Credit scoring for small medium enterprises using transaction data

W. J. Verkade

Master thesis

April, 2018

THESIS SUBMITTED TO THE

DELFT INSTITUTE OF APPLIED MATHEMATICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

IN

APPLIED MATHEMATICS

GRADUATION COMMITTEE

Dr. P. Cirillo (Delft University of Technology)

Prof. dr. ir. G. Jongbloed (Delft University of Technology)

R. Waaijer (Deloitte)

Abstract

Managing credit risk is a vital part of financial institutions. While the research into credit risk models is extensive, transaction data is a relatively untapped data source in these models. We investigate the explanatory value of transaction data for the Bank by developing default classification models for their small medium enterprises (SME) portfolio. We develop measures that summarize the transaction behaviour on a client level for different time windows. Variables that are included into traditional models are positive income shocks, balance returns, zero transactions (indicating rejected direct debits), and relative cash expenditure. By combining these variables with client characteristics and loan behaviour information, we develop a hierarchical logistic regression model which has a good overall classification performance, reflected by an area under curve (AUC) of 0.850. Tolerating 2 out of 3 false warnings, the model identifies more than 50% of the defaults on average. We investigate relational classification methods, which classify clients according to similarity in terms of their transaction behaviour. The relational neighbour classifier achieves an AUC of 0.768, using similarity between to clients that are determined according to a flexible weight function of the number of shared entities. By combining this approach with the aggregated transaction variables, we develop a model which is solely based on transaction data. The strong performance of this model is reflected by an AUC of 0.804, illustrating the effectiveness of transaction data in default classification.

keywords: credit scoring, monitoring, transaction data, default classification, relational neighbour, hierarchical logistic regression

Contents

Introduction

1

1.1 Motivation

In this research we investigate the explanatory power of transaction data on the credit risk of small and medium-sized enterprises (SME) clients of a medium sized Dutch Bank (the Bank). Currently, in the loan application process at the Bank, client characteristics are used in order to decide whether an applicant is considered creditworthy. If a loan is granted to a client, the performance of this loan is monitored. This is done in a qualitative fashion by for example client visits, and in a quantitative manner by using an internal credit risk model. This model provides monthly updates of the well-known credit metrics such as Probability of Default (PD), Exposure at Default (EAD), Loss Given Default (LGD), and Expected Loss (EL). By closely monitoring these metrics, the Bank is able to keep track of the performance of their loans and take preventive measures if needed. In this research, we focus on default classification for monitoring purposes.

We investigate monitoring, rather than the acceptance decision of the loan. If a new client applies for loan, the Bank has little information available about the applicant's financial situation or payment behaviour. Transaction data could fill this gap and provide a better informed decision. However, for this decision, the transaction data of a client before their loan application is needed. In contrast, a research on monitoring requires the transaction data after a loan has been accepted. The former data is limited in comparison to the latter and insufficient to construct a proper model. It must be noted however that if this research shows the benefits of using transaction in monitoring, it immediately serves as a preliminary research for including transaction data in the acceptance decision.

Current credit risk models of the Bank use a combination of company characteristics, credit behaviour, and external ratings. Company characteristics consist of the client's industry, region, and size, yearly revenue, etc. This information is acquired through annual reports. Loan payment behaviour is gathered by the Bank itself and essentially contains the client's behaviour with respect to their existing loan(s). For example, late payments and an amount of overdraft lower the creditworthiness of a client. The external rating is a credit assessment of the client, which is provided by an external rating agency. These ratings are essential to many credit risk models and widely accepted in the financial sector.

Transaction data is a relatively untapped data source within the Bank. The data contains information of the financial behaviour of the client, which means it is potentially useful in default classification. For every client with an active bank account, all incoming and outgoing transactions are available on a (nearly) live basis. This immediate inflow of information means that financial distress can potentially be spotted earlier. Additionally, by including more financial knowledge of the client into existing models, classification accuracy might be improved. As a result, preventive measures can be taken more effectively, and thus more defaults (and therefore credit losses) can be averted. Furthermore, the credit risk buffer of the Bank can also be lowered. Hence, monitoring improvements by means of transaction data would be in favour of both the client as well as the Bank.

We formulate the following research question:

- *In what way can transaction data effectively be used for default classification models?*

We attempt to answer this question by means of the following sub questions:

- *How can we incorporate transaction data into traditional modelling techniques?*
- *Which novel default classification techniques can be applied to the transaction data?*
- *Can we develop an effective default classification model that is solely based on transaction data?*

This paper is organised as follows: In chapter 2, we discuss the data that is provided by the Bank. Chapter 3 describes how we translate this data into mathematical measures. Modelling techniques that incorporate these measures as modelling variables are explained in chapter 4, and the results of these models are evaluated. In chapter 5, a relational classification approach is explained and applied to the transaction data. Naturally, the predictive performance of this model is also analysed. Chapter 6 combines the traditional approach of chapter 4 with the relational approach of chapter 5, and investigate the potential of a combined modelling approach. Furthermore, we assess the performance of a model which is solely based on transaction data. Finally, in chapter 7, summarize our findings and make suggestions for further research.

1.2 Literature

As managing credit risk is a vital part of many financial institutions, there is a big interest in developing quantitative techniques for this purpose. Consequently, in literature there is a growing number of researches that investigate credit scoring approaches. These approaches aim to accurately make a distinction between good and bad borrowers, which is done by identifying and analysing borrower's characteristics. This distinction is useful for the loan application as well as monitoring purposes on both consumer and corporate level. In Louzada et al. (2016), the number of published credit scoring researches in the past two decades is illustrated over time. This shows a strong growing interest in the topic, which is attributed to the Basel II (credit risk) regulations introduced in 2004. As a result of this growing interest, as well as improvements in computational power, the number of techniques has grown significantly. Some techniques that are often encountered in literature (Brown and Mues, 2012,Louzada et al., 2016), are: logistic regression (LR), linear discriminant analysis (LDA), support vector machines (SVM), and neural networks (NN).

The LDA and LR are comparable techniques that are both based on regression. The independent variables are assumed to linearly relate to the dependent variable. The benefit of these approaches is that they are simple to implement and provide interpretable results (Gurný and Gurný, 2013). The main difference is that LDA assumes normality of the independent variables, which is why LR is preferred if this assumption is unreasonable (Press and Wilson, 1978). The logistic regression can be extended to an hierarchical logistic regression (HLR), also known as mixed-effect or multi-level modelling. The HLR allows for modelling dependent observations within the portfolio and is applied in McNeil and Wendin (2003). The HLR requires more modelling effort than the LR but is useful for handling group specific effects in the portfolio.

While the LR is considered to be the industry's practice (Lessmann et al., 2015), the technique is currently used as a benchmark for more sophisticated approaches. As a result of increased computing power and data size, machine learning techniques such as SVM and NN have become popular in the field of classification. These techniques are able to capture complex, non-linear patterns in the data and have proven to be accurate in default classification (West, 2000, Huang et al., 2007). Despite this success, these approaches have severe drawback for practical purposes. The algorithms are considered 'black box', meaning that insight of the algorithm's inner workings is lacking (Martens et al., 2008). This is an important aspect for financial institutions, because regulations require them to use explainable and transparent models.

In consumer credit risk, it is common practice to use socio-demographic variables (such as: age, income) and previous loan behaviour (Crook et al., 2007). Corporate credit assessment is mainly based on company characteristics, financial ratios (such as loan-to-balance), and loan behaviour (Wilson et al., 2000). The use of transactional data is still uncommon in credit risk literature, which is likely due to the private nature of this data. However, the development of the FinTech[1] industry and the introduction of Payment Service Directive II (PSDII) has resulted in a growing interest in transaction data (Romānova et al., 2018). There are some examples in which transaction data is analysed for a credit scoring purpose. In Khandani et al. (2010), individual transactions of credit card issuers are used for predicting consumer creditworthiness by classifying each transaction into a certain category. On a corporate level, checking account information and credit usage are used a modelling variables in Norden and Weber (2009). While the use of transaction data is novel in these researches, the applied techniques are still (relatively) traditional.

Recently, there have been several studies in which the transaction data is used in a fine-grained, non-aggregated manner. In these researches, a relational network is constructed in which clients are connected if they have transacted with each other or the same entity (Martens et al., 2016). Highly connected clients in the network are classified to the same class by means of various relational classifiers (Macskassy and Provost, 2007). In Tobback and Martens (2017), relational classification through transactional data shows promising results on a large consumer portfolio, whereas relationships between managers is an informative bankruptcy predictor on an SME level (Toback et al., 2017). These approaches are inspired by social networks, in which friendships or shared hobbies can indicate similarity in properties of interest (Cnudde et al., 2015). To our best knowledge, there are no studies which apply relational classification based on transactional data to an SME portfolio.

[1] Spotcap is an example of a FinTech company that issues loans based on transaction data.

Data

2

In this chapter, we describe and analyse the data that is used in our research. The source of this data is extensive, containing all information about the SME portfolio of the Bank. As not all information is representative for our monitoring purpose, we set requirements that limit number of clients in scope. First, we introduce these requirements. The resulting data in scope is divided in three datasets which are described in detail. An exploratory analysis is performed, which provides insight in the data. The main contribution of this research lies in researching the explanatory value of the transaction data. Therefore, we provide a separate analysis of the transaction data.

The content of this chapter is confidential.

Covariates 3

In chapter two, we have discussed the available data. We illustrated the transaction as well as the loan behaviour of clients. The next step is to translate this information into modelling variables. In section 3.1, we construct measures that summarize the data in certain time frames. Section 3.2 describes which measures are considered as potential modelling variables and specifies the time frames. We summarize all developed modelling variables in section 3.3, and briefly assess their significance with respect to the default event.

3.1 Mathematical measures

3.1.1 Transactional measures

Before we construct the measures, we introduce some notation. Let $t = 1, 2, \dots, T$ denote the time in days. The set $\mathcal{X}_j$ contains all transaction amounts $x_t^i(j)$ for client j. In this formulation $i \in \mathbb{N}$ represents the i-th transaction and t indicates the day the transaction occurred. The following three subsets are defined, which distinguish positive, negative, and zero transactions.

$$X_t^+(j) = \{x_t^i(j) \in \mathcal{X}_j | x_t^i(j) > 0\} \tag{3.1}$$

$$X_t^-(j) = \{x_t^i(j) \in \mathcal{X}_j | x_t^i(j) < 0\} \tag{3.2}$$

$$X_t^0(j) = \{x_t^i(j) \in \mathcal{X}_j | x_t^i(j) = 0\} \tag{3.3}$$

The size of each set corresponds to the number of positive, negative, and zero transactions per day. We denote these sizes as $c_t(j)$ and define them in 3.4.

$$c_t^+(j) = |X_t^+(j)|, \quad c_t^-(j) = |X_t^-(j)|, \quad c_t^0(j) = |X_t^0(j)| \tag{3.4}$$

A distinction is made between the positive and negative transaction volume per day. The aggregated daily volumes are denoted by $Y_t^+(j)$ and $Y_t^-(j)$ for positive and negative volumes respectively, and defined as follows:

$$Y_t^+(j) = \sum_{x \in X_t^+(j) x} \tag{3.5}$$

$$Y_t^-(j) = \sum_{x \in X_t^-(j) x} \tag{3.6}$$

Using the daily transaction volume, we define the balance at time t as $B_t(j)$, in which $B_0(j)$ is the starting balance of the client j.

$$B_t(j) = B_0(j) + Y_t^+(j) + Y_t^-(j) \tag{3.7}$$

Based on these definitions, we construct several covariates that capture the transaction information in varying time windows. For notational simplicity, we drop the j, and keep in mind that we focus on the transactions on a client level.

Log balance return

The balance of a client is easily interpretable from a risk perspective. Generally, a decreasing balance means an increasing risk, and inversely. Note that the time window in such trends is important. A client can have an increasing balance in the past month, but a strongly decreasing balance in the past three months.

We consider relative changes in the balance rather than absolute, i.e. returns. For mathematical convenience, it is common practice in finance to consider log returns rather than discrete returns. Equation 3.8 defines the log returns r_t of the balance.

$$r_t = \log\left(\frac{B_t}{B_{t-1}}\right) \tag{3.8}$$

A problem arises in case a clients' account allows for negative amounts. If either B_t or B_{t-1} is negative, the log of a fraction between the two is negative and therefore does not exist. To deal with this, we introduce an absolute shift value a. We set the shift value equal to 1.25 times the credit limit. Consequently, the balance is always larger than zero. The absolute shift method is crude, however does not have any severe consequences in this research. More sophisticated and refined approaches to deal with returns of negative prices are discussed by Fries et al. (2017).

Thus, we use the log returns of the shifted balance r_t^* as defined in equation 3.9.

$$r_t^* = \log\left(\frac{B_t + a}{B_{t-1} + a}\right) \tag{3.9}$$

in which $a = 1.25 \cdot |\min_t B_t|$.

To capture trends rather than the point-in-time returns, we calculate the τ-day average in equation 3.10.

$$R_\tau = \frac{1}{\tau} \sum_{t=T-\tau}^{T} r_t^* \tag{3.10}$$

in which τ is the size, and T the end-day of the time window.

Volatility of balance returns

The volatility of the balance returns provides an indication of the stability of the balance. We determine the volatility by calculating the standard deviation of of the log (shifted) balance returns. Again, we introduce τ to measure the log return volatility of a certain time period.

$$\sigma_{R_\tau} = \sqrt{\frac{1}{\tau} \sum_{t=T-\tau}^{T} (R_t - R_\tau)^2} \tag{3.11}$$

Frequency volatility

Stability in the number of transactions is an interesting property of the transaction behaviour. Hence, we also examine the volatility of the number of transactions. We distinguish between positive and negative transactions.

We define the average number of transactions in a given time window:

$$K_\tau^+ = \frac{1}{\tau} \sum_{t=T-\tau}^{T} c_t^+ \tag{3.12}$$

$$K_\tau^- = \frac{1}{\tau} \sum_{t=T-\tau}^{T} c_t^- \tag{3.13}$$

Using these averages, we define the volatility as the standard deviation of the number of transactions. The formulas are as follows:

$$\sigma_{K_\tau^+} = \sqrt{\frac{1}{\tau} \sum_{t=T-\tau}^{T} \left(c_t^+ - K_\tau^+\right)^2} \tag{3.14}$$

$$\sigma_{K_\tau^-} = \sqrt{\frac{1}{\tau} \sum_{t=T-\tau}^{T} \left(c_t^- - K_\tau^-\right)^2} \tag{3.15}$$

Return shocks

Shocks are sudden sharp spikes or drops in the balance returns. These extreme events can have a big impact on the financial situation of a client. We measure both positive, as well as negative shocks. We define:

$$E_\tau^+ = \frac{1}{\tau} \sum_{t=T-\tau}^{T} \mathbf{1}\left\{r_t - R_\tau > \alpha\right\} \tag{3.16}$$

$$E_\tau^- = \frac{1}{\tau} \sum_{t=T-\tau}^{T} \mathbf{1}\left\{r_t - R_\tau < \alpha\right\} \tag{3.17}$$

These variables denote the average number of shocks in the time window $[\tau, T]$. The boundary value α gives the threshold a return needs to surpass in order to be marked as an extreme.

Zero transactions

The occurrence of a zero transaction is an interesting event. We measure the total number of zero transactions in the time window $[\tau, T]$ as follows:

$$K_\tau^0 = \frac{1}{\tau} \sum_{T-\tau}^{T} c_t^0 \tag{3.18}$$

Rather than considering the absolute value, we examine the number of zero transactions relative to the number of total transactions in the period. The formula is given in equation 3.19.

$$Z_\tau = \frac{K_\tau^0}{(K_\tau^0 + K_\tau^+ + K_\tau^-)} \tag{3.19}$$

Categories

The relative frequency and volume of transactions of a certain category is summarized for different time windows. Let us define $Y_t^+(c)$ and $Y_t^-(c)$ as the incoming and outgoing volume of transactions of category c at day t. We construct the following measures for a category c.

$$J_\tau^+(c) = \frac{1}{\tau} \sum_{t=T-\tau}^{T} \frac{Y_t^+(c)}{Y_t^+} \tag{3.20}$$

$$J_\tau^-(c) = \frac{1}{\tau} \sum_{t=T-\tau}^{T} \frac{Y_t^-(c)}{Y_t^-} \tag{3.21}$$

Similarly, we define $c_t^+(c)$ and $c_t^-(c)$ as the incoming and outgoing number of transactions. The frequency measures per category are given by:

$$H_\tau^+(c) = \frac{1}{\tau} \sum_{t=T-\tau}^{T} \frac{c_t^+(c)}{c_t^+} \tag{3.22}$$

$$H_\tau^-(c) = \frac{1}{\tau} \sum_{t=T-\tau}^{T} \frac{c_t^-(c)}{c_t^-} \tag{3.23}$$

Using these measures, we construct modelling variables of the transactions of a client on a category level.

3.1.2 Credit measures

The formulas in the previous section consolidate the transactional data into mathematical measures. In this section, we achieve the same for credit data. To develop the measures, we assume that a client has a credit limit M_j^*. Note that this value is negative, as it represents the most negative value a client's account is allowed to reach. Again, we omit the j in the formulas, and remember that we focus on one particular client. These measures are only valid in case the client has a dynamic credit product.

Credit use

The credit limit indicates the maximum amount a client can withdraw. By comparing this to the withdrawn credit, we determine the percentage of credit that is used by the client at any point in time. Clearly, a client with a positive balance has a credit use of zero. The credit use at time t is given by:

$$m_t = \begin{cases} 0 & \text{if } B_t \geq 0 \\ B_t/M^* & \text{if } B_t < 0 \end{cases} \tag{3.24}$$

We focus on time windows rather than the point-in-time values of m_t. Hence, we define,

$$M_\tau = \frac{1}{\tau} \sum_{t=T-\tau}^{T} m_t \tag{3.25}$$

The value for M_τ represents the average limit use in the period $[\tau, T]$.

Limit exceedances

It can occur that the maximum credit limit is exceeded by a client. These limit exceedances are important events for the credit issuer because it indicates that the current credit agreement is not sustained. We define these exceedances as follows:

$$O_\tau = \frac{1}{\tau} \sum_{t=T-\tau}^{T} \mathbf{1}\{m_t > 1\} \tag{3.26}$$

The variable O_τ corresponds to the average number of limit exceedances in the period $[\tau, T]$.

3.2 Covariate choices

To further specify the introduced measures into modelling variables, we make two decisions. First, we discuss which categories are included as potential candidates. Then, we select an appropriate time frame for each of the measures.

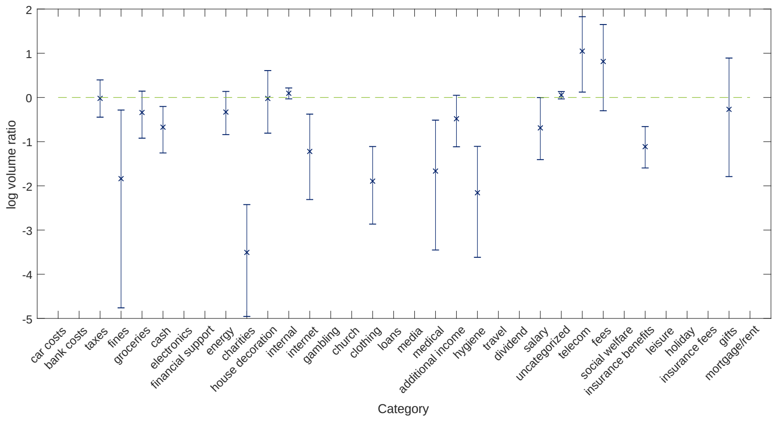


Fig. 3.1.: Log ratio of incoming transaction volume between default and non-default.

3.2.1 Selection categories

The transaction categories as presented in chapter 2 are further analysed. We investigate whether defaulting clients have a significant different relative volume in certain categories. If so, these categories are considered as risky or safe from a default perspective, and modelling variables are constructed from them. We focus on transaction volume rather than frequency. While the two are similar, relative transaction volume contains more interesting information in our opinion.

We distinguish the portfolio in a default and a non-default group. The average relative transaction volume of both groups is calculated for each category. By dividing these averages, we find a ratio which indicates the riskiness of the categories. For illustrative purpose, we take the log of this ratio. A ratio higher than zero indicates that the relative volume in a category is larger among defaulting clients, a ratio lower than zero means the category has higher volume among non-defaulting clients.

Figures 3.1 and 3.2 show the resulting log-ratios for every category for incoming and outgoing transactions respectively. In addition to the observed ratio, a 95% confidence interval is also constructed via bootstrap sampling. If zero falls outside a confidence bound, we conclude that the corresponding category is regarded as either 'safe' (lower than zero) or 'risky' (higher than zero). All other categories are considered neutral.

For every variable that we include into the model, it is important that it is intuitive explainable for the risk expert. Therefore, we introduce categories as potential covariates that have both quantitative as well as intuitive support. Together with experts from the Bank, it is decided to include the following categories as potential modelling covariates: cash, mortgage, bank costs, gambling, and charities.

We consider the relative outgoing volume for these categories. Figure 3.2 shows that the log-ratio of the cash, mortgage, and bank costs category are significantly above zero. For these categories, the purpose of the transaction is also clear. Cash transactions are ATM machine withdrawals, mortgage reflect property loan payments, and bank costs are costs associated with any bank. The latter can consist of interest payments, additional loans,

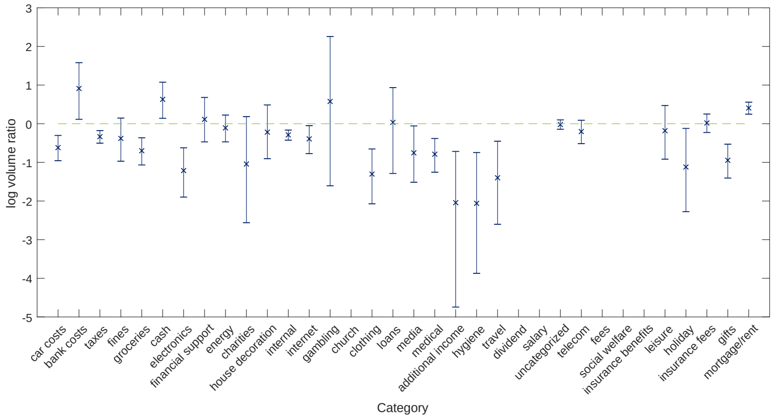


Fig. 3.2.: Log ratio of outgoing transaction volume between default and non-default.

or fees, which explains the observed quantitative risk. The other two categories are not significantly different from zero. However, both have a very intuitive association. The gambling category is associated with risk, regarding the opportunistic nature of this activity. Expenses in charities is regarded as reliable, as it indicates that there is both room and incentive to provide financial assistance.

3.2.2 Selection time windows

As stated in **??**, T_j denotes the last date of relevant information for each client j. Based on T_j, we determine three different time windows, namely: short, medium, and long term. Together with experts from the Bank, we determine time windows to correspond to three, six, and twelve months respectively. We define these windows as follows:

$$\tau_3 = [T_j - 91, T_j] \tag{3.27}$$

$$\tau_6 = [T_j - 182, T_j] \tag{3.28}$$

$$\tau_{12} = [T_j - 365, T_j] \tag{3.29}$$

For each of the constructed measure, we choose one of these time windows. Table 3.1 shows these choices. Shocks, exceedances, and zero transactions are distinctive events, that possibly have an immediate effect on the financial situation. Therefore, we choose to analyse these variables within the short term. The balance returns and all category measures are considered within the long period. This choice is made to include important yearly transactions. To find a balance between a too stable and an overly sensitive volatility, the volatility measures are chosen to be measured on a medium term.

Last, we define the threshold for which a balance return is seen as an extreme event. We choose $\alpha = 2 \cdot \sigma_{R_6}$, which means that a balance return is labelled as an extreme if it exceeds the short term mean by twice the medium term standard deviation.

3.3 Analysis covariates

The purpose of constructing the covariates is to develop a model to predict defaults. Due to the different nature of fixed and dynamic credit products, we aim to develop two separate models. Hence, the portfolio is split into two groups, clients with a dynamic credit (DC), and clients with a fixed credit (FC) product. To develop these models the DC and FC set are divided into a training and a test part according to the ratio 2:1. The training part is used for model building, whereas the test part is used to evaluate the final model. Note that the final model depends on the partitioning between training and test data. We analyse this effect in chapter 4.

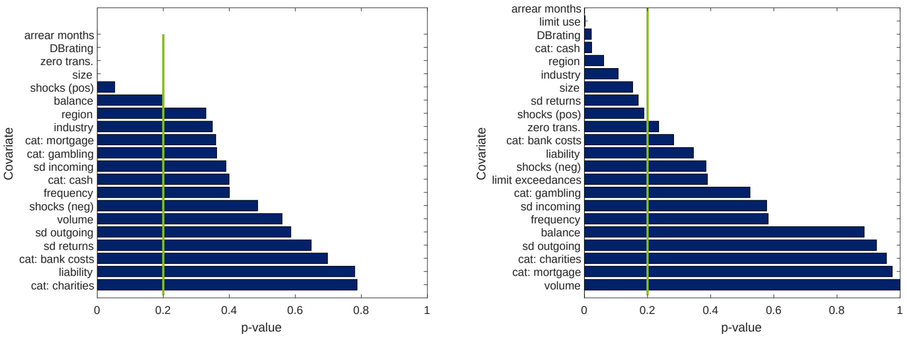
The model building process consists of selecting independent variables that are important in terms of explanatory value. A standard approach to do this is stepwise regression. The stepwise regression consists of selecting predictors in a iterative manner, based on model performance criteria. However, stepwise regression is often criticized for being susceptible to overfitting (Tibshirani, 1996). This is caused by the greedy strategy of the approach, which can result in locally optimal predictors. An other approach is to use regularization techniques such as Ridge regression (L_1), LASSO (least absolute shrinkage and selection operator) (L_2), or the elastic net (Hoerl and Kennard, 2000). The latter is a combination of the former two techniques. These techniques are based on adding penalizing terms to the likelihood function. A drawback of these penalizing terms is that the resulting coefficient values lose interpretability.

To have full control of the model-building process, we prefer to use a manual step-by-step approach as presented Hosmer and Lemeshow (2005). The first step of this approach is to perform a univariable analysis on all possible variables. In this section, we analyse all discussed variables in such a manner. This includes the client characteristics (as discussed in chapter 2), and the credit and transaction behaviour variables as developed in the previous section. A full overview of variables is given in table 3.2. We only include linear predictors into our model and neglect any higher order, or interaction terms. These terms are generally difficult to interpret and significantly complicate the model. We have many variables available and there are no intuitive non-linear relations present. Thus, we assume that a model without higher order terms should be sufficient.

With the univariable analysis, we assess whether a covariate has a direct effect on the independent variable. For categorical variables, we use Pearson's chi-squared test for independence as discussed in appendix B.1. We test the null hypothesis that the different

Short	**Medium**	**Long**
zeros	sd returns	balance
positive shocks	sd incoming	cat: cash
negative shocks	sd outgoing	cat: mortgage
limit exceedances		cat: bank costs
limit use		cat: gambling
		cat: charities

Tab. 3.1.: Time window choices for the mathematical measures.



(a) p-values of the univariable analysis for covariates of the fixed credit model.

(b) p-values of the univariable analysis for covariates of the dynamic credit model.

Fig. 3.3.: Univariable analysis of the covariates for the credit models. The blue bars represent the calculated p-values, whereas the green line illustrates the cut-off point of 0.20.

levels of the categorical covariate follow the same distribution with respect to the dependent variable. Rejection of the null hypothesis suggests that the covariate is a candidate for the final regression model.

For continuous variables, there is no direct statistical test to assess the effect on the dependent variable. Instead, we fit a univariable logistic regression with the continuous covariate as the only independent variable. With the resulting estimated coefficient and standard deviation, we use Wald's test to evaluate whether the coefficient is significant. If so, the continuous covariate is a potential candidate to include in our model. The use of a logistic regression is motivated in the chapter 4.

The decision to include covariates in the next step of our covariate selection process is based on the p-values of the analyses. Figure 3.3 presents these, and further statistics can be found in the appendix C.1. Mickey and Greenland (1989) have shown that using a traditional significance level of 0.05 as a cut-off point often disregards variables known to be important. Hence, we follow the suggestion of Hosmer and Lemeshow (2005) and consider variables with a p-value lower than 0.20 as potential candidates.

Figure 3.3a shows 6 variables that are significant on a 0.20 level. These variables are potential candidates for the fixed credit model. The dynamic credit model has 9 variables that are significantly smaller than 0.20, which is illustrated in figure 3.3b. The two models have 4 variables in common, namely: arrear months, size, DB rating, and positive shocks. Besides traditional covariates such as the client characteristics and arrear months, both models also show transactional variables at a significant level. This is promising for our research.

This analysis provides the first step into determining important variables for our models. In the next chapter, we will take the next steps of the model building strategy.

Covariate	Description	Levels
region	location of the client	region 1, region 2, region 3
industry	branche in which the client operates	industry 1, industry 2, industry 3
size	size of the company, based on the yearly revenue	small, medium, large, extra large
DB rating	credit rating by the external bureau Dun & Bradstreet	unrated, minimal, low, above average, and significant risk
liability	indicator of personal liability within the legal form	liable, non-liable
volume	yearly transaction volume	continuous
frequency	yearly transaction frequency	continuous
balance (1 year)	average balance returns	continuous
zeros (3 month)	percentage of zero transactions	continuous
sd returns	standard deviation of balance returns	continuous
sd incoming	standard deviation of number of incoming transactions	continuous
sd outgoing	standard deviation of number of outgoing transactions	continuous
arrear months	months a client has been in arrears in the past 24 months	continuous
shocks (pos)	positive income shocks	continuous
shocks (neg)	negative income shocks	continuous
cat: cash	relative expenditure in cash	continuous
cat: mortgage	relative expenditure in mortgage/rent	continuous
cat: bank costs	relative expenditure in banking costs	continuous
cat: gambling	relative expenditure in gambling	continuous
cat: charities	relative expenditure in charities	continuous
limit exceedances	exceedances of credit limit	continuous
limit use	percentage of credit	continuous

Tab. 3.2.: Summary of all variables that can be included in the models. A short description is given for each variables, and the possible levels of categorical variables are described.

Traditional Modelling 4

In the previous chapter we discussed the covariates that summarize the available information of the clients. We construct models for the classification of defaults with these covariates. These models are traditional in the sense that the dependent variable is predicted via multiple independent variables. In this chapter, we describe the model building process and show the results of the constructed models. In section 4.1, we develop a logistic regression model, and asses its performance according to several introduced evaluation measures. We distinguish between clients with a different credit type and create separate models for these groups. Section 4.2 extends the logistic regression to a hierarchical version which is applied to the full portfolio.

4.1 Logistic regression

In credit scoring, the logistic regression is considered the industry's practice. The logistic regression (and the similar probit regression) belong to the family of generalized linear models (GLM). Fox (2008) describes the GLM as consisting of three main components:

- A random component, specifying the conditional distribution of the response variable.
- A linear predictor, i.e. a linear function of the covariates.
- A smooth and invertible link function, which transforms the linear prediction to the expectation of the response variable.

GLMs specify linear regression models that allow modelling for response variables that are not necessarily Gaussian. Rather than that, the response variable is assumed to be a member of the exponential family. The clients in the portfolio are either in default or non-default, which can be described by a binomial distribution.

To transform the linear predictor to the response variable a link function is used. The link functions corresponding to a binomial family in the GLM are: the logit, probit, log-log, and complementary log-log. We focus on the logit, which is the most commonly used due to some practical advantages. One such an advantage is that the results of the linear predictor directly correspond to the log-odds of a default event versus a non-default event.

4.1.1 Formulation

We formally describe the logistic regression and follow the notation of Hastie et al. (2001) and Fox (2008). The objective is to model the distribution of Y conditional on a set of covariates $X_1, \ldots, X_M$. In other words, we want to find $\mathbb{P}(Y|X_1, \ldots, X_M)$. To achieve this, we specify the linear predictor η, and a smooth, invertible link function g that maps η to the expected value of Y.

The linear predictor η is a linear function of the covariates $X_1, \ldots, X_M$. These covariates are described in chapter 3 and reflect the (possibly transformed) continuous variables, and dummy-coded versions of the categorical variables. The linear predictor is defined as follows:

$$\eta = \beta_0 + \beta_1 X_1 + \cdots + \beta_M X_M \tag{4.1}$$

$$E(Y) = g^{-1}(\eta) \tag{4.2}$$

The density $\mathbb{P}(Y|X_1, \ldots, X_m)$ represents the class probabilities. For each realization $Y = y$, the outcome is either default or non-default. We view the outcomes as the binomial distribution with probability of success π. Thus, we write:

$$\mathbb{P}(Y = y|\pi) = \pi^y (1 - \pi)^{(1-y)} \tag{4.3}$$

The probability π is restricted to have a value between 0 and 1. The linear predictor η is theoretically defined between minus infinity to infinity and therefore we require a link function that maps these values towards the range $(0, 1)$. This is achieved via the logit link function:

$$\log(\frac{\pi}{1 - \pi}) = \eta \tag{4.4}$$

By taking the inverse of this expression and inserting the linear predictor, we find the following expression for the probability π.

$$\pi = \frac{e^\eta}{1 + e^\eta} \tag{4.5}$$

$$= \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_M X_M}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_M X_M}} \tag{4.6}$$

Altogether, the three components of the logistic regression are summarized in the following equations:

$$\eta = \beta_0 + \beta_1 X_1 + \cdots + \beta_M X_M \tag{4.7}$$

$$\pi = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_M X_M}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_M X_M}} \tag{4.8}$$

$$Y \sim Binomial(1, \pi) \tag{4.9}$$

4.1.2 Parameter estimation

Now we have developed the framework for the logistic regression we have to fit the model, i.e. determine the parameter estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \dots, \hat{\beta}_M)$. The most common way to do this is through the method of maximum likelihood. The available data consists of N observations of the dependent variable y_i, and N observations of the M variables x_{iM}. The $N \times (M+1)$ design matrix $\boldsymbol{X}$ consists of a vector of ones and the vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M$ denoting the observations of the independent variables. The likelihood of our parameters given the data is given by:

$$
\begin{aligned}
Lik(\boldsymbol{\beta}, \boldsymbol{X}) &= \mathbb{P}(Y|\boldsymbol{X}, \boldsymbol{\beta}) && (4.10)\\
&= \prod_{i=1}^{N} \mathbb{P}(y_i|\boldsymbol{X}, \boldsymbol{\beta}) && (4.11)\\
&= \prod_{i=1}^{N} \pi_i^{y_i}(1-\pi_i)^{1-y_i} && (4.12)
\end{aligned}
$$

This yields the following log-likelihood in which $\boldsymbol{x}'_i$ represents the i-th row of the design matrix $\boldsymbol{X}$. Further details on the derivation are provided in the appendix A.1.1.

$$
\begin{aligned}
l(\boldsymbol{\beta}, X) &= \log\left(\prod_{i=1}^{N} \mathbb{P}(y_i|\boldsymbol{X}, \boldsymbol{\beta})\right) && (4.13)\\
&= \sum_i y_i \boldsymbol{x}'_i \boldsymbol{\beta} - \log\left(1 + e^{\boldsymbol{x}'_i \boldsymbol{\beta}}\right) && (4.14)
\end{aligned}
$$

The optimal parameters are found if the log likelihood is maximized. Therefore, we set the derivatives to zero for each parameter β_j. After some manipulations, we find for $j = 0, \dots, M$:

$$
\frac{\mathrm{d}l(\boldsymbol{\beta}, X)}{\mathrm{d}\beta_j} = \sum_i y_i x_{ij} - \sum_i x_{ij} \frac{e^{\boldsymbol{x}'_i \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}'_i \boldsymbol{\beta}}} = 0 \qquad (4.15)
$$

By solving these equations, the parameter estimates $(\hat{\beta}_0, \dots, \hat{\beta}_M)$ are determined. Since there is no closed-form solution, a common technique is to numerically solve them using the Newton Raphson technique. Details of this technique are found in appendix A.1.2.

4.1.3 Goodness of fit tests

We have described a methodology to construct a logistic regression model. After developing such a model, it is important to check whether the model is appropriate. Goodness of fit tests exists to assess the model quality. The test are developed to test for the following misspecifications of the logistic model: missing higher order terms, missing interactions, and a wrong link function.

A well-known statistical test for categorical data is the Pearson chi-squared test (Agresti, 2006). The test is applicable if the data observations can be aggregated into unique cases, in which each case has exactly the same predictor values. For each case, the difference between the observed proportion and the expected proportion of events is calculated. The null hypothesis is that these proportions are similar and under this hypothesis the corresponding test statistic (equation 4.16) follows a chi-squared distribution with $N - M$ degrees of freedom (in which N and M are the number of observations and predictors respectively). An intuition for the degrees of freedom is sketched in the appendix B.1.

$$X^2 = \sum_j \sum_i \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{4.16}$$

in which O_{ij} and E_{ij} are the observed and expected proportions of an event i for the case j.

The test shows good properties if the number of observations per case is sufficient (McCullagh, 1985). However, we are working with continuous predictor values, resulting in one unique case for every observation. Consequently, the test-statistic does not necessarily follow a chi-square distribution under the null hypothesis, and the classic Pearson test can not reliably be applied.

In a simulation study, Evans and Li (2005) show that goodness of fit tests on a logistic model gives varying results. They advice to use multiple test to complement each other. We follow this advice and introduce three tests with which we assess the goodness of fit of our models.

Standardized Pearson

In case there is a unique case per observation the classic Pearson test statistic can be written as in equation 4.17. A derivation is given in the appendix B.2.

$$X^2 = \sum_i \frac{(y_i - \hat{\pi}_i)}{\hat{\pi}_i(1 - \hat{\pi}_i)} \tag{4.17}$$

in which y_i is the dependent variable, and $\hat{\pi}_i$ the predicted probability that $y_i = 1$ for observation i.

In this standardized version, the number of groups is equal to the number of observations. However, the problem with this statistic is that X^2 does not have a chi-squared distribution if the data is ungrouped (i.e. continuous). In Osius and Rojek (1992), it is shown that the statistic has an asymptotic normal distribution with an approximated mean of ν and standard deviation 2ν, where ν indicates the degrees of freedom. By subtracting the mean and dividing by the standard deviation we find a test statistic that follows an asymptotic standard normal distribution.

Hosmer and Lemeshow

Hosmer and Lemeshow (1980) introduce a goodness of fit test for logistic models. This test is based on allocating the observations and corresponding predicted values to G unique groups (ten is often recommended). Specifically, the predicted values are ordered from low to high, and separated into groups of approximately equal size. For each group, the observed and expected proportion of both defaults and non-defaults is calculated in each group. The observed proportions are determined according to the class labels, whereas the expected proportions are calculated using the predicted values.

To determine whether the model is calibrated well, the proportions are compared for each group. Formally, the test statistic is defined as follows:

$$H = \sum_{g=1}^{G} \left(\frac{(O_{1g} - E_{1g})^2}{E_{1g}} + \frac{(O_{0g} - E_{0g})^2}{E_{0g}} \right) \tag{4.18}$$

in which $O_{1g}(E_{1g})$ and $O_{0g}(E_{0g})$ represent the observed (expected) proportions of defaults and non-defaults of group g. Note that this is similar to the classic Pearson test statistic.

With the statistic H, we test the null hypothesis whether the actual and predicted event rates are similar. If H is larger than some threshold, the null hypothesis is rejected and the model should be altered. The threshold value is determined by considering the distribution of the test statistic. It is shown in Hosmer and Lemeshow (1980) that H follows a chi-square distribution with $G - 2$ degrees of freedom under the null hypothesis. A drawback of this approach is pointed out by Homer and Lemeshow themselves. The the results of the test are unstable under the number of considered groups (Hosmer et al., 1997). Consequently, the test statistic may lack power.

Stukel test

Stukel (1988) introduces a generalization of the logistic regression model. This generalization includes two additional variables which allow for deviations from the logistic curve. The additional variables are constructed according to the results of the original model according to equations 4.19 and 4.20. In these equations g_i denotes the i-th row the matrix $\boldsymbol{X}$ multiplied with the predicted estimators, i.e. $g_i = \boldsymbol{x}_i' \cdot \hat{\boldsymbol{\beta}}$.

$$z_i^a = \begin{cases} g_i^2 & \text{if } g_i \geq 0 \\ 0 & \text{if } g_i < 0 \end{cases} \tag{4.19}$$

$$z_i^b = \begin{cases} 0 & \text{if } g_i \geq 0 \\ g_i^2 & \text{if } g_i < 0 \end{cases} \tag{4.20}$$

According to these formulas, the additional variable vectors $\boldsymbol{z}^a$ and $\boldsymbol{z}^b$ are calculated. The variables are added to the logistic regression model and the null hypothesis is tested that both their corresponding coefficients are zero. Failure of the test indicates deviations from the link function.

4.1.4 Predictive power tests

The main reason to develop our model is to correctly classify unseen data. Hence, besides the goodness of fit of the model, we also assess the predictive quality. In a linear model, the predictive quality is often measured by using the coefficient of determination R^2. The estimates for hte logistic regression are determined by ML, and hence the R^2 does not exist. Alternatives have been developed, which are often referred to as pseudo-R^2 measures. A comparison between such measures is provided in Mittblöck and Schemper (1996). We discuss two of these measures, namely the commonly used McFadden (McFadden, 1973) and the more recent Tjur (Tjur, 2009). While the R^2 values do measure predictive quality, the results are not easily interpretable (Hu et al., 2006). It is stated by L. McFadden (1977) that values between 0.2. and 0.4 for these measures represent an excellent fit. In addition to the pseudo-R^2, we also present methods to assess classification performance.

MCfadden

The MacFadden R^2 compares the likelihood of the fitted model, to a model with only an intercept. The formula is given by:

$$R^2_{MF} = 1 - \frac{\log(L_c)}{\log(L_{null})} \tag{4.21}$$

in which L_c is the likelihood of the current fitted model, and L_{null} the likelihood of a fitted model with only an intercept.

Tjur

A recent test measure of predictive quality is introduced by Tjur. The motivation behind this measure is that labels of a certain class should have high estimated probabilities for this class in case of a good fit. If this is the case, the model achieves its goal. Let us assume the binary case in which we have the classes 0 and 1. The R_{Tjur} measure is given by:

$$R_{Tjur} = \overline{\hat{\pi}}_1 - \overline{\hat{\pi}}_1 \tag{4.22}$$

in which $\overline{\hat{\pi}}_c$ denotes the average estimated probability of class c. A nice property is that this measure is intuitive.

Classification measures

The output of our models are predictions of the probability of default $\hat{\boldsymbol{\pi}}$, which are continuous values. Whether a client is classified as default or not, depends on a threshold value T. If $\hat{\pi}_i \leq T$, client i is predicted to belong to the non-default class and if $\hat{\pi}_i > T$, the prediction is that the client defaults. The classification quality can be assessed by comparing the predicted class labels with the actual ones. A client that has been correctly identifies as default is called a true positive (TP), whereas a client that has been classified incorrectly is a false positive

(FP). For different values of T, the true positive rate (TPR) and false positive rate (FPR) are defined as in equations 4.23 and 4.24 respectively.

$$TPR(T) = \frac{TP(T)}{FP(T) + TP(T)} \tag{4.23}$$

$$FPR(T) = \frac{FP(T)}{FP(T) + TP(T)} \tag{4.24}$$

The TPR and FPR can be calculated for different values of T. The TPR is likely to be high if the threshold is set close to 1. However, this is accompanied by a high FPR because it means that almost all instances are labelled as default. Similarly, choosing T close to zero not only results in a low FPR, but also a low TPR. The rate of classification (ROC) curve describes this tradeoff by plotting the FPR and TPR for different values of T. Examples of a ROC curves will be presented later in this chapter.

A measure that is directly associated with the ROC curve is the area under curve characteristic (AUC). Like the name suggests, this value is equal to the area under the ROC curve. It reflects the probability that a randomly chosen default client will be assigned a higher predicted probability than a randomly picked non-default client. In general, a classifier that randomly assigns a client to either default or non-default has an AUC of 0.5. According to the definitions of the Bank we qualify the AUC performances as follows: bad: 0.5-0.6, moderate: 0.6-0.7, sufficient: 0.7-0.8, good: 0.8-0.9, excellent: 0.9-1.0. We remark that these judgements subjective and are domain dependent.

In addition to true and false positives, the classification quality is also measured by the number false negatives (FN). A false negative means that a client is classified as non-default, but actually does default. We evaluate the classification performance according to two additional measures, namely recall and precision. Recall is defined in equation 4.25 and reflects the recognized defaults as a fraction of the total number of default in the data set. Precision is the fraction between correctly classified defaults and the total number of classified defaults, and is defined in equation 4.26.

$$Recall : \frac{TP}{TP + FN} \tag{4.25}$$

$$Precision : \frac{TP}{FP + TP} \tag{4.26}$$

For every classification problem a balance between these two measures is determined by the selected threshold T. In our problem of recognizing defaults, there is a trade-off between false warnings and missing defaults. Practitioners should decide on this trade-off by researching the financial impact of both events.

Variable	Coefficient	Standard error	p-value
Intercept	**-3.072**	**0.238**	**<0.001**
Size M	0.381	0.311	0.221
Size L	**1.325**	**0.405**	**0.001**
Size XL	**1.443**	**0.427**	**<0.001**
DB rating 0	0.296	0.304	0.331
DB rating 1	**-1.889**	**0.760**	**0.013**
DB rating 3	**0.657**	**0.380**	**0.083**
DB rating 4	**1.263**	**0.506**	**0.012**
Arrear months	**0.991**	**0.149**	**<0.001**
Balance returns	**-0.174**	**0.105**	**0.097**
Positive shocks	**-0.334**	**0.141**	**0.017**
Zero trans.	**0.237**	**0.111**	**0.032**

Tab. 4.1.: Statistics for the preliminary and final FC model. The maximum likelihood estimates of the coefficients, the standard errors, and the p-values are given. Significant variables ($p < 0.10$) are shown in bold.

4.1.5 Estimation results

As discussed in chapter 3, a distinction is made between clients with a dynamic credit product (DC) and clients with a fixed credit product (FC). Furthermore, the two data sets has been split between a training and test set. Based on the training sets, we have identified variables that are likely to be significant predictors in the final models for both client groups. The next step in the model building process is to fit a multivariate logistic regression, containing all identified candidates. We examine the coefficients of this fitted model and test for significance using the Wald test (details in appendix B.3). To determine whether a variable is significant we use a (relatively large) significance level of 0.10. Note that, the different groups of categorical covariates are included in the model as dummy variables. The largest group has been chosen as the reference group.

If there are any insignificant variables they are removed from the model and the model is refitted. After refitting the model, it is important to check whether any remaining coefficient has changed considerably. If so, this suggests that the excluded variable provides a needed adjustment for the remaining variables, and is therefore essential. Following the approach of Hosmer and Lemeshow (2005), a considerable change is defined as a coefficient change of more than 20%. Furthermore, a likelihood ratio test (details in appendix B.4) is performed to verify whether the full model is not significantly better than the reduced model.

Table 4.1 presents the model statistics of the preliminary FC model. The p-values indicate that there are two non-significant variables, namely: size M and DB rating 0. These variables can only be excluded from the model together with all other levels of the corresponding categorical covariates, which are significant. Hence, we only exclude these variables if the likelihood ratio test indicates that the removal does not significantly yield a worse model in terms of likelihood. The test statistic for removal of DB rating is $LR = 23.02$, which with four degrees of freedom, yields a p-value of < 0.001. Exclusion of size yields a test statistic of $LR = 16.80$, which with three degrees of freedom, also results in a p-value of < 0.001. From these results, we conclude that the full model performs significantly better than the reduced model, and thus no variables are removed. If one would only allow significant variables

Variable	Coefficient	Standard error	p-value
Intercept	**-3.086**	**0.381**	**<0.001**
Size M	0.409	0.388	0.291
Size L	-0.888	1.056	0.401
Size XL	-1.484	0.915	0.105
Industry 1	**-1.997**	**1.003**	**0.047**
Industry 2	-0.437	0.339	0.197
region 1	-0.536	0.448	0.200
region 3	**-1.060**	**0.389**	**0.007**
DB-rating 0	**0.856**	**0.413**	**0.043**
DB-rating 1	-0.409	0.774	0.470
DB-rating 3	-0.445	0.542	0.442
DB-rating 4	0.672	0.575	0.274
Arrear months	**0.341**	**0.110**	**0.002**
Positive shocks	**-0.442**	**0.192**	**0.017**
Return volatility	-0.152	0.299	0.611
Limit use	**1.174**	**0.158**	**<0.001**
Category: cash	**0.271**	**0.098**	**0.005**

Tab. 4.2.: Coefficient statistics for the preliminary DC model. The maximum likelihood estimates of the coefficients, the standard errors, and the p-values are given. Significant variables ($p < 0.10$) are shown in bold.

in the model, insignificant groups can be merged with other, similar groups. However, this would decrease interpretability.

The same strategy is used for the DC model. First, the return volatility has the highest p-value, namely 0.611. The likelihood ratio test yields $LR = 0.286$ and $p = 0.593$. Hence, the variable is removed from the model. Second, none of the size groups are significant at a 0.10 level. The likelihood ratio test gives $LR = 6.142$ corresponding to $p = 0.105$, and thus size is excluded from the model. Next, the DB-rating is only significant for DB rating 0, while the other groups have high p-values. Exclusion from the model is supported by the statistics $LR = 6.760$ and $p = 0.149$. The region variable has one significant and one insignificant group. Test results are $LR = 6.863$ and $p = 0.032$, which means we keep the variable in the model. Last, removal of the industry is supported by the test results $LR = 5.647$ and $p = 0.059$. Note that during this iterative process the model is refitted after each removal.

Table 4.3 shows the coefficient values after removing the several variables. The largest change is 14% for the coefficient of arrear months. This does not exceed our criterion of 20% and hence no further alterations need to be made. The next step is to review the coefficient values of the models and decide whether their effect is intuitive according to experts. If not, the variable is removed from the model. The coefficients in tables 4.1 and 4.3 are in line with experts' expectation and thus none of the variables are removed from the model.

Last, we assess whether there is any collinearity present in these final models. If there is collinearity between two covariates in the model, small data changes can drastically influence the coefficient estimates. The correlation between variables is low and does not exceed 0.4. Hence, we conclude that there is no collinearity between the variables.

Variable	Coefficient	Standard error	p-value
Intercept	**-3.263**	**0.292**	**<0.001**
region 1	-0.669	0.407	0.100
region 3	**-0.914**	**0.377**	**0.015**
Arrear months	**0.293**	**0.099**	**0.003**
Positive shocks	**-0.452**	**0.181**	**0.012**
Limit use	**1.160**	**0.214**	**<0.001**
Category: cash	**0.265**	**0.102**	**0.009**

Tab. 4.3.: Coefficient statistics for the final DC model. The maximum likelihood estimates of the coefficients, the standard errors, and the p-values are given. Significant variables ($p < 0.10$) are shown in bold.

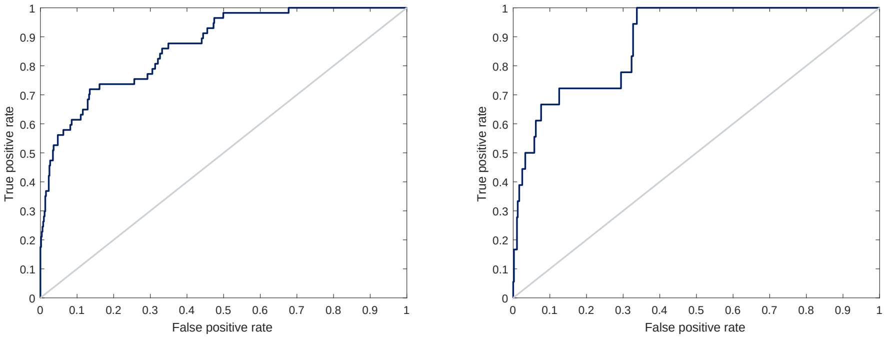
Measure	Fixed credit	Dynamic credit
MacFadden	0.265	0.233
Tjur	0.258	0.174
AUC	0.868	0.887
Precision	0.552	0.240
Recall	0.561	0.667

Tab. 4.4.: Predictive performance of the models.

Now that we have developed two final models, we test their goodness of fit and assess their predictive power. The goodness of fit tests assesses the adequacy of the models. All tests are passed at a critical level of 0.05, which suggest that there are no reasons to revisit any of the model choices we have made. Table 4.4 presents the Macfadden and Tjur for both models. The FC model has a value of 0.265 and 0.258 for these measures respectively. Both values are between 0.2 and 0.4, which represents an excellent fit. Both pseudo-R^2 measures for the DC model are lower, meaning the DC model fits the corresponding clients worse than the LP model. The value for the Tjur measure is 0.174, meaning that the average difference between PD predictions of defaults and non-defaults is 17.4 percent. The MacFadden of 0.233 still represents an excellent fit.

The previous statistics are all based on the training data. A measure that indicates the performance of classifying the test data is the ROC curve. Figure 4.1a and 4.1b show the ROC curves for the FC and DC model respectively. The curves present the false positive rate versus the true positive rate of the models for different threshold values T. An increase along the vertical axis means that more defaults are correctly classified, whereas an increase with respect to the horizontal axis represents a rise in false positives. A model that perfectly distinguishes defaults from non-defaults would have a curve with straight lines connecting $(0, 0)$ to $(0, 1)$ and $(0, 1)$ to $(1, 1)$. The grey lines of 45 degrees in the figures represent the ROC of a random classification model.

The AUC's corresponding to the curves of the FC and DC model are 0.868 and 0.887, and indicate a high overall classification performance. One can interpret these values as the probability that a uniformly drawn random default is assigned a higher PD than a uniformly drawn random non-default. As such, the performance of the models in distinguishing defaults is good. Note that a random model and a perfect model have an AUC of 0.5 and 1 respectively.



(a) Rate of classification curve for the fixed credit model. The corresponding AUC is 0.868.

(b) Rate of classification curve for the dynamic credit model. The corresponding AUC is 0.851.

Fig. 4.1.: Rate of classification curves for the fixed and dynamic credit model.

While the AUC gives a good indication of the overall quality, it does not fully capture the performance of the models for a monitoring purpose. To illustrate this, let us consider point on the curve in figure 4.1a where the FPR is approximately 35% and the TPR is around 88%. To achieve this performance, the number of warnings signals corresponds to 40% of the whole test set. It requires a lot of effort to analyse all warning signs. Therefore, it is useful to investigate the recall and precision of the models.

To determine the recall and precision, we must select a threshold T. We choose T such that the number of warnings is 10% of the size of the test set. Table 4.4 presents the precision and recall of this approach. The recall of the FC model is 56.1%, which indicates the percentage of defaults are detected by the warnings system. Furthermore, the precision is approximately 50%, which means that 1 in 2 warnings is a false alarm. The precision for the DC model is lower, having 3 out of 4 false alarms. This does result in a recall of 66.7%. The high false alarm rates are not optimal. However, it must be taken into account that the prevalence of defaults is low (in particular for the DC set), making it difficult to distinguish defaults. The FC and DC model respectively score 5.5 and 6.7 times better on precision than random classification.

We have presented results on the performance of the two models. Next, we investigate the stability of these results.

4.1.6 Stability

The data is randomly partitioned into a training and a test part according to the ratio 2:1. The data sets in scope are relatively small, and hence it is likely that the proposed models are dependent on the split choice. It is therefore important to examine the stability of the models for different partitions of training and test set. To test this stability, we use a simulations. For 1000 random partitions we calculate the predictive measures and perform all 4 goodness of fit tests.

Table 4.5a shows average and standard deviation of the classification measures and the Pseudo-R^2. The average AUC is 0.827, which corresponds to a good overall performance.

Measure	Result
AUC	0.827 (0.028)
Precision	0.432 (0.061)
Recall	0.510 (0.054)
MacFadden	0.298 (0.023)
Tjur	0.293 (0.026)
HL	1.000
Pearson	1.000
Stukel	0.645

(a) Average AUC and GoF pass rates for the FC model. These statistics are based on 1000 random partitions of the training and test set.

Measure	Result
AUC	0.844 (0.037)
Precision	0.242 (0.053)
Recall	0.527 (0.086)
MacFadden	0.257 (0.029)
Tjur	0.184 (0.028)
HL	0.999
Pearson	1.000
Stukel	0.932

(b) Average AUC and GoF pass rates for the DC model. These statistics are based on 1000 random partitions of the training and test set.

Tab. 4.5.: Stability analysis of the GoF tests and the AUC for the FC and DC model. The tables present averages of 1000 random partitions of the training and test set.

The precision and recall are 0.432 and 0.510 respectively, meaning that 50% of the defaults are identified with a false alarm rate of 57.8%. The Macfadden and Tjur are both close to 0.3, indicating an excellent fit of the model to the data. The Homer & Lemeshow and standardized Pearson test are both passed for all 1000 partitions. Stukel's test is rejected 35.5% of the time. This indicates that in the corresponding partitions, deviations from the link function result in a more appropriate fit. The results for the DC model are presented in table 4.5b. The AUC of 0.844 represents a good overall classification performance and the recall and precision show that 50% of the defaults are identified with a 75% false alarm rate. The goodness of fit tests are rejected in almost none of the partitions, indicating that a proper model is specified. The pseudo-R^2 provide acceptable values, close to 0.20.

Overall, the models yield stable results for the different partitions. The overall classification quality is good, having AUC's above 0.8. The judgement whether or not the precision and recall are satisfactory are very dependent on the practitioners' trade-off between false alarms and unidentified defaults. However, taking the low prevalence of defaults (in particular for DC clients) into account, the results are acceptable. The rejection decision of Stukel's test for a large number of partition suggest that the FC model can be improved. However, the classification performance is good and the other tests are satisfied. Therefore, we decide not to revisit modelling choices.

4.2 Hierarchical logistic regression

In the previous section we developed two separate classification models. We have constructed a model for both the FC and DC set. Because of the different nature of the type of credit there are differences in the model regarding the variables and the coefficients. One difference is that the DC model contains limit variables which are not defined for the FC model. However, besides the models also contain similarities such as the variables arrear months and positive shocks. For the Bank it is favourable to have one model rather than two. Therefore, in this section we investigate the possibility to combine the two separate models into one model that is applicable to the entire portfolio.

To construct this model we use the hierarchical generalized linear models (also: mixed effects model, multi-level model). In particular, we focus on the hierarchical logistic regression. Hierarchical models are popular for modelling data that arises from a clustered structure (Levy, 2011) and are used in the field of credit risk by McNeil and Wendin (2003). There are two natural clusters in the data set, namely client with a fixed credit product and clients with a dynamic credit product. The clients in both clusters share variables with which we perform default classification. Besides these shared variables, the clusters also exhibit cluster-specific variables. These cluster-specific variables are present to a lesser extent (or absent) in the other cluster(s). Within the hierarchical framework it is possible to model using variables on both a shared and a cluster-specific level.

4.2.1 Formulation

Let us develop the hierarchical framework for the logistic model we have reviewed in section 4.1. The formulas, which form the hierarchical logistic regression are given in 4.27-4.30. The parameter vector $\boldsymbol{\beta}$ are still present in the linear predictor and are referred to as fixed effects. Random, cluster-specific effects are also included in the linear predictor and denoted by b_j for cluster j. The vector $\boldsymbol{b}$ consisting of b_j's is generally assumed to follow a multivariate normal distributions with a mean vector of all zeros, and covariance matrix Σ_b. The random effects can be considered as deviations from the fixed effects within a cluster. Together, the fixed and random effects result in a cluster-specific linear predictor vector η_j. Through the inverse logit function, this is linked to a cluster-specific probability vector π_j. The class distribution of the cluster j follows a binomial distribution with parameters π_j, and cluster size m_j.

$$\boldsymbol{b} \sim N(0, \Sigma_{\boldsymbol{b}}) \tag{4.27}$$

$$\eta_j = \beta_0 + b_{0j} + (\beta_1 + b_{1j})X_1 + \cdots + (\beta_M + b_{Mj})X_M \tag{4.28}$$

$$\pi_j = \frac{e^{\eta_j}}{1 + e^{\eta_j}} \tag{4.29}$$

$$\boldsymbol{y}_j \sim Binomial(m_j, \pi_j) \tag{4.30}$$

Figure 4.2 shows a plate representation of the structure of the model. This representation is typically used in illustrating the hierarchical set-up of the model. The circles represent the ingredients of the model, containing data and parameters. The arrows represent the dependencies that are in place. The figure shows boxes, which we refer to as plates. The plate around the $\boldsymbol{b}$ parameter represents the parameters for M clusters. The other plate indicates the N events for which we have corresponding data and the cluster identity.

4.2.2 Parameter estimation

Parameter estimation for the hierarchical logistic models requires some effort than the non-hierarchical. The structure of the covariance matrix needs to be determined beforehand. We do not have any assumption on the covariation between the two clusters and therefore

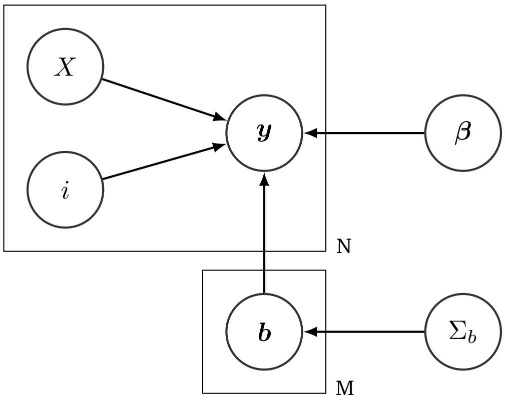


Fig. 4.2.: Plate representation of hierarchical modelling set-up.

assume they are independent. The likelihood includes the cluster specific terms b_j. Before we state the likelihood function, we define:

$$f(\boldsymbol{y}_j|\boldsymbol{b}_j,\boldsymbol{\beta}) = \prod_{i=1}^{n_j} \pi_{ij}^{y_{ij}}(1-\pi_{ij})^{(1-y_{ij})} \tag{4.31}$$

in which n_j denotes the number of observations in cluster j. The likelihood is now given by equation 4.32.

$$Lik(\beta,\Sigma_{\boldsymbol{b}_j}) = \mathbb{P}(\boldsymbol{y}_j|\boldsymbol{\beta},\Sigma_{\boldsymbol{b}_j}) = \prod_{j=1}^{J}\int f(\boldsymbol{y}_j|\boldsymbol{b}_j,\beta)r(\boldsymbol{b}_j,\Sigma_{\boldsymbol{b}_j})\mathrm{d}\boldsymbol{b}_j \tag{4.32}$$

This likelihood can not be evaluated exactly, and therefore an approximating technique is used. By means of this approximation, we determine the parameters for the fixed effects $\boldsymbol{\beta}$, and the covariance matrix $\Sigma_{\boldsymbol{b}_j}$, which reflects a diagonal matrix containing the standard deviation of the M estimated random effects. A further explanation on the derivation of the likelihood can be found in the appendix A.1.3.

The optimization process yields estimates for $\boldsymbol{\beta}$ and $\Sigma_{\boldsymbol{b}_j}$. As we are interested in the size of the cluster-specific effects, we infer the $\boldsymbol{b}_j$'s by using the point estimates $\hat{\boldsymbol{\beta}}$, and $\hat{\Sigma}_{\boldsymbol{b}_j}$:

$$\hat{\boldsymbol{b}}_j = \arg\max \mathbb{P}(\boldsymbol{b}_j|\hat{\boldsymbol{\beta}},\hat{\Sigma}_{\boldsymbol{b}_j},\boldsymbol{y}_j) \tag{4.33}$$

4.2.3 Estimation results

For the hierarchical model we require one data set which contain all clients. Some clients have both a FC and DC product, which means the corresponding data sets have overlapping entries. Hence, combining the two sets would results in non-independent observations in the full set. To correct for this, we preserve the entry of the DC data set and discard the FC

Variable	Coefficient	Standard error	p-value
Intercept	-3.182	0.188	<0.001
Arrear months	0.681	0.124	<0.001
Positive shocks	-0.341	0.130	0.009

Tab. 4.6.: Statistics of the estimated fixed effect parameters for the hierarchical logistic regression model. The estimates of the coefficients, the standard errors, and the p-values are given.

entry in case of overlapping entries. This decision results in the most balanced data set in terms of credit product.

The FC and DC model share two variables, namely arrear months and positive shocks. Besides these shared variables, both models have additional model-specific variables. The hierarchical model includes the shared variables as fixed effects, whereas the additional variables of both models are incorporated as cluster-specific effects. By construction, two clusters in the full set are the FC and the DC product type.

We split the data into a training and test set according to the ratio 2:1. In results not shown, the random effects for the balance returns are not significant in any cluster. The likelihood ratio test supports removal with the test statistic of $LR < 0.001$ and a corresponding p-value of $p = 1.000$. Table 4.6 and 4.7 show the parameter estimates of the final hierarchical model fitted on the training set. Table 4.6 contains statistics for the fixed effects (or shared variables) of the fitted model. The results in the table show that the fixed variables are significant. The coefficient values are in the same range as the values in table 4.1 and 4.3, indicating similar, and intuitively correct effects.

Table 4.7 shows the statistics of the estimated cluster-specific parameters per cluster. It is important to note that the limit use does not exist for the clients from the FC cluster (as they do not have a limit). To account for this, we set the limit use to zero for each client in this cluster, which results in an estimated variable coefficient of zero. As the limit use effect is a cluster-specific effect, the alteration of the data does not affect the estimate for the FC cluster. The variables size and zero transactions are significant in the FC cluster, whereas they are not in the DC cluster. Similarly, we find that the region, limit use, and cash expenditure are significant in the DC cluster, whereas there are not in the FC cluster. This behaviour corresponds to the selected variables in the separate FC and DC models. The DB-rating is significant in both clusters.

Figure 4.3 present the rate of classification curve which illustrates the performance of the model. The overall classification performance is excellent, having an AUC of 0.900. The precision and recall are 0.374 and 0.620 respectively. Rather than focussing on the results for this single partition, we again examine the average performance on multiple partitions of training and test set.

4.2.4 Stability

By including the additional variables into the model, the possibility exists that we are overfitting the training data. Therefore, we carefully examine the predictive performance

Variable	Cluster	Coefficient	Standard error	p-value
DB rating 0	FC	0.182	0.181	0.315
DB rating 1	**FC**	**-1.718**	**0.557**	**0.002**
DB rating 3	**FC**	**0.782**	**0.302**	**0.010**
DB rating 4	**FC**	**1.174**	**0.397**	**0.003**
region 1	FC	0.034	0.180	0.845
region 3	FC	0.426	0.283	0.132
Size M	**FC**	**0.842**	**0.260**	**<0.001**
Size L	**FC**	**1.576**	**0.378**	**<0.001**
Size XL	**FC**	**1.150**	**0.338**	**<0.001**
Zero trans.	**FC**	**0.389**	**0.094**	**<0.001**
Limit use	FC	0.000	0.977	1.000
Cat. cash	FC	0.011	0.129	0.933
DB rating 0	**DC**	**0.366**	**0.182**	**0.044**
DB rating 1	DC	-0.569	0.395	0.150
DB rating 3	DC	-0.273	0.318	0.390
DB rating 4	**DC**	**0.479**	**0.283**	**0.090**
region 1	DC	-0.248	0.182	0.173
region 3	**DC**	**-0.658**	**0.305**	**0.031**
Size M	DC	-0.035	0.300	0.907
Size L	DC	-0.653	0.527	0.216
Size XL	DC	-0.687	0.475	0.149
Zero trans.	DC	0.065	0.073	0.369
Limit use	**DC**	**0.968**	**0.180**	**<0.001**
Cat. cash	**DC**	**0.185**	**0.084**	**0.027**

Tab. 4.7.: Statistics of the estimated random effect parameters for the hierarchical logistic regression model. The estimates of the coefficients, the standard errors, and the p-values are given. Significant variables ($p < 0.10$) are shown in bold.

on unseen data. Rather than testing the performance on the single partition, we simulate the results for 100 partitions. Ideally, we would have run 1000 simulations, however the parameter estimation of the hierarchical model is computationally too intensive for this.

The model has an average AUC of 0.850, indicating a good classification performance. The precision and recall suggest that the model identifies 50% of the defaults in the set and 2 out 3 warnings are false. This indicates that the precision is approximately 5 times better than random. Both pseudo-R^2 measures are close to 0.3, indicating an excellent fit. Furthermore, the HL and standardized Pearson test are passed in each partition. In contrast, Stukel's test rejects the null hypothesis of a correct model fit in all of the partitions.

The several models in this chapter show that we can develop effective classification models in which transaction data plays a significant role. These models utilize the transaction data in a traditional way, by including aggregated measures as modelling variables. In the next chapter, we approach the transaction data in a fine-grained manner by using relational classification techniques.

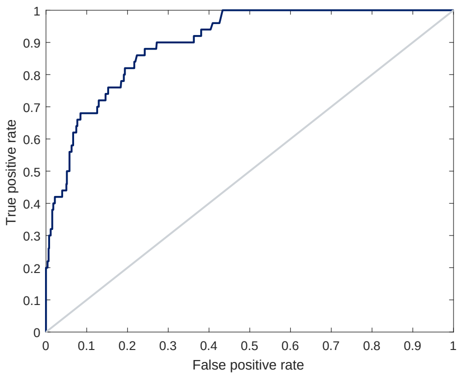


Fig. 4.3.: Rate of classification curve for the dynamic credit model. The corresponding AUC is 0.900.

Measure	Result
AUC	0.850 (0.024)
Precision	0.353 (0.038)
Recall	0.515 (0.049)
MacFadden	0.294 (0.021)
Tjur	0.298 (0.022)
HL	1.000
Pearson	1.000
Stukel	0.000

Tab. 4.8.: Statistics of the hierarchical logistic regression for 100 partitions of the training and test set. The table shows the predictive performance measures with corresponding standard deviation and average pass rates of the goodness of fit tests.

Relational Classification

5

The previous chapter describes the development of the default classification models. For both credit product types, we have constructed a separate logistic regression model. This is extended to a hierarchical logistic regression which allows for modelling all clients in the portfolio. These techniques are traditional in the sense that the data is used as variables for explaining the response variable. For this purpose, the transactions of a client are summarized into modelling variables that describe quantitative properties (e.g. volume, frequency). In this chapter, we approach the transaction in a non-aggregated manner. Section 5.1 describes a network approach to relate clients by means of their transaction similarity. The properties of such a network are discussed in section 5.2 and 5.3. The constructed network is used for a default classification model by using the relational classifiers defined in section 5.4. The classification performance is investigated in section 5.5.

5.1 Network construction

In this section we explain the network approach through which we relate clients. Instead of summarizing them in aggregated measures, we consider all individual transactions in a fine-grained manner. By relating clients to each other that have transacted with the same entities, a behavioural network can be constructed. In a recent paper, Tobback and Martens (2017) use consumer transactions in this manner, to provide a credit score for individuals. In marketing, transactions are used in order to develop targeted marketing strategies (Martens et al., 2016). Looking beyond transactional data, a common example of a behavioural data set is a network of social connections. For example in Cnudde et al. (2015), Facebook connections are used to determine the creditworthiness of individuals. These approaches originate from the assortativity theory in the sociological domain, which assumes that people or companies with similar behavioural characteristics tend to cluster. Our hypothesis builds upon this concept by assuming that clients with similar transactional behaviour have comparable creditworthiness.

Two types of relational networks exists, namely a direct and an indirect network. In the former, links are constructed if two clients have directly transacted with each other, whereas the latter forms links if clients have transacted with the same entity. The portfolio is relatively small, meaning the direct network will be very sparse. Hence, we focus on the indirect network. Figure 5.1 shows how an indirect network is derived from transactional data. The left figure shows an example of possible transactions, whereas the corresponding (indirect) network is presented in the right figure.

To construct the network, let us introduce the following notation: Let $\mathcal{I} = \{1, \ldots, N\}$ represent the clients and let $\mathcal{K} = \{1, \ldots, K\}$ be the set of existing entities. We define the binary variable e_{ik} as 1 if $i \in \mathcal{I}$ has existing transactions with $k \in \mathcal{K}$, and 0 if not. Using this variable, we construct the set $\mathcal{K}_i = \{k \in \mathcal{K} | e_{ik} = 1\}$ representing all entities that i

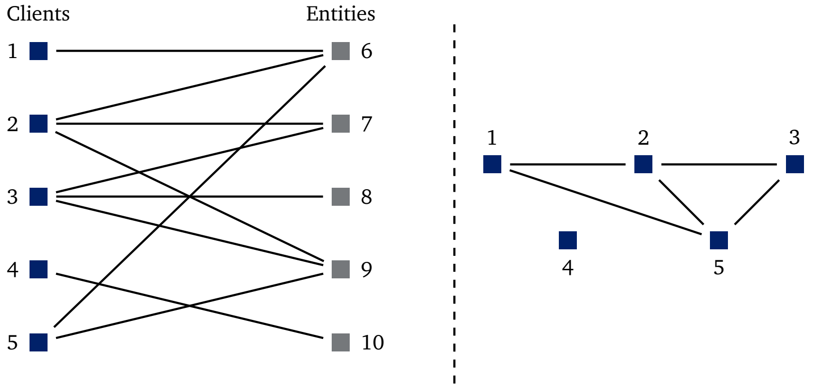


Fig. 5.1.: Example of how the indirect network is constructed from transactions between clients and entities.

has transacted with. The degree d_k is the number of clients that k is connected to, i.e. $d_k = \sum_i \mathbf{1}\{k \in \mathcal{K}_i\}$.

5.1.1 Client similarity

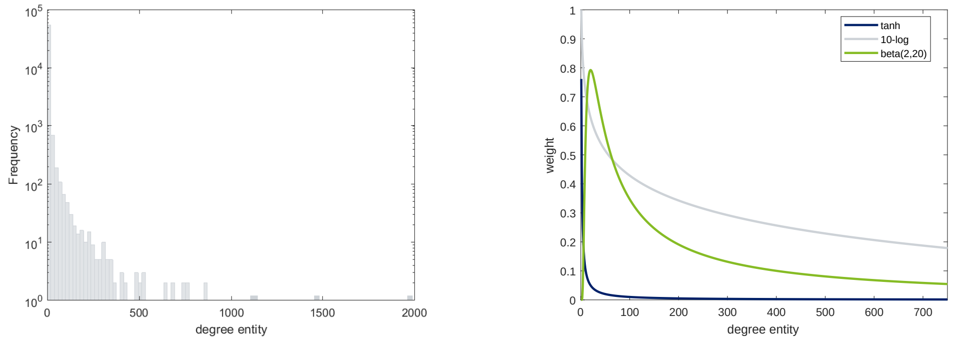
The purpose of constructing a relational network is to describe the similarity between clients in terms of their transaction behaviour. Before we formally define the similarity, we must note that not every shared entity provides the same information. Most clients in the SME portfolio have a shared entity, namely the government. The question arises whether a link through the government provides much information. A link formed through an uncommon entity, such as a local grocery store, provides more valuable information in terms of similarity. To account for this we introduce a weight function w_k which represents the weight of entity k. The similarity between two clients i and j is defined as the summed weight over the shared entities. Equation 5.1 formalizes this.

$$s_{ij} = \sum_{k \in (\mathcal{K}_i \cap \mathcal{K}_j)} w_k \tag{5.1}$$

The s_{ij} form the entries of the $N \times N$ similarity matrix S. Next, we describe several weight functions.

Degree based weight

A basic form of a weight function is the reciprocal function of the degree of an entity, given in equation 5.2. A hyperbolic tangent function of the inverse degree is used in Tobback and Martens (2017) and stated in equation 5.3. This function behaves similarly to the reciprocal function, decreasing very rapidly as the degree increases. The difference is that the hyperbolic tangent assigns a lower weight to entities with few connections compared to the reciprocal function. Furthermore, Martens et al. (2016) propose a logarithmic function as stated in 5.4. The logarithmic function has a more gradual descend then the other two



(a) Distribution of the degree of the entities.

(b) Weight functions of the entity degree.

Fig. 5.2.: The figures show a histogram of the entity degree (on a log-scale) and three possible weight functions that require the degree as an input.

functions. Martens et al. (2016) also introduce a weighting function based on the pdf of the beta distribution $(f_{\beta(a,b)})$ with parameters a and b as defined in equation 5.5. This offers a flexible way of determining the weight function. However, the optimal a^* and b^* need to be learned from the data, which increases the complexity of our problem significantly.

$$w_k = \frac{1}{d_k} \tag{5.2}$$

$$w_k = \tanh\left(\frac{1}{d_k}\right) \tag{5.3}$$

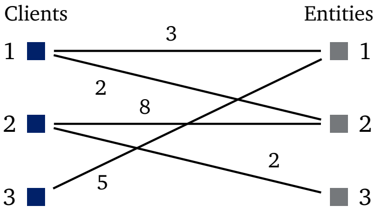
$$w_k = \log_{10}\left(\frac{\max_k d_k}{d_k}\right) \tag{5.4}$$

$$w_k = f_{\beta(a,b)}\left(\frac{1}{d_k}\right) \tag{5.5}$$

To choose a proper weight function, we examine the degree distribution of the entities. Figure 5.2a shows an histogram of the degree of the entities. Entities with a degree of 1 attribute nothing in terms of client connectivity and are therefore removed from the analysis. For illustrative purposes, the frequency of the histogram is given in a log-scale. This shows that the number of entities having a small degree is large compared to the entities with a degree of 500 or larger. Figure 5.2b shows different weight functions that translate the degree to a weight. The figure shows the log, tangent hyperbolic, and a beta(2,20)-function. The first two assign a decreasing weight as the degree increases, in which the log function has a more gradual decrease. The beta function also assigns a gradually decreasing weight, however it also downweighs entities of low degree. One could argue that these connections are formed by chance and are therefore uninformative. The beta function has the strength of being flexible, and is able to represent such a belief.

Volume based weight

With the degree-based approach, the similarity between clients depends on the degree of their shared entities. A client is either connected to an entity or not, depending on whether



(a) Example client-entity relations.

Client	1	2	3
1	-	$\frac{8}{10} \cdot \frac{2}{5}$	$\frac{3}{5} \cdot \frac{5}{5}$
2	$\frac{8}{10} \cdot \frac{2}{5}$	-	0
3	$\frac{3}{5} \cdot \frac{5}{5}$	0	-

(b) Resulting similarity matrix.

Fig. 5.3.: Example of how similarity can be determined through (volume or frequency)-weighted entity connections.

a transaction has occurred at least once. This is a binary event and does not take any further transaction information into account. Consequently, a client that has transacted an amount of €0.01 to an entity once, has the same connection towards that entity as a client that has transacted a total amount of €10.000 in several transactions. To account for this, we define an approach that takes the transaction volume into account.

The strength of the connection of a client with the entity is determined via the relative transaction volume. Let us denote ω_{ik} as the weight between a client i and an entity k as follows:

$$\omega_{ik} = \frac{m_{ik}}{M_i} \tag{5.6}$$

in which m_{ik} is the total transacted volume between m and k, and $M_i = \sum_{k=1}^{K} m_{ik}$. Instead of using equation 5.1, the similarity is calculated as follows:

$$s_{ij} = \sum_{j<>i}^{N} \sum_{k=1}^{K} \omega_{ik}\omega_{jk} \tag{5.7}$$

For further explanation, figure 5.3 shows an example of this method.

Frequency based weight

An alternative to the volume based approach, is the frequency-based similarity. Rather than considering the relative volume that is transacted with an entity, the number of transactions is considered. In calculating the similarity, this is incorporated by replacing ω_{ik} in equation 5.6 by:

$$\omega_{ik} = \frac{a_{ik}}{A_i} \tag{5.8}$$

in which a_{ik} is the total number of transactions between m and k, and $A_i = \sum_{k=1}^{K} a_{ik}$. To construct the similarity matrix, formula 5.7 is used.

For the remainder of this paper, we refer to the different similarity matrices as follows:

- S_1: hyperbolic tangent (degree-based)

- S_2: logarithmic (degree-based)
- S_3: beta (degree-based)
- S_4: volume-based
- S_5: frequency-based

5.1.2 Implementation

To effectively calculate the similarity matrices we rely on matrix multiplications. Let us introduce the client-entity adjacency matrix X. This is a $N \times K$ matrix which denotes a one if a client i is connected to an entity k and zero otherwise. The transactional data set contains hundreds of thousands unique entities and most clients are adjacent to only a few hundred of them. Thus the matrix X is sparse.

Performing the matrix multiplication $S = X \cdot X^T$ would result in the number of shared entities between two clients for each entry of the matrix S. To include the entity weight we introduce X_w. For S_4 and S_5, the entries of X_w represent the volume and frequency that the client has transacted with the entity. To construct X_w for the degree-based similarity matrices, the weight function w_k is applied to the matrix X, and the square root is taken element-wise. The similarity matrices are found by performing the multiplication $S = X_w X_w^T$.

Proposition 1. *The matrix $S = X \cdot X^T$ is symmetric.*

Proof. We have constructed the similarity matrix $S = X \cdot X^T$. For S to be symmetric, it must hold that $S = S^T$.

$$S^T = (X \cdot X^T)^T = (X^T)^T \cdot X^T = X \cdot X^T = S$$

which concludes our proof. □

All similarity matrices are constructed through a multiplication of a matrix with its transpose, making them symmetric. Asymmetry could be reflected by introducing directed edges. However, this would further complicate the network and future calculations.

5.1.3 Network example

The similarity matrices represent relationships between clients. These relationships can be represented by a network, in which the nodes reflect clients and the edges the existence of a connection between them. By including edge weights, the strength of the connection (i.e. similarity) is captured. Figure 5.4 shows a subset of 15 (defaulted and non-defaulted) clients of the total portfolio. The thickest edges, representing high similarity, are found between the client 3,4, and 13, of which the former two are defaulted. The idea behind relational classification is that this relation increases the default probability of client 13. Furthermore, an interesting observation is that client 5 is disconnected from the network, meaning it has no relation to the other 14 clients. Note that this sample only reflects a fraction of the full network, and therefore no conclusions must be drawn from this.

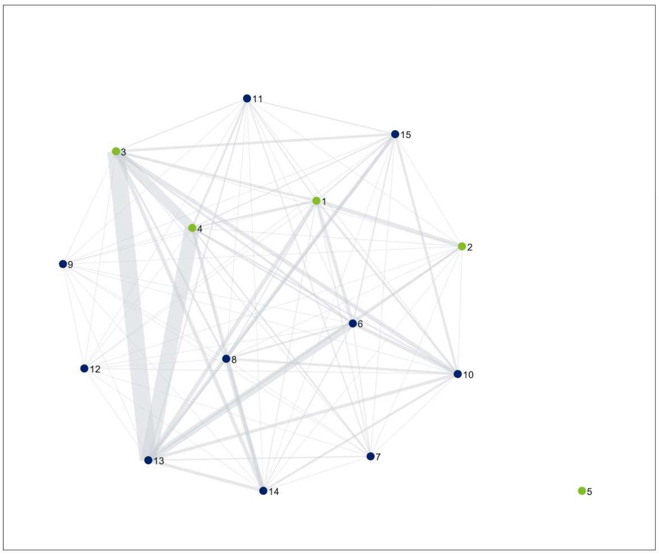


Fig. 5.4.: Network of a sample of 15 clients from the portfolio. The green and blue nodes represent defaulted and non-defaulted clients. The thickness of the edge is respective to the similarity between the clients.

In addition to the default status of clients, it also possible to illustrate different properties, such as industry, region, or size. While these properties do not provide immediate information on the default status, it is interesting for portfolio insight. For example, dependencies (or non-dependencies) within certain industries could be discovered. On a larger scale, it is difficult to see structure by eye. Therefore, we have to conduct quantitative tests in order to draw conclusions. In the next sections we will conduct such tests.

5.1.4 Statistically validated network

In constructing the similarity matrices, the existence of all shared entities between clients is taken into account. We assume that all client-entity connections provide information in some way, which is represented by the several weight functions. Rather than assuming that all connections are informative, it is possible that some connections merely contribute noise to the system. As a result, client connections may exist merely as a result of random occurrences of shared entities. Tumminello et al. (2011) propose a method to statistically validate each client-entity connection in terms that they are not explainable by randomly occurring connections. The resulting, statistically validated, network is represented by a similarity matrix containing the number of statistically validated shared entities.

The method works as follows. Let us have the bipartite network B of all client-entity connections. Consider the subset $K^d = \{k \in \mathcal{K} | d_k = d\}$, representing the subset of entities that have degree d. We construct the subsystem B^d, which contains all entities in K^d and the clients I^d connected to them. Furthermore, let us denote N_i^d and N_j^d as the degree of the client i and j in the subsystem B^d. Under the hypothesis that clients randomly form connections with entities in K^d, the probability that i and j share X entities in the system B^d is given by a hypergeometric distribution as given in equation 5.9.

$$h(X|N_K^d, N_i^d, N_j^d) = \frac{\binom{N_i^d}{X}\binom{N_K^d - N_i^d}{N_j^d - X}}{\binom{N_K^d}{N_j^d}} \tag{5.9}$$

The actual number of shared entities between i and j in the subsystem is denoted by N_{ij}^d. The distribution H allows us to construct a p-value representing the probability that, under the assumption of randomly forming connections, a number of N_{ij}^d or more shared entities between clients are observed. This probability is calculated as follows:

$$p(N_{ij}^d) = H(X \geq N_{ij}^d) = 1 - H(X \leq N_{ij}^d - 1) \tag{5.10}$$

$$= 1 - \sum_{X=0}^{N_{ij}^d - 1} h(X|N_K^d, N_i^d, N_j^d) \tag{5.11}$$

in which $H(X)$ represents the cumulative distribution function corresponding to $h(X)$.

If the p-value is lower than some threshold γ, we conclude that the number of shared entities within the subsystem is different from random. Hence, the client similarity provided by the entities in K^d are statistically valid. To validate all client connections in the network, we repeat this method for all subsystems. There are 248 unique degrees within the set of entities, meaning we analyse 248 subsystems.

We find that 48.2% of the client-entity connections are statistically valid at a 0.05 level of significance. This indicates that more than half of the client-entity connections provide noise and should not be included in client similarity calculations. However, we must note that as a result of the large heterogeneity in entity degree, approximately 75% of the subsystems have six or less entities. The minimum p-value in these systems is larger than 0.05 for all combinations of N_i, N_j and N_{ij}. Thus, none of the client-entity connections are significantly different from random.

The statistical validation of the client entity connections shows that not all the similarity between clients from shared entities is different from noise. However, due to the relatively high heterogeneity in entity degree, the hypergeometric test is often rejected. The relational classification approach we will incorporate benefits from a connected network. For this reason, we focus on the full network (as represented by the similarity matrices), rather than the statistically validated one. The downside is that we are likely to include noise into the network.

5.2 Network topology

The previous section describes how the relational network is developed by means of client transactions. The resulting transaction-based (TB) network represents the similarity between clients. Before we investigate the propagation of information through the network, we examine the topological structure. We examine whether this structure resembles known types of network and whether it differs from a randomly created network.

5.2.1 Network type

In Solé and Valverde (2004), real-life networks are discussed and categorized according to their underlying properties. The most prominent types are random, scale-free, and small-world networks (SWN). Random networks are generally constructed by randomly generating edges between existing nodes. A network is considered to be scale-free if the degree distribution follows a power law. These networks exhibit hubs that have a very high node degree. SWNs are typical in social networks and exhibit properties such as high clustering and low shortest path lengths (Bialonski et al., 2010). To compare the TB network to these known types, we analyse three properties, namely: the degree distribution, the (unweighted) clustering coefficient C, and the efficiency E.

There are several ways to define a random network. A basic form is the Erdös Rényi (ER) network (Erdös and Rényi, 1960). Other possibilities are to construct conditional random graphs, in which constraints on the edge selection are imposed (Squartini and Garlaschelli, 2011). In the TB network there are no natural constraints. Hence, we focus on ER networks, which is defined in definition 1.

Definition 1. *For a fixed number of nodes n and N edges, an Erdös Rényi (ER) network is defined by choosing N of the possible $\binom{n}{2}$ edges. Each edge is chosen with equal probability namely $\frac{1}{\binom{n}{2}}$.*

We examine the degree distribution of the TB as well as the ER network. Figure 5.5 presents a histogram of the distribution of the node degree of these networks. The degree of the ER network is normally distributed with mean of 1659, and does not exhibit a power law tail. While the average degree in the TB network is also 1659, there is more heterogeneity in the degree distribution. The degree of many nodes is higher than 2000, resulting in a peak right of the ER histogram. Low node degrees are also present in the PB network, resulting in a long tail as node degree decreases. The peak at degree 0 shows the existence of approximately 200 unconnected nodes. The corresponding clients are special cases in the relational classification. If we make a distinction in default status, we find that the average degree among non defaulted clients is 1.2 times higher. Thus, defaulted clients seem less connected.

Figure 5.6 shows a log-log (or zipf) plot of the node degree. The plot is used to detect power law behaviour in the degree distribution. The cumulative distribution function of the Pareto (type I) distribution is given by:

$$F_\gamma(x) = \begin{cases} 1 - \left(\frac{x}{x_0}\right)^{-\gamma} & \text{for } x \geq x_0 \\ 0 & \text{for } x < x_0 \end{cases} \tag{5.12}$$

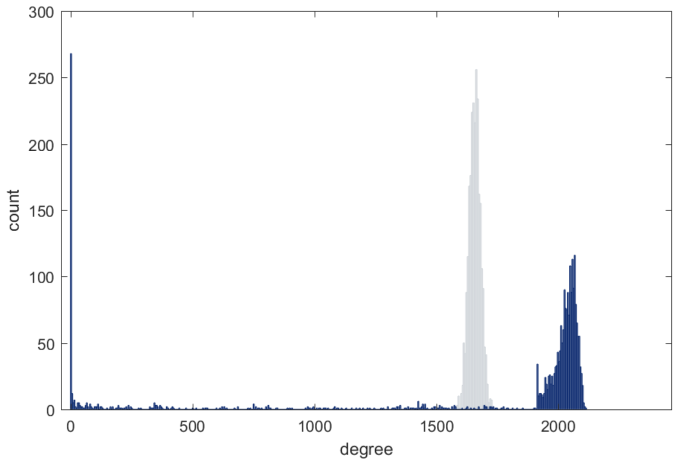


Fig. 5.5.: Histogram of the node degree of the payment-based network (blue) and an ER network (grey).

By taking the log of the survival function we find the following relation:

$$\log(1 - F_\gamma(x)) = \log\left(\left(\frac{x}{x_0}\right)^{-\gamma}\right) = \gamma\log(x_0) - \gamma\log(x) = C - \gamma\log(x) \qquad (5.13)$$

in which C and x_0 are constants. Similar relationships can be derived for other types of power law distributions.

Equation 5.12 implies the necessary condition that the log-log plot shows a linear relationship as x increases. Figure 5.6 clearly shows that this relation is not present and hence we conclude that there is no power law behaviour in the degree distribution. Consequently, the PB network is not a scale-free network. This implies that the network has a relatively homogeneous distribution, and that there are no nodes that have many more connections than others (often called hubs). This is beneficial for our network analysis, as the absence of hubs means that all clients are (more or less) equally influential. This is intuitively correct as our portfolio does not contain clients on which a large portion of the portfolio is highly dependent.

The shortest path length between two nodes i and j is the smallest number of edges needed to reach i from j. The average shortest path length L is calculated by taking the mean shortest path length between all pairs (i, j). The formula for calculating this value is given in appendix D.4. A problem arises for unconnected nodes, which have a shortest path of infinity to every other node. To resolve this, we replace L with the efficiency E, which is defined in appendix D.5.

The clustering coefficient C is a measure of how tightly knit a network is and is defined in D.1. It represents the number of closed triplets as a fraction of the total number of connected triplets. Generally, C and L are both low in a random network, whereas a SWN is characterized by a low L, and a high C (Bialonski et al., 2010). A network is often characterized as small world if it satisfies the conditions $\frac{L}{L_r} \approx 1$ and $\frac{C}{C_r} > 1$, in which L_r

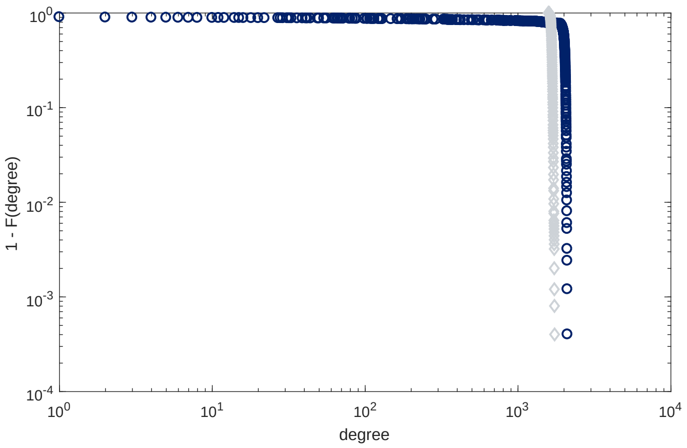


Fig. 5.6.: The figure shows the degree vs the empirical distribution function on a log-scale. The blue circles and the grey diamonds represent the payment based network and the ER network respectively.

Measure	Payment-based	Erdös Rényi
Clustering (C)	0.865	0.667 (<0.001)
Efficiency (E)	0.834	0.756 (<0.001)
Betweenness (B)	<0.001	<0.001 (<0.001)

Tab. 5.1.: Statistical measures of the PB network and the ER network. Averages and standard errors (between brackets) of the 1000 randomly generated ER networks are given.

and C_r denote the measures for random networks. By replacing the shortest path length with the efficiency, we find $\frac{E_r}{E} \approx 1$, which is equivalent to the former condition.

Table 5.1 shows the values for C and E for the PB and ER network. The maximum C of a network is 1, which means that the clustering in the TB network is high. However, the clustering in the ER network is also significant, which can be explained by the high connectivity of the network. The average node degree is approximately $2/3$ of the total node degree, which results in a relatively high clustering coefficient. The efficiency of both the TB as the ER network is high, indicating that the average path length is low. This is a result of the fact that many nodes are directly connected, resulting in shortest paths of length 1.

The combination of a high heterogeneity in the degree distribution and a high clustering coefficient indicate that TB network is different from random. Furthermore, the power law behaviour of the degree distribution is absent, showing that the network is neither scale-free. The network does show similarities to the small world network type. However, the high average degree of the network results is uncommon in classic small world networks (Solé and Valverde, 2004). The properties of a small world network are not needed in our further analysis, therefore we remain indifferent in classifying the PB network as small world.

5.2.2 Robustness

Robustness is a measure of the sensitivity of the network to removal of edges or vertices. If the network's information propagation is dependent on a small set of nodes, removal or wrong information could disrupt the whole network (Ellens and Kooij, 2013). This is especially important in scale-free networks, which are characterized by hubs of highly connected nodes. We quantify the robustness through additional statistical properties.

We have already determined that the PB-network has a high clustering coefficient, which indicates the existence of alternative paths. Furthermore, the high value for E and high average degree all indicate that the network is highly interconnective. To further analyse the robustness, we introduce the betweenness measure B (details can be found in appendix D.3). For each node, this measure calculates the number of shortest paths between vertices that contain that particular node. Hence, it determines the importance of nodes in the network.

Table 5.1 shows that the average betweenness is very low for both the PB and the ER network. This is related to the low average shortest path length in the graph. A shortest path of 1 means that two nodes are directly connected, meaning there are no other nodes on the shortest path between them. As many pairs in the graph are directly connected, this results in a low betweenness value for all nodes. Consequently, the average betweenness is also low. A low betweenness means that removal of the node does not have a large effect on the number of paths. Hence, it indicates a high robustness.

Together with the high C, and high E we conclude that the TB network is robust. Thus, information propagation is not reliant on a subset of nodes.

5.3 Network properties

After discussing the topological structure of the network, we now examine the information contained in it. For the purpose of credit scoring, we expect high similarity between defaulting clients. It is also interesting to examine the similarity between clients among properties such as industry, size, and region.

5.3.1 Homophily

The tendency of clients to connect with others with similar properties is called homophily and can be measured for different characteristics. Social networks, for instance, have a tendency to be strongly linked according to race and language (David and Jon, 2010). In contrast, a network can also show inverse homophily. For example, romantic relationships in a school would display strong inverse homophily with respect to gender. Naturally, we investigate whether clients show homophily according to their default status. For portfolio insight, we also consider other properties such as size, industry, region, and DB-rating.

To investigate the existence of homophily, the network is tested whether it shows a significantly different structure than a random network. Let us illustrate a general example. Consider a graph G with node set V and unweighted edges E. Every node has either property c_1 or c_2. We test if there exists homophily with respect to c_1. We make inferences via a non-parametric approach. We construct several random networks by assigning class labels

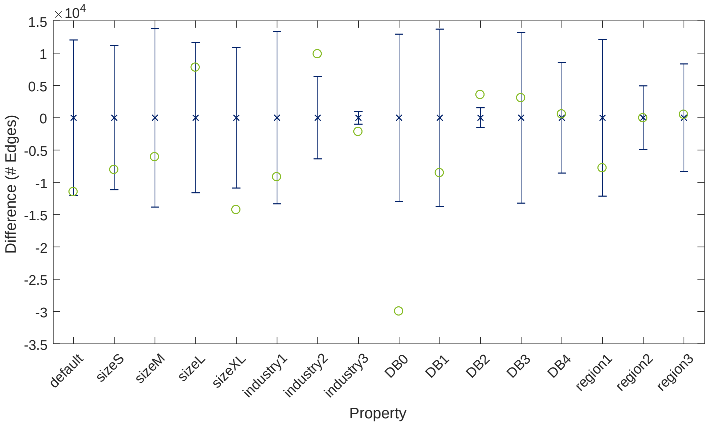


Fig. 5.7.: Homophily test for different client properties. The difference of cross-class edges between TB network and a random network is presented, including a confidence interval.

randomly to each node, while respecting the proportions of the classes. After constructing 1000 samples, we infer the mean number of cross class edges in a random network, and construct a 95% confidence interval. To test for homophily of a property, we investigate whether the number of cross class edges in the TB network falls within the confidence interval of the random networks.

Figure 5.7 shows the difference between the number of cross-class edges in the TB network and the mean number of cross-class edges in the random samples, including the confidence interval. There are significantly less cross-class edges than random if we consider the properties: default, DB-rating of 0, or industry 3. Hence, the network exhibits homophily w.r.t. these properties. In contrast, inverse homophily is displayed for the properties DB rating 2 and industry 2.

The observation of homophily within defaults is promising for the credit scoring purpose. It indicates that a relationship between defaults can be distinguished that is different from non-defaulted clients. The other properties potentially provide interesting insights in the portfolio.

5.3.2 Clustering

Homophily tests are based on the existence of a connection, i.e. the edges between clients. These tests neglect the strength of a connection, which in the TB network represents the client similarity. In section 5.2, the clustering coefficient is used. An extension to this is the weighted clustering coefficient (WCC), which allows for networks with weighted edges. Different approaches to calculating the WCC are discussed in Pahn et al. (2013). The definition and calculation of the WCC is given in appendix D.2.

The WCC of the network is calculated through this formula. To determine whether clients cluster according to specific properties, we calculate the WCC for subgraphs G^p. These subgraphs are defined by $G^p = (V^p, E^p)$ in which $V^p = \{i \in V | i \text{ has property } p\}$ and $E^p \subseteq E$ are the corresponding edges of this subset of nodes. The clustering coefficient for a specific property WCC_p, is found by calculating the WCC for the corresponding subgraph G_i.

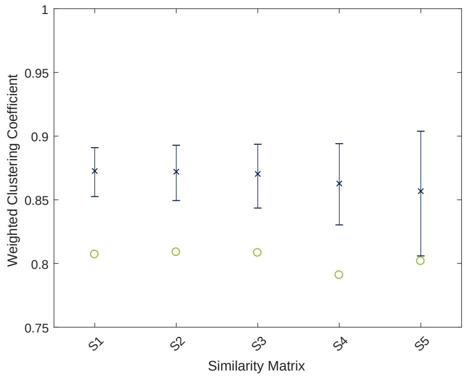


Fig. 5.8.: Average weighted clustering coefficient of defaulted clients for all five similarity matrices.

By comparing the WCC for a specific property with the WCC_p, we can determine whether or not the graph shows clustering amongst clients with that property. The higher the coefficient, the stronger the clustering.

To assess the WCC of a property p, we compare it to the mean WCC of 1000 sample networks. Similar to the homophily tests, these samples are constructed by randomly assigning nodes the property p, accordingly create G^p and calculate the WCC_p for each sample. From the results, we infer a mean and a confidence interval. If the WCC_p of the original graph falls outside the confidence bounds, it means the property shows different clustering behaviour from random.

Calculating the WCC and the confidence intervals is time-consuming, especially in large networks. Additionally, the calculations need to be performed for all similarity matrices, as these contain different weights. For this reason, we have chosen to only perform these calculations for the default status. Figure 5.8 shows the WCC, for the five different similarity matrices. The mean and confidence interval of the random samples are also presented.

Figure 5.8 shows that the WCC for all networks is significantly lower than random selections of nodes in the network. This indicates that the similarity among the subgraph of all defaulted clients is relatively low. This may affect the performance of the relational classification approaches described in the next section. However, these approaches use the similarity between clients on an individual, rather than a group level. In addition, the average degree of defaulted directly lowers the WCC. Therefore, the relational classification may still be effective.

5.4 Relational classification

By making use of the relationships in the network, information can propagate throughout the network. Applied to credit scoring, clients that are known as defaults increase the probability of default for strongly related neighbours. This technique is known as relational classification. There are different relational classifiers. In Macskassy and Provost (2003), the relational neighbour (RN) and probabilistic relational neighbour (pRN) are introduced. The RN classifier uses the known class labels of neighbouring nodes to estimate the class probability of a node. Information of the unlabelled nodes is not taken into account. The pRN does include unlabelled nodes and assigns a class prior to them. Smoothed versions of the RN and pRN exists (Tobback and Martens, 2017; Toback et al., 2017), which are also defined for unconnected nodes. The class distribution relational neighbour (cdRN) is used in both Macskassy and Provost (2007) and Thuraisingham et al. (2016). In this classifier, class probabilities are determined by learning a distribution of the class labels of neighbouring nodes.

5.4.1 Classifiers

We provide the definitions of the mentioned classifiers. We have the similarity matrix S with entries s_{ij}. In addition, let us introduce the following notation: The variable l_i denotes the class label of node i and the nodes in the set $N(i) = \{j \in V | s_{ij} > 0\}$ are the neighbours of i. Note that i is not an element of $N(i)$.

Relational neighbour

The relational neighbour is given in equation 5.14. The probability that the client i belongs to class c is equal to the sum of the similarity of all neighbouring nodes that have known class labels c. In this equation $Z_i = \sum_j s_{ij}$ represents a normalizing constant. The normalizing constant ensures that the resulting probabilities lie within the interval $[0, 1]$.

$$\mathbb{P}(l_i = c | N(i)) = \frac{1}{Z_i} \sum_{j | l_j = c} s_{ij} \tag{5.14}$$

If a node is disconnected, the neighbouring set is empty and the probability for all classes is set to zero. Furthermore, the normalizing constant is zero and the fraction is undefined. To resolve these issues, we introduce a smoothed version of the RN, given in equation 5.15.

$$\mathbb{P}(l_i = c | N(i)) = \frac{\sum_{j | l_j = c} s_{ij} + 2\mu_c}{Z_i + 2} \tag{5.15}$$

The smoothed version assures that unconnected clients are assigned the probability of the class prior μ_c.

Probabilistic relational neighbour

The relational neighbour uses known class labels. Especially in situations of class imbalance, the number of labelled neighbours might be limited, which means few neighbours are used in determining the class probability of a node. By using probabilities rather than labels, all neighbours are included in calculating the class probabilities. The pRN, defined in 5.16, is based on this principle.

$$\mathbb{P}(l_i = c|N(i)) = \frac{1}{Z_i} \sum_{j \in N(i)} s_{ij} \mathbb{P}(l_j = c|N(j)) \tag{5.16}$$

Again, a smoothed version is used to allow for unconnected clients.

$$\mathbb{P}(l_i = c|N(i)) = \frac{\sum_{j \in N(i)} s_{ij} \mathbb{P}(l_j = c|N(j)) + 2\mu_c}{Z_i + 2} \tag{5.17}$$

Class distribution relational neighbour

The cdRN uses the neighbouring class labels to determine a class distribution. Let us define the class vector $\boldsymbol{CV}(i)$ for a node i in equation 5.18 and the reference vector $\boldsymbol{RV}(c_k)$ in equation 5.19. The class vector represents the total similarity to the various classes, whereas the reference vector is the average of all class vectors known to be of class c_k.

The reference vector needs to be trained. Hence, we divide the data into a training and test part. For all nodes in the training set, $\boldsymbol{CV}(i)$ is calculated as follows:

$$CV_k(i) = \sum_{j \in N(i), l_j = c_k} s_{ij} \tag{5.18}$$

in which k represents the k-th entry in the vector, corresponding to the class label c_k. Then, according to the vectors $\boldsymbol{CV}(i)$, we construct the reference vector.

$$\boldsymbol{RV}(c_k) = \frac{1}{|V_{c_k}^L|} \sum_{i \in V_{c_k}^L} \boldsymbol{CV}(i) \tag{5.19}$$

in which $V_{c_k}^L = \{i \in V | l_i = c_k\}$, i.e. the set of known nodes of class label c_k.

Now, for the test set, we alter the definition of the class vector such that it includes the class probabilities.

$$CV_k(i) = \sum_{j \in N(i), l_j = c_k} s_{ij} \mathbb{P}(l_j = c_k|N(j)) \tag{5.20}$$

Using these definitions, the class probabilities for the test set can now be estimated using the cdRN, given in equation 5.21.

$$P(l_i = c_k|N(i)) = D(\boldsymbol{CV}(i), \boldsymbol{RV}(c_k)) \tag{5.21}$$

in which D is a vector distance measure, e.g. L_1, L_2, or the cosine. In this research, we use the latter, which is defined as:

$$D_{cos}(\mathbf{A}, \mathbf{B}) = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2}\sqrt{\sum_i B_i^2}} \tag{5.22}$$

in which A_i and B_i represent the i-th entry of the vectors $\mathbf{A}$ and $\mathbf{B}$.

5.5 Results

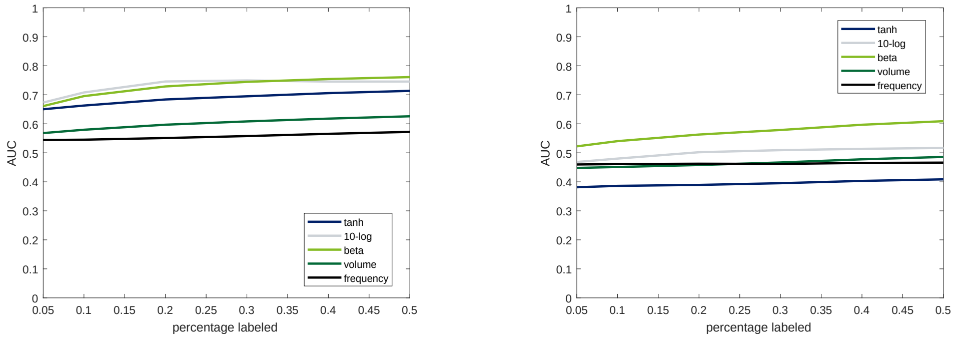
In this section, we compare different classification approaches applied to the several similarity matrices in order to estimate class probabilities. We investigate the classification techniques for all five similarity matrices. The classes we consider are default and non-default. To apply the classification techniques, we use the historical defaults in the combined data set (containing FC and DC).

To test the performance of our relational classifiers, we split the data into a training and a test set. The size of the training set is varied from 5 to 50 percent. The results are dependent on the training/test partition of the data. To correct for this, we vary the partition 100 times and present the average performance. For the pRN, and cdRN, we assume that the probability of default of the non-labelled instances is equal to the prior PD. We choose this prior to be the overall default rate of the portfolio. This is a basic assumption which could be extended to size or credit type specific priors.

Construction of S_3 requires the most computational effort because of the tuning of the weight function parameters. This is done as follows: For several a and b, the similarity matrix is constructed. Next, the labelled instances are split into two equal subsets: a training, and a validation set. Using a relational classifier, the classification performance of the validation set is measured for all similarity matrices in terms of the AUC. The parameters that yield

Algorithm 1 Parameter Tuning

```
Given: Parameter sets $A, B$;
procedure OPTIMIZE($a, b$)
    for $a \in A$,$b \in B$ do
        $w(d_k) \leftarrow f_{\beta(a,b)}(d_k)$
        Construct $S$                                  ▷ Equation 5.1
        Calculate $\mathbb{P}(l_i = c|N(i))$           ▷ Equations 5.14,5.16 or 5.21
        Evaluate $AUC$ of validation set
        if $AUC > AUC^*$ then
            $AUC^* \leftarrow AUC$
            $a^* \leftarrow a$
            $b^* \leftarrow b$
        end if
    end for
    return $a^*, b^*$
end procedure
```



(a) Average AUC of the RN.

(b) Average AUC of the pRN.

Fig. 5.9.: Average AUC of the RN and the pRN for multiple weight functions and different percentages of labelled data. The average is calculated according to 100 partitions of the training and test set for each percentage.

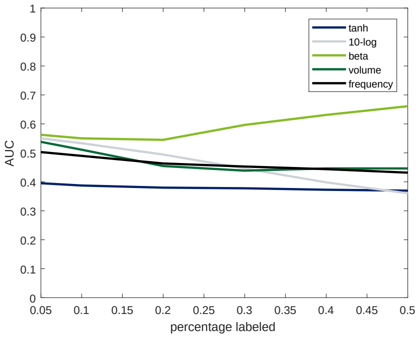


Fig. 5.10.: Average AUC of the cdRN for multiple weight functions and different percentages of labelled data. The average is calculated according to 100 partitions of the training and test set for each percentage.

the best performance for the validation set are selected. This process is described in the pseudo-algorithm 1.

Figures 5.9a, 5.9b, and 5.10 show the AUC of the RN, pRN, and cdRN classifier respectively. With the RN classifier the AUC is the lowest for the frequency based similarity matrix. The two weight functions that perform best are the 10-log and the beta function, having an AUC that is consistently above 0.65. This performance gradually increases to 0.75 as the percentage of labelled instances rises to 50%.

The performance of the pRN is significantly lower than the RN for all weight functions. Only the AUC for beta function is consistently above 0.5, which is the AUC of random classification. The pRN classifier includes all connected nodes in the classification process. A possible explanation for the worse performance is that the probabilities are more similar to the network's average and therefore less distinctive in recognizing defaults.

Figure 5.10 shows that the cdRN is affected by the percentage of labelled instances. The AUC decreases for all but the beta function similarity matrix as the information in the network grows. The performance of the beta function significantly increases when more than 20% of

Similarity matrix	RN	pRN	cdRN
S_1 (tanh)	**0.725 (0.026)**	0.423 (0.035)	0.361 (0.031)
S_2 (10-log)	**0.749 (0.030)**	0.524 (0.036)	0.312 (0.024)
S_3 (beta)	**0.773 (0.022)**	0.653 (0.043)	0.696 (0.039)
S_4 (volume)	**0.641 (0.035)**	0.517 (0.039)	0.436 (0.036)
S_5 (frequency)	**0.589 (0.035)**	0.469 (0.037)	0.420 (0.032)

Tab. 5.2.: Comparison of the average AUC for different combinations of weight function and classifiers with 50% labelled data. The average and standard deviation are calculated for 100 random partitions. The best performing classifier for each similarity matrix is given in bold.

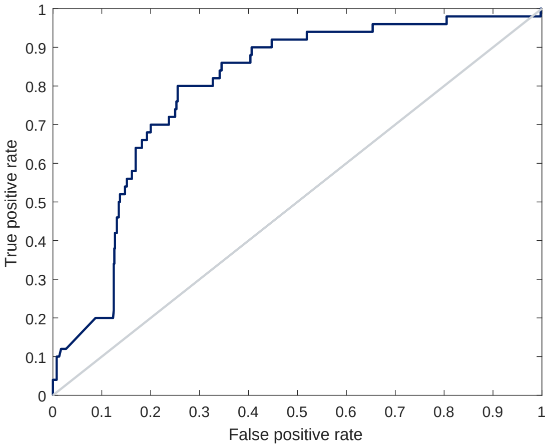


Fig. 5.11.: Rate of classification curve of the RN-beta relational model.

the labels are known. This shows that sufficient calibration is required before accurate class distribution can be constructed.

Table 5.2 shows the statistics for the classifiers at 50% of labelled instances. Besides the averages, these statistics also give an indication of the standard deviation of the AUC. In addition to the best overall performance, the RN also has the lowest standard deviation. Of all combinations of similarity matrices and classifiers, the relational neighbour with a beta-weighted similarity function result in the best performance. With 50% labelled instances we achieve an average AUC of 0.773 with a standard deviation of 0.022.

We further investigate this performance of this relational model. Figure 5.11 shows the ROC curve on the same partition as for the hierarchical (figure 4.3). The corresponding AUC is 0.794, which corresponds to a decent overall classification performance. Before a FPR of 0.12 is reached, the model does not perform well. This can be explained by the large number of disconnected clients, for which the model has no information other than the class prior. The precision and recall suffer from this, having values of 0.115 and 0.200 respectively. This means that of all (10% of the test set) warning signals, 8 out of 9 are false alarms. Furthermore, only 1 out of 5 of the total number of defaults would be detected.

From these results we conclude that the relational model would not suffice as a monitoring model. However, the overall classification performance is promising, especially considering the fact that only relational connections in the data are used. These relational connections are a data source that are currently untapped, which means we can combine it with the traditional data. In the next chapter, we investigate this.

Combined model & Applications

6

In chapter 4, we described models with which we make classification predictions. These models constructed in a traditional manner, namely by estimating predictors based on the given data. In chapter 5, we take a different approach, focussing on relational connections between clients that are implicit in the payment data. Both approaches have shown promise in predictive quality, while making use of different information in the data. Hence, a combination of the models might further improve the predictive performance of the model. This ensemble approach is discussed in section 6.1.1. The low customer effort required in providing transaction information makes it interesting to develop a model which only uses transaction data. Therefore, we develop such a model in section 6.1.2. In section 6.2 we compare the PD estimates of the different (hierarchical) logistic regression models.

6.1 Additional models

6.1.1 Combined model

The approaches in chapter 4 and 5 use the transaction data in a different manner. A combination of these approaches could further improve the classification model. A way to combine the approaches is to include network properties as a new covariate in the hierarchical logistic regression. We examine the following three properties: client connectivity (degree), local weighted clustering coefficient, and the RN PD estimations for the beta similarity matrix. We have chosen the RN classifier in combination with the beta-weighted similarity as this combination has achieved the best performance.

Like all variables in the model, the properties should have an intuitive effect in the model. We have seen that the client connectivity among default clients is lower than for non-defaults. Thus, we expect a negative coefficient for this covariate, reflecting that a low degree should increase the default probability. The local weighted clustering coefficient is the weighted clustering value for a particular node (further details can be found in appendix D.2). Figure 5.8 shows that the clustering among default clients is lower than a randomly selected clients in the network. This could be an indication that default clients are less connected to the entire network. Hence, we expect a negative sign for the coefficient corresponding to the local WCC covariate. The interpretation of the RN estimations is clear, a higher RN estimation should increase the PD of the ensemble model.

The network properties are included in the hierarchical logistic regression as follows. First, the data is split into a training, and test set according to the ratio 2:1. Furthermore, the one third of the training set is used as the validation set. Given the training data, the parameters of the beta function are then tuned on this validation set. Based on the optimal parameters,

Measure	Significant	Expected sign	Both
Local node weighted clustering	0.32	1.00	0.32
Degree	0.00	0.40	0.00
RN estimates	1.00	1.00	1.00

Tab. 6.1.: Percentage of the 100 partitions in which the network properties are significant at a 5% level in the combined model.

the similarity matrix and corresponding network are constructed. The properties of the network are included in the hierarchical logistic regression as fixed effects and the model is fitted on the training set (including the validation part). To examine whether the network properties are a significant improvement to the hierarchical logistic regression, we randomly generate 100 partitions of the training and test set and construct the combined model for each network property. For each partition we examine whether the added property is significant and has the expected coefficient sign. Table 6.1 shows the percentage of partitions in which the network property is significant at a 10% confidence level, intuitively correct, and both.

Based on these results we include the RN estimations to the hierarchical logistic regression. We will refer to this combined model as the HLR-RN model. Table 6.3 presents the performance of the HLR-RN for 100 random partitions. The table also includes the results the hierarchical model (without RN) of chapter 4. By comparing the statistics of the HLR and the HLR-RN, we find that the models perform very similar. The AUC, precision, and recall are nearly equal. The averages of the pseudo-R measures are higher, indicating that the combined model has a better fit. However, the standard deviation of both measures has also increased, meaning that this does not hold for every partition. Furthermore, the average pass rates for the HL, standardized Pearson are 100%, whereas Stukel's test is never passed in both models.

6.1.2 Transaction only model

The improvement of combining the network classification with traditional modelling is marginal. In terms of average AUC, an improvement of 0.013 is not a persuasive result in favour of adding the computationally intensive network analysis. An explanation for this is that hierarchical logistic regression model is already refined, achieving good performance. Hence, it is difficult to further improve the model and identify the most undistinguishable defaults.

An other question is whether credit scoring models can rely on only transaction data. In traditional models, external ratings, and client characteristics are important variables. These variables are collected via an external party or annual client reports. Acquiring them therefore requires money and customer effort. In contrast, the transaction data is all available in-house, which means it requires no customer effort and is of low financial costs. Therefore, it is interesting to examine the performance of a transactional only model.

The following transaction-based variables are significant in either the FC or DC model: balance returns, positive shocks, zero transactions, cash volume. Furthermore, the network

Variable	Coefficient	St. dev.	p-value
Intercept	-2.934	0.121	<0.001
Positive shocks	-0.235	0.120	0.051
Zero	0.293	0.057	<0.001
Cash	0.259	0.068	<0.001
RN estimates	0.903	0.122	<0.001

Tab. 6.2.: Statistics for the transactional model. The maximum likelihood estimates of the coefficients, the standard errors, and the p-values are given.

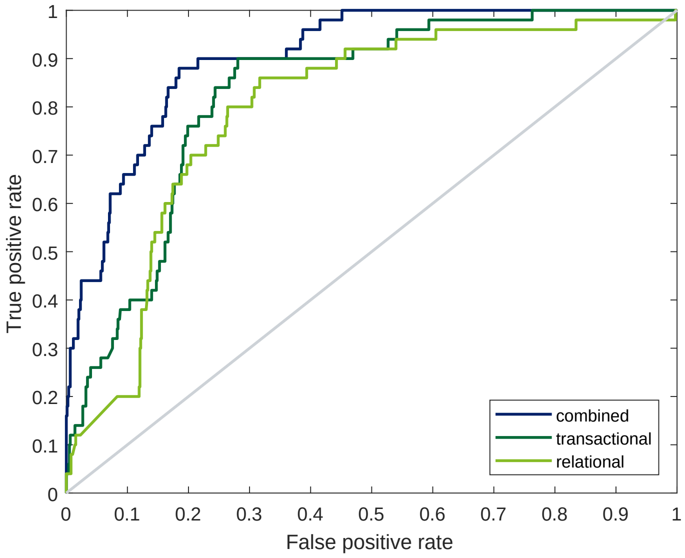


Fig. 6.1.: Rate of classification curve of the relational, transactional, and combined model. The corresponding AUC's are 0.780, 0.804, and 0.903 respectively.

PD predictions has also shown predictive power. Analogous to the strategy in chapter 4, we iteratively develop a logistic regression containing variables on the combined dataset (containing both FC and DC).

After fitting the multivariate model, the balance variable is not significant at a 10% level. The likelihood ratio test supports removal according to the test statistic $LR = 0.025$ and corresponding p-value $p = 0.875$. There are no large (>20%) changes in the coefficient value and no collinearity between the resulting variables. The final transaction based model is presented in table 6.2.

Table 6.3 presents the results of the transaction model for 100 random partitions. For comparison, we have also included the results for the relational model (of chapter 5). Allowing 10% of the total set as the number of warnings, we find a precision of 0.230, and a recall of 0.340. These values indicate that approximately 1/3 of the defaults is detected and 3 out of 4 warnings are false alarms. These are poor statistics for a monitoring model. This performance can be attributed to the absence of the important variable arrear months. If a client has overdue payments, its creditworthiness decreases drastically. The transactional variables are more subtle and therefore defaults are less clearly distinguishable.

Model	Hierarchical	HLR-RN	Relational	Transactional
AUC	0.850 (0.024)	**0.860 (0.025)**	0.768 (0.023)	0.804 (0.026)
Precision	0.353 (0.038)	**0.355 (0.039)**	0.151 (0.033)	0.230 (0.042)
Recall	0.515 (0.049)	**0.519 (0.053)**	0.226 (0.046)	0.341 (0.057)
MacFadden	0.294 (0.021)	**0.390 (0.068)**	-	0.254 (0.093)
Tjur	0.298 (0.022)	**0.385 (0.080)**	-	0.224 (0.102)
HL	1.000	**1.000**	-	0.800
Pearson	1.000	**1.000**	-	1.000
Stukel	0.000	0.000	-	0.000

Tab. 6.3.: Comparison of the average performance measures and GoF pass rates between different models. The statistics are based on 100 random partitions. The best performances are given in bold.

Figure 6.1 shows the ROC curve of the transactional model of the first partition. Furthermore, it also shows the curves of the relational model (corresponding to figure 5.11), and the combined model. The figure illustrates that model performance increases as more data sources are utilized. The relational model uses the similarity between clients in terms of their shared entities, resulting in an AUC of 0.780. The relational model is extended by the transactional model, and includes aggregated transaction characteristics, which capture trends or events that occurred in certain time windows. The model yields an AUC of 0.804. The HLR-RN model is the most comprehensive model and adds client characteristics and loan behaviour to the transactional model, achieving an excellent performance of 0.903.

The transactional model performs worse than the HLR-RN for all evaluation measures. While the performance is not convincing for a monitoring model, the transactional model does has potential for a loan acceptance model. An acceptance model requires a high overall classification performance, but not necessarily high recall and precision on a subset of the portfolio. The AUC of 0.804 indicates a good overall classification performance, and the data can be acquired with low cost or customer effort. We elaborate this further in chapter 7.

6.2 Comparison probability estimates

The output of our (hierarchical) logistic regression models reflects $\mathbb{P}(Y = 1|X)$, i.e. the conditional probability of Y being default given the data. The hierarchical, HLR-RN, and transactional model are all logistic regression models. Table 6.4 gives a comparison of the probability estimates of the three models for two defaulting and two non-defaulting clients in our portfolio.

For all clients, the PD estimate of the HLR-RN is between the estimates of the hierarchical and the transactional model. This behaviour can be explained by the fact that the HLR-RN combines the hierarchical model with the RN approach, which is an important ingredient of the transactional model. For client 1 and 3, the PD estimates of the hierarchical and the combined model are high, given that the average default rate of the portfolio is approximately 7%. In comparison, the PD estimates of the transactional model are lower. It is therefore likely that for the client has 'risky' traditional characteristics, such as a high DB-rating, but does not have financial distress indicators in its transaction behaviour. Client 3 defaulted,

Model	Hierarchical	HLR-RN	Transactional
Client 1 (ND)	0.160	0.134	0.048
Client 2 (ND)	0.010	0.049	0.307
Client 3 (D)	0.187	0.118	0.050
Client 4 (D)	0.042	0.071	0.141

Tab. 6.4.: Comparison of the output of the different (hierarchical) logistic regression models. The estimates of two non-defaulting (ND) and two defaulting (D) client are presented.

which indicates that the traditional drivers are important for correctly classifying this client. In contrast to client 1 and 3, clients 2 and 4 shows the highest PD estimates for the transactional model. Thus, these clients show financial distress in their transactional behaviour, rather than in the traditional variables. Client 4 defaulted, which shows the effectiveness of the transactional variables in this case.

Conclusion & Discussion

7

7.1 Conclusion

In this research we have developed a default classification model for the purpose of credit risk monitoring. We have developed several models that are able to accurately distinguish defaults for the SME portfolio of the Bank. To construct these models, we have used traditional techniques, such as the logistic regression. A more unconventional approach is to identify defaults through relational classification. While the final models can be of practical use, the main achievement of this research lies in the use of transactional data, which is a relatively untapped data source. We contribute to current research in three-fold. First, we introduce measures that allow transaction data to be used in current credit risk models. Second, we develop an innovative transaction based relational classification approach that use the transaction data in a fine-grained manner. Last, we have developed a model that is solely based on transaction data.

To incorporate transaction data into traditional models, we develop mathematical measures for different time windows. With these measures, we are able to capture trends and events in the transaction behaviour of clients. The time windows of these measures are important, and are determined together with experts. Our analysis has shown that zero transactions, positive income shocks, and yearly balance returns are significant variables in classifying defaults. Furthermore, we incorporate and analyse categories reflecting the purpose of transactions. We found that a high expenditure in cash is related to a higher default rate.

Together with client characteristics and loan payment variables, these variables are included in separate logistic regressions for the fixed credit (FC) and dynamic credit (DC) products. Both models have a good overall classification performance with average AUC's of 0.82 and 0.84 respectively. Allowing for a limited number of warnings, the recall of both models is slightly above 0.5, meaning that more than 50% of the defaults are correctly identified. Due to the limited number of defaults in the DC set, 3 out of 4 signals are false alarms. This is almost twice as much as for the FC set. A hierarchical logistic regression combines the two models, yielding results that are decent for a monitoring model, identifying 50% of the defaults with 2 out of 3 false alarms. Due to the low prevalence of defaults in the data, the precision is relatively low. However, we remark that it is 5 times higher than random classification.

Rather than aggregating the transaction information into modelling variables, we also examine the transactions in a non-aggregated manner. We investigate different relational classification methods, which classify clients according to similarity in terms of their transaction behaviour. By using relationships between clients according to the number of shared entities, and using a flexible beta weighing function, the relational neighbour achieves a classification performance of 0.768. Merging this approach with the hierarchical logistic regression improves the performance marginally. We also construct a model which is solely based on transaction data, by combining the relational approach and the (aggregated) transaction variables. This results in a good AUC of 0.804, but a low recall and precision. We conclude that these results are unsatisfactory for a monitoring model. However, the transactional model does have potential as a loan acceptance model, as a good discriminatory power is the most important requirement for loan acceptance.

To summarize, we conclude that transactional data can effectively be used in default classification models. Transactions can be aggregated on a client level into modelling variables, which are easily included in current modelling techniques. Furthermore, the transactions can be analysed on an individual level to construct relation classification models. Combining these two techniques leads to an effective transactional model that shows a good overall classification performance.

7.2 Discussion & recommendations

The transaction data is a data source that is currently untapped by the Bank. Some of the aggregated transaction variables have shown a significant effect in the constructed models. These variables can be added to current models of the Bank without much effort. We therefore recommend the Bank to investigate the added value of the aggregated transaction variables into the models currently in use. We have demonstrated that 1 of the 34 categories (relative expenditure in cash) is a significant variable in some of the models. To further improve the use of the transaction categories, the internal algorithm should be designed purposefully for credit risk in the SME portfolio. We recommend that the number of categories is decreased, and the possible categories are chosen according to an intuition of creditworthiness per category.

Research into an acceptance model is also of interest for the Bank. In general, little information is known about a new applicant. For this reason, the application process requires a lot of customer and employee effort to gather relevant data. Transaction data is easily acquired and can be of value for both the efficiency and accuracy of the loan application process. However, transaction data before one's loan application is limited, which is the reason we focussed on monitoring in this research. A possible solution for this arises from the, recently introduced, Payment Service Directive II (PSDII). Roughly stated, these regulations enable access to account information the client agrees upon, even if it is stored at an other bank. Note, that not all transactions are electronic, meaning that access to all accounts still not necessarily provides the complete financial behaviour of a client. Besides transaction data, the Bank can also consider to include alternative data sources, such as social media information or psychometric tests. [1] These data can also be used in other applications, such as fraud detection. However, it is important to recognize the amount of personal data that can be extracted from transaction data and these alternative data sources. Therefore, there should be a high alertness on the protection of privacy (and legal boundaries) if the data is indeed used.

[1] The use of these data has recently gained publicity as a result of the controversy regarding Cambridge Analytics.

Further steps can be taken into researching relation classification for credit risk in general. Rather than transaction data, similarity scores can be determined as a function of shared client characteristics. Such an approach is used in Macskassy and Provost (2003) to predict blockbusters based on movie characteristics such as: actors, producers, or genre. On a portfolio scale, the relational network can be used to measure the interdependency of regions or industries. By having insight in these dependencies, macro economic effects could be studied. Furthermore, it would be interesting to study the network development over time, and determine the speed at which new clients are connected. Models can be built from this, which could predict the probability of a new client forming certain connections. This would in turn reflect a growth model of the relational network. An important thing to note is that the approach suffers from a lack of interpretability and hence can be classified as 'black box'. Therefore, it is valuable to research information extraction techniques. This can be achieved by identifying entities that are a contributing factor to the similarity between clients. Such an analysis also provides more insight into the inner workings of relational classification and hence is a valuable contribution to literature.

Even with proper information extraction, we remark that relational classification requires additional attention before considering using it in practice. Hurley and Adebayo (2016) elaborate on the development of credit scoring models in the recent decade and discuss the concept of "creditworthiness by association". This concept entails that a credit rating is determined by affiliations rather than one's individual actions. To some extent, this is also present in traditional regression models, in which regional or industry characteristics are used. However, this is more immediate in machine learning models, which may include significant variables that are intuitively unrelated to creditworthiness. Relational classification is a perfect example of this and even completely based on the assumption of creditworthiness by association. While this may have strong classification performance, the ethical question arises whether credit eligibility should indeed be based on one's affiliations.

Parameter estimation techniques

A

A.1 Maximum likelihood

A.1.1 Logistic Regression

In order to find estimates for the logistic regression, we use maximum likelihood estimation.

$$\begin{aligned} Lik(\boldsymbol{\beta}; X) &= \mathbb{P}(Y|\boldsymbol{X}, \boldsymbol{\beta}) && \text{(A.1)} \\ &= \prod_{i=1}^{N} \mathbb{P}(y_i|\boldsymbol{X}, \boldsymbol{\beta}) && \text{(A.2)} \\ &= \prod_{i=1}^{N} \pi_i^{y_i}(1-\pi_i)^{1-y_i} && \text{(A.3)} \end{aligned}$$

Hence the log-likelihood is given by:

$$\begin{aligned} l(\boldsymbol{\beta}, X) &= \log\left(\prod_{i=1}^{N} \mathbb{P}(y_i|\boldsymbol{X}, \boldsymbol{\beta})\right) && \text{(A.4)} \\ &= \sum_i \log\left(\pi_i^{y_i}(1-\pi_i)^{1-y_i}\right) && \text{(A.5)} \\ &= \sum_i y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i) && \text{(A.6)} \\ &= \sum_i y_i \log\left(\frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right) + (1-y_i)\log\left(1-\frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right) && \text{(A.7)} \\ &= \sum_i y_i \log\left(\frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right) + (1-y_i)\log\left(\frac{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}-\frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right) && \text{(A.8)} \\ &= \sum_i y_i \log\left(\frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right) + (1-y_i)\log\left(\frac{1}{1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}\right) && \text{(A.9)} \\ &= \sum_i y_i \log\left(e^{\boldsymbol{x}_i'\boldsymbol{\beta}}\right) - y_i\log\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}\right) - (1-y_i)\log\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}\right) && \text{(A.10)} \\ &= \sum_i y_i\boldsymbol{x}_i'\boldsymbol{\beta} - \log\left(1+e^{\boldsymbol{x}_i'\boldsymbol{\beta}}\right) && \text{(A.11)} \end{aligned}$$

in which $\boldsymbol{x}_i'$ represents the i-th row of the design matrix $\boldsymbol{X}$.

The optimal parameters are found if the log likelihood is maximized. Therefore, we set the derivatives to zero for each entry β_j of the parameter vector. By making use of the substitution rule, we find the following equation for β_j.

$$\frac{\mathrm{d}l(\boldsymbol{\beta}, X)}{\mathrm{d}\beta_j} = \sum_i y_i x_{ij} - \sum_i x_{ij} \frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}}} = 0 \tag{A.12}$$

By solving these equations for all $j = 0, \ldots, M$, the parameters estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_M)$ are determined. There is no closed-form solution to these equations. Hence, we approximate the solution. The most common technique to do this by using the Newton Raphson method, which is described in section A.1.2. Furthermore, we need to verify whether this solution represents a global maximum. To determine whether the solution is a maximum, we consider the second derivative of the log-likelihood function w.r.t. β_k. By making use of the quotient rule, we find the following equations.

$$\frac{\mathrm{d}^2 l(\boldsymbol{\beta}, X)}{\mathrm{d}\beta_j \beta_k} = -\sum_i x_{ij} x_{ik} \frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{(1 + e^{\boldsymbol{x}_i'\boldsymbol{\beta}})^2} \tag{A.13}$$

The solution $\hat{\boldsymbol{\beta}}$ is a maximum if the matrix of these second order partial derivatives is negative definite.

A.1.2 Newton Raphson

The Newton Raphson approach is used to determine the roots of a function f by using the Taylor series approximation. Given an initial starting point x_0 in which f is differentiable at least once, the value of f given x_0 can be written as follows:

$$f(x) \approx f(x_0) + \frac{f'(x_0)}{1!}(x - x_0)^1 \tag{A.14}$$

By setting this equation to zero, we find the following formula for x.

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{A.15}$$

The value for x is the new approximation for the root. The Newton Raphson method is based on using this formula in an iterative manner until there is convergence. We set $x_1 = x$, and use the formula (replacing x_0 by x_1) to determine the new estimate x_2. If the series converges, a solution is found.

We describe the algorithm for finding the parameter estimates $\hat{\beta}$ that fit the data best. Full details can be found in Fox (2008)). To determine the parameter estimates, we have to solve the system of equations consisting of:

$$\frac{\mathrm{d}l(\boldsymbol{\beta}, \boldsymbol{X})}{\mathrm{d}\beta_j} = \sum_i y_i x_{ij} - \sum_i x_{ij} \frac{e^{\boldsymbol{x}'_i \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}'_i \boldsymbol{\beta}}} = 0 \tag{A.16}$$

for $j = 0, \ldots, M$. These equations need to be solved simultaneously.

Let $\boldsymbol{\beta}^{(k)}$ represent the estimates of the k-th step of the algorithm. We use the formula:

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \left(-l''(\boldsymbol{\beta}^{(k)})\right)^{-1} l'(\boldsymbol{\beta}^{(k)}) \tag{A.17}$$

The derivatives can be written as follows:

$$l'(\boldsymbol{\beta}^{(k)}) = \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{p}^{(k)}) \tag{A.18}$$

$$-l''(\boldsymbol{\beta}^{(k)}) = \boldsymbol{X}^T \boldsymbol{V}^{(k)} \boldsymbol{X} \tag{A.19}$$

in which $\boldsymbol{p}_i^{(k)} = \frac{e^{\boldsymbol{x}'_i \boldsymbol{\beta}^{(\boldsymbol{k})}}}{1+e^{\boldsymbol{x}'_i \boldsymbol{\beta}^{(\boldsymbol{k})}}}$ and $V^{(k)} = \text{diag}(\boldsymbol{p}^{(k)}(1 - \boldsymbol{p}^{(k)}))$, i.e. a diagonal matrix that varies per iteration.

By inserting equations A.18 and A.19 into A.17, we find:

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + (\boldsymbol{X}^T \boldsymbol{V}^{(k)} \boldsymbol{X})^{-1} \boldsymbol{X}^T (\boldsymbol{y} - \boldsymbol{p}^{(\boldsymbol{k})}) \tag{A.20}$$

which provides the solution to our system of equations if $\boldsymbol{\beta}^{(k+1)} \approx \boldsymbol{\beta}^{(k)}$ to some desired degree of accuracy.

A.1.3 Hierarchical Logistic Regression

The probability density function for the hierarchical logistic regression is written as follows:

$$f(\boldsymbol{y}_j | \boldsymbol{b}_j, \boldsymbol{\beta}) = \prod_{i=1}^{n_j} \mathbb{P}(y_{ij} = 1 | \boldsymbol{b}_j, \boldsymbol{\beta})^{y_{ij}} \mathbb{P}(y_{ij} = 0 | \boldsymbol{b}_j, \boldsymbol{\beta})^{1 - y_{ij}} \tag{A.21}$$

$$= \prod_{i=1}^{n_j} \pi_{ij}^{y_{ij}} (1 - \pi_{ij})^{(1 - y_{ij})} \tag{A.22}$$

in which n_j denotes the number of observations in cluster j. The vector $\boldsymbol{y}_j$ is a $n_j \times 1$ response vector, and $\boldsymbol{b}_j$ is a $r \times 1$ vector containing the random effects.

The random effects are unobserved. Marginalizing over them yields,

$$L(\boldsymbol{\beta}, \Sigma_{\boldsymbol{b}_j}) = \prod_{j=1}^{J} \int_{-\infty}^{\infty} f(\boldsymbol{y}_j | \boldsymbol{b}_j, \boldsymbol{\beta}) r(\boldsymbol{b}_j, \Sigma_{\boldsymbol{b}_j}) \mathrm{d}\boldsymbol{b}_j \tag{A.23}$$

in which $r(\boldsymbol{b}_j, \Sigma_{\boldsymbol{b}_j})$ is the probability distribution of $\boldsymbol{b}_j$. We assume that $\boldsymbol{b}_j$ follows a multivariate normal distribution with covariance matrix $\Sigma_{\boldsymbol{b}_j}$, i.e.

$$\boldsymbol{b}_j \sim \mathcal{N}\left(\mathbf{0}, \Sigma_{\boldsymbol{b}_j}\right) \tag{A.24}$$

The formula A.23 is analytically intractable. Hence, the likelihood must be estimated in order to find estimates for $\boldsymbol{\beta}$ and $\Sigma_{\boldsymbol{b}_j}$. This can be achieved using Laplace approximations. The idea behind this is to approximate the marginal likelihood with Taylor Series such that the integration can be performed. Others approaches are Gaussian quadrature and Penalized quasi-likelihood approaches, for which the details can be found in McNeish (2016). The benefit of using Laplace approximation is that the approach is relatively fast. A downside is that the performance is dependent on the assumption of an independent cluster structure, i.e. $\Sigma_{\boldsymbol{b}_j}$ is a diagonal matrix.

Evaluation Measures

B

B.1 Classic Pearson test

The Pearson test statistic is defined as follows:

$$X^2 = \sum_{i=1}^{n} (\frac{O_i - E_i}{E_i}) \tag{B.1}$$

In this formula, n denotes the number of cases and E_i and O_i represent the expected and observed proportion of case i.

The classic Pearson test follows a chi-square distribution. We show that this holds for the special situation of binary events and one case. The test statistic can be written as follows:

$$\begin{align} X^2 &= \frac{(O_1 - E_1)^2}{E_1} + \frac{(O_0 - E_0)^2}{E_0} \tag{B.2} \\ &= \frac{(O_1 - E_1)^2}{E_1} + \frac{((n - O_1) + (n - E_1))^2}{n - E_1} \tag{B.3} \\ &= \frac{(n - E_1)(O_1 - E_1)^2}{E_1(n - E_1)} + \frac{E_1(E_1 - O_1)^2}{E_1(n - E_1)} \tag{B.4} \\ &= \frac{n(O_1 - E_1)^2}{E_1(n - E_1)} \tag{B.5} \end{align}$$

in which O_0 and E_0 are the observed and expected proportions of the event 1 and non-event 0, and n is the number of observations. Let p the probability of an event under the null hypothesis. The expected number of events is distributed as $Binomial(n, p)$. Hence, we can write:

$$\begin{align} X^2 &= \frac{n(O_1 - E_1)^2}{E_1(n - E_1)} \tag{B.6} \\ &= \frac{n(O_1 - np)^2}{np(n - np)} \tag{B.7} \\ &= \frac{(O_1 - np)^2}{np(1 - p)} \tag{B.8} \\ &= \left(\frac{O_1 - np}{\sqrt{np(1 - p)}} \right)^2 \tag{B.9} \end{align}$$

For large n the binomial distribution can be approximated by the normal distribution with a mean and variance equal to the first two moments of the binomial distribution. Thus,

$$E_1 \sim Binomial(n, p) \approx \mathcal{N}(np, np(1-p)) \tag{B.10}$$

From this we can conclude that the test statistic as given in equation B.9 follows a chi-squared distribution with one degree of freedom. This derivation lays the foundation for the general case. A formal derivation of the general case can be found in Buonocore and Pirozzi (2014).

B.2 Standardized Pearson test

The classic Pearson test statistic is given by:

$$X^2 = \sum_j \sum_i \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{B.11}$$

in which O_{ij} and E_{ij} are the observed and expected proportions of an event i for the case j.

Let us consider the binary case in which the independent variables are estimated through a logistic regression. We show that the standardized Pearson can be written as follows:

$$X^2 = \sum_i \frac{(y_i - \hat{\pi}_i)}{\hat{\pi}_i(1 - \hat{\pi}_i)} \tag{B.12}$$

in which y_i is the dependent variable, and $\hat{\pi}_i$ the predicted probability that $y_i = 1$ for observation i.

Proof. In the binary case, the classic Pearson test is denoted as follows:

$$X^2 = \sum_j \frac{(O_{0j} - E_{0j})^2}{E_{0j}} + \frac{(O_{1j} - E_{1j})^2}{E_{1j}} \tag{B.13}$$

$$= \sum_j \frac{((N_j - O_{1j}) - (N_j - E_{1j}))^2}{E_{0j}} + \frac{(O_{1j} - E_{1j})^2}{E_{1j}} \tag{B.14}$$

in which N_j denotes the size of case j. We assume that each observation is its unique case, and hence $N_j = 1$. Furthermore, we can write $O_{1j} = y_j$ and $O_{0j} = 1 - y_j$, where y_j is the observed binary value. The predicted values follow from a logistic regression meaning that the expected proportions are binomially distributed with parameters N_j and π_j. Hence, the proportion E_{1g} is equal to $\hat{\pi}_j$ and E_{0j} is equal to $1 - \hat{pi}_j$.

Including this information in the formula yields.

$$X^2 = \sum_j \frac{\left((N_j - O_{1j}) - (N_j - E_{1j})\right)^2}{E_{0j}} + \frac{\left(O_{1j} - E_{1j}\right)^2}{E_{1j}} \tag{B.15}$$

$$= \sum_j \frac{\left((1 - y_j) - (1 - \hat{\pi}_j)\right)^2}{(1 - \hat{\pi}_j)} + \frac{\left(y_j - \hat{\pi}_j\right)^2}{\hat{\pi}_j} \tag{B.16}$$

$$= \sum_j \frac{\left(\hat{\pi}_j - y_j\right)^2}{(1 - \hat{\pi}_j)} + \frac{\left(y_j - \hat{\pi}_j\right)^2}{\hat{\pi}_j} \tag{B.17}$$

$$= \sum_j \frac{\hat{\pi}_j \left(y_j - \hat{\pi}_j\right)^2}{\hat{\pi}_j(1 - \hat{\pi}_j)} + \frac{(1 - \hat{\pi}_j)\left(y_j - \hat{\pi}_j\right)^2}{\hat{\pi}_j(1 - \hat{\pi}_j)} \tag{B.18}$$

$$= \sum_j \frac{\left(y_j - \hat{\pi}_j\right)^2}{\hat{\pi}_j(1 - \hat{\pi}_j)} \tag{B.19}$$

which concludes our proof. □

B.3 Wald's test

The parameter coefficients $\hat{\beta}$ are estimated via the maximum likelihood procedure. To test the significance of an estimated coefficient β_i we use Wald's test. The test statistic for testing if a coefficient is significantly different from zero is calculated as follows:

$$W = \frac{\hat{\beta}_i}{\hat{\sigma}_{\hat{\beta}_i}} \tag{B.20}$$

in which $\hat{\sigma}_{\hat{\beta}_i}$ represents the estimated standard error of the coefficient estimate $\hat{\beta}_i$.

The Wald statistic W is asymptomatically distributed as a standard normal distribution. Furthermore, for the logistic regression the variance of the residuals is related to the mean and therefore we do not need to estimate it. As a result of these two properties, the significance of the coefficient can be determine using the z-score. We test the null hypothesis that the coefficient $\hat{beta}_i$ is zero. The null hypothesis is rejected if:

$$\mathbb{P}(|W| > Z_{\alpha/2}) \leq \alpha \tag{B.21}$$

in which α denotes the critical value, and $Z_\alpha = F^{-1}(\alpha)$ in which F^{-1} denotes the inverse cumulative distribution function of the standard normal distribution. If the test is rejected, we conclude that the estimated coefficient is significantly different from zero.

B.4 Likelihood ratio test

The likelihood ratio test compares two different models and is used in practice to test whether removal of parameters does not yield a significantly worse model. The alternative model represents the full model, and the null model represents the reduced model (with removed parameters). These models are fitted on the data according to the maximum likelihood procedure and the likelihoods are given by L_{null} and L_{alt}. The test statistic is as follows:

$$D = -2 \ln \left(\frac{L_{null}}{L_{alt}} \right) \tag{B.22}$$

$$= 2 \ln \left(\frac{L_{alt}}{L_{null}} \right) \tag{B.23}$$

$$= 2 \left(\ln(L_{alt}) - \ln(L_{null}) \right) \tag{B.24}$$

The alternative model (with more parameters) will always have a likelihood that is at least as high as the null model. We test whether it is also significantly higher. The test statistic D approximately follows a chi-squared distribution with degrees of freedom equal to the number of removed parameters p. We reject the null hypothesis if $\mathbb{P}(D > \chi^2_p(\alpha)) \leq \alpha$. Rejection of the null hypothesis indicates that the null model is significantly worse than the full model. Therefore, removal of the parameters is not supported in case of rejection.

Modelling results

C

C.1 Univariable analysis

Variable	Test statistic	p-value
region	0.854	0.653
industry	**8.426**	**0.015**
size	**33.20**	**<0.001**
DB rating	**49.28**	**<0.001**
liability	0.000	0.995

Tab. C.1.: Classic Pearson test results for the categorical variables of the fixed credit model. The test statistic and p-value are presented. Significant variables ($p < 0.20$) are shown in bold.

Variable	Coefficient	Standard error	p-value
volume	**-0.083**	**0.053**	**0.113**
frequency	**-0.084**	**0.048**	**0.080**
balance (1 year)	-0.055	0.048	0.251
zero trans. (3 month)	**0.633**	**0.091**	**<0.001**
sd returns	0.028	0.048	0.559
sd incoming	-0.049	0.055	0.364
sd outgoing	**-0.081**	**0.048**	**0.095**
arrear months	**3.808**	**0.249**	**<0.001**
pos. shocks	**-0.118**	**0.048**	**0.015**
neg. shocks	-0.054	0.048	0.256
cat. cash	0.039	0.049	0.431
cat. mortgage	**0.110**	**0.048**	**0.022**
cat. bank costs	0.040	0.049	0.420
cat. gambling	0.047	0.053	0.371
cat. charities	-0.016	0.050	0.747

Tab. C.2.: Statistics of the univariable logistic regressions for the continuous variables of the fixed credit model. Significant variables ($p < 0.20$) are shown in bold.

Variable	Test statistic	p-value
region	**3.071**	**0.079**
industry	**3.843**	**0.146**
size	2.338	0.505
DB rating	**34.34**	**<0.001**
liability	0.730	0.393

Tab. C.3.: Classic Pearson test results for the categorical variables of the dynamic credit model. The test statistic and p-value are presented. Significant variables ($p < 0.20$) are shown in bold.

Variable	Coefficient	Standard deviation	p-value
volume	-0.022	0.052	0.681
frequency	-0.028	0.052	0.609
balance (1 year)	-0.023	0.052	0.662
zero trans. (3 month)	**0.142**	**0.063**	**0.024**
sd returns	**-0.070**	**0.052**	**0.177**
sd incoming	-0.023	0.052	0.654
sd outgoing	-0.038	0.053	0.476
arrear months	**1.743**	**0.305**	**<0.001**
pos. shocks	**-0.067**	**0.051**	**0.192**
neg. shocks	-0.025	0.052	0.633
cat. cash	**0.093**	**0.056**	**0.093**
cat. mortgage	0.031	0.052	0.550
cat. bank costs	0.058	0.053	0.271
cat. gambling	0.042	0.053	0.436
cat. charities	-0.010	0.052	0.849
limit exceedances	-0.062	0.052	0.232
limit use	**0.195**	**0.052**	**<0.001**

Tab. C.4.: Statistics of the univariable logistic regressions for the continuous variables of the dynamic credit model. Significant variables ($p < 0.20$) are shown in bold.

Network measures

D

This chapter describes the several measures used in the analysis of the network. The definitions are based on Ellens and Kooij (2013), Opsahl and Panzarasa (2009), and Pahn et al. (2013).

D.1 Clustering coefficient

Let us have the graph $G = (V, E)$ with corresponding adjacency matrix A. The average clustering coefficient is calculated by taking the average of the local clustering coefficients c_i. For $i \in V$, the c_i is calculated as follows:

$$c_i = \frac{2}{d_i(d_i - 1)} e_i \tag{D.1}$$

in which d_i represents the degree of node i and e_i the number of edges between neighbours of i. Thus, c_i represents the existing number of edges divided by the total possible number of edges between neighbouring nodes, which is given by $(d_i(d_i - 1))/2$. Using the local clustering coefficients, we calculate the average clustering coefficient C as follows:

$$\begin{align}
C &= \frac{1}{|V|} \sum_{i \in V | d_i > 1} c_i \tag{D.2}\\
&= \frac{1}{|V|} \sum_{i \in V | d_i > 1} \frac{2}{d_i(d_i - 1)} e_i \tag{D.3}\\
&= \frac{1}{|V|} \sum_{i \in V | d_i > 1} \frac{1}{d_i(d_i - 1)} \sum_{j=1}^{n} \sum_{k=1}^{n} a_{ij} a_{jk} a_{ki} \tag{D.4}\\
&= \frac{1}{|V|} \sum_{i \in V | d_i > 1} \frac{1}{d_i(d_i - 1)} (A^3)_{ii} \tag{D.5}
\end{align}$$

in which a_{ij} denotes the i-th row and j-th column of the adjacency matrix.

D.2 Weighted clustering coefficient

The weighted clustering coefficient (WCC) for the weighted graph $G = (V, E)$ is defined by:

$$WCC(G) = \frac{\text{Total weight of closed triplets in } G}{\text{Total weight of triplets in } G} \tag{D.6}$$

The total weight is closed triangles (triples) is calculated as a proportional to the number of unclosed triangles. To determine the value of the WCC, we take the average of the local WCC for every node. Let A be the adjacency matrix of the graph G. The matrix S extends A such that for every one in the adjacency matrix, the edge corresponding weight is given. To calculate the weighted clustering coefficient we use the following formula:

$$wcc(i) = \frac{1}{M_i(d_i - 1)} \sum_{j \in V | a_{ij}=1} \sum_{k \in V | a_{jk}=1} \frac{s_{ij} + s_{ik}}{2} \qquad \forall i | d_i \geq 1 \tag{D.7}$$

in which a_{ij} and s_{ij} denote the entries of row i and column j of the matrices A and M. The value $M_i = \sum_{j \in V} s_{ij}$ represents the importance of the node i and is used as a normalizing constant together with the degree d_i of the node. If the degree of i is smaller than 1, $wcc(i)$ is zero.

Using equation D.7 , the weighted clustering coefficient for the graph G is calculated by:

$$WCC = \frac{1}{|V|} \sum_{i \in V} wcc(i) \tag{D.8}$$

D.3 Betweenness

For the graph $G = (V, E)$, the betweenness of a node $i \in V$ is defined by:

$$b(i) = \frac{2}{(|V| - 1)(|V| - 2)} \sum_{j \neq i \neq k} \frac{\sigma_{jk}(i)}{\sigma_{jk}} \tag{D.9}$$

in which σ_{jk} denotes the total number of shortest paths between j and k, and $\sigma_{jk}(i)$ is the number of those paths that contain i. The node betweenness is scaled according to the total number of paths not including i, such that $b(i) \in [0, 1]$.

To determine the betweenness B of the graph G, we use the formula:

$$B = \frac{1}{|V|} \sum_{i \in V} b(i) \tag{D.10}$$

D.4 Shortest path length

For the graph $G = (V, E)$, the average shortest path length L is defined as follows:

$$L = \frac{2}{|V|(|V| + 1)} \sum_{i \leq j} l_{ij}$$

in which l_{ij} represents the shortest path length between nodes i and j. The value for l_{ij} is defined between 1, representing a direct path, and infinity for unconnected nodes.

D.5 Efficiency

The shortest path length is infinity if there are any unconnected nodes in the graph $G = (V, E)$. To correct for this, we define the efficiency of G as follows:

$$L = \frac{2}{|V|(|V| + 1)} \sum_{i \leq j} \frac{1}{l_{ij}}$$

The value for L lies between 0 and 1, in which 1 represents high efficiency (and therefore short paths). 0 corresponds to low efficiency, i.e. long average path lengths.

Bibliography

Agresti, Alan (2006). “Contingency Tables”. In: *An Introduction to Categorical Data Analysis*. John Wiley & Sons, Inc., pp. 21–64 (cit. on p. 20).

Bialonski, Stephan, Marie-Therese Horstmann, and Klaus Lehnertz (2010). “From brain to earth and climate systems: Small-world interaction networks or not?” In: 20, p. 013134 (cit. on pp. 42, 43).

Brown, Iain and Christophe Mues (2012). “An experimental comparison of classification algorithms for imbalanced credit scoring data sets”. In: *Expert Systems with Applications* 39.3, pp. 3446 –3453 (cit. on p. 2).

Buonocore, Aniello and Enrica Pirozzi (2014). “On the Pearson-Fisher Chi-Squared Theorem”. In: *Applied Mathematical Sciences* 8.134, pp. 6733–6744 (cit. on p. 70).

Cnudde, Sofie De, Julie Moeyersoms, Marija Stankova, et al. (2015). *Who Cares About Your Facebook Friends? Credit Scoring for Microfinance*. Research Paper. University of Antwerp (cit. on pp. 3, 35).

Crook, Jonathan N., David B. Edelman, and Lyn C. Thomas (2007). “Recent developments in consumer credit risk assessment”. In: *European Journal of Operational Research* 183.3, pp. 1447 –1465 (cit. on p. 3).

David, Easley and Kleinberg Jon (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. New York, NY, USA: Cambridge University Press (cit. on p. 45).

Ellens, Wendy and Robert Kooij (2013). “Graph measures and network robustness”. In: (cit. on pp. 45, 75).

Erdös, Paul and Alfred Rényi (1960). “On the evolution of random graphs”. In: *Publ. Math. Inst. Hungary. Acad. Sci.* 5, pp. 17–61 (cit. on p. 42).

Evans, Scott and Lingling Li (2005). “A comparison of goodness of fit tests for the logistic GEE model”. In: *Statistics in Medicine* 24.8, pp. 1245–1261 (cit. on p. 20).

Fox, John (2008). *Applied Regression Analysis and Generalized Linear Models*. Sage (cit. on pp. 17, 67).

Fries, Christian P., Tobias Nigbur, and Norman Seeger (2017). “Displaced relative changes in historical simulation: Application to risk measures of interest rates with phases of negative rates”. In: *Journal of Empirical Finance* 42, pp. 175 –198 (cit. on p. 8).

Gurný, Petr and Martin Gurný (2013). “Comparison of Credit Scoring Models on Probability of Default Estimation for Us Banks”. In: *Prague Economic Papers* 2013.2, pp. 163–181 (cit. on p. 2).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc. (cit. on p. 17).

Hoerl, Arthur E. and Robert W. Kennard (2000). “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. In: *Technometrics* 42.1, pp. 80–86 (cit. on p. 14).

Hosmer, D. W., T. Hosmer, S. Le Cessie, and S. Lemeshow (1997). “A comparison of goodness-of-fit tests for the logistic regression model”. In: *Statistics in Medicine* 16.9, pp. 965–980 (cit. on p. 21).

Hosmer, David and Stanley Lemeshow (2005). “Model-Building Strategies and Methods for Logistic Regression”. In: *Applied Logistic Regression*. John Wiley & Sons, Inc., pp. 91–142 (cit. on pp. 14, 15, 24).

Hosmer, David W. and Stanley Lemeshow (1980). “Goodness of fit tests for the multiple logistic regression model”. In: *Communications in Statistics - Theory and Methods* 9.10, pp. 1043–1069 (cit. on p. 21).

Hu, Bo, Jun Shao, and Mari Palta (2006). “Pseudo-R in logistic regression model”. In: *Statistica Sinica* 16.3, pp. 847–860 (cit. on p. 22).

Huang, Cheng-Lung, Mu-Chen Chen, and Chieh-Jen Wang (2007). “Credit scoring with a data mining approach based on support vector machines”. In: *Expert Systems with Applications* 33.4, pp. 847 –856 (cit. on p. 3).

Hurley, Mikella and Julius Adebayo (2016). “Credit Scoring in the Era of Big Data”. In: *Yale Journal of Law and Technology* 18.1, pp. 148–216 (cit. on p. 63).

Khandani, Amir E., Adlar J. Kim, and Andrew W. Lo (2010). “Consumer credit-risk models via machine-learning algorithms”. In: *Journal of Banking and Finance* 34.1, pp. 2767 –2787 (cit. on p. 3).

L. McFadden, Daniel (1977). “Quantitative Methods for Analyzing Travel Behaviour of Individuals: Some Recent Developments”. In: *Behaviour Travel Modelling*. Ed. by David A. Hensher and Peter R. Stopher. Vol. 13. London: Croom Helm (cit. on p. 22).

Lessmann, Stefan, Bart Baesens, Hsin-Vonn Seow, and Lyn C. Thomas (2015). “Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research”. In: *European Journal of Operational Research* 247.1, pp. 124 –136 (cit. on p. 3).

Levy, Roger (2011). *Probabilistic Models in the Study of Language* (cit. on p. 29).

Louzada, Francisco, Anderson Ara, and Guilherme B. Fernandes (2016). “Classification methods applied to credit scoring: Systematic review and overall comparison”. In: *Surveys in Operations Research and Management Science* 21.2, pp. 117 –134 (cit. on p. 2).

Macskassy, Sofus and Foster Provost (2007). “Classification in Networked Data: A Toolkit and a Univariate Case Study”. In: *Journal of Machine Learning Research* 8, pp. 935–983 (cit. on pp. 3, 48).

Macskassy, Sofus A. and Foster Provost (2003). *A Simple Relational Classifier*. Public Research. New York University (cit. on pp. 47, 63).

Martens, David, Johan Huysmans, Rudy Setiono, Jan Vanthienen, and Bart Baesens (2008). “An Overview of Issues and Application in Credit Scoring”. In: *Rule Extraction from Support Vector Machines*. Ed. by Joachim Diederich. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 33–63 (cit. on p. 3).

Martens, David, Foster Provost, Jessica Clark, and Enric Junqué de Fortuny (2016). “Mining Massive Fine-Grained Behavior Data To Improve Predictive Analytics”. In: *MIS Quarterly* 40, pp. 869–888 (cit. on pp. 3, 35–37).

McCullagh, Peter (1985). “On the Asymptotic Distribution of Pearson’s Statistic in Linear Exponential-Family Models”. In: *International Statistical Review / Revue Internationale de Statistique* 53.1, pp. 61–67 (cit. on p. 20).

McFadden, D. (1973). “Conditional Logit Analysis of Qualitative Choice Behaviour”. In: *Frontiers in Econometrics*. Ed. by P. Zarembka. New York, NY, USA: Academic Press New York, pp. 105–142 (cit. on p. 22).

McNeil, Alexander and Jonathan Wendin (2003). “Generalized linear mixed models in portfolio credit risk modelling”. In: (cit. on pp. 3, 29).

McNeish, Daniel (2016). “Estimation Methods for Mixed Logistic Models with Few Clusters”. In: 51 (cit. on p. 68).

Mickey, Ruth M. and Sander Greenland (1989). “The impact of confounder selection criteria on effect estimation”. In: *American Journal of Epidemiology* 129.1, pp. 125–137 (cit. on p. 15).

Mittblöck, Martina and Michael Schemper (1996). “Explained Variation for Logistic Regression”. In: *Statistics in Medicine* 15.19, pp. 1987–1997 (cit. on p. 22).

Norden, Lars and Martin Weber (2009). “Credit Line Usage, Checking Account Activity, and Default Risk of Bank Borrowers”. In: 23, pp. 3665–3699 (cit. on p. 3).

Opsahl, Tore and Pietro Panzarasa (2009). “Clustering in weighted networks”. In: *Social Networks* 31, pp. 155–163 (cit. on p. 75).

Osius, Gerhard and Dieter Rojek (1992). “Normal Goodness-of-Fit Tests for Multinomial Models with Large Degrees of Freedom”. In: *Journal of the American Statistical Association* 87.420, pp. 1145–1152 (cit. on p. 20).

Pahn, Binh, Kenth Engø-Monsen, and Øystein D. Fjeldstad (2013). “Considering clustering measures: Third ties, means, and triplets”. In: *Social Networks* 35, pp. 300–308 (cit. on pp. 46, 75).

Press, S. James and Sandra Wilson (1978). “Choosing between Logistic Regression and Discriminant Analysis”. In: *Journal of the American Statistical Association* 73.364, pp. 699–705 (cit. on p. 3).

Romānova, Inna, Simon Grima, Jonathan Spiteri, and Marina Kudinska (2018). “The Payment Services Directive 2 and Competitiveness: The Perspective of European Fintech Companies”. In: *European Studies Journal* XXI, pp. 5–24 (cit. on p. 3).

Solé, Ricard V. and Sergi Valverde (2004). “Information Theory of Complex Networks: On Evolution and Architectural Constraints”. In: *Complex Networks*. Ed. by Eli Ben-Naim, Hans Frauenfelder, and Zoltan Toroczkai. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 189–207 (cit. on pp. 42, 44).

Squartini, Tiziano and Diego Garlaschelli (2011). “Analytical maximum-likelihood method to detect patterns in real networks”. In: *New Journal of Physics* 13.8 (cit. on p. 42).

Stukel, Therese A. (1988). “Generalized Logistic Models”. In: *Journal of the American Statistical Association* 83.402, pp. 426–431 (cit. on p. 21).

Thuraisingham, Bhavani, Satyen Abrol, Raymond Heatherly, et al. (2016). *Analyzing and Securing Social Networks*. Boston, MA, USA: Auerbach Publications (cit. on p. 48).

Tibshirani, Robert (1996). “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society; Series B (Methodological)* 58.1, pp. 267–288 (cit. on p. 14).

Tjur, Tue (2009). “Coefficients of Determination in Logistic Regression Models-A New Proposal: The Coefficient of Discrimination”. In: *The American Statistician* 63.4, pp. 366–372 (cit. on p. 22).

Toback, Ellen, Tony Bellotti, Julie Moeyersoms, Marija Stankova, and David Martens (2017). “Bankruptcy prediction for SMEs using relational data”. In: *Decision Support Systems* 102, pp. 69–81 (cit. on pp. 3, 48).

Tobback, Ellen and David Martens (2017). *Retail Credit Scoring Using Fine-grained Payment Data*. Research Paper. University of Antwerp (cit. on pp. 3, 35, 36, 48).

Tumminello, Michele, Salvatore Miccichè, Fabrizio Lillo, Jyrki Piilo, and Rosario N. Mantegna (2011). “Statistically Validated Networks in Bipartite Complex Systems”. In: *PLOS ONE* 6.3, pp. 1–11 (cit. on p. 40).

West, David (2000). “Neural network credit scoring models”. In: *Computers and Operations Research* 27.11, pp. 1131 –1152 (cit. on p. 3).

Wilson, Nicholas, Barbara Summers, and Robert Hope (2000). “Using Payment Behaviour Data for Credit Risk Modelling”. In: *International Journal of the Economics of Business* 7.3, pp. 333–346 (cit. on p. 3).

List of Figures

List of Tables