Credit Scoring Prediction using Graph Features

Master Thesis

Lorena Poenaru-Olaru

TUDelft

Credit Scoring Prediction using Graph Features

Master Thesis

by

Lorena Poenaru-Olaru

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on the 8th of July 2020 at 11:00

Student number:	4824881	
Thesis committee:	Associate Prof. dr. Pablo Cesar,	TU Delft, (Chair)
	Assistant Prof. dr. Huijuan. Wang,	TU Delft (Supervisor)
	Assistant Prof. dr. Christoph Lofi,	TU Delft
	Director Data Science. dr. Judith Redi,	GeoPhy (Exact Supervisor)

Disclaimer

The information made available by Exact for this research is provided for use of this research only and under strict confidentiality.

Acknowledgements

This project conducted in collaboration with Exact concludes my 2 years experience as a student at TU Delft. I wish to express my sincere thanks to my supervisors Dr. Judith Redi and Dr. Huijuan Wang for guiding me during this process and indirectly teaching me how to view a problem in a scientific manner. I would like to express special gratitude for Arthur Hovanesyan, who explained to me the key concepts of the data and provided me with meaningful feedback on my approaches. I also want to thank everyone from the Exact Data Science team for their constant moral support and the wonderful working environment.

I am most grateful for my parents supported me both financially and morally during these two years of Masters. Without your help, this experience would have been not possible. I would like to express my gratitude to Şerban, who always encouraged me and reminded me of my potential. Lastly, I want to thank all my friends for their constant support.

Abstract

Small and medium enterprises (SMEs) bring a significant contribution to each country's economy, ensuring both a high employment rate and financial prosperity. Despite their essential role, these type of companies presents a higher vulnerability to default than large corporates. The default event implies that the SME could not properly reimburse the money owned to its suppliers. If their default could be forecast, experts could adopt measures in order to prevent it or its impactful consequences. For this reason, developing a credit scoring model which is able to predict which SMEs are endangered of default is crucial.

There are plenty of credit scoring prediction models available in the literature. However, most of them are only relying on the financial status of one company. In this work, we are presenting a novel method for credit scoring prediction which does not only take into account the SMEs' financial situation, but also their position and role within a transactional network. A transactional network is a graph, whereby the nodes are represented by SMEs and the edges show that between two nodes there should be at least one transaction.

In our work, we highlight the limitations that traditional models face and provide an alternative to overcome them. Furthermore, our findings show that combining network features with financial features could lead to a more accurate prediction and increase the robustness of the model. For this reason, we believe that the transactional network carries significant insights and could be a meaningful addition to financial based credit scoring prediction models.

Contents

List of Figures

List of Tables

Acronyms

SME(s)

Small and Medium Enterprise(s)

ROS

Random Over-Sampling

RUS

Random Under-Sampling

SMOTE

Synthetic Minority Oversampling Technique

EOL

Exact Online

API

Application Programming Interface

KVK

Chamber of Commerce

ASPN

Average Shortest Path of the entire Network

ASPD

Average Shortest Path of the Defaulted Nodes

ASPND

Average Shortest Path of the Non-Defaulted

EBITDA

Earnings Before Interest, Taxes, Depreciation and Amortization

EBIT

Earnings Before Interest and Taxes

RCSFI

Reference Classification System of Financial Information

1 Introduction

Econometrics is a research field which studies methodologies that could be used in order to gather insights from financial data. Its outcomes are usually employed by governments and businesses in their decision-making processes. *Credit scoring* is a research sub-field derived from econometrics which investigates novel methods of judging the creditworthiness of a person or enterprise. This term was initially introduced by banks as a method to assess the likelihood of a borrower to pay back its loans. In this context, the borrower can be either a person or an organization, such as a corporate or a small or medium enterprise (SME). Special attention is usually paid to SMEs due to the fact that their financial stability is fragile compared to large corporates. For this reason, the chance that they are defaulting without being able to refund the borrowed amount is considerably high. Moreover, the term credit scoring was further used with a different connotation in literature, namely to predict the default of a company without being directly linked to a loan.

In econometrics, especially in the *financial contagion* field, researchers are studying the effect of one SME's default on other SMEs. This situation is similar to the bank credit scoring, considering that if an SME defaults it cannot reimburse its debts to its counterparties, which could lead to a significant impact on their financial stability. Additional insights into the SMEs financial status could be gathered by analyzing the interconnections between these organizations. This could be achieved by employing a transactional network, whereby the nodes are represented by companies and the links show that there should be at least one transaction between them. In this project, we are also referring to the transactional network as a business network. In order to extract meaningful insights out of the transactional graph, we employ notions from the *complex networks* research field. Thus we are expecting an improvement into the prediction accuracy by integrating network based features into the traditional credit scoring models.

1.1. Problem Statement

Small and medium enterprises play a key role in the Dutch economy. They are contributing not only by generating 61.8% of the overall value of the country but also by maintaining a high employment rate[1] (64.2% of the total employment). Furthermore, according to Chong et. al. [19], they act as important suppliers for large corporates, ensuring in this way the country's product exports and, thereby, the economical growth.

Although considered the backbone of the economy, SMEs present a higher exposure rate to default than large corporates. The main reason for this is the fact that they are more vulnerable to economic change. Thus, their financial stability is considered weak.

Given the above, developing a credit scoring prediction model for SMEs in the Netherlands is crucial in order to take appropriate measures that could prevent the eventual default of the organization. In this paper we want to address this problem and, thereby, develop a model which is able to predict which Dutch SMEs are endangered.

We tackle the credit scoring prediction problem as a binary classification problem of SMEs into defaulted

[1] Small Business Act for Europe (SBA) Fact Sheet - Netherlands

and non-defaulted. Our goal is to adapt this model such that at the end of each year it is able to predict the company's status for the following year. Furthermore, besides financial information, we also take into consideration the position and importance of the SMEs in the transactional network. We believe that this aspect is meaningful and can facilitate determining whether one company is exposed to the default event.

In essence, in this project, we are presenting a novel Dutch credit scoring prediction model for SMEs which relies not only on learning the financial behaviour, but also the location and importance of a company in a real-world transactional network.

1.2. State of the Art Solutions

A plethora of models was proposed in literature when it comes to credit scoring prediction. We aim to further provide extensive motivation for our literature selection criteria and the limitations of the state of the art solutions.

In this study we only considered models which were created for SMEs. Our choice is motivated by the fact that the financial stability between SMEs and large corporates is significantly different and, thereby, a distinction between the two needs to be made [6].

Another selection criteria was the model's adaptability to the economy of different countries. Thus, we only considered models which were applied for SMEs from various geographical regions.

Following the two aforementioned criteria, we considered as the state-of-the-art model the credit scoring prediction model developed by Altman et al. [6] which was tested on SMEs from the US market. This model is further referred to in the report as the *Altman model*. It proved to be efficient when reproduced by other researchers from other countries. This model only relies on financial features, which are ubiquitous features in credit scoring prediction.

However, the financial features are not always sufficiently informative when predicting default. The authors of [5] claimed that *the exposure to default* also influences the credit scoring prediction. The exposure to default refers to the likelihood of a company's default given the default of its counterparties. However, not enough attention was paid on this factor within literature. Thus, it would be interesting to observe the performance of a credit scoring model which also incorporates knowledge about the partnerships between companies.

This limitation was overcome by Misheva et al. in [41] where they employ both financial features and graph features to cover the exposure to default factor. Their research proves the fact that graph features could be useful in improving the accuracy of the model. However, the main constraint of their work is the fact that they used a synthetic graph of Italian SMEs. The authors, however, mention in the future work the fact that the model needs to be tested on a real transactional network. Considering that this type of data is extremely sensitive, only companies have access to it. Thus, an substantial limitation in terms of credit scoring prediction using graph features is the inaccessibility to a real-world SMEs network.

1.3. Research Question

We first need to mention that after conducting the literature study we observed that most credit scoring models are relying on financial features. For this reason, we are referring to credit scoring models based on financial features as *traditional models*.

Taking into consideration our beliefs regarding the value of the business network information, during this study we aim to answer the following research question:

- *What is the impact of adding network based features on traditional credit scoring models in terms of effectiveness?*

In this paper we target two types of network features: *network centrality based features* (network topological properties features) and *graph embedding features*. The main question can be further split into the following sub-questions:

- *What is the impact of adding network centrality based features on traditional credit scoring models in terms of effectiveness?*
- *What is the impact of adding graph embedding based features on traditional credit scoring models in terms of effectiveness?*

- *What is the impact of adding combined network centrality based features and graph embedding features on traditional credit scoring models in terms of effectiveness?*

1.4. Contribution

As previously mentioned, the state-of-the-art credit scoring models need to be reproduced for each country. Thus, our first contribution was reproducing the Altman model for Dutch SMEs. By doing so we discovered one impactful limitation of this model caused by missing data. We are providing an in-depth explanation into these constraints in Section 5.

The Altman model was reproduced with the purpose of being used as a baseline. However, considering the constraints that it had on our data, a new financial model needed to be developed and set as the baseline. This leads to our second contribution, namely the *financial coefficients model* which is also presented in Section 5. The financial coefficients model is composed of only financial features and was an essential step in answering our research question. What we want to achieve is determining whether graph features bring a significant improvement when combined with financial features.

One important aspect that we need to mention is the fact that we did not only rely on the state-of-the-art solutions, but we performed a systematic analysis of the problems that could occur when predicting credit scoring. One general issue with this type of models is the high class imbalance between the two classes, defaulted and non-defaulted. We, indeed, used the solutions from the state of the art to handle this challenge, but we also reviewed literature, came up with other approaches and critically analyzed them in terms of their impact on our data. Thereby, we carefully examined each step required to develop the model and provide clear motivation into our decisions.

The last, but most important contribution of ours is developing a credit scoring model in which we take into account both financial and graph features. We call this model the *hybrid model* and we provide an extensive explanation into how the two types of features are integrated and which graph features are employed. Our collaboration with Exact proves to be crucial for this analysis as we have access to a financial network in which the nodes are Dutch SMEs and the edges represent the partnership between two companies. For this reason, we exceed the constraints that Misheva et. al. [41] encountered and we tested the benefit of graph features on a real-world transactional network. Furthermore, we extended their research by not only exploiting network topological properties as features but also graph embedding features.

1.5. Report Structure

This report is organized as follows. In Section 2 we present some essential background information regarding the business terminology that we are using further. Section 3 shows the literature study that we conducted in order to properly understand the credit scoring research field, as well as the complex networks. Details regarding the data that we are using in our model are provided in Section 4. A comprehensive explanation regarding the 3 models that we developed is given in Section 5. We show the results of our experiments in Section 6 and we provide a reflection on our work and outcome in Section 7. In Section 8 we conclude our work and create a summary of the conducted research. Lastly, we show possible research opportunities derived from the limitations of our model in Section 9.

2 Background

We start the project by presenting some essential background information regarding the researched domain and its specific terminologies. Thus, this chapter contains a detailed explanation of the following items: **SMEs**, **credit scoring**, **defaulted and non-defaulted**.

2.0.1. Small and Medium Enterprises (SMEs)

There are multiple definitions for SMEs across the different regions of the world. However, considering that our data is composed of Dutch SMEs, in this study we are considering the latest definition given by the European Union (2003)[1].

SMEs are, thereby, companies that have less than €50 milion sales or the number of employees less than 250.

2.0.2. Credit Scoring

The term **credit scoring** was initially used by financial institutions, such as banks, and lenders. In the process of lending money, there are situations in which the consumer is not able to repay its debt. This usually results in significant financial losses for the bank. In order to minimize the losses, the bank needed a system that would automatically indicate the creditworthiness of the borrower. Thus, the credit scoring was created and its purpose is to determine the likelihood of a borrower to repay the loan.

In our research we will use the term credit scoring in order to refer to an SME's probability to pay back its full debts to its suppliers.

2.0.3. Defaulted and Non-Defaulted

Determining credit scoring can also be seen as a binary classification problem. In this case, the challenge that rises is how to distinguish between **defaulted companies** and **non-defaulted** companies. According to Levratto [34], the concept of defaulted company is defined differently within literature. Some authors are considering that defaulted is the equivalent of bankrupt, while other authors are correlating the default status with delays in payments (with respect to Basel II criteria [1]). Levratto, thereby, observed that these interpretations of the *default* concept came from both a juridical and an economical point of view.

In our study we combine both perspectives. We consider a company in the default state if it was declared bankrupt or if the court needed to be involved in order to minimize the amount of money that the company owed due to their current financial status. The main reason for this approach was the fact that in the two aforementioned cases the reimbursement is lowered, thus the supplier would not be able to receive its full payment.

[1] Commission Recommendation 96/280/EC of April 3, 1996, updated in 2003/361/EC of May 6, 2003, enacted from January 1, 2005

3 Related Work

This chapter aims to give a detailed explanation into the literature that has been reviewed in order to conduct this study. We start by analyzing the **credit scoring prediction** literature in a chronological manner. Thus, we firstly introduce the *traditional credit scoring prediction models*. Thereafter, we investigate modern techniques of predicting credit scoring, which imply the use of *machine learning*.
In order to clearly combine the network information with traditional credit scoring, we initially investigate the impact of *complex networks in economics*. We continue by presenting a *credit scoring model which incorporates graph features* and its strong limitations that we aim to overcome in this study. In the last two parts of our review, we investigate the two types of graph features, *network centrality features* and *network embedding features*.

3.1. Credit Scoring Prediction

3.1.1. Traditional Techniques for Credit Scoring Prediction

The most well known and cited credit scoring models were developed by Beaver et al. (1966) [55] and Altman et al. (1968) [4] using an univariate model and a multiple discriminant analysis, respectively. The models were relying on a set of financial ratios which were calculated for each company. Their outputs consisted of a score that could be used in order to determine whether a company is financially stable, close to bankruptcy or in the gray zone. The gray zone is an uncertainty interval in which a company could not be properly classifier into one of the two other classes.
Although the two aforementioned methods were the most popular, in [6] Altman et al. underlines the importance of the company size when determining credit scoring. Thus, the models for SMEs need to be different than the ones for large corporates. In [6] the authors are also presenting a new logistic regression credit scoring model specially adapted for SMEs. As our data-set is composed solely of SMEs from the Netherlands, we only considered literature which presented models that were specially designed for SMEs.
The Altman model became very popular due to its high performance when detecting default [8]. As specified by Balcaen [9], plenty of researchers are using it as a baseline to their models. For instance, the Altman model was applied to predict credit scoring of SMEs from Pakistan [31], UK [27], Hungary [10] and Italy [7]. For this reason, we also considered testing the Altman model on our data-set.

3.1.2. Machine Learning Techniques for Credit Scoring Prediction

The current credit scoring models employed machine learning techniques due to the huge growth of this field. We discovered that these models are strongly affected by the issue of class imbalance. Thus, as we previously mentioned, we are conducting a systematic review, which also covers the most commonly used techniques for the problems that could occur while developing these models. Furthermore, we also research which classifiers were employed for both predicting credit scoring and classifying between two highly imbalanced classes.

Features

Financial features significantly contribute to the problem of credit scoring prediction. The most employed features in the literature were sets of *financial ratios*. In [4] Altman et al. defines 5 categories of financial features: *liquidity*, *leverage*, *coverage*, *profitability* and *activity*.

Liquidity: Is a company performance indicator, which illustrates how likely it is for a company to pay its debts in time given its assets (without borrowing any other amount).

Leverage: Is a company performance indicator, which refers to the amount that comes from loans in order to fulfil its financial commitments.

Coverage: Is a company performance indicator, which highlights whether a company is able to pay its debts based on its current financial status.

Profitability: Is a company performance indicator, which reveals the company's ability to generate earnings after paying off its costs.

Activity: Is a company performance indicator, which shows the company's evolution and health over time.

The process of creating a data-set with financial ratios involved literature review. We, thereby, extracted the financial ratios which were meaningful for different studies of credit scoring prediction for SMEs. We further considered the 5 aforementioned categories as they play an important role in reproducing the Altman model for SME. For this reason, when choosing the appropriate features we filter papers based on whether they divide the used financial ratios into these particular categories. In Table 3.1 we show the most significant financial features from each category, as well as the paper that used them.

		Paper				
	Financial Ratio	Altman [6]	Gupta [28]	Yoshino [58]	Altman [7]	Sophocleous [51]
Liquidity	Cash/Total Assets	x	x	x	x	
	Working Capital/Total Assets	x	x	x	x	x
	Cash/Net Sales	x		x		
	Intangible/Total Assets	x			x	
	Current Ratio		x		x	x
	Quick Ratio		x		x	x
	Intangible/Total Assets					
Leverage	Short Term Debt/Equity Book Value	x	x		x	
	Equity Book Value /Total Liabilities	x		x	x	
	Liabilities/Total Assets	x		x		
	Short Term Debt/Total Debt				x	
	Short Term Debt/Total Assets				x	
	Debt/EBITDA				x	
	Net Debt/EBITDA				x	
	Total Debt/Total Assets				x	x
	Total Liabilities/Tangible Total Assets		x			
	Total Liabilities/Net Worth		x			
	Capital Employment/Total Liabilities		x			
Coverage	Ebitda/Interest Expenses	x	x		x	
	Ebit/Interest Expenses	x		x	x	x
	Interest Expenses/Sales				x	
	Debt Service Coverage Ratio					x
Profitability	Ebit/Sales	x		x		
	Ebitda/Total Assets	x	x		x	
	Net Income/Total Assets	x	x		x	x
	Retained Earnings/Total Assets	x	x	x		x
	Net Income/Sales	x		x		x
	Return of Equit				x	
	Operating Profit/Net Income		x			
	Sales Growth					x
Activity	Sales/Total Assets	x	x		x	x
	Account Payable/Sales	x	x	x	x	
	Account Receivable/Liabilities	x		x	x	
	Working Capital/Sales		x			

Table 3.1: Financial Ratios Literature Review.

Class Imbalance

In practice, the likelihood of having a defaulted company is much lower than the one of having a non-defaulted company. For this reason, we can claim that the former is a rare event compared to the latter. As a consequence, the real-world data-sets contain more samples of non-defaulted than defaulted. This makes classifying companies into defaulted and non-defaulted extremely challenging due to the fact that classifiers tend to predict all samples as non-defaulted (the majority class). Thus, an ubiquitous issue when predicting credit scoring is the high imbalance between the two classes.

The problem of class imbalance is omnipresent in fraud detection [13] or anomaly detection [40]. Given that, several techniques were researched in order to solve it.

In [33] the authors presented the state of the art methods that were used for this particular issue. Their focus was solely on high-class imbalance in big data. The covered techniques were divided into *Data-Level Methods* and *Algorithm-Level Methods*.

Data-Level Methods: These techniques were mostly composed of data sampling approaches such as *Random Over-Sampling* (ROS), *Random Under-Sampling* (RUS) and *Synthetic Minority Over-Sampling Technique* (SMOTE) [16]. ROS involves randomly selecting samples from the minority class and duplicating them. This could, thereby, contribute to increasing the classifier's over-fitting probability. RUS implies randomly selecting samples for the majority class and deleting them. Andrea et al. [47] investigated the benefits of undersampling. The authors observed the fact that this is an efficient technique to handle class imbalance, but they also claim that it can only be used when the *a priori probabilities* are known. SMOTE, however, was created to overcome the limitations of ROS. It is an over-sampling technique which, unlike ROS, synthetically generates samples from the minority class by interpolating the already existing ones that are similar. More et al. [42] highlights the fact that the combination of SMOTE and under-sampling techniques might yield to better results.

Algorithm-Level Methods: This method is implemented at the classifier level, thus there is no data adjustment required. The technique assigns a *higher weight* to the *minority class*. By doing so, when a sample from the minority class is misclassified, the prediction cost is higher than the one resulted from an erroneous classification of a majority class sample. Thereby, the classifier learns that the minority class has a higher importance than the majority class and avoids classifying everything as non-defaulted in order to preserve the accuracy.

Classifiers

In terms of the chosen classifiers, we also firstly performed a literature review. We initially considered classifiers that were used in order to predict credit scoring. Furthermore, given the previously mentioned class imbalance problem, we also believed that employing classifiers used in order to solve this issue might help achieving good results. It needs to be mentioned that also in this step the literature selection criteria of only considering papers that try to address a binary classification problem was preserved.

	Paper				
Classifier	**Shahwan [50]**	Mselmi [43]	Bastos [12]	Brown [14]	Wang [57]
Multiple Discriminant Analysis	x				
Linear Discriminant Analysis				x	
Logistic Regression	x	x		x	x
Naive Bayes					x
AdaBoost			x	x	
Random Forests				x	x
KNN				x	
XGBoost					x

Table 3.2: Classifiers Used in Credit Scoring Literature.

We started by exploring classifiers that were employed in credit scoring prediction. Shahwan et al. [50] tested the performances of 3 classifiers, *multiple discriminant analysis* (MDA) and *logistic regression* (LR). He observed that LR outperformed the traditional MDA. Mselmi et al. [43] also utilized LR in his experiments. Bastos et al. [12] research the whether AdaBoost improves the performance in contrast to LR. The findings of his work stated that Boosting significantly contributes to increasing the model accuracy. According to Brown et al. [14] Random Forest and AdaBoost proved to perform significantly better than Linear Discriminant Analysis (LDA) or LR. They mention that the reason for this is the fact that the first two are more optimized to deal with class imbalance. Wang et al. [57] tested LR, Naive Bayes, SVM, Random Forests and XGBoost and shown that the last one achieved the highest accuracy. In Table 3.2 we show a summary of the classifiers that we found in the literature and the reference to the paper that was using them.

The second part of this study involves investigating classifiers that were hitherto used when performing binary classification on highly imbalance classes. Thus, we explored review papers on this subject and extracted the classifiers that were mentioned. In Table 3.3 we show a summary of our findings.

	Paper				
Classifier	**Leevy [33]**	**Longadge [38]**	Heixiang [29]	**Albashrawi[3]**	**Niu [44]**
Logistic Regression	x	x	x	x	x
Decision tree			x	x	x
Naive Bayes			x	x	
KNN			x	x	x
Random Forest	x			x	x
AdaBoost	x	x	x		
XGBoost	x				x
RUSBoost	x				

Table 3.3: Classifiers Used in Class Imbalance Literature.

We, thereafter, choose to utilize in our study *LR*, *AdaBoost*, *Random Forest* and *XGBoost*. We made this decision based on both popularity of the classifier (number of papers that are using it) for both credit scoring prediction and class imbalance issue. We also took into consideration the reported performance. In addition, Leevy et al. [33] mentions that tree based-classifiers (Random Forest, XGBoost [17] and AdaBoost) show better performance when compared with linear ones. Based on this, we also considered testing another gradient boosting decision tree, namely *LightGBM* [30]. The authors of the new gradient boosting algorithm specify that it improves both the accuracy and the computational time.

3.2. Complex Networks in Economics

The first introduction of graph theory in the field of financial literature was realized by Mantegna et al. [39] in 1999. In their research, the authors highlighted the complex interactions between economical entities that could be mapped by means of graphs. Furthermore, these interactions are the link between two fundamental research fields, namely *economics* and *complex networks*. In this relationship the former is focusing on financial aspects, the latter aims to model the financial connections between different entities.

Ferraro et al. [22] analyzed the importance of bringing valuable information in the field of economy through the means of an innovation network. The innovation network is defined as a network in which the nodes are represented by countries or companies and the edges represent the economical relationships between them. The authors claim that the network could bring significant knowledge exceeding the current economical situation of a particular country or company. Their work was focused on analyzing the network composed of independent organizations from different countries and their business interaction from the Enterprise European Network (EEN). The complex networks research field was employed to further perform this study and gain access to essential details with respect to the economical growth.

An example of representation of such type of unweighted innovation network is depicted in Figure 3.1 and was extracted from [22]. The nodes of the graph represent the countries from EEN and the links the interactions between them. What can also be remarked from this plot is the insights that the network representation could bring. We can observe the fact that some countries such as Germany (DE), Italy (IT) or United Kingdom (UK) have more interactions than others. Thus, the importance of the country in the EEN is highlighted by a network representation, emphasizing the knowledge that could be gathered by using it.

Given its strong contribution to the field of economics, complex networks was further utilized in the *financial contagion* research field which focuses on modelling financial systems. Gai mentions in [23] that the network representation of economics problems could provide substantial insights. The authors also highlight the fact that having one defaulted company could have a tremendous effect on its counterparties (direct neighbours), given the strong financial connection between them. Thus, the innovation network in this case is a graph whereby the nodes represent companies and the links represent whether there are transactions between them.

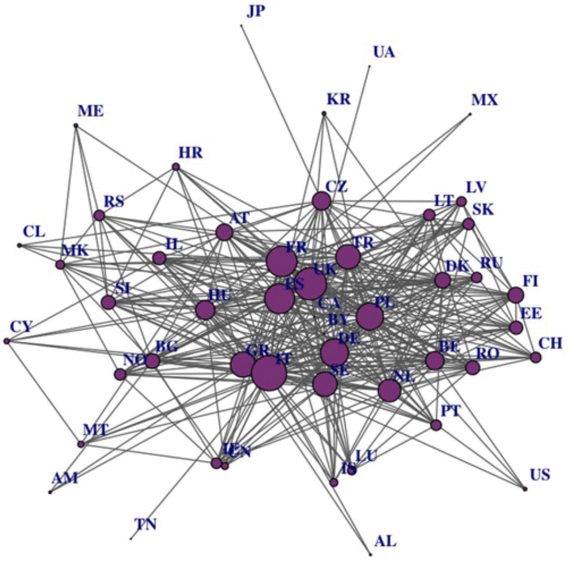


Figure 3.1: EEN countries and their interactions in 2012. [22]

3.3. Credit Scoring Prediction using Graph Features

When it comes to credit scoring prediction literature, Altman et al. [5] mentioned that *exposure to default* can be a significant contributor to a company's loss. They further said that, however, most of the traditional credit scoring models do not incorporate this aspect. Having a network based approach would be extremely beneficial to understand whether a company is exposed to the event of default and, thereby, fill in the credit scoring literature gap.

Misheva et al. [41] contributed to bridging this gap by creating a credit scoring prediction model which incorporates both financial and network based features. Hereby, the network is composed of Italian SMEs from the manufacturing sector. However, due to high limitation when it comes to data access, the authors are not using a real world transactional network, but a synthetic one.

In complex networks the interaction between nodes are mapped using the adjacency matrix. This is a matrix in which the rows and columns are the nodes and its values are represented by whether there is a link between two nodes (for unweighted graphs) and the weight of the link between two nodes (for weighted graph). As this adjacency matrix is not available due to data constraints, the authors found an alternative to it. They substituted it with a correlation matrix that was derived from the available financial data. The motivation behind this approach is the work of Giudici et al. [24] in which the authors demonstrate that conventional network measures derived from the correlation matrix are similar to the ones derived from a real world transactional network.

The synthetic network is generated using an unsupervised technique, which relies on correlations between companies. Thus, the authors of [41] are assessing the correlation between two companies as the correlation between their financial indicators. Each correlation coefficient is a value introduced into the correlation matrix, which, in this particular case, is the same as the adjacency matrix. This matrix is the direct correspondent of a weighted network.

Once the correlation matrix was computed, the authors started building the credit scoring model. They used financial ratios as financial features and they further added the node degree and closeness as network features. In their results, they observed the fact that the network features helped improving the accuracy of the credit scoring prediction model and they concluded that graph derived features can bring significant knowledge.

Although the work of Misheva [41] is a valuable addition to the credit scoring prediction research, assessing such approach using a real-world network is essential. Given the fact that we have access to a transactional network composed of Dutch SMEs, we are researching to what extend can the credit scoring prediction traditional model be improved by the addition of graph features. We further expand their research by not only employing node degree and closeness as graph features. We, additionally, look into other network centrality based features, such as clustering coefficient and eigenvector central-

ity. Furthermore, since graph embeddings have shown significant potential in the in complex networks research, we are also investigating their impact on credit scoring prediction models.
Given the aforementioned, we bridge two significant gaps in the credit scoring prediction research. Firstly, we incorporate network features as a technique to map the interactions between companies to traditional credit scoring models. Lastly, we evaluate the benefits adding 2 types of network features (network centrality features and network embedding features) on a real world network composed on Dutch SMEs.

3.4. Network Centrality Features

According to Bar-Yam [11], a network is heterogeneously organized. This means that some of the nodes have more connections than others. Due to this heterogeneous characteristic of the graph, some nodes are considered more important than others. The importance of one node is assessed by the network centrality metrics. Thereby, these metrics were used extensively when performing network analysis. For instance, Martin relied on network centrality in order to analyze the financial contagion effects [52].
However, according to Rogdrigues [48], the significance of one particular network component is not quantifiable. For this reason, plenty of network centrality metrics were developed in order to assess the importance of the nodes from different perspectives. In his survey, Rodrigues mentioned the most well known network centrality metrics:

- Node Degree Centrality;
- Closeness Centrality;
- Betweeness Centrality;
- Eigenvector Centrality;
- Clustering Coefficient Centrality.

Cong et al. [36] employed the eigenvector centrality, betweenness and closeness and studied their correlation by using the Pearson correlation coefficient. They further introduce a new centrality measure, namely the *degree mass*. However, due to time constraints, this metric was not used in this study. The authors also found a strong correlation between the closeness, betweenness and node degree. Cong et al. [35] demonstrated that the clustering coefficients is strongly correlated with the average shortest path in functional brain networks, which are undirected and unweighted graphs.
The **node degree** refers to the number of edges that are connected to a target node *i*. Thus, this centrality metric considers that one node is influential if it has many connections.
The **closeness centrality** is a metric calculated by summing the shortest paths from a target node *i* to all the other *n-1* nodes in the network. After calculating this sum, the centrality metric is determined by taking the reciprocal of it. This metric considers that an important node should be close in distance to the others.
The **betweeness centrality** is a metric determined through the shortest paths that traverse a target node *i*. Comparing it with the closeness, this metric does not only take into account the length of the shortest path, but also the number of existing shortest paths that pass through node *i*. The betweenness is a metric for which an important node is a node which receives plenty of information.
The **eigenvector centrality** is calculated from the adjancency matrix, which contains information regarding the neighborhood of a target node *i*. This metric evaluate the importance of the node based on whether it is connected to other important nodes in the network.
The **clustering coefficient centrality** is computed by determining the probability of two neighbors of a target node *i* to have a link between them. Thus, this metric judges the importance of a node by how strongly connected its neighborhood community is. According to the authors of [37], a node can be important within its neighborhood community, even though it has a low degree on the overall network.

3.5. Network Embedding Features

The graph representation of the data could be extremely beneficial in gathering insights out of powerful graph analytics. However, effective graph analytics is extremely difficult especially for large graphs.

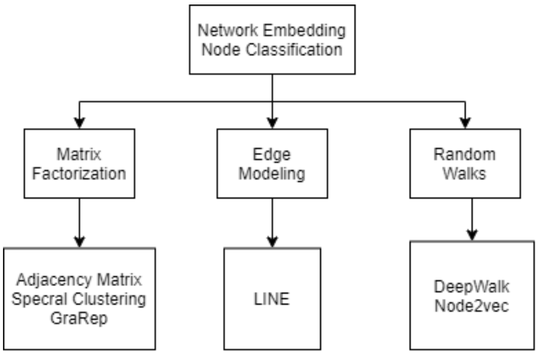


Figure 3.2: Taxonomy of Network Embedding Techniques for Node Classification

The reason for this is that the computational complexity is extremely high and modern machine learning techniques could not be applied given that they need independent vectors represented in a vector space.

Graph embedding, also known as network embedding, is a practical method to represent graph data into low dimensional vectors and concurrently to preserve the graph structure. They are extremely beneficial when solving the problems of the graph representation of the data. Furthermore, they allow parallel and distributing computing, which was a bottleneck for the traditional network data.

Peng et. all [20] created the most recent review of the state-of-the-art network embedding techniques. They can be used for tasks such as node importance, community detection, network distance, link prediction, node classification, network evolution, etc. As our problem is node classification, we will further only focus on techniques related to this particular issue.

There are 3 classes in which network embedding techniques for node classification could be divided: *matrix factorization*, *edge modeling* and *random walks*. In the following part we are presenting only the state-of-the-art methods. They were selected based on the fact that were strongly employed by other papers. In Figure 3.2 we depict the taxonomy of the embedding techniques that we considered for the review.

Traditionally, networks were embedded using **matrix factorization** techniques. The most common was representing a network based on its *adjacency matrix*, which shows each node's direct neighbors. The disadvantage of this method it is extremely sparse, which might have an impact on machine learning technique. In order to address the sparsity issue, the spectral clustering [54] method was proposed. Instead of considering the adjacency matrix, the authors used the Laplacian matrix. Each node was further represented in a vector space using the top d eigenvectors from the normalized matrix as features. GraRep [15] learns the direct neighborhood of the source node, thus it could capture the global structure. However, according to [25] it has the downside of not being able to scale to large networks. These methods have the advantage of capturing the global structure of the network.

Another approach of network embeddings is represented by the **edge modeling** based methods [60]. The most popular technique in this category is LINE [53]. It operates with a 2-phase learning. In the former, there are learned d/2 dimensions of the direct neighbors of the nodes, while in the latter there are learned d/2 dimensions of the 2-hop distance selected nodes from the source one. The learning process involves *Breadth-first Sample*-style simulations. Breadth-first Sampling restricts the neighborhood of a specific node only to the ones which are directly connected to it. The edge modeling based methods are, indeed, only capturing the local structure, but they are efficient and scalable for large networks.

Random Walks approaches are another type of node classification techniques. Deep Walk [45] learning method is relying on simulating random walks in order to explore the neighborhoods of a seed node. It was created with reference to the Skip-Gram model for word representation learning since the nodes distribution from random walks was similar to word distribution in natural language. Node2Vec [25] was developed as an improvement of DeepWalk since the latter was not able to capture connectivity patterns' diversity within the network. Node2Vec is capable of learning nodes community and embed

them properly. It also constrains similar nodes to have similar embeddings. Thereby, Node2Vec is an extension of DeepWalk, but more flexible when it comes to the explored neighborhood.
Given the high computation required by the matrix factorization techniques, we did not considered employing any of the methods in our further research. We focused only on the edge modeling and random walks methods and we examined whether we have clear evidence of choosing one method over the other. It needs to be mentioned that by doing so we assumed that the global structure will not be covered, but we believed that these techniques are a good compromise between performance and computational costs. In [25] the authors of Node2Vec investigated its performance over LINE and DeepWalk. They concluded that Node2Vec achieved a higher performance when reconstructing the graph from the embeddings than the other two.

4

Data

In this chapter we aim to explicitly describe the data that we used in order to perform our research. As the research was conducted together with Exact, some of the data was already available and is described in the first sub-chapter. However, the data-set provided did not contain information regarding to the companies' status (defaulted or non-defaulted). For this reason, we conducted a process of gathering data, which is described in the second sub-chapter.

4.1. Available Data

In this section we are discussing about the data that was provided by Exact. Firstly, we had access to the *network data*, which contains data that was further used in order to create the transactional network. Furthermore, we also had access to the *financial data* of mostly all SMEs that were included into the network.

4.1.1. Network Data

The data provided by Exact consisted of two main data-sets: the **nodes** data-set in which each node is a SME and the **edges** data-sets which represents which companies do business with each other. The companies from the data are clients of Exact, which are using the Exact Online (EOL) software for accountancy and bookkeeping tasks. It needs to be mentioned that according to our definition, two companies do business with each other if there is at least one transaction between them.

The nodes data-set contained initially 289413 records corresponding to 289413 companies. The edges data-set 117615325 records corresponding to 117615325 links in the graph. As these two data-sets were noisy at the beginning we are further explaining the cleaning process.

The first cleaning step was removing all the records from the nodes data-set, which were demos. The EOL software allowed some customers to create two accounts: a demo account and their official account. Although most of them were cleaned afterwards, there were still some records which were left. We considered that the noise caused by these accounts could harm our results, thus we decided to remove them.

The data-sets were derived from an entity resolution algorithm, in which all the entities that were referring to the same company were mapped into a node corresponding to that particular company. This mapping was done by an algorithm and, thereby, can contain noise. Thus, each node in the nodes data-set has two unique attributes: a *NodeID* and a *DivisionCode*. The DivisionCode attribute is the unique identifier of the company in EOL. The NodeID is a unique identifier for each node of the graph. There were situations in which the algorithm erroneously mapped different companies (with different DivisionCode) into the same NodeID. We decided to remove these cases due to the fact that when one of the companies defaulted all the companies with the same NodeID would be considered defaulted.

After the pre-processing step, the final **nodes** data-set contains 228080 records corresponding to SMEs. The edges were filtered, such that they would only contain links between the remaining companies in nodes. The final edges data-set contains 2121361. Therefor, our network is composed of 228080 nodes and 2121361 edges.

4.1.2. Financial Data

In order to thoroughly explain the financial data that Exact provides, we must initially present how the financial data is organized in the Netherlands. We start by explaining what a *Reference Classification System of Financial Information (RCSFI)* or, in Dutch, *Referentie Grootboekschema (RGS)* is.

RCSFI is a standardization of transactions mapping created in order to automate the processes of financial administration. It appeared as a necessity to properly group transactions in order to further develop analysis and provide insights into one company's financial situation. The main reason why RCSFI was created is the fact that SMEs have no standardized format for bookkeeping and accountancy. For this reason, they used their own format and no comparison could be created in order to judge the performance of one company.

Exact has its financial data mapped into this administration convention. Thereby, all transactions are mapped according to RCSFI in specific codes. For example, all the transactions that are referring to the purchase of auxiliary materials required to manufacture a product are mapped into the code WKprKvg.

With the purpose of fulfilling our goal of calculating the *financial ratios* that the Altman model uses we firstly needed to calculate the *financial coefficients* that will facilitate deriving them. Hence, we computed 15 financial coefficients from the RCSFI codes for each of the companies in our data-set: revenue, direct cost, indirect cost, taxes, ebitda (earnings before interest, taxes, depreciation, and amortization), total assets, short term debt, equity book value, cash, interest expenses, retained earnings, current assets, working capital, total liabilities, ebit (earnings before interest and taxes). Some of the financial coefficients required additional coefficients to be calculated beforehand. One example is EBITDA, which relies on calculating the revenue, direct cost and indirect cost in order to be computed. For this reason, in Appendix A we provide both the formulas and the RCSFI codes combinations that are used for each financial coefficient.

4.2. Collected Data

In order to enrich our data-set with the label for each company, we performed a **web-scrapping** technique. We collected the data from the `https://www.faillissementen.com` web-site. We particularly chose this web-site due to the fact that it is the most complete database with company statuses in the Netherlands. Furthermore, it updates its data on a daily basis.

4.2.1. Process Description

Web-scrapping is a commonly used crawling method to gather data which is public but not available to download. This technique becomes extremely useful especially when the web-site does not provide an API from which the data can be extracted.

The pipeline of the collecting data is depicted in Figure 4.1 and more details about each block are provided further.

As it can be seen, the data extraction process takes as input an *URL* in which it needs to be specified the web source. For instance, if we want to collect the 3rd page of data from the aforementioned website, we need to specify as URL:

`https://www.faillissementen.com/insolventies/nederlandse-insolventies/3/`.

The former step of the process is the **scraping** step. The scraper extracts the HTML format data corresponding to the given URL. In our approach, we scraped all the pages which contained data regarding the status of Dutch companies.

The latter step of the process is the **parsing** step. In this part, the HTML data is pre-processed and only essential information is kept. In our case, we only kept information regarding: *KVK*, *Date*, *Name*, *Location*, *SBI*, and *Status*.

The output of the pipeline is the data containing the previously mentioned attributes. We will further refer to this data-set as the *company status* data-set. It needs to be mentioned that the data-set contains the status of Dutch companies from 2005 until 2019.

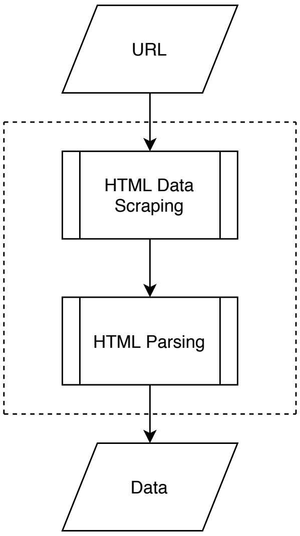


Figure 4.1: Pipeline of Data Extraction with Web-Scrapping

4.2.2. Collected Data Integration

In this subsection we aim to present how we integrated the company status data-set with the other two aforementioned data-sets, namely nodes and financial data-set. For a better interpretation and visualization, we present the data model in Figure 4.2.

As we can observe, both nodes data-set and company status data-set contain the *KVK* attribute.

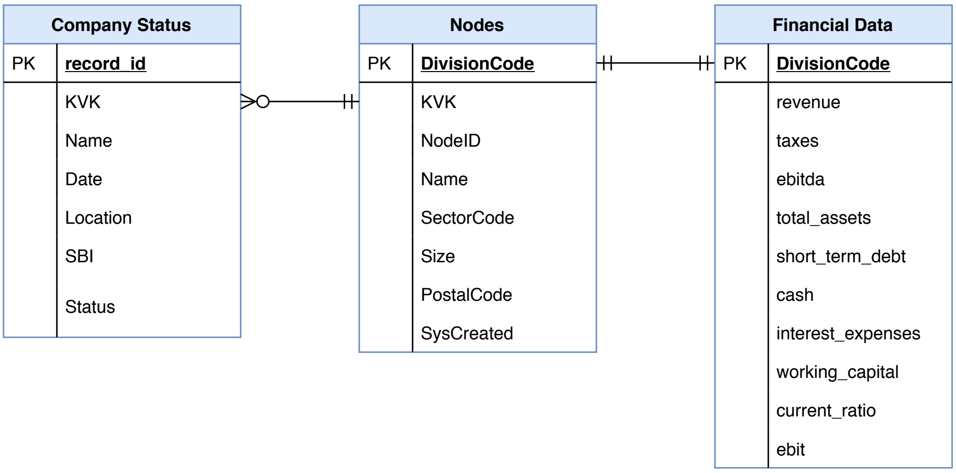


Figure 4.2: Data Model Diagram

The KVK, also known as chamber of commerce, is a unique identifier composed of 8 digits for each company in the Netherlands. Considering that two companies cannot have the same KVK according to the Dutch law, we decided to perform an inner join between company status data-set and nodes data-set in order to link companies with their corresponding status. However, we firstly needed to conduct a cleaning process on both collected data and available data.

The cleaning part involved deleting noisy records from both the nodes data-set and the company status data-set. We define noisy record a record for which the KVK is '00000000' or '12345678' as they are not valid KVKs. We gathered these information by consulting experts in the domain.

Thereafter, we performed an inner join of the company status data-set with the nodes data-set. We, thereby, managed to obtain the company status for 808 companies from our nodes data-set. In the following section, we will extensively explain how do we derive the defaulted and non-defaulted label from the Status attribute. It needs to be mentioned that one company could be reported multiple times under a different status. For instance, one company can be reported in 2014 under a certain status

and in 2015 under a different status.
When it comes to the financial data-set, each coefficient was calculated for each company according to the previous section. For both the nodes data-set and the financial data-set the DivisionCode is included. Thus, the link between the two is represented by this attribute.

4.3. Exploratory Data Analysis

4.3.1. Defining Labels

This subsection aims to properly define the defaulted and non-defaulted labels based on the Status attribute included in the company status data-set.
The status of one company is decided in a juridical manner by the court, which takes into consideration its economical situation. In our crawled data, a company can have one of the following statuses:

- *Suspension of Payment (Surseance)*;
- *Debt Restructuring (Schuldsanering)*;
- *End of Suspension of Payment (Einde Surseance)*;
- *End of Debt Restructuring (Einde Schuldsanering)*;
- *Bankruptcy (Faillissement)*;
- *End of Bankruptcy (Einde Faillissement)*;
- *Distruction of Bankruptcy (Vernietiging Faillissement)*

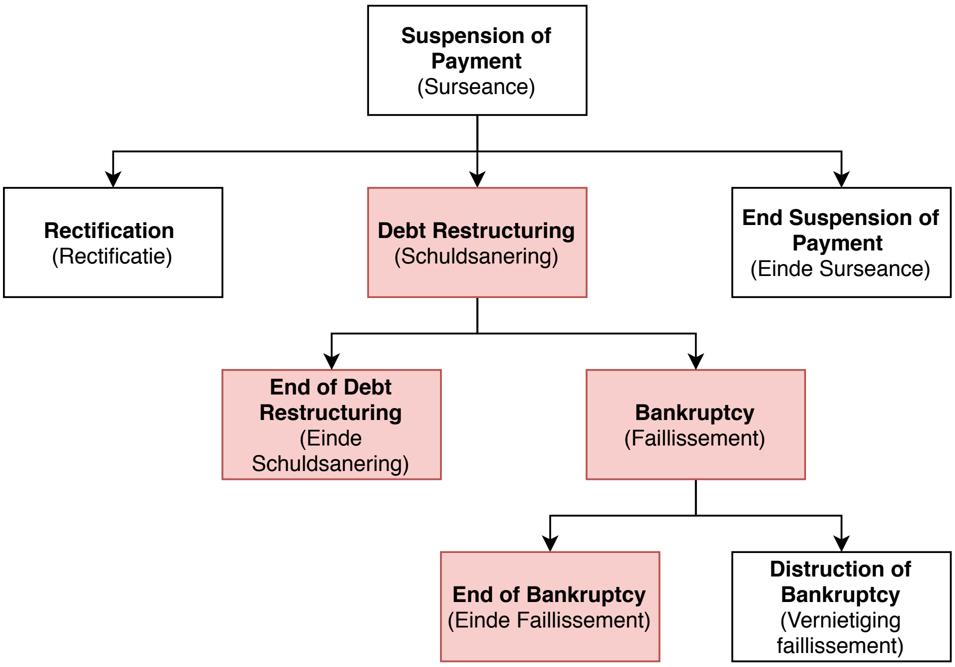

Suspension of Payment (Surseance)
Rectification (Rectificatie)
Debt Restructuring (Schuldsanering)
End Suspension of Payment (Einde Surseance)
End of Debt Restructuring (Einde Schuldsanering)
Bankruptcy (Faillissement)
End of Bankruptcy (Einde Faillissement)
Distruction of Bankruptcy (Vernietiging faillissement)

Figure 4.3: Status Company Chain Crawled Data

In order to properly understand their meaning we consulted experts in the domain. We discovered that the states correspond to the chain depicted in Figure 4.3.
Whenever a company has latency in paying its debts such that the court needs to be involved, it goes into the **suspension of payment** state. If the outcome of the judicial process is the fact that the company was mistakenly accused, the following state is **rectification**. If the company pays all its debts after the court intervention its status changes into **end suspension of payment**. The problem rises when the authorities decide that the company cannot fulfill its financial obligations due to its current economical situation. Thus, the court minimizes the amount of money that the company needs to pay.

For example, if a company owns its supplier €5000 and but cannot afford paying it due to its current financial situation, the court decides to adjust this debt to €3000. At this point, the status of the company becomes **debt restructuring**. From this state, the company could pay the restructured debt and go into the **end of debt restructuring** state or not be able to pay its obligations at all and go into the **bankruptcy** state. If a company was wrongly declared bankrupt (there was a mistake in the juridical process), then its state will become **destruction of bankruptcy**. There are also some rare cases in which a company succeeds in paying its financial debts after being declared bankrupt and, thereby, change its state into **end of bankruptcy**.

We needed to deeply understand these states before determining what are we going to consider a defaulted company or a non-defaulted company. Given the explanations above, we decided that the states which imply that the supplier is not able to receive its full reimbursement are: *debt restructuring*, *end of debt restructuring*, *bankruptcy* and *end of bankruptcy*. The *end of bankruptcy* state was, however, a special one due to the fact that there is no clear evidence that after this case a company could still be active or not. For this reason, we decided to check whether companies classified still make transactions after being declared into this state. We discovered that in majority of the cases they do not have, thus we decided to further link this state with the default event.

It needs to be mentioned that in real data, the chain is not respected. For this reason, the state *destruction of bankruptcy* appears after a bankruptcy was declared and no debt restructuring was initially performed. We, thereby, considered that this state reflects just the situation of a mistakenly formulated accusation. Thus, we assumed that this state is similar to the rectification state.

All things considered, we define as a **defaulted company** a company which was declared in the following states: *debt restructuring*, *end of debt restructuring*, *bankruptcy* and *end of bankruptcy*. We further define a **non-defaulted company** a company which was not declared in any state (was not reported on the aforementioned web-site) or was declared in the following states: *suspension of payment*, *rectification*, *end suspension of payment* and *destruction of bankruptcy*.

4.3.2. Analysis

Defaulted Companies

We began by performing an in-depth analysis regarding the companies which were reported defaulted. Considering that the web-site that we scrapped in order to obtain the labels contains temporal information, we decided that an analysis of the number of reported defaults can be beneficial. Hereby, we show in Figure 4.4 the number of defaulted companies each year. We need to mention that this is not a cumulative sum, but the exact number of companies which defaulted each year.

From Figure 4.4 we can observe that the number of annual defaults is increasing. The reasoning for this behaviour is extremely interesting to explore.

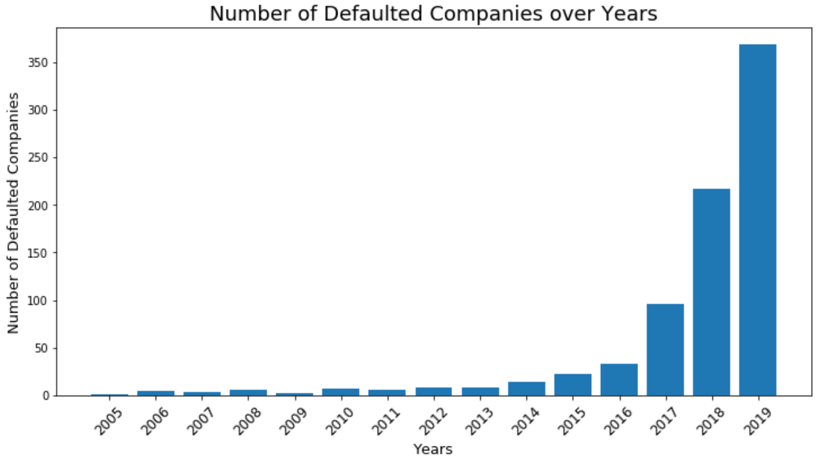


Figure 4.4: Number of Defaulted SMEs over Time.

Given the fact that the web-site was created in 2005, one explanation could be that in the beginning it was not reporting intensively due to its low popularity and trustworthiness. However, until now, the web-site's evolution was substantial, being considered one of the most up to date web-sites when it comes to the financial situation of the Dutch companies.

Another explanation could be given in terms of the EOL platform in which all the SMEs from our data-set are collected. The EOL platform stated its activity in 2005, exactly in the same manner as the web-site. Initially it was composed only of 168 SMEs, but in the course of time, new customers joined, reaching around 228000 SMEs in 2019. The fact that the EOL platform is a growing platform makes it more likely to include more defaulted companies. For this reason we would expect the number of defaulted companies to continuously increase, and, in consequence, the model that we are building needs to deal with this issue. Taking these aspects into consideration, we decided that we should keep the samples of 2019 as test set and use the others in order to train it. Thus, besides the fact that 2019 is the latest year for which we have data, we also take into account the growing trend of the amount of companies which default each year. Furthermore, on thorough analysis of the financial data, we observed that plenty of records corresponding to years 2005 - 2009 were missing. For this reason, we believed that using samples from this interval would have as consequence introducing noise into the model. For this reason, only financial data starting from 2010 was considered accurate enough to be introduced into the model.

Proof of Concept Network

In this project we believe the fact that if one company defaults it is extremely likely to also affect its counterparties from a financial point of view. If the counterparties are financially triggered, there is a chance that they will also affect their direct neighbors. This could lead to a financial contagion (a phenomenon which was previously described) within the network and the most affected nodes are the ones close to the default seed node. For this reason, we hinge our research on the following assumption:

Assumption: *Defaulted nodes are close to each other in the network.*

In order to validate our assumption, we decided to explore the companies from a network point of view. Thus, we used the network, which contains companies that do business with each other. The network's nodes are represented by companies and the network's links show that two nodes do business with each other. We are further referring to this network as the **business graph**. The business graph is an undirected fully connected graph.

In order to validate our assumption, we employed 2 techniques: **defaulted sub-network extraction** and **calculating the distance between nodes**. The first technique aims to find whether the defaulted nodes are connected to each other, thus enforcing the idea that the default of one company has a tremendous effect on its direct neighbors. The second technique implies a more general analysis. We do not consider that only the direct neighbors are the ones affected, but we also believe that the the default of one node has an impact on the ones located at a small distance from it. We are briefly explaining the outcome of these techniques.

Defaulted sub-network extraction is also used by Yoshiyuki et al. [59] to determine whether the bankruptcy event could be considered a spreading process.

Having all companies in the business graph, we extracted only the defaulted ones together with their corresponding edges. We discovered that the sub-network of defaulted nodes is not fully connected, it is actually composed of 642 components. The distribution of the sub-network's components is depicted in Figure 4.5. With the purpose of a better visualization of the differences between bins, we plotted the y-axis using logarithmic scale.

From Figure 4.5 can observe the fact that most of the components are composed of only one node. This means that most of the defaulted companies have non-defaulted direct neighbors. In consequence, there is insufficient evidence that the default event could be modeled by spreading models. However, we can also observe some large components which contain 7, 8 or even 24 defaulted nodes. These components present great interest to us and we decided to further investigate them.

In our analysis we examined 4 components: one component of 24 nodes, 2 components of 8 and one component of 7 nodes. We observed that one component (component 2), which contained 8 nodes, corresponded actually to 8 companies belonging to the same franchise. We, additionally, noticed that all defaulted in exactly the same date. We discovered afterwards that some franchises have the same KVK. Thereby, we decided to remove this component as it could have been mistakenly created while merging by KVK.

Our focus remained solely on the 3 left components (7, 8 and 24 nodes). We depict them in Figure 4.6 on the left side. In the right side of Figure 4.6 we show the same components, but this time coloured based on the year the companies defaulted. We considered that including the time dimension would

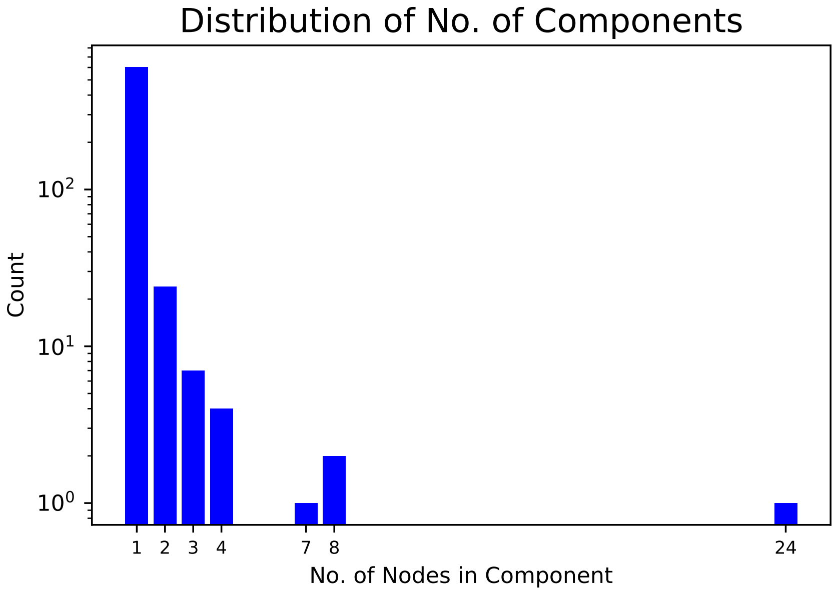


Figure 4.5: Distribution of the number of components in the sub-network of defaulted companies. The Y-Axis is represented using logarithmic scale.

bring supplementary insights into how one node's default influences the others. We, thereby, assumed that if a node defaulted at timestamp *t* and its default affects its neighbors, then its neighbors will default at timestamp *t* or *t+1*. We analyzed the nodes default at a granularity of one year. All things considered, our purpose with this analysis is to check whether the default of one node is responsible for the default of its neighbors.

As we aforementioned, we are considering the granularity of the analysis of one year. In the case of components 1 and 3, most of the nodes defaulted in 2017, 2018, 2019, but there were also nodes which defaulted much earlier, such as 2014. We believe that the default of one node in 2014 would not impact the default of one node in 2017 due to high time difference. Thereby, we removed these nodes from the temporal representation of the component.

From Figure 4.6 we can observe that in the situation of component 1, there is very few evidence that our assumption is valid. However, when looking at components 3 and 4, we observe that the time dimension could provide meaningful insights. We can see that nodes which defaulted in 2017 or 2018 are directly connected with the ones that defaulted in 2018 and 2019, respectively. This strengthens our assumption and it proves that in some cases taking into consideration the relationships between companies when predicting credit scoring might be beneficial.

Calculating the distance between nodes is our method of verifying whether the defaulted nodes are close to each other. We formerly observed that some of them are connected to each other, but we want more insights into whether the non-connected components are closely located within the business graph.

The most common way to do this is to calculate the average shortest path of only defaulted nodes (ASPD) and compare it with the average shortest path of the entire network (ASPN). If the ASPD is smaller than ASPN, then we could say that defaulted nodes are closer to each other in the graph.

One crucial problem of this approach is the fact that our network is extremely large, containing 228080 nodes in total. The employed method for calculating the average shortest path has a complexity of O(n^2), translated into O(228080^2) in our case. Due to this fact, we cannot calculate the ASPN considering the high computational demand.

In order to solve this issue we decided to take a different approach. We calculated the ASPD and the average shortest path between x randomly chosen non-defaulted nodes (ASPND). In this case x represents the number of total defaulted nodes, namely 735. We performed the random selection of the non-defaulted nodes 20 times and took the mean of the 20 resulting ASPND. By doing so, we observed that the ASPD (3.26) is smaller than the mean of the 20 ASPND (3.49), which means that defaulted

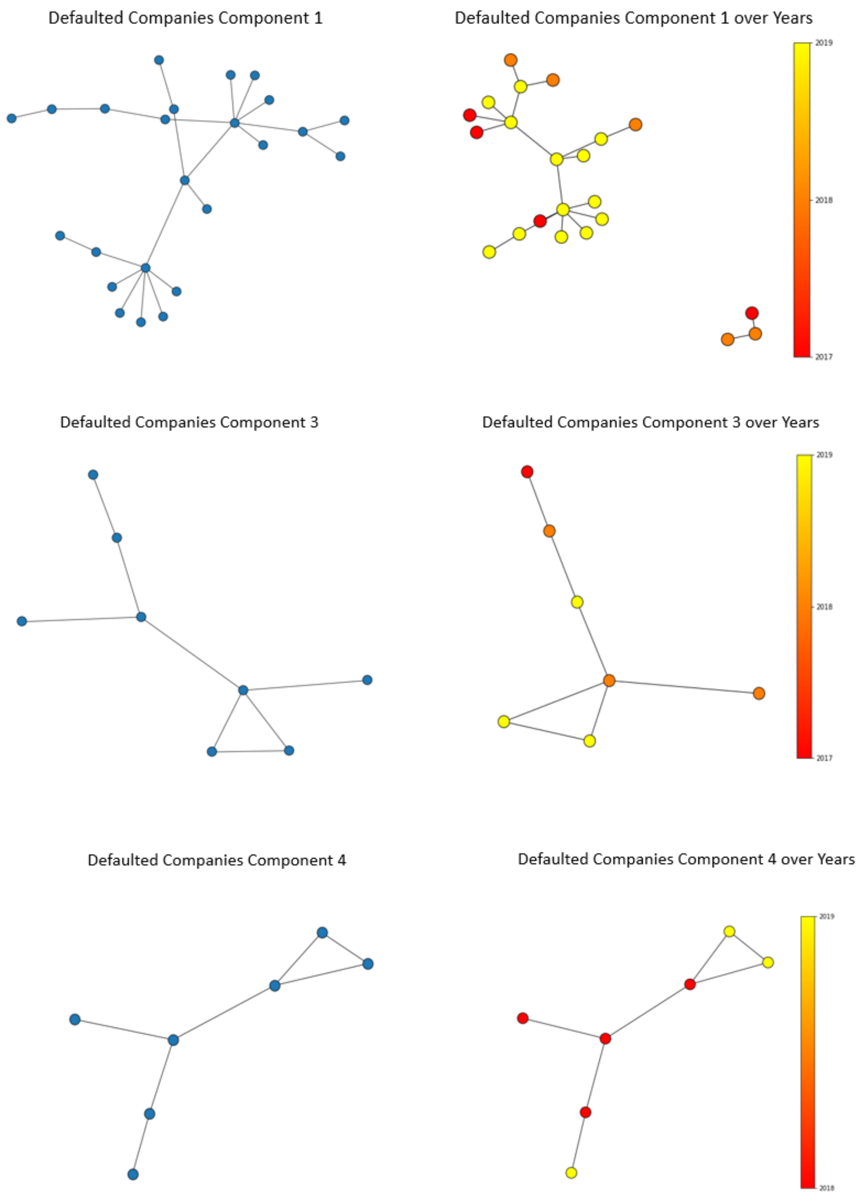

Defaulted Companies Component 1
Defaulted Companies Component 1 over Years
2019
2018
2017
Defaulted Companies Component 3
Defaulted Companies Component 3 over Years
2019
2018
2017
Defaulted Companies Component 4
Defaulted Companies Component 4 over Years
2019
2018

Figure 4.6: Analysis 3 largest components defaulted sub-network. The analysis was done without considering the temporal aspect (left) and including the temporal aspect (right).

nodes are closer to each other in the graph.

Given that our method is dependent on random selection, we needed a statistical test to validate the fact that the difference between ASPD and ASPND is meaningful. We chose to apply a paired difference test which is usually performed in order to judge whether the difference between the means of two populations is statistically significant. In our case the two populations are the shortest paths between the defaulted nodes and the shortest paths between the non-defaulted nodes (for all the 20 cases). We further utilize the *Wilcoxon signed-rank test* which is based on the fact that the difference between the two populations may not be normally distributed. We use this test due to the fact that both populations are composed of distances (categorical variables). For this reason, we cannot say that the difference between the two is normally distributed.

The **null hypothesis** of this test is: *There is no distance difference between the shortest paths of the defaulted nodes and the shortest paths of the non-defaulted nodes.*

We verified this hypothesis for 20 scenarios: the defaulted nodes shortest paths and each of the 20 cases of non-defaulted nodes shortest paths. Thus, we applied the Wilcoxon signed-rank test for 20 populations. The chosen confidence level was 0.05. We observed that in all cases the null hypothesis was rejected. Thereby, we concluded that there are differences between the shortest paths of the defaulted nodes and the non-defaulted nodes.

Given the two presented technique, defaulted sub-network extraction and calculating the distance between node, we validated our assumption that defaulted nodes are close to each other in the graph. For this reason we considered that graph could provide significant insights in credit scoring prediction.

5
Solutions

With this project we contributed to the credit scoring literature twofold. We firstly reproduced the state of the art credit scoring prediction model, the *Altman model*, and tested its performance on Dutch SMEs data. We highlighted strong limitations of this the state of the art and created a model, the *financial coefficients model*, which is able to overcome these constraints. Secondly, we created a novel model which does not only include the financial features, but also graph features. This model is a significant contribution to the credit scoring prediction literature as it does not only take into account the financial state of one company, but also the interactions mapped through a transactional network. Furthermore, the graph features are extracted from a real-world transactional network. We call this model the *hybrid model*. Furthermore, we performed a systematic analysis into the problems that occur when predicting credit scoring and possible solutions that could diminish them.

5.1. Financial Ratios Model - Altman Model

The Altman model that we consider was created in order to predict credit scoring of SMEs [6]. It was trained and tested on companies from the United States of America. However, the authors mention that the model needs to be adapted according to each geographical region. As it was intensively used within literature, we considered reproducing the model and adapting it to Dutch SMEs, using the data provided by Exact. We are further explaining the steps that we take in order to design this model.

The Altman model is based on financial ratios corresponding to the 5 aforementioned categories, *liquidity*, *leverage*, *coverage*, *profitability* and *activity*. The financial ratios were calculated with respect to the financial coefficients derived from the RCSFI codes. In [6] the authors clearly state that the ratios are calculated only for companies with non-missing data. In our data-set, however, we encounter the problem of missing values for some financial coefficients. This is a result of missing values of RCSFI codes used to compute them. Thus, we discovered the first limitation of the most popular credit scoring model for SMEs in literature. We managed to calculate the financial ratios for 71 out of 735 samples of defaulted companies and 81382 out of 227345 samples of non-defaulted companies. Thus, the data-set used in order to test the Altman model was composed of 71 defaulted companies and 81382 non-defaulted companies. The time horizon of the defaulted companies in this specific data-set is from 2014 until 2019. Thus, we have SMEs which defaulted within this interval.

As part of the reproductibility of the Altman model on Dutch SMEs 3 steps were required: *undersampling*, *feature selection* and *classification*. This chapter is carefully detailing each step. However, the results of this model are presented in the Experiments chapter.

Undersampling

In this methods the authors did not, however, include all the available non-defaulted companies. When selecting them, they took into account the average default rate of the SMEs in the USA, which is 6%. The **average default rate** is a percentage which tells how many companies that received a loan defaulted and were not able to reimburse it. In order to incorporate the default rate, the authors performed a random selection of samples from the non-defaulted companies per year such that they preserve this percentage. By doing so, they created a model which is adapted to a real world scenario of data distribution and they also used the 6% default rate as a prior probability to the employed classifier. We are

further referring to this **undersampling technique** as a **domain driven method**.
We performed the same domain driven undersampling method on our data. It is crucial that the default rate is adapted on the geographical region. For this reason, we used the default rate of 2.5%, which is a pessimistic approximation given the previously reported default rates of SMEs in the Netherlands in 2018 (2.2%) [1], 2015 (1.6%) and 2013 (2.3%) [2]. We, thereby, selected non-defaulted SMEs each year such that the 2.5% average default rate is conserved. Our time horizon covers 6 years due to the limitation of calculating the financial ratios. For this reason, we considered it important to sample the companies taking into account whether they existed or not in that particular year. We do not have information in our data-set which tells when each SME was founded. For this reason, we ensured that we sample for one specific year by considering the year the company joined EOL. For instance, when under-sampling non-defaulted companies for 2016, we firstly ensured that the companies that were random chosen were included in EOL in 2016 or earlier. It needs to be mentioned that a more correct approach would be using a stratified random sampling on the sector and company size. However, as a result of having few samples for which the features could be calculated, we considered to give up on the stratification. The domain driven undersampling method is depicted for each year in Table 5.1.

Feature Selection

When it comes to creating the data-set of financial ratios, we selected the ones that were the most effective in credit scoring prediction. It needs to be mentioned that the literature selection criteria was preserved, thus we only the collected financial ratios divided into the 5 categories. We further had some data availability issues. Thus, we excluded all candidates that could not be computed due to the fact that the data could not be accessed.

Year	No. Defaults	No. Non-Defaults	Total Sample
2014	1	40	41
2015	1	40	41
2016	3	120	123
2017	2	80	82
2018	23	920	943
2019	41	1640	1681
Total	**71**	**2840**	**2911**

Table 5.1: This table presents how the data-set for the Altman Model was created taking into consideration the default rate of the Netherlands. We show precisely how many defaulted companies we have available (second column) and how many non-defaulted companies we randomly select (third column) each year in order to preserve the default rate. Our final data-set contains 2911 Dutch SMEs out of which 71 are defaulted and 2840 are non-defaulted.

The feature selection method explained in [6] is composed of an unsupervised feature selection followed by a supervised feature selection. The unsupervised feature selection implied verifying the correlation between features from each category. The supervised feature selection is done using a log likelihood test. This specific statistical test is used in order to determine whether a feature is statistically significant for the logistic regression classifier.
Table 5.2 presents details about the financial ratios that we chosen from literature (first column), the financial ratios that we could compute with the available data (second column), the result of the first feature selection (third column) and the result of the second feature selection (fourth column). The financial ratios depicted in the fourth column are the ones that we further use in our model. From the Table 5.2 we can also observe the fact that no financial ratio could be computed from the **activity** category. Furthermore, our model only includes financial ratios from the **coverage** and **leverage** categories due to the fact that the other groups did not contain statistically significant features. This could, indeed, reduce the complexity of the model, but we prioritized the fact that our model contains only significant features. We make this claim due to the fact that the Altman model relies on a logistic regression classifier, which is extremely sensitive to features which are correlated or irrelevant. The two feature selection methods are further explained.

Correlation based feature selection (unsupervised). This step was necessary in order to ensure the fact that the features included into the model are independent. We considered that, according to their definitions, each category has the purpose of highlighting a different performance indicator. However,

[1] Dutch SME bank financing, from a European perspective - Centraal Planbureau - July 2019

[2] Netherlands - technical note-financial stability and stress testing of the banking, household and corporate sectors - IMF Country Report No. 17/95

some financial ratios from each category could be correlated with each other.

	Financial Ratios in Literature	**Financial Ratios for which Data is Available**	**Financial Ratios after Correlation Selection**	**Financial Ratios after Significance Selection**
Liquidity	Cash/Total Assets Working Capital/Total Assets Cash/Net Sales Intangible/Total Assets Current Ratio Quick Ratio Intangible/Total Assets	Cash/Total Assets Working Capital/Total Assets	Working Capital/Total Assets	Working Capital/Total Assets
Leverage	Short Term Debt/Equity Book Value Equity Book Value /Total Liabilities Liabilities/Total Assets Short Term Debt/Total Debt Short Term Debt/Total Assets Debt/EBITDA Net Debt/EBITDA Total Debt/Total Assets Total Liabilities/Tangible Total Assets Total Liabilities/Net Worth Capital Employment/Total Liabilities	Short Term Debt/Equity Book Value Equity Book Value /Total Liabilities Short Term Debt/Total Assets	Short Term Debt/Equity Book Value Equity Book Value /Total Liabilities Short Term Debt/Total Assets	Equity Book Value /Total Liabilities Short Term Debt/Total Assets
Coverage	Ebitda/Interest Expenses Ebit/Interest Expenses Interest Expenses/Sales Debt Service Coverage Ratio	Ebitda/Interest Expenses Ebit/Interest Expenses	Ebitda/Interest Expenses	-
Profitability	Ebit/Sales Ebitda/Total Assets Net Income/Total Assets Retained Earnings/Total Assets Net Income/Sales Return of Equit Operating Profit/Net Income Sales Growth	Ebitda/Total Assets Retained Earnings/Total Assets	Ebitda/Total Assets	-
Activity	Sales/Total Assets Account Payable/Sales Account Receivable/Liabilities Working Capital/Sales	-	-	-

Table 5.2: This table shows the process of filtering financial ratios found in the literature (first column) according to whether the data is available (second column), whether they are correlated with others in the same category (third column) and whether they are statistically significant (fourth column).

In order to preserve the independence of the features we checked whether features within each category are correlated. If so, from one pair of correlated financial ratios we only kept an arbitrary chosen one.

After performing this type of selection we will further keep the following financial ratios: *working capital/total assets*, *short term debt/equity book value*, *equity book value/total liabilities*, *short term debt/total assets*, *ebitda/interest expenses*, *ebitda/total assets*.

Log Likelihood based feature selection (supervised). This feature selection step ensures the fact that the features that we include into our model are significant to the logistic regression classifier. In order to select the most meaningful features for the model, the log-likelihood significance statistical test could be employed. This test's desire is to determine whether the performances of the model are affected by removing some of the features. In other words, it verifies if a simpler model performs better than a more complex model.

In this project we validated the significance of all the 6 aforementioned financial ratios using the log-likelihood significance statistical test. Similarly to [6], we chose the confidence level of 25. Thereafter, we removed features one by one and determine whether the significance level is above or below the confidence level. Finally, we observed that 3 out of 6 features, namely *working capital/total assets*, *equity book value/total liabilities* and *short term debt/total assets*, are the statistically significant features that we can further include into our model.

5.2. Financial Coefficients Model

As we previously saw, the Altman model has considerable limitations due to missing data. We considered that excluding almost 90% of the 735 defaulted companies is a serious loss of information. For this reason, our aim was to further develop a model that is not that affected by the missing data and could incorporate all the default samples that we have available. Thus, we considered that instead of developing a model which is based on *financial ratios*, we could only rely on the information offered by the *financial coefficients*, which we are the nominators and denominators of the ratios. It needs to be mentioned that this model is not designed with a comparison purpose with the Altman model due to the fact that they do not use similar features.

Our financial model was originally computed with 12 financial coefficients features. These financial coefficients are: *cash*, *current assets*, *ebit*, *ebitda*, *equity book value*, *interest expenses*, *retained earnings*, *revenue*, *short term debt*, *total assets*, *total liabilities* and *working capital*. In Figure 5.1 we are depicting the model's pipeline. In the following subsections we are clearly explaining the first 3 steps of the pipeline, namely the *train/test split*, the *stratified random sampling* and the *feature selection*. The last two parts are detailed in the *Experiments* chapter. This model is further referred as the *baseline* model.

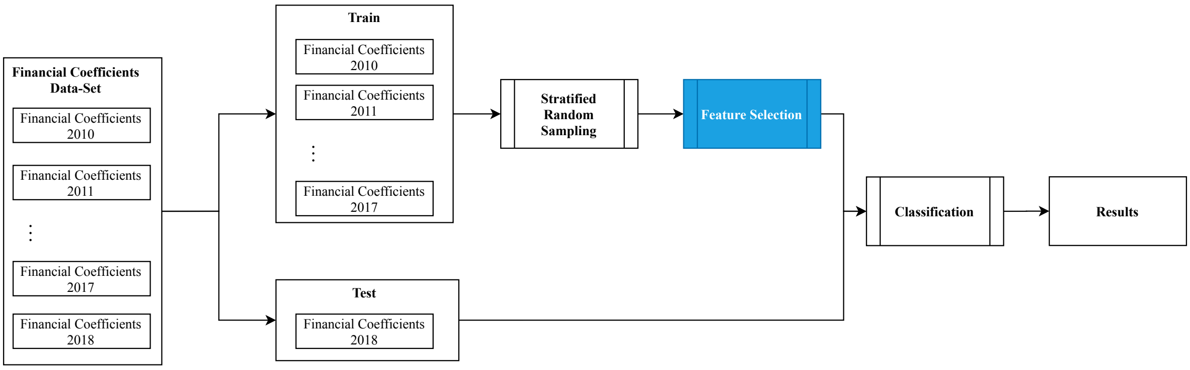

Financial Coefficients Data-Set
Financial Coefficients 2010
Financial Coefficients 2011
⋮
Financial Coefficients 2017
Financial Coefficients 2018
Train
Financial Coefficients 2010
Financial Coefficients 2011
⋮
Financial Coefficients 2017
Test
Financial Coefficients 2018
Stratified Random Sampling
Feature Selection
Classification
Results

Figure 5.1: The Pipeline Corresponding to the Financial Coefficients Model. The Feature selection step is applied only when using a linear classifier. When we are using a non-linear classifier, we do not perform feature selection.

5.2.1. Train Test Split

Before splitting our data into train and test, we firstly needed to properly understand the most suitable way of evaluating our model. Thus, we designed an evaluation methods that will be referred to throughout this thesis as *production scenario*. It needs to be mentioned that this evaluation technique is designed such that the model could be deployed into production afterwards.

As mentioned earlier, we have complete financial data from the end of 2010 until the end of 2018. Our labels are extracted from 2011 until 2019 inclusive. Given that our aim is to design a credit scoring model which is able to predict the default event in the following year, we decided to use the year of 2019 as the test set and other years (from 2011 until 2018) as training set. As mentioned before, the financial coefficients calculated at the end of each year correspond to the label of one company in the following year. For instance, the financial coefficients calculated at the end of 2016 correspond to the label of that company in 2017. In other words, we develop our model based on the idea that if a SME defaults in year *t*, the financial coefficients that resulted at the time step *t-1* should be meaningful enough to predict the default event. In the same manner, if an SME is declared non-defaulted at time step *t*, then the financial coefficients calculated at the end of *t-1* should be significant to predict the non-default event.

In the light of these aspects, our **training set** is composed of defaulted and non-defaulted companies from the year of 2011 until the year of 2018 (with financial coefficients calculated at the end of 2010 until the end of 2017) and our **test set** consists of defaulted and non-defaulted companies of 2019 (with

financial coefficients calculated at the end of 2018).
We believe that we can predict the companies that are at risk of defaulting in the following year. Thus, when it comes to deployment, the model is retrained at the end of each year with the new labels (defaulted and non-defaulted) that are available until that time and used in order to predict whether the companies which were still declared non-defaulted are endangered.

5.2.2. Stratified Random Sampling

In a similar manner as the Altman model, we considered employing a *domain driven data undersampling method* in order to preserve the default rate of the Netherlands. We believe that in this manner we are able to adapt the model such that it could be properly used in a real world scenario. Taking into consideration the fact that statistically 2.5% of the Dutch SMEs default every year without being able to pay their loans, we believed that in a test set we could expect the same percentage of defaults. Thus, we performed a stratified random sampling technique on the training set as shown in Figure 5.1.
The non-defaulted SMEs in the training set are stratified random sampled based on sector and size. We took the default rate into account each year when performing stratified random sampling. However, in our data-set we did not have defaulted samples for sectors U, O and T and for companies for which the size was bigger than 500 workers. This could be due to the fact that the defaults in these situations are infrequent, but it can also be due to the fact that they were not reported. For this reason, we excluded them from the data-set. Thereby, the number of defaulted companies in one year should be 2.5% of the entire data-set on that specific year (including both defaulted and non-defaulted). In Table 5.3 we give an overview of how many defaulted and non defaulted companies we obtained each year, as well as how many defaulted and non-defaulted companies are in total in the training set. When performing the stratification, we needed to determine how many samples per each category we need to randomly choose. For this used the following formula:

$$stratified\_random\_sample = \frac{sample\_size \cdot stratum\_size}{population\_size} \tag{5.1}$$

The terms of this equations are the following:

- **stratified random sample** - the number of non-defaulted selected;
- **sample size** - the number of non-defaulted samples needed in order to preserve the default rate;
- **stratum size** - number of non-defaulted in a specific category;
- **population size** - the number of non-defaulted existing in the data-set.

As a disclaimer, some of the numbers of non-defaulted might be slightly bigger than the ones that the stratified sampling expects due to the rounding caused by floating number.
After performing the stratified random undersampling sampling technique, our **training set** is composed of **325 defaulted** companies and **13505 non-defaulted** companies. It is necessary to clarify that the domain undersampling technique is only applied on the training set. The **test set** contains **336 defaulted** companies and **216371 non-defaulted** companies.

Year	No. Defaults	No. Non-Defaults	Total Sample
2011	2	140	142
2012	1	141	141
2013	1	146	147
2014	6	308	314
2015	15	650	665
2016	29	1203	1232
2017	77	3112	3189
2018	194	7785	7979
Total Train	**325**	**13505**	**13830**

Table 5.3: This table presents how the data-set for the Financial Coefficients Model was created taking into consideration the default rate of the Netherlands. We show precisely how many defaulted companies we have available (second column) and how many non-defaulted companies we randomly select (third column) each year in order to preserve the default rate. Our final training data-set contains 13830 Dutch SMEs out of which 735 are defaulted and 13505 are non-defaulted.

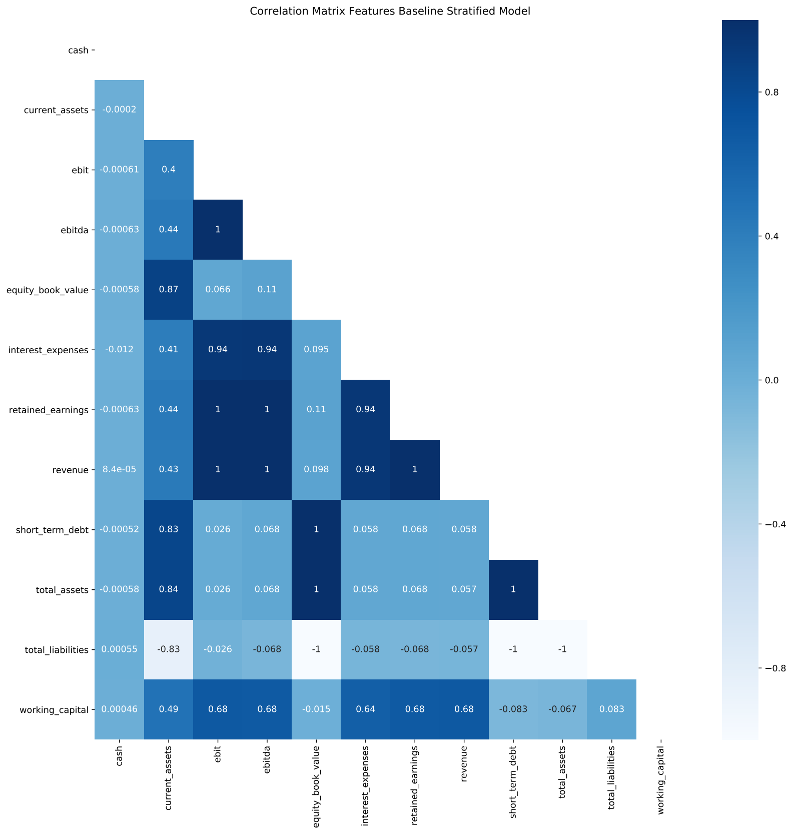


Figure 5.2: Correlation Matrix Between Features of the Baseline Model after Stratify Random Sampling

5.2.3. Feature Selection

In terms of feature selection techniques we only employ unsupervised feature selection based on feature correlation. As the economy is changing annually and some features could be useful for certain years, we considered that employing any supervised feature selection could lead to including bias our model on the training set. Our aim is to create a general model that could predict the credit scoring in any economical situation. For this reason, biasing it on the samples from the training set would not help in the situation of an economical crisis in which the samples in the test set could be different than the ones that we already included in the train set (from 2011 until 2018).

We need to mention that the feature selection was only applied for the linear classifier. For this type of classifiers it is important that each feature is bringing new valuable information. Thus, the selected features should be linearly independent variables in this case. We perform this selection only on the linear classifiers due to the fact that they are extremely sensitive to correlated features. We did not use it for non-linear classifiers due to the fact that their performance can be boosted by having correlated features. The reason for this is the fact that this type of classifiers can gather valuable information from correlated features.

In order to test the dependency of features, the correlation matrix can be employed. This matrix contains correlation coefficients between all features. Those which have high correlation coefficient (positive or negative) will be filtered such that only one of them will remain in the model and the other one will be removed. We considered a high correlation the situation in which the correlation coefficients is smaller than -0.9 (negative correlation) or bigger than 0.9 (positive correlation).

In Figure 5.2 we display the correlation matrix between the features of the model. We can observe that we have strong correlation between *ebit*, *ebitda*, *retained earnings*, *revenue* and *interest expenses*. From these 5 features we chose to drop 4 and only include ebitda in the model. In Figure 5.2 we can

also observe a strong correlation between *equity book value*, *short term debt* and *total assets*. For this reason we dropped the first two and only kept the total assets feature.
After performing this unsupervised feature selection, our final model was composed of 4 out of 12 financial coefficients, namely *cash*, *ebitda*, *current assets*, *working capital*. Only these features were employed for the linear classifier. For the non-linear ones, all the 12 features were kept.
The correlation between the financial coefficients could be either a consequence of the formula used to calculate it or the fact that two financial coefficients can have similar economic meaning.

5.3. Hybrid Model

This section aims to give a clear explanation about how we integrate financial features and graph feature. We call the resulted model the *hybrid model*. This model is further compared with the baseline, the financial coefficients model, in order to understand whether graph features bring a significant contribution in predicting credit scoring.

5.3.1. Hybrid Model Design

The hybrid model is composed by both financial and graph related features. Its pipeline is depicted in Figure 5.3.

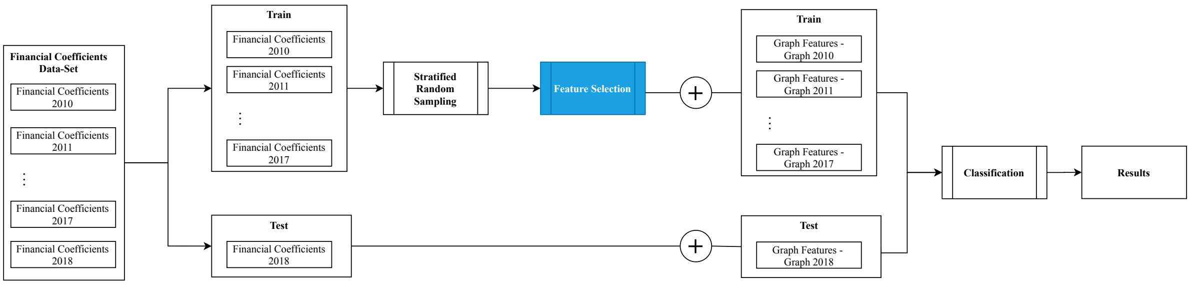

Financial Coefficients Data-Set
Financial Coefficients 2010
Financial Coefficients 2011
⋮
Financial Coefficients 2017
Financial Coefficients 2018
Train
Financial Coefficients 2010
Financial Coefficients 2011
⋮
Financial Coefficients 2017
Test
Financial Coefficients 2018
Stratified Random Sampling
Feature Selection
+
+
Train
Graph Features - Graph 2010
Graph Features - Graph 2011
⋮
Graph Features - Graph 2017
Test
Graph Features - Graph 2018
Classification
Results

Figure 5.3: The Pipeline Corresponding to the Hybrid Model. The Feature selection step is applied only when using a linear classifier. When we are using a non-linear classifier, we do not perform feature selection.

The first step of this model is splitting the data into test (financial coefficients calculated at the end of 2018) and train (financial coefficients calculated from the end of 2010 until the end of 2017). Afterwards a **stratified random sampling** is applied. For linear classifiers, the following step is a correlation based **feature selection** in which we filter out financial coefficients that do not bring new valuable information. These two processes are executed in the same manner as in the previous model.
What makes this model different than the baseline is the addition of graph features. It is important to be highlighted the fact that the time domain plays an important role. Thereby, for each sample for which the financial coefficients are calculated at the end of year *t*, we add network features extracted from the graph at the end of year *t*. For instance, if the financial features are calculated at the end of year 2015, the corresponding graph features needs to be extracted from the graph at the end of 2015.

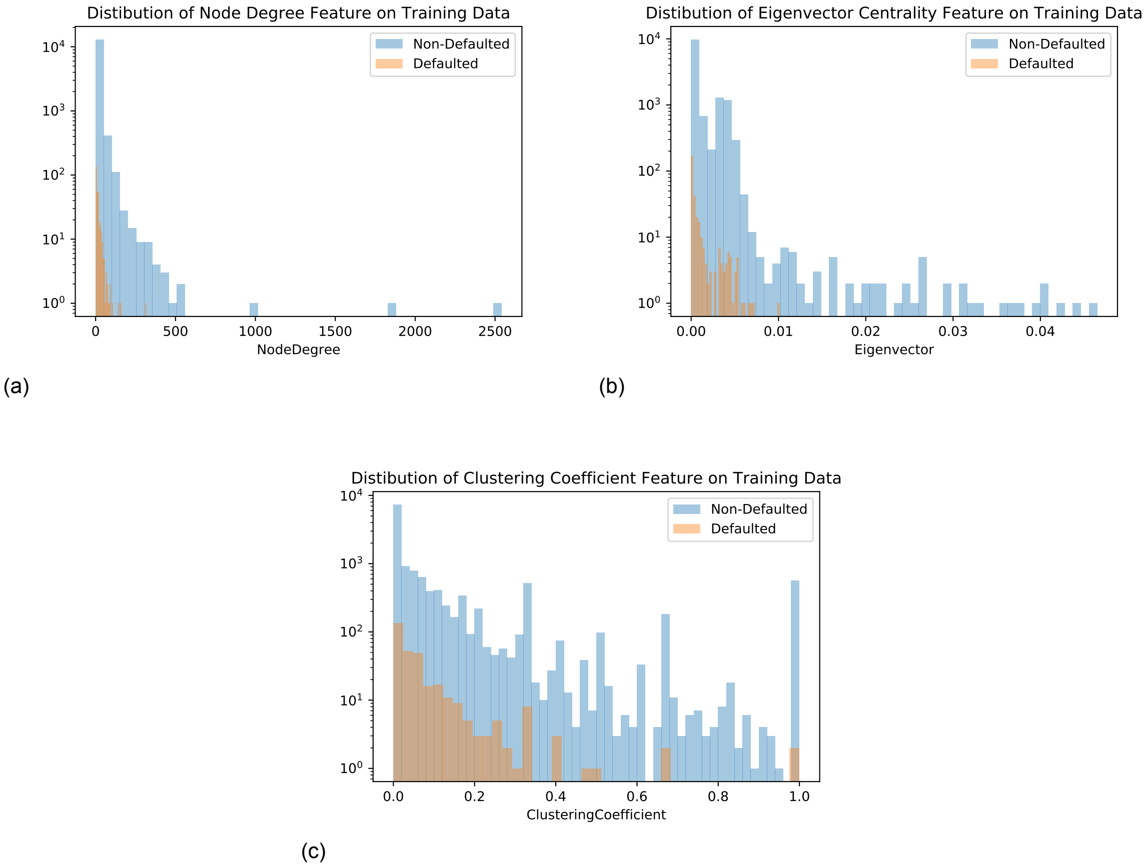
5.3.2. Graph Features

This sub-chapter aims to give a detailed explanation into which graph features were considered for this study. We classify these features into 2 categories: **network centrality features** and **graph embedding features**. Furthermore, we provide motivation into why some features were computed and we also show why we believe they could bring benefits to the financial coefficients model.

Network Centrality Based Features

Network centrality is a term used in complex networks theory in order to indicate the most important nodes in a graph. A node, also known as vertex, is considered important if it propagates information faster than the others. Adapting this definition to our problem, credit scoring, one node (represented by a company) is considered important if its default can have a significant impact on others.
In the Related Work Section we mentioned the 5 most popular network centrality metrics: *node degree*, *clustering coefficient*, *eigenvector centrality*, *closeness centrality* and *betweenness centrality*.

However, we did not use all these features into our hybrid model and we are further explaining the motivation behind this.
The most important selection criteria for network centrality features was determining the computational expenses. The most computational expensive features are the betweenness centrality and the closeness centrality. The reason for this is the fact that they both require calculating the average shortest paths, which has a complexity of O(n^2), whereby n is the number of nodes of the graph. Given that our biggest network contains more than 216000 nodes, we decided that these 2 network centrality metrics are too computational expensive to compute.
We only considered the 3 remaining network centrality features for our hybrid model: *node degree*, *clustering coefficient* and *eigenvector centrality*. In order to understand how meaningful they are in differentiating between defaulted and non-defaulted SMEs, we analyzed the distribution of the training data with respect to these features. These distributions are plotted in Figure 5.4.



(a)

(b)

(c)

Figure 5.4: Distributions of the 3 network centrality features on the training data for both defaulted and non-defaulted SMEs.

In Figure 5.4 (a) we can see that the node degree features can be a good discriminant between defaulted and non-defaulted. It is interesting to observe the fact that non-defaulted nodes tend to have a high node degree, while the defaulted ones are characterised by a low node degree. This means that a company's financial stability is strongly dependent on its number of counter-parties. Thereby, we can claim that companies which have a few number of business partners have a higher default likelihood.
In Figure 5.4 (b) we present the the distribution of defaulted and non-defaulted SMEs with respect to the eigenvector centrality metric. It is interesting to observe that also in this case the two classes could be separated. We can remark that a high eigenvector centrality is observed for non-defaulted companies, while the defaulted ones usually have a low eigenvector centrality. As an interpretation of this feature, a company whose counter-parties have many connections has a low default likelihood.
In Figure 5.4 (c) we depict the distribution of defaulted and non-defaulted SMEs with respect to the clustering coefficient. We can observe that in this case the clustering coefficient of each node is not a good feature to distinguish between defaulted and non-defaulted. However, we still employed it as it

can become a good discriminant in combination with other features.
As an overview of our analysis on the network centrality features, we can say that the node degree and eigenvector centrality are potential candidates, which can help distinguishing between defaulted and non-defaulted. We remarked the fact that the financial stability of one SME is strongly dependent on its number of connections and also on the number of connections of the counter-parties. Thus, we would expect that these two features to be a good addition to the financial coefficients model.

Graph Embedding Features

Another category of graph features that we utilized are **graph embeddings**. We created the embedding of each year starting with 2011 until 2019 and used the features given by the embeddings in combination with the financial coefficients. The graph embedding features were generated using node2vec [26] with the following configurations: p = 1, q = 1, number of walks = 10 and walk length = 80. According to [26], this node2vec configuration corresponds to DeepWalk [46] graph embeddings framework. Thus, we can say that DeepWalk was employed in order to embed the network into a vector feature space. We used node2vec in stead of DeepWalk as the authors of [26] claim that this framework is more efficient and we need efficient solutions given that we have a big network. We, initially, generate 128 graph features from the embedded network. In this case we did not perform a distribution analysis as in the situation of network centrality features due to the high dimensionality.
The embedding of the graph requires that each node's vicinity needs to be explored. For extremely large network, the computational time of the embeddings increases significantly. In our case, the smallest network (corresponding to year 2011) contains 18768 nodes and the biggest one (corresponding to year 2018) contains 217092 nodes. The time that the algorithm takes in order to embed the graph is approximately 45 minutes and 14 hours respectively. Although node2vec requires tuning its 5 parameters in order to achieve the optimum embedding, we decided to keep the DeepWalk configuration throughout our experiments due to the high computation demand.
The network embedding has the purpose of learning the nodes representation such that the graph could be reconstructed from the embedding space. We believe that these features could be meaningful considering that they are describing the node's position in the network. In addition, two nodes which have similar position could share some similarities. As we previously observed, the default nodes are closer to each other in the network. Thus, we can probably introduce meaningful information in the model by employing features which are learning the node representation within the network.
To sum up, the graph features employed in the hybrid model are: **node degree**, **clustering coefficient**, **eigenvector centrality** and **network embedding features**

5.4. Evaluation Metrics

As our purpose is classifying companies into 2 classes, defaulted and non-defaulted, we are experiencing a binary classification problem. Thus, our data is composed of two classes, the positive class and the negative class, represented by the defaulted SMEs and the non-defaulted SMEs, respectively. The classification would produce 4 possible outcomes:

- defaulted classified as defaulted (TP);
- defaulted classified as non-defaulted (FP);
- non-defaulted classified as defaulted (FN);
- non-defaulted classified as non-defaulted (TN).

In terms of evaluation metrics, which assess the correctness of our classification, we need to take into account the high class imbalance between the two classes. After applying the stratified random sampling our data-set is composed of around 2.5% defaulted companies and 97.5% non-defaulted companies.
The most well known and widely used metric for assessing the classification performance is the accuracy. According to [2], in the situation of imbalance data, this metric is extremely misleading. The reason for this is the fact that the accuracy assesses how well the classifier is performing overall. Thus, classifying correct most of the majority class samples could lead to a very high accuracy. Hence, the

minority samples, which are the most important, are not taking into consideration.

We continue by further looking into the metrics which are suitable for dealing with class imbalance. We divide these metrics into two categories: metrics which focus on the performance of correctly classifying both classes (*compromise metrics*) and metrics which underline the performance of correctly classifying the positive class (*minority metrics*).

5.4.1. Compromise Metrics

From this categories of metrics we employed the *ROC AUC score* and the *Matthews Correlation Coefficients*. We are further detailing these two and critically analyzing them in terms of compatibility with our problem. We did not choose other metrics, such as Kappa due to its high sensitivity when it comes to the marginal totals of the confusion matrix.

ROC AUC Score

The most popular metric when facing the class imbalance issue is the **ROC AUC score**. This metric gives a score between 0.5 and 1, in which 0.5 corresponds to random guessing and 1 corresponds to a perfect classification. This metric is calculated by plotting the True Positive Rate with respect to the False Positive Rate and calculating the area under the curve of the two. The formulas for the two are the following:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

In [49], the authors, however, highlighted the disadvantages of ROC AUC. They conducted a study and analyzed this metric for different data-sets with different ratios of positive and negative samples. They concluded that the ROC AUC may be extremely inaccurate for highly imbalanced data-sets. The reason for this is the fact that this metric relies on the False Positive Rate, which in the situation of a high imbalance data-set could be extremely low as a result of a big number of true negatives. The consequence of this could be a very low False Positive Rate and, thereby, an erroneous ROC AUC score. We decided to employ the ROC AUC metric, but due to its downsides which were presented in the literature, we are not fully relying on its outcome in order to asses the performance of our model.

Matthews Correlation Coefficients

Matthews Correlation Coefficients (MCC) is considered by literature an adequate metric for class imbalance due to its strong criticism when evaluating the performance. It should be reliable due to the fact that it generates a score with respect to the 4 classification outcomes. [18] Thereby, a high score is achieved only in the situation of having results for FP and FN close to 0 and the ones for TP and TN close to their maximum value. This metric produces a score which ranges between -1 and +1. A score of -1 corresponds to a perfect misclassification, a score of 0 corresponds to random guessing and a score of 1 corresponds to a perfect classification.

The MMC formula is the following:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) + (TP + FN) + (TN + FP) + (TN + FN)}} \tag{5.2}$$

5.4.2. Minority Metrics

When it comes to the minority metrics we employed the *sensitivity (true positive rate)* metric. In depth insights into our intuition are presented further for this metric.

Sensitivity

This metric aims to evaluate how well does the model predict the negative class. We opted for this metric as we desired a clear overview into the percentage of defaulted companies classified as defaulted. The value of this metric ranges from 0 to 1, where 0 means that no defaulted was predicted as defaulted and 1 means that all defaulted were predicted as defaulted.

Given the true positive and false negative, the sensitivity can be calculated using the following formula:

$$sensitivity = \frac{TP}{TP + FN} \tag{5.3}$$

6 Experiments

In this chapter we are covering the experiments that we performed on the 3 aforementioned models: the Altman Model, the Financial Coefficient Model and the Hybrid Model. All the 3 models were evaluated using the *production scenario*. In all 3 cases, we employed 3 metrics, 2 compromise metrics (ROC AUC and MMC) and 1 minority metric (sensitivity).

We split experiments chapter into two main Sections that refer to which types of financial features were employed. Thus, the first Section includes experiments with financial ratios features (Altman Model), while the second Section contains experiments with financial coefficient features (baseline and hybrid model).

6.1. Financial Ratios Model - Altman Model

This section shows the performance of the Altman Model presented in the first part of chapter 5. As we aforementioned, the data-set is composed of 2911 samples of non-defaulted SMEs and 71 samples of defaulted SMEs.

The Altman Model requires a logistic regression classifier in order to determine whether a sample is defaulted or non-defaulted. In [6] the authors mention that they are using the undersampling technique followed by another class imbalance technique which was highlighted by the authors of [32]. When it comes to the latter, Kind et al. [32] are targeting adapting logistic regression for rare events classification. They conclude that an algorithm-level method that assigns a higher weight to the minority class was the best approach in their experiments. Thus, in a similar manner we applied this class imbalance technique when evaluating this model.

We used the production scenario to evaluate the Altman model, namely we trained on samples from 2014 until 2018 and tested on those from 2019. We used the logistic regression classifier and weighted the minority class. We depict our results in Table 6.1.

Classifier	ROC AUC
Logistic Regression Classifier (weighted)	0.58446

Table 6.1: Altman Model Evaluation ROC AUC

We can observe from 6.1 that the performance of this model is extremely poor. We were further interested into the reason for such bad results. Thus, we decided to reduce the dimensionality of the features from 3D to 2D using principal component analysis (PCA) and visualize them. In Figure 6.1 we show the plotted features.

From Figure 6.1 it can be observed the fact that the two classes, defaulted and non-defaulted are strongly overlapping. Due to this, the decision boundary could not properly separate the two classes. This could be the main reason why the logistic regression classifier underperforms on this data.

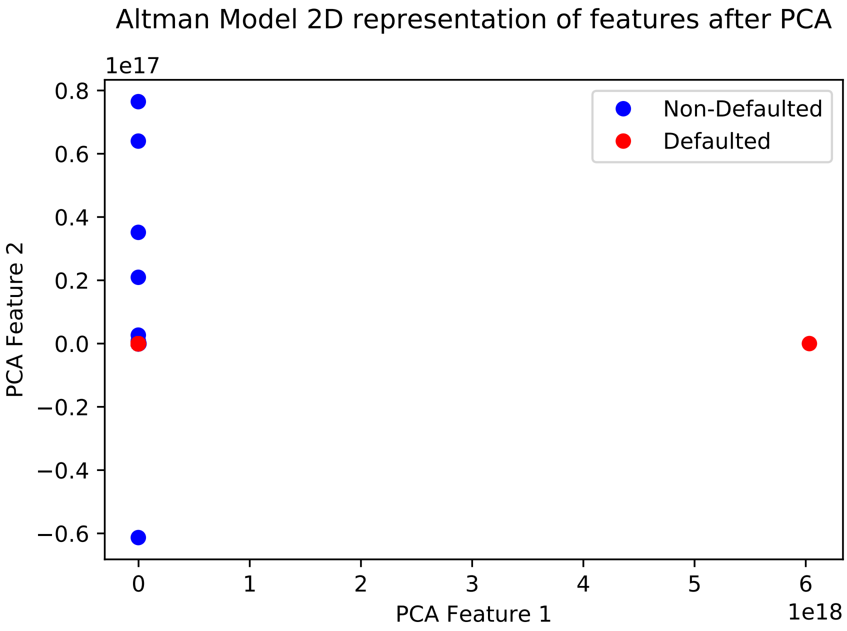


Figure 6.1: 2D visualization of the Altman model features.

6.2. Financial Coefficients Models

The financial coefficients model contains only financial features derived from the RCSFI codes. We need to specify the fact that for the linear classifier (logistic regression) we performed a correlation based feature selection, reducing the dimensionality from 12 to 4 features. For other classifiers all the 12 financial coefficients were employed. This model is used as a **baseline** and its performances are compared to the hybrid model's outcome. The reason for this is the fact that we want to understand whether graph features are beneficial in increasing the accuracy of determining defaulted companies and, thereby, answer our main research question.

In all situations the classifier parameter tuning was employed. For this part, in stead of using exhaustive grid search in order to find the best configuration, we used randomized search with 10 fold cross validation. The reason for this is the fact that the parameters' space explored by both approaches is similar, but the run time for chosen one is substantially lower.

In a similar way to the Altman model, we are initially performing a stratified random undersampling technique. We are also combining this undersampling method with other class imbalance techniques. Thereby, we are examining the performances of our models in 3 situations: one situations in which class imbalance is not handled and 2 situations in which we are trying to solve the issue of having the classes extremely imbalanced. In the former we weighted the minority class and in the latter we used SMOTE in order to tackle class imbalance. We need to clarify that the scenario in which the class imbalance is not handled corresponds to the the situation in which only the undersampling method was employed and no other technique was applied.

In addition to the above, we need to mention that our aim is to have as many defaulted companies predicted correctly. For this reason, the performances of the models are critically evaluated using this criteria. For two models which are similar in terms of the number of correctly predicted defaulted SMEs, we consider that the best model is the one which is also able to properly predict a high number of non-defaulted SMEs.

When comparing the hybrid model with the baseline we employed the McNemar's statistical test [21]. This is a paired non-parametric test, which was designed for the situations in which the performances between two classifiers need to be compared and they are evaluated on only one test set. We have, indeed, only one test set in our production scenario (composed of samples of 2019), thus we considered that employing this test would be suitable to our issue. McNemar's test verifies the predictions of two models and checks whether they make similar errors on the test set. Its null hypothesis is: *The two models have the same error rate.*

6.2.1. Experiment 1: Hybrid Model - Network Centrality Features

In this subsection we are evaluating the performance of the hybrid model, which contains both financial and network centrality features against the baseline. These experiments aim to answer our first research sub-question and, thereby, provide us with a clear understanding of this type of network features.

We divided this subsection into two parts: in the first part we show the results that we obtained when performing the experiments, while in the second part we conduct an in-depth analysis of the results.

In the following tables we are using the following notations:

- **B** - Financial Coefficients Model (Baseline);
- **B + ND** - Hybrid Model using financial features and Node Degree;
- **B + CC** - Hybrid Model using financial features and Clustering Coefficient;
- **B + EC** - Hybrid Model using financial features and Eigenvector Centrality.

Results of Experiments

Class Imbalance Not Handled

The first set of experiments was performed without handling the class imbalance issue. Thus, in Table 6.2, Table 6.3 and Table 6.4 we show ROC AUC score, MCC score and sensitivity of all classifiers in the situation in which we do not perform other class imbalance technique besides the stratified random sampling that was previously mentioned.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	0.5	0.5	0.5	0.5
Decision Tree	0.73719	0.68236	0.7362	0.7362
Random Forest	0.70944	0.70211	0.70807	0.69608
XGBoost	0.7977	0.80507	0.80666	**0.81397**
AdaBoost	0.63533	0.64871	0.64948	0.64918
LightGBM	0.78988	0.78096	0.7721	0.77912

Table 6.2: This Table shows the ROC AUC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	-0.00508	0.002	-0.0047	-0.00508
Decision Tree	0.20342	0.16247	0.21552	0.21552
Random Forest	0.44035	0.46948	**0.48139**	0.42877
XGBoost	0.11254	0.11513	0.11595	0.11838
AdaBoost	0.13222	0.06815	0.07405	0.06351
LightGBM	0.10875	0.10544	0.10561	0.10401

Table 6.3: This Table shows the MCC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	0.10416	0.00297	0.10119	0.10416
Decision Tree	0.48214	0.37202	0.47916	0.47916
Random Forest	0.41964	0.40476	0.41666	0.39285
XGBoost	0.63988	0.65476	0.65773	**0.67261**
AdaBoost	0.27678	0.32738	0.3244	0.33333
LightGBM	0.625	0.60714	0.10561	0.60416

Table 6.4: This Table shows the sensitivity of all classifiers for all the analyzed models. The best result is shown with Bold.

Weighting the Minority Class

In this set of experiments we are using weighting based technique to overcome the problems which rise due to high class imbalance. More precisely, we are weighting the minority class (defaulted) and, by doing so, we punish the classifier more for mistakenly predict this particular class. We depict our results in Table 6.5, Table 6.6 and Table 6.7, which correspond to the ROC AUC score, MMC score and sensitivity metric.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	0.68669	0.70298	0.69095	0.703
Decision Tree	0.77999	0.780921	0.76701	0.77489
Random Forest	0.72647	0.67171	0.72837	0.68198
XGBoost	0.76975	0.75346	0.75195	0.76208
AdaBoost	0.75378	0.77545	0.77545	0.76314
LightGBM	0.7211	0.80399	0.72958	**0.83952**

Table 6.5: This Table shows the ROC AUC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	0.03817	0.03649	0.03529	0.03649
Decision Tree	0.0739	0.0719	0.06499	0.07313
Random Forest	0.081209	0.09602	0.08497	**0.15486**
XGBoost	0.10276	0.09677	0.0994	0.11045
AdaBoost	0.0656	0.07079	0.070728	0.065052
LightGBM	0.068	0.08262	0.07829	0.03119

Table 6.6: This Table shows the MCC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	0.55357	0.66369	0.61904	0.66369
Decision Tree	0.65773	0.66666	0.65178	0.64583
Random Forest	0.50297	0.36309	0.50297	0.37202
XGBoost	0.58333	0.55059	0.54464	0.55952
AdaBoost	0.61011	0.65476	0.65773	0.63988
LightGBM	0.5119	0.6994	0.51488	**0.74404**

Table 6.7: This Table shows the sensitivity score of all classifiers for all the analyzed models. The best result is shown with Bold.

SMOTE

In this set of experiments we are using a data based technique in order to address the class imbalance issue. We are generating synthetic samples from the minority class until the two classes are equal in size. We display our results in Table 6.8, Table 6.9 and Table 6.10, which correspond to the ROC AUC score, MCC score and sensitivity metric.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	0.70206	0.70198	0.69127	0.70206
Decision Tree	0.82255	0.80573	0.8003	0.80843
Random Forest	0.81743	0.81211	0.81175	0.82021
XGBoost*	0.88158	0.87847	0.87247	**0.88437**
AdaBoost	0.75999	0.76505	0.76646	0.76
LightGBM	0.86212	0.85435	0.83842	0.86859

Table 6.8: This Table shows the ROC AUC score of all classifiers for all the analyzed models. The best result is shown with Bold. McNemar statistical test was applied in order to assess whether the improvement was significant.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	0.03637	0.03635	0.03554	0.03637
Decision Tree	0.07412	0.07073	0.07418	0.0676
Random Forest	0.10553	0.10745	0.11248	0.1178
XGBoost	**0.13345**	0.13205	0.1334	0.12754
AdaBoost	0.05818	0.05952	0.06203	0.05224
LightGBM	0.1299	0.12638	0.12174	0.04003

Table 6.9: This Table shows the MCC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + ND	B + CC	B + EC
Logistic Regression	0.66071	0.66071	0.69127	0.66071
Decision Tree	0.77976	0.74404	0.71428	0.76785
Random Forest	0.69345	0.67858	0.67261	0.6875
XGBoost	0.81547	0.80952	0.79464	**0.82738**
AdaBoost	0.66369	0.67261	0.66369	0.708
LightGBM	0.7738	0.75892	0.72619	0.79166

Table 6.10: This Table shows the sensitivity score of all classifiers for all the analyzed models. The best result is shown with Bold.

Analysis of Results

From the experiments that we performed we observed that in all cases, the ROC AUC and sensitivity increased after the class imbalance issue was handled. We make this comment based on the fact that both increased significantly after applying SMOTE or weighting the minority class. In consequence, we believe that the two metrics are correlated and, thus, we can say that the ROC AUC is sensitive to the number of correctly identified defaulted SMEs.

Another interesting remark that we can make is the fact that the MCC score's highest value was reached when the class imbalance was not handled. This score is obtained for a relative low percentage of correctly predicted defaulted companies. For this reason, we can claim that the MCC score proved not to be strongly influence by the number of correctly identified defaulted companies.

The best overall performance in terms of correctly identified defaulted SMEs was achieved by XGBoost after synthetically generating samples of the minority class using SMOTE. It is interesting to observe that, in most of the cases the highest performance is achieved by the hybrid model. In all the 3 situations (class imbalance not handled, weighted minority class and SMOTE) the B + EC model was the best model in terms of the percentage of correctly identified defaulted SMEs. Moreover, the increase in performance of this model proved to be statistically significant when compared to the baseline according to McNemar's test.

6.2.2. Experiments 2: Hybrid Model - Graph Embedding Features

This subsection aims to evaluate the performance of the hybrid model, composed of financial and graph embedding features, against the baseline. We perform these experiments in order to answer our second research sub-question.

This subsection is divided into 3 parts. The first part has the purpose of offering a better understanding into the impact of the number of embedded features on each classifier's performance. In the second part we show some results that we obtain using this model. The last part contains a detailed analysis of the obtained results.

In the following tables we are using the following notations:

- **B** - Financial Coefficients Model (Baseline);
- **B + GE** - Hybrid Model using financial features and Graph Embedding features.

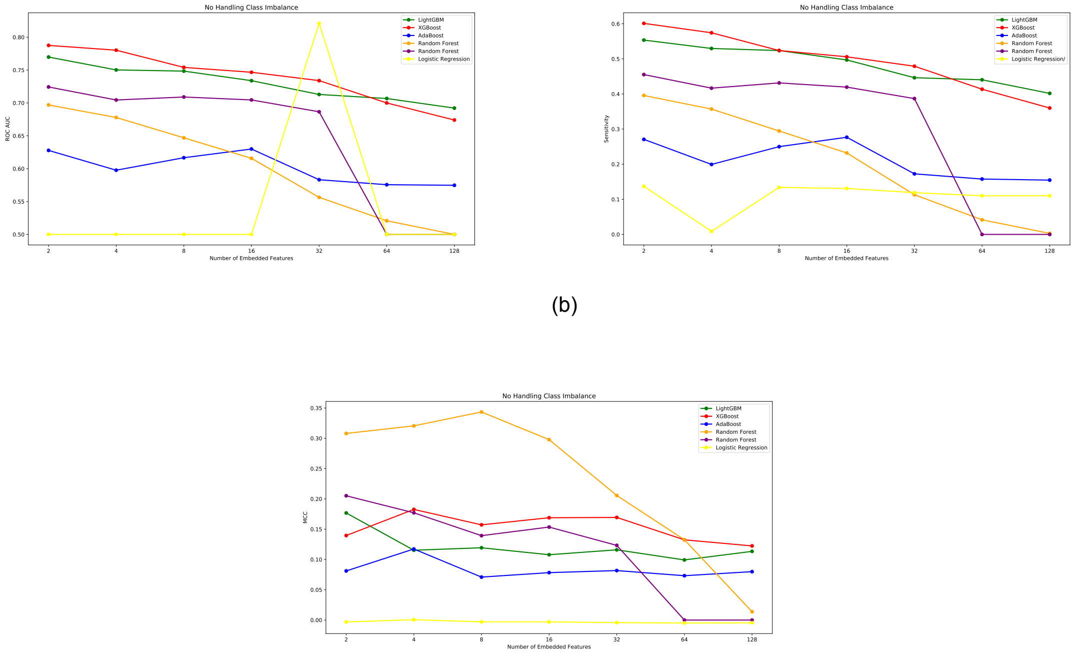
Experiments Different Number of Embedding Features

We started by generating graph embedding features using the node2vec framework with the following configurations: p = 1, q = 1, random_walk_length = 80, number_of_walks = 10 and number_of_dimensions = 128. This corresponds to the DeepWalk configuration of embedding the network and it generates 128 embedded graph features for each node in the network (each SME in the data-set).

We remarked the fact that the feature space was extremely high (128 graph embedding features + 12 financial features for boosting classifiers and 128 graph features + 4 financial features for linear classifiers). When performing experiments with this amount of features, we observed that some of classifiers were overfitting in mostly all situations (XGBoost, LightGBM), while others were incapable of learning due to the high feature dimensionality (Logistic Regression). We, thereafter, decided that a reduction into the number of graph embedded features needs to be performed. For this reason, we tuned the number of graph embedding features generated by node2vec.

The tuning was realized by constantly decreasing the dimensionality from 128 until 2. We performed a grid search for the optimal number of features. The number of features tested are the following: 2, 4, 8, 16, 32, 64, 128. In addition, we analyze all the 3 cases: no handling class imbalance, weighting the minority class and SMOTE.

From this experiment, we observed the fact that a high number of features has a tremendous effect on the classifiers' performance. We further noticed that decreasing the number of graph embedding feature significantly improved all the 3 analyzed metrics: ROC AUC, MCC and sensitivity.
In Figure 6.2, Figure 6.3 and Figure 6.4 we depict the previously mentioned metrics with respect to the number of graph embedding features. The purpose of these plots is to give a better understanding of the effect of feature dimensionality on each classifier's performance.
As a conclusion of this experiment, we can say that adding a high number of graph embedded features has a tremendous effect on the model's performance. For this reason, in order to properly understand whether these features have a beneficial impact in improving the baseline, the number of embeddings needs to be lowered.



(a)

(b)

(c)

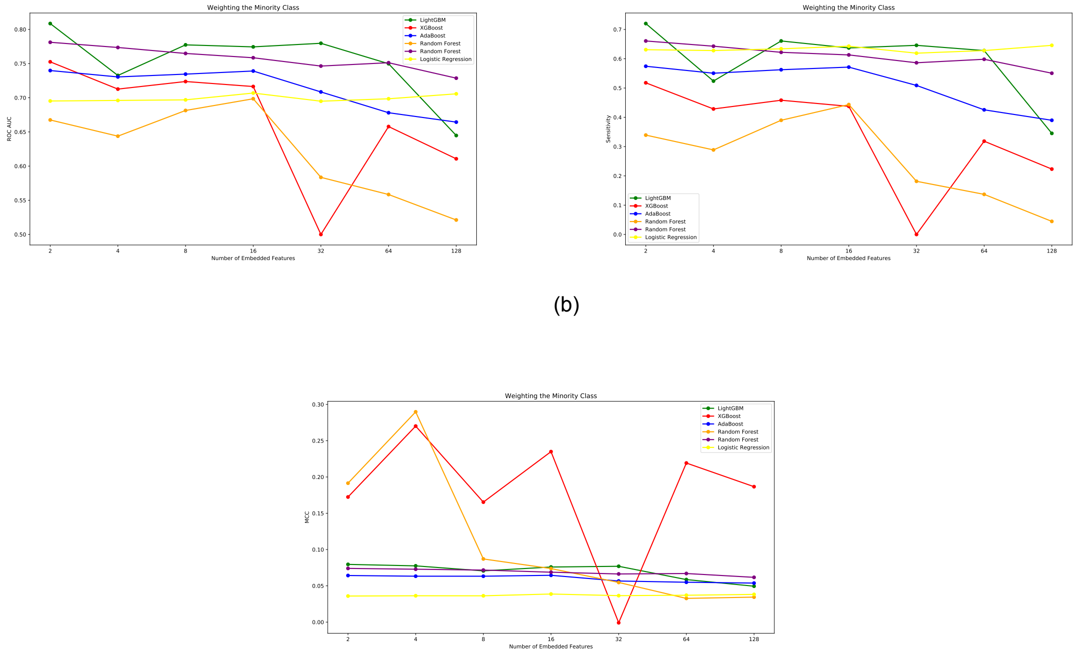
Figure 6.2: The ROC AUC (a), Sensitivity (b) and MCC (c) of the model B + GE when when varying the number of embedded features. These plots correspond to the case in which the class imbalance is not handled.

Results of Experiments

In this part we are depicting the results that we obtained for graph embeddings in comparison with the baseline. It needs to be mentioned that for the model B + GE all tables contain the best value obtained with the optimal number of feature for each classifier. For instance, the optimal number of features for XGBoost with SMOTE was 4, while for LightGBM with SMOTE was 8 when looking at the ROC AUC score metric. In the table which shows the performance using SMOTE we present the ROC AUC obtained with 4 features for XGBoost and with 8 features for LightGBM.

Class Imbalance Not Handled

As in the previous subsection, we firstly evaluate the performance of each classifier without solving the issue of the class imbalance. In Table 6.11, Table 6.12 and Table 6.13 we show the ROC AUC score, MCC score and sensitivity of this experiment.

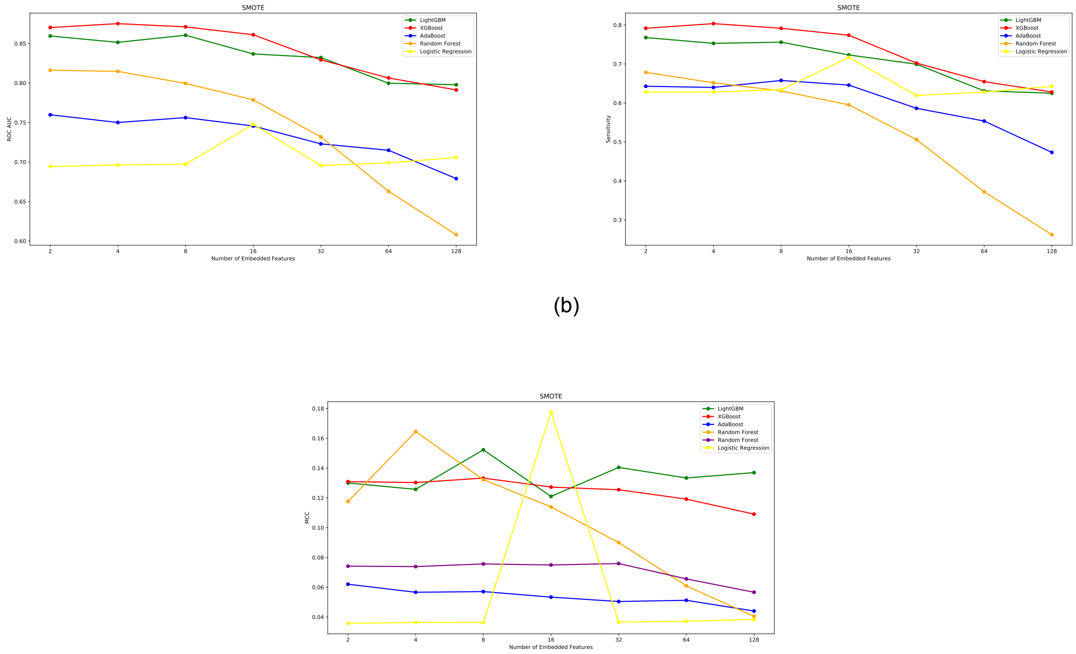


(a)

(b)

(c)

Figure 6.3: The ROC AUC (a), Sensitivity (b) and MCC (c) of the model B + GE when when varying the number of embedded features. These plots correspond to the case in which the class imbalance is handled by weighting the minority class.



(a)

(b)

(c)

Figure 6.4: The ROC AUC (a), Sensitivity (b) and MCC (c) of the model B + GE when when varying the number of embedded features. These plots correspond to the case in which the class imbalance is handled by applying SMOTE.

Classifier	**B**	**B + GE**
Logistic Regression	0.5	0.5
Decision Tree	0.73719	0.72429
Random Forest	0.70944	0.69695
XGBoost	0.7977	0.78752
AdaBoost	0.63533	0.6278
LightGBM	**0.78988**	0.7698

Table 6.11: This Table shows the ROC AUC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + GE
Logistic Regression	-0.00508	0.00053
Decision Tree	0.20342	0.20515
Random Forest	**0.44035**	0.32057
XGBoost	0.11254	0.18263
AdaBoost	0.13222	0.11729
LightGBM	0.10875	0.17678

Table 6.12: This Table shows the MCC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + GE
Logistic Regression	0.10416	0.1369
Decision Tree	**0.48214**	0.45535
Random Forest	0.41964	0.39583
XGBoost	0.63988	0.5744
AdaBoost	0.27678	0.27678
LightGBM	0.625	0.55357

Table 6.13: This Table shows the sensitivity of all classifiers for all the analyzed models. The best result is shown with Bold.

Weighting the Minority Class

In this experiments, we are weighting the minority class as a method of handling the high imbalance between the two classes. We depict the ROC AUC score, MCC score and sensitivity in Table 6.14, Table 6.15 and Table 6.16.

Classifier	B	B + GE
Logistic Regression	0.68669	0.7069
Decision Tree	0.77999	0.78118
Random Forest	0.72647	0.69852
XGBoost	0.76975	0.75258
AdaBoost	0.75378	0.73984
LightGBM	0.7211	**0.80859**

Table 6.14: This Table shows the ROC AUC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + GE
Logistic Regression	0.03817	0.0387
Decision Tree	0.0739	0.0387
Random Forest	0.081209	**0.28976**
XGBoost	0.10276	0.27014
AdaBoost	0.0656	0.0645
LightGBM	0.068	0.07954

Table 6.15: This Table shows the MCC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + GE
Logistic Regression	0.55357	0.64583
Decision Tree	0.65773	0.66071
Random Forest	0.50297	0.44345
XGBoost	0.58333	0.51785
AdaBoost	0.61011	0.51785
LightGBM	0.5119	**0.72023**

Table 6.16: This Table shows the sensitivity of all classifiers for all the analyzed models. The best result is shown with Bold.

SMOTE

In this experiment we are synthetically generating samples of the minority class into the training set in order to tackle the learning problems caused by class imbalance. We display the results of ROC AUC score, MMC score and sensitivity in Table 6.17, Table 6.18 and Table 6.19.

Classifier	B	B + GE
Logistic Regression	0.70206	0.74772
Decision Tree	0.82255	0.82416
Random Forest	0.81743	0.81628
XGBoost	**0.88158**	0.87521
AdaBoost	0.75999	0.75984
LightGBM	0.86212	0.85956

Table 6.17: This Table shows the ROC AUC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + GE
Logistic Regression	0.03637	0.1775
Decision Tree	0.07412	0.074944
Random Forest	0.10553	**0.16453**
XGBoost	0.13345	0.1309
AdaBoost	0.05818	0.06204
LightGBM	0.1299	0.13003

Table 6.18: This Table shows the MCC score of all classifiers for all the analyzed models. The best result is shown with Bold.

Classifier	B	B + GE
Logistic Regression	0.66071	0.71692
Decision Tree	0.77976	0.78571
Random Forest	0.69345	0.67857
XGBoost	**0.81547**	0.80357
AdaBoost	0.66369	0.64285
LightGBM	0.7738	0.76785

Table 6.19: This Table shows the sensitivity of all classifiers for all the analyzed models. The best result is shown with Bold.

Analysis of Results

One important remark after performing these experiments is the fact that a high number of graph embedding features has a tremendous effect on the performances of the classifiers. This is due to overfitting for some of them or inability to learn such high dimensionality for others. Thus, for most of the classifiers we observed that the optimal number of embedding features is between 2 and 16.

When analyzing the experiments' results, we observed the fact that graph embedding features do not bring a significant contribution to the model's performance neither in the situation in which the class imbalance is not handled, nor in the case in which we apply SMOTE. For these two experiments, the best performance was achieved by the baseline. We further observe the fact that the graph embedding features prove to be extremely impactful in the situation in which the minority class is weighted. We can notice the fact that there were 20% more correctly identified defaulted SMEs by the model B + GE compared to the baseline.

Similarly with the previous experiment, the ROC AUC score and sensitivity achieved the best performance for the same model. This could strengthened our belief that these two metrics are correlated and the ROC AUC score is, indeed, sensitive to the percentage of correctly identified defaulted SMEs. Additionally, also in this situation the MCC is not extremely correlated with the sensitivity.

As an overall view, the graph embedding features contributed, indeed, to increase the performance of the baseline when the minority class was weighted. However, they did not manage to improve the best score obtained with the baseline, namely XGBoost with SMOTE. For this reason, we consider that taken alone graph embedding features do not carry meaningful enough information to improve the traditional credit scoring models.

6.2.3. Experiments 3: Hybrid Model - Combined

In this subsection the hybrid model is composed of financial features combined with both network centrality measure features with graph embedding features. As our purpose is to answer our third research subquestion, we compare the performance of this hybrid model with the baseline. In this experiment we are using the following notations:

- **B** - Baseline
- **B + EC** - Baseline + Eigenvector Centrality;
- **B + EC + ND** - Baseline + Eigenvector Centrality + Node Degree;
- **B + EC + CC** - Baseline + Eigenvector Centrality + Clustering Coefficient;
- **B + EC + GE** - Baseline + Eigenvector Centrality + Graph Embeddings;
- **B + EC + ND + CC** - Baseline + Eigenvector Centrality + Node Degree + Clustering Coefficient;
- **B + EC + ND + GE** - Baseline + Eigenvector Centrality + Node Degree + Graph Embeddings;
- **B + EC + CC + GE** - Baseline + Eigenvector Centrality + Clustering Coefficient + Graph Embeddings;
- **B + EC + ND + CC + GE** - Baseline + Eigenvector Centrality + Node Degree + Clustering Coefficient + Graph Embeddings;

Results of Experiments

In this experiment we are not, however, considering all combinations. We are using the best performing hybrid model from the ones that we already evaluated, namely the B + EC model and we are adding the other graph features to this model. Furthermore, only the best classifier, XGBoost, and the best class imbalance method, SMOTE, were considered for this experiment.
We depict our results in Table 6.20 in terms of ROC AUC score, sensitivity and MCC score.

Case	ROC AUC	Sensitivity	MCC
B	0.88158	0.81547	**0.13345**
B + EC *	0.88437	0.82738	0.12754
B + EC + ND *	**0.89414**	**0.84821**	0.12944
B + EC + CC	0.86361	0.78273	0.12379
B + EC + GE *	0.89218	0.84226	0.13087
B + EC + ND + CC	0.8788	0.81547	0.12646
B + EC + ND + GE	0.8748	0.80654	0.12607
B + EC + CC + GE	0.86406	0.78273	0.12487
B + EC + ND + CC + GE	0.8617	0.77678	0.12539

Table 6.20: This Table shows performance of XGBoost with SMOTE on the previously mentioned combined models. We highlight with bold the highest ROC AUC score, sensitivity and MCC score. The models which are shown with * were verified with McNemar's statistical test and their improvement to the baseline is statistically significant with a confidence level of 0.01.

Analysis of Results

When analyzing the combined model, we again observe the same correlation between ROC AUC score and sensitivity. Furthermore, we believe that, due to the fact that the MCC is not sensitive to the percentage of defaulted companies, we do not consider it a good metric to evaluate our performance. In the following subsection we are closely analyzing what type of factors influence this metric. For this reason, we are only focusing on the results according to the ROC AUC score and the sensitivity. Thus, in this situation, the graph embedding features proved to be a significant contribution to the traditional credit scoring model.
We can observe from Table 6.20 that there are 3 types of hybrid models which achieve higher performance than the baseline (B + EC, B + EC + ND, B + EC + GE). We observe that the outcome of adding the CC (clustering coefficient features) is a decrease in accuracy. For this reason, we believe that this features is not meaningful when it comes to predicting the which companies are endangered of default. Furthermore, incorporating the ND (node degree) and GE (graph embeddings) features into the B + EC model results in improving the model's accuracy by around 1% when compared to the baseline. For this reason, we can claim that the combination of network centrality based features and graph embedding features could result in improving the performance of the traditional credit scoring model if the class imbalance issue was handled with SMOTE.

6.2.4. Robustness

In our data-set we have accumulated samples corresponding to 9 years, which were divided into 8 years of training and one year of testing. The way we treated this problem in our evaluation method (the production scenario) was to incorporate samples from 2011 until 2018 into our training set and assess the performance of the model by testing on the samples from 2019.
In our data analysis process which was included into the Data chapter, we argued that the model needs to deal with the up-growing trend of having more defaulted samples in the test compared to the ones in the train. We made this remark due to the fact that we observed that the number of defaulted SMEs is increasing over time.
For this reason, we decided to choose 2019 as the most reliable year to test on. In addition, in credit scoring literature the testing is also performed on the recent year for which labeled data is available. The reason for this is the fact that, although the economy is changing over time, one can observe insignificant changes for consecutive years.
However, we also take into account the situation in which an economical crisis can occur. This is extremely impactful as it can bring significant changes in the economy for two consecutive years. For this reason, we decided that our model needs to be capable of dealing with such a tremendous event. Hence, we considered having an experiment in which we are checking whether the model that we developed is robust.

Results of Experiments

The robustness is assessed by continuously changing the test set from samples of 2019 to samples from other years and training on the remaining ones. For example, if we want to evaluate the performance of the model on the 2012 year, our test set contains samples of 2012, while our train set is composed of samples from {2011, 2013, ... , 2018, 2019}.
In Table 6.21 and Table 6.22 we show the ROC AUC and sensitivity, respectively, obtained from the robustness experiment. Our main purpose in this experiment is to understand whether the models are suitable to correctly predict as many defaulted SMEs as possible. We did not choose to further consider the MCC metrics due to the fact that in the previous experiments it is not extremely sensitive when it comes to how many defaulted companies the model properly predicts. Furthermore, the two tables we also show the number of defaulted SMEs in the training set and in the test set.
We evaluated the robustness of the best performing 4 models that we obtained, namely the baseline, 2 hybrid model which includes network centrality features and one hybrid models which combine network centrality features with graph embedding features.

Test Year	B	B + EC	B + EC + ND	B + EC + GE	No. of Defaulted in Train	No. of Defaulted in Test
2011	0.74647	0.7483	**0.74848**	0.7483	659	2
2012	0.99618	**0.99764**	0.99761	0.5	660	1
2013	0.99671	0.99735	0.99749	**0.99752**	660	1
2014	0.99662	0.9141	0.91434	**0.99735**	655	6
2015	**0.92977**	0.83099	0.89802	0.8312	646	15
2016	**0.96224**	0.9285	0.92897	0.91084	632	29
2017	0.95788	**0.95824**	0.93271	0.92545	584	77
2018	**0.92463**	0.90339	0.90643	0.80927	467	194
2019	0.88158	0.88437	**0.89414**	0.89218	325	336

Table 6.21: This Table shows how robust is each model when different years are considered for test. The first column shows which year is considered for test. The last two columns show the number of defaulted samples in the training and test set, respectively. The evaluation was made by ROC AUC score. We highlight with bold the highest score per year.

Test Year	B	B + EC	B + EC + ND	B + EC + GE	No. of Defaulted in Train	No. of Defaulted in Test
2011	0.5	0.5	0.5	0.5	659	2
2012	**1**	**1**	**1**	0	660	1
2013	1	1	1	1	660	1
2014	**1**	0.8333	0.8333	**1**	655	6
2015	**0.86666**	0.6666	0.8	0.6666	646	15
2016	**0.93103**	0.86206	0.86206	0.82758	632	29
2017	**0.92207**	**0.92207**	0.87012	0.85714	584	77
2018	**0.85567**	0.81443	0.81958	0.61855	467	194
2019	0.81547	0.82738	**0.84821**	0.84226	325	336

Table 6.22: This Table shows how robust is each model when different years are considered for test. The first column shows which year is considered for test. The last two columns show the number of defaulted samples in the training and test set, respectively. The evaluation was made by the sensitivity. We highlight with Bold the highest sensitivity per year. If the highest sensitivity is obtained by different models it is highlighted multiple times. If all the models have the same sensitivity, we do not highlight it.

Analysis of Results

When taking into consideration the ROC AUC score in Table 6.21 we can observe that in most of the cases the highest performance was achieved by combining financial and graph features. However, when also prioritizing a high percentage of correctly identified defaulted SMEs, we can observe in Table 6.22 that the baseline was the best model in most of the cases. The ROC AUC was probably also extremely influenced by the fact that a high number of non-defaulted was properly predicted.

An extremely interesting observation is the fact that the baseline correctly identified defaulted companies when the number of defaulted samples in the training set was much highers than the one in the test set. The baseline performance's in terms of sensitivity starts decreasing with the increase in defaults in the test set. Thus, the baseline model is outperformed by the 3 hybrid models when there are not enough defaults available in the training set.

As an overview, the baseline achieves high results when the training set is composed of a much higher number of defaults compared to the test set. However, the hybrid model is suitable to use when having more defaulted companies in the test set compared to the training set.

As we previously observed while performing the data analysis, the number of defaulted companies increases annually. This could be a consequence of the fact that the scrapped web-site gathers more entries and EOL grows in terms of customers. For this reason, we expect that this trend continues. From the results of our robustness experiment we observed that the hybrid model could handle the situation of having more defaulted samples in the test set compared to the ones in the training set. Given these two aspects we believe that this model would be more suitable than the baseline in detecting SMEs endangered to default. Moreover, in a situation of an economical crisis we would again expect more defaults in the test set. For this reason, we consider the hybrid model to be the most suitable to this problem.

6.2.5. Overall Analysis of Results

We observed in most of our experiments that the best model in terms of ROC AUC score was the best model in terms of sensitivity. However, when it comes to the MCC, the best model according to this metric was different than the previous ones. In order to understand what influences this metric, we plotted the confusion matrix corresponding to the highest MCC, which is model B + CC using Random Forest without handling the class imbalance. In comparison, we also plotted the confusion matrix of the same configuration (without handling class imbalance) for the model which obtained the highest ROC AUC score. Thus, we are observing the differences between two models for which the class imbalance was not handled: B + EC with XGBoost and B + CC with Random Forest. We show the two confusion matrices in Figure 6.5. From Figure 6.5 we can notice that in the situation in which we achieved the highest MCC score, a great majority of the non-defaulted are predicted correctly. Thus, this means that the MCC metric prioritizes more having the majority class properly predicted. This could be also justified by the fact that this metric was created to achieve a high performance only in the situation in which the 4 components of the confusion matrix (TP, TN, FP, FN) are close to their ideal values.

Given the fact that we prioritize the correctly classification of the minority class, we do not consider the MCC metric suitable for our problem. This metric could be, indeed, employed when both classes are

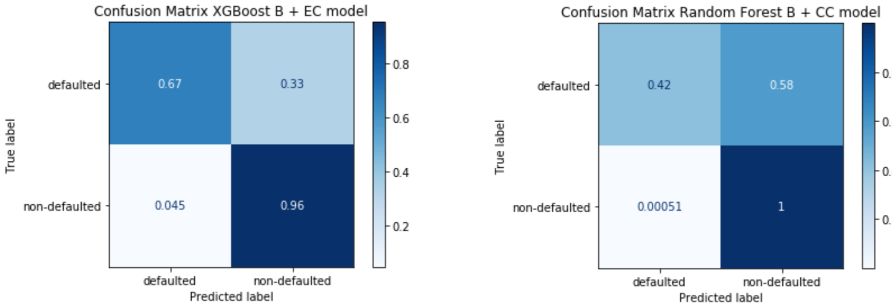


Figure 6.5: Confusion matrix of the best models obtained for the scenario in which the class imbalance is not handled. The left confusion matrix corresponds to the best ROC AUC score and sensitivity obtained. The right confusion matrix corresponds to the best MCC score obtained.

equally important.

7 Discussion

In this chapter we aim to overall discuss the results of our previous experiments. We, thereby, want to highlight the most interesting observations that we made while conducting the experiments.

7.1. Financial Models

The first step in creating a credit scoring model suitable for the Dutch SMEs was an extensive literature review of the existing methods. We employed the Altman model as it proved to be extremely accurate when reproduced by researchers from other countries and we were expecting a similar outcome for our data-set. However, this model was not suitable for our provided financial data, due to its strong limitations when it comes to missing data. We could have, indeed, used the Altman model in our further investigation with the limited amount of samples that we could calculate. However, we considered that it implies a tremendous loss in information when it comes to the defaulted companies. The reason to say this is the fact that we could only include 71 defaulted samples, which is only 10% of the total available defaulted samples.

Our intention was to use this model as baseline, but, considering the aforementioned, we created our own model composed of only financial features. This model is using financial coefficients, which are the nominator and denominator of the financial ratios from the Altman model. Besides these, the financial coefficients model proved to be efficient when predicting SMEs that are endangered of default, even though it contains a different type of financial features compared to the literature in credit scoring. Thus, in our case the financial coefficients proved to be significant key performance indicators of companies. Moreover, we contributed to the credit scoring research by developing a model suitable for Dutch SMEs.

7.2. Hybrid Model

This model was created in order to evaluate the impact of the graph features on the credit scoring model. With this model our contribution to the credit scoring prediction research is twofold. Firstly, most of the credit scoring models are only focusing on the financial status of the companies without taking into account their role in a transactional network. Secondly, within the credit scoring literature which takes into account the partnerships of the companies, Misheva et al. [41], the model is not evaluated on a real-world transactional network due to the inaccessibility of this type of data. Furthermore, we do not only rely on the graph features that were included into this model, but also explore other network features that could bring a substantial contribution.

It is important to mention the fact that we do not place full confidence on the state of the art methodology. We perform a systematic analysis in order to solve the ubiquitous problem of credit scoring prediction, specifically, class imbalance. In this manner, we employ a method of handling class imbalance which was not evaluated in the state of the art, namely, the combination between a domain based undersampling technique and a data based technique (SMOTE). In spite of these, different metrics of assessing the model's performance were critically evaluated based on their outcome. It needs to be mentioned that the accuracy of these metrics was strongly debated within literature, thus we felt the need to also perform this analysis. Moreover, the model was created in such a manner that it can be deployed into the production and used in the following years. Thus, when developing it, we took into consideration the aspect of creating a maintainable product.

When it comes to our results, we examined the performance of both hybrid model and financial coefficients model on the production scenario. We observed the fact that the eigenvector centrality brought significant contribution to the financial coefficients model. We expected this due to the fact that we observed in our previous analysis that the non-defaulted SMEs tend to have a higher eigenvector centrality than the defaulted ones. In terms of interpretability, this means that companies, which are doing business with other companies that have many connections, are more likely to be non-defaulted. In the same manner, the ones which have direct partners that are not linked with many other companies have a higher likelihood to default. This is also an interesting insight into the business workflow. We can observe that the financial stability of one company can be strongly dependent on the stability of its direct counterparties.

We further remarked that, contrary to our expectations, the node degree was not a good addition to the financial coefficients model. Based on the distribution analysis, we observed that non-defaulted companies usually have a high node degree, while the defaulted ones are characterised by a low node degree. However, in our case it can also be the fact that there are also plenty of non-defaulted companies which do not have many connection. We believe that this could be a consequence of the fact that some companies were recently funded and they did not yet have time to gather plenty of customers or suppliers, but their financial status is stable. Another reason for this could also be the fact that those companies probably have other connections that we did not know about due to the fact that they are not Exact customers.

According to our expectations after seeing the distribution, the clustering coefficient was not a good discriminator between defaulted and non-defaulted. Thus, whether companies tend to cluster together is not significant in the credit scoring prediction problem. This could be due to the fact that companies deliver different products, which means that they need suppliers and customers according to the type of product that they manufacture. For example, if a company delivers food it needs electric cooking systems from its electronics supplier. The same supplier is providing electric construction equipment to a construction company. However, it is highly unlikely that a connection could be created between the food delivery company and the construction company. The reason for this is the fact that there is a low chance that they rely on each others' services.

We also observed that graph embeddings did not contribute to increasing the accuracy of traditional credit scoring prediction models. For this reason, a vector representation of a company in a transactional network is not a good indicator of the company's likelihood to default. From our proof of concept analysis, we would expect the contrary. We based our expectations on the fact that we detected some subgraphs of only defaulted nodes and we also showed that the defaulted nodes are closer to each other in distance compared to the non-defaulted ones. However, the embedded location in this case does not take into account the distance between two nodes. As the embeddings are created based on the neighborhood, nodes from the same neighborhood would have similar representations. Thus, even though the distance is smaller between defaulted nodes, they are probably in different neighborhoods. However, there is one limitation that we need to mention. We only generated the embeddings using the DeepWalk configuration due to the time constraints and computational expenses. Node2vec is more flexible than DeepWalk and could probably generate more meaningful embeddings if adjusted accordingly. Thus, a future step is the optimization of graph embeddings by varying the node2vec parameters. Furthermore, a thorough analysis into the meaning of the neighborhood also needs to be performed.

We further considered to combine the network features using the best classifier and class imbalance technique and observe whether they also help improving the accuracy. Thus, we observed that the hybrid model composed of eigenvector centrality and node degree, as well as the hybrid model composed of eigenvector centrality and graph embedding features outperformed the baseline. Furthermore, they also outperformed the hybrid model containing only the eigenvector centrality feature. This means that the best option to assess the importance of a node is by taking into consideration both its number of connections and its direct neighbors' number of connections. Thus, a company which does business with many companies and its counterparties also have plenty of connections is more unlikely to default. Furthermore, also considering the embedding of the location of a node within the network and the number of connections of its counterparties is extremely significant for predicting its default likelihood. This could be the case that the graph embeddings bring additional information of the neighborhood to the number of connections of one company's counterparties.

Last, but not least, we also considered evaluating the robustness of the models. In our production scenario we were training on samples from 2011 until 2018 and testing on 2019. However, the economy is changing over time. Hence we would like to have a model which is able to predict the credit scoring without being biased on one specific year. We consider that introducing samples from different years in our model is a good indicator of robustness. However, we also evaluated this by testing each year and training on the other remaining ones.

Most of our defaulted samples were from 2019. As a consequence, when evaluating on other years the training set contained plenty of defaulted companies, while the test set contained fewer samples of defaults. We observed that a high performance was obtained for the situations in which the training set contained plenty of defaulted samples. This means that our model is robust over years, but also that having as many defaulted samples in the training set plays a key role in predicting credit scoring. We observed that in all cases except for testing on 2019 the financial coefficients model outperformed the hybrid model in terms of the number of correctly predicted samples from the positive class. However, in all of these situations the number of samples in the training was significantly high. Thus, we can conclude that the financial coefficients model achieves high performance when there are enough defaulted samples in the training set. If not, the hybrid model achieves a higher performance. For this reason, we can say that the graph features have a significant contribution when the model is trained on few samples from the defaulted class. In a situation of an economical crisis, there can be expected to have plenty of defaults in the test set, but not enough in the training set. Hence, we consider the hybrid model more robust than the financial coefficients model as it achieves a high prediction accuracy without being provided with plenty of defaulted samples during training.

8 Conclusion

In this report we developed a new credit scoring prediction model which includes both financial and graph features. Our aim was answering the main research question:
MRQ *What is the impact of adding network based features on traditional credit scoring models in terms of effectiveness?*.
Our research question was based on extensive analysis of the transactional network and its relationship with the defaulted and non-defaulted companies. Thus, we did not only considered intuition in formulating our problem, but we also demonstrated that some defaulted companies are connected to each other and in general the average shortest path between the defaulted companies is lower than the average shortest path of non-defaulted companies. By demonstrating these aspects, we showed that the defaulted companies are close in our transactional network. For this reason, we considered that it is appropriate to use graph based features in order to improve the credit scoring prediction.
Graph features is an extremely general term. In this report we classified graph features into: network centrality features and graph embeddings features. The former are metrics which assess the importance of a node in a network, while the latter are methods to translate the graph into a vector space. We further split our main research question into 3 research sub-questions:
SRQ1 *What is the impact of adding network centrality based features on traditional credit scoring models in terms of effectiveness?* For this sub-question, the network centrality features employed were *node degree*, *clustering coefficient* and *eigenvector centrality*. Other metrics were not included due to their high computational expenses. In our experiments we observed that the eigenvector centrality proved to be the most efficient feature, contributing in correctly predicting 83% of the defaulted samples in the test set. Furthermore, the model which included financial features and eigenvector centrality outperformed the baseline model. However, the addition of either the node degree or the clustering coefficient to the financial coefficients model did not improve its accuracy.
SRQ2 *What is the impact of adding graph embedding features on traditional credit scoring models in terms of effectiveness?* When it comes to this sub-research question, we remarked that graph features did not have a positive impact on properly predicting the defaulted class when added to the financial features. They contributed significantly in correctly predicting the non-defaulted class. However, our main focus was precisely classifying as many defaulted SMEs. Thus, we did not consider the model including graph embedding features and financial features a suitable model for our issue.
SRQ3 *What is the impact of adding combined network centrality features and graph embedding features on traditional credit scoring models in terms of effectiveness?* We further performed different combinations of network features in order to test their performance and observe whether they improve the predictions of the baseline. We remarked that incorporating both graph embedding features and network centrality features to our baseline significantly contributes to increasing the model's performance when predicting the situation of 2019. By the means of our robustness experiment, we observed the fact that the baseline outperformed the others when the training set contained enough defaulted samples. However, the model which combines network centrality features and graph embedding features is more suitable to use when not enough defaulted samples are available for training. For this reason, we can conclude that the graph embedding features carry important information. Combining them with the financial features and the eigenvector centrality contributes to creating a model that outperforms

the baseline.

9 Future Work

This chapter aims to give some directions for future research that could be done in order to improve the credit scoring prediction model for Dutch SMEs. We are initially discussing it from a network point of view and then showing a business point of view.

9.1. Network

In our work we are using a transactional network (business network) which is unweighted and undirected. The network shows which companies are doing business with each other. This means that between every two nodes of our network there should be at least one transaction.

Inspired by the work of Wang et al. [56], we believe that a future work would be to explore the benefits of using a weighted graph in stead of unweighted. This could be done by weighting the edges between nodes by the amount of money that was exchanges during each year. We believe that this might bring even more insights into the business workflow and it can also improve the credit scoring prediction model. The motivation behind our assumption is the fact that when having two companies that do business with each other, the default of one might not cause the default of the other if the amount of money exchanged is small. However, if they exchange a large amount of money, the default of one company could have a tremendous effect on the other one.

Another improvement to the network representation is using a directed in stead of an undirected graph. The direction should map the customer-supplier relationship. Thus, the source node is the customer which needs to reimburse a certain amount of money to the supplier, which is the target node. This mapping of customer-supplier relationship by the mean of a directed graph was frequently employed in the financial contagion research and proved to bring significant insights. The motivation for considering this is the fact that in a customer-supplier relationship, the supplier is usually the one affected by the default of customers due to their inability to return their payment.

An additional advantage of using a directed network is lowering the computation required to embed the network. This is a consequence of the fact that the number of random walks required by the framework is restricted according to the direction. This also results in decreasing the computational expenses of performing hyper-parameter tuning for node2vec. Thereby, the generated embeddings could bring even more information than the ones that we used for this project.

9.2. Business

In this section we want to analyze how the model can be improved from the business point of view. We are starting by firstly explaining some of the issues that Exact is facing with missing data and then justify how they correlate with our work and affect our results.

As we mentioned in this project, we calculate our financial coefficients based on the RCSFI codes. These codes are generated from mapping transactions which were registered by accountants and bookkeepers in EOL. However, the EOL software does not impose restrictions to its users when it comes to the deadline of introducing the transactional records. In consequence, many records are delayed and this leads to having missing transaction registered. Thereby, some RCSFI codes have incomplete values. This can affect our model as we rely on the RCSFI codes to calculate the financial features.

One possible solution to this issue is *now-casting*. Now-casting is a forecasting method that could be employed to predict the incomplete records. This technique could be applied in order to overcome the problem of missing transactions recorded and could lead to an accurate value for the RCSFI codes. Thus, one possible improvement of this model is to firstly predict the missing values of RCSFI codes using now-casting and, thereafter, utilize the predicted RCSFI in order to calculate the financial coefficients. We believe that it can be possible that some of the companies are mistakenly predicted as defaulted due to the fact that their financial coefficients features were not complete, which made them similar to the defaulted SMEs samples. Thus, integrating now-casting for predicting late transactions into our model could contribute to increasing its prediction accuracy.

A

Appendix

Financial coefficients according to RCSFI 1.1.

EBITDA

Formula:

EBITDA = revenue - direct costs - indirect costs

RCSFI codes:

Revenue - (-1)*WOmz + (-1)*WOvb
Direct Costs - WKpr
Indirect Costs - WAad + WAfs + (-1)*WBbe + WBed + WBel + (-1)*WFbe + (-1)*WNer + WPer + (-1)*WVhe + WWiv
EBITDA = (-1)*WOmz + (-1)*WOvb - WKpr - WAad - WAfs + WBbe - WBed - WBel + WFbe + WNer - WPer + WVhe - WWiv

Total Assets

RCSFI Codes:

Total assets = BIva + BMva + BFva + BEff + BLim + BVor + BVrd + BPro

Short Term Debt:

RCSFI Codes:

Short Term Debt = (-1)*BSch

Equity Book Value:

RCSFI Codes:

Equity Book Value = (-1)*BEiv

Cash:

RCSFI Codes:

Cash = BLimKas

Interest Expenses:

RCSFI Codes:

Interest Expenses = WkprKra + WBedKse + (-1)*WFbeOrb + WFbeOrl + WFbeRls

Retained Earnings:

Formula:
Retained Earnings = revenue - direct costs - indirect costs -taxes
RCSFI Codes:
taxes = (-1)*BSchBtw + (-1)*BSchVpb + (-1)*BschOvb;
Retained Earnings = (-1)*WOmz + (-1)*WOvb - WKpr - WAad - WAfs + WBbe - WBed - WBel + WFbe + WNer - WPer + WVhe - WWiv + BSchBtw + BSchVpb + BschOvb

Working Capital:

Formula:
Working Capital = current assets - short term debt
RCSFI Codes:
Current Assets = BLim + BEff + BVor + BPro + BVrd;
Working Capital = BLim + BEff + BVor + BPro + BVrd + BSch

Total Liabilities:

RCSFI Codes:
Total Liabilities = (-1)*BVrz + (-1)*BLas + BSch

EBIT:

EBIT = EBITDA - taxes
RCSFI Codes:
EBIT = (-1)*WOmz + (-1)*WOvb - WKpr - WAad - WAfs + WBbe - WBed - WBel + WFbe + WNer - WPer + WVhe - WWiv + (-1)*BSchBtw + (-1)*BSchVpb + (-1)*BschOvb

Bibliography

[1] Comitato di basilea per la vigilanza bancaria. *International convergence of capital measurement and capital standards: a revised framework. Bank for International Settlements*, 2004.

[2] Josephine Sarpong Akosa. Predictive accuracy : A misleading performance measure for highly imbalanced data. 2017.

[3] Mousa Albashrawi. Detecting financial fraud using data mining techniques: A decade review from 2004 to 2015. 2016.

[4] Edward I. Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. 1968.

[5] Edward I. Altman. Default recovery rates and lgd in credit risk modelling and practice. 2011.

[6] Edward I. Altman and Gabriele Sabato. Modeling credit risk for smes: Evidence from the us market. 2007.

[7] Edward I. Altman, Maurizio Esentato, and Gabriele Sabato. Assessing the credit worthiness of italian smes and mini-bond issuers. 2020.

[8] Jackson Arroyave. A comparative analysis of the effectiveness of corporate bankruptcy prediction models based on financial ratios: Evidence from colombia. 2018.

[9] Sofie Balcaen and Hubert Ooghe. 35 years of studies on business failure: an overview of the classic statistical methodologies and their related problems. *British Accounting Review*, 38:63–93, 2006.

[10] Ádám Banai, Gyöngyi Vargáné Körmendi, Péter Lang, and Nikolett Vágó. Modelling the credit risk of the hungarian sme sector. 2016.

[11] Yaneer Bar-Yam. *Dynamics of Complex Systems*. Perseus Books, USA, 1997. ISBN 0201557487.

[12] Joao Bastos. Credit scoring with boosted decision trees. MPRA Paper 8034, University Library of Munich, Germany, April 2007. URL `https://ideas.repec.org/p/pra/mprapa/8034.html`.

[13] Peter J. Brennan. A comprehensive survey of methods for overcoming the class imbalance problem in fraud detection. 2012.

[14] Iain Brown and Christophe Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Syst. Appl.*, 39:3446–3453, 2012.

[15] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *CIKM '15*, 2015.

[16] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

[17] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *ArXiv*, abs/1603.02754, 2016.

[18] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21, 2020.

[19] Stephen Chong, Rutger Hoekstra, Oscar Lemmers, Ilke Van Beveren, Marcel Berg, Ron Wal, and Piet Verbiest. The role of small- and medium-sized enterprises in the dutch economy: an analysis using an extended supply and use table. *Journal of Economic Structures*, 8, 12 2019. doi: 10.1186/s40008-019-0139-1.

[20] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31:833–852, 2019.

[21] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998. doi: 10.1162/089976698300017197.

[22] Giovanna Ferraro and Antonio Iovanella. Technology transfer in innovation networks: An empirical study of the enterprise europe network. *International Journal of Engineering Business Management*, Vol. 9:1–15, 11 2017. doi: 10.1177/1847979017735748.

[23] Prasanna Gai and Sujit Kapadia. Contagion in financial networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466:2401 – 2423, 2010.

[24] Paolo Giudici, Peter Sarlin, and Alessandro Spelta. The interconnected nature of financial systems: Direct and common exposures. *Journal of Banking Finance*, 05 2017. doi: 10.1016/j.jbankfin.2017.05.010.

[25] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[26] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 855–864. ACM Press, 2016. doi: 10.1145/2939672.2939754.

[27] Jairaj Gupta, Nick L. Wilson, Andros Gregoriou, and J. V. Healy. The effect of internationalisation on modelling credit risk for smes: Evidence from uk market. 2014.

[28] Jairaj Gupta, Andros Gregoriou, and Tahera Ebrahimi. Empirical comparison of hazard models in predicting bankruptcy. 2016.

[29] Guo Haixiang, Yijing Li, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Bing Gong. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73, 12 2016. doi: 10.1016/j.eswa.2016.12.035.

[30] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*, 2017.

[31] Usama Ehsan Khan. Bankruptcy prediction for financial sector of pakistan: Evaluation of logit and discriminant analysis approaches. 2018.

[32] Gary King and Langche Zeng. Logistic regression in rare events data. *Political Analysis*, 9:137–163, Spring 2001.

[33] Joffrey L. Leevy, Taghi M. Khoshgoftaar, Richard A. Bauder, and Naeem Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5:1–30, 2018.

[34] Nadine Levratto. From failure to corporate bankruptcy: a review. *Journal of Innovation and Entrepreneurship*, 2:1–15, 2013.

[35] Cong Li, Huijuan Wang, W Haan, C. Stam, and Van Mieghem. The correlation of metrics in complex networks with applications in functional brain networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011, 11 2011. doi: 10.1088/1742-5468/2011/11/P11018.

[36] Cong Li, Qian Li, Piet Mieghem, H. Stanley, and Huijuan Wang. Correlation between centrality metrics and their application to the opinion model. *The European Physical Journal B*, 88, 09 2014. doi: 10.1140/epjb/e2015-50671-y.

[37] J. Liebig and Asha Rao. A clustering coefficient to identify important nodes in bipartite networks. 06 2014.

[38] Rushi Longadge and Snehalata Dongre. Class imbalance problem in data mining review.

[39] Rosario N. Mantegna and H. Eugene Stanley. *An Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge University Press, USA, 1999. ISBN 0521620082.

[40] Chandresh Kumar Maurya, Durga Toshniwal, and Gopalan Vijendran Venkoparao. Online anomaly detection via class-imbalance learning. *2015 Eighth International Conference on Contemporary Computing (IC3)*, pages 30–35, 2015.

[41] Branka Hadji Misheva, Paolo Giudici, and Valentino Pediroda. Network-based models to improve credit scoring accuracy. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 623–630, 2018.

[42] Ajinkya More. Survey of resampling techniques for improving classification performance in unbalanced datasets. abs/1608.06048, 2016.

[43] Nada Mselmi, Amine Lahiani, and Taher T. Hamza. Financial distress prediction: The case of french small and medium-sized firms. 2017.

[44] Xuetong Niu, Li Wang, and Xulei Yang. A comparison study of credit card fraud detection: Supervised versus unsupervised. *ArXiv*, abs/1904.10604, 2019.

[45] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. *ArXiv*, abs/1403.6652, 2014.

[46] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, pages 701–710. ACM Press, 2014. doi: 10.1145/2623330.2623732.

[47] Andrea Dal Pozzolo, Olivier Caelen, and Gianluca Bontempi. When is undersampling effective in unbalanced classification tasks? In *ECML/PKDD*, 2015.

[48] Francisco Aparecido Rodrigues. Network centrality: An introduction. *arXiv: Physics and Society*, pages 177–196, 2019.

[49] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, 10, 2015.

[50] Tamer Mohamed Shahwan and Maisara Ahmed Fadel. Machine learning models and financial distress prediction of small and medium- sized firms: Evidence from egypt. 2020.

[51] Maria Sophocleous. Access to credit for small business in south africa towards a value-based decision framework. 2019.

[52] Martin Summer. Financial contagion and network analysis. *Annual Review of Financial Economics*, 5(1):277–297, 2013. doi: 10.1146/annurev-financial-110112-120948.

[53] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. *ArXiv*, abs/1503.03578, 2015.

[54] Lei Tang and Huan Liu. Leveraging social media networks for classification. *Data Min. Knowl. Discov.*, 23:447–478, 11 2011. doi: 10.1007/s10618-010-0210-x.

[55] Beaver W. Financial ratios as predictors of failure. *Journal of Accounting Research (Supplement)*, 4:71–111, 1966.

[56] Huijuan Wang, Edgar Boven, A. Krishnakumar, M. Hosseini, Hans van Hooff, Tamihiro Takema, Nico Baken, and Piet Mieghem. Multi-weighted monetary transaction network. *Advances in Complex Systems*, 14:691–710, 10 2011. doi: 10.1142/S021952591100330X.

[57] Maoguang Wang, Jiayu Yu, and Zijian Ji. Credit fraud risk detection based on xgboost-lr hybrid model. 2018.

[58] Naoyuki Yoshino and Farhad Taghizadeh-Hesary. A comprehensive method for the credit risk assessment of small and medium-sized enterprises based on asian data. 2018.

[59] Arata Yoshiyuki. Bankruptcy propagation on a customer-supplier network: An empirical analysis in japan. 2018.

[60] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE Transactions on Big Data*, 6:3–28, 2020.