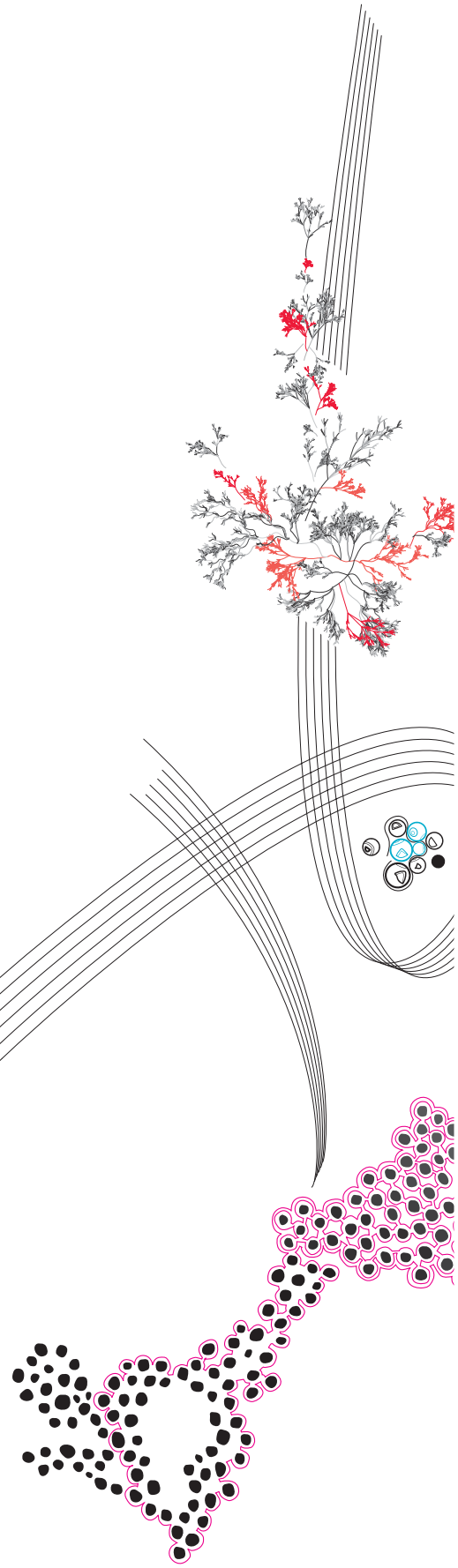

MSc Industrial Engineering and Management

Master Thesis

Predicting the Credit Risk Classification of Retail clients at Rabobank

Sander Constantijn Bijn van den Berg

1st Supervisor UT: Laura Spierdijk
2nd Supervisor UT: Berend Roorda
1st External Supervisor: Frank van Ingen
2nd External Supervisor: Liam Eykhout

February, 2024

Department of Industrial Engineering and Management
Faculty of Behavioural, Management and Social Sciences
Financial Engineering and Management Specialization
University of Twente

UNIVERSITY OF TWENTE.

Contents

Abstract

This thesis undertakes a comprehensive case study focusing on the development of a prediction model aimed at classifying the prospective credit risk of retail clients within Rabobank. Rabobank considers 4 ordinal Credit Risk Classification (CRC) classes: CRC [Good], CRC [Early Warning], CRC [Financial difficulties] and CRC [Default]. The primary objective is to enrich the existing early warning credit risk monitoring system, SAMAS, by incorporating a machine learning-based forward looking approach. The central research question driving this thesis is: "How can machine learning algorithms be applied to predict the CRC class of retail clients at LBBB three months into the future for clients currently on the watchlist as CRC [Early Warning]?"

The problem approach involves an extensive systematic literature review (SLR) delving into the concept of Machine Learning in Early Warning Systems and Credit Risk. These findings, coupled with insights from the data review, formed the empirical study design. This thesis strives to contribute to the field of the multi-class classification problems characterized by imbalanced classes and within the context of credit risk. The research deploys a branch of under- and oversampling techniques, in conjunction with a multitude of Machine Learning techniques including Random Forest (RF), XGBoost (XGB), Support Vector Machine (SVM), Feedforward Neural Networks (NN), Multinomial Logistic Regression (MLR), and Linear Discriminant Analysis (LDA), which are compared to a Naive Bayes model. Hence in total 28 models are compared and an optimal model for this RSME client dataset is selected.

The evaluation of these models hinges on two accurateness perspectives. One describes an accurate model to be a model that classifies each class equally well while the other accounts the pure accurateness of the prediction model. It becomes apparent that the models struggle to predict the minority class, CRC [Default]. Because of this, the problem is restructured as a 3-class multi-class classification problem by rewriting all CRC [Default] cases to CRC [Financial Difficulties]. This mutation yields to a model that show promising results that align with our definition on an accurate model. The optimal model emerges as an ensemble method combining Random Forest, eXtreme Gradient Boosting, Support Vector Machine, and Multinomial Logistic Regression. Subsequently, this final model undergoes additional assessments across another time generalization, boasting an average balanced F-1 score of 0.7005 and an average weighted F-1 score of 0.7429 with a confident balanced avg. precision of 0.8252. The results affirm the model's potential to generalize over time and indicate potential cost savings for practical implementation. A proof of concept is also developed on how the extension can be implemented into practice. Nevertheless, further research, such as window forward cross-validation, is imperative to establish a more confident feasibility of its practical use.

In conclusion, this research presents a valuable practical contribution to the field of credit risk analysis, while it also demonstrated its solutions potential to forecast the CRC with a proof of concept. The innovative tool enhances the current risk monitoring systems and can be utilized to form risk mitigation strategies. Additionally, it underscores theoretical contributions applicable to the broader banking industry, emphasizing the potential of machine learning in reshaping credit risk early warning monitoring systems.

Keywords: Prediction model, Rabobank, Machine Learning, Early Warning System (EWS), Credit Risk Credit Risk Classification, Multi-class classification problems

List of Abbreviations

- AI - Artificial Intelligence
- AUC - Area Under The Curve
- BCBS - Basel Committee on Banking Supervision
- CRC - Credit Risk Classifier
- DL - Deep Learning
- DPD - Days Past Due
- DT - Decision Tree
- EAD - Exposure at Default
- ECB - European Central Bank
- EW - Early Warning
- EWS - Early Warning System
- FD - Financial Difficulties
- RFE-RF - Recursive Feature Elimination - Random Forest
- GS - Global Standards
- LBBB - Lokale Banken Bedrijf Business
- LR - Logistic Regression
- MCC - Matthews Correlation Coefficient
- ML - Machine Learning
- NN - Neural Networks
- NPL - Non-Performing Loans
- PD - Probability of Default
- RF - Random Forest
- QA - Qualative Assesment
- RC - Regulatory Capital

- RRR - Rabobank Risk Rating
- RSME - Retail Small Medium Enterprise
- RWA - Risk-Weighted Assets
- SCE - Single Credit Exposure
- SLR - Systematic Literature Review
- SVM - Support Vector Machine
- XGB - XG-Boost

Chapter 1

Introduction

1.1 Credit Risk Class monitoring with Early Warning Systems

Commercial banks are a type of financial institution that provide financial services such as storing deposits, making business loans, and offering basic investment products. The services that include credit expose institutions to credit risk i.e. in the event of non-performing loans [29]. It is therefore vital that commercial banks monitor the credit risk of client adequately and continuously [30].

Continuous and adequate monitoring aims to detect problematic credited obligors as early as possible. Adequate and continuous monitoring of the credit risk exposure on client level does not only provide more internal financial stability to the firm, but also contributes in meeting regulatory demands. One of the methods used by banking institutions to detect, adequately monitor and report on these obligors are early warning systems (EWS) [39]. EWS use triggers and financial credit risk metrics to detect obligors in financial distress as early as possible. Triggers act as binary flag variables that activate when certain conditions are met, while credit risk metrics denote a non-negative variable that is continuously present on client level. Based on this data, clients are automatically classified in the severity of their financial distress position. Depending on the classification, timely actions – such as forbearance measures – can be taken by the bank in order to avoid or mitigate losses. Classification can be conducted on a multitude of classes. In general, a high probability of loan repayment - or similarly a low probability of default - are classified in a financial stable client group and customers with a high probability of default are classified in the financially distressed client group [4] . In general the class that concerns a classification of first signs of financial distress can be considered watchlist classes, note that this definition may vary per financial institution.

Banks with efficient use of EWS can reduce unsecured exposures for clients on a watchlist by about 60 percent within nine months, whereas average banks reduce only around 20 percent unsecured-exposure [6]. Therefore, successful implementation of EWS can significantly reduce lost exposure due to defaults. While at the same time, a timely alert and financial guidance proactively stimulates the bank-client relationship. Besides it contributes from a compliance perspective to cope with the Basel rules imposed by the Basel Committee on Banking Supervision (BCBS), the implications of these rules in the credit monitoring system are discussed in chapter 2. The BCBS is not a regulatory body in itself but a forum for cooperation among banking supervisory authorities [9]. It sets international standards and guidelines for banking regulations on (credit) risk management. The European Central Bank (ECB) takes on a supervisory role and provides unbinding

guidance in the regulation on these standard and guidelines, but the underlying financial institution is responsible for their own methods and standards on monitoring and reporting credit risk [9]. This case-study focuses on the credit monitoring at Rabobank Group.

In the following section, we explore positioning of Rabobank's credit monitoring system within the case study's problem context.

1.2 Problem Context

Rabobank Group is a Dutch multinational commercial banking and financial services company with the second-largest total assets in the Netherlands. Rabobank contributes to resp. 17% and 35% on the domestic market shares in mortgages and savings [36]). Private Loans such as mortgages are only a small part of the credit portfolio at Rabobank, other asset classes involve retail small medium enterprises, wholesale, rural and some other. Rabobank Group also monitors the credit-risk portfolio of its other business units such as The Lage Landen and Obvion. Rabobank EWS is called the Credit Risk Classifier CRC. The CRC considers four classes that indicate the severity of the clients credit risk, in ascending order of severity Rabobank considers CRC: [Good (G)], [Early Warning (EW)], [Financial Difficulty (FD)] and [Default (D)] [43]. From now on, we refer to a status as class and denote the specific class i.e. good as CRC [G] - without a hyperlink to the abbreviation. In figure 1.1, an ordinal overview of the CRC classes is presented.

FIGURE 1.1: This figure shows the four CRC classes in ascending (left to right) order from severity and is taken from the internal report Global Standards on Credit Risk Parameters [43]

The class is mostly determined based on the presence of triggers. Triggers - as indicated earlier - function as a binary flag mechanism that indicate if certain corresponding thresholds - by means of a change in underlying credit risk metrics over time - are breached. Triggers are mostly automatically computed, but can also be manually added to the clients data profile by a portfolio holder at Rabobank. When (a set of) certain triggers are reported, an automatic CRC transition between the four possible statuses immediately occurs [43]. This research focuses on if and when transitions occur, therefore the so-called entry and exit criteria for each CRC status - essentially a decision tree following pre determined rules - will not be deeply discussed in the research. However, if necessary, a comprehensive flowchart including a set of rules of these criteria is available in Appendix 1. For now it is important to understand what a certain CRC class defines. CRC [G] follows from no indication that the client is in financial distress. A transition to CRC [EW] shows first financial distress i.e. a trigger indicating that the clients payment is 30 days overdue, but to an extent that no forbearance measures are given. CRC [FD], the CRC [EW] class has become more severe, additional triggers e.g. a change in Rabobank Risk Rating (RRR) status has been flagged and forbearance measures are applied to the client. CRC [D] is labeled to a client when forbearance measures fail to get the client out of financial distress and the last measures are taken to avoid default or too mitigate losses. A client should "ideally" follow through all of these statuses before being labeled as CRC [D] but in reality, some triggers allow clients to immediately transition to default [35]. Furthermore although the classification are ordinal, its true placement is uncertain. We therefore, don't assume true ordinality within the classification problem. The question arises, can we foresee that

immediate transitions occur and does the reported data on client level posses predictive power to accomplish this? To do so we will check if a snapshot of one years date in time contain information on potential transitions 3 months into the future. An answer on this question remains central in this research and will be elaborated to further extent in the next section: the problem statement.

1.3 Problem Statement

We discussed that CRC automatically classifies clients their risk status based on set of triggers immediately in real-time. The set of triggers and other relevant financial data on clients are assembled, accessed and reported through one single application: SAMAS. SAMAS is Rabobank's application that ensures effective internal reporting to monitor on the CRC of the client, the triggers and other credit risk metrics. The triggers form the basis on how classes take form. I.e. certain triggers can indicate in realtime that thresholds are breached and thus the client (likely) shows financial distress. The watchlist class at Rabobank is CRC[EW], and functions as a temporary class. It is expected that these clients soon have to be transitioned to their actual corresponding class where reasonable possible. This transition occurs both automatically and manually based on qualitative assessment or by means of additional triggers.

Currently, the systems shows to effectively monitor their exposure to credit risk [36] and the CRC has no drastic problem required solving. However, we can provide a (problem) statement on potential extensions to the SAMAS application. It is important to take into account that such additions remain in line with SAMAS objectives - which rely on transparent and objective reporting. Additionally, the defined objectives should not merely focus as a consulting report for Rabobank, but also reflect on general monitoring and reporting methods of credited clients at financial institutions – using Rabobank's CRC as context - and contribute to the literature on effective credit monitoring and EWS.

Aligned with these objectives, we propose to implement a predictive mechanism into SAMAS to predict the CRC, by using data available to assess the client's prospective risk. It is imminent that thee CRC transition occur automatically, as a result of triggers reported, in real-time. Triggers, in their place, only occur when threshold are breached over time. So real-time predictions based on a snapshot of data have no useful impact as there is no uncertainty there. However, an attempt can be made to use the historical data available to discover patterns in certain prospective CRC transitions. Furthermore, additional data that is not used for the initial CRC[1] determination but which is available on client level can also be utilized to enhance the predictive power in pattern recognition i.e. the operating sector and the current absolute value of the exposure at default (EAD). While it also improves useful insights on the use of these metrics. Additionally, literature shows that by implementing ML into big data, accurate predictions on classification can be realized [14] [18], more about the approach is discussed in the next chapter.

Concluding, the problem statement is to be interpreted as an innovation to the current system. It aims to make the EWS more effective in indicating early financial distress of clients by predicting the class they are likely to be in into the future. In the next chapter we dive into how we approach the innovation - we remain to call it the problem approach - by means of objectives, research question and a research design. Also, we asses the scope of the clients investigated in this research.

[1] Last hyperlink for CRC Abbreviation

1.4 Research Objectives

The primary objective is to accurately predict the CRC class of RSME (LBBB) clients three months into the future based on the clients data. In order to achieve this, we train a multitude of Machine Learning (ML) algorithms to asses if patterns between the data exists, hence separability of classes can be determined. The best performing ML algorithm can then be selected. The selected subset of data that is considered significant can be further analyzed to gain insights on the most important predictors to asses the CRC [Class] of the client. While CRC transitions in real-time are solely based on triggers, more data is available on client level and can thus also be exploited in the prediction model, further improving timeliness. It also contributes to answer what defines an adequate prediction model. Furthermore, we want to gain insight if the implementation of the prediction model truly contributes to practical insights and is feasible to implement into SAMAS. In other words, does the model generalized over time and can losses be mitigated by extending the EWS with the prediction model. A Proof of Concept is also made to emphasize its potential and feasibility.

In short, we formulate the following objectives to contribute to the credit risk monitoring system:

- **Forward Looking Approach:** Currently, SAMAS primarily includes backward looking approaches on analysing the CRC. The proposed extension contributes to a forward looking approach on the prospective risk of the client by means of forecasting. This method incorporates historical data of clients to discover patterns used to predict a CRC Classification 3 months into the future.
- **Feature Importance:** Investigate which features (predictors) are considered to contribute on the separability of CRC [Classes]. And also, highlight if additional data - apart from triggers - contribute to the separability of CRC [Classes].
- **Timeliness:**: CRC classification is conducted in real time. This means that classification is done instantly and involves no uncertainty. The proposed extension attempts to detect a CRC transition at an earlier stage, which enables Rabobank to undertake forbearance measures in advance.
- **Practical insights:** The predictions can be tested on associated costs and measure if losses can be mitigated by following the prediction models outcome, improving the watchlist functionality. Also, a proof of concept is developed to stimulate its potential use.

1.5 Research Questions

Based on the aforementioned research objectives, we define the following main research question:

"How can machine learning algorithms be applied to predict the CRC status of RSME clients at LBBB three months into the future?"

The main research question is divided into a number of subquestions, each subquestion is either answered by means of at least one of the following: Literature Research (L), Modelling (M), Data Analyses (DA) and Internal expertise opinions (I). Additionally, the

chapter in which the research question is (partly) answered is denoted next to the corresponding subquestion.

1. What is the existing literature on the topic so far and what are the results found?
    a. To what extent is machine learning applied in monitoring credit risk? What are possible interesting algorithms to include? (L) - Chapter 2
    b. to what extent are EWS used to monitor credit risk? (L) - Chapter 2
2. What scope should be defined and how what properties does the data within this scope have?
    a. What scope should be defined and where should the data be filtered on? (I) Chapter 1.7
    b. How does the raw data and the properties of the features look like? (DA) Chapter 3.4
    c. What type of predictors have been proven to be effective or present in current CRC transitions? (DA) (L) (I) Chapter 3.4
3. How can a prediction model be created to classify the clients based on their prospective risk?
    a. What constitutes an adequate prediction model? (L) Chapter 2
    b. What kind of pre-processing methods are necessary for the dataset to conduct? (L) & (M) Chapter 4.1 & 4.2 & 5.1
    c. Which ML and DL methods can be used in credit risk and how can we implement these? (L) & (M) Chapter 3.4 & 4.2 & 5
    d. What kind of evaluation methods and metrics are suitable to reflect on the predictive power for each of the models? (L) Chapter 4.3 & Chapter 5.6
4. How can we implement the prediction model into SAMAS?
    a. How are the outcomes of the model best compared against the current performance? (L) Chapter 5.6
    b. How can the model be implemented into SAMAS? And how is the intuition behind the model best explained for practical use? (L) & (I) Chapter 6.4 6.5
5. How are the results interpreted?
    a. How can we back test the results to confirm consistent model performance? (L) & (M) Chapter 6.3
    b. What kind of implication does the result have on both practical and theoretical insights? (DA) & (L) Chapter 7.3 & 7.2

1.6 Research Design

We organize our research systematically by employing "The Standard Thesis Structure" outlined by David Evans in his work cited as \cite{Evans2014}. This structure helps us segment our content into distinct chapters. The structure – found in figure 1.2 - serves as

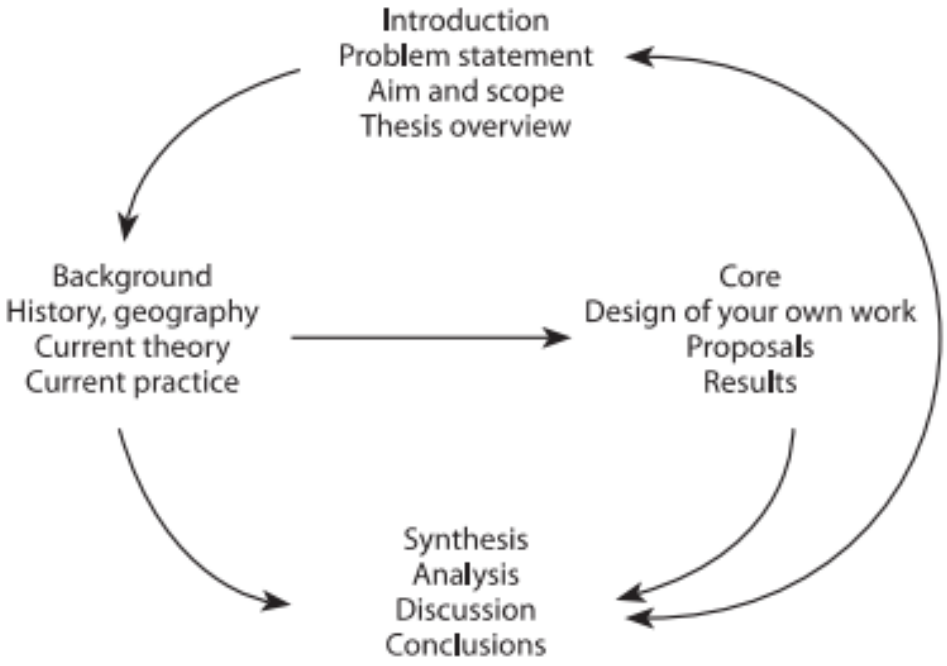

FIGURE 1.2: Structed research design of the thesis according to the Standard Thesis Approach by David Evans [20]

a foundation for this research and is useful for keeping the project orderly as it unfolds. We follow the steps as proposed by Evans as follow. Prior to the introduction, an outline is given that briefly states what is discussed on chapter level. The introduction, problem statement, aim and scope has been put in the introduction chapter. In the first chapter we stated that no direct problem is indicated by Rabobank but we propose a innovation the on current application by extending it with a predictive mechanism in.

Consequently in the same chapter, we establish an approach, that includes research objectives and research questions. The design strategy, as part of the research questions, encompasses the methods used to answer the research question by means of either literature research, results from modelling, data analyses or internal expert opinions.

From this point we start with specifying the scope, background and related work. First related work - depcited as background, history and current theory - is investigated in chapter 2 and is deployed by use of a systematic literature review (SLR). Then, we dive into the data properties by means of a data review.

After, we execute the empirical study design by Akkad [3]. This study design gives insight on the implementation of machine learning classification problems. We start by discussing technical intuition behind the applied methods in chapter 4 Consequently, we walk through the empirical implementation in chapter 5. Then model performance across the different machine learning algorithm is presented in chapter 6. The conclusions are discussed in chapter 7, here we also reflect on the research objectives and the main research question. In the same chapter, we end with the implications of the prediction model both in terms of its limitations and the theoretical and practical contribution.

1.7 Thesis Scope

In this section we elaborate on the chosen scope and answer research question 2.a. In section 1.2 we already stated that entry and exit criteria of CRC classification at Rabobank depend on both the assets class and business unit. Rabobank distinguishes several different asset Classes in their entire credit portfolio. Each asset class has its own level in credit granting, credit monitoring and credit reporting. A mixture of assets classes in the data set is not desired, as this also takes on a mixture of different guidelines and criteria assessing the CRC [class]. Therefore, this research solely takes on the data on the asset class Retail Small Medium Enterprise RSME.

From a similar perspective, including a mixture of different business units in the dataset is also not desired. Because of this, our research also filters on clients from the business unit "Lokale Banken Bedrijven Business" (LBBB).

Additionally, we focus on clients that are already in CRC [EW], as these clients are already on the watchlist and therefore expected to make a transition in the near future.

Ultimately, this means that we focus on RSME clients from the business unit LBBB that are currently in CRC [EW] and predict their CRC 3 months into the future.

Chapter 2

Literature review

Credit risk is most simply defined as the potential that a borrower or obligor will fail to meet its obligations in accordance with agreed terms [32], when this occurs we speak of an obligor in default. According to LCD, the 10-year default rate average for US Leveraged Loans is 1.57% [13]. Without any proper intervention or forbearance measures, defaults can result in huge and permanent losses for the issuer. To mitigate the risk of default, certain credit-risk metrics are monitored. Monitoring of these metrics is either done manually or automatically [41].

With the immense growth in the area of artificial intelligence in the past two decades, ML is becoming an interesting method to contribute to monitoring credit decision making. Such monitoring systems involve EWS, capable of indicating clients in potential financial distress and default [24]. There are already internal credit scoring systems and credit rating agencies that provide their analysis of a customer to banks, but the exploration of using various ML techniques to improve the accuracy level of these ratings is still limited [8]. Predictive ML techniques are capable to identify the pattern behavior and classification of the imminent credit risk that potential clients pose to the financial institutions [45]. However, one of the significant challenges that has been integral to the method is about the alignment of the ML models to the internal monitoring, reporting and information systems that can support overall system enhancement and comprehensibility [40].

For this reason, a systematic literature review (SLR) on the implementation of ML in credit risk is conducted. This SLR attempts to reflect the research questions 1.1 as defined in section 1.5: To what extent is machine learning applied in monitoring credit risk and what are feasible algorithms to include? The SLR found 342 articles in Scopus using the following search terms: (TITLE-ABS-KEY (machine AND learning) AND TITLE-ABS-KEY (bank) AND TITLE-ABS-KEY (credit AND risk) OR TITLE-ABS-KEY (early AND warning)). Filtering on Language: “English”, selecting Keyword: “Credit Risk”, Filter on Relevance (TOP 25), excluding topic: “Credit Card” and include: accessible open source literature. This resulted in a total of 10 papers suitable for comparison. From this SLR we conclude that the SLR is somewhat limited as most articles do not solely focus on the RSME scope but to loans in general. Another conclusion is that no direct relationship to EWS or similar CRC classes are present in many of the literature as most articles focused on other types of prediction models than a classification problem i.e. probability of default. Regarding the methods used in model development, most papers followed a similar theoretical framework as developed by [3]: Pre-processing, feature selection, cross fold, multiple machine learning algorithms and evaluation methods are included in the

same order. Also, we found that most articles are not older than 2020, indicating a recent uprise of machine learning applications in credit risk.

Additionally, we include papers that conduct similar SLRs on the topic [8][39]. We combine our own SLR together with these SLRs and find that the majority of the papers include models based on first taking the best features from a feature selection method and by investigating potential correlations between variables. Consequently they discuss on supervised machine- and deep learning methods including Multinomial Logistic Regression (MLR), Linear Discriminant Analyses (LDA), Support-Vector-Machine (SVM), Decision Tree (DT), Random Forrest (RF), Neural Networks (NN), XG-Boost (XGB). Almost all paper include some kind of cross fold validation before training the models. However, no undersampling problems seems to be present in the data sets of these papers. Furthermore, the models are evaluated on a test set based on several metrics. Most of the measures are based on metrics from the confusion matrix such as accuracy, precision, recall, f-1 score. While some also articles also include metrics as Area under the Curve, and two additional metrics from the confusion matrix, Mathews correlation coefficient and the G-Mean. F-1 score seems to work as a promising metric as it is able to represent multiple metrics into one comprehensible metric [22], as the f-1 score provides a balanced metric on the false positive rate as the false negative rate. However, the evaluation metric usually differs based on the research objective. We also found that DL techniques such as NN are mostly defined as DL techniques, to enhance clarity we treat DL methods as ML methods from this point onwards. Similarly, we regard the MLR as a ML algorithm too.

In the second SLR we aim to answer research question 1.2: "to what extent are EWS used to monitor credit risk?". Therefore, we perform an additional literature review on the application of EWS in credit risk by banking institutions. The key terms used in Scopus located 37 papers and were: (TITLE-ABS-KEY (early AND warning AND system) AND TITLE-ABS-KEY (banking) AND TITLE-ABS-KEY (credit AND risk) OR TITLE-ABS-KEY (early AND warning)) Filters based on articles limited to Language: “English”, Keyword: “Credit Risk”, Include: Only free-access or licensed literature. This resulted in 6 new papers (compared to the prior SLR to be taken into account. Likewise to the previous SLR, detailed answers on this SLR can be found in appendix .2. Unfortunately, little useful insights were found in the second SLR. No similar study considers its EWS in a similar way as at Rabobank. Where we hoped to find studies that had more than 3 classes in the EWS, but only less than 3 were found. Most of the researches emphasize on creating an EWS by solely introducing a single EWS trigger just before default. However, the multi-class classification problem is widely researched, although less than two class problem. Here, the aforementioned ML remain viable but the evaluation metrics require some alteration compased to a two class problem. We further dive into the technical insight on this in chapter 4.3.

Ultimately from both SLRs, the paper that relates most to our work is a case study on monitoring credit risk of ING by University of Twente student Daniel Chen [14]. He defines the EWS as a early warning system that enables the effective monitoring of the credit portfolio by providing Early Warning Indicators and triggers to alert stakeholders - such as risk and account managers - when there are early signs of financial distress. The graduate author tries to effectively classify wholesale clients at ING on a watchlist based on their prospective credit risk. ING uses an application called ARIA, which has similar functionality as Rabobank’s SAMAS, to classify clients based on their prospective credit risk. Opposed to the four classes from SAMAS, Aria includes three stages and thus has

only one watchlist status. At ING, the CRC stages are mostly determined manually based on indication of triggers, whereas at Rabobank CRC transitions are performed in real time and are automatically based on triggers following the decision tree. The predictors that were considered most important following the feature selection method called mutual information – and therefore also included in the models – were predictors that were non-triggers but values from internal data such as exposure at default and risk-weighted assets.

In conclusion, the SLRs in appendix .2 contributed to answer both research questions 1.a and 1.b. From the first SLR we asses that literature on successfully implementing machine learning in monitoring credit risk exist. Combining the SLR with existing ones [8], [39] we form a list of algorithms that are deployed in our empirical study: Multinomial Logistic Regression (MLR), Linear Discriminant Analyses (LDA), Support-Vector-Machine (SVM), Decision Tree (DT), Random Forrest (RF), Neural Networks (NN), XG-Boost (XGB). Methodology on the specific machine learning methods used will be discussed in chapter 4. Furthermore, we answer the use of EWS in credit monitoring in the second SLR, where we found that the existing literature is rather limited on this topic, But the concept of multi-class classification problem is prominent in all cases. This involves theory on general methods for multi-class classification problems and is not necessarily related to credit risk. Also, the SLRs enhanced our view of potential predictors to include. Which opposed to the CRC entry- and exit criteria not only included triggers but also continuous predictors. Specifically the probability of default is frequently denoted as the most significant predictor. In the paper of Chen [14], we find that triggers are not as contributing as other metrics such as exposure at default and risk-weighted assets. Based on this expected feasibility of ML into credit risk systems and its prior insights, we are confident that we are able to contribute to the current credit monitoring system at Rabobank and also enhance both theoretical and practical insights on the topics of ML and EWS.

Chapter 3

Data Review

In this section we delve into the dataset used to train and test the prediction model. The objective of this chapter is to get more familiar with the dataset and to gain intuition behind the predictors and its relation to the outcome variable. This intuition is described both in statistical terms as well as its use in credit risk. To conclude the chapter, we summarize our hypotheses regarding the significance of the predictors and their impact on the model's separability.

In section 1.7 we stated that we focus on RSME clients from the business unit LBBB that are currently on the watchlist in CRC [EW] and want to predict their CRC 3 months into the future. Of course, to train such a model, we need to specify the specific months that are used. We choose to predict the CRC most recently reported at the time of research, which is Aug-23 and is called the outcome variable. Consequently, 3 months prior to the outcome variable, is data reported at May-23 and the variables used to do so are called predictors. Together they form to be the so-called merged data frame. To increase comprehensibility we consider the predictors to come from three pipelines. An overview of these pipelines (including corresponding reporting months) is given in figure 3.1.

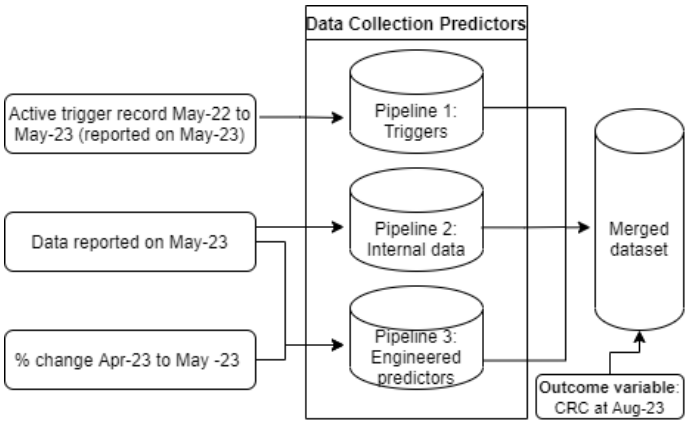

FIGURE 3.1: This figure shows the three pipelines that constitute the predictors, grouped by the client ID, the data is merged into one dataset.

This means that the latest known data before the prediction is at may-23, the data reported and used at this moment in time is called the snapshot. Based on this snapshot either one of the four transitions as denoted by figure 3.2 occur 3 months into the future. Initially, the three pipelines that are considered contain all the data made available by Rabobank through the SAMAS application. A straightforward method is to let data mining do the work and automatically dilute the significant predictors. However, this

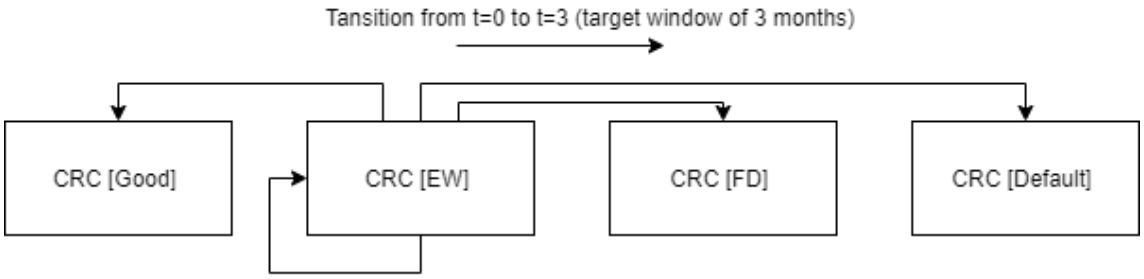

FIGURE 3.2: This figure shows the possible transitions from all clients that are in CRC [EW] at May-23 to the four possible CRC [Classes] at Aug-23

does not allow us to gain a comprehensive understanding of the initial dataset before any predictors are removed. Therefore, throughout this chapter, we conduct a thorough review of each of these predictors, examining their relevance and significance in the context of credit risk.

3.1 Data Constitution

The data from the three pipelines in figure 3.1 is extracted through Rabobank's Cuberouter and is accessed through a query. The query is able to extract all indicative dimensions and convert this into a readable data frame. Rabobanks level to determine contracts individually is on a so-called regulatory facility level. However, to enhance interpretibility we define the lowest level to be on "Client_ID" level. We merge the data frames from the pipelines and ensure that no clients are lost when matched by Client_ID and thus all clients at least indicate a CRC [Class] at Aug-23. The total number of clients - called the sample - in the particular dataset used is not presented in this thesis due to confidentiality reasons, furthermore a client will also be referred to as a "row". The exact methods used to filter and merge the merged dataset can be found in Appendix .3. After merging, the client_ID is removed, in this way information is protected on client level. To further enhance the clients privacy, no data on individual client level is published in this thesis.

To ensure consistent use of terminology, a feature, column or independent variable used as an input for the prediction model will be called a predictor. We consider three types of predictors according to the three pipelines in figure 3.1: triggers, internal data and feature engineered predictors. The actual variable indicating the class to predict is called the outcome variable - and has a equal meaning to target variable or dependent variable. Elaboration on the included predictors and their relation to the outcome variable is given throughout the rest of the thesis. All the modelling is conducted in R.

3.2 The Predictors

3.2.1 Triggers

Definitions

The inputted predictors can be seen as a vector X for each client. Across all predictors inside the vector X a pattern is hoped to be conceived that corresponds to a certain class, this classification is denoted the outcome variable (Y). In this section we delve into each of these predictors. We start with the first pipeline: Triggers. We start with an overview of the trigger types including a description in table 3.1.

Symbol	Description	Type
T009	DPD $>$ 30 days. Indicates 30 days past due, up to 60 days, for 100 EUR exposure. No forbearance needed. Automatically registered. If triggered twice in 3 months, QA needed.	Integer
T025	Direct external indication of unlikeliness to pay. Manually registered if notified by other financial institutions of client's financial distress.	Integer
T029	Other signs of unlikeliness to pay of the obligor. Manual registration based on indirect signs of financial distress (e.g., payment refusals, untraceable client).	Integer
T030	Cross default trigger. Activates if a related facility or client is in default. Definite CRC determined through QA. Automatically registered.	Integer
T043	Director/owner passes away trigger. Manually registered if client's director, owner, or partner deceases.	Integer
T044	Portfolio trigger. Automatic trigger if client falls under risk category following from sector or portfolio analyses. Automatically transfer to CRC [EW]	Integer
T070	QA/CF analysis concluding non-CRC [FD], but better. Manually registered, can only trigger if client is in CRC [FD]. Based on QA/CF, client transitions to CRC [EW] or CRC [G].	Integer
T100	DPD $>$ 1 day of at least 1 EUR. Early indication of potential financial distress. No CRC transition. Automatically registered.	Integer
T104	DPD $>$ 90 days. Considered a technical default, not an official CRC default status but all relevant info must be logged and available. Automatically registered.	Integer
T117	Emergency funding trigger. Manually added for emergency funding or contractual changes on short notice. QA needed for further CRC assessment.	Integer
T119	Trigger for client contact with Special Asset Management. Client unable to fulfill obligations without bank help. Manually added.	Integer
T130	Rabo Bank Risk Rating (RRR) = R18 or R19 trigger. Detects if client is in RRR18 or RRR19, indicating a high degree of financial distress by RRR model. Automatically registered.	Integer
T131	Rabo Bank Risk Rating (RRR) $<$ RRR18 trigger. Detects if client goes from R18-20 to RRR $<$ 18, indicating decreasing financial distress. Automatically registered.	Integer

TABLE 3.1: Description of Triggers

Triggers act as binary flag variables that activate when certain conditions - i.e. threshold breach in a underlying metric - are met. Only the triggers that are active at the current reporting date May-23 are considered in our model, moreover the same trigger can be present more than once. However, the latter is barely the case. It is worth noting that in SAMAS much more triggers exist, but since we focus solely on clients in CRC [EW] these triggers are are just not reported within this scope. Furthermore, triggers occur automatically or are manually added by portfolio holders at the Rabobank W&R department. Some of the automatic triggers give rise to a qualitative assessment (QA) or cashflow analysis(CA), which can result in a CRC transition to occur. This makes the prediction model especially useful since it can try to perceive a pattern in the outcome of the QA and CA and take away some uncertainty there.

Data Properties

In this section we describe the Data Properties of the Triggers. We start with a summary of descriptive statistics.

Predictor	mean	sd	median	min	max	range	skew	kurtosis	se
T009	0.0035	0.0587	0	0	1	1	16.92	284.40	0.0008
T025	0.0760	0.3210	0	0	5	5	5.24	35.71	0.0042
T029	0.0136	0.1204	0	0	2	2	9.26	92.17	0.0016
T030	0.0124	0.1124	0	0	2	2	9.16	86.01	0.0015
T043	0.0339	0.1973	0	0	3	3	6.45	47.55	0.0026
T044	0.0750	0.2724	0	0	2	2	3.58	12.37	0.0036
T070	0.0119	0.1101	0	0	2	2	9.38	90.41	0.0014
T100	0.0603	0.2387	0	0	2	2	3.73	12.14	0.0031
T104	0.0003	0.0186	0	0	1	1	53.76	2889.00	0.0002
T117	0.0098	0.1087	0	0	2	2	12.22	166.76	0.0014
T119	0.0003	0.0186	0	0	1	1	53.76	2889.00	0.0002
T130	0.7972	0.4102	1	0	3	3	-1.30	0.44	0.0054
T131	0.1064	0.3869	0	0	12	12	9.37	201.58	0.0051

TABLE 3.2: Statistical summary of triggers

The findings from this table are listed below.

- The **range** indicates the spread of data. Some predictors like "T009" and "T104" have a narrow range (0 to 1), while others like "T130" and "T131" have a wider range (0 to 3 or even 12). A histogram that specifies the spread per count can be found in appendix .4.
- The **mean** and **median** statistics provide insight into the central tendency of the data. For instance, "T130" has a mean of approximately 0.7972, indicating that this predictor is flagged for most clients.
- The **skew** statistic indicates the skewness of the data distribution. Positive skewness (values greater than 0) suggests a right-skewed distribution, while negative skewness (values less than 0) suggests a left-skewed distribution. For example, "T130" has a negative skew of approximately -1.30, indicating a left-skewed distribution. The **kurtosis** column measures the degree of peakedness or flatness in the data distribution. Higher kurtosis values indicate more peaked distributions, while lower values suggest flatter distributions. "T104" and "T131" stand out with an extremely high kurtosis of approximately 2889.00, indicating a highly peaked distribution. This is likely because the of the trigger is very rare.

Furthermore, we stated that triggers are either active or inactive and if they are active, they can also be reported more than once. This statement is reflected in table 3.5, a range of 0-1 indicates a binary variable, while the vast majority contains a wider range. If the trigger is of binary value, the variance, skewness and kurtosis may not be as informative or interesting as it is for continuous predictors and wider intervals. This is because binary predictors take on only two possible values (0 and 1), which limits the variation within the predictor. As a result, the variance of a binary predictor tends to be relatively small. This also holds true in case that triggers with a wider range most of the time only report one or zero triggers. As such, more than one counts should be seen as outliers. To check this, we provided an overview of the total number of triggers reported, and include the proportion that they are reported at least one from this total.

Trigger code	Total percentage trigger is present	Percentage trig >1 over total
T009	0.35%	0.00%
T025	6.25%	17.96%
T029	1.31%	3.95%
T030	1.23%	1.41%
T043	3.09%	8.94%
T044	7.25%	3.33%
T070	1.17%	1.47%
T100	6.01%	0.29%
T104	0.03%	0.00%
T119	0.03%	0.00%
T117	0.88%	11.76%
T130	79.43%	0.33%
T131	9.62%	4.85%

TABLE 3.3: Table overview of frequency that triggers are present. The 2nd column indicates the percentage that the trigger on client level is present at least once. Where the 3rd column indicates the proportion the trigger is present >1 in from the percentage_tot.

We see that in the 2nd and 3rd column most triggers - apart from T130 - are quite unique per client, most triggers are $< 10\%$ present per client. Looking purely at three of the most frequent triggers: T130, T131 and T044. T130 is a trigger that follow from a high RRR and indicates decent sign of financial distress, hence the high percentage is intuitive. Additionally, trigger T131 indicates decreasing financial distress and is probably associated from clients that were previously CRC [FD] or worse, it is worthwhile to investigate in the exploratory analyses what proportion of these clients continue on further decreasing financial distress by checking if they are also are in CRC [G] in Aug-23. T044 is a Portfolio trigger, and is a automatic trigger if client falls under risk category following from sector or portfolio analyses. Its relatively high proportion might contain overly precociously watchlisted clients, that in fact are not as financially distressed as they appear in the CRC. Furthermore in the last column, from the total number of triggers, only a small percentage is present more than once per client. Intuitively, it remains questionable to regard the trigger as a numerical variable. However, since machine learning is able to cope with both types of variable types, we include the triggers both as a numerical variable - indicating the count - and as a binary variable - indicating its individual presence. These binary variable will be stated throughout the rest paper i.e. as T131_Once for the triggers applicable.

3.2.2 Internal Data

Definitions

The second pipeline contains internal data available through SAMAS, but of which the stand-alone metrics are not directly included in the CRC transition criteria. Because of this reason, valuable information might be contained in this data on CRC Transitions which are not directly associated as such. Likewise to the previous section an overview describing these predictors can be found in table 3.2.2. It is worth to mention that the

numerical variables from this pipeline are common credit risk metrics (i.e. EAD). Also a part of these metrics are explicitly mentioned in multiple Basel accords i.e. Basel IV denoted that the regulatory capital RC is a mandatory capital that should be kept in by the bank in cash on the balance sheet for the specific loan and is to be at least 8% of the RWA [35].

Symbol	Description	Type
RRR	Rabobanks Risk Rating framework. A Framework based on Basel and EBA regulation and is used for assessing the probability of Default (0-100%) for Obligors/Facilities to indicate the financial distress. It can be interpreted as substages of the possible CRC statuses. Stages RRR:R1-R20 can be applied to CRC [G] CRC[EW] CRC[FD]. RRRD1-D4 can be given to CRC[D].	Categorical
CRC_prev	Previous CRC that the client had before the client became CRC[EW]. Useful to include past information of client.	Categorical
New_EW	Indicating if the client is new in EW this month	Binary
New_EW_3	Indicating if the client was new in EW 3 months ago	Binary
New_EW_6	Indicating if the client was new in EW 6 months ago	Binary
New_EW_12	Indicating if the client was new in EW 12 months ago	Binary
Sector	Indicating the clients sector. It is indicated the least detailed category as then options are most limited.	Categorical
O_Exp	"Original Exposure" of the client. Which is to actually be considered the current exposure that the client has without the expected 3 interest fees.	Numerical
EAD	Exposure at Default (EAD) indicates the predicted amount a bank may be exposed to when a debtor defaults on a loan. For retail loans, the amount is calculated based on the IRB approach. EAD = O_exp * . Where indicates the on- & off balance outstanding exposures, which includes lost interest.	Numerical
PD	Probability of default of the client on a one year time horizon. This is not the same as the probability that the client is going to CRC [D]. PD is calculated based on approved internal model calculations and the outcome widely varies based on the clients attributes.	Numerical
RWA	The Risk-Weighted-Asset value of the client. It indicates the minimum capital that Rabobank should keep as a reserve to reduce the risk of insolvency for this specific client. It is calculated using Standardized approach (obligatory since Basel IV to be implemented in 2025). Calculated using RW% $\bar{E}$AD. RW% is determined by client attributes such as LGD, PD and the client's risk profile.	Numerical
RC	Minimum regulatory capital requirement to hold as bank on the balance sheet according to regulators for the specific client. Since Basel IV calculated to be at least 8% of the RWA. Including both RWA and RC is expected not to bring much value to the model as the formula provides a linearly correlation	Numerical

Symbol	Description	Type
LGD	Loss given default (LGD) is the amount of money a financial institution loses when a borrower defaults on a loan, after taking into consideration any recovery, represented as a percentage of total exposure at the time of loss. LGD = EAD*(1-Recovery Rate)/ EAD	Numerical

Data properties

In table 3.5 we provide descriptive statistics of the internal data. This includes the ordered categorical and binary data, which is rewritten to numerical data types and excludes the unordered categorical variable sector. Solely looking at table 3.5. We see that **New_EW_12** contains NA values, this is because it only has zero values and therefore some statistics can not be calculated. This indicates that the predictor does not contain any useful predictive power and can thus be removed. Furthermore, the descriptive statistics of RWA and RC are identical. This can be explained as **RC** is at least 8% of the **RWA**. The statistics show that Rabobank does not deviate from this minimum and therefore the underlying metric RC is not included as a final predictor. Another interesting insight is that **PD** predictor has a mean value of 0.074 and a low standard deviation of 0.067, with a minimum value of 0. This suggests that **PD** is primarily concentrated around low values.

Predictor	mean	sd	median	min	max	range	skew	kurtosis	se
RRR	5.466	0.851	6	2	7	5	-1.726	2.665	0.011
CRC_prev	1.623	0.951	1	1	4	3	0.920	-1.003	0.012
New_EW_6	0.048	0.213	0	0	1	1	4.245	16.022	0.003
New_EW	0.111	0.314	0	0	1	1	2.475	4.127	0.004
New_EW_12	0	0	0	0	0	0	NA	NA	0
New_EW_3	0.160	0.366	0	0	1	1	1.859	1.455	0.005
PD	0.074	0.067	0.062	0	1	1	3.771	34.871	0.001
EAD	0.121	0.146	0.059	0	1	1	1.636	2.121	0.002
O_Exp	0.118	0.142	0.062	0	1	1	1.615	2.032	0.002
RC	0.031	0.067	0.008	0	1	1	5.110	36.819	0.001
RWA	0.031	0.067	0.008	0	1	1	5.110	36.819	0.001
LGD	0.127	0.089	0.123	0	1	1	1.636	8.961	0.001

TABLE 3.5: Summary statistics of predictors coming from the second pipeline internal data. The numerical values are scaled from 0-1 to enhance confidentiality of clients data.

Additionally, histograms and bar-charts are provided in appendix .4 to check how the data is distributed. Three plots of interest are taken from the appendix and shown in figure 3.3. These are useful to visualize the categorical variables. We are of specific interest in the unordered categorical predictor: sector. We see that the sectors distribution is not distributed uniformly. It is shown that wholesale and retail trade make up the biggest part of the client sector, whereas there are many sectors that are not as much present. It is worthwhile to investigate transitions against the fraction at which that sector is present as it helps us to hypothesise about their separability per class. Furthermore, we see in the same appendix that higher RRR stages are more frequently present in the dataset. Which is expected as the RRR>18 trigger directly sends clients to CRC[EW]. The CRC_Prev histogram is shows that the previous CRC count decreases for more severe previous classes,

thus most clients in EW came from CRC[Good]. To further delve into the previous CRC we will also explore the relation between CRC_prev to the CRC in Aug-23 in section 3.4.2.

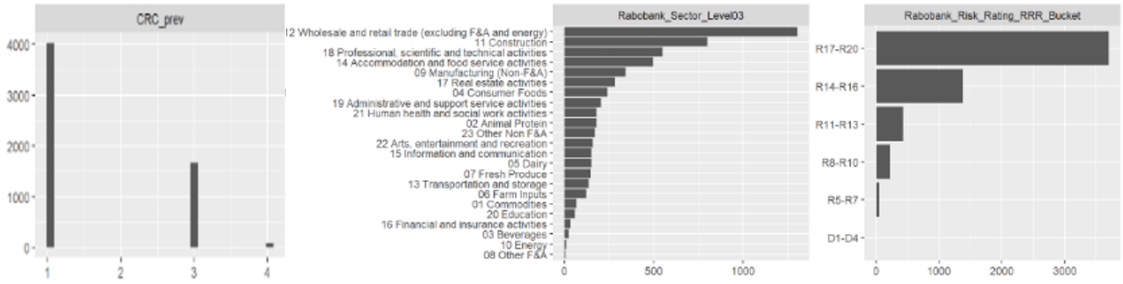

FIGURE 3.3: This figure shows a barchart of CRC_Prev with the y axis indicating counts and the x-axis the CRC from [G] = 1 to [D] = 4. The histograms show the x-axis as counts and the y-axis indicates the category.

3.2.3 Feature Engineered Data

Definitions

Variable	Description	Type
CRC_n_days	Represents the number of days that the client is in CRC [EW]. Calculated based on the range of days between two removed variables: CRC_date_changing_trigger and reporting_date_[month_yr].	Numerical
d_EAD, d_PD, d_O_Exp, d_RC, d_LGD	Represents the percentual change for the months from apr_23 to may_23 and form a new engineered predictor to our model	Numerical

TABLE 3.6: Feature Engineered Variables

The third pipeline are feature engineered predictors. Feature engineering is done by manipulating data from the raw dataset to constitute a new predictor that aims to increase the model performance. The feature engineered predictors calculate the percentual change of the numerical predictors at may_23 against apr_23 and will be refered to use as the delta predictors. Also, the number of days that client is in CRC[EW] is calculated to give insight on how long the client is already labeled as CRC[EW] since its transition to CRC[EW]. An overview is provided in table 3.6.

Data properties

Histograms on the distribution of the delta predictors can be found in appendix .4. Purely looking at table 3.7 we see that the range can take huge positive values compared to its negative values, this is because in theory a percentual increase can be infinitely high while a decrease can never be lower than 100%. Because of this outliers are expected to make the distribution right skewed. Intuitively, outliers are expected to posses the most predictive power as they indicate abrupt changes on the short term. In the next section we discuss the outcome variable that specifies the four CRC classes at Aug-23.

Predictor	mean	sd	median	min	max	range	skew	kurtosis	se
d_PD	0.220	1.132	0.045	-0.979	18.923	19.902	7.872	91.813	0.014
d_EAD	0.022	0.541	-0.005	-0.999	14.883	15.883	16.788	379.475	0.007
d_O_Exp	-0.001	0.299	-0.004	-0.980	14.354	15.334	40.023	1857.242	0.004
d_RWA	0.452	11.202	0.003	-0.999	649.319	650.319	42.555	2159.864	0.147
d_LGD	0.096	1.694	0.000	-0.978	64.853	65.831	24.438	707.364	0.022
CRC_n_days	108.652	89.661	96	0	364	364	1.048	0.262	1.178

TABLE 3.7: Summary statistics of predictors from the third pipeline: feature engineered predictors. The percentual change metrics indicate a time lag of 1, calculating a delta between the months Apr-23 and May-23. While CRC_n_days indicate how long the client has been in CRC[EW] up until the reporting date of the snapshot at May-23.

3.3 Outcome Variable

3.3.1 Definitions

In machine learning the dependent variable is referred to as the Outcome Variable (Y), it is sometimes also regarded as the target variable or dependent variable. In our model we refer to our outcome variable as CRC_Y_Outcome.

CRC_Y_Outcome is a categorical variable that specifies the CRC Status at August 23 based on four classes, this means that we have multi class classification problem at hand. We addressed that transitions occur instantaneously in real time without any uncertainty. Therefore we are less interested to why transitions occur, we specifically care about if and when they occur, such that the transitions is measured.

Based on the constituted merged dataset from section 3.1, the following transitions from CRC [EW] occurred between May-23 up until Aug-23.

at_aug_23	**CRC [G]**	**CRC [EW]**	**CRC [FD]**	**CRC [D]**
% of total	26.682%	60.069%	12.538%	0.711%

TABLE 3.8: Outcome variable: the CRC transtitions from May-23 (all clients are in CRC [EW]) to Aug-23 as a proportion of the total set

We see that most clients (60.069%) stay in CRC [EW] after the target window of 3 months and only 0.711% transition to CRC[D], naturally a low percentage of CRC [D] is considered a good thing. Though for our prediction model, this means that this specific transition might be under sampled in the dataset. This invokes an imbalanced (skewed) class distribution. AFexn imbalanced class distribution will have one or more classes with only a few examples - the minority classes - and one or more classes with many examples - the majority classes. This might imbalance training pattern recognition and makes a test set much harder to accurately predict upon. We should therefore consider techniques that can cope with imbalance during the training model and stratify a test set to ensure the minority classes are represented, we further dive into this in section 4.1.

In the next section we conduct an exploratory analyses on the relation between predictors and the classes of the outcome variable.

3.4 Exploratory Analyses

In the previous section we already mentioned some predictors that required additional analyses. In the exploratory analyses we conduct this analyses and seek a relation between the predictors and true classes. The objective of the section is to gain further intuition behind the predictors and to hypothesise about the separability of classes from these predictors. The predictors include the start date in EW: CRC_n_days and the prev_CRC. Furthermore, we delve into the sector and the trigger frequency per class. Additionally, we compute a Spearman correlation matrix to see if relations between predictors and the outcome variable exist. The aim of the exploratory analyses is to hypothesise about the predictive power that the predictors might have, while it also contributes to becoming more familiar with the data set as a whole.

3.4.1 Spearman Correlation Matrix

A Spearman correlation matrix is computed to examine potential relationships between the predictors and the outcome variable. The corresponding correlation plot can be found in Fig 3.4. The Spearman correlation coefficient (ρ) measures the strength and direction of monotonic relationships between pairs of variables by assessing the covariance of the ranks of the variable. It is not as effective on non ordinal categorical data. But it certainly fits better on these predictors than the more commonly used Pearson correlation matrix, which is limited to continuous numerical data types [25],.

In total, this analysis involved 33 predictors, resulting in 1089 pairwise correlations. With so many relations to consider, interpreting the correlation plot can become overwhelming. To address this, we chose to focus our attention on Spearman correlation coefficients exceeding 0.6 and -0.6, indicating the substantial monotonic relationships.

It's important to note that a high correlation between variables in Spearman's rank correlation coefficient does not necessarily imply causation, just like in the case of Pearson correlation. Spearman's correlation measures the strength of a monotonic association between two variables based on their ranks rather than their actual values. This method is less sensitive to outliers, making it more robust in the presence of extreme values [23].

However, it should be noted that Spearman's correlation - like Pearson - cannot establish a definite causation. Nevertheless, we can hypothesize that these predictors are essential features contributing to the separability of the model, based on the findings in the Spearman correlation plot.

1. 1. $\rho = 1$. Relation between RC and RWA they have exact linearity as following Basel [32] banking institutions are required to hold at least 8% capital for each loan, hence RC = 8% * RWA. Of course, the same holds for the deltas of RWA and RC.

2. $\rho = 0.99$. Relation between EAD and O_exp, which mostly are of equal value. In some cases, EAD indicates a higher value when loans include expected interest that are lost at default.

3. $\rho = 0.861$ - 0.802. All combinations EAD, RC, RWA, O_exp. EAD is an input variable for all formulas and explains why the r coefficient of the combination of these variables(the blue block in the upper right). Furthermore because of that d_EAD correlates with the corresponding delta's of RC, RWA and O_exp too.

4. $\rho = 0.703$. Relation between RRR and PD relative high correlation. The RRR stages are mostly based on ranges of PD value and a relation is intuitive (also explain a low positive with T130).

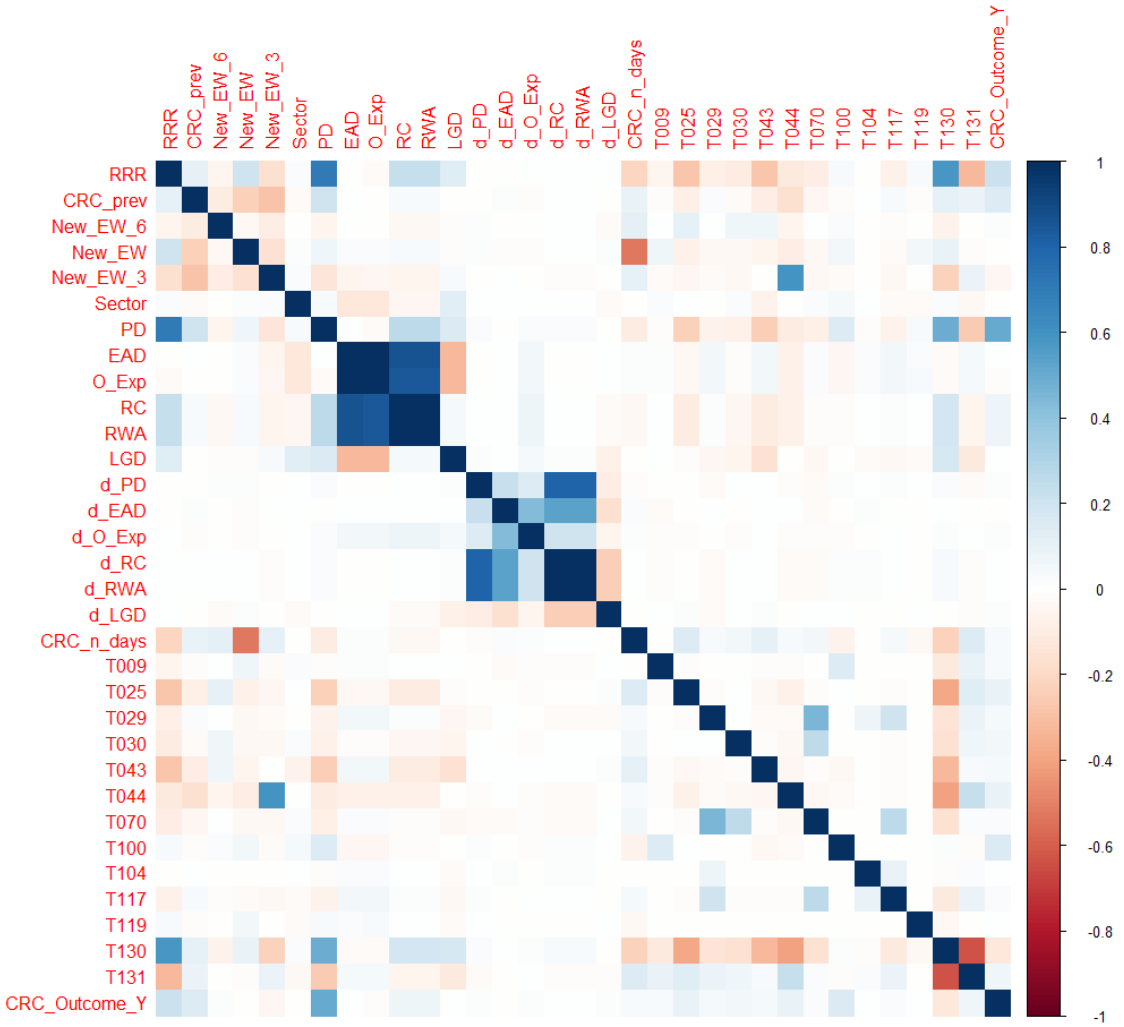

FIGURE 3.4: This figure shows the Spearman Correlation of all viable predictors

5. ρ = 0.653-0.612. Relation between T130 and resp. PD and RRR. One of the few trigger that shows relative high positive correlation with predictors. T130 indicates that the RRR=18 or 19 and thus a relation to RRR is expected.

6. ρ = 0.622. Relation between T044 and New_EW_3. T044 is a portfolio trigger that sends clients to CRC [EW] and is quite apparent for clients whom specifically arrived exactly 3 months ago.

7. ρ = -0.638. Relation between T130 and T131. T130 triggers with an increase in financial distress whereas T131 triggers with decreasing financial distress. So this negative coefficient is intuitive.

8. The Outcome Variable does not indicate a strong coefficient with one of the predictors. However PD is considered the most significant (Positive Correlation) predictor. And therefore also expected to contribute on the separability for the predictions.

9. A further interesting indication (wide white cross in the matrix) is that the delta predictors do not seem to show any consistent relation with any of the predictors (except for d_EAD as explained in 3.). We therefore question their contribution the the separability of classes and the effect of small scale time series to the predictions.

3.4.2 Previous CRC and days in [EW]

In this section we dive into the relation between the previous CRC and the number of days in EW against the outcome variable. This analysis first delves into the entry dates

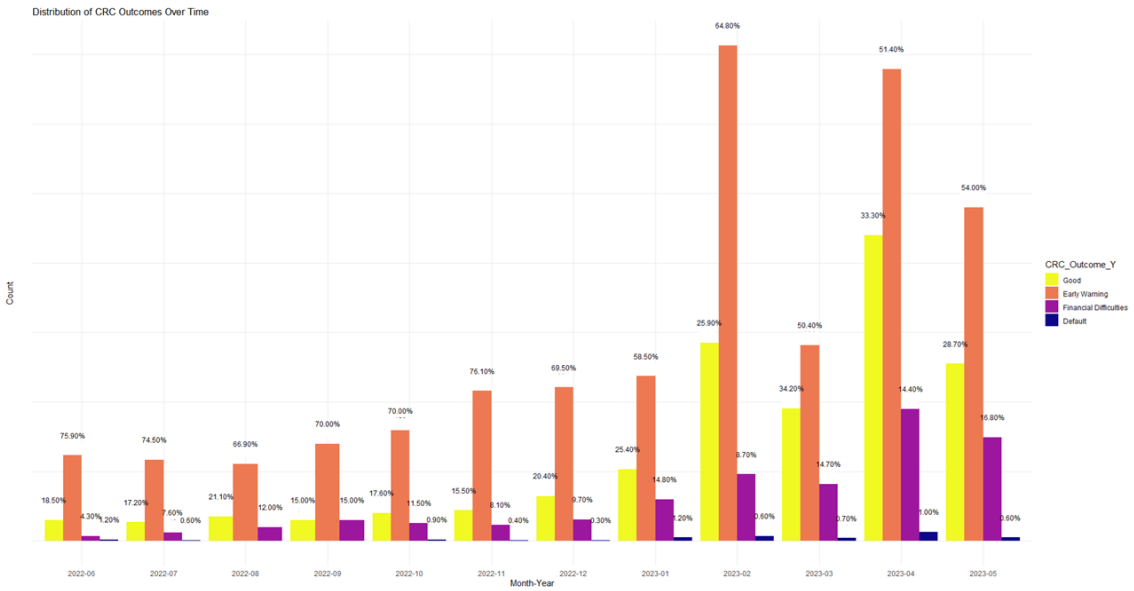

FIGURE 3.5: This figure shows histogram plots for the proportion of the outcome variable against the entries per month at which the client transitioned to CRC [EW]

of clients (month based) in the CRC [EW] class and plots them against the respective CRC_Outcome_Y observed in August 2023. The plot can be found in figure 3.5 It shows both the total occurrences per CRC_Outcome_Y each month, as well as the class distribution for that month (as a proportion of the total transition). Notably, the majority of entries in CRC [EW] appear to be relatively recent. This observation aligns logically with the primary objective of CRC [EW], as its serves as a watchlist status and is thus meant as a temporary state for clients within the system. Furthermore, the proportion of non CRC [EW] transition are more dominant in recent months. Indicating that the clients that entered recently in CRC [EW] are also transitioning. We do notice a kink in observations for March, however the class distribution remains is in proportion to neighboring months February and April, so we still expect that the models might find a separability on lower CRC_n_days.

Additionally, we plot the CRC_prev against the class CRC_outcome_Y in Aug-23 in Figure 3.6. In the violin plot the thickness presents the count distribution of the combinations of values for CRC_prev and class CRC_outcome_Y . In the violin plot we preferably see an ascending or descending order of thickness. As we care about the class distribution, we primarily look at the plot on column level. We see that clients in CRC[G] at Aug-23 are likely previously from CRC [G] and thus return back to their class. Whereas the remainders of CRC [EW] also came mostly from CRC [G]. One could argue that a big part of clients are overly cautious put on the watch list. For the transitions to CRC [FD], no conclusion can be made as both the previous CRC for [FD] and [G] shows equal thickness. CRC [D] also shows no clear indication for a relation between previous [CRC] thicknesses are widely spread. Briefly looking at the plot on row level, the previous CRC also shows no consistent relation, indicating no clear relation with CRC_prev and the CRC[D] class.

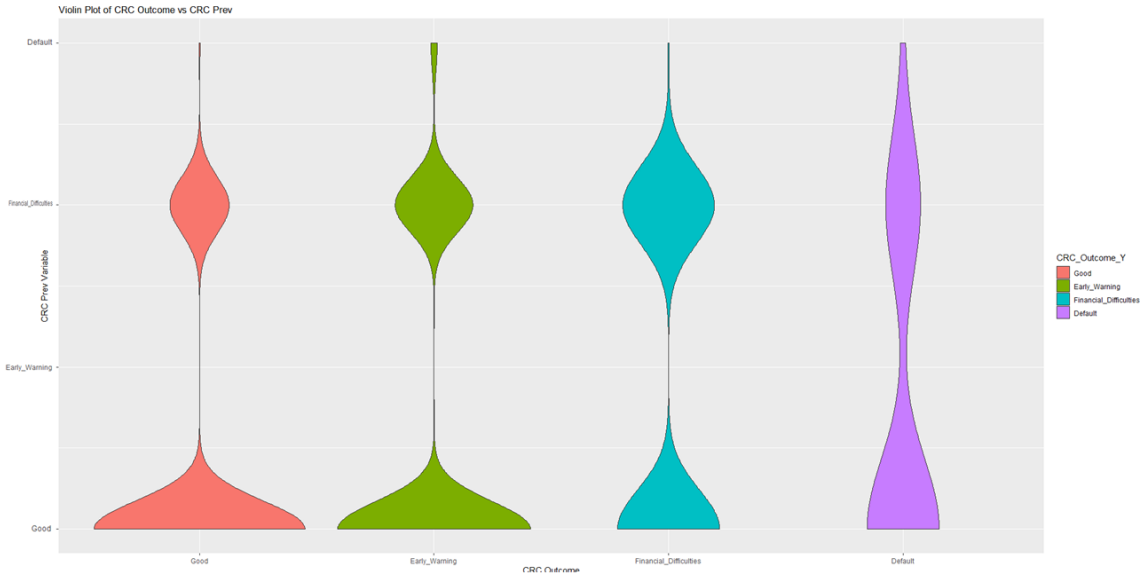

FIGURE 3.6: This figure shows a violin plot of the previous CRC [EW] against the CRC_Outcome_Y. All clients are currently in CRC [EW], so their CRC_Prev cannot be CRC [EW], hence we see no density there. We see that most clients who previously came from CRC [G] are most likely to be in CRC[G] again in August. Whereas, when we look at clients previously in CRC [FD], are as likely to go back to CRC [FD] as to go to CRC [G] (the width is about equal).

3.4.3 Sector

In figure 3.7 the proportion of the total occurrences of the specific sector are plotted against the CRC in Aug-23. This relation was specifically requested for investigation by Rabobank. Also, for our prediction model it is an interesting predictor, as it as unordered categorical predictor and therefore has no intuitive generalization. This is because categorical data that is unordered consists of categories with no inherent numerical order or meaningful distance between categories. For unordered categorical data, you cannot calculate correlation coefficients because there is no natural way to assign ranks or determine a linear or monotonic relationship between the categories. In the plot a dashed lined representing the class proportion from table 3.8, the sector all show to be around this dashed line line with only some individual outliers per class(from a batch of 23 options). This indicates that their is likely no consistent separability of the class based on the sector. CRC [D] indicates an outlier for "Other F&A" (2.929%), but this only account for 3 samples and likely does not have significant influence on the predictors separability. Concluding, we expect that the predictor "sector" will not provide significant importance on the separability of classes.

3.4.4 Trigger distribution

In this section we want to investigate the relation between the trigger types and the outcome variable. To visualize this, we plot the trigger frequency in a heat map in figure 3.8. This heatmap shows how sensitive a trigger is to the CRC Outcome variable i.e. the block in the upper-right shows that from all clients in CRC[G] at Aug-23 97.48% have a active T130 trigger reported. The dark-red blocks are expected not to be of significant importance on the separability of the class, hence do not contain predictive power. The most significant

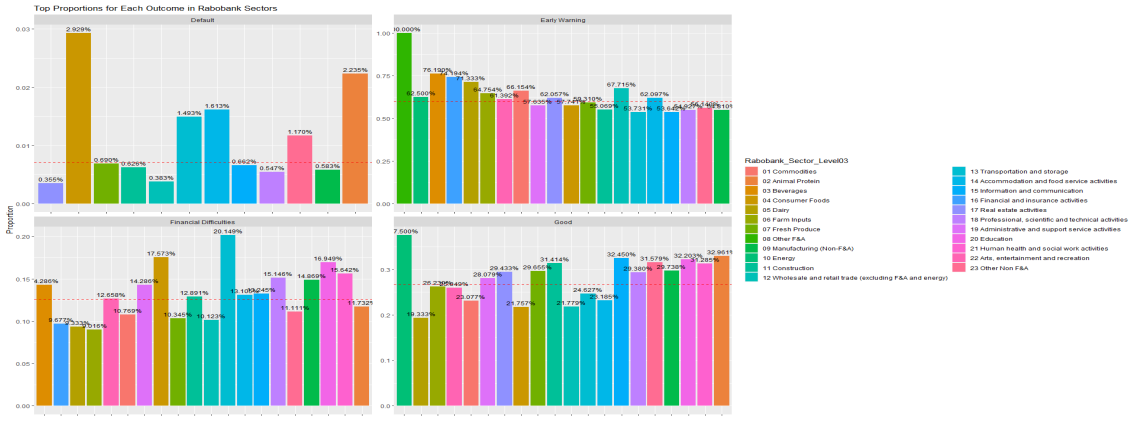

FIGURE 3.7: This figure shows the proportion of total occurrences per sector against the CRC [class] in Aug-23. We see that there is no distinct sector overly present in one of the classes.

cells following from the plot are as follows.

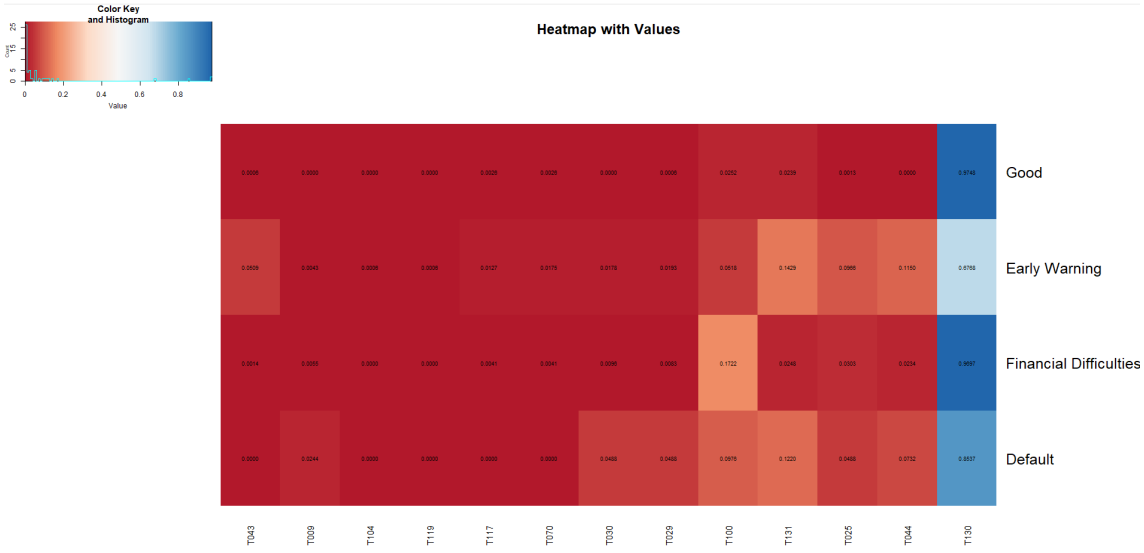

FIGURE 3.8: This figure visualizes the trigger frequency distribution

- T131 is relatively high for both CRC [EW] and CRC [D] and might contain a significant contribution to the separability of classes. Intuitively, as T131 detects decrease in financial distress, one would expect them to be in CRC[G]
- T130 is frequently present for all CRC_outcome_Y. However CRC [EW] is drastically lower than the other CRC_outcome_Y classes. Hence, in case that the T-130 is not reported, a likeleniss that the customer remains in CRC[EW] at Aug-23 can be interpreted.
- T100 is distinctively high for CRC [FD] and indicates early financial distress (1 DPD).
- T44 is a portfolio trigger and awaits qualitative assessment the outcome is either positive (resulting in stay in CRC [EW] or negative and go to CRC[D] - just like the

other client in portfolio. It is uncertain if separability by models is found as both have quite an equal value.

- Note that significant influence on CRC [D] by triggers is hard to determine for the minority class is undersampled (the most sensitive trigger after T-130) is only for 5/41= 12.20 % of the samples.

3.5 Key findings & Hypotheses

The data review helped us to gain knowledge and insight on the predictors, outcome variable and their relation. It also helped to accumulate an answer on research questions 2b and 2c. The key findings following these research questions are as follows. The findings are formulated as hypotheses on the predictors power to separate the CRC_Y_Outcome class. This is useful as we can reflect if these predictors indeed contribute to the predictive power of the model, which confirms a solid intuition behind the dataset.

- New_EW_12 shows 0 variance (all clients indicate 0) and is thus removed as a predictor.
- RC and d_RC are perfectly linearly correlating with resp. RWA and d_RWA and do not uniquely contribute on the separability of classes. RC and d_RC is thus removed to reduce computation time.
- PD is expected to be of signifcant importance as it shows a relative high pearson correlation to the outcome variable
- Since PD is positive correlated to the outcome variable. Its own relation EAD, O_Exp, RWA are also expected to be of statistically significant and posses predictive power. We chose to remain both the EAD and O_Exp in the model - instead of removing one - as they are not perfectly correlated (although very high).
- CRC_n_days is expected to provide some predictive power as most transition occur for the more recent entries.
- CRC_prev is likely to posses predictive power for determining the CRC [G] and [EW] are likely from clients that came previously from CRC[G].
- Sector is expected not to provide any predictive power, as all sector are somewhat evenly spread around the expected uniform distribution.
- T130 and T100 are expected to be the triggers - if any - to be most significant and posses some predictive power.

Chapter 4

Methodologies

In this chapter we provide technical insight on the methodologies used to develop and interpret the prediction model. The methodologies addressed in this chapter follow from the methods found in the literature chapter 2. Furthermore, we found in chapter 2 that the outcome variable showed an imbalanced class distribution. We therefore start this chapter by explaining the methods to cope with imbalanced datasets.

4.1 Imbalanced multi-class classification problem

In the vast field of Machine Learning, the general focus is to predict an outcome using the available data. The prediction task is a called a "classification problem" when the outcome represents different classes, hence discrete outcomes. A "regression problem" problem on the other hand, would address the prediction of a continuous numeric measurement [37]. We focus on four classes and therefore our problem is called a "multi-class classification problem".

Multi-class classification can invoke class imbalance, which is considered to be a crucial problem in machine learning. In an imbalanced dataset with respect to classes, the number of one or more classes present are much greater - majority classes- than the other – minority classes [28]. In our case, the minority classes are CRC [FD] and CRC [D], where specifically CRC [D] contributes to only a small proportion of the total sample. Models trained on imbalanced dataset as such tend to favor the majority class, leading to biased predictions [10]. This leads to predictions performing well on the majority class but poorly on the minority class, even though the minority class, in our case CRC [D] is considered as equally important to predict. This imposes a lot of issues to interpret the "accurateness" of the prediction model, as the accuracy becomes quite a misleading metric for multi-class imbalanced datasets. In example, a model could already achieve a relatively high accuracy ($> 60\%$) by naively predicting the majority class CRC [EW] for all cases, even if it fails to identify any of the minority class instances correctly. This problem is specifically addressed in multi-class classification, as certain classes become more unique the more classes are possible [1]. Therefore, we should be aware that interpreting the accurateness of a prediction model should not be mismatched against the accuracy metric, as they are not the same. How accurateness is to be interpreted is presented from three perspectives below.

- The first perspective considers predicting each class equally important, hence a balanced accuracy has to be measured.

- The second perspective considers the pure accurateness from the model by weighting the classification against the proportion that it is present in the data. Here a weighted accuracy constitutes. It aims not to punish false predictions on the minority class as much.
- If the imbalanced class is not able to capture an accurate prediction model, we train the models as a 3-class problem by rewriting the biggest minority class CRC [D] to CRC [FD].

Each of these perspectives gives another meaning to the accurateness and provides more interpretation possibilities on the effectiveness of the prediction model. Further elaboration on the evaluation metrics from each of these perspectives is given in chapter. What is important for now is that the class distribution imbalance can lead to a model's bias towards the majority class during training, making it less sensitive to the minority classes and potentially resulting in under fitting for those classes. To improve the representation of classes, multiple under- and oversampling strategies are considered. The methods considered originate from the paper [2] by Agrawal and is cited throughout the rest of this section. Under- and oversampling are methods to manipulate the dataset such that the dataset becomes less imbalanced. Oversampling involves techniques that adds artificial samples of the minority class instances. Its objective is to improve its representation in the dataset, while also preserving the true information within the original rows. However, the generation of synthetic samples could in their place introduce over fitting, especially if the synthetic samples do not represent the true underlying distribution accurately. Undersampling on the other hand, makes that dataset less imbalanced by applying discarding techniques to reduce the representation of the majority class. Here again, discarding the majority class samples can lead to the loss of valuable information present in the dataset, potentially affecting the model's ability to learn the underlying patterns and lead to biased predictions particularly if the retained samples do not sufficiently represent the actual distribution of the majority class. The benefit from this is that the minority classes increase its proportion of occurrence and the overall computational time decreases. Concluding, this means that under- and oversampling techniques can increase the performance of the prediction model, but its use come at a cost and its impact should therefore always be compared when choosing the best performing prediction model. The oversampling technique considered in this research is called SMOTE (Synthetic Minority Over-sampling Technique). SMOTE generates synthetic samples by interpolating between existing minority class samples, aiming to maintain the underlying characteristics of the minority class. The undersampling technique used is called Expectation-Maximization (EM), which forms a subset of majority classes based on the probability distribution formed by a mixture of Gaussian's of these samples. SCUT is a technique that combines both of these techniques into one algorithm. The implementation of these method is discussed in chapter 5.3.

4.2 Machine Learning Algorithms

In this chapter, we delve into the details of each machine learning algorithm technique utilized in our empirical study, explaining their principles, strengths and weaknesses. The chosen techniques follow from the SLR conducted in chapter 2 and covers a diverse set of machine learning approaches, including deep-learning methods as neural networks, ensemble methods like Random Forest and XGBoost, but also statistical method like multinomial logistic regression, support vector machines (SVMs), linear discriminant analysis,

and Naïve Bayes. As you might have noticed we regard all of these techniques as machine learning algorithms throughout the thesis. Each method brings its unique set of capabilities and advantages, making them suitable for different types of classification tasks and applicable to compare the best performing metrics against each other. It is worth to mention that none of the algorithms account for the order of classes. This is because the exact nature of the ordinal relationship of CRC is uncertain. However at a later stage this will be included by means of a cost matrix, punishing the false prediction based on the mismatch of classes.

Furthermore, we explore the concept of feature selection and dimensionality reduction using Recursive Feature Elimination with Random Forest (RFE-RFE). High-dimensional datasets can pose challenges in terms of computational complexity and model performance. RFE-RF offers a solution to address these issues by identifying relevant subsets of features. While other techniques like Lasso or Ridge regression could also be considered, we opt for RFE-RF to maintain consistency with the intuition behind the use of Random Forest in our study.

4.2.1 Random Forest

Random Forest Algorithm

The Random Forest (RF) algorithm is a machine learning technique capable of handling multi-class classification problems. The algorithm uses multiple decision trees (DT) to asses its prediction. In most cases the power to accurately predict using RF is stronger than using DT on its own [42], for this reason DT as a standalone algorithm is not considered in this research. However, the underlying concept of DT is important to understand how the RF algorithm operates.

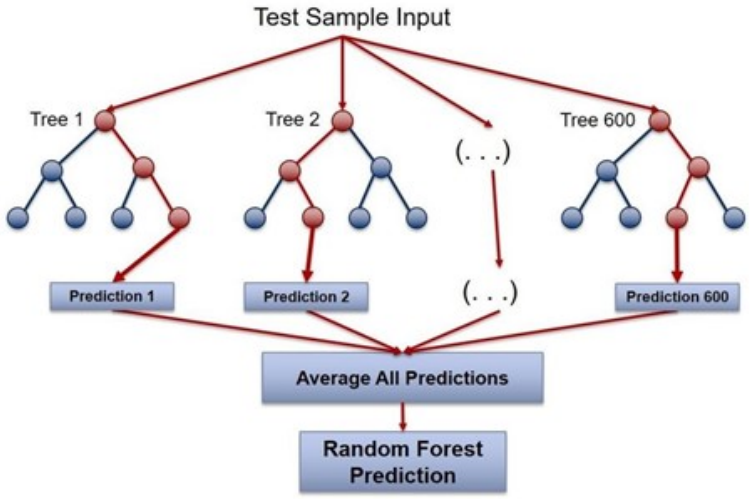

FIGURE 4.1: This figure shows how RF uses decision trees to asses the definite prediction that follows from the average of all predictions (the most voted class)

In figure 4.2 you can see how the decision trees act like a flowchart following decisions made at each node. The set of splits form a tree that always ends with one of the four classifications as an output. Each tree learns from a subset of the data and makes its own set of rules to form a decision. This is how the standard DT operates. Now RF does not create one big decision tre (i.e. Tree 1 of figure 4.2), but creates many trees. Each tree is trained on random different subsets of predictors. As such, Random Forest is basically

an Ensemble of DTs in a parallel matter [27]. It thereby aims to predict the class label or category to which the data belongs. The order or ranking of classes is generally not considered necessary, as Random Forest focuses on correctly assigning data points to their respective classes based on the provided features neglecting any order. The prediction that follows is based on output of the multitude class of trees that are created, where each tree will "vote" for a particular class. The class with the most votes is chosen as the prediction outcome. The prediction outcome is compared to the actual test data, hence a confusion matrix constitutes that can be evaluated on different metrics.

To have a good ensemble, base classifiers are to be diverse (i.e. they predict differently) and accurate whereby adding endless amounts of tree does not improve accuracy (hence, there is an optimum) [26]. Therefore one should choose hyper-parameters to optimize the trained model. This optimization can be achieved using 5-cross fold validation tuning the parameters on the one fold as the validation set while using the other 4 to train them - more on this method is discussed in section 5.5. The eventual hyper tuned parameters for RF - being the optimal number of trees and the maximum depth of each tree - are selected as input parameters the final model. Random forest is widely used as a ML algorithm for classification with its strong ability to automatically reduce over-fitting, handle large data sets and is capability of reflecting on its significant predictors [26]. Therefore we also utilize RF as a feature selection technique to reduce dimensionality.

Recursive Feature Elimination using Random Forest

High dimensionality poses a challenge for training machine learning models, as they can become prone to overfitting. This is because more dimensions (predictors) increases the complexity of models, making them more likely to capture noise and spurious correlations in the training data. While the same pattern is not assessed in the test data. Feature (predictor) selection techniques have proven to be useful in processing high-dimensional data and in enhancing learning efficiency of the trained models i.e computation time. It is referred to as the process of obtaining a relevant subset from an original feature set according to certain feature selection criterion. It plays a role in compressing the data processing scale, where the redundant and irrelevant features are removed [11].

Our research uses the Recursive feature elimination - Random Forest (RFE-RF technique, which is a supervised machine learning methods (based on RF) capable to work with high dimensional data and multi-class classification. As the name suggests, it does so recursively, starting with all features in the set and iterates over all possible subset of features. The Gini Importance is calculated for each iteration and computes the total decrease in node impurity, which indicates the features marginal contribution to minimizing the impurity (uncertainty) impact on the model's decision-making process. Features causing higher impurity decreases are considered more important. Once a certain subset outperforms another, it overwrites the selected features as the most important features and form to be the eventual vector of predictors inputted in the model [16].

4.2.2 eXtreme Gradient Boosting

Extreme Gradient Boosting (XGB) is another commonly used machine learning algorithm ([14] [15] [18] [22]). XGB works quite similarly as random forest. Like Random Forest, it is an ensemble method of decision trees, but instead of using multiple decision trees in parallel (as in Random Forest), XGBoost builds trees sequentially in a gradient-boosting framework. This means that the algorithms builds trees one at a time, but corrects each tree on the errors made. It uses a gradient descent optimization technique to correct its

errors, which is an iterative procedure that aims to minimize a loss function by adjusting the model's parameters in the direction that reduces the error ([15]). Throughout the iterations, XGB monitors the performance on a separate 5-cross fold validation dataset to determine that best hyper parameters.

4.2.3 Radial - Support Vector Machine

Support Vector Machine (SVM) is another powerful machine learning algorithm used for multi-class classification [8] [5] [14] [17], relying on statistical principles. It aims to find the optimal hyperplane that maximizes the margin between different classes while minimizing classification errors. This wider margin signifies greater classification confidence. SVM efficiently handles high-dimensional datasets using kernel functions, such as the radial basis function (RBF), to map data into higher-dimensional spaces. The kernel trick avoids explicit data mapping, enhancing computational efficiency, particularly beneficial for large datasets. Hyperparameters, including the choice of kernel and regularization strength (C), are fine-tuned through techniques like 5-fold cross-validation to optimize the model's performance for robust multi-class classification. The algorithm identifies support vectors, data points closest to the hyperplane, to determine its position and orientation, ensuring balanced class separation.[17].

4.2.4 Neural Networks

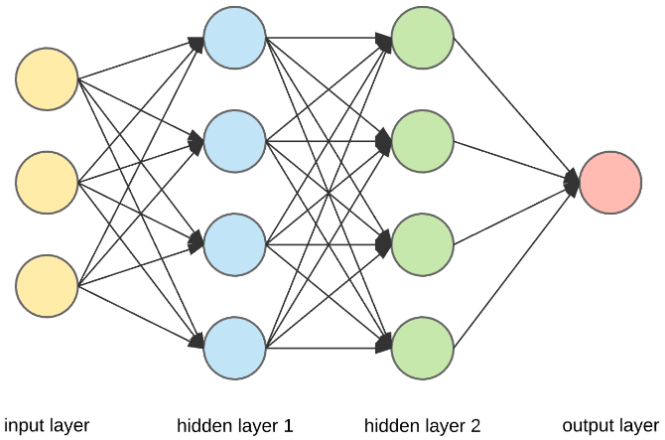

FIGURE 4.2: This figure shows how Feed-forward NN uses nodes from input, hidden and output layers to constitute a prediction (in our case there are 4 output nodes in the layer)

Neural networks (NN) is a machine learning algorithm inspired by the structure and functioning of the human brain. Many different types of NN exist. In this research we deploy Feed-forward Neural Network. It is commonly used for classification problems [8] [37] [14] [34]. Figure 4.2 shows how NN is composed of interconnected nodes (called neurons) that are organized into layers, the layers function as the foundation to make the predictions. First the data is put into the input layer, than one or more hidden layers make decisions, and an output layer outputs the class. Each layer in a neural network performs specific computations on the data it receives. The interconnected nodes (neurons) within a layer – the hidden layers - are responsible for processing information and passing it to the next layer. The layers collectively work as the foundation for the network to process input data and make predictions as an output. During training, the network adjusts the weights assigned to connections between the neurons in each layer

to minimize the difference between predicted and actual outputs. This process, known as back-propagation, uses optimization techniques to fine-tune the weights, improving the network's ability to make as accurate predictions as possible [34].

4.2.5 Multinomial logistic regression

Multinomial logistic regression (MLR is an extension on logistic regression specifically designed for multi-class classification problems. It uses a softmax function (also known as the normalized exponential function) to model the probabilities of each class. It utilizes the linear combination of predictors into probabilities that sum up to one across all classes. The model estimates coefficients for each predictor variable per class. These coefficients are trained using the maximum likelihood estimation. The method assumes that the relationships between the predictors and the log-odds of each class are linear. Log odds refer to the logarithmic likelihood of an event happening compared to the likelihood of it not happening. It also assumes that the predicted errors are independent and follow a multinomial distribution. Since the exact nature of the ordinal relationship is uncertain, we opt not to run the algorithm as an ordinal logistic regression.

4.2.6 Linear Discriminant Analyses

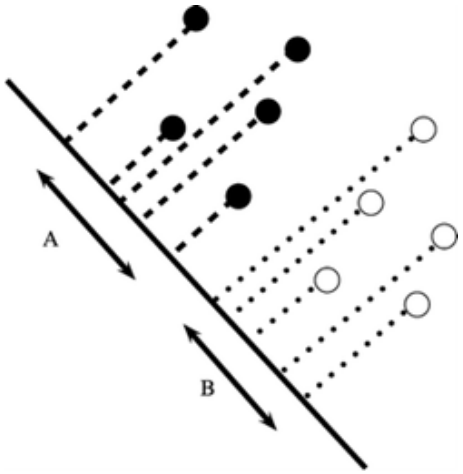

FIGURE 4.3: This figure is to improve intuition behind LDA. Suppose you plot data points in two dimensions (a flat surface) that represent different classes. When these points are projected onto a line (a lower-dimensional space), the line should be chosen in such a way that the projected points from different classes are as far apart from each other as possible along that line in order to maximize separately.

The linear discriminant analysis (LDA) is a fundamental data mining method originally proposed by R. Fisher dating back to 1936 [19]. The concept behind Linear Discriminant Analysis (LDA) is to find a lower-dimensional subspace in which the data points from the original high-dimensional dataset become more separable or distinguishable. In other words, LDA aims to transform the data into a new space where the different classes or categories are well-separated, making it easier to classify or discriminate between them. Figure 4.3 tries to visualize this concept by plotting two dimensions into one. LDA is a statistical method where the separability is defined in terms of statistical measures of mean value and variance. The original algorithm was proposed for binary class problems but multi-class generalizations have been developed in more recent years [44]. The LDA

assumes linear separability in the dataset, so it might not perform well as not all predictors are linearly separable.

4.2.7 Naïve Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem. It is called "naive" because it assumes that the presence of a particular feature in a class is independent of the presence of other features. It use conditional probabilities to determine the prediction. This means that when Naive Bayes calculates conditional probabilities, it's assessing the likelihood of a specific class being the correct prediction given the observed values of individual features, assuming that these features are independent within each class. The Naive Bayes model that we use assumes features to follow a Gaussian distribution. Limitations with this method is expected as the assumption of feature independence might not hold true to our dataset. Because this technique is quite naïve, we regard the naïve classification method as our minimum metric to improve in order to see if other algorithms are truly viable [33].

4.3 Evaluation Metrics for multi-class classification problems

In this section we delve into the evaluation metrics used to asses the performance of our prediction model. Each prediction model is trained based on a specified machine learning algorithm. Once trained, a prediction is deployed based on a new set of unseen data, called test data. This unseen data contains the same type of predictors as have been put into the training model. The most likely class predicted – based on the prediction algorithm used - will be checked for its actual outcome of the specific client. Further elaboration on the train-test-validation split is discussed in chapter 5.2. For now, we only focus on the intuition and metrics used to asses the performance of the prediction model.

One should consider the predictions as a vector $\hat{y}$ and the actual classifications a vector y. In this way both vectors can be compared. Within both vectors, each row denotes one of the class possibles discrete classifications. Given a total range of 4 classifications, n equals 4. Hence, the possible unique combination of values for both of these vectors are given to be $n^2 = 4^2 = 16$. The occurrences of these combinations are presented in a so-called confusion matrix. Interpreting a binary confusion matrix is more straightforward compared to multi-class confusion matrices. In this section we dive into the evaluation metrics of the multi-class confusion metrics as denoted by Grandini [21].

ACTUAL classification	Classes	PREDICTED classification: a	b	c	d
	a	TN	FP	TN	TN
	b	FN	TP	FN	FN
	c	TN	FP	TN	TN
	d	TN	FP	TN	TN

ACTUAL classification	Classes	PREDICTED classification: a	b	c	d	Total
	a	50	37	24	39	150
	b	10	480	5	3	498
	c	14	10	765	1	790
	d	0	2	9	101	112
	Total	74	529	803	144	**1550**

FIGURE 4.4: This figure shows a fictionalized example of a 4x4 confusion matrix. The matrices are viewed from the b-class perspective. In practice the evaluation metric loops the TP over the diagonal n-class= 4 times, thus n=4. Consequently the FP, TN, FN change following the TP position on the diagonal. The single metrics that form is an (weighted) average of all these perspectives combined.

In figure 4.4 an example of the 4x4 confusion matrix is shown. The sum of the total number of times that a combination is observed is denoted in each corresponding cell of the matrix. Here each possible observation is denoted either a true positive (TP), false positive (FP), true negative (TN) or false negative (FN) observation. Considering 4-class classification, one needs to measure the metrics for each class perspective, after all the TP can be either one of the cells on the diagonal. To do so, the metric iteratively runs the TP class perspective along the diagonal. Consequently, a single metric can be constructed by taking the average of all perspectives combined [21]. Repeating this for all n iteration the ith iteration also switches position of the TP, TN, FN. Here the True Positive (TP) indicate the correctly classified units for a class, False Positive (FP) and False Negative (FN) indicate the wrongly classified elements on the predictions and actual classes respectively. True Negative (TN) are all the other tiles, and will later show not to be very important to calculate any of the metrics. This research considers a multitude of metrics to evaluate the performance of each model inspired from the paper on multi-class classification metrics by Grandini [21]: Accuracy, Avg. Balanced Recall (Accuracy), Avg. Weighted Recall (Accuracy), Avg. Balanced Precision, Avg. Weighted Precision, Avg. Balanced F-1 Score, Avg. Weighted F-1 Score and the G-Mean.

$$Accuracy = \sum_{i=1}^{n} \frac{TP(i)}{TP(i) + TN(i) + FP(i) + FN(i)} \tag{4.1}$$

Accuracy: Accuracy is considered to be the most used classification evaluation metric, the accuracy returns an overall measure on the models power to correctly predict the classification of a single individual client. It is an average measure which is - as previously discussed - a rather unsuitable metric for imbalanced datasets. This is because it does not consider the class distribution. In our dataset CRC[D] only makes up for 0.711% of the dataset, if we would predict all other classes correctly and all CRC[D] falsely, we would still achieve an accuracy of 99.289%. Which is considered a model that can predict very accurately, but misses out on the objective to accurately classify each class.

To gain more intuition behind the metric we calculate it using the fictionalized example in figure 4.4:

$$Accuracy = \frac{50}{1550} + \frac{480}{1550} + \frac{765}{1550} + \frac{101}{1550} = 0.90 = 90\% Accuracy \tag{4.2}$$

To check if the accuracy is not misleading due to the imbalanced classes, we review the rest of the metrics from both a balanced and a weighted perspective. By providing the metrics both as a balanced metric as well as a weighted metric, we can interpret the accurateness of the model from the perspectives in chapter 4.1. Which first considers accurateness as predicting each class equally important (hence balanced) and then from the weighted proportional perspective. Here the weights can be any desired vector, but we regard the weights as the proportion that the class is present in the dataset as denoted in table 3.8. Because of this the weighted metrics allow us to indicate the pure accurateness of the model across all classes. We first start with the recall/accuracy metrics of the 4x4 confusion matrix.

$$Avg.\ Balanced\ Recall = \frac{1}{n} \sum_{i=1}^{n} \frac{TP(i)}{TP_{(i)} + FNs_{(i)}} = \frac{1}{n} \sum_{i=1}^{n=4} Recall(i) \tag{4.3}$$

The Avg. Balanced Recall calculates the proportion of the correct predictions over the total amount of actual samples for that class, it recalls the models sensitivity to correctly identify positive instances among all actual positive instances. Recall is therefore sometimes also referred to as the sensitivity. It can be regarded as the row performance metric over the diagonal. After each iteration is calculated it balances the measurements over the diagonal by taking the arithmetic mean of each iteration, hence the Balanced Recall constitutes. This is in fact the same as calculating the balanced accuracy, where each iteration of the accuracy would be weighted by a factor 1/n = 0.25. It is thus "balanced" because every class has the same weight and the same importance. A consequence is that smaller classes eventually have a more than proportional influence on the formula. Intuitively this make sense if the data set is quite balanced, i.e. the classes are almost the same size, Accuracy and Balanced Recall/Accuracy tend to converge to the same value [21]..

Given the fictionalized example in figure 4.4, the avg. Balanced Recall is calculated as follows.

$$\begin{aligned} \textit{The Avg. Balanced Recall} &= \frac{1}{4}\left(\frac{37}{50+27+24+39} + \frac{480}{10+480+5+3}\right. \\ &\quad \left. + \frac{765}{14+10+765+1} + \frac{101}{0+2+9+101}\right) \\ &= 0.7746 \\ &= 77.46\% \text{ Avg. Balanced Recall} \end{aligned} \tag{4.4}$$

From this point we assume that the examples demonstrate the calculation of the metrics sufficiently and we only continue with the plain formula for the other metrics.

$$\textit{Avg. Weighted Recall} = \sum_{i=1}^{n} \frac{TP(i)}{TP_{(i)} + FNs_{(i)}} \times w_{(i)}, \text{ where } \sum_{i}^{n} w_{(i)} = 1 \tag{4.5}$$

The Avg. Weighted Recall takes advantage of the Balanced Recall formula by multiplying each recall by the weight of its class([i]), which we consider to be a vector with the proportion that each class is present in the entire dataset. As argued, one can consider any weights by changing value in the vector w([i]). Avg. Weighted Recall is useful as it allows to separate algorithm performances based on different class weights, because of this we may influence classes that are of less importance to classify. Note that the avg. balanced recall/accuracy is the same as the avg. weighted recall/accuracy if w_([i])=0.25. Moreover, if - as in our case - we assign the weight vectors equal to the class proportion distribution, Hence, for our case, $w_{(i)} = [0.2668, 0.60069, 0.1254, 0.711]$. By doing so the avg. weighted recall/accuracy becomes just the same as the accuracy, proof on this can be found in appendix.5.

$$\textit{Avg. Balanced Precision} = \frac{1}{n}\sum_{i=1}^{n} \frac{TP(i)}{TP_{(i)} + FPs_{(i)}} = \frac{1}{n}\sum_{i=1}^{n} Precision(i) \tag{4.6}$$

$$\textit{Avg. Weighted Precision} = \sum_{i=1}^{n} \frac{TP(i)}{TP_{(i)} + FPs_{(i)}} \times w_{(i)}, \text{ where } \sum_{i}^{n} w_{(i)} = 1 \tag{4.7}$$

The Avg. Balanced Precision and **Avg. Weighted Precision** follows the same principle. However, it indicates how "precise" the true positive prediction are over the

total number of attempts to predict a certain class. In the test set the total number of actual classifications per class is always the same, but the total number of attempted predictions is not. The precision is therefore especially interesting as the metric also shows how convinced the algorithm is that a classification pattern is discovered and newly observed for a certain class. It can be regarded as the column performance metric over the diagonal. Both are put into one metric by calculating the metric balanced as well weighted to its proportion [21].

$$\textit{Avg. Balanced F1-score} = \frac{1}{n}\sum_{i=1}^{n} 2 \times \frac{\text{Precision}_{(i)} \times \text{Recall}_{(i)}}{\text{Precision}_{(i)} + \text{Recall}_{(i)}} \tag{4.8}$$

$$\textit{Avg. Weighted F1-score} = \sum_{i=1}^{n} 2 \times \frac{\text{Precision}_{(i)} \times \text{Recall}_{(i)}}{\text{Precision}_{(i)} + \text{Recall}_{(i)}} \times w_{(i)}, \text{ where } \sum_{i=1}^{n} w_{(i)} = 1 \tag{4.9}$$

The F-1 score is commonly used metric in both binary and multi-class classification to measure a model's accuracy. It is calculated using the harmonic mean of precision and recall and provides a balance between these two measures. The F-1 provides a harmonic mean of precision and recall, giving a balanced view that includes the algorithms power to make predictions as well as not miss out on actual classes. Both the precision and recall are calculated for each iteration across TP over the diagonal and the single metric is both constituted as **the Avg. Balanced F-1 score** and **the Avg. Weighted f-1 score** [21].

$$\textit{G-Mean} = \sqrt[n]{\prod_{i=1}^{n} \text{recall}(i)} \tag{4.10}$$

Finally, the geometric mean (G-mean) is the higher root product of each class-wise recall. The g-mean alongside the other metrics can provide a more nuanced evaluation of a machine learning model's performance. It considers the geometric mean across multiple recall values, which can provide a fairer evaluation in scenarios where classes are disproportionate in size such as our dataset. Note that its formula is the balanced accuracy formula but than using the geometric mean, this makes the metric vulnerable to zero values in case there are 0 TPs for a class. It is a use full metric as it immediately shows when a certain class has missed out on predicting a certain class [21].

A common metric (given a 2x2 confusion matrix) that is left out is Specificity. Specificity focuses on the model's ability to correctly identify true negatives and does not consider the classes on its own. This makes the metric unable to provide a comprehensive assessment of the model's overall accurateness across classes.

In this chapter we delved into the methodologies that we use in the next chapter: Empirical Implementation. Furthermore the chapter helped us to answer research question 3b, 3c and 3d.

Chapter 5

Empirical Implementation

In this chapter we reflect on the model development. In Figure 5.1 the concept of the prediction model is visualized. As stated in the problem statement, it functions as an extension on the current EWS application of SAMAS.

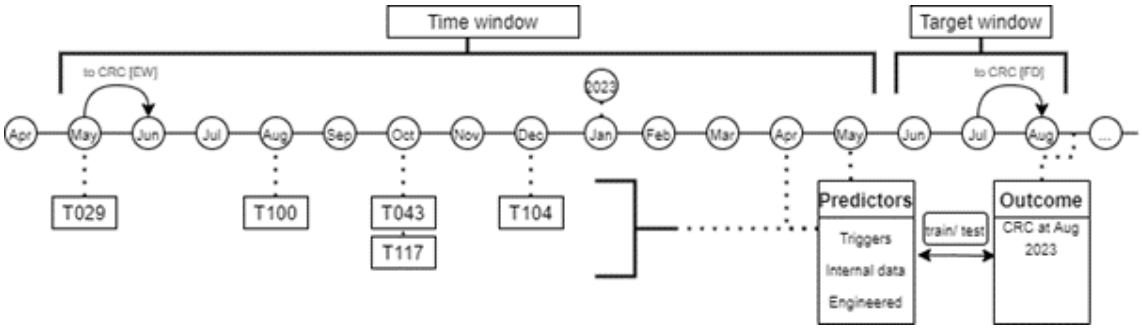

FIGURE 5.1: This figure shows a fictionalised example of how the model operates. The data is reported at May-23 on client level. The predictors contain data from the so-called "time window" interval, internal data is taken at May-23, Feature engineered predictors are take from Apr-23 and May-23 and triggers are active one year to date (May-23). The "target window" describes the time interval that we measure and predict the CRC transition, being Jun-23 to Aug-23.

The figure shows an example of how the model works on client level. In this example, the client received a total of 5 triggers during the time window of 12 months. The time interval is 12 months as clients cannot be in CRC[EW] longer than 12 months. The triggers that are reported however, did not lead to any further CRC transition, hence the client is in CRC[EW]. During the target window - the time that (potential) transitions are measured - at July 2023, it is known that a CRC transitions to CRC[FD] takes place. The clients data known at May 2023 – the snapshot - will therefore be trained for a CRC transition to CRC [FD]. In real time CRC transitions occur as a consequence of CRC Changing triggers, while our prediction model also takes into account internal data and feature engineered predictors known at the time of the snapshot. Because of this, more data can be utilized to discover patterns (based on the ML algorithms to classify the clients. Of course in reality, a mixture of likewise examples occur. The total number of these clients constitute the rows in our merged data-set. The next figure 5.2 depicts a general overview of the steps conducted during the model development.

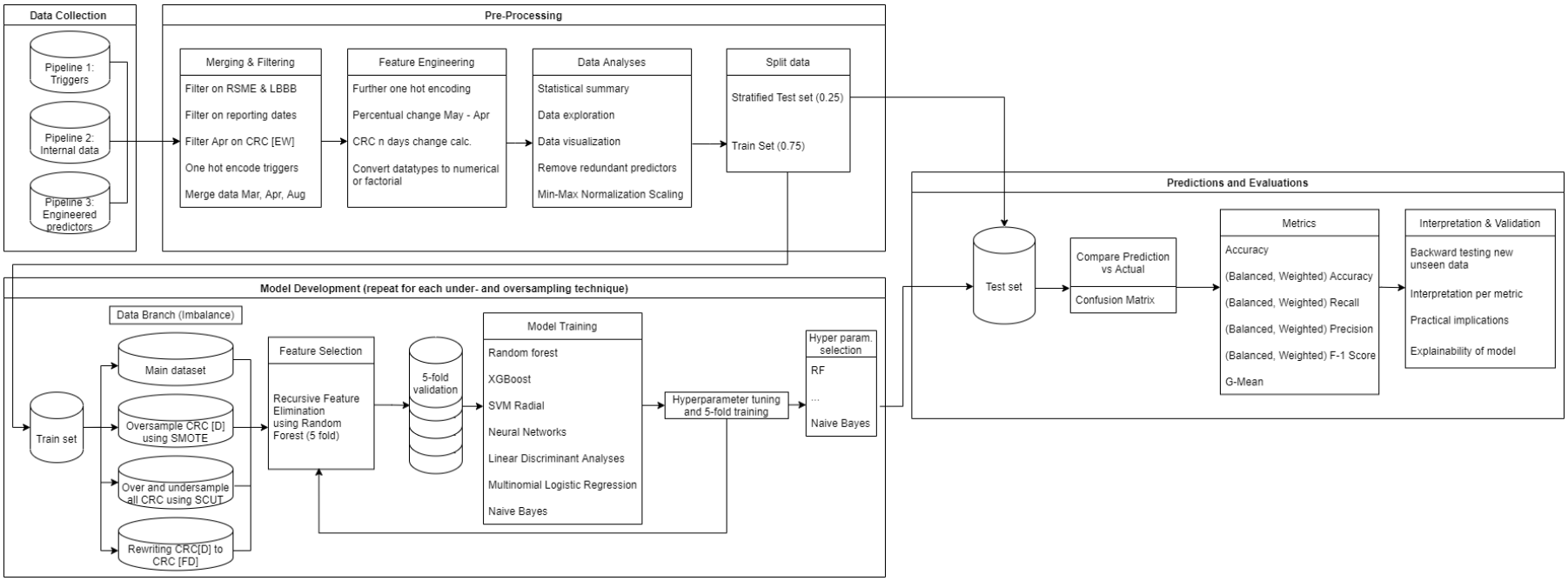

FIGURE 5.2: This figure depicts a step-wise overview of the task conducted during the model development. Elaboration on the steps conducted will be discussed throughout the chapter.

5.1 Pre-processing

Data pre-processing involves several tasks aimed at cleaning, transforming, and organizing raw data into a format that is more suitable and understandable for analysis or machine learning purposes. In this section we delve into the pre-processing steps from figure 5.2. Some of the pre-processing steps have already been put forward in chapter 2, these include steps merging filtering, feature engineering and data analysis (which lead to removal of redundant predictors). After the data analysis on the merged dataset took place, the predictors are scaled using Min-Max Scaling [12], this benefits the computation time of the machine learning algorithms. While it also ensures that all predictors have values within the same range, which can help prevent certain features from dominating the learning process simply because of their larger scale. The three pipelines are extracted from Rabobank server. The data is filtered, converted and merged according to the steps in appendix .3. A total of n clients constitute the merged dataset and are now ready to be split for training and testing.

5.2 Train-Test Split

Each dataset is split into a train-test ratio of 0.75, which is considered a fair rate for an (imbalanced) dataset of our size [31]. It results that 75% of the clients is used for training the models, the training set will also be split up into 5 folds - where each time one fold functions as a validation set - and will be discussed in section 5.5. The test-set makes up for the leftover of 25% and is used as unseen data to test the final performance of each prediction by means of the confusion matrix. The test-set is always kept as a subset of unseen data and is not used for training or tuning the model and its parameters.

5.3 Under- and oversampling Techniques

In chapter 2 we mentioned that we have an imbalanced dataset at hand, To tackle this issue, we propose a branch of the trainset to 4 different under- and oversampling strategies as discussed in section 4.1. Each of the following 4 techniques will be compared to eventually choose the best performing model:

1. The "main dataset" leaves that dataset intact as it was after merging and splitting in the train dataset.

2. The "smote dataset" volves adding minority classes to the dataset based on the over-sample SMOTE (Synthetic Minority Over-sampling Technique) technique. SMOTE generates synthetic samples by interpolating between existing minority class samples, aiming to maintain the underlying characteristics of the minority class.

3. The "SCUT dataset" combines both SMOTE and undersampling methods called SCUT (SMOTE and Clustered Undersampling Technique). SCUT applies under sampling techniques to the class that (in case of 4 classes) present more than 25% of the sample and likewise under sampling for classes less than 25% of the sample. In the unlikely case that the classes rpresent exactly present 25% of the sample, they remain intact. Undersampling is based on the Expectation-Maximization (EM) technique, which forms a subset of majority classes based on the probability distribution formed by a mixture of Gaussians of these samples. The oversampling technique on the minority class follow the SMOTE technique.

4. The "3-class dataset" rewrites all CRC [D] to CRC[FD], this reduces the number of classes to 3 and consequently significantly reduces the class imbalance.

Because of this the class proportion of the train dataset on changes as denoted in table 5.1. We see that we have significantly increased the CRC [D] sample for the 2nd and 3rd method, We also rewritten all CRC[D] to CRC[FD] in the 4th method, to see how the algorithms would operate without the CRC [D] class at all. By comparing each implementation, we can to determine if the prediction model improves in case that the imbalance is mitigated.

#	Aug-23			
	CRC [G]	CRC [EW]	CRC [FD]	CRC [D]
1. Main dataframe (no changes)	26.68%	60.07%	12.54%	0.71%
2. Smote	25.58%	57.58%	12.02%	4.82%
3. SCUT	25%	25%	25%	25%
4. CRC[D] to CRC[FD]	26.68%	60.07%	13.25%	0.0%

TABLE 5.1: The class distribution after a branch in under and oversampling techniques on the original training set. The three methods are implemented to tackle the issue of imbalance of the severe minority class: CRC [D].

5.4 Recursive Feature Elimination - Random Forest

In section 4.2.1 we discussed that we utilize Random Forest as a Recursive Feature Elimination Technique to select the optimal predictors for each branched training set. During the process the model evaluates per subset of predictors the accuracy using cross fold validation. Based on this metric the RFE-RF selects the subset of features that maximizes the performance metric. The RFE-RF gives us the following set of predictors per branched dataset as depicted in table 5.2.

TABLE 5.2: Selected and removed predictors for each branched training set following the RFE-RF.

Main df	Smote df	Scut df	CRC [D] to CRC [FD] df
Selected n = 37	**Selected n = 31**	**Selected n = 30**	**Selected n = 22**
PD	PD	PD	PD
RRR	RRR	CRC_n_days	RRR
T130_once	T130	CRC_prev	T130
T130	T130_once	Sector	T130_once
O_Exp	RWA	d_EAD	O_Exp
T044	EAD	d_LGD	RWA
T044_once	CRC_n_days	d_O_Exp	T044
RWA	O_Exp	RRR	EAD
EAD	CRC_prev	T130_once	T044_once
CRC_n_days	T100	T130	CRC_n_days
T025_once	T100_once	RWA	CRC_prev
T025	T030_once	d_RWA	T025
CRC_prev	T044	LGD	T025_once

Continued on next page

Table 5.2 – Continued from previous page

Main df	Smote df	Scut df	CRC [D] to CRC [FD] df
T043_once	T044_once	EAD	T043_once
T043	T030	d_PD	T043
d_RWA	T131_once	O_Exp	T100
T100	Sector	T100	T100_once
T030_once	T131	T100_once	d_RWA
T100_once	d_RWA	T030_once	T030
T030	T025	T030	T030_once
T070	d_EAD	T131_once	T070
LGD	LGD	T044_once	CRC_Outcome_Y
d_PD	d_PD	T131	**Removed n = 19**
d_EAD	d_O_Exp	T044	New_EW_6
T029	T025_once	New_EW	New_EW
New_EW_3	d_LGD	T025	New_EW_3
T029_once	T029	T025_once	Sector
d_LGD	T029_once	New_EW_3	LGD
d_O_Exp	T043	New_EW_6	d_PD
T131	New_EW_3	CRC_Outcome_Y	d_EAD
T117	CRC_Outcome_Y	**Removed n = 11**	d_O_Exp
T131_once	**Removed n = 10**	T009	d_LGD
Sector	New_EW_6	T029	T009
New_EW	New_EW	T043	T029
T117_once	T009	T070	T104
T009_once	T070	T104	T117
CRC_Outcome_Y	T104	T117	T119
Removed n = 4	T117	T119	T131
New_EW_6	T119	T009_once	T009_once
T009	T009_once	T029_once	T029_once
T104	T043_once	T043_once	T117_once
T119	T117_once	T117_once	T131_once

As can be seen, the over- and under sampling techniques are able to significantly to reduce the number of predictors required to asses separability of classes. Here the triggers and "new_EW_" predictors are eliminated in most cases.

5.5 Cross-fold validation and Hyper-parameter tuning

Now that we have determined and selected the final set predictors for each of the branched, the training set is prepared for 5-fold cross-validation. This method divides the training set into 5 equally sized subsets (folds), each being 1/5*75% (training set)= 12.5% of the merged dataset. These folds are used to iteratively evaluate the model during training. The amount of iterations are equal to the amount of folds. This is because one fold serves as the validation set while the remaining four folds act as the training set. This allows the model to learn from the training data while validating its performance against the fold kept aside for validation. Additionally, within each iteration, the model fine-tunes its hyper-parameters by exploring the entire grid of potential options to identify the optimal settings. The 5 fold cross validation offers a more robust estimation of the model's performance by averaging the results from the 5 rounds of training and by using the validation fold for

hyper-parameter tuning [38].

5.6 Trained Models and Hyper-parameters

A multitude of models are trained according to the methodologies of the machine learning algorithms discussed in section 4.2. During training the cross-fold validation set allows to find the optimal hyper-parameters over 5-folds from the training set. These optimal parameters are put into the algorithms final model of which this optimization is either performed using ransom search. A grid search would provide more precise parameters, but increase computational time significantly [7]. The eventual optimal hyper parameters vary depending on the under and oversampling technique. We include 7 algorithms being:

- **Random Forest**: Tuned hyperparameters include `ntree` (number of trees) and `mtry` (maximum features per split).
- **XGBoost**: Tuned hyperparameters include `nrounds` (number of boosting rounds), `max_depth` (maximum tree depth), and `eta` (learning rate).
- **SVM Radial**: Tuned hyperparameters include `cost` (regularization parameter) and the radial kernel parameter `sigma`.
- **(Feedforward) Neural Networks**: Tuned hyperparameters include the number of hidden layers, neurons per layer, `learning_rate`, and `dropout`.
- **Linear Discriminant Analysis**: No hyperparameter tuning was used.
- **Multinomial Logistic Regression**: No hyperparmeter tuning was used.
- **Naïve Bayes**: No hyperparmeter tuning was used. The Naïve Bayes is considered to be the dummy model and is used to compare if the other algorithms indicate improvements.

5.7 Model Evaluation

Each branched training set is trained on each of the 7 algorithms. After cross-fold validation and hyper parameter tuning, the predictions are compared against the unseen test data. Figure 5.2 indicates that the test set is split accordingly. The vector containing the predicted classifications are compared against the vector of actual classification and are constituted in the confusion matrix as discussed in section 4.3. This confusion matrix allows us the evaluate the model by means of calculating the metrics as listed below.

- **Accuracy**: Measures the proportion of correctly predicted instances.
- **Avg. Balanced Recall (Accuracy)**: Calculates balanced recall for each class and takes the average. It accounts for class imbalances.
- **Avg. Weighted Recall (Accuracy)**: Similar to Avg. Balanced Recall but uses weighted averages based on class sizes.
- **Avg. Balanced Precision**: Calculates balanced precision for each class and takes the average. It measures the precision of predictions.
- **Avg. Weighted Precision**: Similar to Avg. Balanced Precision but uses weighted averages based on class sizes.

- **Avg. Balanced F-1 Score**: Calculates balanced F-1 score for each class and takes the average. It balances precision and recall.
- **Avg. Weighted F-1 Score**: Similar to Avg. Balanced F-1 Score but uses weighted averages based on class sizes.
- **G-Mean**: The geometric mean of sensitivity and specificity. It balances the trade-off between sensitivity and specificity, useful for imbalanced datasets.

The accurateness of the model is perceived from the two perspectives denoted in chapter 4.1 and is assessed primarily through the F-1 score.

1. **Avg. Balanced F-1 Score:** Accurately predicting each class is considered equally important, any imbalance in the classes is balanced out. Conducted first as 4-class then as a 3-class.
2. **Avg. Weighted F-1 Score:** Accurately predicting each class is considered weighted to the proportion that the class is present in the data set, any imbalance is balanced to its proportionalized weights. Conducted first as 4-class then as a 3-class.

The other metrics will be used to dive deeper into the models performance and can be utilized to capture distinct aspects of a model's performance. In short, and more deeply elaborated in section 4.3, accuracy focuses on overall correctness, Precision measures the preciseness of positive classes, recall measures the sensitivity for false negative classes, while F-1 shed light on both the model's performance concerning positive and false negative classes. Using multiple metrics provides a more comprehensive understanding of the model its behavior.

This chapter reflected on the steps conducted during the empirical implementation. It argued why certain decisions where made and how they were implemented. Additionally they helped to answer research questions 3b, 3c, 3d and 4a. In the next chapter we discuss the interpretation of the empirical results.

Chapter 6

Emperical Results

6.1 Result interpretation & Model Comparison

In this chapter we present the results, they are put in an overview in table 6.1 and denote each ML algorithm per under- and oversampling technique each denoted as a 'branched dataset". We use this section to compare and select the best model based on the accurateness perspectives defined.

DF_Main	*RF*	*XGB*	*SVM*	*NN*	*MLR*	*LDA*	*NB*
Accuracy	0.7863	0.7815	0.7642	0.6003	**0.7911**	0.7545	0.6217
Avg. Balanced Recall	0.5119	0.5194	**0.5776**	0.2588	0.5064	0.4765	0.3275
Avg. Weighted Recall	0.7863	0.7815	0.7606	0.6003	**0.7911**	0.7545	0.6217
Avg. Balanced Precision	0.5886	0.5802	0.4665	0.3539	**0.6002**	0.5710	0.3135
Avg. Weighted Precision	0.7850	0.7807	0.7642	0.5213	**0.7883**	0.7561	0.4538
Avg. Balanced F-1	0.5259	**0.5319**	0.4868	0.2100	0.5315	0.5058	0.3013
Avg. Weighted F-1	0.7725	0.7713	0.7416	0.4644	**0.7767**	0.7415	0.5110
Avg. G-Mean	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
DF_Smote	*RF*	*XGB*	*SVM*	*NN*	*MLR*	*LDA*	*NB*
Accuracy	0.7828	**0.7863**	0.7234	0.7593	0.7835	0.7483	0.5546
Avg. Balanced Recall	0.5134	**0.5178**	0.4892	0.4608	0.5036	0.4681	0.4896
Avg. Weighted Recall	0.7828	**0.7863**	0.7157	0.7593	0.7835	0.7483	0.5546
Avg. Balanced Precision	0.5894	0.5828	0.4710	0.5559	**0.5921**	0.5683	0.4567
Avg. Weighted Precision	**0.7838**	0.7831	0.7234	0.7489	0.7832	0.7515	0.7572
Avg. Balanced F-1	0.5295	**0.5337**	0.4789	0.4786	0.5287	0.4985	0.4147
Avg. Weighted F-1	0.7716	**0.7757**	0.7182	0.7361	0.7720	0.7338	0.5838
Avg. G-Mean	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
DF_SCUT	*RF*	*XGB*	*SVM*	*NN*	*MLR*	*LDA*	*NB*
Accuracy	0.7268	**0.7420**	0.6902	0.5145	0.5802	0.5539	0.4039
Avg. Balanced Recall	0.5396	**0.5495**	0.4496	0.5402	0.5485	0.5049	0.4198
Avg. Weighted Recall	0.7268	**0.7420**	0.6958	0.5145	0.5802	0.5539	0.4039
Avg. Balanced Precision	0.5041	**0.5088**	0.4621	0.4423	0.4788	0.4622	0.4768
Avg. Weighted Precision	0.7607	**0.7677**	0.6902	0.6700	0.7319	0.7054	0.7307
Avg. Balanced F-1	0.5143	**0.5225**	0.4530	0.4237	0.4611	0.4427	0.3092
Avg. Weighted F-1	0.7331	**0.7462**	0.6895	0.5512	0.6208	0.5964	0.3865
Avg. G-Mean	0.0000	0.0000	0.0000	**0.5228**	0.5143	0.4526	0.3386
DF_3class	*RF*	*XGB*	*SVM*	*NN*	*MLR*	*LDA*	*NB*
Accuracy	**0.7939**	0.7911	0.7856	0.7773	0.7891	0.7586	0.4613

Table 6.1 continued from previous page

DF3_ class	*RF*	*XGB*	*SVM*	*NN*	*MLR*	*LDA*	*NB*
Avg. Balanced Recall	0.6932	**0.6960**	0.6606	0.6681	0.6694	0.6339	0.5149
Avg. Weighted Recall	**0.7939**	0.7911	0.7856	0.7773	0.7891	0.7586	0.4613
Avg. Balanced Precision	0.7941	**0.7963**	0.7885	0.7531	0.7857	0.7534	0.6377
Avg. Weighted Precision	0.7977	**0.7984**	0.7864	0.7720	0.7882	0.7582	0.7518
Avg. Balanced F-1	**0.7162**	0.7161	0.6926	0.6950	0.7008	0.6705	0.4410
Avg. Weighted F-1	**0.7842**	0.7820	0.7712	0.7671	0.7761	0.7447	0.4390
Avg. G-Mean	0.6534	**0.6558**	0.6098	0.6322	0.6237	0.5968	0.4273

TABLE 6.1: Overview of the evaluation metrics for each of the branched datasets. Note that the zero value at the G-mean indicates that one of the classes has zero true positives.

As the overview can be quite overwhelming, we consistently zoom in on the specific results of the table necessary. But first some general notes on the result interpretation. A G-mean of zero is explained because one class has 0 True Positive predictions, after investigating this always occurs for the CRC [D] class. The highest value for each of the metric per under and oversampling branch has been emphasized in bold, the under performing algorithms (<0 bold s cells excluding g-mean metrics) are not considered for further inspection to determine the best model. These algorithms include NN and LDA - NB remains included for comparison purposes.

Furthermore, when we look at the metrics of df_main on its own, we see that df_main shows a variety of best performing models per metric. The Avg. Balanced F-1 score is the highest for XGB but, MLR scores better on almost any other metric. After the techniques of SMOTE and SCUT are applied, the XGB-Smote tend to out perform MLR-main again on the balanced F-1 score. However, the rest of the metrics do not necessarily increase. More importantly, even after applying the over- and undersample techniques, the CRC[D] class remains not to have any True Positives and does not indicate consistent improvements on metrics at all. This is even the case after the SCUT method allowed the model to train on 1448 CRC[D] samples. We can therefore argue that the under- and oversampling techniques on the dataset have no significant impact on the separability of classes. Its main purpose was to get a grip on the ability to predict the CRC[D] class and even though the balanced metric seems to improve marginally in XGB Smote, this remains unachieved. Further explanation on the problematic class will be discussed in more detail in the next question. For now we conclude that the over and undersample techniques did not manage to get a hold on the CRC [D] class, and its overall implementation did not benefit the performance metrics consistently disregarding expectations.

1. Which model performs best when we consider that accurately predicting each class equally important?

To answer this question we have to look at the balanced metrics, we argued that the Avg. F-1 score is our most indicative metric to answer the question. Therefore ,we zoom in on the Avg. F-1 score per branch in table 6.2

Avg. Balanced F-1	RF	XGB	SVM	NN	MLR	LDA	NB
Main	0.5259	0.5319	0.4868	0.2100	0.5315	0.5058	0.3013
Smote	0.5295	**0.5337**	0.4789	0.4786	0.5287	0.4985	0.4147
SCUT	0.5143	0.5225	0.4530	0.4237	0.4611	0.4427	0.3092

Table 6.2 continued from previous page

Avg. Balanced F-1	RF	XGB	SVM	NN	MLR	LDA	NB
3-Class	**0.7162**	0.7161	0.6926	0.6950	0.7008	0.6705	0.4410

TABLE 6.2: Avg. balanced F-1 score performance per branch and model

Perceiving the prediction model as a 4-class problem, We see that XGB is the best model outperforming the NB by +0.1189. However a f-1 score around 0.5 might be considered acceptable in some cases, it is in general not showing sufficient predictive powers. We therefore further inspect the confusion matrix of XGB-Smote to gain a more detailed view of the models performance in table 6.3.

Prediction (row) Actual (Col)	CRC [G]	CRC[EW]	CRC[FD]	CRC[D]
CRC [G]	308	78	0	0
CRC[EW]	95	756	17	1
CRC[FD]	23	84	73	1
CRC[D]	2	6	2	0

TABLE 6.3: This table displays the transposed confusion matrix for XGB-Smote.

In this table, we see that CRC [D] has only 2 positive predictions (row) of which zero are true positive prediction. Because of this the balanced metrics and thus the Avg. Balanced F-1 score relatively low. Purely looking at the accurateness of XGB-smote we see that indeed that 0.7853 accuracy rate can be misleading when the objective is to accurately predict each class equally important.

Clearly, a problem occurs for the balanced metrics when the CRC[D] class is not mitigated. When we consider the problem as 3-class problem, significant improvements can be perceived in the avg. balanced F-1 score, with RF indicating the highest. Considering the same metric, this model is outperforming XGB-Smote by +0.1825 and the NB-3class by +0.2752. The corresponding confusion matrix used to calculate these metric is depicted below in table 6.4. Together with table 6.1 it shows a strong balanced precision metric (where CRC[FD] contributes a precision of 75/(75+15)=0.83 to this average. Also it indicates strong avg. balanced recall where CRC[EW] recall 759/(759+95+15)=0.87 contributes nicely to the 0.6960 avg. balanced recall.

Prediction (Col) Actual (Row)	CRC [G]	CRC[EW]	CRC[FD]
CRC [G]	314	72	0
CRC[EW]	95	759	15
CRC[FD]	22	94	75

TABLE 6.4: This table displays the transposed confusion matrix for RF-3Class.

This means that the XGB model performs best when we consider accurately predicting each class equally important for a 4-class and RF for a 3-class. However, the 4-class best performing balanced metric remains to be considered too low. While as a 3-class problem the metric significantly jumps to a sufficient balanced metric. This means that one can only achieve an equally accurate prediction model across all classes if the CRC [D] is mitigated and the 3-class problem is considered.

To check if CRC [D] is truly the bottleneck class in these classification. We review the metrics also from the perspective of the models pure accurateness, this is done by weighting the classification based on the actual proportion of the class in the dataset. We would expect that mitigating class imbalance by multiplying these weight will provide equal f-1 scores for both the 3-class and 4-class.

2. Which model performs best when we consider pure model accurateness and weigh the metrics to the class proportions?

We recall that the weighted metrics in our case are a vector of the class proportionalized weights, as stated in chapter 4.3. In theory, the vector can be made interactive by altering its weight accordingly. But sticking to our purpose, the proportionalized weights show the model pure accurate ability to classify. This phenomenon is emphasized by our prove that the avg. weighted recall becomes the same as the accuracy, as shown in appendix.5. Earlier, we stated that the Avg. Weighted F-1 score is the most indicative metric to answer this question on the models pure accurateness. We therefore zoom in on the Avg. F-1 score per branch in table 6.5.

TABLE 6.5: F-1 score performance

Avg. Balanced F-1	RF	XGB	SVM	NN	MLR	LDA	NB
Main	0.7725	0.7713	0.7416	0.4644	**0.7767**	0.7415	0.5110
Smote	0.7716	0.7757	0.7182	0.7361	0.7720	0.7338	0.5838
SCUT	0.7331	0.7462	0.6895	0.5512	0.6208	0.5964	0.3865
3-Class	**0.7842**	0.7820	0.7712	0.7671	0.7761	0.7447	0.4390

TABLE 6.5: Avg. Weighted F-1 score performance per branch and model

If we perceive the problem as a 4-class problem we see that the MLR-Main trained set performances best, with a score of 0.7767. This value can be considered as fairly accurate prediction model and a big improvement on the avg. balanced accuracy of XGB Smote. It also scores the highest on both the Avg. Weighted Recall and Precision. The Confusion matrix is found below in table 6.6. What we see is that the model is quite sensitive on the recall for CRC[EW] and fairly precise for CRC[FD]. However, again it misses out on accurate classifications of CRC[D]. This time however the metric diminishes this impact on the pure accuracy as it limits the class influence by the small proportion of 0.7%. Looking at the descending order of false positives per class, the MLR indicates an ordinal pattern of the CRC. Even though this is not explicitly indicated during training (i.e. we would train an ordinal logistic regression). Furthermore, it shows that machine learning methods (RF, XGB) can still be outperformed by more traditional statistical methods as MLR.

Prediction (Col) Actual (Row)	CRC [G]	CRC [EW]	CRC [FD]	CRC [D]
CRC [G]	286	100	0	0
CRC [EW]	64	790	14	1
CRC [FD]	11	102	68	0
CRC [D]	0	9	1	0

TABLE 6.6: This table displays the confusion matrix for the MLR-Main model.

From a 3-class perspective on the avg. weighted F-1 score, RF is again the best performing model. This confusion matrix can be found in table 6.4.

Now to come to an overall conclusion which model should we take the metric according to the following research objective perspectives as put in table 6.7

	4-Class	3-Class
Avg. Balanced F-1	0.5319	0.7162
Avg. Weighted F-1	0.7767	0.7842

TABLE 6.7: This table shows the average balanced and weighted F-1 scores for both the 4-class and 3-class scenarios.

The first research objective considers predicting each class equally important, To achieve this XGB-Smote is the best for the 4-class problem but still performs poorly with its Avg. Balanced F-1 score of 0.5319 (<0.6). Considering it as a 3-class problem however, the objective can be met and is best predicted by MLR-main with 0.7162 (>0.7). Thus this objective is only achieved for the 3-class problem. When we view the accurateness from the second perspective objective, RF is the best choice for both the 4 and 3 class problems (resp. 0.7767 0.7842). These weighted metrics are in line with the expectations that these values should be about equal, indicating that the minority CRC [D] class is indeed problematic. It is thus clear that continuing the problem as 4 class problem will keep the the CRC[D] class to cause poor predictions, if it is able to predict the class at all. Therefore, we state that under the current methods CRC[D] is an unpredictable and problematic class. Its further reasoning is argued as follows.

1. Using the time window of clients from [EW] does not possess the power to predict CRC [D] within a target window of 3 months. It can be argued that most of the information to why the CRC[D] occurs is only present during the target window, and thus is not used in the training data.

2. The reason that CRC [D] occur are not available in the data at all. Following from direct defaults that are by definition abrupt and therefore hard to predict.

3. There are just too little CRC [D] in the test set. Broadening the scope to other clients apart from RSME might enhance the separability of the class.

Concluding, only accurateness across each class equally is reached when the CRC[D] class influence is mitigated or removed. From the models pure accurate view, the model already shows to be a promising prediction model, hence the problem originates from CRC[D]. Furthermore, the model is also able to separate the order of classes rather well, even though this is not assumed in the algorithms (i.e. Multinomial is used opposed to Ordinal). We can show this in the confusion matrix, here the number of FP and FN per class are decreasing with the CRC [class] order.

Now that we assessed that RF is the best model to predict the classification of the 3-class problem, we also check how the model performed on the validation set. During the 5 cross fold validation an average accuracy of 0.7973 is perceived on the validation sets. The difference is not substantial. Both accuracy's are relatively high, indicating that the model performs reasonably well on unseen data. Furthermore, we check that the estimates the Out of the Bag error rate to be 20.20%. Which is in line with the expectation that the Out of the Bag error rate is equal to 1 - (observed accuracy).

From the point we further embrace the prediction model as a solution to the 3-class problem, as interpreting the 4-class solely by means of weighting metrics is considered cumbersome. This new transition diagram is depicted in figure 6.1. To become more confident

on the definite prediction we explore the options of ensemble methods. After which, we test the model generalization across test data on other time intervals and calculate the cost impact of the FP FN predictions using a cost matrix. We end with conceptual tool on how the prediction model can be implemented into the SAMAS application.

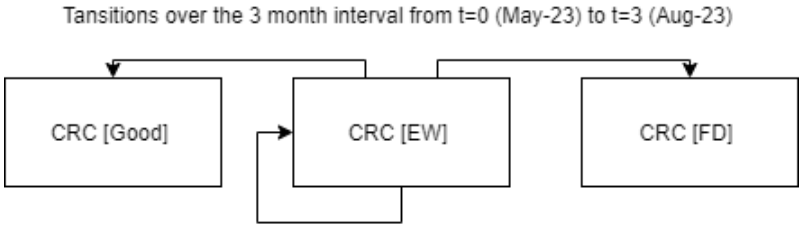

FIGURE 6.1: This figure indicates the transition diagram of the 3-class problem. Here all CRC [FD] classes are rewritten to CRC [D] classes.

6.2 Model optimization: Ensemble Methods

In the previous section, we determined to continue on optimizing and the analysing the problem further as a 3-class. This is because regarding the problem as 4-class classification is cumbersome with an unpredictable CRC [D] class. In table 6.1 we see that that RF is the best performing metric on both the Avg. Balanced F-1 and Avg. Weighted F-1. XGB is performing better on some of the other metrics. SVM and MLR are runner-ups with relatively the same scores per metric, however they showed to be promising in other under- and oversampling branches. We therefore consider an ensemble method for the combination of each of these models. The method is called weighted model voting and uses multiple machine learning model for its final prediction. Here, each model predicts a class and the class with the majority vote is chosen, if there is a tie, the prediction from in order of RF, XGB or MLR is chosen. By means of this ensemble method we are able to to improve the RF model if we combine the votes of the all 4 methods. Here each of the metrics are improved, as can be seen in table 6.8.

DF_ class	*RF*	*Ensemble*	*NB*
Accuracy	0.7939	**0.7974**	0.4613
Avg. Balanced Recall	0.6932	**0.6935**	0.5149
Avg. Weighted Recall	0.7939	**0.7974**	0.4613
Avg. Balanced Precision	0.7941	**0.8006**	0.6377
Avg. Weighted Precision	0.7977	**0.8010**	0.7518
Avg. Balanced F-1	0.7162	**0.7191**	0.4410
Avg. Weighted F-1	0.7842	**0.7880**	0.4390
Avg. G-Mean	0.6534	**0.6550**	0.4273

TABLE 6.8: The performance metrics for the ensemble method by combining the weighted model voting of RF, XGB, SVM, and MLR for the 3-class problem, showing a decent prediction model compared to the Naive Bayes model.

This model will be used to asses to what extent the prediction model can be used in practise. First we will test how well the trained data generalizes on test data from other months.

6.3 Consistent model generalization

The ensemble method using XGB, RF, SVM and MLR is considered the best method to accurately predict each class of the CRC equally well, while it also the more pure accurate model as addressed in the the weighted metrics. However, these metrics are solely tested on unseen data from the same target window Jun-23 until Aug-23. In order to test if the model remains consistent for future use, we should also put in unseen data from time intervals - time window + target window - from other months. By doing so, we can provide insights into the model's generalization across time.

We therefore deploy the trained ensemble prediction model on a random other time interval from the query. Unfortunately, we are somewhat limited in finding new data as the query only possess data from Feb-Aug. We therefore take March as our snapshot such that Feb can be used to feature engineer the percentual change between these months. We aim to predict the CRC in June and do not include clients that are also in the train set, otherwise the samples in the train and test set can overlap. The new test set allows to stratify a significantly larger test set out to make predictions on. The following confusion matrix in table 6.9 is constituted.

Metric	Ensemble_may_aug	Ensemble_mar_jun	Perc. Change
Accuracy	**0.7974**	0.7501	-5.93%
Avg. Balanced Recall	**0.6935**	0.6538	-5.72%
Avg. Weighted Recall	**0.7974**	0.7695	-3.50%
Avg. Balanced Prec	0.8006	**0.8252**	+3.08%
Avg. Weighted Prec	**0.8010**	0.7685	-4.05%
Avg. Balanced F-1	**0.7191**	0.7005	-2.59%
Avg. Weighted F-1	**0.7880**	0.7429	-5.72%
Avg. G-Mean	**0.6550**	0.6263	-4.39%

TABLE 6.9: Performance Metrics Comparison

We see that almost all of the metric seems to perform slightly less on unseen data from another time interval, except for the balanced precision. Moreover, the metrics remain to show fairly good metrics with scores above 0.7. More importantly, the Balanced and weighted F-1 have proven to remain above the threshold. This indicates that the model is able to generalize decently well over time. However this generalization is achieved quite straightforward and we need to emphasize that there is still a decrease in most metrics and a certain consistency is therefore not reassured. Unfortunately, testing the data on more time interval is limited due to the query restriction.

6.4 The prediction model put into practice

In this section we implement a cost matrix to asses the performance against associated costs. As such, we determine if implementing the prediction model is considered costly or cost saving. Previously we already used the confusion matrices its TP, TN, FP, and FN to calculate the precision, recall, and F1-score for each class. Now we will use the TP, TN, FP, and FN and assign costs and rewards to each of these predictions made.

This method of multiplying a cost matrix with a confusion matrix is a technique used in cost-sensitive classification and identifies the impact of different types of misclassifications. In this context, a confusion matrix is typically of a multi-class classification problem with

three classes [38]. The rows of the confusion matrix represent the predicted classes, and the columns represent the actual classes.

The confusion matrix is computed using the ensemble method and is tested against the initial test data from Aug-23 as well as the new test data from jun-23 in table 6.9. This constituted the following confusion matrix.

Prediction (Col) Actual (Row)	CRC [G]	CRC[EW]	CRC[FD]
CRC [G]	843	547	1
CRC[EW]	207	2567	25
CRC[FD]	41	345	343

TABLE 6.10: This table displays the combined matrix of the test data from both time intervals.

The first thing that we want to see is that the falsely predicted (both positive and negative) classification are in descending order, this is indeed the case. Also, we see that the model is more precise than that is sensitive. From a practical point of view, this is preferred because the prediction model is to be seen as an extension to the SAMAS application and thus we care that it provides confident predictions rather than missed predictions that would otherwise be missed without the model anyhow. Anyway, this quick notion is further investigate in the next section. To compute the total cost, we multiply the values of the confusion matrix by the corresponding values of the cost matrix and sum them up. This will give you us overall cost value that accounts for the different types of classifications and their associated costs.

The reason classifications further away from the diagonal should be more punished is because they represent cases where the model's predictions are further from the true classes. In our case, this means that our model is to reward the true class (diagonal entries) of transitions a bit more, and punishes errors farther from the diagonal should be penalized more because they indicate more severe misclassifications. By using a cost matrix, we allow to emphasize the importance of different types of errors based on their impact in a real-world context. [38]

Prediction (row) Actual (Col)	CRC [G]	CRC[EW]	CRC[FD]
CRC [G]	€ 2.00	€ -2.00	€ -4.00
CRC[EW]	€ -2.00	€ 1.00	€ -2.00
CRC[FD]	€ -6.00	€ -4.00	€ 3.00

TABLE 6.11: This table displays a fictionalized cost matrix for punishing and rewarding the predictions made.

Table 6.11 is a simple version of a fictionalized cost matrix that punishes false predictions and rewards true predictions. In practise the costs are much bigger, but for now the ratio is important. We see that we punished predicting financial distress when this is not the case less than predicting no financial distress when it actually is the case. Now we simply multiply the matrices from both tables and then take the sum of the multiplied matrix, this leads to a positive (saved) cost value of + **€2143** . This indicates that by implementing the predictions of the ensemble model, the extended EWS is able use of to save costs by €2143 (given the fictionalized confusion matrix). Of course, this method is

quite a simple implementation to the associated costs in the real world, as different client also have i.e. different EAD, additionally many other metrics and factors have to be implemented to get to a more accurate cost output. Nevertheless, this approach represents an initial phase in demonstrating the efficacy of the prediction model within the context of the 3-class problem, showcasing its ability to make accurate class predictions and suggesting the potential for cost-saving benefits.

6.5 Implementing the prediction model into SAMAS

In this section we briefly describe a proof of concept on how the prediction model can be implemented into SAMAS. We consider the ensemble method for the 3-class to be the final prediction model that is put into practise. Here, Random Forest is prioritized in its votes above XGB, SVM and MLR. The confusion matrix of this ensemble method can be found in table 6.10. In the previous section we also punished the missed classification, however this does not necessarily has to be the case. In example, if we solely regard the implementation of the model, the precision of the changing classes forms an important metric to measure. This is especially the case since all starting clients are by definition of the scope regarded as CRC [EW], hence the recall is less important. This is intuitive as without a model, no prediction is deployed anyhow.

Let's deploy this theory on our confusion matrix. We calculate the precision of the CRC [FD] class to be $\frac{343}{369} = 0,93$, likewise the CRC [G] gives us 0,77. If we would follow this intuitive thinking, the model is highly precise in predicting [FD] and fairly in CRC [G]. That it did not recognize all actual classes and thus missed - by means of recall - predicting 386 CRC [FD] and 548 CRC [G] is not as important as they would be missed without the model anyhow. The question thus remains is it useful to blindly follow the precision metrics when a non CRC [EW] prediction is made. The answer upon this question is not straightforward and requires additional testing. But a proof of concept can already be made that implies how we can implement this forward looking approach.

Client_ID	Predictors	i.e. sort EAD	Actual	Prediction	Prob [G]	Prob [EW]	Prob [FD]
Client111	...	15000	CRC [FD]	CRC[FD]	0.012	0.11	0.878
Client222	...	14000	CRC[G]	CRC[G]	0.582	0.362	0.056
Client333	...	13000	CRC[G]	CRC[G]	0.582	0.362	0.056
Client444	...	12000	CRC[EW]	CRC[EW]	0.012	0.758	0.23
Client555	...	11000	CRC[FD]	CRC[EW]	0.002	0.856	0.141

TABLE 6.12: This figure illustrates an example of how the prediction model can be implemented into practise. The predictors shown are removed or fictionalized but the outcome variable and prediction are truly observed.

Let us suggest that a portfolio holder monitors the credit risk of a set of clients. By using the prediction model as an extension to its current monitoring system, a forward looking approach is implemented that checks if transitions from the CRC [EW] class is most likely to occur (the actual class is of course not available in real-time). Given its preciseness across [FD] and [G], the portfolio holder can determine if a notified transitions requires additional inspection. It can use the prediction model as an automatic trigger that simply follow from the majority voted class or to provide additional insight into the models estimated class probabilities during a qualitative assessment. Now, this is only a proof of concept, the implementation can be developed to more technical extent i.e. creating a dashboard. Note that additional steps might come at a cost of the prediction

models interpretibility.

Furthermore, we enhance the interpretibility of the model by delving into each of the predictors contribution in determining the class. To do so, we inspect into the primary algorithm of the ensemble method: the Random Forest model. Random Forest allows to sort the most significant predictors that contributed to the separability of the three classes. The technical process operates by selecting the predictor that maximizes the reduction in Mean Gini impurity (mean Decrease Gini). Predictors with a higher Mean Decrease Gini values are more likely to be selected for splitting because the greater impurity reduction indicates more predictive power. This method is already used in the Recursive Feature Elimination technique to reduce dimensionality, for a technical overview we refer to chapter 4.2.1 and the paper on RFE-RF from Darst[16]. We repeat this algorithm to asses the predictive power of each predictor within the eliminated subset. In figure 6.2 below we present a decreasing order of the Mean Decrease Gini together with the percentage per predictor of the total metric. The significance of these predictors are compared with the hypotheses on their separability in section 7.1.

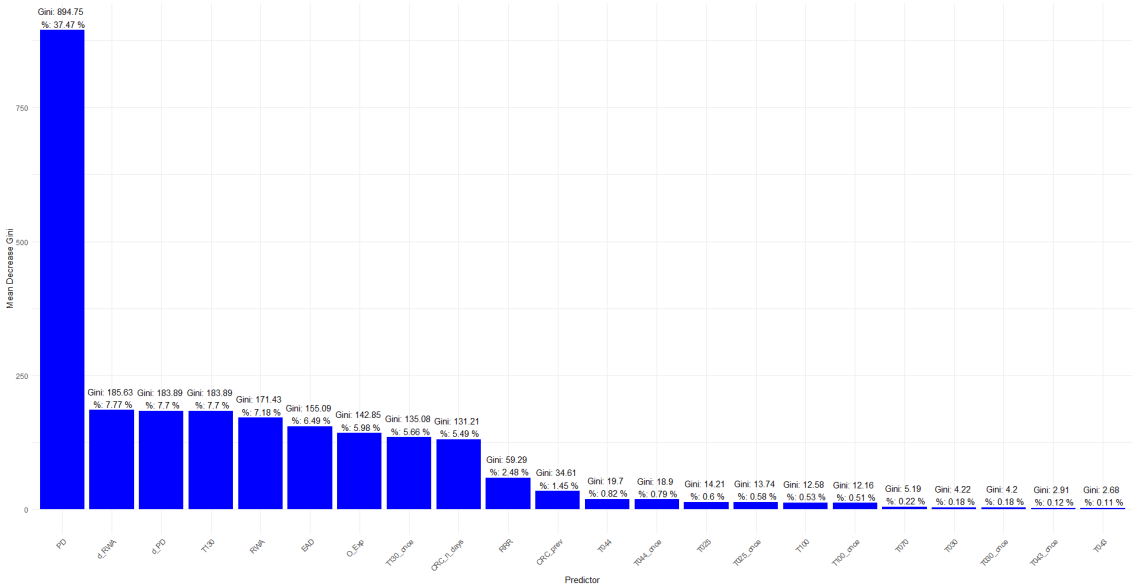

FIGURE 6.2: This figure shows the decreasing order of the Mean Decrease Gini together with the percentage per predictor of the total metric. Indicating pure distinction of the PD metric, followed by d_RWA and d_PD and a minor significance from most Triggers apart from T130.

In this chapter we discussed the empirical results. Consequently, we assessed the best performing model based on two accurateness perspectives. We concluded that the CRC [D] class is not able to be predicted under the current methods. Therefore the problem is further investigated as 3-class problem. The models are optimized by means of an ensemble voting method between RF, XGB, SVM and MLR and is generalized across different time intervals and put into practise. We end by proposing a proof of concept on how the prediction model can be implemented and interpretted into the SAMAS application. The topics reflected upon the research questions 4b amd 5a.

Chapter 7

Discussion & Conclusion

In this chapter we delve into the discussion and conclusion. We conduct the chapter constructively by answering and reflecting on the research question and objectives. Furthermore, we summarize the limitations and elaborate on the practical and theoretical insights. We end with recommendations, limitations and prospects for further research.

7.1 Reflecting on research objectives

We recall the primary objective: To accurately predict the CRC class of RSME (LBBB) clients three months into the future based on the clients data. We conclude that a model is developed that is able to significantly improve naive predictions. However, accurateness should be measured from two perspectives to form a definite answer if an adequate prediction model is developed.

First we consider that each class has to be predicted - given the balanced metrics - decently well, this fails to be acheived for the 4-class model. This is because none of the models is able to predict the CRC [D] class, hence the bad balanced performance metrics. Efforts haven been put in to enhance the predictability of CRC [D], including under- and oversampling techniques, but effects were negligible. Reviewing the problem from the models pure accurateness - given the weighted metrics - the 4-class MLR model shows decent accurateness with a F-1 score of 0.7767. However, in practice this model would become quite cumbersome as it of no use to implement a model of which one class is unpredictable. Thus, the methods were reran but than as 3-class CRC problem, here CRC [D] is rewritten to CRC[FD]. Ultimately the 4 best performing models - RF, XGB, SVM, MLR - were assembled into one prediction model. This one was able to provide a robust model that indicated decent metrics across both the balanced and weighted metrics. This model is later reran on a test set from months of an independent time interval and remained to show decent performance metrics. Testing the viability into practise using a cost matrix, indicated that by implementing the prediction model, costs can be saved. Therefore, we conclude that we are able to accurately predict the CRC class of clients three months into the future, but only by mitigating the CRC [D].

Furthermore, reflecting upon the secondary objectives, the latter paragraph indicates that the model successfully regards the models forward looking approach and timeliness objective. Also, the practical viability is proven with independent new test data and uses the same confusion matrix to asses that its implementation is able to save costs. Furthermore, the model is able to indicate the predictors that are of significant importance on determining the CRC transitions, which is specifically requested by the problem owner Rabobank. An interesting conclusion from the model separability of classes is seen when

the RFE approach showed that the more the imbalance of the CRC [D] class was mitigated the less predictors the model required. Now regarding the 3-class model as our main model, we can also reflect that the key findings in chapter 3.5 are in line with the outcome. Given table 5.2 a first subset is made of important predictors. Within this subset we repeat the RFE-RF approach and find that PD is indeed of significant importance to the model, while also the expected correlated metrics as O_Exp, EAD and RWA are of expected significance. The lagged feature between the delta d_PD and d_EAD indicate decent significance too. Furthermore, from the exploratory analyses the conclusion that CRC_n_days, CRC_Prev is of some importance is correct and likewise that Sector is not. Lastly, the triggers T100 and T130 were also correctly expected of significant importance, while the other triggers did not posses as much predictive power. To conclude, this thesis contributes to further include machine learning methods into the Rabobank's credit monitoring system. Although further research is required for definite use, its considered an innovative contribution to the EWS and presents a proof of concept on its implementation.

7.2 Theoretical Contributions

This section delves into the significance and impact of theoretical contributions of the research. The contributions are outlined below. Note that the contributions are based upon the 3-class problem where the CRC[D] class is rewritten to the CRC [FD] class.

Imbalanced Multi-class classification Problems: The first noteworthy theoretical contribution is to multi-class classification as whole, with especially addressing the challenge of severe class imbalance. Specifically, it emphasizes the limitations of conventional algorithms in multi-class scenarios with severe imbalance. We find that by mitigating the most imbalanced class , substantial performance improvements are seen. Here RF, XGB, SVM and MLR are considered the best performing algorithms. This study also highlights the limited existing research dedicated to multi-class imbalance, indicating the necessity of further exploration on multi-class classification problem compared to binary classification problems. Furthermore, it provides anti-intuitive insights by showing that commonly used under and oversampling techniques like SCOPE and SCUT do not consistently enhance multi-class classification models. Also, we found that given a multi-class confusion matrix, the avg. weighted recall when weighted to the proportion is equal to the accuracy. We may question the relevance of the metric as proposed by Grandini [21]
.

Reevaluation of Feature Importance in credit risk: The thesis challenges existing research by demonstrating that triggers in traditional credit risk models, allow to contribute to credit risk assessment. Related work showed that the concept of triggers is not always included, let alone as a forward looking model. One research that included triggers, ended up not including them into the model as they did not provide as much significance. We show to contradict this action. Of course, this is highly dependent of how triggers are defined within the problems context but it highlights the need for a more nuanced understanding of feature importance in credit risk management for financial institutions.

Literature on Early Warning Systems: The core theoretical contribution of this thesis is the development of a prediction model that classifies clients' prospective risk. While traditional early warning systems primarily focus on identifying and alerting about existing or imminent risks, our model goes a step further by forecasting future risks as-

sociated with clients. This prospective risk classification enhances the proactive nature of early warning systems, enabling preemptive actions - such as forbearance measures - to mitigate potential problematic clients and by providing more timeliness to the Early Warning System.

Interpretability and Explainability: The thesis explores methods to make machine learning models more practicable and interpretive, Giving the algorithms a context to work on and evaluating its performance in the credit risk context. Likewise, it contributes to the exploitation of Machine Learning methods in the Credit Risk Department at Rabobank. Enhancing mutual interest of a rather new and theoretical tool in the practical field of credit risk.

7.3 Practical contributions Recommendations

In this section we reflect on the practical contributions and recommendations concluded from the thesis.
Again note that the contributions are based upon the 3-class problem where the CRC[D] class is rewritten to the CRC [FD] class. This is because within the target window of 3 months, the models are not able to grasp any consistent pattern of this class. Over-sampling the class did not contribute to any improvement, as the algorithm were still not able to predict any of the classes right (number of prediction attempts also remained low). Indicating that the class is just very unpredictable in the current target window. It can be such that the data becomes more indicative to separate CRC [D] clients in smaller target windows as indication are rather abrupt and close to the actual transitions, however this is only speculation and out of scope for this research. One key limitation that thus remains is on the CRC[D] class their unpredictability, which is indeed quite logical as the financial institutions are ever seeking for the secret formula to foresee its defaulting clients, while it will possibly never find it.

Forward looking approach: Now as a 3-class model, the ensemble model is significantly able to contribute to credit risk early warning system. It is able to show that not only triggers are of influence on (future) CRC transitions, but also extra data known to the client should be taken into account. As such, it provides an forward looking model, that distinct itself from the real-time transitions on which most of the internal documents are based. The model has a good f-1 score able to grasps both a good precision and recall. Here considering the most important metric for practical use is the precision. Which is scoring especially high the merged confusion matrix across multiple target windows. Indicating that if predictions are made, they are most of the time correct. Whereas a lower recall states that missed some predictions, which without a prediction model would have been missed anyhow. Following this line of thought, it does not necessarily matter if the model missed a decent bunch of predictions, but as soon as it did predict, the predictions are likely to be confident and precise. Therefore, this increased precision allows for more accurate identification and classification of credit risk classes among clients, leading to better-informed credit monitoring decisions and forbearance measures.

Interactive evaluation and cost Efficiency: The algorithm's practical contribution includes multiple evaluation methods and analyses its cost efficiency. In this research pre-determined weights (by class proportion) are used as well as cost matrix. These models remain viable on its standalone use and the interpretation can be altered to the problems

owner preferences. In the current settings, we show that Rabobank can achieve substantial cost savings while maintaining the quality of credit risk evaluation when the prediction model is implemented.

Consistency Across Time: Even though huge additional steps can be performed on this aspect. Our model already offers the practical advantage of being relatively consistent across time. Metrics decrease in new unseen data but only to little extent. This allows to consistency facilitate long-term planning and decision-making, as it provides a stable foundation for assessing credit risk over extended periods. However, it should be performed more in depth to provide a more robust conclusion on to this. Such that it can be said that even in the face of evolving cyclic and non-cylic economic conditions, the model remains accurate and precise.

Improved Risk Mitigation Strategies: The algorithm's precision empowers Rabobank to develop more effective risk mitigation strategies. By accurately identifying clients with varying class credit risk, the bank can tailor risk management approaches to specific client profiles, reducing potential financial losses. While we cannot guarantee absolute accuracy in credit risk assessment due to the inherent uncertainties in financial markets, our model's is quite adaptable to changing conditions and thus a practical contribution. It equips Rabobank with the tools to respond dynamically to evolving risk factors and market dynamics.

Future-Proofing with Window Forward Cross-Methods: The recommendation to develop a more representative model using window forward cross-methods is a practical contribution aimed at enhancing the algorithm's robustness over time. By incorporating forward-looking data and retraining the model iteratively, Rabobank can proactively address emerging risks and opportunities.

In summary, our machine learning algorithm contributes practically by offering a prediction model with enhanced precision, cost efficiency, and shows consistency across other timer intervals. While it cannot guarantee absolute accuracy and precision, it is fairly decent, hence the balanced en weighted F-1 score of resp 0.70051 & 0.7429. That being said, we provided a first step to show that machine learning methods has the potential to be used into practise for more efficient credit risk monitoring at Rabobank. More exploitation on this topic has to power to enable better risk mitigation strategies and consistently adapts to evolving conditions. The recommendation for window forward cross-methods further bolsters the algorithm's representatives and long-term viability, ensuring Rabobank remains well-equipped to navigate the complex landscape of credit risk management and the application of Early Warning Systems.

7.4 Limitations & Further Research

There are some limitations to this research which can be overcome in further research. Regarding the 4-class problem, the CRC[D] class made us miss out on the main primary research objective to predict each class about equally well. To potentially succeed in the research as a 4-class problem, we should improve the predictability of this minority class by focusing on a different or wider scope. The three factors that can be altered are the asset class of the clients - here RSME - , the business unit - here LBBB - and the target window - here 3 months. Especially evaluating on a larger target window is expected to account

for more randomness in the clients behaviour and thus likely that more transitions occur, including defaults. A method to achieve this can be done with the use of window forward cross methods. With this method, the algorithms become more generalized across time, taking into accounts lagged features to have a robust trained model across i,e. an entire year. This would provide the current model with additional time series expansion, of which we were currently in limited reach of. This is due to a rather recent gathering of data in the SAMAS application evoking data issues while and inability to match clients across a multitude months as they get lost the more lagged features are added. Furthermore, in this research we assumed that during training there is no ordinal nature of the outcome variable, as classification from the dataset might indicate inaccurate financial distress in some of its cases. Therefore, we trained the model using multinomial regression, ideally we should also have trained an ordinal logistic regression model, assuming ordinal nature of the outcome variable. Regardless, the conclusion is that prediction models using machine learning are able to accurately predict if transitions occur from the CRC [EW] 3 months into the future. The model is able to generalize across other time intervals with a balanced f-1 score of 0.7005 and shows potential for practical implementation with a confident balanced precision of 0.82125.

References

[1] Shaza M Abd Elrahman and Ajith Abraham. "A Review of Class Imbalance Problem". In: *Journal of Network and Innovative Computing* 1 (2013), pp. 332–340. URL: www.mirlabs.net/jnic/index.html.

[2] Astha Agrawal, Herna L Viktor, and Eric Paquet. "SCUT: Multi-class imbalanced data classification using SMOTE and cluster-based undersampling". In: *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*. Vol. 01. 2015, pp. 226–234.

[3] Khaled Akkad, Hassan Mehboob, and Rakan Alyamani. "A Machine-Learning-Based Approach for Predicting Mechanical Performance of Semi-Porous Hip Stems". In: (2023).

[4] Soner Akkoç. "An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data". In: *European Journal of Operational Research* 222.1 (2012), pp. 168–178. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2012.04.009. URL: https://www.sciencedirect.com/science/article/pii/S0377221712002858.

[5] Kayode Omotosho Alabi et al. "Credit Risk Prediction in Commercial Bank Using Chi-Square with SVM-RBF". In: *Communications in Computer and Information Science* 1350 (2021), pp. 158–169. DOI: 10.1007/978-3-030-69143-1_13. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102777115&doi=10.1007%2F978-3-030-69143-1_13&partnerID=40&md5=3bebd1581236cfa75fda57b855dbef64.

[6] Bernhard Babel et al. "First-mover matters". In: *McKinsey Working Papers on Risk* 37 (2012), pp. 1–16.

[7] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization." In: *Journal of machine learning research* 13.2 (2012).

[8] Siddharth Bhatore, Lalit Mohan, and Y Raghu Reddy. "Machine learning techniques for credit risk evaluation: a systematic literature review". In: *Journal of Banking and Financial Technology* 4.1 (2020), pp. 111–138. ISSN: 2524-7964. DOI: 10.1007/s42786-020-00020-3. URL: https://doi.org/10.1007/s42786-020-00020-3.

[9] BIS. *Core principles for effective banking supervision*. Tech. rep. 2023. URL: https://www.bis.org/bcbs/publ/d551.pdf.

[10] Kay Henning Brodersen et al. "The Balanced Accuracy and Its Posterior Distribution". In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 3121–3124. DOI: 10.1109/ICPR.2010.764.

[11] Jie Cai et al. “Feature selection in machine learning: A new perspective”. In: *Neurocomputing* 300 (2018), pp. 70–79. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.11.077. URL: https://www.sciencedirect.com/science/article/pii/S0925231218302911.

[12] Xi Hang Cao, Ivan Stojkovic, and Zoran Obradovic. “A robust data scaling algorithm to improve classification accuracies in biomedical data.” eng. In: *BMC bioinformatics* 17.1 (Sept. 2016), p. 359. ISSN: 1471-2105 (Electronic). DOI: 10.1186/s12859-016-1236-x.

[13] David Carruthers. “Default Rate Forecast 2023/24: US Speculative Grade Borrowers and US Leveraged Loansle”. PhD thesis. 2023. URL: https://www.creditbenchmark.com/default-rate-forecast-2023-24-us-speculative-grade-borrowers-and-us-leveraged-loans/.

[14] Daniel Tianfu Chen. “Development of Financial Distress Prediction Model for the Watchlist Classification of Wholesale Banking Clients at ING by”. PhD thesis. University of Twente, 2023. URL: http://essay.utwente.nl/95110/1/ING_Thesis_Daniel_Chen_Final_Version.pdf.

[15] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *CoRR* abs/1603.0 (2016). URL: http://arxiv.org/abs/1603.02754.

[16] Burcu F Darst, Kristen C Malecki, and Corinne D Engelman. “Using recursive feature elimination in random forest to account for correlated variables in high dimensional data”. In: *BMC Genetics* 19.1 (2018), p. 65. ISSN: 1471-2156. DOI: 10.1186/s12863-018-0633-8. URL: https://doi.org/10.1186/s12863-018-0633-8.

[17] Xiaojian Ding et al. “Random radial basis function kernel-based support vector machine”. In: *Journal of the Franklin Institute* 358.18 (2021), pp. 10121–10140. ISSN: 0016-0032. DOI: https://doi.org/10.1016/j.jfranklin.2021.10.005. URL: https://www.sciencedirect.com/science/article/pii/S0016003221006025.

[18] Merryta Djakaria and Tuga Mauritsius. “Artificial intelligence model as an early warning system for fraudulent transactions in mobile banking”. In: *ICIC Express Letters, Part B: Applications* 14.7 (2023), pp. 747–753. ISSN: 21852766. DOI: 10.24507/icicelb.14.07.747.

[19] The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics. “Fisher, R.A.” In: 7 (1936), pp. 179–188.

[20] David Evans, Paul Gruba, and Justin Zobel. *How to Write a Better Thesis*. April. 2014. ISBN: 9783319042855. DOI: 10.1007/978-3-319-04286-2.

[21] Margherita Grandini, Enrico Bagli, and Giorgio Visani. “Metrics for Multi-Class Classification: an Overview”. In: (2020), pp. 1–17. arXiv: 2008.05756. URL: http://arxiv.org/abs/2008.05756.

[22] Pedro Guerra, Mauro Castelli, and Nadine Côrte-Real. “Machine learning for liquidity risk modelling: A supervisory perspective”. In: *Economic Analysis and Policy* 74 (2022), pp. 175–187. ISSN: 03135926. DOI: 10.1016/j.eap.2022.02.001. URL: https://doi.org/10.1016/j.eap.2022.02.001.

[23] Jan Hauke and Tomasz Kossowski. “Comparison of values of Pearson’s and Spearman’s correlation coefficients on the same sets of data”. In: *Quaestiones geographicae* 30.2 (2011), pp. 87–93.

[24] Xiaoyi Hu and Ke Wang. "Bank Financial Innovation and Computer Information Security Management Based on Artificial Intelligence". In: *2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*. 2020, pp. 572–575. DOI: 10.1109/MLBDBI51377.2020.00120.

[25] Franz Kronthaler and Silke Zöllner. *Data Analysis with R Studio: An Easygoing Introduction*. 2020, pp. 1–125. ISBN: 9783662625187. DOI: 10.1007/978-3-662-62518-7.

[26] Vrushali Y Kullarni and Pradeep K Sinha. "Random Forest Classifier: A Survey and Future Research Directions". In: *International Journal of Advanced Computing* 36.1 (2013), pp. 1144–1156.

[27] Xianglong Liu. "Towards Better Banking Crisis Prediction: Could an Automatic Variable Selection Process Improve the Performance?*". In: *Economic Record* 99.325 (2023), pp. 288–312. ISSN: 14754932. DOI: 10.1111/1475-4932.12721.

[28] Juan Luis Olmo et al. "Binary and Multiclass Imbalanced Classification Using Multi-Objective Ant Programming". In: *2012 12TH INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS (ISDA)*. Ed. by A Abraham et al. International Conference on Intelligent Systems Design and Applications. Machine Intelligence Res Labs (MIR Labs); IEEE Syst, Man & Cybernet Soc, Czechoslovakia Chapter; IEEE Syst, Man & Cybernet Soc, Spanish Chapter; IEEE; Cochin Univ Sci & Technol. 2012, pp. 70–76. ISBN: 978-1-4673-5117-1; 978-1-4673-5118-8.

[29] Bernardo Maggi and Marco Guida. "Modelling non-performing loans probability in the commercial banking system: efficiency and effectiveness related to credit risk in Italy". In: *Empirical Economics* 41.2 (2011), pp. 269–291. ISSN: 1435-8921. DOI: 10.1007/s00181-010-0379-2. URL: https://doi.org/10.1007/s00181-010-0379-2.

[30] Somayeh Moradi and Farimah Mokhatab Rafiei. "A dynamic credit risk assessment model with data mining techniques: evidence from Iranian banks". In: *Financial Innovation* 5.1 (2019). ISSN: 21994730. DOI: 10.1186/s40854-019-0121-9.

[31] Ismail Olaniyi Muraina. "IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS :" in: February (2022).

[32] Basel Committee on Banking Supervision. "Basel Committee on Banking Supervision High-level summary of Basel III reforms High-level summary of Basel III reforms iii Contents". In: (2017). URL: www.bis.org.

[33] Organisation for Economic Co-operation and Development (OECD). "Artificial Intelligence, Machine Learning and Big Data in Finance". In: *OECD Paris* (2021), pp. 1–72. URL: https://www.oecd.org/finance/financial-markets/Artificial-intelligence-machine-learning-big-data-in-finance.pdf%0Ahttps://www.oecd.org/daf/fin/financial-markets/artificial-intelligence-machine-learning-big-data-in-finance.htm.

[34] Guobin Ou and Yi Lu Murphey. "Multi-class pattern classification using neural networks". In: *Pattern Recognition* 40.1 (2007), pp. 4–18. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2006.04.041. URL: https://www.sciencedirect.com/science/article/pii/S0031320306002081.

[35] Rabobank. *Basel IV implications Rabobank*. Tech. rep.

[36] *Rabobank Annual Report*. Tech. rep. Rabobank Group, 2022, pp. 61–64.

[37] Boran Sekeroglu, Shakar Sherwan Hasan, and Saman Mirza Abdullah. "Comparison of Machine Learning Algorithms for Classification Problems BT - Advances in Computer Vision". In: ed. by Kohei Arai and Supriya Kapoor. Cham: Springer International Publishing, 2020, pp. 491–499. ISBN: 978-3-030-17798-0.

[38] B. H Shekar and Guesh Dagnew. "Grid Search-Based Hyperparameter Tuning and Classification of Microarray Cancer Data". In: *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*. 2019, pp. 1–8. DOI: 10.1109/ICACCP.2019.8882943.

[39] Si Shi et al. "Machine learning-driven credit risk: a systemic review". In: *Neural Computing and Applications* 34.17 (2022), pp. 14327–14339. ISSN: 1433-3058. DOI: 10.1007/s00521-022-07472-2. URL: https://doi.org/10.1007/s00521-022-07472-2.

[40] Solidar. "Review of Machine Learning models for Credit Scoring Analysis". In: (2022), pp. 1–16. URL: doi:%2010.16925/2357-6014.2020.01.11..

[41] Erika Spuchľáková, Katarína Valašková, and Peter Adamko. "The Credit Risk and its Measurement, Hedging and Monitoring". In: *Procedia Economics and Finance* 24 (2015), pp. 675–681. ISSN: 2212-5671. DOI: https://doi.org/10.1016/S2212-5671(15)00671-1. URL: https://www.sciencedirect.com/science/article/pii/S2212567115006711.

[42] Prajwala T R. "A Comparative Study on Decision Tree and Random Forest Using R Tool". In: *IJARCCE* (Jan. 2015), pp. 196–199. DOI: 10.17148/IJARCCE.2015.4142.

[43] E R Creditcore Tribe. "Global Standard on Credit Risk Parameters". In: (2018).

[44] Petros Xanthopoulos, Panos M Pardalos, and Theodore B Trafalis. "Linear Discriminant Analysis". In: *Robust Data Mining*. New York, NY: Springer New York, 2013, pp. 27–33. ISBN: 978-1-4419-9878-1. DOI: 10.1007/978-1-4419-9878-1_4. URL: https://doi.org/10.1007/978-1-4419-9878-1_4.

[45] Lasheng Yu and Zeko Mbumwae. "Towards a Multi-Agent System (MAS) based Credit Reference Bureau". In: *Proceedings - 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009* 1 (2009), pp. 728–732. DOI: 10.1109/ICICISYS.2009.5358392.

Chapter 8

Appendices

.1 Appendix A: Determination of CRC in real-time

The determination of the CRC [Class] should be based on ability of the obligor to meet its financial commitments on a going concern basis and should not take into account any elements related to actions of the bank to claim its security position (e.g. collateral, guarantees etc.). Hence, it should be objective. The CRC can be determined at any point in time during the credit life cycle. It is based on specific trigger events which lead to either: The direct assignment of a CRC class, or in some cases further analysis to determine the appropriate CRC class. CRC transition for RSME clients are depicted in figure 1 including a table of entry and exit criteria in table 1.

CRC Good	No triggers are hit
Entry EW	>EUR 100 is more than 30 but no more than 60 days past due and one of the following applies (qualitative assessment needed); i. the arrears can be resolved without help from the bank, or; ii. the arrears can be resolved with help from the bank that fits within regular commercial underwriting criteria; iii. The amount past due relates to a forborn, but not reclassified from Default, contract and this contract. A hit of EW Triggers no. 7, 8, or 12 in the Qualitative Assessment concludes EW; A hit of EW Trigger(s) (no. 5, 6, 9, 11, 14) assigns the Obligor to the EW list; Trigger 9 (R18-R19) isn't applicable for Standardized Approach portfolios without a RRR model; A management decision; An SCE is in the Forbearance Probation period and FD is not applicable.
Exit	Contract has no Forbearance probation period anymore; EW triggers aren't hit anymore. De-activation depends on specific EW triggers; No >30 DPD in the last 3 months; RRR should be better than R18 for at least 3 consecutive months.

TABLE 1: This table contains an overview of entry and exit criteria and is taken from the internal document [43]

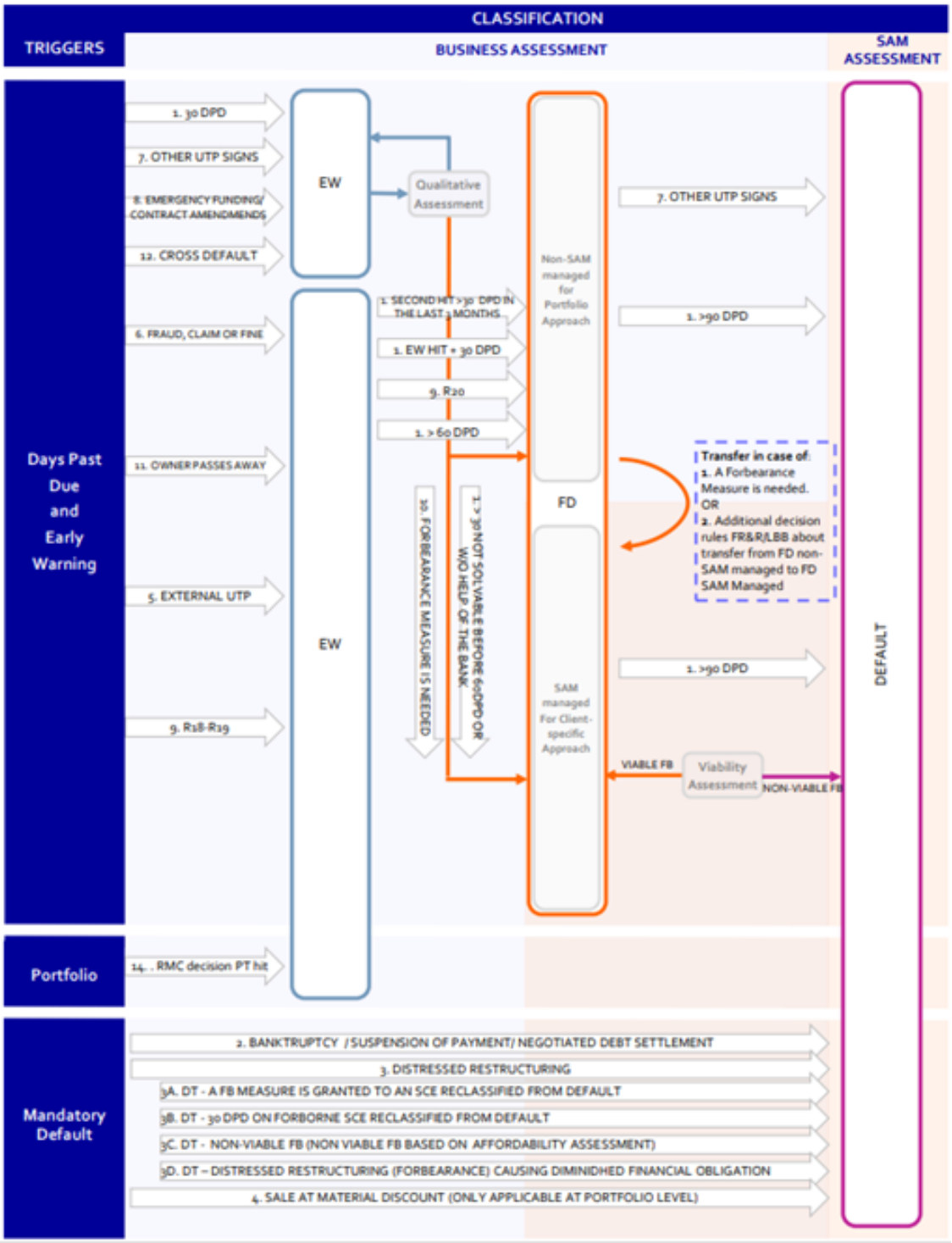

FIGURE 1: This figure shows the four CRC classes in ascending (left to right) order from severity and is taken from the internal report Global Standards on Credit Risk Parameters [43]

.2 Appendix B: SLR

Machine Learning in Credit Risk

Scopus: (TITLE-ABS-KEY (machine AND learning) AND TITLE-ABS-KEY (bank) AND TITLE-ABS-KEY (credit AND risk) OR TITLE-ABS-KEY (early AND warning))	n = 370
Language: “English”	n = 361
Keyword: “Credit Risk”	n = 53
Filter on Relevance	n = 25
Exclude topic: “Credit Card”	n = 17
Include: Only free-access or licensed literature	n = 9
Manually added University of Twene Library: Same keys as above	n = 10
SLR Research questions to answer: To what extent is machine learning already used in Credit Risk Monitoring? 1. Unsupervised ML methods applied if yes which? 2. Which ML algorithms is used for training, which for cross fold and validation? 3. What type of ML validation methods are used? 4. Is the research focused on a case study or purely theoretical? 5. How successful is the research? Is it addressed in terms of metrics? 6. Is there a mention of CRC stages and or a watchlist/early warning system? 7. Is the research focused on predicting (probability of) default or any other related CRC stage? What software is used? 8. What are the main findings and result? 9. What is the client scope (rural, wholesale, RSME etc.)?	

Source	1	2	3	4	5	6	7	8	9
(Alabi et al. 2021)	Feature selection: Chi- Square	SVM-(RBF), trained with 10-crossfold	Confusion matrix. Acc main metric	No, theoretical	Yes accuracy of 93%	No	Predicting credit scoring loan. Using matlab	The result shows that RBF and SVM-RBF accuracy was 93% shown in the predictive performance	General loans of Taiwan bank credit data set

								table. This study confines with the prediction of credit loan	
(Kalayci and Arslan 2017)	Extra features are added	RF vs LR, SVM, DT	Confusion matrix, acc main metric	No theoretical	Unknown, accuracy 82,25%	Yes	Predict NPL classification. Uknown software.	The proposed system aims to support a warning 6 months ahead to detect NPL state. RF outperforms other ML methods	SMEs turkey
(G. 2016)	Correlation variables, ranking importance using RF	DT	Confusion matrix, all metrics used.	No theoretical on public dataset	Not compared but has accuracy of 94,3%	Not directly, uses PD without any thresholds mentioned to determine a class label	Predicting PD of new loan applications of a bank. Using R	DT is used to predict the class labels of the new loan applicants, based on their Probability of Default.	Not mentioned
(T. Liu and Huang 2022)	Ranking importance using RF	LR, RF incl. hyperparameter tuning	Confusion matrix, main accuracy	Not mentioned	"Relative high accuracy"	EW is mentioned	Predict the loan viability using classification. Unknown software.	Designed a rf algorithm to make risky early warning classifiers	Credit loan businesses
(L. Wang and Zhang 2023)	Not used deliberately, as it is not always	Two stage N-model, SVM, DT, RF , MLP , LTSM.	Confusion matrix matrix and	Only used theoretically	Two staged N-model outperformed common ML	Compared with corporate credist risk	Predict EW. In order to mitigate risk of default.	This article builds a new two-stage	Unknown

	better to feature select	Uses 5 fold cross	F1-score + G-mean		algorithms with F1 87,29% and G-mean 89,47%	early warning indicators. But no mentioned of CRC like banking stages	Unknown software.	ensemble model using a variety of machine learning methods represented by deep learning for corporate credit risk early warning	
(Jiang and Wang 2022)	Unknown	NN, XGBoost, LGB, CatBoost, LR	Confusion matrix and AUC	Only used theoretically	All show accuracy >88%. Not compared to current method. NN has best AUC score.	Not mentioned	Predict classification problem and use the data from LendingClub company. Unknown software.	This paper considers the use of six kind of machine learning methods to model credit risk management problems for a classification of clients	Not mentioned
(Hegde et al. 2023)	Some features selection in Keras Library is used unknown which one	K-fold is used. DT, SVM, LR, RF, XGBoost are used.	Confusion matrix on Accuracy	Only used theoretically	Focused on accuracy. LR is best with slight margin over RF and XGBoost	Not mentioned	Prediciting the chance that loans are granted. Not a certain CRC stage.	The experimental results indicates that logistic regression model is more accurate for the	Unknown

								credit risk predicition	
(Tan and Lin 2023)	Lasso regression	XGboost with 5 cross fold	accuracy, recall, specificity, AUC, G-mean to evaluate the XGBoost model.	Used as case study on unknown company	AUC of 95,8 and G-mean 91,7%	Own early warning system without specific stages	Classification stages not specified. But prediction on some sort of classification is the idea. Python used.	Credit risk early warning method for companies based on XGBoost and SHAP.	General banking clients.
(D. T. Chen 2022)	Mutual information and many other pre-process methods	DT, LDA, LG, SVM, GBM, RF, XGB, ANN, (dummy)	Acc, Prec, F1, MCC, AUC, Recall, Correlation	Theoretical applied on ING database	Assessed on all metrics. RF does best on all metrics except accuracy.	Good, EW, Def. Almost same as Rabo. Though research focuses on negative mitigations and not specific transitions.	Tries to effectively classify WB clients at ING on a watchlist based on their prospective credit risk. Using machine learning in Python	This financial distress prediction uses machine learning based on internal triggers, external triggers, and internal client data as input to predict if a client will be in financial distress. RF scores best on chosen metrics	Wholesale at ING
(Guerra, Castelli, and Côrte-Real 2022)	RF with 85% feature importance threshold. PCA	LR, SVM, NBC, RFC, XGB. 5-fold cross for training.	The confusion matrix (Precision and recall)	Practical and theoretical implications	Assessed on all metrics. Following F1-score the most. Best on	The best performing model can be set up as a decision	This work investigates whether machine learning	The results show that extreme gradient	General

	considered but less accurate though giving model more efficiency. Correlation matrix between features and targets		and the f1-score.		XGB followed by RF.	support system, either as stand-alone stress-testing tool, or as part of an EWS. No mention of any or similar CRC stages.	techniques can successfully predict liquidity risk, thus providing insights for stress-testing scenarios. Uses python	boosting (XGBoost) outperforms other methods for this classification problem. The resulting model can be set up for a production environment and provide scenarios for stress-testing, or as an early warning system (EWS), thus supporting the overall SREP exercise.	

Early warning systems in credit risk

Scopus: (TITLE-ABS-KEY (early AND warning AND system) AND TITLE-ABS-KEY (banking) AND TITLE-ABS-KEY (credit AND risk) OR TITLE-ABS-KEY (early AND warning))	n = 37
Limited to Language: "English"	n = 33
Keyword: "Credit Risk"	n = 11
Include: Only free-access or licensed literature	n = 4
Manually added University of Twene Library: Same keys as above	n = 6
SLR Research questions to answer: To what extent are early warning systems already used in Credit Risk Monitoring? 1. What type of credit risk is monitored? 2. How is the early warning system defined 3. Are there more stages to except good and default in the EWS and if yes how many? 4. Can we find any Machine learning methods in the paper? 5. Main findings and conclusion?	

Source	1	2	33	4	5
(Kalayci and Arslan 2017)	Credit risk for RSME at bank Turkey	The EWS, is an early watchlist system that supports classifying SME customers as non-performing or performing and is targeted during lifetime of the credit	1. Non performing and performing. With one Watchlist status in between.	yes, RF algorithm is compared with different machine learning algorithms like Logistic Regression, Supp ort Vector Machine and Decision Tree	The proposed system aims to support a warning 6 months ahead to detect NPL state. RF outperforms other ML methods

(D. T. Chen 2022)	Credit risk for Wholesale and rural clients at bank (ING)	An Early Warning System (EWS) enables the effective monitoring of the credit portfolio by providing Early Warning Indicators (EWI) and triggers to alert stakeholders such as risk and account managers when there are early signs of default.	1. Non performing and performing. With one Watchlist status in between.	Yes. Both unsupervised and supervised methods are trained to find best suitable model.	Tries to effectively classify WB clients at ING on a watchlist based on their prospective credit risk. Using machine learning in Python
(Koyuncugil and Ozgulbas 2012)	Credit risk for RSME at bank Turkey	An early warning system (EWS) is a system which is using for predicting the success level, probable anomalies and is reducing crisis risk of cases, affairs transactions, systems, phenomena, firms and people. Furthermore,	No immediate case EWS with stages is considered. They consider 15 indicators and dilute 2 indicators that can be used as an EWS.	Data mining is used but no specicifc ML methods are mentioned	Writers developed a financial EWS based on financial risk by using data mining. CHAID algorithm has been used for development of the EWS. Developed EWS can be served like a tailor made financial advisor in decision making process of the firms with its automated nature to the ones who have inadequate financial background.
(Jin and Nadal De Simone 2014)	32 European banking groups	Not given	No immediate stages though combining the	Not given	This study proposes a novel framework which combines marginal probabilities of default estimated

			GDFM applied to a large macrofinancial database with a structural credit risk model not only produces an "early warning indicator", but also can help identying the economic forces driving the increase in vulnerabilities		from a structural credit risk model with the consistent information multivariate density optimization (CIMDO) methodology and the generalized dynamic factor model (GDFM) supplemented by a dynamic t-copula.
(Lin and Wu 2011)	Banks	Not given	No mention	No use GRA approach compared to machine learning. Not very helpful paper on EWS.	The results illustrate that in the prediction of financially crisis as well as financially sound banks, the proposed GRA model demonstrates better prediction accuracy than the conventional ones. The results also imply that the financial data set one year before the crisis leads to the best accuracy. It is helpful for the establishment of early warning models of financial crisis for banks
(Bakurova, Pasichnyk, and Tereschenko 2021)	Banking in Ukraine	Early warning signals provide information about the credit quality of the debtor, as well as a wide range of credit analytics and scenario analysis for companies.	No mentioned	No mentioned	The main result of the work is a light ontology based on the analysis of bank documents in the OWL language in the Protégé editor and the production system to support credit decision-making in banking institutions of Ukraine. The

.3 Appendix C: Data merging & filtering

Since our scope is pre-defined and we use data of multiple moments in time, we have to filter and merge our data based on these specifications. Because this research involves merging of multiple data frames, we slightly differentiate from the theoretical framework by [3], as some pre-processing steps are put forward in the model development process. Following the GS on CRC, data is not reported on client level but on a so-called regulatory facility level. For consistency throughout the thesis we will regard a unique regulatory facility as a unique client_ID.

The query already included a filter on the clients asset class RSME and the chosen business unit LBB, as defined in section scope. To account for data at a specific moment in time, we filter the data based on the dimension "Reporting Data Month". The last day of the concerning month is always taken as the snapshot to be reported. Three reporting months of data need to merged: April 23 (to feature engineer changes compared to May 23), May 23 (including all relevant variables from the query) and August 23 to attribute the outcome variable (CRC) at August 23 for each client present in data of May 23. Table ref indicates the filters on row basis that follow from merging and filtering.

Step	Description	Resulting Rows
1	Merge aug_23 CRC variable to may_23	n (confidential)
2	Remove clients containing F4LF in client_ID	(-1265)
3	Unfound match based on client_ID	(-2067)
4	Merge apr_23 numerical variables to new may_23	(-419)
5	Filter out unspecified previous CRC values	(-2729)
6	Filter out missing values for PD and LGD	(-171)
7	Merge rows specifying triggers based on Client_ID	(-1162)

TABLE 2: Summary of Data Manipulation Steps

1. The core data frame, resulting from a query on may_23 with filters on RSME and LBB, initially contains n rows.

2. Combining dimensions into one query resulted in an incorrect extraction due to the CRC being determined at the client level while clients can have subsets of contracts (single credit exposures). Rows with "F4FL" in their names are removed, resulting in the removal of 1265 rows from the data frame.

3. Merging the may_23 data frame with the CRC outcome variable from August 23 necessitates grouping both data frames by client_ID, resulting in the loss of 2067 rows due to mismatches in client_ID. This could occur if client contracts with Rabobank were terminated between the reporting dates.

4. A similar process to step 3 is executed to merge the updated core data frame with numerical variables from apr_23. This inclusion allows for the potential feature engineering of differences between these variables monthly.

5. Unspecified CRC or missing data are not imputed but instead removed. This approach prevents potential bias introduced by imputation, allowing the data to be treated as true observations.

6. Similarly, a few missing values for PD and LGD are removed without imputation.

7. Within SAMAS, each trigger reported adds a separate row, potentially resulting in multiple identical rows differing only in the trigger code variable. To merge all client_IDs successfully, one-hot encoding of the trigger ID variable is required. After encoding, rows based on Client_IDs can be merged, summing binary triggers while retaining identical variables.

.4 Appendix D: Visuals from predictors in Data Review chapter

Removed for confidentiality reasons

.5 Appendix E: Mathematical intuition behind proportionalized weights and accuracy

This appendix shows that accuracy equals (average) weighted accuracy when the class proportion is equal to the weight each iteration.

$$\text{Accuracy} = \sum \frac{TP}{TP+FP+FN+TN}$$

Revisiting weighted accuracy:

$$\text{Weighted accuracy} = \sum \frac{TP \times \text{propdata}}{\text{Row sums of confusion matrix}}$$

If the dataset is weighted such that the proportions in 'propdata' match the actual distribution of classes in the confusion matrix:

$$\text{Weighted accuracy} = \sum \frac{TP \times \text{actual\_distribution}}{\text{Row sums of confusion matrix}}$$

Where the actual distribution (hence the 'propdata') can be obtained from the confusion matrix due to the test set being stratified. Each iteration represents the actual proportion of the class:

$$\begin{aligned}
\text{Actual distribution} &= \left[\frac{TP+FN}{TP+FP+FN+TN}\right] \\
\text{Weighted accuracy} &= \sum \frac{TP \times \left[\frac{TP+FN}{TP+FP+FN+TN}\right]}{TP+FN} \\
\text{Weighted accuracy} &= \sum \frac{TP}{TP+FP+FN+TN}
\end{aligned}$$

Therefore, the weighted accuracy in this case would be equal to the accuracy when the weights are proportionolized to the class distribution.