and Lessmann et al. 2015) that the tree-based ensemble classifiers have an advantage compared to traditional statistical models. The difference in PCC is mainly due to the large gain in PCC for the default class. There is not a large gain in the PCC for non-defaults because naive classifiers such as logistic regression tend to predict the majority class, such that the PCC for this class is very high.

Applying the SAFE ML method to the complex classifiers also improves upon the standard logistic regression. Again, a substantial improvement is observed for the Gini coefficient, particularly for the SAFE ML method in combination with random forest. This indicates that discriminatory power is gained by changing the data representation as to better present the relationships between the predictors and the probability of default.

On the other hand, adding interaction terms to the original set of predictors in the logistic regression has a negative impact on the performance. The number of added interaction terms are based on the performance of the Hybrid model on the validation set. Figure 9a and 9b show the Gini Coefficient of the Hybrid Approach applied on random forest and XGBoost, respectively, for different numbers of added interaction terms for the Low Default Rate. The graphs corresponding to the performance of the Hybrid models for the other default rates can be found in Figure 11 in the Appendix.



(a) Random forest

(b) XGBoost

Figure 9: Gini Coefficient of Hybrid Approach in combination with random forest **(a)** and XGBoost **(b)** for varying number of interaction terms.

From the graphs we observe that the performance regarding the combination with random forest increases for a certain number of interactions and then stabilizes around that value. As such, we choose the number of interactions for a high performance, but also taking into account the interpretability regarding the number of features to include in the model, and the stability regarding the error in estimation of the parameters. For these reasons, thirteen interaction terms are added to the model with the Low Default Rate. For the Hybrid Approach in combination with XGBoost we observe another pattern. Namely, for a low number of interactions a high performance is realized, which drops heavily when more interactions are added, to subsequently increase slightly and stabilize at a relatively low performance of the Gini Coefficient of around 50%. As such, we augment the set of original predictors with a low number of interaction terms. For the Hybrid Approach in combination with both random forest and XGBoost the performance is worse than the benchmark in terms of all the five metrics. There is particularly an underperformance for the correctness of predictions for the default class. These results imply that the generated second-order interaction terms do not seem to expose relevant relationships with the probability of default. Another reason might be that the interaction terms have a non-linear relationship with the target variable as identified by random forest or XGBoost, which cannot be captured by the logistic regression. This can cause a lot of noise, leading to a bad performance.

However, when we transform the original set of predictors augmented by the identified interaction terms by means of the SAFE ML method, there is a substantial improvement regarding all performance measures. In particular, when we apply this technique in combination with random forest, this results in the best-performing model for all the metrics, except for the PCC for the default class which is second-best among the interpretable models. The biggest outperformance is observed for the Gini coefficient, where the combination of the Hybrid Approach with the SAFE ML method on random forest shows an improvement of more than 10% compared to the benchmark. Taking into account Henley et al. 1997 who state that an increase in discriminatory power even by a fraction of a per cent could translate into significant future savings, this improvement can have a very large impact. Additionally, the PCC for defaults increases from 35.10% to 44.25%. Since the costs associated with accepting bad applicants can be enormous, this gain in correctly classifying these applicants is financially very beneficial. There is still a lot of room for improvement, but compared to the benchmark this is a great step.

The Hybrid Approach in combination with the SAFE ML method applied to XGBoost also outperforms the benchmark model, but performs worse than applying solely the SAFE ML method in terms of the majority of the performance measures. This insinuates that the interactions found by XGBoost do not constitute to relevant relationships with the probability of default, even after changing the data representation with the SAFE ML method. As such, XGBoost performs the best as an individual classifier but does not contribute to the performance when we extract information in terms of interaction terms or single-variable transformations from the model.

The revised logistic regression as constructed by the Hybrid Approach in combination with the SAFE ML method applied to random forest does not outperform the complex machine learning models, but it does perform competitively, especially in comparison to the random forest. In particular, the difference in the Gini coefficient is slightly more than 2% and the Brier Score differs by just 0.0052. This implies that the discriminatory power and the accuracy of probability predictions is very comparable between random forest and the revised logistic regression, whereas there is a huge gain in interpretability.

4.2.2 Interpretability characteristic

In this section we discuss some insights that are obtained using the interpretability characteristic of the revised logistic regression. In particular, several coefficients of predictors in the best-performing interpretable model are compared to those of the standard logistic regression. We focus on two different aspects: (1) the impact of individual variables in the standard LR versus the individual variables transformed to binary features by SAFE ML in the revised LR, and (2) the impact of individual variables in the standard LR versus the interaction terms as constructed by the Hybrid Approach and transformed by SAFE ML in the revised LR. For both aspects we consider one variable as an example in order to reveal the interpretability characteristic.

Table 6 presents the impact of the variable `Days without Tax Payments` as individual predictor in the standard logistic regression with LASSO and as transformed binary features in the revised logistic regression (Hybrid & SAFE with RF). The SAFE ML method has transformed the continuous variable into four binary variables corresponding to four different intervals. The first binary variable is excluded from the set to prevent multicollinearity among the independent variables. The remaining variables obtain a coefficient as estimated by the logistic regression. The negative coefficients for customers that have 13-28 days without paying tax (`Days without Tax Payments` $\left[13.48,22.54\right]$ and `Days without Tax Payments` $\left[22.54,28.06\right]$) indicate that all else being equal, these customers are less likely to go in default than customers that have less than 13 days or more than 28 days without paying tax. The p-values indicate that these newly constructed features are statistically significant. The standard logistic regression estimates the coefficient of `Days without Tax Payments` as positive without making any distinction for different values of the variable. For this the interpretation is that all else being equal, customers with more days without paying tax are more likely to default. Hence, in addition to improving the performance of the standard framework of the logistic regression, we also gain more insights into the drivers in the model.

Table 6: Predictor variable `Days without Tax Payments` with estimated coefficients for benchmark model (LR with LASSO) and revised logistic regression (Hybrid & SAFE with RF)

LR with LASSO			Hybrid & SAFE with RF			
Predictor	Coefficient	p-value	Predictor	Interval	Coefficient	p-value
`Days without Tax Payments`	0.0507	0.0000	`Days without Tax Payments`	$[-\infty, 13.48]$	-	-
			`Days without Tax Payments`	$[13.48, 22.54]$	-0.1874	0.0000
			`Days without Tax Payments`	$[22.54, 28.06]$	-0.2772	0.0000
			`Days without Tax Payments`	$[28.06, \infty]$	0.3882	0.0000

Table 7 presents the impact of the interaction term {`Number of Transactions` · `Percentage of Type 3 Transactions`} as generated by the Hybrid Approach and transformed to binary features by means of the SAFE ML method. The individual variables `Number of Transactions` and `Percentage of Type 3 Transactions`, their estimated coefficients in the standard logistic regression and corresponding p-values are also shown. The newly generated interaction term is easy to interpret, namely it corresponds to the number of type 3 transactions. When we include the variables individually in the standard framework, the estimated coefficient for `Number of Transactions` is negative (-0.0217), and for `Percentage of Type 3 Transactions` is positive (1.9598), both statistically significant. In contrast to this, Table 7 shows that the estimated coefficient for the interaction term is positive but deviating for all the different intervals. The p-values also show that three of the four features are statistically significant for a significance level of 5%, and all four for a significance level of 10%. Hence, we are not limited to the argument that all else being equal, customers are more likely to default for more type 3 transactions, but we can specify the impact for several ranges. This yields additional insights into the impact of the predictors, in this case the types of transactions made by the customer.

Table 7: Generated interaction terms {`Number of Transactions` · `Percentage of Type 3 Transactions`} with estimated coefficients for the revised logistic regression (Hybrid & SAFE with RF) and corresponding individual variables with estimated coefficients in the benchmark model (LR with LASSO).

LR with LASSO			Hybrid & SAFE with RF			
Predictor	Coefficient	p-value	Predictor	Interval	Coefficient	p-value
`Number of Transactions`	-0.0217	0.0000	{`Number of Transactions` · `Percentage of Type 3 Transactions`}	$[-\infty, 2.52]$	-	-
`Percentage of Type 3 Transactions`	1.9598	0.0000	{`Number of Transactions` · `Percentage of Type 3 Transactions`}	$[2.52, 3.53]$	0.9131	0.0983
			{`Number of Transactions` · `Percentage of Type 3 Transactions`}	$[3.53, 5.49]$	0.5262	0.0455
			{`Number of Transactions` · `Percentage of Type 3 Transactions`}	$[5.49, 13.01]$	0.3482	0.0203
			{`Number of Transactions` · `Percentage of Type 3 Transactions`}	$[13.01, \infty]$	1.1833	0.0143

The variables that are used as examples are chosen based on their significant impact on the probability of default for the different intervals and their ease of interpretation. Other insights that can be obtained from this model are for example the number of type 1 transactions (`Number of Transactions · Percentage of Type 1 Transactions`) together with the money that is earned with it (`Net Inflow · Percentage of Type 1 Transactions`) or the relation between the number of days that a company exists and the days that it is registered at Knab (`Registered Days · Existing Days`). The main takeaway from the results in Table 6 and 7 is that the revised logistic regression retains the ability to interpret the outcome of the model by providing the exact relationship between the predictors and the target variable, and additionally yields more detailed insights from the discretization and new insights from the generated interaction terms.

4.2.3 Sensitivity analysis

The focus of the sensitivity analysis regarding the class imbalance is divided in three aspects: (1) the performance of the classifiers, (2) the generated interaction terms, and (3) the created clusters of categorical variables. We evaluate among the best-performing model per default rate to what extent these aspects differ, in order to determine how sensitive the model is with respect to class imbalance. As shown in Table 5 in Section 4.2.1 the best-performing model for the Low Default Rate is the combination of the Hybrid Approach with the SAFE ML method applied on random forest. We find the same result for the other two default rates, which can be seen in Table 15 in the Appendix. As such, we evaluate solely this best-performing model for the varying degrees of class imbalance for the sensitivity analysis.

Table 8 presents the performance of the model for the different default rates. We find several interesting results and patterns by comparing the five performance measures. To start, the classifier shows an increasing pattern in performance for higher class imbalance in terms of the Brier Score, the PCC and the PCC for the non-defaults class. Particularly, as the default rate decreases from 9.64% to 8.35% to 6.78%, an increasing performance in the Brier Score is observed of 0.0675 to 0.0618 to 0.0492, respectively. From this we can argue that the accuracy of probability predictions actually improves for higher class imbalance. The same inference is drawn for the correctness of categorical predictions, since both the overall PCC and PCC for the non-defaults class show a similar pattern. Another remarkable finding is the difference in the Gini Coefficient between the High Default Rate and the Low Default Rate. In particular, the classifier in the Low Default Rate scores almost 4.5% higher, indicating that also the discriminatory ability of the model improves for higher class imbalance.

Table 8: The performance of the Hybrid Approach in combination with SAFE ML applied to random forest for the different default rates.

	Performance		
	High Default Rate	Medium Default Rate	Low Default Rate
Gini	74.93%	73.70%	79.41%
Brier Score	0.0675	0.0618	0.0492
PCC	89.45%	90.15%	92.41%
PCC defaults	45.16%	41.12%	44.25%
PCC non-defualts	94.18%	94.61%	95.91%

Since credit scoring data sets often experience high class imbalance (Brown and Mues 2012), naive classifiers tend to predict the majority class, such that the performance on both the correctness of categorical predictions for the default class as the accuracy of probability predictions is relatively poor (Baesens et al. 2003). The results as shown in Table 8 proof that the performance of the revised logistic regression actually improves on all the three notions of performance for higher class imbalance for the given default rates, which is a very desirable finding for practitioners of credit scoring.

Table 9 shows the interaction terms as constructed by the model for the different default rates. We use a specific notational convention whereby the interaction terms that occur for two different default rates are denoted in bold face, and interactions that occur for all three default rates are underlined and denoted in bold face.

A first clear observation is that the number of identified interaction terms differ among the varying class imbalance: nine for the model corresponding to the High Default Rate, sixteen for the Medium Default Rate, and thirteen for the Low Default Rate. These results show that higher class imbalance demands more interaction terms to reach the optimal performance. As such, the number of identified interaction terms is not robust with respect to class imbalance.

Another interesting result is that eight of the nine interaction terms for the High Default Rate are also identified for the other two default rates. For the Low Default Rate twelve out of the thirteen also occur in the model with the Medium Default Rate. This indicates that there is a high recurrence of interaction terms for varying default rates, meaning that almost the complete set of interactions is a subset of a model with a larger set. This is because the interaction terms are included one by one based on the highest contribution of the pairwise interaction. From these two findings we can conclude that the number of identified interaction terms is not robust with respect to class imbalance, but the importance of the interactions is.

Table 9: The generated interaction terms for the different default rates. Interactions indicated in bold occur for two default rates and interactions that are underlined and indicated in bold occur for all three default rates.

Interaction terms		
High Default Rate	Medium Default Rate	Low Default Rate
{**Net Inflow · Number of Transactions**}	{**Net Inflow · Number of Transactions**}	{**Net Inflow · Number of Transactions**}
{**Number of Transactions · Percentage of Type 1 Transactions**}	{**Number of Transactions · Percentage of Type 1 Transactions**}	{**Number of Transactions · Percentage of Type 1 Transactions**}
{**Percentage of Type 1 Transactions · Percentage of Type 3 Transactions**}	{**Percentage of Type 1 Transactions · Percentage of Type 3 Transactions**}	{**Percentage of Type 1 Transactions · Percentage of Type 3 Transactions**}
{**Number of Transactions · Percentage of Type 3 Transactions**}	{**Number of Transactions · Percentage of Type 3 Transactions**}	{**Number of Transactions · Percentage of Type 3 Transactions**}
{**Existing Days · Number of Transactions**}	{**Existing Days · Number of Transactions**}	{**Existing Days · Number of Transactions**}
{**Net Inflow · Percentage of Type 3 Transactions**}	{**Net Inflow · Percentage of Type 3 Transactions**}	{**Net Inflow · Percentage of Type 3 Transactions**}
{**Registered Days · Existing Days**}	{**Registered Days · Existing Days**}	{**Registered Days · Existing Days**}
{**Net Inflow · Percentage of Type 1 Transactions**}	{**Net Inflow · Percentage of Type 1 Transactions**}	{**Net Inflow · Percentage of Type 1 Transactions**}
{**Registered Days · Number of Transactions**}	{Net Inflow · Tax}	{**Registered Days · Number of Transactions**}
	{**Percentage of Type 3 Transactions · Percentage of Negative Transactions**}	{**Percentage of Type 3 Transactions · Percentage of Negative Transactions**}
	{**Registered Days · Percentage of Type 3 Transactions**}	{**Registered Days · Percentage of Type 3 Transactions**}
	{**Net Inflow · Percentage of Negative Transactions**}	{**Net Inflow · Percentage of Negative Transactions**}
	{Existing Days · Percentage of Type 3 Transactions}	{Existing Days · Net Inflow}
	{Number of Transactions · Percentage of Negative Transactions}	
	{Percentage of Type 1 Transactions · Percentage of Negative Transactions}	
	{Net Inflow · Days without Tax Payments}	

The last focus point of the sensitivity analysis is the difference in the created clusters of categories for the three categorical features: Legal Entity Type, Industry and Province. The corresponding clusters are shown in Table 10 as created by the model for the varying degrees of class imbalance. It can be seen that for Industry and Province the number of created clusters is similar among the three default rates. This is however not the case for Legal Entity Type, where the model with the Low Default Rate creates a separate cluster for each category. It can be argued that the number of clusters is semi robust to the class imbalance. However, few to almost no similarities can be observed within the created clusters among the models. This indicates that the specific construction of clusters is not robust for varying default rates, such that clusters can be very different for higher or lower class imbalance.

Table 10: The created clusters of categorical variables for the different default rates.

Categorical Clusters		
High Default Rate	Medium Default Rate	Low Default Rate
Legal Entity Type 1. Besloten Vennootschap-Maatschap-Stichting 2. Eenmanszaak-Venootschap Onder Firma	Legal Entity Type 1. Eenmanszaak-Stichting-Venootschap Onder Firma 2. Besloten Vennootschap-Maatschap	Legal Entity Type 1. Eenmanszaak 2. Maatschap 3. Stichting 4. Vennootschap 5. Besloten Vennootschap
Industry (SBI Codes) 1. A-B-K-L-M-O-S-T 2. E-F-I-N 3. C-J-R 4. D-H-P-Q 5. G	Industry (SBI Codes) 1. A-D-E-H-J-K-M-O-P-T 2. F-G 3. C-L-N-R-Q-S 4. B-I	Industry (SBI Codes) 1. A-B-E-F-M-O-P-Q-T 2. C-J-K-N-R-S- 3. D-G-I 4. H-L
Province 1. Drenthe-Zuid Holland-Noord Holland 2. Flevoland-Friesland-Groningen-Noord Brabant 3. Gelderland-Zeeland 4. Limburg-Overijssel-Utrecht	Province 1. Drenthe-Friesland-Zeeland-Zuid Holland 2. Flevoland-Gelderland-Limburg 3. Groningen 4. Noord Brabant-Noord Holland-Overijssel-Utrecht	Province 1. Drenthe-Zuid Holland-Noord Holland-Friesland -Noord Brabant 2. Flevoland-Zeeland 3. Gelderland-Limburg-Utrecht 4. Groningen-Overijssel

5 Conclusion

In this paper is investigated whether we can improve the standard framework of the logistic regression by means of complex machine learning techniques in credit scoring applications. The traditional logistic regression is the industry workhorse in credit scoring because of its ease of interpretation, which is an important characteristic for both regulators and credit risk managers. Hence, the revised model does not only endeavor to improve the performance of the standard framework, but should also preserve its simple interpretation. Additionally, we investigated whether the revised logistic regression performs competitively to the stand-alone black box models. The benchmark model is constructed by applying LASSO regularization to the logistic regression. We use random forest and XGBoost as individual classifiers and as underlying models to extract relevant information from in terms of transformed single-variables and constructed interaction terms, to include in the revised logistic regression. We assess the comparison between the benchmark model and the competing models on five performance measures covering the different facets of classifier performance. The study is conducted on a credit scoring data set containing characteristics and transactional behavior of self-employed persons and small business owners. The data does not contain defaults of the customers, such that we have approximated this by insolvency. In doing so, we have constructed a varying default rate among the customers, resulting in three different sets of the target variable in order to perform a sensitivity analysis.

We find that the two black box machine learning models substantially outperform the benchmark model on all five performance measures. This confirms the prior expectation that more advanced classifiers have an advantage compared to the traditional statistical model. This might indicate that the respective data set is non-linear. Changing the data representation by discretizing the continuous variables and clustering the categorical variables by means of the SAFE ML method also improves performance compared to the benchmark. This is the first model to combine machine learning techniques with the logistic regression, and proofs that the performance of the standard framework can be improved, while preserving the ability to identify the drivers and interpret the corresponding relationships with the probability of default.

However, the second technique that we use in order to reach this goal results in less promising findings. Namely, enhancing the original set of predictors with interaction terms as identified by the Hybrid Approach leads to underperformance compared to the benchmark, in particular a worse performance for all the measures results from applying this technique. In contrast to this, we find the best performance among all interpretable models when the Hybrid Approach and the SAFE ML method are combined. In other words, when the identified interaction terms are added to the original set of predictors and transformed into a set of new binary features, a substantial outperformance of the benchmark is realized. From these results we can conclude that the interaction terms identified by the Hybrid Approach lead to relevant features, but the true relationships with the probability of default can only be captured after transforming the respective set to newly engineered features. Combining these newly engineered features with the interpretability characteristic of the revised logistic regression, allows us to gain new and more detailed insights into the predictors in the model. Additionally, the sensitivity analysis also proofs that the performance of the model is robust with respect to varying default rates and performs well in the presence of large class imbalance. Most importantly, the results proof that we can improve the framework of the traditional logistic regression by means of machine learning techniques, while preserving its simple interpretation. An additional inference can be drawn, that not only the revised logistic regression outperforms the standard framework, it also performs competitively to the random forest classifier. By using the proposed methodology, any financial institution could increase future savings (Henley et al. 1997), accurately determine the financial buffer as required by the Basel capital accord and simultaneously easily interpret the marginal contribution of each predictor.

We encountered several limitations while conducting this research. The methodology as described in this paper determines the optimal number of interaction terms based on the performance of the Hybrid Approach. This set is added to the original set of predictors and used for both the Hybrid Approach, as well as for the combination with the SAFE ML method. However, it might not be the optimal set of interaction terms after performing the transformations by means of the SAFE ML method. This causes higher uncertainty with respect to parameter estimation because the set is assumed to be good. Secondly, the applied methodology demands extensive computation power, such that running the different algorithms for large data sets takes a lot of time to complete. This is less practical when the model has to be recalibrated by, for example, credit risk managers on a frequent base.

Further research that could be interesting based on the findings in this paper would be to firstly develop an algorithm that entwines the methodologies of the Hybrid Approach and the SAFE ML method in one step. Such an algorithm would allow the model to include the optimal number of interaction terms, while taking into account that these will be discretized after including them in the original set of predictors. In this way, the uncertainty regarding the selection of the interaction terms of the first step is eliminated. Another interesting topic for future work is to apply SHAP values in the SAFE ML method as alternative to the partial dependence function. In this way, a consistent feature attribution method is applied to determine the model's output. Lastly, another practical extension of this research would be to create a faster algorithm or heuristic to lower the required computational power and consequently diminish the running time, such that recalibration of the model is more pragmatic.

References

[1] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. “Benchmarking state-of-the-art classification algorithms for credit scoring”. In: *Journal of the Operational Research Society* 54.6 (2003), pp. 627–635.

[2] A. N. Berger, W. S. Frame, and N. H. Miller. “Credit scoring and the availability, price, and risk of small business credit”. In: *Journal of Money, Credit and Banking* (2005), pp. 191–222.

[3] K. Bijak and L. C. Thomas. “Does segmentation always improve model performance in credit scoring?” In: *Expert Systems with Applications* 39.3 (2012), pp. 2433–2442.

[4] A. Blöchlinger and M. Leippold. “Economic benefit of powerful credit scoring”. In: *Journal of Banking & Finance* 30.3 (2006), pp. 851–873.

[5] L. Breiman. “Bagging predictors”. In: *Machine Learning* 24.2 (1996), pp. 123–140.

[6] L. Breiman. “Random forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32.

[7] G. W. Brier. “Verification of forecasts expressed in terms of probability”. In: *Monthly Weather Review* 78.1 (1950), pp. 1–3.

[8] I. Brown and C. Mues. “An experimental comparison of classification algorithms for imbalanced credit scoring data sets”. In: *Expert Systems with Applications* 39.3 (2012), pp. 3446–3453.

[9] T. Chen and C. Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794.

[10] J. N. Crook, D. B. Edelman, and L. C. Thomas. “Recent developments in consumer credit risk assessment”. In: *European Journal of Operational Research* 183.3 (2007), pp. 1447–1465.

[11] E. Dumitrescu, S. Hue, C. Hurlin, and S. Tokpavi. “Machine Learning for Credit Scoring: Improving Logistic Regression with Non Linear Decision Tree Effects”. PhD thesis. Doctoral dissertation, 2018.

[12] T. Fawcett. “An introduction to ROC analysis”. In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874.

[13] S. Finlay. “Multiple classifier architectures and their application to credit risk assessment”. In: *European Journal of Operational Research* 210.2 (2011), pp. 368–378.

[14] Y. Freund, R. E. Schapire, et al. “Experiments with a new boosting algorithm”. In: *ICML*. Vol. 96. Citeseer. 1996, pp. 148–156.

[15] J. H. Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of Statistics* (2001), pp. 1189–1232.

[16] K. Fujimoto, I. Kojadinovic, and J.-L. Marichal. “Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices”. In: *Games and Economic Behavior* 55.1 (2006), pp. 72–99.

[17] A. Gosiewska, A. Gacek, P. Lubon, and P. Biecek. “SAFE ML: Surrogate Assisted Feature Extraction for Model Learning”. In: *arXiv preprint arXiv:1902.11035* (2019).

[18] D. J. Hand. “Good practice in retail credit scorecard assessment”. In: *Journal of the Operational Research Society* 56.9 (2005), pp. 1109–1117.

[19] W. Henley et al. “Construction of a k-nearest-neighbour credit-scoring system”. In: *IMA Journal of Management Mathematics* 8.4 (1997), pp. 305–321.

[20] C.-L. Huang, M.-C. Chen, and C.-J. Wang. “Credit scoring with a data mining approach based on support vector machines”. In: *Expert Systems with Applications* 33.4 (2007), pp. 847–856.

[21] P. R. Kumar and V. Ravi. “Bankruptcy prediction in banks and firms via statistical and intelligent techniques–A review”. In: *European Journal of Operational Research* 180.1 (2007), pp. 1–28.

[22] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas. “Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research”. In: *European Journal of Operational Research* 247.1 (2015), pp. 124–136.

[23] J. J. Levy and A. J. O’Malley. “Do not dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning”. In: *BMC Medical Research Methodology* 20.1 (2020), pp. 1–15.

[24] S. Lundberg, G. G. Erion, and S.-I. Lee. “Consistent individualized feature attribution for tree ensembles”. In: *arXiv preprint arXiv:1802.03888* (2018).

[25] S. Lundberg and S.-I. Lee. “A unified approach to interpreting model predictions”. In: *arXiv preprint arXiv:1705.07874* (2017).

[26] C. Luo, D. Wu, and D. Wu. “A deep learning approach for credit scoring using credit default swaps”. In: *Engineering Applications of Artificial Intelligence* 65 (2017), pp. 465–470.

[27] M. Pal. “Random forest classifier for remote sensing classification”. In: *International Journal of Remote Sensing* 26.1 (2005), pp. 217–222.

[28] G. Paleologo, A. Elisseeff, and G. Antonini. “Subagging for credit scoring models”. In: *European Journal of Operational Research* 201.2 (2010), pp. 490–499.

[29] L. Rokach and O. Maimon. “Clustering methods”. In: *Data Mining and Knowledge Discovery Handbook*. Springer, 2005, pp. 321–352.

[30] C. Truong, L. Oudre, and N. Vayatis. “A review of change point detection methods”. In: *arXiv preprint arXiv:1801.00718* (2018).

[31] T. Van Gestel, B. Baesens, P. Van Dijcke, J. Suykens, J. Garcia, and T. Alderweireld. “Linear and nonlinear credit scoring by combining logistic regression and support vector machines”. In: *Journal of Credit Risk* 1.4 (2005).

A Appendix

A.1 Additional Feature Information

A.1.1 Feature Descriptions

Table 11: Descriptions of the features corresponding to the characteristics of the customers and their transactional behavior.

Features	Type	Description
Employees	Characteristic	The number of employees in the company.
Registered Days	Characteristic	The number of days that the company is registered at Knab.
Existing Days	Characteristic	The number of days since the foundation of the company.
SBI Sector	Characteristic	The industry in which the company is active, where SBI stands for Standaard Bedrijfsindeling. In english this is referred to as Standard Industrial Classification (SIC).
Province	Characteristic	The province in which the company is located.
Legal Entity Type	Characteristic	The type of the entity of the company. The different entity types are: Besloten Vennootschap, Eensmanszaak, Maatschap, Stichting and Vennootschap Onder Firma
Net Inflow	Transactional behavior	The net amount of all incoming (+) and outgoing (-) transactions.
Tax	Transactional behavior	The net amount of all incoming (+) and outgoing (-) transactions related to tax.
Number of Transactions	Transactional behavior	The total number of transactions made.
Days without Tax Payments	Transactional behavior	The number of consecutive days that no tax is paid.
Crypto Trader	Transactional behavior	Indicator whether the company trades in cryptocurrency.
Money Service	Transactional behavior	Indicator that the company might be registered at another bank. Determined by checking the contra account holder on a list of money service businesses.
Fraud Victim	Transactional behavior	Indicator that the company might be a victim of fraud. Determined by keywords in the transaction description.
High Risk Country	Transactional behavior	Indicator whether the company has incoming or outgoing transactions to a high risk country.
Medium Risk Country	Transactional behavior	Indicator whether the company has incoming or outgoing transactions to a medium risk country.

Transaction Type	Transactional behavior	The type of the transaction. The different transaction types are:
		Type 1 {Urgent Incoming, International Incoming, SEPA Incoming, Payment Request Incoming, SEPA Direct Ideal, SEPA ID Deposit Ideal, Term Deposit, Automatic Rebalancing, Fee, Interest}
		Type 2 {Urgent Outgoing, International Outgoing, SEPA Outgoing Ex Ideal, Payment Request Outgoing, SEPA Direct Debit Denial, SEPA Outgoing Ideal, Instant Payment, Costs International Outgoing, Costs Urgent Outgoing, International Return}
		Type 3 {ATM Foreign, ATM Euro, Fuel Foreign, Fuel Euro, POS Foreign, OVB Manual OPS, SEPA Internal, Unknown}

A.1.2 Descriptive Statistics: Characteristics

Table 12: Descriptive statistics of features corresponding to the characteristics of the customers.

Features	Continuous/Categorical	Mean	Stand. Dev.	Min	Max	Skew	Kurt
Employees	Continuous	0.9597	1.9610	0	200	62.3768	6018.4151
Registered Days	Continuous	740.2282	408.6101	24	1976	0.5350	-0.4193
Existing Days	Continuous	2196.8935	3111.8139	24	43812	6.1467	64.7674
SBI Sector	Categorical	-	-	-	-	-	-
Province	Categorical	-	-	-	-	-	-
Legal Entity Type	Categorical	-	-	-	-	-	-

A.1.3 Descriptive Statistics: Transactional Behavior

Table 13: Descriptive statistics of features corresponding to the transactional behavior of the customers. There is a total of approximately 9,6 million transactions over the time period January 2018 to December 2020.

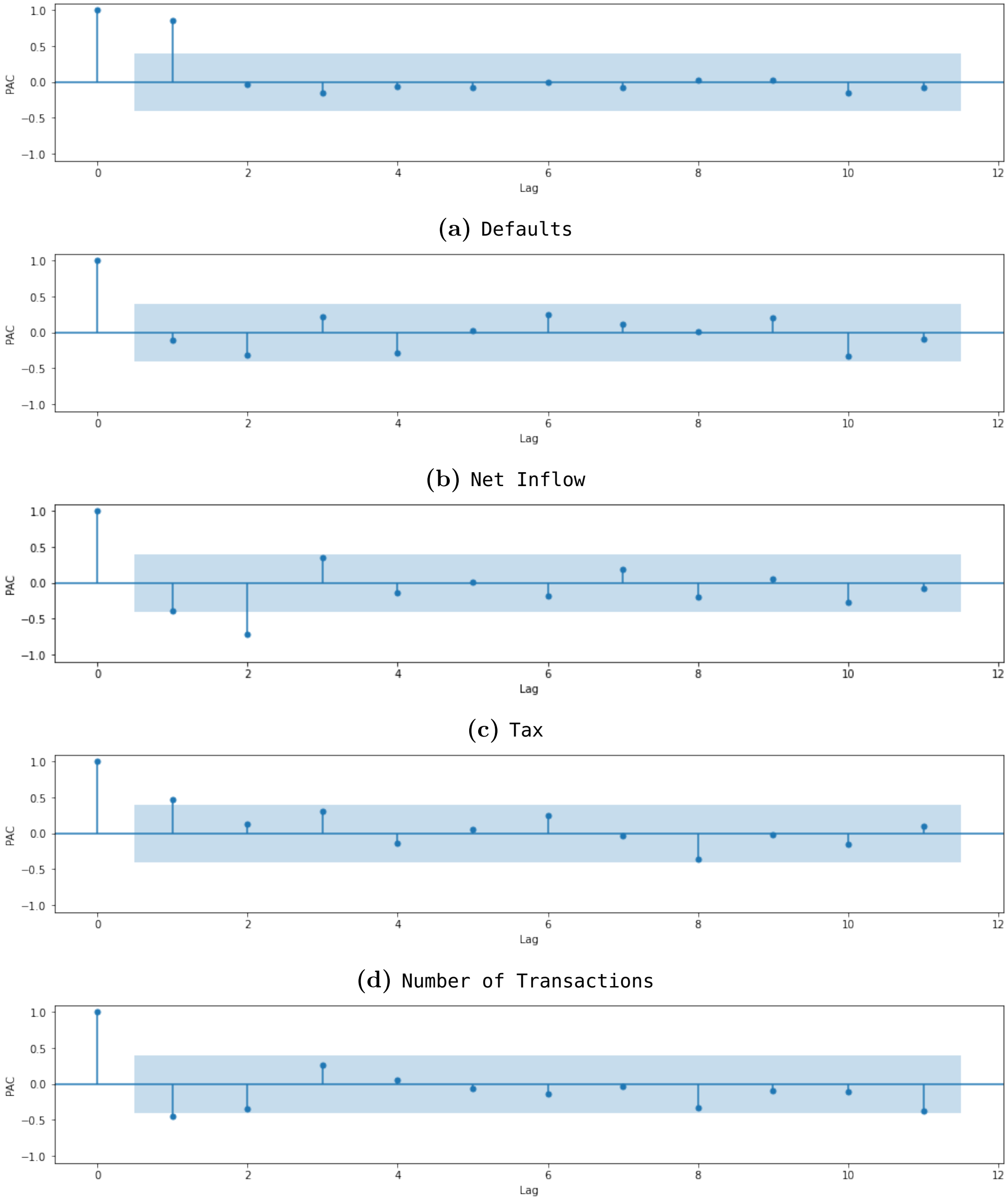
Features	Type	Mean	Stand. Dev.	Min	Max	Skew	Kurt
Net Inflow	Continuous	390.6020	20220.1900	-4192296	1793273	-15.8417	5636.6415
Tax	Continuous	-1037.1693	4056.0458	-291407	214899	-11.8663	355.9228
Number of Transactions	Continuous	23.2934	32.0348	0	2007	7.1150	172.4530
Days without Tax Payments	Continuous	26.9099	4.8144	1	30	-2.0332	4.0008
Crypto Trader	Binary	0.0010	0.0316	0	1	31.5725	994.8211
Money Service	Binary	0.0508	0.2196	0	1	4.0902	14.7210
Fraud Victim	Binary	0.0112	0.1052	0	1	9.2914	84.3294
High Risk Country	Binary	0,0005	0.0225	0	1	44.3734	1966.9959
Medium Risk Country	Binary	0.0131	0.1139	0	1	8.5508	71.1156
Percentage of Type 1 Transactions	Continuous	0.5478	0.2984	0	1	-0.6459	-0.7306
Percentage of Type 2 Transactions	Continuous	0.2329	0.2257	0	1	0.8184	-0.1422
Percentage of Type 3 Transactions	Continuous	0.2131	0.2883	0	1	1.9279	2.5271
Percentage of Negative Transactions	Continuous	0.7052	0.2160	0	1	-0.6071	0.1469

A.1.4 Industry SBI Codes

Table 14: Standaard Bedrijfsindeling (English: Standard Industrial Classification). Source: https://www.kvktoegankelijk.nl/sbi2019nederlands/

SBI Code	Industry
A	Landbouw, bosbouw en visserij
B	Winning van delfstoffen
C	Industrie
D	Productie en distributie van en handel in elektriciteit, aardgas, stoom en gekoelde lucht
E	Winning en distributie van water; afval- en afvalwaterbeheer en sanering
F	Bouwnijverheid
G	Groot- en detailhandel; reparatie van auto's
H	Vervoer en opslag
I	Logies-, maaltijd- en drankverstrekking
J	Informatie en communicatie
K	Financiële instellingen
L	Verhuur van en handel in onroerend goed
M	Advisering, onderzoek en overige specialistische zakelijke dienstverlening
N	Verhuur van roerende goederen en overige zakelijke dienstverlening
O	Openbaar bestuur, overheidsdiensten en verplichte sociale verzekeringen
P	Onderwijs
Q	Gezondheids- en welzijnszorg
R	Cultuur, sport en recreatie
S	Overige dienstverlening
T	Huishoudens als werkgever; niet-gedifferentieerde productie van goederen en diensten door huishoudens voor eigen gebruik
U	Extraterritoriale organisaties en lichamen

A.1.5 Partial Autocorrelation Functions



(a) `Defaults`

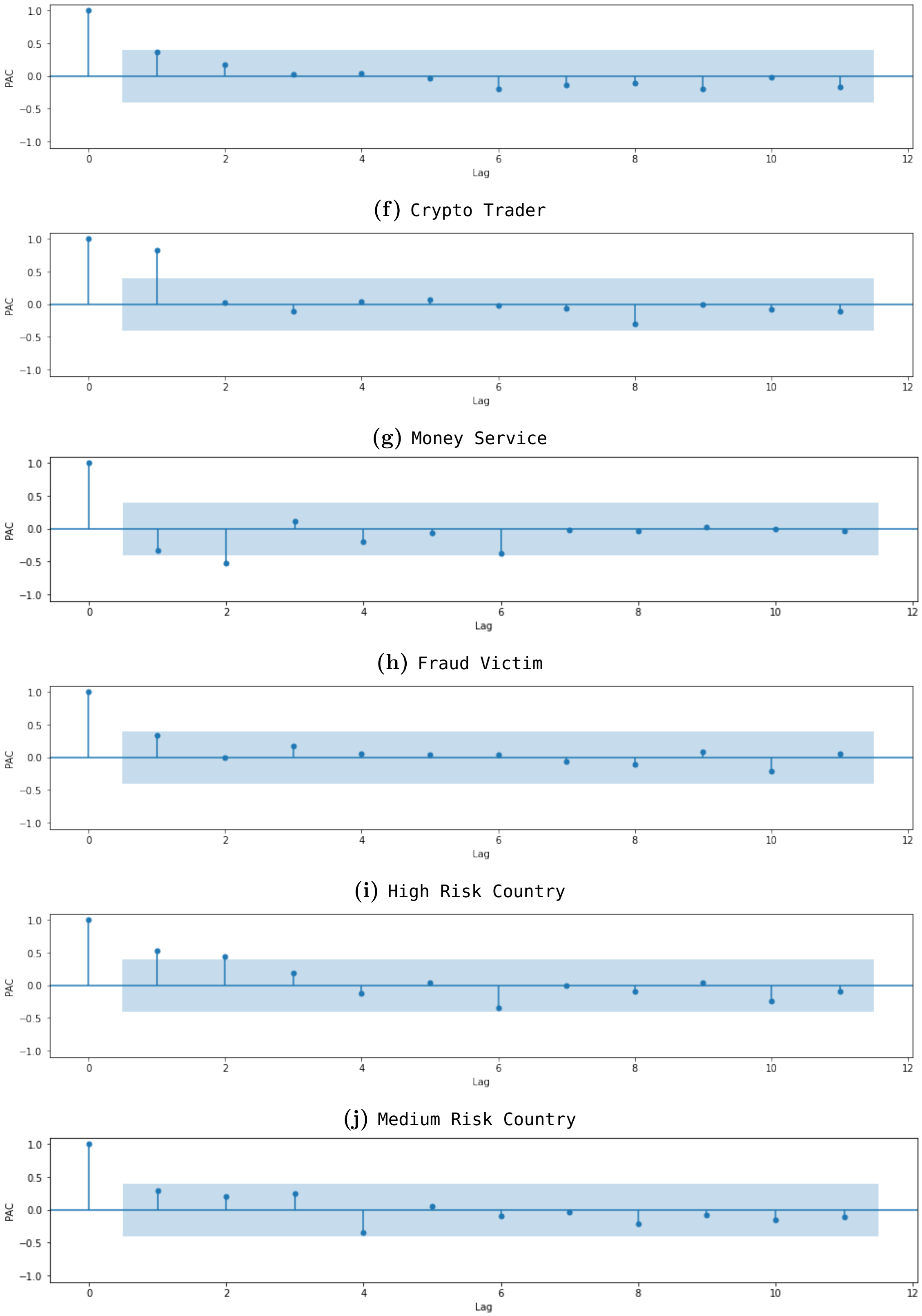
(b) `Net Inflow`

(c) `Tax`

(d) `Number of Transactions`

(e) `Days without Tax Payments`

Figure 10: Partial Autocorrelation Functions for the target variable `Defaults` and all the variables reflecting the transactional behavior together with the corresponding confidence level. This is done for the set with the overall default rate of 8.35%.



(f) `Crypto Trader`

(g) `Money Service`

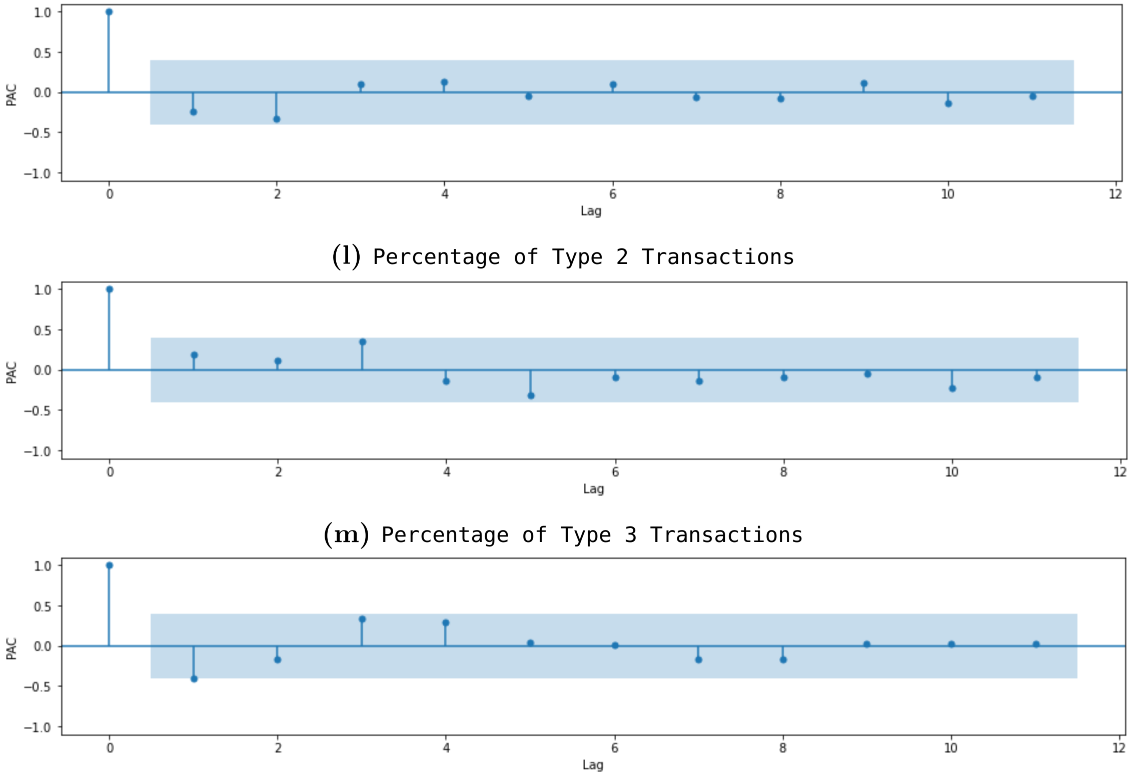
(h) `Fraud Victim`

(i) `High Risk Country`

(j) `Medium Risk Country`

(k) `Percentage of Type 1 Transactions`

Figure 10: Partial Autocorrelation Functions for the target variable `Defaults` and all the variables reflecting the transactional behavior together with the corresponding confidence level. This is done for the set with the overall default rate of 8.35%.



(l) `Percentage of Type 2 Transactions`

(m) `Percentage of Type 3 Transactions`

(n) `Percentage of Negative Transactions`

Figure 10: Partial Autocorrelation Functions for the target variable `Defaults` and all the variables reflecting the transactional behavior together with the corresponding confidence level. This is done for the set with the overall default rate of 8.35%.

A.2 Classifier Performance

Table 15: Performance measures of the different models for the different default rates. Bold face indicates the best performance per metric over all the models. Bold face and underlined indicates the best performance among the interpretable models.

High Default Rate

	Gini	BS	PCC	PCC defaults	PCC non-defaults
LR with LASSO	68.99%	0.0718	88.41%	39.93%	93.59%
Random forest	80.91%	**0.0579**	91.39%	55.39%	95.24%
XGBoost	**83.29%**	**0.0579**	**91.56%**	**56.22%**	**95.33%**
SAFE ML with RF	74.23%	0.0685	89.33%	44.76%	94.09%
SAFE ML with XGB	72.89%	0.0693	89.00%	43.01%	93.91%
Hybrid with RF	65.79%	0.0789	87.07%	33.06%	92.84%
Hybrid with XGB	67.77%	0.0759	87.96%	37.58%	93.34%
Hybrid & SAFE with RF	**74.93%**	**0.0675**	**89.45%**	**45.16%**	**94.18%**
Hybrid & SAFE with XGB	72.08%	0.0695	88.59%	40.81%	93.69%

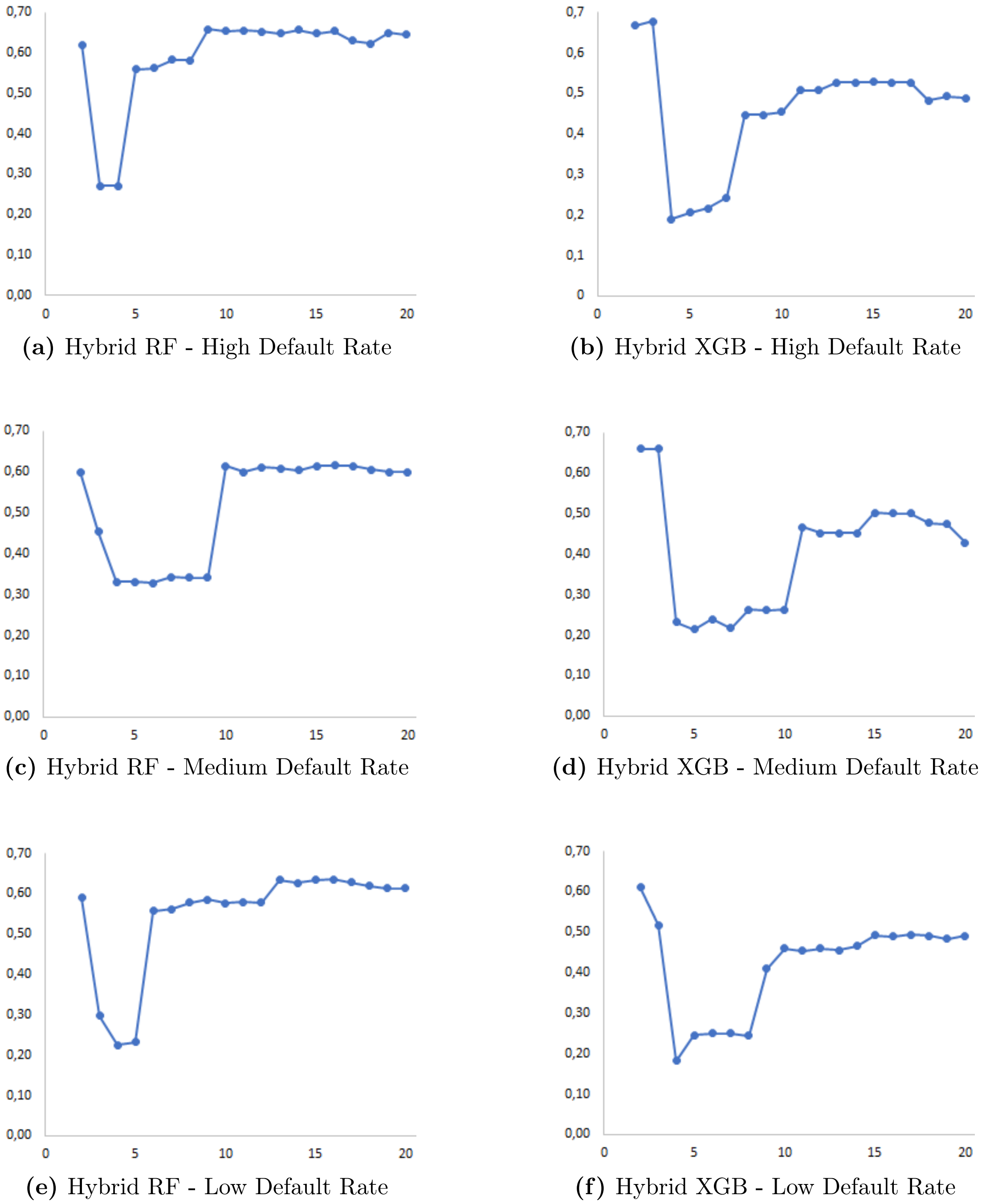
Medium Default Rate

	Gini	BS	PCC	PCC defaults	PCC non-defaults
LR with LASSO	69.59%	0.0639	89.73%	38.48%	94.39%
Random forest	81.90%	**0.0515**	92.28%	53.77%	95.79%
XGBoost	**84.04%**	**0.0515**	**92.54%**	**55.32%**	**95.93%**
SAFE ML with RF	70.07%	0.0635	90.00%	40.19%	94.52%
SAFE ML with XGB	70.24%	0.0637	89.57%	37.53%	94.31%
Hybrid with RF	61.61%	0.0830	88.83%	33.18%	93.89%
Hybrid with XGB	66.01%	0.0688	88.53%	31.47%	93.73%
Hybrid & SAFE with RF	**73.70%**	**0.0618**	**90.15%**	**41.12%**	**94.61%**
Hybrid & SAFE with XGB	70.88%	0.0636	89.76%	38.69%	94.41%

Low Default Rate

	Gini	BS	PCC	PCC defaults	PCC non-defaults
LR with LASSO	69.15%	0.0542	91.20%	35.10%	95.28%
Random forest	81.62%	0.0440	93.39%	51.27%	96.46%
XGBoost	**84.39%**	**0.0435**	**93.57%**	**52.54%**	**96.55%**
SAFE ML with RF	74.08%	0.0522	91.59%	39.08%	95.41%
SAFE ML with XGB	72.43%	0.0519	91.10%	**45.11%**	94.44%
Hybrid with RF	63.03%	0.0696	89.64%	23.56%	94.44%
Hybrid with XGB	61.19%	0.0588	90.05%	27.35%	94.68%
Hybrid & SAFE with RF	**79.41%**	**0.0492**	**92.41%**	44.25%	**95.91%**
Hybrid & SAFE with XGB	71.70%	0.0525	91.57%	37.93%	94.47%

A.3 Performance Hybrid Approach



(a) Hybrid RF - High Default Rate

(b) Hybrid XGB - High Default Rate

(c) Hybrid RF - Medium Default Rate

(d) Hybrid XGB - Medium Default Rate

(e) Hybrid RF - Low Default Rate

(f) Hybrid XGB - Low Default Rate

Figure 11: Gini Coefficient of Hybrid Approach in combination with random forest **(a)**, **(c)**, **(e)** and XGBoost **(b)**, **(d)**, **(f)** for varying number of interaction terms for the different default rates.

A.4 Programming Codes

The programming codes are contained in two separate files: (1) *Main Code.py*, (2) *Performance Measures.py*. Both codes are briefly described in the following sections.

A.4.1 Main Code

The models that are described in the Methodology, Section 3, are developed and programmed in *Main Code.py*. The structure of the code follows the structure of the paper:

- Data connection
- Install packages
- Load & structure data
- Logistic Regression
- Random forest
- XGBoost
- SAFE ML method
- Hybrid Approach
- Combination of Hybrid and SAFE ML

The data is first imported from the database of Knab, after which required open source packages are installed. The tables are loaded and structured to implement in the different models. This includes among other things constructing dummy variables for the categorical variables. The logistic regression is first developed for all the variables, then in combination with stepwise selection (backward and forward) and lastly with LASSO. LASSO first requires tuning of the regularization term. Subsequently, the other models are tuned and trained as described in the Methodology section, after which they are run in order to obtain the corresponding performance on the test set.

A.4.2 Performance Measures

The performance measures that are described in the Methodology, Section 3.6, are developed and programmed in the code called *Performance Measures.py*. The structure of the code is as follows:

- Gini Coefficient
- Brier Score
- Overall PCC
- PCC default class
- PCC non-default class

These are the metrics used to evaluate and compare the different models.