Exploring the accuracy-explainability trade-off on credit scoring classifiers

by

Sibusiso Mtiyane

Dissertation submitted in accordance with the requirements for the degree of

Master of Science

in the subject

Operations Research

at the

UNIVERSITY OF SOUTH AFRICA

Department of Decision Sciences
College of Economic and Management Sciences

Supervisor: Prof. KM Malan
Co-supervisor: Prof. MD Jankowitz

February 2024

Open Rubric

Declaration of Authorship

Name	**Sibusiso Mtiyane**
Student Number	**36777900**
Course	**DFOPR93**

I declare that the research project/dissertation/thesis "Exploring the accuracy-explainability trade-off on credit scoring classifiers" is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I have not previously submitted this work, or part of it, for examination at UNISA for another qualification or at any other higher education institution.

February 2024

Signature **Date**

Abstract

Recent research has highlighted the significance of accuracy and explainability of classification models applied across various disciplines. A wide range of classification models and combinations of models have been extensively studied to determine those with superior performance. These studies demonstrate that models that tend to be more accurate are also difficult to understand; there appears to be a trade-off between accuracy and explainability. Consequently, this has led to an increased focus on explainable artificial intelligence, a field of research concerned with explaining model predictions.

Although explainable artificial intelligence is an area of research with growing popularity in the science community, there are still limited case studies that explore its applications in credit default risk. Credit default risk refers to the potential financial loss or risk that is incurred by a credit provider when an obligor fails to meet their debt obligations. To quantify, mitigate and manage the risk associated with granting credit proactively, credit providers utilise scoring classifiers to assess the risk of credit applicants prior to granting credit. Furthermore, credit risk providers are legally required to explain predictions of scoring classifiers.

Popular classifiers used in credit risk include logistic regression, discriminant analysis, decision trees, random forests, bootstrap aggregation, neural networks, support vector machines and gradient boosting algorithms. Logistic regression and discriminant analysis are widely adopted in the financial industry because they perform reasonably well and are inherently interpretable. However, these approaches are giving way to alternative approaches that offer improved accuracy in risk assessment, even though these alternatives lack interpretability; they are less comprehensible and are often regarded as black boxes. This lack of interpretability has resulted in a reluctance to adopt these alternative techniques in credit granting.

The aim of this study is to remove the aforementioned barrier of using black box models by utilising explainable artificial intelligence methods, such as Shapley additive explanations and local interpretable model-agnostic explanations. The study also examines the accuracy-explainability trade-off of different classifiers by developing and evaluating eight classification models on two publicly available credit datasets.

Eight classification models were constructed, including decision trees, logistic regression, linear discriminant analysis, support vector machines, artificial neural networks, bootstrap aggregation, random forest, and light gradient boosting classifier. Their performance and interpretability were assessed after training and tuning the hyperparameters for optimal comparison on training, testing and validation subsets of the data. Performance accuracy was measured using the area under the curve on 30 random subsets generated from the validation data. Furthermore, the Kruskal Wallis test and Dunn's multi-comparison test were used to rank the predictive models by accuracy and to determine if the differences in mean accuracy are statistically significant. The interpretability of these classifiers was conducted for both transparent and black box models. To achieve these ends, key preprocessing steps were developed to reduce the complexities of local and global model interpretation. In addition, Shapley additive explanations and local interpretable model-agnostic explanations were utilised to analyse the relative importance of features and the impact on predictions.

The experiments show that the artificial neural network, ensembles and other tree-based algorithms significantly outperform logistic regression and linear discriminant analysis in the first case study. However, contradictory results are obtained for the second case study, as the performance of the classifiers are relatively comparable. This indicates that model performance depends on the data from which the models are constructed. These two case studies show that the perceived trade-off between accuracy and explainability does not always hold true. Furthermore, Shapley additive explanations yielded results that are consistent with the intrinsic interpretability results of the transparent methods. This post-hoc interpretability enables us to understand how the predictions are made and what factors contributed to the prediction. This is important to create a reliable and trustworthy framework that uses black box models for credit decisions.

The research highlights the benefits of using alternative methods for credit risk scoring, showing that the performance can vary significantly. It also demonstrates the effectiveness of Shapley additive explanations and local interpretable model-agnostic explanations to explain predictions of black box classifiers. However, it identifies challenges in using the Shapley additive explanations. The mean absolute value may be sensitive to outliers, which could have an impact on feature importance. Therefore, further work is required to enhance the efficiency of calculating Shapley additive explanations' values for linear classifiers and some ensembles.

Opsomming

Onlangse navorsing het die belangrikheid uitgelig van die akkuraatheid en verduidelikbaarheid van klassifikasiemodelle wat dwarsoor verskeie dissiplines toegepas word. 'n Wye reeks klassifikasiemodelle en modelkombinasies is omvattend bestudeer om daardie modelle met voortreflike prestasie te bepaal. Hierdie studies het gedemonstreer dat modelle wat neig om meer akkuraat te wees, ook moeilik is om te verstaan; dit kom voor of daar 'n kompromie is tussen akkuraatheid en verduidelikbaarheid. Dit het gevolglik aanleiding gegee tot 'n verhoogde fokus op verduidelikbare kunsmatige intelligensie, 'n navorsingsveld wat met die verduideliking van modelvoorspellings gemoeid is.

Alhoewel verduidelikbare kunsmatige intelligensie 'n navorsingsgebied is wat besig is om in gewildheid toe te neem binne die wetenskapgemeenskap, is daar steeds beperkte gevallestudies wat die toepassing daarvan op kredietwanbetalingsrisiko ondersoek. Kredietwanbetalingsrisiko verwys na die potensiële finansiële verlies of risiko waaraan 'n kredietverskaffer blootgestel word wanneer 'n skuldenaar in gebreke bly om hul skuldverpligtinge na te kom. Ten einde die risiko wat met kredietverskaffing geassosieer word proaktief te kwantifiseer, versag en bestuur, moet kredietverskaffers kredietgraderingsklassifiseerders gebruik om die moontlike risiko te evalueer wat kredietaansoekers inhou, voordat krediet toegestaan word. Voorts is kredietrisikoverskaffers volgens wet verplig om die voorspellings van kredietgraderingsklasifiseerders te verduidelik.

Gewilde klassifiseerders wat in kredietrisiko gebruik word, sluit logistieke regressie, diskriminantanalise, besluitnemingsbome, ewekansige woude, skoenlussamevoeging, neurale netwerke, ondersteuningsvektormasjiene en gradiëntversterkingsalgoritmes in. Logistieke regressie en diskriminantanalise is algemeen deur die finansiële bedryf aanvaar aangesien hulle redelik goed presteer en inherent verduidelikbaar is. Hierdie benaderings skep egter ruimte vir alternatiewe benaderings wat verbeterde akku-

raatheid ten opsigte van risiko-assessering bied selfs al gaan hierdie alternatiewe benaderings mank aan interpreteerbaarheid; hulle is nie so verstaanbaar nie en word dikwels as swartkissies (black boxes) gesien. Hierdie gebrek aan interpreteerbaarheid het tot gevolg dat daar 'n traagheid is om hierdie alternatiewe kredietverleningstegnieke aan te neem.

Hierdie studie het ten doel om die voorafgenoemde versperring tot die gebruik van swartkissiemodelle te verwyder deur verduidelikbare kunsmatige intelligensiemetodes soos Shapely se additiewe verduidelikings en plaaslike interpreteerbare model-agnostiese verklarings te gebruik. Die studie ondersoek ook die akkuraatheid-verduidelikbaarheidskompromie van verskillende klassifiseerders deur agt klassifikasiemodelle vir twee openbaar beskikbare kredietdatastelle te ontwikkel en te evalueer.

Agt klassifikasiemodelle is saamgestel, naamlik besluitnemingsbome, logistieke regressie, linière diskriminantanalise, ondersteuningsvektormasjiene, kunsmatige neurale netwerke, skoenlussamevoeging, ewekansige woud en ligte gradiëntversterkingsklassifiseerder. Hul prestasie en interpreteerbaarheid is geassesseer na opleiding en instelling van die hiperparameters vir optimale vergelyking van opleiding, toetsing en geldigverklaring van deelversamelings van die data. Prestasie-akkuraatheid is gemeet deur van die area onder die kurwe van 30 ewekansige deelversamelings wat uit die geldigverklaarde data gegenereer is, gebruik te maak. Voorts is daar van die Kruskal Wallis-toets en Dunn se multivergelykingstoets gebruik gemaak om die voorspellingsmodelle ten opsigte van akkuraatheid te klassifiseer en te bepaal of die verskille in gemidddelde akkuraatheid statisties beduidend is. Die interpreteerbaarheid van hierdie klassifiseerders is vir beide deursigtige en swartkassiemodelle uitgevoer. Om hierdie resultate te verkry, is belangrike voorverwerkingstappe ontwikkel om die kompleksiteite van plaaslike sowel as globale modelinterpretasie te verminder. Daarbenewens is Shapley se additiewe verduidelikings en plaaslike interpreteerbare model-agnostiese verduidelikings ook ingespan om die relatiewe belangrikheid van kenmerke en die impak op voorspellings te ontleed.

Die eksperimente toon dat die kunsmatige neurale netwerk, ensembles en ander boomgebaseerde algoritmes in die eerste gevallestudie beduidend beter as die logistieke regressie en linière diskriminantanalise presteer het. Die tweede gevallestudie het egter teenstrydige resultate opgelewer. In die tweede gevallestudie is die prestasie van die klassifiseerders relatief vergelykbaar. Dit is 'n aanduiding dat modelprestasie afhanklik is van die data waaruit die modelle saamgestel is. Hierdie twee gevallestudies toon dat die waargenome kompromie tussen akkuraatheid en verduidelikbaarheid nie altyd waar is nie. Boonop het die Shapley additiewe verduidelikings resultate opgelewer wat met die intrinsieke interpreteerbaarheidsresultate van die deursigtige metodes ooreenstem. Hierdie post-hoc interpreteerbaarheid help ons om te verstaan hoe die voorspellings gemaak word en watter faktore tot die voorspellings bygedra het. Laasgenoemde is belangrik ten einde 'n betroubare en geloofwaardige raamwerk te skep wat van swartkassiemodelle vir kredietbesluite gebruik maak.

Die navorsing beklemtoon die voordele van die gebruik van alternatiewe metodes vir kredietrisikogradering; dit toon dat die prestasie aansienlik kan varieer. Dit demonstreer ook die doeltreffendheid van die Shapley additiewe verduidelikings en plaaslike interpreteerbare model-agnostiese verduidelikings in die verduideliking van voorspellings van swartkissieklassifiseerders. Dit is egter so dat dit uitdagings ten opsigte van die Shapley additiewe verduidelikings identifiseer. Die gemiddelde absolute waarde mag dalk sensitief wees vir uitskieters wat 'n impak op die belangrikheid van kenmerke kan hê. Daarom is verdere werk nodig om die doeltreffendheid van die berekening van Shapley se additiewe verduidelikings se waardes vir liniêre klassifiseerders en sommige ensembles te versterk.

Kgutsufatso

Diphuputso tsa morao tjena di totobaditse bohlokwa ba ho nepahala le ho hlaloswa ha mefuta ya dihlopha e sebediswang dikarolong tse fapaneng. Mefuta e mengata e fapaneng ya dihlopha le motswako wa mefuta e nnile ya ithutwa haholo ho fumana hore na ke efe e nang le tshebetso e phahameng. Diphuputso tsena di bontsha hore mehlala e atisang ho nepahala haholwanyane le yona e thata ho e utlwisisa; ho bonahala ho e na le kgwebo pakeng tsa ho nepahala le ho hlalosa. Ka lebaka leo, sena se lebisitse tlhokomelong e eketsehileng ho bohlale bo hlakileng ba maiketsetso, lefapha la dipatlisiso le amanang le ho hlalosa dikgakanyo tsa mohlala.

Leha bohlale ba maiketsetso bo hlaloswang e le sebaka sa dipatlisiso se ntseng se hola setumo se ntseng se hola setjhabeng sa mahlale, ho ntse ho na le dithuto tse fokolang tse hlahlobang tshebediso ya yona kotsing ya ho se be teng ha mekitlane. Kotsi ya ho se be teng ha mokitlane e bolela tahlehelo ya ditjhelete e ka bang teng kapa kotsi e hlahiswang ke mofani wa mokoloto ha motho ya tlamang a hloleha ho fihlela mekoloto ya hae. Ho lekanya, ho fokotsa le ho laola kotsi e amanang le ho fana ka mokoloto ka potlako, bafani ba mekitlane ba sebedisa dihlopha tsa dintlha ho lekola kotsi ya bakopi ba mekitlane pele ba fana ka mokoloto. Ho feta moo, bafani ba kotsi ya mokoloto ba hlokwa ka molao ho hlalosa dikgakanyo tsa dihlopha tsa dintlha.

Dihlopha tse tsebahalang tse sebediswang e le kotsi ya mokoloto di kenyelletsa ho theola maemo, hlahlobo ya kgethollo, difate tsa diqeto, meru e sa rerwang, pokello ya bootstrap, marangrang a neural, metjhini ya divector ya tshehetso le dialgorithms tse matlafatsang. Phokotso ya dintho le hlahlobo ya kgethollo di amohelwa haholo indastering ya ditjhelete hobane di sebetsa hantle ka mokgwa o utlwahalang mme ka tlhaho di ka tolokwa. Leha ho le jwalo, mekgwa ena e fana ka mokgwa wa mekgwa e meng e fanang ka ho nepahala ho ntlafetseng ha ho hlahlojwa kotsi, le hoja mekgwa ena e meng e se na tlhaloso; ha di utlwisisehe mme hangata di nkwa e le mabokose a matsho. Kgaello ena ya hlaloso e bakile ho qeaqea ho sebedisa mekgwa ena e meng

ya ho fana ka mekoloto.

Sepheo sa thuto ena ke ho tlosa mokwallo o boletsweng ka hodimo wa ho sebedisa mehlala ya diblackbox ka ho sebedisa mekgwa e hlakileng ya bohlale ba maiketsetso, jwalo ka dihlaloso tsa tlatsetso tsa Shapley le dihlaloso tsa sebaka sa habo bona tsa agnostic. Boithuto bona bo boetse bo hlahloba kgwebo e nepahetseng le hlaloso e nepahetseng ya dihlopha tse fapaneng ka ho theha le ho lekola mefuta e robedi ya dikarolo ho didatabase tse pedi tse fumanehang phatlalatso ya tsa mekoloto.

Ho ile ha ahwa mefuta e robedi ya dikarolo, ho kenyeletswa lifate tsa liqeto, ho theoha ha thepa, hlahlobo ya kgethollo e tshwanang, metjhini ya divector tse tshehetsang, marangrang a maiketsetso a neural, aggregation ya bootstrap, moru o sa rerwang, le sehlopha se matlafatsang se bobebe. Tshebetso ya bona le hlaloso ya bona di ile tsa hlahlojwa ka mora ho kwetliswa le ho lokisa di-hyperparameters bakeng sa papiso e nepahetseng mabapi le kwetliso, diteko le ho netefatsa dikarolwana tsa data. Ho nepahala ha tshebetso ho ile ha lekanyetswa ho sebediswa sebaka se ka tlasa lekgalo ho disubsets tse 30 tse sa rerwang tse hlahisitsweng ho data ya netefatso. Ho feta moo, teko ya Kruskal Wallis le ya Dunn ya ho bapisa dintho tse ngata di ile tsa sebediswa ho beha maemo a ponelopele ka ho nepahala le ho fumana hore na diphapano tsa ho nepahala ha moelelo di bohlokwa ho latela dipalo. Hlaloso ya dihlopha tsena e ile ya etswa bakeng sa mehlala ya dibox tse bonaletsang le tse ntsho. Ho finyella diphello tsena, mehato ya bohlokwa ya ho lokisa esale pele e ile ya ntlafatswa ho fokotsa ho rarahana ha hlaloso ya mohlala ya lehae le ya lefatshe. Ntle le moo, dihlaloso tsa tlatsetso tsa Shapley le dihlaloso tsa sebaka sa sebaka sa motlolo wa agnostic di ile tsa sebediswa ho sekaseka bohlokwa bo lekanyeditsweng ba dikarolo le phello ya dikgakanyo.

Diteko di bontsha hore marangrang a maiketsetso a methapo ya kutlo, di-ensembles le di-algorithms tse ding tse thehilweng sefateng di feta haholo ho theoha ha thepa le hlahlobo e fapaneng ya kgethollo thutong ya pele. Leha ho le jwalo, diphetho tse hanyetsanang di fumanwa bakeng sa thuto ya mohlala ya bobedi, kaha tshebetso ya dihlopha di batla di bapiswa. Sena se bontsha hore tshebetso ya mohlala e itshetlehile ka data eo mehlala e ahilweng ho yona. Dithuto tsena tse pedi tsa dinyewe di bontsha hore phapang pakeng tsa ho nepahala le ho hlalosa ha se kamehla e leng nnete. Ho feta moo, dihlaloso tsa tlatsetso tsa Shapley di hlahisitse ditholwana tse tsamaellanang le sephetho sa ho toloka ha mekgwa e pepeneneng. Hlaloso ena ya post-hoc e re thusa ho utlwisisa hore na dikgakanyo di etswa jwang le hore na ke dintlha dife tse tlatseditseng ho bolela esale pele. Sena ke sa bohlokwa ho theha moralo o ka tsheptjwang le o ka tsheptjwang o sebedisang mehlala ya lebokose le letsho bakeng sa diqeto tsa mokitlane.

Patlisiso e totobatsa melemo ya ho sebedisa mekgwa e meng bakeng sa dintlha tsa kotsi ya mokoloto, e bontsha hore tshebetso e ka fapana haholo. E boetse e bontsa katleho ya dihlaloso tsa tlatsetso ya Shapley le dihlaloso tsa sebaka seo ho ka tolokwang tsa mohlala-agnostic ho hlalosa dikgakanyo tsa dihlopha tsa diblackbox. Leha ho le jwalo, e supa mathata a ho sebedisa dihlaloso tsa tlatsetso ya Shapley. Theko ya boleng bo felletseng e kanna ya ameha ho barekisi ba kantle, e ka amang bohlokwa ba karolo. Ka hona, mosebetsi o mong o a hlokahala ho ntlafatsa bokgoni ba ho bala boleng ba dihlaloso tsa tlatsetso tsa Shapley bakeng sa dihlopha tsa linear le diensembles tse ding.

Acknowledgement

I would like to express my sincere gratitude to my supervisors, Prof. Katherine Malan and Prof. Mardi Jankowitz. Their continuous support, patience, insightful comments and suggestions were invaluable to me throughout the research and writing process. Their willingness to share their immense knowledge and experience has greatly enhanced my technical skills and development in this field and professional career. I will be forever grateful.

My wife, Nongazi Melita Mtiyane, and my children, Sinethemba, Unathi, and Mpilonhle Mtiyane, from whom I derive my inspiration and sense of purpose, deserve special appreciation. I appreciate all of their prayers and encouragement, which kept me going. I also want to thank my mother, Panie Agnes Mtiyane, and my sister, Sandra Mabaso, for supporting my decisions.

I am grateful to my in-laws, Sipho and Maleshano Gxota, for their continuous support in overcoming obstacles that may have affected my academic journey.

Finally, I want to thank the almighty God for granting me the strength and wisdom to overcome all the challenges I had while I was pursuing my education. I extend my appreciation to everyone who has supported me directly and indirectly. I will continue to place my trust in you for my future endeavours.

Contents

List of Figures

List of Tables

Acronyms

GA genetic algorithm. 20

GBDT gradient boosting decision trees. 22

gboost gradient boosting. 12, 13

GLM generalized linear model. 21

ILIME influence-based local interpretable model-agnostic explanations. 13

IV information value. 31, 47, 48, 59, 60

KNN k-nearest neighbour. 21, 22, 24

Lasso R. lasso regression. 60

LBFGS limited-memory Broyden–Fletcher–Goldfarb–Shanno. 8, 47

LDA linear discriminant analysis. 1, 3, 6, 8, 13, 19, 29, 36, 38, 46, 47, 49–52, 54, 58, 62, 63, 65, 68, 73, 75

LGBM light gradient boosting machines. 13, 15, 29, 36, 46, 47, 49, 50, 54, 58, 60–63, 72, 73, 75

Liblinear library for large linear classification. 8, 47

LIME local interpretable model-agnostic explanations. 4, 13–15, 18, 25, 26, 28, 38, 47, 58, 72, 74, 75

LORE local rule-based explanation. 13

LR logistic regression. 1–4, 6–8, 11, 13, 19–22, 29, 36, 38, 46, 47, 49–51, 54, 58, 61–63, 65, 68, 73, 75

LSQR least squares solution. 47

MAPLE model-agnostic supervised explanations. 13

MCS multiple classifier system. 6, 20–22, 28, 75

MDA multiple discriminant analysis. 9, 20

NB naive Bayes. 21

newton-cg Newton method. 8, 47

PCC percentage correctly classified. 16, 18

PDP partial dependence plot. 13, 14, 18

QDA quadratic discriminant analysis. 8

RBF radial basis function. 9

RF random forest. 1, 3, 6, 11, 20–22, 29, 35, 36, 46, 47, 49, 50, 58, 60, 61, 63, 73, 75

RFE recursive feature elimination. 35, 60, 61

Ridge R. ridge regression. 60

ROC receiver operating characteristic. 17

RPA recursive partitioning algorithm. 3

SAGA stochastic average gradient descent. 8, 47, 49

SARB South African Reserve Bank. 3

SFFS sequential forward feature selection. 35

SHAP Shapley additive explanations. 4, 13, 15, 18, 25, 28, 38, 47, 54, 58, 68, 72–75

SMOTE synthetic minority oversampling technique. 36

SVD single value decomposition. 47, 49, 63

SVM support vector machine. 1, 3, 6, 7, 9, 20–22, 29, 33, 36, 38, 46, 47, 49, 50, 58, 62, 63, 68, 73, 75

TAX4CS transparency, auditability and explainability for credit scoring. 26

TN true negatives. 16

TP true positives. 16

VIF variance inflation factor. 35, 38, 48, 60

VSI variables stability index. 25, 26

WoE weight of evidence. 59

XAI explainable artificial intelligence. 1, 4, 6, 15, 18, 23–26, 73

XGBoost extreme gradient boosting. 12, 13, 22, 32, 43

XML explainable machine learning. 23

CHAPTER 1

INTRODUCTION

The field of explainable artificial intelligence (XAI) is a fast growing field of interest in the science community. This is due to the increase in the applications of prediction models, availability of large data as well as reported failures of complex predictive models, which can be traced back to the lack of transparency [Bücker et al., 2022]. Traditionally, prediction models were based on domain knowledge and were easy to understand. However, recent predictive modelling approaches have become more complex, resulting in higher accuracy but less transparency. Thus, there is a trade-off between the performance and explainability of prediction models. Often the terms explainability and interpretability are used interchangeably. Interpretability refers to the degree to which an observer can understand the cause of a decision [Miller, 2019; Molnar, 2022]. The aim of XAI is to provide insights as to how and why complex predictive models produce predictions [Markus et al., 2021].

XAI assists with the adoption of complex predictive models in areas such as credit risk management, which entails the approval or rejection of credit applications. In the context of credit risk management, these predictive models are referred to as credit scoring classifiers. Over the last few decades, credit approval decisions progressed from judgemental or intuitive approaches to automated scoring systems [Abdou and Pointon, 2011]. Traditional credit scoring approaches, such as logistic regression (LR) and linear discriminant analysis (LDA), involve the formalisation of relationships between variables in the form of mathematical equations. Moreover, they provide a fine balance between predictive ability and ease of interpretation. Alternative scoring classifiers, including support vector machine (SVM)s, artificial neural network (ANN)s, bootstrap aggregation (bagging), boosting methods and random forest (RF), utilise algorithms that can learn from data without relying on rule-based programming and have shown superior performance ability. The main challenge in utilising alternative approaches is that, despite the potential high predictive accuracy, they often lack transparency and interpretability [FSB, 2017]. Consequently, these methods are often referred to as black box models. This accuracy-explainability trade-off has hindered the adoption of complex predictive models for credit scoring. Figure 1 illustrates the trade-off between performance accuracy and explainability.

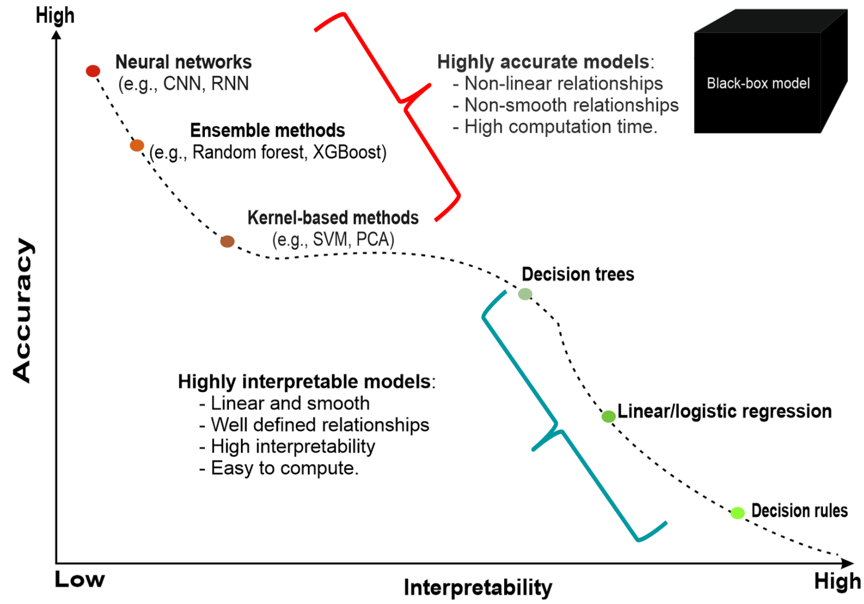

Figure 1: Accuracy-explainability trade-off (Figure 1.4 in Karim [2022]).

Figure 1 shows that complex models, which are capable of learning non-linear and non-smooth relationships in data, exhibit higher accuracy compared to traditional models such as decision tree (DT) and LR. However, these complex models are less interpretable than their traditional counterparts. The aim of this dissertation is to investigate the accuracy-explainability trade-off on credit scoring classifiers by assessing the performance and explainability of the classifiers for two case studies.

1.1 Background and rationale

Historically, credit approval decisions were based on an expert judgement approach that involved evaluating a customer's creditworthiness based on the 5Cs: character (reputation), capital (amount), capacity (earnings volatility), collateral, and condition (economic cycle) [de Servigny and Renault, 2004]. The success of the judgemental process is dependent on the credit analyst's or expert's experience and common sense. This approach has the advantage of considering the qualitative aspects of a customer. However, the disadvantage is the potentially subjective, inconsistent, and biased evaluations [Abdou and Pointon, 2011].

The credit lending landscape has shifted significantly from judgemental to automated credit scoring systems. Technological advancements resulted in the deployment and widespread utilisation of automated credit scoring systems, and the adoption of statistical scoring methods to aid in credit decision making. Popular credit scoring approaches include LDA, LR and recursive partitioning algorithm (RPA) [van Gestel and Baesens, 2008; Thomas et al., 2002]. These are classification scoring approaches that are used to support credit strategies and decision-making throughout the credit life cycle, namely acquisitions or origination, account management, and collections.

The main purpose of credit scoring is to differentiate between good and bad credit customers which has lead to improved credit processing times, reductions of process costs, and the minimisation of errors [Abdou and Pointon, 2011]. Therefore, the performance in terms of predictive accuracy plays a critical part in the success of credit scoring. De Servigny and Renault [2004] argue that an optimal scoring model must have high accuracy and feasibility. This entails low error rates resulting from reasonable assumptions, as well as efficiency and ease of implementation.

De Servigny and Renault [2004] also state that an optimal scoring model must meet other criteria, namely parsimony and transparency. This means using a reasonable number of explanatory variables, along with producing explainable results. Creditors are required to be able to explain reasons behind credit decisions [Dastile et al., 2020]. Consequently, creditors prefer to use models that are transparent and interpretable, sometimes compromising on accuracy and performance. In addition, primary lenders such as banks are regulated by international committees, such as the Basel Committee on Banking Supervision (BCBS), local regulators, such as the South African Reserve Bank (SARB) and auditors to ensure that they comply with lending regulations. This is to prevent reckless lending, biases when lending and to manage credit risk proactively. Decisions made using automated scoring systems must be free of biases and in line with lending legislation and regulations.

Scoring approaches can be used to overcome issues around bias and inconsistency when making decisions to grant credit where lending to customers remains largely intuitive. In recent years, there has been a rapid advancement of credit scoring classifiers that serve as alternative to conventional techniques like LR and LDA and can be used to model complex multivariate non-linear relationships in contrast to traditional linear techniques [van Gestel et al., 2005; Abdou and Pointon, 2011]. These alternative classifiers are deemed to be black boxes because often they are difficult to understand (lack transparency and interpretability). The literature on these classifiers, which include SVM, ANN, bagging, boosting methods and RF, suggests that they outperform the traditional approaches. In addition, these alternative classifiers are broadly categorised as neural networks, ensemble methods and kernel-based methods as shown in Figure 1.

1.2 Problem statement

Upon receiving applications for credit, lenders must decide whether or not to grant credit and to which customers. The decisions are usually aided by the use of scorecards and automated systems. Nonetheless, lenders must be able to accurately discriminate between good and bad customers in a fair manner. Furthermore, credit decisions must be in line with the objectives of the business, generally to minimise risk and maximise profit or, equivalently to minimise losses [Witzany, 2017].

The likelihood of customers defaulting is estimated using a statistically sound approach, such as a classification model. An accurate assessment of a customer's degree of risk or probability of default is imperative for a lender. Lenders must determine their risk appetite or the level of risk that they are willing to accept. They must decide whether to approve or decline credit applications depending on their risk appetite. This research will assist with predicting of default risk and enable explanations for predictions. The research was conducted using publicly available data from the Kaggle and UCI machine learning online repositories.

1.3 Aims and objectives of the research

The aim of this study is to investigate the accuracy-explainability trade-off on credit scoring classifiers.

The main objectives of this project are to:

- Explore the advantages and effectiveness of alternative approaches in the context of credit applications, as this can improve the accuracy of predictions to discriminate between good and bad customers. There is a large body of literature on LR and other transparent approaches, but limited studies and recommendations on the use of black box models.
- Analyse the challenges and limitations of using machine learning techniques to score customers within the credit risk management framework. Many machine learning classification models are deemed as black box models, i.e. outcomes are not explainable. This has resulted in the reluctance to adopt and utilise these models in practice. This study explores the use of XAI methods, such as Shapley additive explanations (SHAP) and local interpretable model-agnostic explanations (LIME), to explain reasons behind predictions.

The work on these aspects is currently limited. This study contributes to the ongoing research on credit scoring approaches and their application in credit risk management, with a view to optimise credit decisions. Furthermore, this research seeks to contribute to a growing field of study on the transparency and explainability of such models, especially within the highly regulated domain of credit risk management.

1.4 Dissertation structure

This research is organised as follows: In Chapter 1, the introduction presents a brief overview of the background, research problem and the research objectives. Chapter 2 presents the theoretical foundation on credit scoring models frequently used in literature. The evaluation of classification models and techniques used to make models transparent and explainable are discussed. Chapter 3 reviews the relevant literature on the accuracy or performance of various credit scoring techniques as well as the challenges of these approaches. A survey of related work on the transparency and explainability of advanced classifiers is presented. Chapter 4 discusses how the research was carried out. The computer application, the data collection and analysis, preprocessing and model construction and approaches on explainability and interpretability are outlined. Chapter 5 presents the results of the data wrangling, analysis and preprocessing. Chapter 6 discusses the results achieved by this research. Chapter 7 provides a summary of the research, stating the research contributions and recommendations for future work.

CHAPTER 2

BACKGROUND CONCEPTS

The Board of Governors of the Federal Reserve System [2011] defines a model as “a quantitative approach that applies mathematical, statistical, economic and financial theories, techniques and assumptions to process input data into quantitative estimates”. Credit scoring involves constructing models that can be used to estimate the default risk associated with credit applicants. The estimated risk is then used to develop credit strategies, such as deciding whether to accept, decline or refer a credit application. These decisions have an impact on the profitability of financial institutions [Thomas et al., 2002; Abdou and Pointon, 2011].

This chapter briefly presents the theoretical foundation of credit scoring classifiers and the explainability of these classifiers. Several classification models commonly used for credit scoring, including DTs, LR, LDA, SVM, ANN, bagging, boosting and RF are presented. Furthermore, the techniques used to understand the behaviour of these classification models are explained. The field of study that deals with explaining and interpreting the behaviour of classification models is referred to as XAI.

2.1 Credit scoring classifiers

Extensive research has been conducted on individual classification models, such as LR, LDA, DT based algorithms, SVM, ANN, as well as multiple classifier system (MCS)s to predict the risk of default. LR and LDA are the most widely used classification models in credit risk management due to their interpretability (the level to which one can understand the reasons behind predictions) [Dastile et al., 2020]. However, these models require the formalisation of relationships between features and a dependent variable in the form of a mathematical formula. Alternative approaches, such as the SVM, ANN and some ensemble systems, employ algorithms that can identify complex patterns in large volumes of data and learn from data without relying on rule-based programming [Dangeti, 2017; FSB, 2017]. These alternative approaches tend to be more accurate in predicting the risk of default. However, they are often difficult to explain [Kollár et al., 2015; Dastile et al., 2020].

2.1.1 Decision trees

A DT is a machine learning algorithm that entails recursively partitioning a data space and fitting a prediction model within each partition. Given a dataset D, with a subspace or feature space of n predictor variables, i.e., $\mathbf{x} = (x_1, x_2, \dots x_n)$ and a dichotomous class variable $y \in \{0, 1\}$, the DT involves partitioning the feature space $\mathbf{x}$, one feature at a time, into a finite number of disjoint subsets until a class can be predicted [Loh, 2011].

A DT is commonly depicted as a tree-like structure providing a hierarchical representation of the feature space and the relationships among the data. A DT is made up of a root node which represents the entire population, branches or subtrees which represent the decisions and leaf nodes which are terminal nodes, i.e., subsets that are usually not partitioned further due a stopping criteria, for example, a specified maximum depth of the tree.

A number of methods, known as measures of impurity, which include the Kolmogorov-Smirnov statistic, the Gini index, entropy index or the chi-square statistic can be used to partition or split the subspace [Witzany, 2017]. These measures provide a measure of the good and bad populations in a partition A_j, in each node or leaf in the tree diagram. The measures that are commonly used in literature are the entropy and Gini index, also referred to as the Gini impurity. However, the best measure of node impurity usually depends on the data set [Brown and Myles, 2013]. The process of splitting or partitioning is recursive and stops when a particular stopping condition is reached.

2.1.2 Logistic regression

A LR model is a parametric statistical technique, developed to discriminate between two or more groups. It uses a mathematical function to determine the relationship between a dependent variable and one or more independent variables.

Consider a dichotomous response variable $y \in \{0, 1\}$ associated with a collection of n independent features denoted by the vector $\mathbf{x} = (x_1, x_2, \dots x_n)$, for each member in a dataset D. Let $\pi(\mathbf{x})$ be the posterior probability $\mathrm{P}(y = 1|x_1, x_2, \dots x_n)$, for each member. Assume that the posterior probability is governed by a logistic or sigmoid function where the input is a linear combination of features x_i for $i = 1, 2, \ldots, n$, i.e.,

$$\pi(\mathbf{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_n}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_n}}. \tag{1}$$

The logistic function, which restricts the outcome to the interval $[0, 1]$, is a bounding function. The name LR is derived from the bounding logistic function utilised. It can be deduced from Equation 1 that

$$\log\left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_n, \tag{2}$$

where $\beta_0, \beta_1, \ldots, \beta_n \in \mathbb{R}$. The parameters β_i, where $i = 0, 1, \ldots n$, are determined using the maximum likelihood estimation and are obtained by fitting Equation 2 to the data. Stochastic average gradient descent (SAGA), Newton method (newton-cg), library for large linear classification (Liblinear) and limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) can be used to estimate these parameters.

2.1.3 Discriminant analysis

Discriminant analysis is a parametric statistical technique, developed to discriminate between two groups. There are different approaches leading to the formulation of the LDA and quadratic discriminant analysis (QDA). These approaches include, the decision theory or probabilistic approach, separating the two groups approach or Fischer's interpretation, and the linear regression approach. This section presents an outline of the decision theory approach described by Thomas et al. [2002] and James et al. [2013].

Consider a dichotomous response variable $y \in \{0, 1\}$ associated with a collection of n independent variables denoted by the vector $\mathbf{x} = (x_1, x_2, \ldots x_n)$ for each member in a dataset D. Each class $y \in \{0, 1\}$ is assigned a prior probability $\pi_y = \frac{N_y}{N}$, where N_y is the number observations in class y and N is the total number of observations. According to Bayes' rule the posterior probability is

$$P(y|\mathbf{x}) = \frac{f_y(\mathbf{x})\pi_y}{\sum_{i=0}^{1} f_i(\mathbf{x})\pi_i}, \tag{3}$$

where$f_y(\mathbf{x})$ is the density of $\mathbf{x}$ given y. Assume that $f_y(\mathbf{x})$ is a multivariate Gaussian density function

$$f_y(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} \det \mathbf{\Sigma}_y^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^T \mathbf{\Sigma}_y^{-1} (\mathbf{x}-\boldsymbol{\mu}_y)\right), \tag{4}$$

where n is the dimension of $\mathbf{x}$, $\mathbf{\Sigma}_y$ is the covariance matrix and $\boldsymbol{\mu}_y$ is the mean vector.

The LDA function is obtained by assuming $\Sigma_1 = \Sigma_0 = \Sigma$ and solving for the decision boundary $P(y = 1|\mathbf{x}) = P(y = 0|\mathbf{x})$.

The discriminant equation is of the form $\mathbf{x}^T\mathbf{M}+\mathbf{C}$ which is a linear function. However, the general form is a quadratic function of the form $\mathbf{x}^T\mathbf{A}\mathbf{x}+\mathbf{B}^T\mathbf{x}+\mathbf{C}$. The quadratic form is obtained when $\Sigma_1 \neq \Sigma_0$. Furthermore, when employing multiple discriminant functions, the technique is referred to as multiple discriminant analysis (MDA).

2.1.4 Support vector machines

An SVM is a machine learning technique commonly used in classification problems. It aims to find an optimal hyperplane with a maximum margin, to discriminate between two classes [Goh and Lee, 2019]. The hyperplane is a function that separates different classes. The distance between support vector points and the hyperplane is called the margin. Fitting an SVM to discriminate between classes requires finding the solution to the following optimisation problem:

$$\underset{\mathbf{w},\epsilon_i,b}{\text{Minimize}} \quad \phi(\mathbf{w},b) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C\sum_i \epsilon_i \tag{5}$$

$$\text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}+b) \geq 1-\epsilon_i, \quad i = 1,2,...,n \tag{6}$$

where $\mathbf{w}$ represents the margin, b is the bias term, C is the penalty hyperparameter and ϵ_i is the slack variable introduced to account for misclassification. The global maximum of the quadratic function can be determined by utilising the Lagrange function. However, when there is no feasible solution, radial basis function (RBF), or polynomial kernels functions are applied to modify the SVM formulation for nonlinear classification [Goh and Lee, 2019; Dangeti, 2017].

2.1.5 Artificial neural networks

An ANN is a machine learning process inspired by biological neural network systems. Biological neural networks comprise neurons which are responsible for receiving information or signals from the internal and external environment. These signals are processed and transmitted to other neurons and to effector organs. Similarly, artificial neural networks receive information or signals in the form of vector inputs $\mathbf{x} = (x_1, x_2, \ldots x_n)$, where $\mathbf{x}$ is a subspace of features of a dataset. Each input feature is associated with a weight and transformed by an artificial neuron made up of a net input function, also referred to as a combination function, and an activation function. Each artificial neuron can connect to another, i.e., contain multiple hidden layers and finally produce an output as depicted in Figure 2.

A single-layer neural network consists of only one hidden layer and is expressed

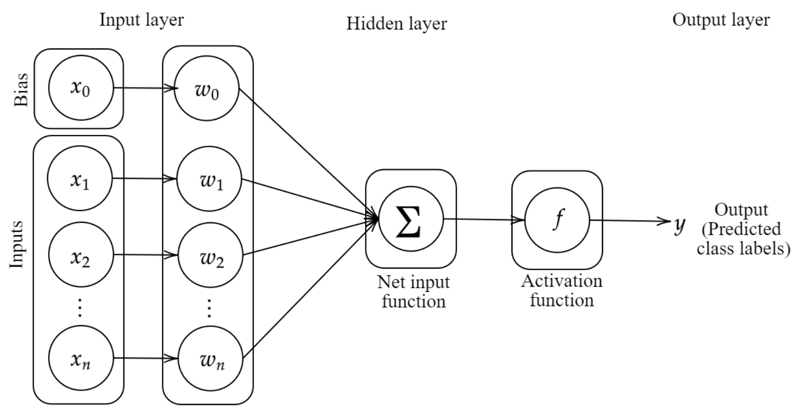

Figure 2: A single-layer neural network classification model.

mathematically as

$$u_k = \sum_{i=0}^{n} w_{ki} x_i \tag{7}$$

$$y = f(u_k) \tag{8}$$

where w_{ki} are the weights. Positive weights are called excitory and they increase the value of the net input function u_k. Negative weights are called inhibitor and they reduce the value of u_k. The net input function need not be a linear function, however the linear form is commonly used in literature and application. k indicates the neuron to which the weight applies and i indicates the variable. Furthermore, x_0 is the bias term as shown in Figure 2 [Thomas et al., 2002]. The activation function f restricts the value generated by the net input function to an interval, often $[0, 1]$ or $[-1, 1]$. Various activation functions are used in the application of neural networks, including the hyperbolic tangent function, logistic function and rectified linear activation function. Furthermore, the gradient descent algorithm is commonly applied to model training to minimise the error in prediction.

2.1.6 Bootstrap aggregation

Bagging is an ensemble method that converts a series of weak or base classifiers into a single strong classifier. A weak classifier, or learner, is a classifier that performs better than random guessing. These weak classifiers are trained on bootstrapped samples generated from the entire training dataset. Additionally, the strong classifier is constructed by aggregating the predictions of the weak classifiers using a voting system. Bagging has the potential to reduce the variance in the final model [Dangeti, 2017]. The bagging algorithm described by Wang et al. [2011] is as follows:

Given a training set D and a base learner $h(x_i)$, then for $t = 1, 2, \ldots, T$ iterations:

1. Generate a subspace or bootstrap sample D_t from D.
2. Fit a learner H_t to each D_t.

The final hypothesis is of the form

$$H(x) = \underset{y}{\mathbf{argmax}} \sum_{i=1}^{T} I(y = h_t(x))$$

where $I(y = h_t(x)) = 1$ when $y = H_t(x)$, otherwise $I(y = h_t(x)) = 0$.

The bagging method used in this study uses LR as base classifiers and is referred to as bagged LR.

2.1.7 Random forests

Closely related to bagging is the RF algorithm which integrates the concept of generating random subspaces (feature subset) and bagging [Nisbet et al., 2009]. In bagging, all the input features are used for each sample, whereas in a RF, a subset of features is selected in addition to the bootstrap samples [Trivedi, 2020]. The RF algorithm described by Han et al. [2020] is as follows:

Given a training set D with n features and T classifiers:

For $t = 1, 2, \ldots, T$

1. Generate a subspace D_t from D.
2. Fit a tree using a subset of random features from D_t.
   For a given node:
   (a) Randomly select $m \approx \sqrt{n}$ or $m \approx n/3$.
   (b) Find the best split features and cutpoints using the feature subset.
   (c) Send down the data using (b).
   Repeat (a) - (c) until terminating conditions are met.
3. Develop trained models C_t.

Use simple majority voting to fuse the T trained models.

2.1.8 Boosting

Boosting is an ensemble technique that converts a series of weak classifiers, also referred to as weak or base learners, to a strong classifier. A weak learner is a classifier that performs better than random guessing. The fundamental assumption of boosting is that a weak learner produces a weak hypothesis that is better than random guessing. This is known as the weak learning assumption [Schapire and Freund, 2012]. The weak learners in boosting are trained sequentially on modified versions of the data, whereas in bagging they are trained in parallel. Moreover boosting does not involve bootstrap sampling, unlike bagging. The learners are then aggregated to create a strong classifier [Dangeti, 2017].

Boosting entails generating a series of classifiers repetitively. At each iteration, a base classifier is trained on a different subset of the training set based on an iteratively computed distribution or weighting over the sample of the training set. Furthermore, a higher weighting is placed on the misclassified observations. The final classifier is determined by computing the weighted average of the preceding classifiers [Theodoridis and Koutroumbas, 2009].

Boosting refers to a family of algorithms, which include adaptive boosting (adaboost), gradient boosting (gboost) and extreme gradient boosting (XGBoost). The adaboost algorithm was formulated by Freund and Schapire [1997]. Friedman [2001] developed the regression and classification gboost algorithms.

The gboost classification algorithm described in Friedman [2001] and Natekin and Knoll [2013] is as follows:

Consider a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ as input, where x_i belongs to some feature space X^m and y_i is a response variable. A differentiable function $L(y_i, \gamma)$ that will be used to evaluate how well the algorithm models the training set is defined. The function $L(y_i, F(x_i))$ is referred to as the loss function. There is a wide range of loss functions that have been developed, the choice of which depends on the response variable y_i. The most frequently used loss functions for classification, i.e., when y_i is a categorical response variable, include the Binomial loss function and the Adaboost loss function. A base-learner $h(x_i)$ and the maximum number of iterations T are then defined.

For $t = 1, 2, \ldots, T$

1. Initialise model with a constant value: $F_0(x) = \underset{\gamma}{\textbf{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$.
2. Compute the pseudo-residuals or negative gradients $g_t(x_i)$.
3. Fit a new weak learner $h_t(x)$.
4. Compute the multiplier or the best gradient step-size:

$$\gamma_t = \underset{\gamma}{\textbf{argmin}} \sum_{i=1}^{n} L(y_i, F_{t-1}(x_i) + \gamma \cdot h_t(x_i)).$$

5. Update the model: $F_t(x_i) = F_{t-1}(x_i) + \gamma_t \cdot h_t(x_i)$.

The most used base learners can be categorised into three model classes, namely linear models, smooth models and decision trees. In addition, a combination of different base learners can be used [Natekin and Knoll, 2013].

There are variants of gboost algorithms such as XGBoost, light gradient boosting machines (LGBM) and CatBoost which are improvements on the original gboost algorithms. A popular variant is the XGBoost, in which the loss function is normalized in order to eliminate model variances. The XGBoost algorithm reduces the likelihood of model overfitting. Furthermore, while gboost uses the first derivative in learning, XGBoost improves the loss function with Taylor expansion [Chang et al., 2018]. LGBM credit scoring classifier using DTs as base classifiers is constructed and used in this study.

2.2 Explainability of classifiers

The explainability and interpretability of classification methods can be challenging and may be a very important aspect of model predictions. Explainability and interpretability enable humans to understand the predictions of the models and they encourage trust in the models. The more complex the architecture of the model, the more difficult the explainability and justification of why a prediction was obtained. Various approaches are utilised in attempt to understand the effects of features on model predictions such as partial dependence plot (PDP) [Friedman, 2001], SHAP [Lundberg and Lee, 2017], LIME [Ribeiro et al., 2016], anchors [Ribeiro et al., 2018], local rule-based explanation (LORE) [Guidotti et al., 2019], influence-based local interpretable model-agnostic explanations (ILIME) [ElShawi et al., 2019] and model-agnostic supervised explanations (MAPLE) [Plumb et al., 2018]. These approaches are broadly categorised as local or global methods. Local interpretation methods explain individual predictions whereas global methods describe the average behaviour of a machine learning model. In addition, approaches that can be used for any classifier are said to can be model-agnostic and those that apply to specific classifiers are said to be model-specific.

2.2.1 Intrinsic explainability

There are classification models that are considered transparent, or glass box models, because they are inherently explainable, such as LR, LDA and DT. In the cases of LR and LDA, the contribution of the features is provided by the model coefficients. Additional analysis of confidence intervals and statistical significance demonstrates the consistency and applicability of feature attributions in order to build trust in the model prediction. A DT is also considered as an interpretable model because it can be displayed visually as a tree diagram or partitions of the feature space, to explain how the prediction was made. However, even DTs can be difficult to visualise and interpret if the depth of the tree is excessively large.

2.2.2 Partial dependence plots

A PDP is a global model-agnostic method that illustrates the dependence of predictions on the joint values of the input features. They depict the marginal effect of one or two features on a classification model's predicted outcome. For a classification problem where the model outputs probabilities, the PDP displays the probability for a certain class given different features values. Additionally, a PDP can show whether the target-feature relationship is linear, monotonic, or more complex [Molnar, 2022]. However, this method of interpretation is difficult to use for high dimensional feature spaces and is therefore limited to a low number of input features. It is useful when there is a low order of interaction between variables or when features are uncorrelated [Friedman, 2001].

2.2.3 Local interpretable model-agnostic explanations

LIME is a local model-agnostic method, in which local surrogate models that are considered interpretable are trained and used to approximate the predictions of less interpretable model. LIME tries to fit a local model using sample data points (interpretable representation) that are similar to the observations being explained. This ensures that explanations are locally faithful, even though they may not be faithful globally or lack global fidelity. The primary objective of LIME is to find a model that is interpretable over the interpretable representation and that is locally faithful to the underlying classifier [Ribeiro et al., 2016].

The optimisation problem to be solved for LIME as proposed in Ribeiro et al. [2016] is formulated as follows: Given a classifier f and a local interpretable surrogate model g, the problem to be solved is

$$\xi(x) = \underset{g \epsilon G}{\textbf{argmin}}\, L(f, g, \pi_x) + \Omega(g) \tag{9}$$

where $\xi(x)$ is the explanation, $L(f, g, \pi_x)$ is a measure of how unfaithful g is in approximating f in the locality defined by π_x, and $\Omega(g)$ is the complexity of the local model g. $L(f, g, \pi_x)$ must be minimised and g must be comprehensible to ensure both local fidelity and interpretability. This formulation can be used with different explanation families G, loss functions L, and complexity measures $\Omega(g)$.

Based on Molnar [2022], the steps for training the approximating model g are as follows:

1. Select an instance for which an explanation of the black box prediction is needed.
2. Generate new weighted samples, based on their distances from to the selected instance.

3. Perturb the new dataset and obtain the predictions of the black box model for these new points.
4. Train a local, interpretable model on the weighted dataset.
5. Use the trained local model to generate explanations for the prediction.

An advantage of LIME is that it can be used to explain any classification model because it does not depend on the original classifier or algorithm used. However, one of the drawbacks of LIME is that it is sensitive to the accuracy of the surrogate model. Gramegna and Giudici [2021] state the importance of explainability in the context of credit risk. It will promote the use of black box models and be used to address ethical and regulatory concerns. Furthermore, they state that LIME is one of the widely recognised and state-of-the-art frameworks in XAI. Given the wide acceptance of this approach, it is used in this study to explain the prediction of the LGBM at a local instance level.

2.2.4 Shapley additive explanations

The SHAP framework, proposed by Lundberg and Lee [2017], is a technique used to explain the outputs of any classification model. It was derived from Shapley values, which are used in game theory to equitably share the gains among players when their contributions are unequal in a coalitional game setting. According to Molnar [2022], an explanation can be obtained by treating each feature value as a player in a game and viewing a prediction as the payout. The underlying assumption of Shapley values is that the features collaborate to influence the model's prediction.

Lundberg and Lee [2017] point out that Shapely values satisfy the following three properties:

1. **Local accuracy:** ensures that the output of the explanation model matches the output of the original model for a specific input.
2. **Missingness:** features that are not part of the prediction of an instance will have a Shapley feature importance values of zero, indicating that they have no impact on the explanation.
3. **Consistency:** if the contribution of a feature x is greater in a model A than model B, then the Shapley feature importance value of x will be higher in A than B. This property also means that, if the impact of x increases in a model, the Shapley feature importance value will also increase.

Furthermore, SHAP can be used as a local model-agnostic method. It is considered to be more robust than LIME, because unlike LIME, it fairly distributes the contributions of features over all subsets of features. SHAP is used for feature attribution and to understand the relationship of the features and predictions.

2.3 Performance evaluation metrics

Several metrics are used in the literature to evaluate the performance of classification models and the most common are the percentage correctly classified (PCC) metrics, area under the curve (AUC) and Gini coefficient. These metrics are used to evaluate the discriminatory and predictive power of the models. Statistical tests, like t-tests, ANOVA, Kruskal Wallis and Dunn's multi-comparison test, are used to compare the performance of different classification models.

2.3.1 Percentage correctly classified

The PCC metrics are a group of ratios calculated from predicted positive and negative outcomes compared to actual positive and negative outcomes. A positive outcome is one in which an event occurs and a negative outcome is one in which an event does not occur. In credit scoring a positive outcome is one in which a customer defaults and a negative outcome is one in which the customer does not default.

True positives (TP) are the number of cases where the predicted outcomes and actual outcomes are positive. True negatives (TN) are the number of cases where the predicted outcomes are negative and actual outcomes are negative. False positives (FP), also referred to as type I error, are when the predicted outcomes are positive, but the actual outcomes are negative. False negatives (FN) or type II error are the total instances where the predicted outcomes are negative, but the actual outcomes are positive.

The main three PCC measures used to evaluate a binary classifier include accuracy, precision and recall. The PCC metrics are defined mathematically as follows:

The accuracy measures the proportion of outcomes that were predicted correctly

$$\text{accuracy} = \frac{TP + TN}{(TP + FP + FN + TN)}. \tag{10}$$

The precision is a measure of the fraction of true positive predictions relative to the total predicted positive outcomes

$$\text{precision} = \frac{TP}{(TP + FP)}. \tag{11}$$

The recall, also referred to as sensitivity or true positive rate, is a measure of the fraction of true positives relative to the total actual positive outcomes

$$\text{recall} = \frac{TP}{(TP + FN)}. \tag{12}$$

The F-measure (or F1-score) is the harmonic mean of the precision and recall.

This measure is interpreted in the same way as the average accuracy, however it is commonly used when the data is imbalanced or skewed

$$\text{F-measure} = 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{13}$$

2.3.2 Area under the receiver operating characteristic curve

The AUC statistic is derived from two measures, namely, sensitivity (Equation 12) and specificity. The specificity (true negative rate) measures the fraction of negatives that are correctly classified relative to actual negatives

$$\text{specificity} = \frac{TN}{TN + FP}. \tag{14}$$

The AUC is used to measure the performance of a classification model at various thresholds. It is a measure of separability for a binary classification model. An AUC value close to 1 indicates that the model has a good measure of separability and a value of 0.5 indicates that the model has no separating power. A value of 0 indicates that the model is reciprocating the outcomes, i.e. defaults and non-defaults are misclassified.

Figure 3 is a graphical representation of the AUC. The receiver operating characteristic (ROC) curve is a probability curve and is obtained by plotting the 1− specificity (false positive rate) on the x-axis against the sensitivity on the y-axis. The AUC is the area under the ROC curve.

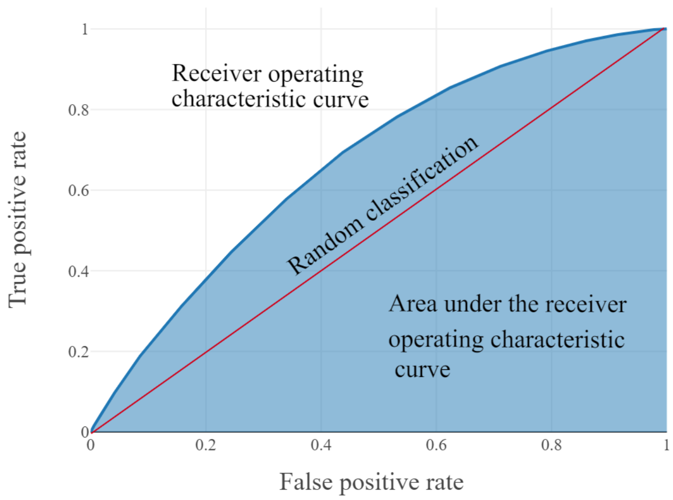

Figure 3: Area under receiver operating characteristic curve

2.3.3 Multiple comparisons tests of mean accuracy

Multiple comparisons of means tests provide a way to determine if the means of the predictive accuracy of each classifier are statistically different. The statistical significance of the means can be assessed using either a set of confidence intervals or a set of hypothesis tests. In order to achieve this ANOVA tests can be conducted. This test is used if three assumptions about the means holds. Firstly, ANOVA assumes that the residuals are normally distributed. Secondly, ANOVA assumes homogeneity of variances, which means that the variance among the groups should be approximately equal. Thirdly, ANOVA assumes that the observations are independent of each other. If the assumptions do not hold, non-parametric tests can be used. In this study, non-parametric tests, such as the Kruskal Wallis test together with the Dunn multi-comparison tests are used to determine the statistical significance of the differences in mean accuracy of classifiers.

2.4 Summary

In this chapter, classification methods that are commonly used in literature on credit scoring classifiers are presented. These methods are often categorised as transparent or non-transparent. Transparent means that the predictions are explainable and can be understood by humans. Various methods, such as PDP, LIME and SHAP are proposed in the literature in an attempt to explain the predictions of non-transparent methods. The ability to understand and explain model inputs and outputs is important for credit providers to meet regulatory requirements, therefore XAI is a crucial field for credit risk management. Different classification methods perform differently. Some methods are more accurate or more efficient than others. The metrics used to measure the performance are explained, this includes PCC metrics, AUC as well as tests to assess if the means of the predictive accuracy of each classification model are different. A detailed literature review on the performance of the different classification models and explainability approaches are explained in Chapter 3. The methodology, data analysis and results of the study are presented in Chapters 4, 5 and 6, respectively.

CHAPTER 3

LITERATURE REVIEW

The literature on credit scoring classifiers indicates that different types of classifiers yield varying levels of performance. Several studies show that transparent models such as LR and DT are often outperformed by alternative approaches. These alternative approaches appear to be more accurate in predicting default risk than transparent models. However, the drawback of adopting these alternative models is their lack of explainability and they fail to meet regulatory requirements. Seemingly, there is a trade-off between accuracy and explainability of classification models.

This chapter provides a literature review of classification models frequently employed in credit scoring research. The research findings of various individuals models are reviewed, followed by studies on combinations of modelling approaches. Additionally, limitations and challenges associated with certain methods are examined. The approaches for improving the explainability of these methods are explored.

3.1 Performance of classification models

The most common and utilised classification models in credit scoring are LR and LDA. Despite the common use, there is criticism against the use of LDA in credit scoring. Several researchers caution against the use of inaccurate prior probabilities, linear functions instead of quadratic functions and potential classification errors [Abdou and Pointon, 2011]. Furthermore, Wang et al. [2011] indicate that techniques like LDA assume that the independent variables conform to a multivariate normal distribution, and this assumption is often not satisfied in practice, rendering these techniques invalid for finite samples. Additionally, Thomas [2000] asserts that LDA and LR assume that the variables have a linear relationship, whereas this relationship is non-linear in general, leading to inaccuracies.

A wide range of techniques, which can be used for scoring, have been studied to ascertain their relative performance over the past two decades. A review by Alaka et al. [2018] explores how MDA, LR, ANN, SVM, rough sets, case based reasoning, DT, and genetic algorithm (GA)s applied to bankruptcy prediction perform when assessed on thirteen criteria. The criteria are broadly classified into three categories: results related criteria, data related criteria and tools' properties related criteria. Results related criteria encompass accuracy, interpretation of results as well as cases where the technique fails to make classifications (non-deterministic output). Data related criteria comprises aspects of the data that may affect the performance of the technique, which includes the size of the sample data, class imbalance (data dispersion), feature selection method, sensitivity to linear correlations between features and the ability to analyse different types of variables. The tools' properties related criteria refers to inherent limitations of the technique used. This covers the limitations of the technique to handle linear or non-linear relationships, assumptions that the data must satisfy for the technique to function optimally, ability to generalise (tendency to underfit or overfit), time to develop the model and the ease with which it can be updated as well as the degree to which it is easily hybridisable (integration ability). Overall, no single method was determined to be significantly superior than others in relation to the thirteen stated criteria. Moreover, it can be concluded that constructing a hybrid model by integrating different methods could yield overall better performance model.

Chopra and Bhilare [2018] carried out a study to examine the superiority of approaches that involve combinations of classifiers (hybrid models) to predict banking loan defaults. The study involved the use of ensembles, a particular class of machine learning techniques involving the combination of multiple classifiers. They investigated the performance of bagging, boosting and RF ensembles and compared them to DT to evaluate the relative performance. The study showed that the gradient boosting model performed better than the benchmark DTs.

In the last few years MCSs attracted great attention in the scientific community across various disciplines like health care, speech, image classification, forecasting and other applications [Ganaie et al., 2022]. In different studies in the literature MCSs are referred to as ensemble based systems, committee of classifiers, classifier fusion and mixture of experts [Abellán and Castellano, 2017]. MCSs involve the amalgamation of two or more individual classifiers into a single super classifier using a heuristic algorithm or combination rule [Zang et al., 2014]. This approach showed potential to enhance the predictive power of classification models [Ala'raj and Abbod, 2016; Ghodselahi, 2011; Lessmann et al., 2015; Yao et al., 2022]. A common combination rule used in literature is that of voting, which can be categorised as hard, soft or weighted voting. Hard voting, also referred to as majority, entails counting the predictions for each class label and predicting the class label with the highest number of votes. Soft voting requires aggregating the probabilities by summing, averaging or taking the maximum and comparing the result to a threshold value to predict the class. Majority voting and weighted average are the most commonly used voting strategies in the literature [Nalić et al., 2020].

Numerous approaches to combined classifiers were developed in literature, given the success of the performance of MCSs. Ala'raj and Abbod [2016] explored studies on MCSs employed for credit scoring that were published between 2005 to 2015. A comparison was made by examining the number of datasets used, homogeneity or heterogeneity of the developed classifier ensembles, rules used to combine the classifiers, performance assessment, and if statistical significance tests were conducted. In the nineteen papers reviewed, the authors point out that most researchers opted to use homogeneous ensemble classifiers. Heterogeneous classifiers were developed in only two studies. There were three papers in which both heterogeneous and homogeneous classifiers were developed in the same study. Over and above that, majority vote was the most used combination rule because of its simplicity, followed by the weighted average rule. Four studies utilised reliability-based methods. Two studies employed stacking, a trainable MCS approach.

Nalić et al. [2020] propose a hybrid ensemble model that incorporates insights from previous research and outperforms standard methods. In the first phase, the authors apply a novel voting system, *if_any*, that demonstrated superior performance compared to all other voting methods, i.e., unanimous and simple hard voting. The method entails using an adjusted version of unanimous majority voting to fuse the outputs of the feature selection algorithms. In the second phase, generalized linear model (GLM), SVM, naive Bayes (NB) and DT were combined using soft voting to form MCSs. The study shows that the MCS comprising of GLM and DT performed better in terms of predictive accuracy (ACC), type I error, F-measure and sensitivity than the other MCSs and individual classifiers. Furthermore, because the MCS uses transparent classifiers as base models and a comprehensible voting system, it is understandable or explainable which makes it suitable to be used for credit scoring purposes. The experiment was conducted on a real-life dataset, consisting of client personal, demographic and credit history data, of a microfinance institution based in Bosnia and Herzegovina.

Anil Kumar et al. [2022] propose an MCS in which LR, k-nearest neighbour (KNN), DT, RF, NB and SVM are used as the base classifiers for the ensemble aggregation. Their study applies stacking in two phases, firstly in the process of training the base classifiers. The outputs of these classifiers are called meta-features because they serve as inputs to the ensemble. Secondly, another set of classifers, specifically three LR, RF and SVM are applied to the meta-features. This second set of classifiers are called meta-classifiers. Majority voting is used to construct the final super classifier. Their study is conducted on the German and Australian datasets from the UCI repository of machine-learning databases. In addition, their ensemble approach outperforms the base classifiers on ACC and AUC.

Runchi et al. [2023] present an MCS, in which data imbalance is taken into account using a heterogeneous balancing approach. Different imbalance ratios are applied to the synthetic minority oversampling technique and edited nearest neighbour balancing algorithm to generate several sub-training datasets. Their ensemble, logistic-BWE (balancing weight effects), involves training multiple LR classifiers on the different sub-datasets and a dynamic weighted voting system is used in the final classifier. The study shows that logistic-BWE outperforms several classifiers: LR, Gaussian Bayes, DT, KNN, SVM, back propagation artificial neural network (BPANN), RF, adaboost, gradient boosting decision trees (GBDT), XGBoost, consistently on AUC, geometric mean, sensitivity and F-measure. It shows that the performance superiority of the logistic-BWE model is statistically significant. Their experiments are conducted on several datasets, namely the Australian, German, Chinese personal loan and default of credit card client from the UCI repository of machine-learning databases.

Many studies on multi-classifiers were conducted on the credit datasets from the UCI repository of machine-learning databases. Furthermore, practitioners are experimenting with heterogeneous as opposed to homogeneous MCSs to improve the accuracy of classifiers. Wang et al. [2011] show through experimentation, using the Australian, China and German credit datasets that bagging performs better than boosting across all datasets. Moreover, stacking and bagging DTs yield the overall best results in terms of average ACC as well as type I and II errors.

The empirical studies on conditions under which MCSs produce improved results is still lacking. Zhu et al. [2001] present a study on the conditions under which the classifiers can be combined to produce improved results. They investigate two criteria, i.e., sufficiency and extraneousness, that are required to ensure that a combination of classifiers will outperform individual classifiers. Sufficiency is used to assess the dominance of a classifier's outputs, whereas extraneousness is used to determine if one classifier's outputs yields information that is useful compared to another. In order for the combination of two classifiers A and B to outperform the individual classifiers, one must dominate, i.e., A must dominate B, and the other B must not be extraneous to the combination. While the work of Zhu et al. [2001] is derived from principles of forecasting, an important finding of the study is that one can construct a single superior classifier by combining the results of individual classifiers, provided that the conditions of sufficiency and extraneousness are satisfied.

3.2 Related work on explainability of classifiers

Some classification techniques, such as ANNs and MCSs have flexible model structures, can analyse enormous amounts of unstructured data, and produce accurate predictions. A common problem regarding these methods is that often they are not transparent, explainable or interpretable, meaning the behaviour and predictions of these systems are not easily understandable to humans, hence they are termed black box models. Furthermore, when these black box models are employed for making decisions, bias that is rooted in datasets that are skewed, inappropriate models, poor formulation of algorithms, or human stereotypes can result in subpar predictions and decisions that are not fair, causing financial and possibly reputational losses [van Giffen et al., 2022]. Therefore, it is crucial that the behaviour of credit scoring models be understood, inputs that might lead to biases be handled appropriately, and learning algorithms be well constructed.

While practitioners are cautious of potential pitfalls and risks associated with black box models, there are socio-economic benefits. Sadok et al. [2022] point out that at the macroeconomic level, the use of artificial intelligence (AI) can contribute positively to economic growth by improving access to credit for traditionally undeserved borrowers. However, Sadok et al. [2022] also caution against the use of AI in credit analysis processes, due to the possible presence of biases and ethical, legal, and regulatory problems. New financial regulations introducing the certification of AI algorithms and of data used by banks is therefore required. Sadok et al. [2022] also point out that AI methods may provide negligible or marginal improvements in predictive power. However, the biggest benefit is that they can be used to model unconventional data from different sources with ease.

There are domains in which models are legally required to be understood and decisions must be explained, such as in retail and business lending institutions [Dastile et al., 2020; Visani et al., 2022]. For this reason, there is ongoing research on methods that seek to make advanced models understandable to remove the black box perception around machine learning techniques, and to establish a model framework that meets legal and regulatory requirements.

3.2.1 What is explainability?

XAI, also referred to as explainable machine learning (XML), is a field of research that seeks to provide insights as to how and why advanced models produce predictions without compromising the performance levels of the models [Markus et al., 2021]. This is an active field of study that aims to overcome the drawbacks of adopting advanced methods. In various studies on XAI the terminology used is inconsistent, may cause confusion, and therefore creates a stumbling block for an agreeable and adoptable framework. Rudin et al. [2022] point out that there is vast and confusing literature on interpretability and explainability. Much literature on explainability confuses it with interpretability or comprehensibility, obscuring the arguments (and thus reducing their precision) and failing to convey the relative importance and practical applications of the two topics.

Gilpin et al. [2018] and Markus et al. [2021], make a distinction between explainability and interpretability as they aim to provide a nomenclature that is clear. A task model is said to be explainable if it is intrinsically interpretable or if it can be complemented by post-hoc explanation that accurately describes the task model and is understandable to a human. An explanation is said to be interpretable if it satisfies two criteria, clarity and parsimony, i.e., the explanation of the task model provides a rationale that is consistent for similar cases and is presented in a compact form. Furthermore, an explanation is said to be faithful or accurately describes a task model if it satisfies the completeness and soundness criteria, i.e., it provides sufficient information to compute the output for a given input and is truthful to the task model. The terms faithful and fidelity are used interchangeably in literature. Figure 4 depicts the definitions of terms related to explainability proposed by Markus et al. [2021].

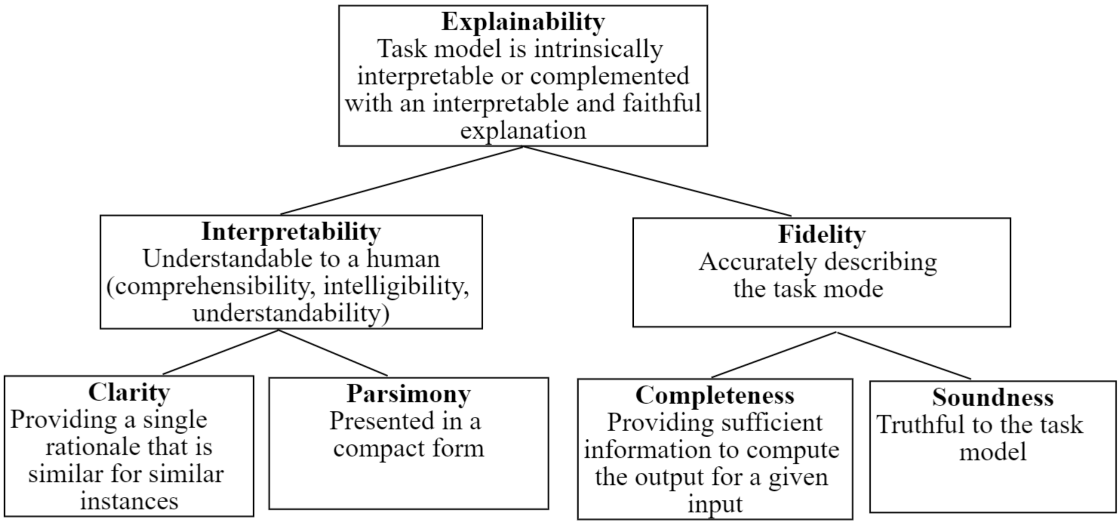

Figure 4: Definitions for terms related to explainability proposed by Markus et al. [2021]

3.2.2 Explainable AI methods

There are various XAI methods described in the literature and often there is an overlap between methods, however each method seems to address different questions. Markus et al. [2021] state that, one approach to accomplish XAI is to utilise models that are deemed transparent or intrinsically explainable. Alternatively, post-hoc explanations can be used to complement the model to make it explainable. Furthermore, Markus et al. [2021] classify explanations into three types, namely, model-based explanations, attribution-based explanations and example-based explanations. Model-based explanations encompass all methods in which an explainable model or a more interpretable surrogate model is created for post-hoc explanations. The class of interpretable models include, sparse linear classifiers, general additive models, rule-based learners, DTs and example based learners (e.g. KNN). Attribution methods, also called feature or variable importance, relevance, or influence methods, provide a measure of the explanatory power of features. Example-based methods explain the task model by selecting instances from the dataset or creating new instances by taking those that are predicted accurately and those that are inaccurate, identifying instances that have an impact on model parameters and creating counterfactual explanations.

In addition, post-hoc explainability can be classified into model-specific or model-agnostic classes and be further subdivided into local and global explanations. Predictions of a model for a large sample of data may be explained using either local (individual) instance explanations or global model interpretation techniques. Local explanations explain why a data point was predicted or not, by segmenting the solution space and giving explanations to a less complex solution subspace, while global explanations explain how attributes influence a decision's behaviour overall. This is useful for examining the fairness of model predictions for choices in a specific data group [Demertzis et al., 2023; Barredo Arrieta et al., 2020]. In some literature, model-specific or model-agnostic techniques are also categorised into explanation by simplification, explanation by feature relevance, visual explanation and local explanation [Saranya and Subhashini, 2023]. Explanation by simplification encompasses techniques in which a whole new system or surrogate is rebuilt based on the trained model to be explained. Feature relevance clarifies the inner functioning of a model by quantifying the impact that a feature has upon the output of the model. Visual explanation covers explainability methods that provide a visualisation of the results [Barredo Arrieta et al., 2020].

3.2.3 Challenges with explainable AI methods

Saeed and Omlin [2023] point out various challenges with respect to the current XAI methods. Scalability can be an issue with local methods, such as LIME, when there is a huge number of cases for which predictions and explanations are needed. Similarly, SHAP can be costly when all combinations of variables must be considered when there are lots of variables to be analysed. Correlation of variables can also cause problems when analysing feature dependence and attribution. Saeed and Omlin [2023] also state that model-based explanations pose a challenge when they cannot predict with reasonable accuracy as practitioners may resort to more accurate models.

In addition, XAI methods must be applied with caution because there is no method that allows for unequivocal, consistent and reliable explanations of machine learning models. Their consistency and reliability are still a discussion topic. Visani et al. [2022] propose two complementary indices, namely coefficients stability index (CSI) and variables stability index (VSI) to measure LIME stability. The CSI assesses whether the coefficients generated by the same variable for different LIME outputs are similar. VSI is used to determine whether different calls of LIME return the same variables. The CSI and VSI give useful information about the consistency of the trained LIME method. In addition, they help understand whether LIME is likely to produce different output at the next call. The CSI and VSI analysis provides a framework that improves trust in LIME as a reliable explanation method [Visani et al., 2022].

3.2.4 Proposed explainability frameworks

The gap between XAI and legal requirements creates a problem for the implementation of transparency, explainability, and interpretability of some classification models. In light of advancements in the utilisation of black box models, there is a need to close the gap between their usage, regulatory and legal requirements.

A study by Bücker et al. [2022] demonstrates that a level of interpretability can be achieved without compromising the predictive power of machine learning techniques. In their study, they propose a systematic model exploration process focused on transparency, auditability and explainability for credit scoring (TAX4CS). Figure 5 shows a schematic representation of the framework proposed by Bücker et al. [2022]. The initial stage is to identify the internal and external stakeholders. Stakeholders include model developers, auditors and regulators as well as bank customers. The second stage is to define the model life cycle, which encompasses the development, validation and production of the model. At every stage the relevant stakeholders are involved in the decisions. The third stage is to recognise the specific needs of the stakeholders. These needs must be aligned with regulatory requirements. Credit officers or managers must comprehend the main features behind credit decisions. Auditors must be able to establish mechanisms to ensure accountability and fairness at every stage of the development process and proper oversight mechanism must be made available to meet regulatory requirements. The fourth stage in the process applies XAI methods and involves exploration at a model-level and local-level. This exploration commences with metrics for assessing the performance of the model and drilling down into examining variable importance (attribution) and effects.

Bücker et al. [2022] also provide an overview of model-agnostic measurements and methods that may be used on any black box model, for each step in the procedure. The proposed framework can be used as a guide to ensure that the necessary level of explainability is attained in fields like credit scoring where explainability is required.

In order to attain an agreeable framework, a consensus of definitions and principles on interpretability must be reached. Principles must be developed on when and how advanced classifiers can be used. Rudin [2019] and Rudin et al. [2022] provide the following principles for interpretability of models:

- Machine learning models must adhere to a domain-specific set of constraints to aid with interpretability.

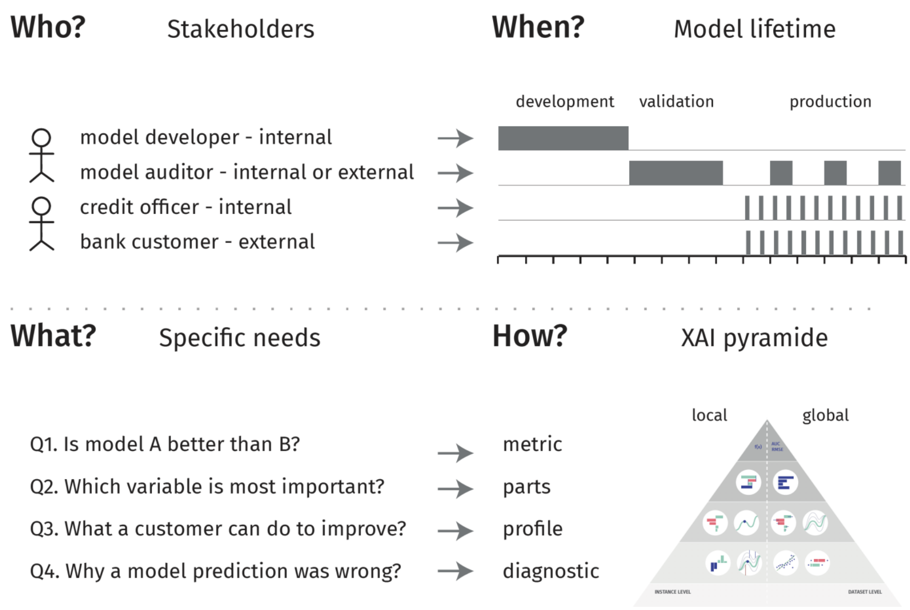

Figure 5: Transparency, auditability and explainability framework proposed by Bücker et al. [2022]

- Interpretable models allow decisions of trust, rather than trust itself.
- In general, the notion of incongruity between interpretability and accuracy is false.
- Metrics for performance and interpretability must be improved through an iterative process.
- Interpretable models should be used for high stakes decisions, if possible, as opposed to explaining black box models.

According to the research and proposed principles by Rudin [2019] and Rudin et al. [2022] there is no accuracy-interpretability trade-off. Furthermore, they propose utilising an interpretable algorithm if the performance is not significantly different. An interpretable model should always serve as a benchmark for model comparison.

There is a need to investigate other strategies that can help practitioners and model users. The value of feedback from stakeholders and subject matter experts is emphasised throughout the studies reviewed. Dastile et al. [2020] present a study on interpretable and black box models and a framework for the interpretability of machine learning models. They propose the rationalisation of predictions, which is a justification of predictions by experts. This approach can be used in addition to the existing local or global model-specific or model-agnostic methods that attempt to make these models understandable.

3.3 Summary

The research on credit scoring techniques indicates that there is no single superior approach to scoring. Furthermore, techniques that are used are problem and data specific. A wide range of methods can be used from individual models as well as hybrid techniques. Wang et al. [2011] point out the need for more experimentation on larger datasets to confirm that MCSs can improve individual base learners substantially when used for credit scoring.

Furthermore, the notion that black box classifiers outperform transparent classifiers is not always correct, which means that the accuracy-explainability trade-off may not always hold. Transparent models must be used as benchmarks to determine if the opaque (black box models) are worth using. In addition, current methods such as SHAP and LIME, utilised for transparency and explainability must be used with caution and tests must be conducted to instil confidence in the explainability and reliability of predictions made. Lastly, a model framework that meets legal and regulatory requirements must be developed and agreed upon to allow for the adoption of black box methods in disciplines where explainability is a requirement.

CHAPTER 4

RESEARCH METHODOLOGY

The purpose of the study is to explore the accuracy-explainability trade-off on classification techniques used for credit scoring. It investigates the perception that black box models outperform transparent models. The study examines the effectiveness of classification models, including DT, LR, LDA, SVM, RF, bagging, LGBM and ANN at predicting credit default risk. It also examines methods utilised to make the predictions of these classification models understandable and explainable. Past research focused primarily on the accuracy of classification methods, comparing black box models to models commonly used in credit risk, such as LR. Recent studies focus on the explainability of black box methods.

This chapter discusses the research methodology used to carry out this study. Section 4.1 describes the Python application and packages used to conduct the experiments described in Chapter 2 and 3 as well as this chapter. The phases of data wrangling and analysis, including data extraction, data assessment, and exploratory data analysis, are discussed in Section 4.2. Section 4.3 discusses the data partitioning. The data preprocessing techniques, i.e., missing value imputation, outlier treatment, feature transformations and engineering are presented in Section 4.4. Section 4.5 discusses a mixed approach to selecting the top features on which to construct the model. The classification methods as well as performance metrics are presented in Section 4.6. The chapter concludes with Section 4.7, in which the methods of interpretability and explainability are discussed. An outline of the research methodology is illustrated in Figure 6.

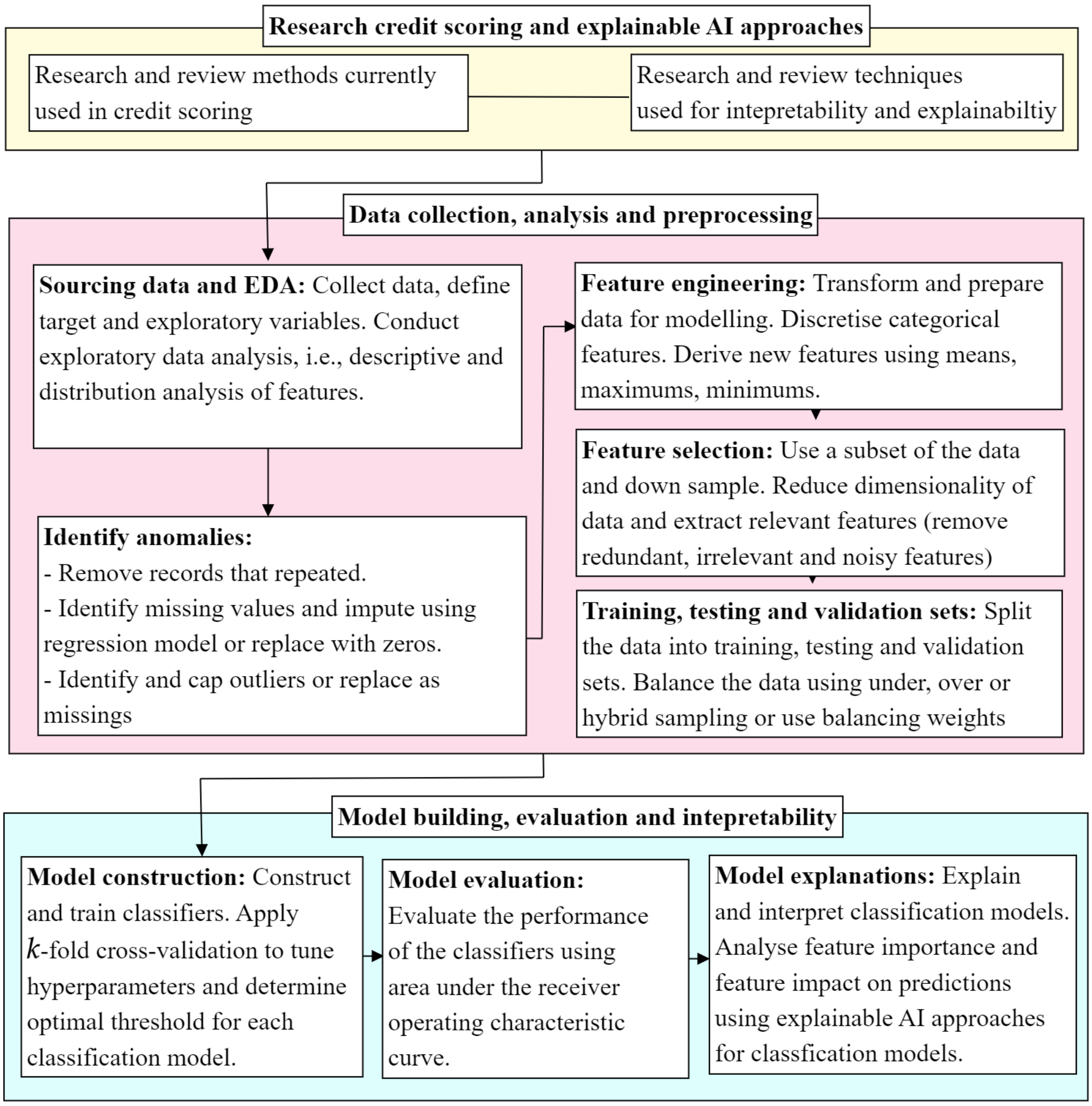

Figure 6: An outline of the research methodology.

4.1 Python application

The experiments for the study, namely, the data wrangling, exploratory data analysis (EDA), feature transformations and extractions, classification model training, performance evaluation and explainability were conducted using Python. Python is an interpreted, object-oriented, high-level programming language that supports modules and packages. The project mainly used the following packages: pandas, numpy and scikit-learn [Pedregosa et al., 2011]. Pandas is used for the manipulation of structured data. Numpy is used for basic numerical operations and matrix operations. Scikit-learn is a Python library integrating several predictive modelling techniques. For data visualisation, the seaborn and matplotlib Python packages were used.

4.2 Data wrangling and analysis

Data wrangling and analysis are essential processes in the development of accurate predictive models, as they inform the techniques to be applied when preprocessing data. The term data wrangling comprises the methods for obtaining raw data and assessing it for the development of classification models.

4.2.1 Data sources and assessment

The data used in this study are publicly available. They contain credit application and default related information on customers. According to Finlay [2010], all consumer datasets contain errors, inconsistencies, and omissions. This could result in a flawed model development training sample, which would make it difficult to determine the relationship between features and modelling objectives. In this study, the data was evaluated in terms of the number of rows and columns, data types, missing values, outliers and duplicates to identify and address anomalies prior to the construction of classifiers.

4.2.2 Exploratory data analysis

EDA refers to the process of evaluating and summarising data in an effort to identify and characterise patterns in the data. The primary goal of this process is to understand the data. In order to identify trends, a variety of statistical methods and graphical representations are used. These methods include univariate reports, distribution summaries, bar charts, heat maps and correlation matrices to understand associations between features.

Despite the fact that graphical representations are often employed in the EDA, one of their main limitations is their inability to show more than two or three aspects of a feature in a single graph. Some of the drawbacks of graphical representations were avoided using a univariate analysis tabular report. The univariate analysis tabular report was used to show the strength of the association between each feature and the target. The measures for degree of association between the feature and target include Gini, chi-square (χ^2) and information value (IV). The IV can be any value from zero to infinity, but common values range from 0 and 1. An IV that is less than 0.05 indicates a weak relationship between the feature and the target, suggesting that the feature is less likely to be predictive. An IV that is between 0.05 and 0.25 signifies a moderate relationship, and values equal to or greater than 0.25 show a fairly strong association [Finlay, 2010].

4.3 Data partitioning

Each dataset was partitioned into three subsets, namely training, testing and validation datasets, using stratified random sampling where the strata was the target variable. The training dataset was used for training, tuning and configuring the classification models. The testing dataset was used for assessing and improving the classification models. The validation dataset was to determine how well the model performs on new data.

4.4 Data preprocessing

Data preprocessing encompasses the methods of transforming, engineering and encoding features so that the data can be used to build effective classification models. It includes implementing techniques to handle missing values, outliers and anomalous data as well removing inconsistencies observed in the data.

4.4.1 Feature transformations and engineering

Features could have missing values if qualitative and quantitative data are not collected, leaving a field empty. The mode can be used to impute missing values for categorical data, and the average or median can be used for numerical data. Depending on the size of the population impacted, entire observations with missing values can also be eliminated. Various techniques may also be used to predict missing values. In this study two approaches are used to impute the missing values. Missing values were either replaced with zeros or an XGBoost regression model was used to impute missing values for features that were deemed predictive.

Outliers can have a negative impact on the model as they introduce bias into the data resulting in under or over-estimates [Kwak and Kim, 2017]. Values that skew the data are treated by either removing the value, capping or removing the entire observations depending on the size of the population affected. The remedial actions for outliers depends on EDA process.

Feature engineering entails the creation of features using domain knowledge and logic to enhance machine learning algorithms. It involves deriving new features, calculating ratios and aggregating existing features using averages, minimums, and maximums, with the aim of introducing new features that may be more predictive than the original features.

4.4.2 Encoding categorical variables

Many machine learning algorithms in the Python scikit-learn library cannot handle qualitative categorical variables. Several encoding techniques, including label encoding, one hot encoding, dummy encoding, and response encoding, can be used to transform these variables into quantitative data. In label encoding the values of a categorical variable are given a distinct integer value [Hancock and Khoshgoftaar, 2020]. In one hot encoding and dummy encoding, a new binary variable is added for each value to indicate the inclusion or exclusion of a value. Furthermore, if a categorical variable has n values, one hot encoding creates n binary variables for each value, whereas dummy encoding creates $n-1$ binary variables. Response encoding involves computing the posterior probabilities of the classes of a given the input of a categorical feature. Response encoding was used in order to keep the dimensions of the data minimal.

4.4.3 Feature scaling

Feature scaling involves the transformation of the values of features so that they lie on a similar scale. The purpose of feature scaling is to reduce the impact of extreme values on algorithms and classification models that are sensitive to such extreme values. Two methods were used to scale features, i.e., standardisation and normalisation.

Standardisation of a feature is obtained by using the formula

$$\hat{x}_i = \frac{x_i - \mu_i}{\sigma_i}, \tag{15}$$

where μ_i and σ_i are the mean and standard deviation of the feature x_i, respectively. Standardisation is commonly used where the data is assumed to follow a normal distribution.

Normalisation of a feature is obtained by using the formula

$$\hat{x}_i = \frac{x_i - x_{i,min}}{x_{i,max} - x_{i,min}}, \tag{16}$$

where $\hat{x}_i$ is a feature in the dataset, $x_{i,min}$ and $x_{i,max}$ are minimum and maximum values of the feature x_i, respectively. Normalisation is mainly used for distance-based algorithms such as SVM.

4.5 Feature selection

Feature selection is the process of selecting a subset of features that have a significant degree of correlation with the target for inclusion in model construction and excluding those that are deemed redundant or unnecessary. It is intended to optimise the learning algorithm so that it works faster and is more efficient. Furthermore, it is intended to improve the performance metrics of the learning algorithm [Oreski and Oreski, 2014; Zhu et al., 2018]. This section describes the steps taken to reduce the dimensions of the data.

The methods used to select features can have a bearing on the accuracy of predictions of a scoring model. Trivedi [2020] presents a detailed study on selection techniques such as information-gain, gain-ratio and χ^2. The study shows that the choice of the selection technique can improve the scoring model. To choose a subset of pertinent features, many statistical techniques can be used, such as low variance, correlation between variables or multicollinearity, filtering and wrapper methods. A combination of the aforementioned techniques was employed to select features using the training subset. Furthermore, the training subset was downsampled, i.e., balanced such that classes are almost equal by reducing the number of observations of the majority class, for the feature selection process. This was done in order to decrease the execution time of the methods used to select features.

4.5.1 Low variance features

Low variance features are constant, approximately constant or quasi-constant across all samples and therefore do not improve model performance. A minimum variance threshold or count of unique values can be used to identify and remove features with a low variance from the dataset. The Python `VarianceThreshold` package can be used to determine the variance of features and remove those with a variance of zero. A count of unique values was used to identify and remove features with unique values less than or equal to one for this research project.

4.5.2 Filter methods

Filter methods select features based on a measure of correlation regardless of the employed modelling algorithm. Additionally, filtering techniques that rank or assess a single feature are known as univariate filters, whereas multivariate filters assess entire feature subsets. Numerous filtering techniques are discussed in the literature and are frequently categorised into information, distance, consistency, similarity, and statistical measurements [Jović et al., 2015].

The common filter methods, filter class and applicable task, whether they are used for classification, clustering or regression and search strategies are discussed in the study by Jović et al. [2015]. Numerous studies show that there is not a single method that outperforms the other and each one depends on the specific task and use case. Also the data type (numeric or categorical) of features that are assessed must be taken into consideration.

In this study, the features were normalised and the χ^2 and Kendall's tau correlation coefficients were utilised for the initial feature selection. Croux and Dehon [2010] present a study on Kendall and Spearman correlation measures. Their literature study suggests that both measures can handle outliers. Furthermore, Kendall's tau is more robust and slightly more efficient than Spearman's rank correlation. The Python `scipy` package is used to compute the Kendall's tau correlation.

4.5.3 Multicollinear features

Collinearity is a linear association between two predictors. Multicollinearity refers to the relationship between two or more predictors that is primarily linear. Multicollinearity is often indicated by an absolute correlation coefficient greater than 0.7 between two or more predictors.

Multicollinearity may result in an algorithm performing poorly. It causes redundancy, meaning that two predictors can provide the same information about the response variable, making the predictors' coefficients inaccurate. It may also cause overfitting, in which case the models perform well on the training dataset but poorly on a testing dataset. Daoud [2017] presents the problems associated with multicollinearity and the use of variance inflation factor (VIF) to quantify the degree of association between features. VIF provides the strength of the correlation between the various independent features. This research uses VIF to identify and reduce multicollinearity. The VIF function from Python `statsmodels` package was used to identify and remove features with VIF above five. A VIF of less than three, indicates low correlation among variables under ideal conditions. A cutoff value of five is commonly used to determine features with high multicollinearity. VIF was applied on a subset of features, i.e., after selecting features using the filter methods, since it is a computationally demanding process.

4.5.4 Wrapper methods

Wrapper methods evaluate and select features based on the classifier performance. It has been shown that wrappers often select subsets of features that are better than those selected by filters because the subsets are evaluated using a real modelling algorithm [Jović et al., 2015]. Rodriguez-Galiano et al. [2018] demonstrate that, despite increased computational requirements, wrapper methods can effectively aid in the selection of the most influential features, improvement of the prediction model and reduction of the dimensionality of the feature space. Moreover, a wrapper composed of a RF learner and a sequential forward feature selection (SFFS) searching strategy performed better than other methods, exhibiting the best accuracy and interpretability.

In this research, the features were normalised and the recursive feature elimination (RFE) wrapper was utilised to select the final features, from features remaining after filtering and removing multicollinear features in the training dataset. RFE seeks to find a subset of features by iteratively removing one feature at a time until the desired number of features is achieved. This involves fitting the predictive model using an initial subset of features, ranking the features according to relevance, removing the least important features, and repeating this process on the remaining features until the specified number of features is obtained.

4.6 Classification methods

The Python scikit-learn library was used to construct and train the LR, LDA, DT, SVM, ANN, bagging, RF and LGBM classification methods, explained in Section 2.1. Furthermore, the features used for LR, LDA, DT, ANN, RF and LGBM were scaled using standardisation, whereas the features used for SVM and bagging were scaled using normalisation (see Section 4.4.3).

Cross-validation was used to train and test the models. This is a resampling procedure used to evaluate the machine learning models in the training phase. Furthermore, random hyperparameter tuning was applied on each classification method to obtain the best performing classification model.

4.6.1 Class imbalance

Credit default risk data tends to be imbalanced, meaning the target is in favour of one class over the other or that the number of data points for a certain class are significantly more. This creates a risk of misclassification since classifiers trained on imbalanced datasets may classify all minority data with majority labels and still produce a high performance measure of accuracy. Kuhn and Johnson [2013] present a detailed study on the impact of imbalanced classes on model development as well as remedies for severe class imbalance in data.

There are numerous balancing approaches that are commonly used in practice and presented in literature to reduce this risk of misclassification. The remedies to handle the risk of misclassification include upsampling, downsampling, as well as using class weights and penalties on the classification methods. The downsampling method involves reducing or eliminating samples from the majority class until there is no substantial difference between the minority and majority classes. Although this method is widely used, caution must be exercised to prevent information loss. Upsampling entails increasing the representation of the minority class examples until there is no substantial difference between the minority and majority classes. This is achieved by either duplicating examples of the minority class or creating synthetic examples using the synthetic minority oversampling technique (SMOTE) [Rendón et al., 2020]. In this study, the balanced class weights built into the Scikit-learn library classification models were used to remedy the effects of the imbalance for each model.

4.6.2 Performance tuning

The k-fold cross-validation, where $k = 4$, was used to configure the classification models. This involved splitting the data into k subsets of equal size as shown in Figure 7. The parameter k refers to the number of groups or folds that the data will be split into. The first fold is treated as a validation set, and the model is fit on the remaining $k-1$ folds. The `RepeatedKFold` and `KFold` Python functions were used to conduct cross-validation.

In addition, cross-validation was used to fine tune the inputs or configurations that are used to control the learning process of the models. The inputs that are configured in the learning or training phase of the model construction are referred to as hyperparameters. A k-fold cross-validation and random search hyperparameter tuning technique were used to determine optimal hyperparameters for each classification model.

Lastly, k-fold cross-validation was used to determine the parameters for the best classification model, which is then used to determine the optimal thresholds to determine classes from the probabilities. The optimal threshold is the maximum distance between the point on the ROC curve and the random line, explained in Section 2.3. The distance between the ROC curve and the random line is referred to as the Youden's J-Statistic or J-Statistic.

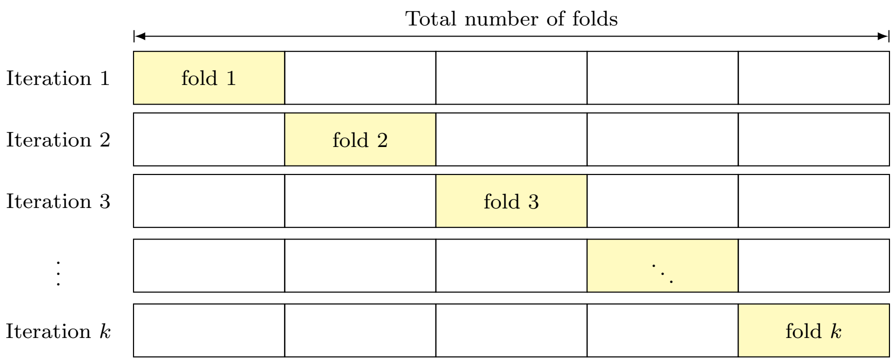

Figure 7: k-fold cross-validation on training dataset

4.6.3 Performance assessment

The classification models were applied to 30 random subsets of data in order to compare the performance in terms of AUC. The `scipy.stats`, `pingouin`, `scikit_posthocs` Python libraries were used to conduct the ANOVA test, the Kruskal Wallis test and Dunn's multi-comparison test, respectively. These tests provide a way to rank the performance of the classifiers and to determine if the difference in performance is statistically significant.

4.7 Explainability and interpretability

The `sklearn`, `shap`, `lime` and `lime.lime_tabular` Python libraries were used to analyse feature contributions and effects in an effort to interpret and explain the classification models. The `shap` package has various methods, which incudes the `KernelExplainer` and `TreeExplainer`. The `KernelExplainer` was utilised for the linear models, which include LR, LDA and SVM. A subset of 6000 observations of the validation data was used, given that `KernelExplainer` takes a long time to process data. The more the observations, the longer it takes. The remaining models were analysed using `TreeExplainer`, since it does not support linear models. Given the effectiveness of `TreeExplainer`, the full validation data subset was used.

4.8 Summary

The methodology provides details of the steps followed to construct the credit scoring classifiers as well as the approaches to explain these classifiers. The experiments were conducted using Python, which was used to analyse data, select features, train classification models and analyse the outcomes. Data analysis is essential for understanding patterns and relationships in the data. It is essential to identify and treat anomalies such as missing values and outliers. Prior to selecting features for modelling and training classifiers, categorical features were encoded and the numerical features were scaled to minimise the adverse effects of different scales and outliers. A number of approaches were applied to identify predictive features and to ensure that the final features selected for training classifiers were not correlated. The VIF was used to identify correlated features and to remove those with a high VIF value. Filter methods, which are model independent methods, were used to identify predictive features. In addition, wrapper methods, which select features based on classifier performance, were also used to select features. The classifiers were trained by tuning hyperparameters and balancing classes. Furthermore, SHAP and LIME were used to explain the outcomes of the classifiers.

CHAPTER 5

DATA ANALYSIS AND PREPROCESSING

This chapter discusses aspects of the data preparation process required for the construction of effective predictive models for case study 1 and 2, i.e., credit card default and home credit default datasets. The data sources, ethical considerations and wrangling are presented. In addition, the exploratory data analysis and preprocessing (transformations and scaling) steps are discussed.

5.1 Case study 1: Credit card default data

The credit card default data is secondary data sourced from the UCI Machine Learning Repository website submitted by Yeh [2016]. The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the development and analysis of machine learning algorithms. This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. This permits the distribution and modification of the datasets for any purpose, under the condition that proper credit is given.

The credit card default data contains 30 000 observations and 25 features. Furthermore, it includes the TARGET, which is a dichotomous response variable where the value zero indicates that the loan was repaid (non-default) and one indicates the loan was not repaid (default). The categorical columns were already encoded. Based on the description of the dataset, it does not contain missing values and duplicates. Therefore, this data was not processed following the full data processing steps described in Section 4.4. Furthermore, the credit card default dataset was partitioned into subsets of sizes 50%, 30% and 20% for training, testing and validation, respectively. The proportions of the partitions are to ensure that there are sufficient volumes in each subset. A low number of observations can result in model instability. The `train_test_split` function from the python Scikit-learn library was used to ensure that the distribution of the targets are representative of the original dataset.

5.2 Case study 2: Home credit default data

The second credit risk data is secondary data sourced from the Kaggle website submitted by Home Credit Group [Home Credit Group, 2018a]. Kaggle is an online hub that hosts data science competitions and often provides data to solve real-world problems with an incentive for providing the best solution. Home Credit Group, which is an international non-bank financial institution, submitted information distributed into several relational datasets containing credit information on borrowers for a competition in Kaggle. The objective of the competition was to develop predictive models to estimate the default risk of a given borrower.

Home Credit Group are the sponsors and rights holders of the Home Credit Default Risk competition. The seventh section under the list of rules provided by Home Credit group grants permission for one to utilise the competition data for purposes of the competition and other non-commercial purposes, such as participation on Kaggle website forums, academic research and education [Home Credit Group, 2018b].

5.2.1 Datasets and structure

The Home Credit Group data is distributed into several data frames containing credit information on borrowers. The structure of the relational data frames is depicted schematically in Figure 8, which provides a brief description of the data frames and the features used to connect each data frame.

The main data frames that were submitted by the Home Credit Group are the application_train and application_test. The subsets in these data frames are mutually exclusive and they contain information about each loan application, identified by the feature SK_ID_CURR. In this study, only the application_train data frame was used to train, test and construct the credit scoring models. The application_train contains 307 511 observations and 121 features. Furthermore, it includes the TARGET, which is a dichotomous response variable where the value zero indicates that the loan was repaid (non-default) and one indicates the loan was not repaid (default). Throughout the research, non-default and default are also referred to as good and bad, respectively.

There are two data frames pertaining to previous loans from other financial institutions reported to the credit bureau for each loan applicant in the applications subset. The first data frame is the bureau, which contains 1 716 428 observations and 17

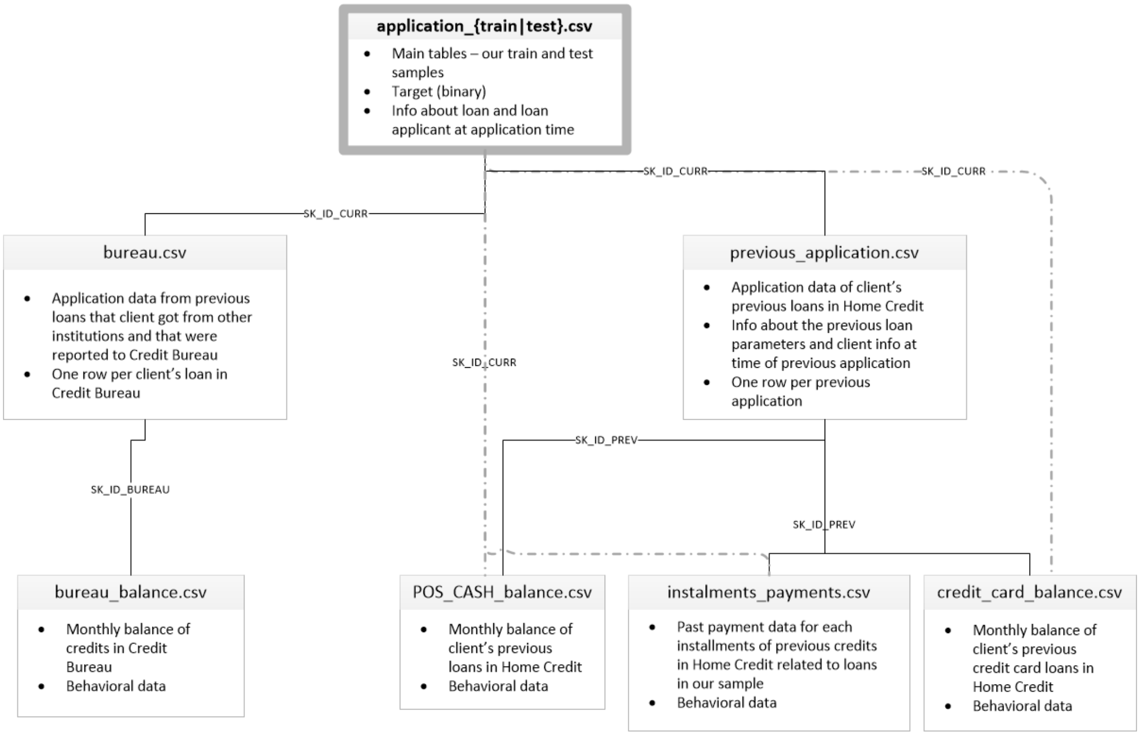

Figure 8: The structure of the relational datasets of the Home Credit competition [Home Credit Group, 2018a].

features. The second is the bureau_balance, which contains 27 299 925 observations and two main features, namely monthly balances and statuses of previous credits. The observations in bureau and bureau_balance are identified by SK_ID_BUREAU. Each loan in the applications data can have multiple previous credits.

There are four data frames, namely the previous_application, POS_CASH_balance, instalments_payments and credit_card_balance, related to previous applications or credits of clients who have loans in the sample of data provided. The previous_-application data frame contains all previous applications for Home Credit loans. Furthermore, each current loan is identified by the SK_ID_PREV feature and it may be linked to multiple previous loans.

The POS_CASH_balance data frame consists of monthly data on previous point of sale and cash loans that the applicants had with the Home Credit Group. Each row in the data frame shows previous credit related to loans in the applications subsets. It contains 10 001 358 observations and eight features.

The credit_card_balance data frame contains monthly data about previous credit cards that the applicant has with the Home Credit Group. Each row in data frame shows the credit card balance for a particular month. Furthermore, a single credit card may have multiple rows.

The instalments_payments data frame comprises the history of payments made for the credits that were previously issued in Home Credit for each applicant. Each row in the data frame reflects a payment that was made, plus one row each for a missed payment.

5.2.2 Data assessment and analysis

The primary objective of this analysis was to obtain a high level overview of the data that would inform the model construction process. Table 1 shows the data assessment and preliminary analysis of the datasets that were used to construct the credit classifiers. A detailed mathematical description overview of the data is presented in Appendix C. The application_train contains 122 variables (121 features and a target variable) and 63% of the features contain missing values. Furthermore, all the datasets excluding installment_payments contain categorical data, which must be encoded. The bureau data has seven features which contain missing values. This study focuses mainly on the application_train datasets for the construction of the classification models. Therefore, the rest of the exploratory data analysis and preprocessing is based on the application_train datasets.

Table 1: The data assessment and preliminary analysis of the home credit default datasets.

	Rows	**Columns**				
Dataset	**No.**	**No.**	**Numeric**	**Categorical**	**Duplicates**	**Missings**
application_train	307511	122	106	16	0	67
bureau	1716428	17	14	3	0	7
bureau_balance	27299925	3	2	1	0	0
credit_card_balance	3840312	23	22	1	0	9
installments_payments	13605401	8	8	0	0	2
previous_application	1670214	37	21	16	0	16
POS_CASH_balance	10001358	8	7	1	0	2

5.2.3 Missing values identification

There are a significant number of columns with a high number of missing values in the application_train. The majority of features with high missing values are related to residential or apartment information. It is expected that these features will be missing if the applicant does not own or rent a property. Figure 9 shows that 41 features contain 50% or more missing values, 16 features have between 10% and 50% missing values and 10 features have less than 10% missing values.

Features with high missing values (above a subjective proportion or threshold) are usually dropped, and those below a certain threshold are imputed. However, dropping features may result in loss of information, therefore it is imperative to understand if these feature have an impact on the models. Features with missing values were kept until the feature selection and modelling phases. Furthermore, various strategies were applied to handle the features with missing values, such as predicting missing values or replacing the missing values with zero. The EXT_SOURCE_1, EXT_SOURCE_2 and EXT_SOURCE_3 features were imputed using XGBoost regression model for predicting, starting with the feature with the least number of missing value columns. Only numeric values were used as input features into the regression model.

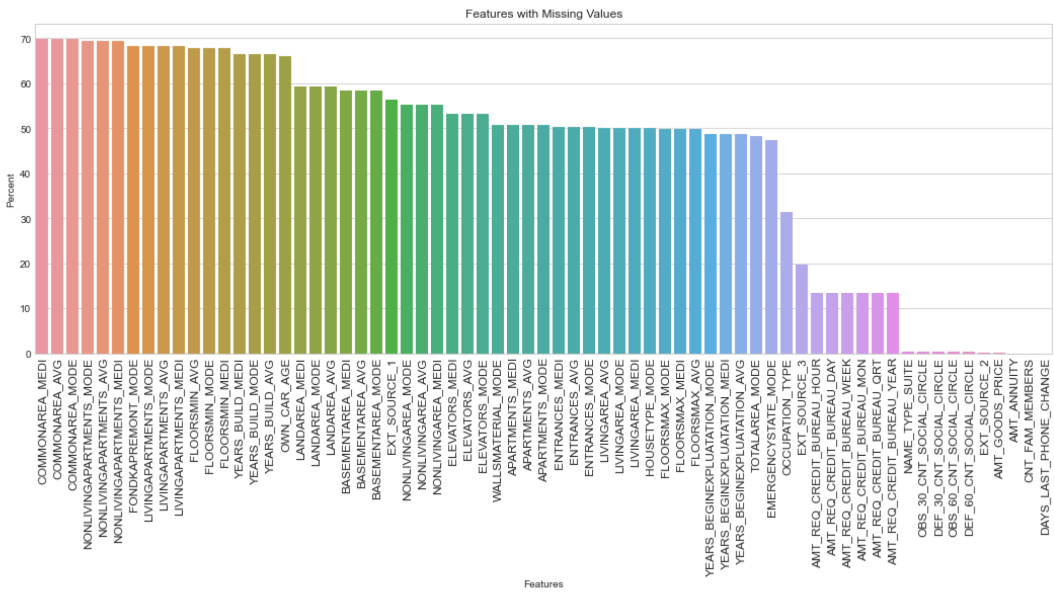

Figure 9: Proportion of missing values for each feature containing missing values in application_train dataset.

5.2.4 Anomalies detection and contradictions

Appendix C provides a statistical description of all the features and shows the distributions, central tendency, quartiles, and extreme values of the numerical features. The analysis shows the presence of anomalies and extreme numbers across all the datasets. Negative values were observed for DAYS_BIRTH. Extreme values are found in DAYS_EMPLOYED, OBS_30_CNT_SOCIAL_CIRCLE and OBS_60_CNT_-SOCIAL_CIRCLE. The DAYS_ BIRTH feature was converted to years and made positive number so that it can be easier to interpret. Erroneous values in some fields such as DAYS_EMPLOYED, OBS_30_CNT_SOCIAL_CIRCLE and OBS_60_CNT_-SOCIAL_CIRCLE were deleted or converted to missings(Nan) and subsequently replaced with 0 for algorithms that cannot handle missing values. There were also four rows with unkown value (XNA) in the Gender feature that were removed. The EXT˙SOURCE features contain missing values and were imputed as described in 5.2.3.

5.2.5 Correlation analysis

The correlation heatmap shows the degree of correlation between the features for the application_train dataset. Highly correlated features can increase the time complexity of the model and increase the complexity of the model interpretation. These highly correlated features are removed, as explained in Section 4.5. Figure 10 shows a high correlation between AMT_GOODS_PRICE and AMT_CREDIT, between DAYS_-EMPLOYED and DAYS_BIRTH as well as the apartments or living area related features.

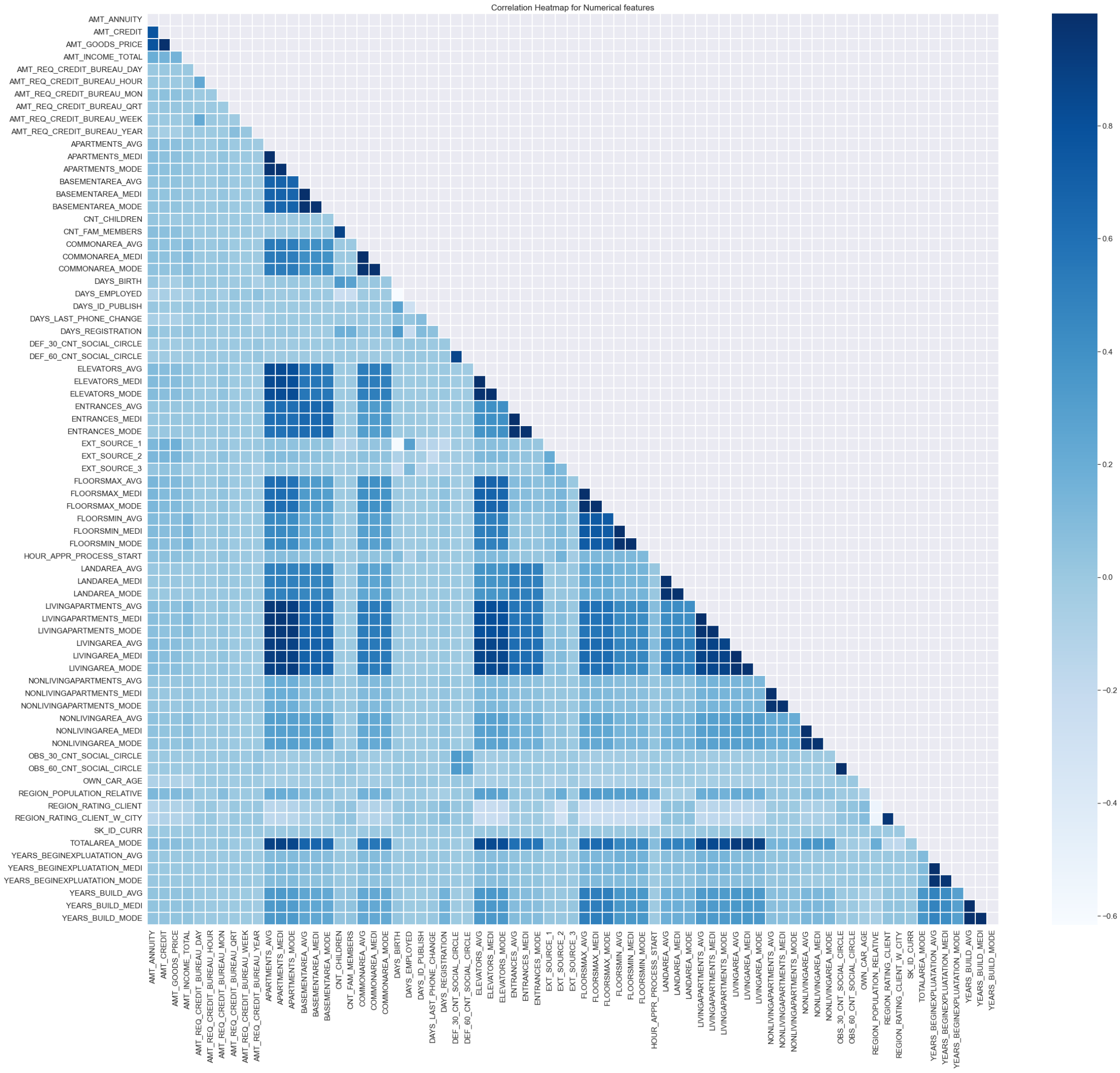

Figure 10: A heatmap of the correlation of each numeric feature with respect to other features in application_train dataset.

5.2.6 Data transformations

Response encoding was used to transform all categorical features into quantitative data because the majority of the algorithms in the Scikit-learn library are unable to handle such features. The categorical features were split into two features (with 1 and 0 suffixes), each of which contains the likelihood that each class label belongs to that category.

5.2.7 Class imbalance analysis

A distribution analysis of the classes indicates that the proportion of defaults (encoded 1) is significantly lower than non-defaults (encoded 0), i.e., the data is highly imbalanced, as shown in Table 2. The low percentage of 8.07% shows that the Home Credit Group is very selective when providing credit and has managed to maintain a low rate of customers that fail to meet their financial obligations or default. Furthermore, when classes are highly imbalanced, some metrics used to measure the performance of the classification models may be misleading. For instance the accuracy (percentage correctly classified) may be misleading in this case because it is biased to the majority class. Other metrics, such as AUC, precision and recall must be applied when assessing the performance of the classification models.

Table 2: The overall class distribution and analysis by loan type.

	Cash loan		Revolving loan		Overall	
Classes	**Total**	**%Total**	**Total**	**%total**	**Total**	**%total**
Non-default (0)	255 011	91.65	27 675	94.52	282 686	91.93
Default (1)	23 221	8.35	1 604	5.48	24 825	8.07
Total	**307 511**	**100.00**	**307 511**	**100.00**	**307 511**	**100.00**

5.2.8 Data partitions

The application_train dataset was partitioned into three subsets made up of 60%, 28% and 12% of the total observations for training, testing and validation respectively. The proportion of subsets is to ensure sufficient volumes in each subset so that the classification models are stable. The `train_test_split` function from the python Scikit-learn library was used to ensure that the distribution of the targets are representative of the original dataset. The imbalance shown by the target distribution may have an adverse effect on the performance of the predictive models and may require additional steps in the construction of the models. In order to optimise the performance of the models, re-sampling, generating synthetic samples, weight class parameters and penalties for some algorithms were considered.

CHAPTER 6

RESEARCH RESULTS AND DISCUSSION

In order to address the research objective, eight classification techniques were constructed and assessed in terms of performance and explainability. The aim being firstly, to examine the effectiveness in terms of accuracy of the transparent and black box models. Secondly, to address the challenges of the explainability of black box techniques in the context of credit default risk predictions.

This chapter presents the results of the study and it is organised as follows: Section 6.1 presents the key hyperparameters that were tuned for optimal performance for each classification model applied to case study 1 and case study 2, i.e., the credit card default dataset and Home-credit default dataset, respectively. Section 6.2 presents the results of the experiments conducted for case study 1. The performance of the classification models as well as pre- and post-explainability modelling results are discussed. In Section 6.3, the results of the experiments conducted for case study 2, are discussed, covering the performance of the classification models as well as pre- and post-explainability modelling results.

6.1 Classifier performance tuning

The classification techniques, namely, ANN, bagging, DT, LDA, LGBM, LR, SVM and, RF discussed in this paper, all required several hyperparameters to be tuned to enhance performance. Given the numerous hyperparameters to be tuned, tuning each one by manual trial and error would be both time consuming and inefficient. Consequently, the hyperparameter optimisation was done with a random search approach. Furthermore, since the data is highly imbalanced, class weights were used to optimise the performance of the classifiers that are influenced by imbalanced classes. Table 3 shows the hyperparameters that were tuned to optimise the performance of the classification models applied to case study 1 and case study 2.

Table 3: Hyperparameters and search spaces for the classifiers applied for case study 1 and case study 2.

Classifier	Hyperparameter	Search Space
ANN	Hidden layers	One layer with 100 nodes and three layers with 120, 80, 40 nodes, respectively
	Activation	Tanh and rectified linear activation function
	Maximum iterations	{10, 20}
Bagging	Number of estimators	{50, 100, 150, . . . , 500}
	Maximum samples	{100, 200, 300, . . . ,1000}
DT	Maximum depth	{1, 2, . . . , 6}
	Maximum leaf nodes	{1, 2, ... , 50}
	Minimum sample per leaf	{1, 100, 200, ... , 1000}
	Class weight	{balanced, none}
LDA	Solver	Single value decomposition (SVD), least squares solution (LSQR) and eigenvalue decomposition (Eigen)
LGBM	Number of leaves	{10, 20, 25, 30, 40, 60, 80, 100}
	Maximum depth	{1, 3, 5, 10, 20}
	Learning rate	{0.01, 0.05, 0.1, 0.2}
	Reg alpha	{0, 0.01, 0.03, 0.05, 0.07}
LR	Class weight	{balanced, none}
	Solver	SAGA, newton-cg, LBFGS, Liblinear
SVM	Class weight	balanced
	Alpha	$10^{-4+i(\frac{9}{49})}$ where $i = 0, 1, \ldots, 49$
RF	Number of estimators	{50, 100}
	Max depth	{6, 9, 12}
	Maximum leaf nodes	{6, 9, 12}

6.2 Case study 1: Credit card default data

This section presents results for the pre- and post-modelling explainability of the classification models applied to case study 1. In pre-modelling explainability, features that served as inputs into the models are described. Post-modelling explainability covers explainability of classification models that are intrinsically explainable or transparent such as LR, LDA, and DT. The post-modelling explainability results for SVM, ANN, bagging, RF, and LGBM achieved using SHAP and LIME are presented.

6.2.1 Pre-modelling explainability

Pre-modelling explainability encompasses methods to understand the data prior to training and applying the classifiers for credit scoring. Pre-modelling explainability can be achieved through univariate analysis of features and quantifying the relationship between features and the target variable. The IV was used to quantify the strength of the relationship between features and target. The results of the univariate analysis for each feature are presented in Appendix A.

Table A.5 shows the analysis of PAY_0, which is the repayment status in September, in relation to the outcome of the loan. The information value of this feature is 0.87, which indicates a strong relationship to the outcome of the loan. PAY_2, defined as repayment status in August, has the second highest IV of 0.54, as shown in Table A.6. Similar analysis was conducted for all features. It is expected that features with a high IV will be deemed as predictive factors in the classification models.

Pre-modelling explainability can also be achieved through explainable feature engineering. The original features were extracted without any modifications from the credit card default dataset and no additional features were derived. This aids in the explainability of features since all the features are defined and computations are explainable and understood. Furthermore, they can be broadly categorised as demographic information, repayment statuses, bill amounts, payment amounts and credit balances. This makes it possible to explain the risk factors or feature contributions towards model predictions.

Given the small size of the feature space, the VIF was used to reduce multicollinearity and eliminate redundant features by excluding those with a VIF above 5. Table 4 shows the 18 features that were selected from the original set of 24 features using VIF.

Table 4: Features selected for case study 1.

Category	**Feature**	**Selected**
Demographics data	SEX	✓
	EDUCATION	✓
	MARRIAGE	✓
	AGE	✓
Repayment statuses	PA_0	✓
	PA_2	✓
	PA_3	✓
	PA_4	✓
	PA_5	✓
	PA_6	✓
Bill statements	BILL_AMT1	✓
	BILL_AMT2	
	BILL_AMT3	
	BILL_AMT4	✓
	BILL_AMT5	
	BILL_AMT6	
Previous payments	PAY_AMT1	✓
	PAY_AMT2	✓
	PAY_AMT3	✓
	PAY_AMT4	✓
	PAY_AMT5	
	PAY_AMT6	✓
	LIMIT˙BAL	✓

6.2.2 Classifier performance tuning

The classification models applied in case study 1 were trained with various hyperparameters. Table 5 lists the hyperparameters that were tuned for each model, as well as optimal values obtained for the search spaces described in Section 6.1. The optimal hyperparameters were obtained using a 5-fold cross-validation random search, repeated 15 times. For each iteration, random samples were extracted for cross-validation and the hyperparameters that produced optimal results were used.

Table 5: Optimal hyperparameters for each classifier for case study 1.

Classifier	Hyperparameter	Optimal value
ANN	Hidden layers	Three layers with 120, 80, 40 nodes, respectively.
	Activation	Tanh
	Maximum iterations	20
Bagging	Number of estimators	5
	Maximum samples	750
DT	Maximum depth	6
	Maximum leaf nodes	43
	Minimum sample per leaf	100
	Class weight	balanced
LDA	Solver	SVD
LGBM	Number of leaves	20
	Maximum depth	3
LR	Class weight	balanced
	Solver	SAGA
RF	Max depth	6
	Maximum leaf nodes	12
SVM	Class weight	balanced
	Alpha	$10^{-4+i(\frac{9}{49})}$ where $i = 7$

6.2.3 Performance evaluation

The performance of each classification model was analysed in terms of AUC. The results were obtained by evaluating the models on 30 randomly generated subsets of data from the validation data. Figure 11 depicts the performance of each classification model in classifying credit card defaults and non-defaults. LGBM achieved the highest average AUC of 76.94%, followed by RF and ANN with average AUCs of 76.85% and 76.32%, respectively. The DT classification model yielded an average AUC of 73.95%. In comparison, bagging, LDA, LR and SVM produced AUCs ranging between 71.18% and 72.21% which are lower than the performance of DT. In this case study, the black box models outperform the transparent models, with the exception of the bagging classifier. This finding suggests that there may be a trade-off between accuracy and explainability.

A further analysis to assess the difference of means was conducted using ANOVA and the Kruskal-Wallis test. However, the p-value on the ANOVA test for normality is less than 0.05. This indicates that data are not normally distributed and therefore ANOVA cannot be used to compare or to draw meaningful conclusions from the means. The Kruskal-Wallis test yields a p-value less than 0.05, which suggests that the means are different. In addition, Dunn's multi-comparison test shows that the average AUCs of ANN, LGBM and RF are not statistically significant since the p-values are greater than 0.05. However, the average AUCs of these classifiers are significantly different compared to those of bagging, DT, LDA, LR and SVM, at a 95% confidence level, as shown in Table 6.

6.2.4 Post-modelling explainability of interpretable models

The DT inherently produces feature rankings since the order of feature splits depends on the discriminatory power of the feature. The sequence of features shown as nodes as well as branches show the relationship between variables. Figure 12 exhibits the first three levels of the DT for case study 1. The PAY_0, and PAY_2 have the highest rank in terms of discriminating between classes. While a decision tree is easier to

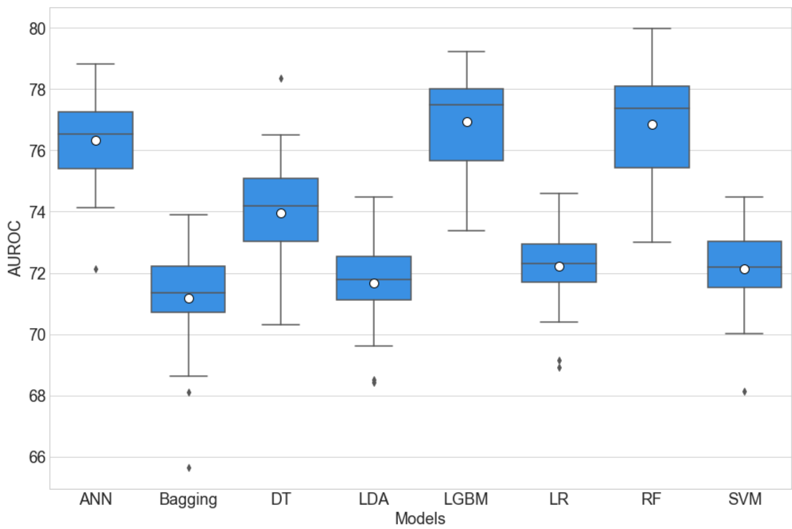

Figure 11: Performance of the classification models for case study 1.

Table 6: Dunn's multi-comparison test of the classification models for case study 1. The average AUCs of ANN, LGBM and RF are significantly different to those of bagging, DT, LDA, LR and SVM since the p-values are less than 0.05.

	AUC	ANN	Bagging	DT	LDA	LGBM	LR	RF	SVM
ANN	76.32	1.00							
Bagging	71.18	**0.00**	1.00						
DT	73.95	**0.02**	**0.00**	1.00					
LDA	71.65	**0.00**	1.00	**0.00**	1.00				
LGBM	76.94	1.00	**0.00**	**0.00**	**0.00**	1.00			
LR	72.21	**0.00**	0.99	0.08	1.00	**0.00**	1.00		
RF	76.85	1.00	**0.00**	**0.00**	**0.00**	1.00	**0.00**	1.00	
SVM	72.14	**0.00**	1.00	0.05	1.00	**0.00**	1.00	**0.00**	1.00

.

interpret because it can be depicted visually, it may be difficult to follow when the size of the tree is large.

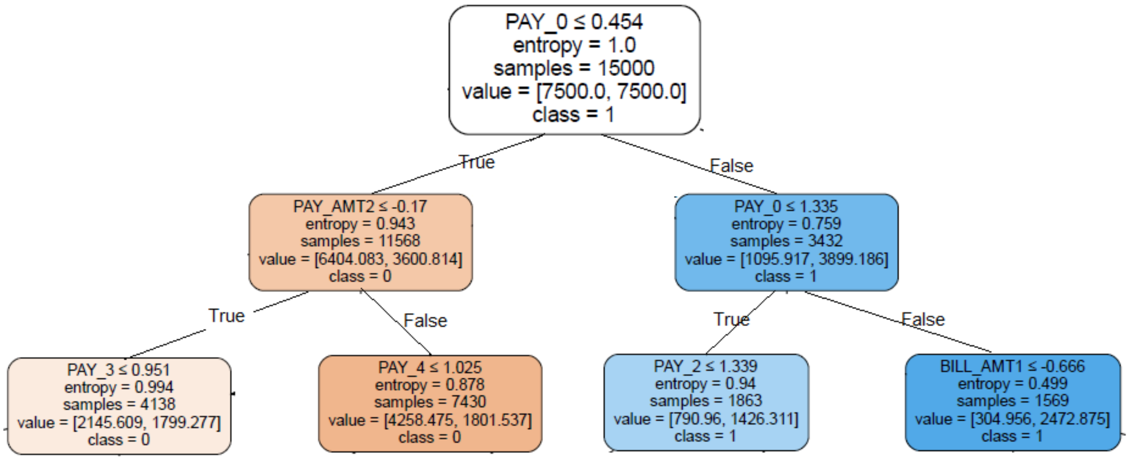

Figure 12: A representation of the DT classifier up to a depth of two for case study 1.

The relative contributions of factors predictive of default were assessed for LR by extracting the coefficients and analysing the statistical significance. Table 7 shows the coefficients, p-values, standard errors, and confidence intervals for each feature for the optimal LR model. The features are ordered in terms of the magnitudes of the contributions to the predictions, by calculating the absolute values of the coefficients and ranking them in descending order. The intercept is used to provide a probability of an outcome when all features are at zero.

The measures of statistical significance and confidence intervals of the LR parameters indicate only 13 features contribute significantly to the model since the p-values are less than 0.05. The p-values for PAY_4, PAY_6, PAY_AMT1, PAY_AMT3, PAY_AMT5, PAY_AMT6 features are higher than 0.05, indicating that those features do not contribute significantly to the scoring models and could be omitted. An added advantage of this approach is that it provides information about features that can be left out of the model without compromising the accuracy.

The measures of statistical significance and confidence intervals of the LDA parameters indicate that only 10 features contribute significantly to this model since the p-values are less than 0.05 as shown in Table 8. The bottom 8 features have p-values higher than 0.05 indicating that the features do not contribute meaningfully to the target and could be excluded from the LDA classification model.

The group means for each feature and each class are depicted in Table 9. The differences in mean values for each feature per class imply that these features have an impact on the classes. Furthermore, the low standard errors and confidence intervals indicate that the mean values are expected to fall within the range of given values at a 95% confidence level. Furthermore, the measures of statistical significance of the LDA parameters for default class indicate that the top 18 features contribute significantly to the model since the p-values are less than 0.05.

Table 7: Feature importance and impacts for the LR classifier for case study 1.

Features	**Coefficients**	**std error**	**z**	**[.025**	**.975]**	$P \geq \|Z\|$
INTERCEPT	-0.19	0.02	-11.79	-0.21	-0.18	0.00
PAY_0	0.52	0.03	17.37	0.49	0.55	0.00
PAY_AMT2	-0.49	0.14	-3.45	-0.63	-0.35	0.00
PAY_AMT4	-0.15	0.05	-3.30	-0.20	-0.11	0.00
PAY_2	0.15	0.04	4.12	0.11	0.18	0.00
LIMIT_BAL	-0.13	0.03	-4.16	-0.16	-0.10	0.00
MARRIAGE	-0.11	0.03	-4.36	-0.14	-0.09	0.00
BILL_AMT1	-0.10	0.05	-1.88	-0.15	-0.05	0.03
EDUCATION	-0.09	0.02	-4.35	-0.11	-0.07	0.00
PAY_3	0.09	0.04	2.04	0.05	0.14	0.02
PAY_AMT1	-0.09	0.06	-1.44	-0.15	-0.03	0.08
SEX	-0.07	0.02	-3.70	-0.09	-0.05	0.00
PAY_4	0.04	0.04	0.99	-0.00	0.08	0.16
AGE	0.04	0.02	1.96	0.02	0.06	0.02
PAY_AMT3	-0.03	0.04	-0.82	-0.08	0.01	0.21
PAY_AMT5	-0.03	0.05	-0.55	-0.08	0.02	0.29
BILL_AMT6	0.01	0.05	0.22	-0.04	0.07	0.41
PAY_AMT6	0.01	0.03	0.19	-0.03	0.04	0.43
PAY_6	-0.00	0.03	-0.08	-0.03	0.03	0.47

Table 8: Feature importance and impacts for the LDA classifier for case study 1.

Features	**Coefficients**	**std error**	**z**	**[.025**	**.975]**	$P \geq \|Z\|$
INTERCEPT	-1.51	0.02	-60.57	-1.53	-1.48	0.00
PAY_0	0.71	0.04	20.06	0.68	0.75	0.00
BILL_AMT1	-0.27	0.04	-6.90	-0.30	-0.23	0.00
PAY_2	0.19	0.05	3.69	0.14	0.24	0.00
PAY_4	0.14	0.05	3.09	0.09	0.19	0.00
EDUCATION	-0.09	0.02	-4.93	-0.11	-0.07	0.00
MARRIAGE	-0.08	0.02	-3.99	-0.10	-0.06	0.00
LIMIT_BAL	-0.08	0.03	-3.19	-0.11	-0.06	0.00
PAY_AMT1	-0.08	0.02	-3.85	-0.10	-0.06	0.00
PAY_AMT5	-0.06	0.02	-2.52	-0.09	-0.04	0.01
AGE	0.06	0.02	2.79	0.04	0.08	0.00
BILL_AMT6	0.05	0.04	1.25	0.01	0.09	0.11
PAY_6	-0.03	0.04	-0.84	-0.07	0.01	0.20
SEX	-0.03	0.02	-1.25	-0.05	-0.01	0.10
PAY_3	0.03	0.05	0.50	-0.03	0.08	0.31
PAY_AMT2	-0.02	0.03	-0.63	-0.05	0.01	0.26
PAY_AMT4	-0.02	0.02	-0.72	-0.04	0.01	0.24
PAY_AMT3	0.01	0.02	0.26	-0.01	0.02	0.40
PAY_AMT6	0.00	0.02	0.06	-0.02	0.02	0.47

Table 9: Post-model analysis of group means for LDA classifier for case study 1.

Features	Non-default class mean						Default class mean					
	Means	**std error**	**z**	**[.025**	**.975]**	**P $\geq$ |Z|**	**Means**	**std error**	**z**	**[.025**	**.975]**	**P $\geq$ |Z|**
PAY_0	-0.18	0.01	-20.71	-0.19	-0.17	0.00	0.61	0.02	30.59	0.59	0.63	0.00
PAY_2	-0.14	0.01	-15.99	-0.15	-0.13	0.00	0.50	0.02	21.24	0.48	0.53	0.00
PAY_3	-0.14	0.01	-15.86	-0.14	-0.13	0.00	0.46	0.02	19.71	0.44	0.49	0.00
PAY_4	-0.13	0.01	-15.91	-0.14	-0.12	0.00	0.41	0.02	17.79	0.39	0.43	0.00
PAY_6	-0.11	0.01	-13.51	-0.12	-0.11	0.00	0.35	0.02	15.00	0.32	0.37	0.00
LIMIT_BAL	0.08	0.01	8.50	0.07	0.09	0.00	-0.29	0.02	-18.96	-0.31	-0.28	0.00
PAY_AMT5	0.04	0.01	3.56	0.03	0.05	0.00	-0.13	0.01	-9.00	-0.15	-0.12	0.00
PAY_AMT4	0.03	0.01	3.61	0.02	0.04	0.00	-0.12	0.01	-9.88	-0.13	-0.10	0.00
PAY_AMT3	0.02	0.01	1.99	0.01	0.03	0.02	-0.11	0.02	-7.02	-0.13	-0.10	0.00
PAY_AMT2	0.02	0.01	1.90	0.01	0.03	0.03	-0.10	0.01	-15.64	-0.11	-0.10	0.00
PAY_AMT1	0.03	0.01	2.81	0.02	0.04	0.00	-0.10	0.01	-9.06	-0.11	-0.09	0.00
SEX	0.03	0.01	3.25	0.02	0.04	0.00	-0.10	0.02	-5.14	-0.12	-0.08	0.00
PAY_AMT6	0.02	0.01	2.49	0.01	0.03	0.01	-0.10	0.01	-7.41	-0.11	-0.08	0.00
MARRIAGE	0.01	0.01	1.18	0.00	0.02	0.12	-0.06	0.02	-3.28	-0.07	-0.04	0.00
BILL_AMT1	-0.02	0.01	-1.97	-0.03	-0.01	0.02	-0.05	0.02	-2.84	-0.07	-0.03	0.00
AGE	0.00	0.01	0.20	-0.01	0.01	0.42	0.04	0.02	1.93	0.02	0.05	0.03
EDUCATION	-0.00	0.01	-0.50	-0.01	0.00	0.31	0.03	0.01	2.32	0.02	0.05	0.01
BILL_AMT6	-0.02	0.01	-1.76	-0.03	-0.01	0.04	-0.02	0.02	-0.95	-0.03	0.00	0.17

6.2.5 Post-modelling explainability using SHAP

SHAP was used to provide insights into the importance of each feature for each classification model. Table 10 exhibits the ranking of features based on the relative magnitudes of the mean absolute SHAP values. The PAY_0 is the most influential feature as it ranks highest across all the models.

The rankings of features for LR and LDA according to SHAP are different to the rankings of features presented in Tables 7 and 8. This can be attributed to the fact that mean absolute values can be easily influenced by extreme values resulting in erroneous rankings and conclusions. Feature importance provides a view of predictive factors of the classifiers.

It can be observed that predictions of the DT classifier depend only on 15 features as shown in Table 10, where the mean absolute SHAP values are not zero. Alternatively, three features, namely PAY_AMT5, MARRIAGE and EDUCATION are not used in predictions since the mean absolute SHAP values are zero. The features that ranked the highest in terms of importance according the mean absolute SHAP values also ranked highest in the graphical representation of the DT. Seemingly, SHAP feature importance rankings produces, but not always, results similar to the intrinsically explainable classifiers. Similar observations regarding feature importance can be made for the other classification models. It is evident that SHAP is also useful for feature selection because it can quantify the importance of each feature. However, a suitable threshold would have to be determined in order to decide which feature to select or remove.

Figures 13a and 13b demonstrate feature dependence plots for the top five features for each classification model. The y-axis has two coordinates, left and right. The right coordinate indicates the feature with the highest interaction. The left coordinate shows the SHAP values. SHAP values that are less than zero contribute negatively towards the predictions. A value of zero indicates no contribution. Whereas values greater than zero contribute positively towards predictions. In the case of predicting default, negative values reduce the expected probability of default and positive values increase the expected probability of default.

The dependence plots provide a view of the relationship between a feature's values and the model's predicted outcomes. The dependence plots reveal that the relationship between SHAP values, feature values and feature interaction are different for each classification model. For example, the LIMIT_BAL is the third most important feature for ANN. Furthermore, as the LIMIT_BAL increases the SHAP values decrease (see the third plot in the first row in Figure 13a). In addition, the LIMIT_BAL has a relatively stronger interaction with PAY_0. However, the LIMIT_BAL is the second most important feature for LGBM. An inverse relationship between the LIMIT_BAL values and SHAP values is observed, similar to that of ANN.

Table 10: Feature importance computed using mean absolute SHAP values for case study 1.

Features	**ANN**	**Bagging**	**DT**	**LDA**	**LGBM**	**LR**	**SVM**	**RF**
PAY_3	1.71×10^{-2}	2.43×10^{-2}	1.50×10^{-2}	1.91×10^{-2}	8.98×10^{-2}	3.62×10^{-2}	2.26×10^{-2}	1.69×10^{-2}
AGE	8.25×10^{-3}	3.44×10^{-3}	1.83×10^{-3}	8.47×10^{-3}	4.20×10^{-2}	4.34×10^{-2}	1.06×10^{-4}	1.32×10^{-2}
BILL_AMT1	6.61×10^{-3}	2.99×10^{-3}	3.09×10^{-2}	1.69×10^{-2}	1.16×10^{-1}	5.60×10^{-2}	3.49×10^{-3}	1.79×10^{-2}
PAY_AMT2	4.53×10^{-3}	2.93×10^{-4}	4.37×10^{-2}	8.20×10^{-4}	9.43×10^{-2}	3.03×10^{-2}	6.77×10^{-3}	7.97×10^{-4}
SEX	2.48×10^{-3}	4.96×10^{-3}	1.60×10^{-3}	7.48×10^{-3}	3.89×10^{-2}	5.06×10^{-2}	1.13×10^{-4}	9.38×10^{-3}
PAY_AMT5	4.89×10^{-3}	7.07×10^{-4}	0.00	1.63×10^{-3}	3.32×10^{-2}	1.54×10^{-2}	2.50×10^{-3}	3.51×10^{-3}
PAY_AMT1	4.57×10^{-3}	7.72×10^{-4}	9.26×10^{-3}	2.49×10^{-3}	9.29×10^{-2}	2.46×10^{-2}	8.19×10^{-3}	4.08×10^{-3}
LIMIT_BAL	2.15×10^{-2}	1.09×10^{-2}	2.93×10^{-2}	9.20×10^{-3}	2.14×10^{-1}	6.29×10^{-2}	1.18×10^{-2}	2.31×10^{-2}
PAY_4	2.77×10^{-2}	7.64×10^{-3}	1.97×10^{-2}	1.23×10^{-2}	7.08×10^{-2}	2.19×10^{-2}	1.61×10^{-2}	4.90×10^{-3}
PAY_AMT4	1.84×10^{-3}	6.68×10^{-4}	1.28×10^{-2}	6.96×10^{-4}	1.05×10^{-1}	7.36×10^{-3}	6.10×10^{-3}	1.11×10^{-3}
MARRIAGE	3.08×10^{-3}	5.91×10^{-3}	0.00	9.30×10^{-3}	3.40×10^{-2}	4.70×10^{-2}	4.15×10^{-4}	9.54×10^{-3}
PAY_0	1.55×10^{-1}	2.75×10^{-2}	1.46×10^{-1}	1.05×10^{-1}	4.71×10^{-1}	2.46×10^{-1}	5.37×10^{-2}	1.55×10^{-1}
EDUCATION	3.35×10^{-3}	1.81×10^{-3}	0.00	8.34×10^{-3}	2.75×10^{-2}	3.80×10^{-2}	4.03×10^{-4}	1.34×10^{-2}
BILL_AMT6	1.74×10^{-3}	1.18×10^{-3}	1.22×10^{-3}	2.15×10^{-3}	3.83×10^{-2}	2.10×10^{-2}	6.67×10^{-4}	5.24×10^{-4}
PAY_AMT3	8.80×10^{-4}	3.84×10^{-4}	6.28×10^{-3}	3.51×10^{-4}	6.17×10^{-2}	5.27×10^{-3}	8.02×10^{-3}	6.15×10^{-4}
PAY_6	5.90×10^{-3}	4.41×10^{-3}	6.35×10^{-3}	1.27×10^{-3}	6.09×10^{-2}	1.65×10^{-2}	7.92×10^{-3}	1.15×10^{-2}
PAY_2	1.52×10^{-2}	2.12×10^{-2}	1.40×10^{-2}	2.15×10^{-2}	5.62×10^{-2}	3.10×10^{-2}	3.30×10^{-2}	3.02×10^{-2}
PAY_AMT6	1.60×10^{-3}	6.72×10^{-4}	1.50×10^{-2}	4.08×10^{-4}	4.07×10^{-2}	1.57×10^{-3}	4.60×10^{-3}	8.13×10^{-4}

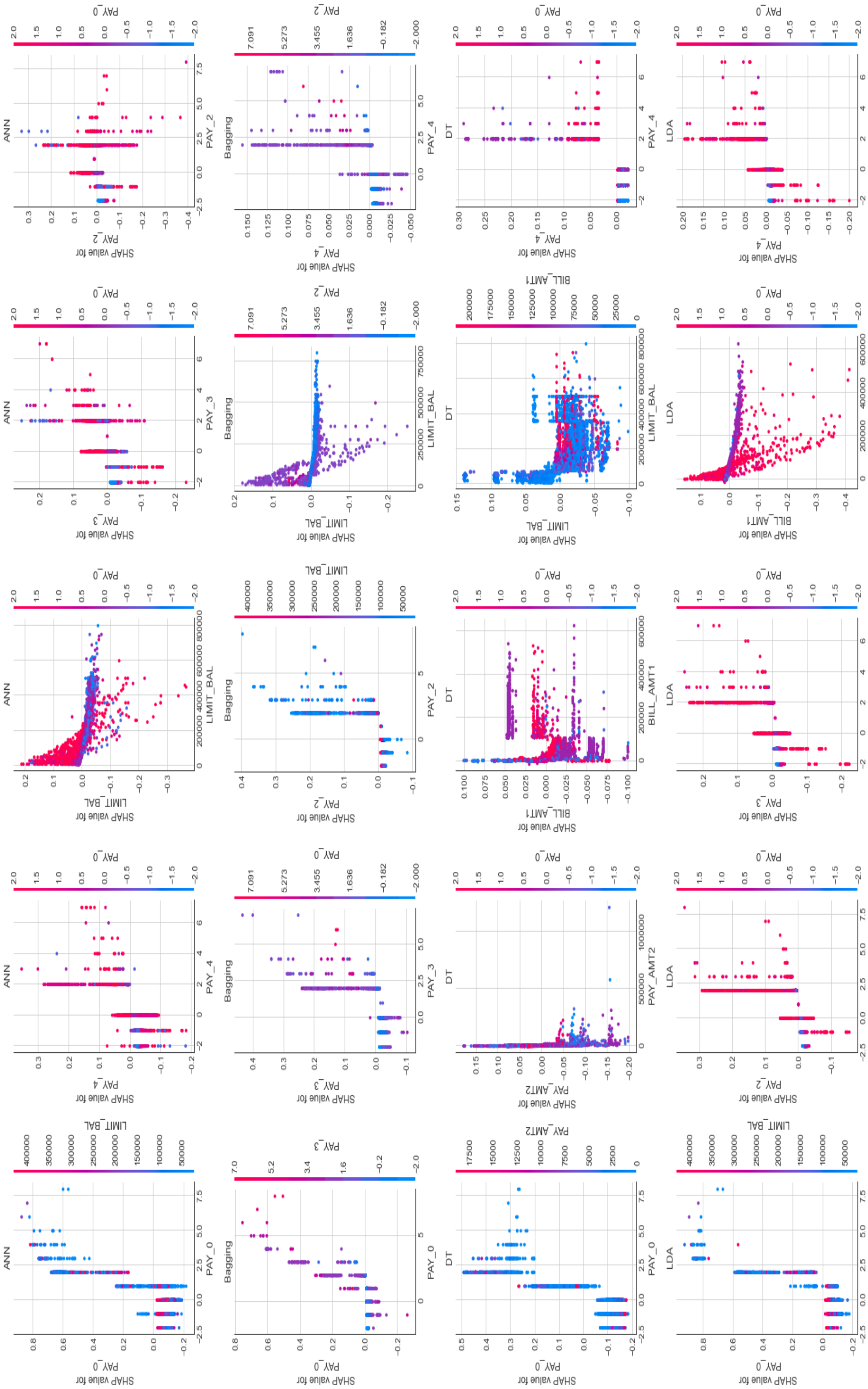

Figure 13a: SHAP feature dependence for classifiers for case study 1.

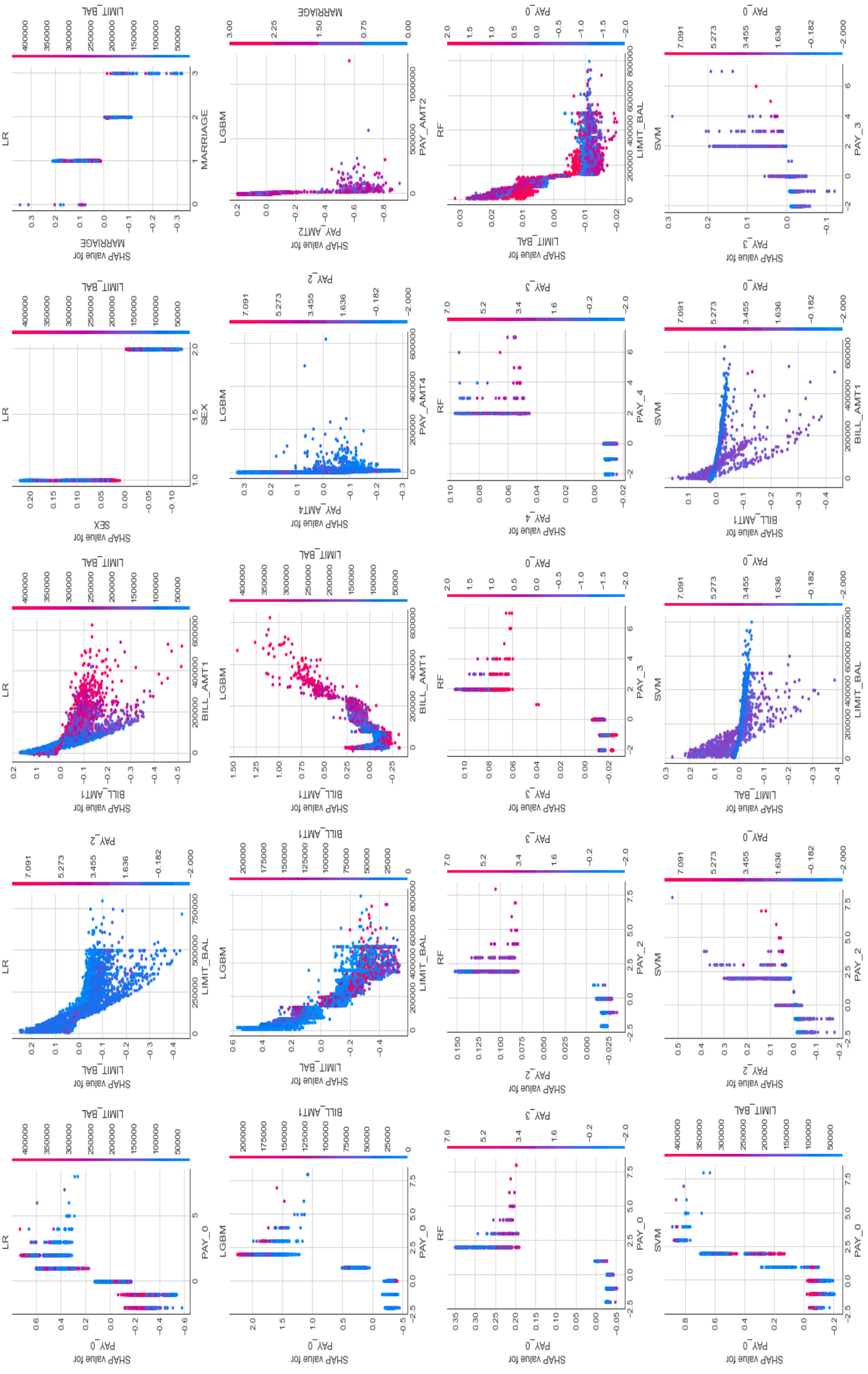

Figure 13b: SHAP feature dependence for classifiers for case study 1.

Furthermore, the LIMIT_BAL has a relatively stronger interaction with BILL_-AMT1 (see the second plot in the fifth row in Figure 13a). In this study, the feature interaction effects are analysed between the feature of interest and the most influential feature, i.e., limiting the interaction effects to the most influential feature.

Figure 14 shows the instance level explanation provided by the LIME framework as predicted by LGBM classification model. These instance level explanations can be generated for all the classifiers since LIME is model agnostic. For this example, LIME explains that this customer is predicted not to default on their credit card and this decision is based mainly on the PAY_0, LIMIT_BAL, PAY_AMT3, PAY_6, PAY_4, SEX, MARRIAGE, PAY_AMT6 and BILL_AMT6. MARRIAGE, highlighted in blue, contributes towards non-default in this case.

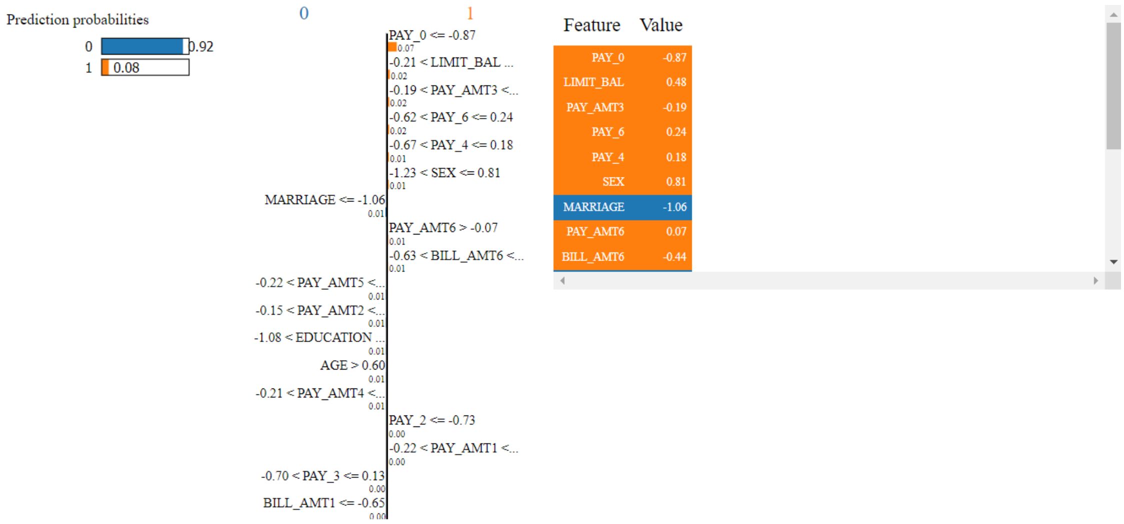

Figure 14: LIME interpretation for LGBM classifier for case study 1.

6.3 Case study 2: Home credit default

This section presents results for the pre- and post-modelling explainability for case study 2. In pre-modelling explainability, features that served as inputs into the classification models are described. In post-modelling explainability the results for intrinsic explainability of LR, LDA, and DT are discussed. Furthermore, the post-modelling explainability results for SVM, ANN, bagging, RF, and LGBM achieved using SHAP and LIME are presented.

6.3.1 Pre-modelling explainability

Pre-modelling explainability encompasses methods to understand the data prior to training of the classifiers for credit scoring. This is achieved through an exploratory analysis of the data, explainable feature engineering, data summaries and feature selection approaches. The results of the data summaries, more specifically using univariate analysis, and feature selection are presented. The univariate analysis is used to show the relationship between features and the target variable. The IV was used to quantify the strength of the relationship between features and target. Given the high number of features for this dataset only the most important features were analysed.

Table D.1 shows the analysis of the education level in relation to the outcome of the loan. Applicants that have a secondary special (Sec. special) education and higher education (higher edu.) constitute 71.02% and 24.34% of applicants, respectively. Applicants with an academic degree make up the lowest percentage of approximately 0.05%. However, applicants with an academic degree also have the lowest bad rate of less than 2%. Lower secondary (Lower sec.) applicants make up 1.24% of applicants, but they have the largest bad rate of 10.93%. This is possibly attributed to the fact that the income of an individual is likely to be higher depending the level of education. Furthermore, the low income earners are likely to be in financial distress and consequently default on loan obligations. The IV of this feature is 0.05, which indicates a moderate relationship to the outcome of the loan.

The analysis of income sources depicted in Table D.2, indicates that most applicants have income sources from working, followed by commercial associate (Com. associate), pensioner and state servant make up 51.63%, 23.29%, 18% and 7.06%, respectively. Applicants from these sources have a bad rate of less than 10%. All other attributes, namely maternity leave, businessman and student were combined under unemployed due to low volumes and similar bad rates, and they make up 0.02% of applicants with a bad rate of 18.18%. The distribution of sources of income indicates that loans are primarily given to individuals who have a stable source of income. Furthermore, the information value of this feature is 0.06, indicating a moderate relationship with the outcome of the loan.

The occupation feature has many occupation types which were grouped based on the low variability of the weight of evidence (WoE), bad rates as well as low volumes. Occupation 1 is mainly made up of low-skill labourers and has an observed default rate of 17.15% as shown in Table D.3. Occupation 8 (accountants) has the lowest default rate of 4.83%. This table shows that there is a conceivable relationship between the level of professional skills and default rates. The observed IV is 0.09, which also shows a moderate relationship between occupation type and default rates.

Similarly, there are many organisation types and they were grouped based on the low variability of the WoE, bad rates as well as low volumes, as presented in Table D.4. Organisation 14 has the lowest default rate of 3.70%. This analysis shows that there is a conceivable relationship between the organisations and default rates. The observed IV is 0.07, which also shows a moderate relationship between organisation type and default rates.

The age of the customers is derived from the DAYS_BIRTH feature by converting days into years. In addition, this feature is converted to a positive value because in the data the age is calculated from the time of application and not from the birth date. The univariate analysis of the age of the applicants, shown in Table D.5, indicates that the younger the applicants the higher the default rate. This could be attributed to the fact that the younger population is still new to the job market and not as financially stable as the older population. Furthermore, the default rate for the age group 20 to 28 years is above average at 11.57%. In this analysis, the age variable was binned such that each interval of the age or age groups are fairly equal in size. The univariate analysis of the age of the applicants yields an IV of 0.08 indicating a moderate association with default rates.

The EXT_SOURCE_1 feature is a normalized score from an external data source. Table D.6 shows this score is not populated for 56% of the population. The bad rate for the population for which the score is blank is slightly above average at 8.52%. The highest bad rates observed is 17.56% for the lower scores and 2.5% for the higher end of the scores. The IV of 0.15 indicates a moderate degree of association between EXT_SOURCE_1 and the target. A similar analysis yields an IV of 0.35 if only the scored population is analysed, i.e., excluding missing values. This shows that the score has a fairly strong relationship with the target for the scored population.

The EXT_SOURCE_2 feature is also a normalized score from an external data source. Table D.7 shows this score is mostly populated, since less than 0.5% are missings. The highest bad rates observed is 18.35% for the lower scores and 2.97% for the higher end of the scores. The IV of 0.31 indicates a moderate degree of association between EXT_SOURCE_2 and the target.

Similarly, EXT_SOURCE_3 is also a normalized score from an external data source. Table D.8 shows this score is mostly populated, since less than 20% are missings. The highest bad rates observed is 20% for the lower scores and 3.23% for the higher end of the scores. The IV of 0.33 indicates a moderate degree of association between EXT_SOURCE_3 and the target.

The subset of relevant features employed for training classifiers was chosen using a combination of feature selection strategies. The initial selection of 100 features was aided by the use of two methods, namely Kendall tau's correlation and χ^2, both of which are categorised as filter methods. The VIF was used to eliminate features that are correlated by excluding features above a VIF threshold of 5. This reduced the number of features from 100 to 65.

Furthermore, lasso regression (Lasso R.), ridge regression (Ridge R.), RF and LGBM RFE wrapper methods were utilised to determine the top ranking features. The performance of the RFE wrapper methods were evaluated using all top ranking 60, 30 and 15 features. As shown in Table 11, selecting the top ranking 15 features for each method produces similar performance results as selecting 60 features. Therefore, the number of features used can be reduced further to 15 without compromising on performance. The final features were selected based on a voting system of the methods on the top 15 features selected by each model, where a feature must be selected by at least one RFE wrapper method.

Table 11: Performance evaluation of RFE wrapper methods tested on 15, 30 and 60 features.

	ROC AUC			Precision			Recall		
No. features	**15**	**30**	**60**	**15**	**30**	**60**	**15**	**30**	**60**
LGBM	67.92	68.06	67.92	13.65	13.98	13.85	80.22	79.53	79.69
Lasso R.	67.67	67.66	67.74	13.65	12.78	13.33	79.86	84.38	81.69
RF	66.33	66.22	66.56	13.01	14.06	14.39	80.16	74.38	73.79
Ridge R.	67.61	67.66	67.73	13.47	12.79	13.32	80.68	84.36	81.76

The final number of features that were selected were 24, where each feature was selected by either one of the RFE wrapper methods as tabulated in Table 12. The final features that were extracted can be broadly categorised as belonging to the following categories: external sources, age related, education and employment, gender, car ownership flag, income and credit characteristics, changes in contact information, social circle observations, car ownership ratios, apartment scores and loan application related.

6.3.2 Classifier performance tuning

Table 13 shows the hyperparameters that were tuned for each model, as well as optimal values for these hyperparameters. The search space is described in Section 6.1. The optimal hyperparameters were obtained using a five-fold cross-validation random search, repeated 15 times.

The AUC was used to assess and rank the classifiers' ability to distinguish between good and bad credit applicants. Table 14 displays the optimal threshold, i.e., best value to classify an outcome as either default or non-default, as well as the AUC for all classifiers for the training and test subsets. On the training subsets, the LGBM classifier had an AUC of 82.54 when applied to 24 features. Furthermore, the LGBM classifier's AUC on training was significantly higher compared to performance on the other subsets. This implies that the LGBM classifier may be overfitting, even though it still performed reasonably well and consistently on those subsets. Overall, the classifiers displayed slightly higher performance on the subset of 24 features.

6.3.3 Performance evaluation

The performance of each classification model applied to the home credit default validation set was analysed in terms of AUC. Figure 15 shows that the DT achieved the lowest average AUC of 70.50% followed by ANN and RF with average AUCs of 72.70% and 72.85%, respectively. The LR classification model achieved the highest

Table 12: Features selected using recursive feature elimination methods for case study 2.

Category	Feature	Lasso R.	Ridge R.	RF	LGBM
Normalised scores	EXT_SOURCE_3	✓	✓	✓	✓
	EXT_SOURCE_2	✓	✓	✓	✓
	EXT_SOURCE_1	✓	✓	✓	✓
	EXT_SOURCE_MAX	-	-	✓	✓
Age related	DAYS_EMPLOYED	✓	✓	✓	✓
	DAYS_BIRTH	✓	✓	✓	✓
Education and employment	ORGANIZATION_TYPE_1	✓	✓	-	✓
	OCCUPATION_TYPE_1	✓	✓	-	✓
	NAME_EDUCATION_TYPE_0	✓	✓	-	✓
Gender	CODE_GENDER_1	✓	✓	-	✓
Car ownership	FLAG_OWN_CAR_1	✓	✓	-	✓
Type of loan	NAME_CONTRACT_TYPE_0	✓	✓	-	-
Income and credit	AMT_INCOME_TOTAL	-	-	✓	-
	AMT_ANNUITY	✓	✓	✓	✓
	CREDIT_GOODS_RATIO	✓	✓	-	✓
	ANNUITY_INCOME_RATIO	-	-	✓	-
	CREDIT_ANNUITY_RATIO	-	-	✓	✓
Personal details change	DAYS_ID_PUBLISH	-	-	✓	✓
	DAYS_REGISTRATION	-	-	✓	-
	DAYS_LAST_PHONE_CHANGE	-	-	✓	-
Social circle	DEF_30_CNT_SOCIAL_CIRCLE	✓	✓	-	-
	NAME_TYPE_SUITE_0	-	-	-	-
Apartment related	REGION_RATING_CLIENT_W_CITY_0	✓	✓	-	-
	REGION_POPULATION_RELATIVE	-	-	✓	-
	WALLSMATERIAL_MODE_1	-	-	-	-
	REG_REGION_NOT_LIVE_REGION	-	-	-	-
	REG_CITY_NOT_WORK_CITY	-	-	-	-
	NONLIVINGAREA_MODE	-	-	-	-
Application related	HOUR_APPR_PROCESS_START	-	-	✓	-
	WEEKDAY_APPR_PROCESS_START_1	-	-	-	-

average AUC of 74.58%. In this experiment the transparent linear models perform relatively well on average compared to the black box models. This is possible if the relationship between the features and target variable is linear and the distributions of the features meet the requirements of linear models. The findings of this experiment suggest that the trade-off between accuracy and explainability may not always apply.

An analysis of the means was conducted using ANOVA and the Kruskal Wallis test. The data fails the test for normality and therefore ANOVA can not be used to compare the means. The Kruskal Wallis test indicates that there is a significant difference in the means of the models, since the p-values are less than 0.05. Furthermore, a multi-comparison analysis using the Dunn test shows that the means of the LR, LDA, LGBM, bagging and SVM are not significantly different as shown in Table 15.

6.3.4 Post-modelling explainability of interpretable models

The DT inherently produces features importance since the order of feature splits depends on their discriminatory power. The classification is visually represented by the branches and terminal nodes of the tree. Figure 16 depicts an example of one of the induced trees illustrating the sequence of features as nodes as well as branches to show the relationship between variables. The features, EXT_SOURCE_3, EXT_SOURCE_2 and EXPECTED_INTEREST_SHARE have the highest rank in terms of discriminating between classes.

Table 16 contains the coefficients, p-values, standard errors, and confidence intervals for each feature for the optimal LR model. The features were ordered in terms of the contribution to the predictions by calculating the absolute value of the coefficients and ranking them in descending order. The p-values for the top 22 features were less

Table 13: Optimal hyperparameters for each classifier for case study 2.

Classifier	Hyperparameter	Optimal value
ANN	Hidden layers	Three layers with 120, 80, 40 nodes, respectively.
	Activation	Tanh
	Maximum iterations	20
bagging	Number of estimators	15
	Maximum samples	750
DT	Maximum Depth	7
	Maximum leaf nodes	48
	Minimum sample per leaf	500
	Class weight	balanced
LDA	Solver	SVD
LGBM	Number of leaves	40
	Maximum depth	5
	Learning rate	0.2
	Reg alpha	0.01
LR	Class weight	balanced
SVM	Class weight	balanced
	Alpha	$10^{-4+i(\frac{9}{49})}$ where $i = 5$
RF	Max depth	6
	Maximum leaf nodes	12

Table 14: The optimal threshold and model performance for the training and testing subsets for case study 2. Results showed that the LGBM classifier outperformed other classifiers, particularly on the 24 selected features. Overall, the classifiers exhibited slightly higher performance on this subset of features.

	J-Statistic				**AUC (Training)**				**AUC (Testing)**			
No. features	8	12	16	24	8	12	16	24	8	12	16	24
ANN	8.01	7.94	7.91	8.69	74.22	75.30	75.58	76.29	73.99	74.84	**75.01**	75.47
bagging	8.19	7.96	8.12	7.95	73.37	73.98	74.19	74.24	73.24	73.72	74.02	74.09
DT	51.44	50.68	50.83	50.98	73.03	73.22	73.16	73.16	71.66	71.65	71.84	71.85
LDA	7.55	7.73	7.77	7.29	73.85	74.73	74.84	74.95	73.70	74.53	74.68	74.83
LGBM	8.34	8.00	8.05	9.10	**75.17**	**76.12**	**76.12**	**82.54**	**74.08**	**74.92**	74.96	**76.46**
LR	48.91	49.33	49.59	8.07	73.89	74.79	74.95	75.06	73.70	74.54	74.72	74.90
RF	8.28	8.24	8.78	8.55	72.70	73.29	73.29	73.72	72.15	72.81	72.95	73.30
SVM	8.02	7.95	8.22	8.28	73.75	74.61	74.73	74.92	73.60	74.37	74.52	74.74

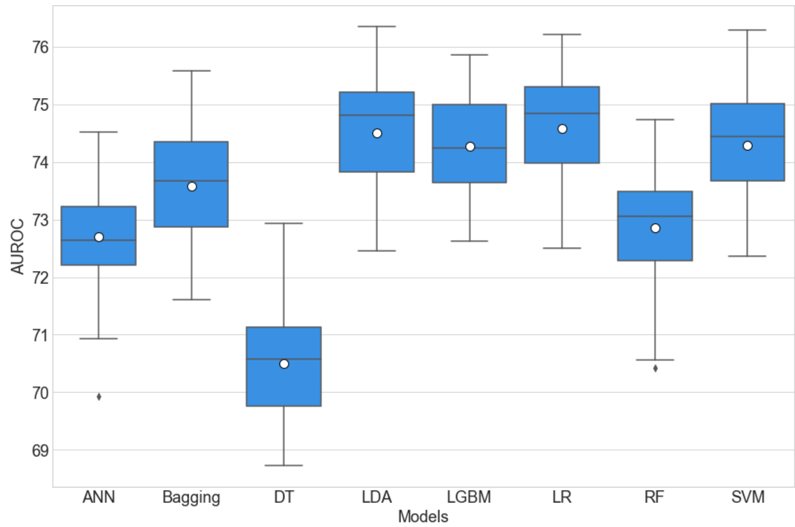

Figure 15: Performance of classification models on the validation set with 24 features for case study 2.

Table 15: Dunn's multi-comparison test for classification models for case study 2. The average AUCs of LR, LDA and LGBM are significantly different to the average AUCs of ANN and DT since the p-values are less than 0.05.

	AUC	**ANN**	**Bagging**	**DT**	**LDA**	**LGBM**	**LR**	**RF**	**SVM**
ANN	72.70	1.00							
Bagging	73.58	0.16	1.00						
DT	70.50	**0.01**	**0.00**	1.00					
LDA	74.50	**0.00**	0.09	**0.00**	1.00				
LGBM	74.27	**0.00**	0.41	**0.00**	1.00	1.00			
LR	74.58	**0.00**	**0.03**	**0.00**	1.00	1.00	1.00		
RF	72.85	1.00	0.43	**0.00**	**0.00**	**0.00**	**0.00**	1.00	
SVM	74.29	**0.00**	0.43	**0.00**	1.00	1.00	1.00	**0.00**	1.00

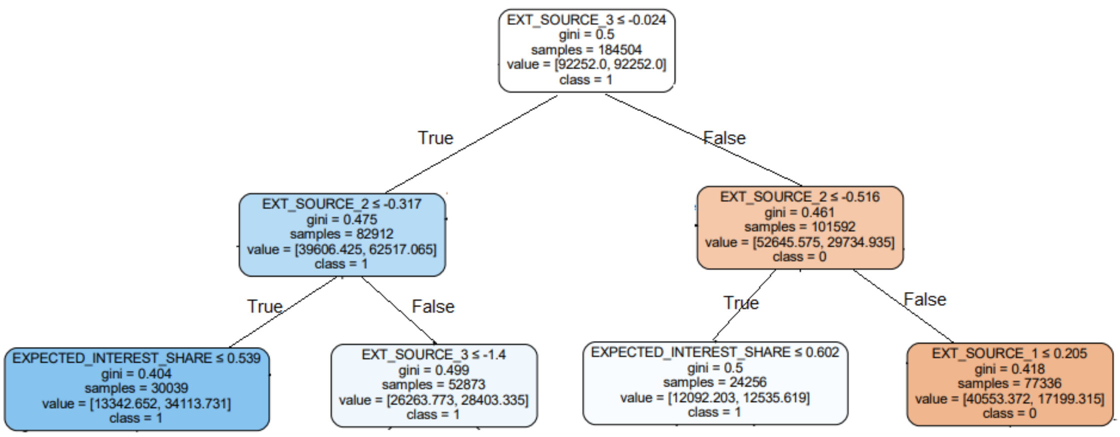

Figure 16: A representation of the DT up to a depth of two for case study 2.

than 0.05, indicating that those features significantly contribute to the scoring models. This was also supported by the relatively low standard error values of these features. The AMT_ANNUITY, EXT_SOURCE_MAX and HOUR_APPR_PROCESS_START were less significant and could be removed from the LR classification model. The intercept is used to provide a probability of an outcome when all features are zero.

Table 16: Feature importance and impacts for the for LR classifier for case study 2.

Features	**Coefficients**	**std error**	**z**	**[.025**	**.975]**	$P \geq \lvert Z \rvert$
INTERCEPT	-2.79	0.01	-251.47	-2.81	-2.78	0.00
EXT_SOURCE_3	-0.48	0.01	-45.53	-0.49	-0.47	0.00
EXT_SOURCE_1	-0.39	0.02	-24.77	-0.41	-0.38	0.00
EXT_SOURCE_2	-0.36	0.01	-32.39	-0.38	-0.35	0.00
DAYS_BIRTH	0.27	0.01	18.44	0.26	0.29	0.00
CREDIT_GOODS_RATIO	0.17	0.01	16.65	0.16	0.18	0.00
DAYS_EMPLOYED	0.14	0.01	11.45	0.13	0.15	0.00
AMT_INCOME_TOTAL	0.13	0.03	3.85	0.10	0.17	0.00
NAME_EDUCATION_TYPE_0	-0.13	0.01	-13.12	-0.14	-0.12	0.00
ORGANIZATION_TYPE_1	0.13	0.01	12.45	0.12	0.14	0.00
FLAG_OWN_CAR_1	0.13	0.01	13.47	0.12	0.13	0.00
CODE_GENDER_1	0.12	0.01	10.58	0.11	0.13	0.00
ANNUITY_INCOME_RATIO	0.10	0.01	7.34	0.09	0.12	0.00
REGION_RATING_CLIENT_W_CITY_0	-0.08	0.01	-8.99	-0.09	-0.07	0.00
DEF_30_CNT_SOCIAL_CIRCLE	0.08	0.01	9.79	0.07	0.08	0.00
OCCUPATION_TYPE_1	0.06	0.01	5.53	0.05	0.07	0.00
DAYS_ID_PUBLISH	0.05	0.01	5.24	0.04	0.06	0.00
NAME_CONTRACT_TYPE_0	-0.04	0.01	-3.29	-0.05	-0.03	0.00
CREDIT_ANNUITY_RATIO	-0.04	0.01	-4.68	-0.05	-0.03	0.00
DAYS_LAST_PHONE_CHANGE	0.04	0.01	3.82	0.03	0.05	0.00
DAYS_REGISTRATION	0.03	0.01	2.66	0.02	0.04	0.00
REGION_POPULATION_RELATIVE	0.02	0.01	1.93	0.01	0.03	0.03
AMT_ANNUITY	0.02	0.02	1.00	-0.00	0.03	0.16
EXT_SOURCE_MAX	0.01	0.02	0.73	-0.00	0.03	0.23
HOUR_APPR_PROCESS_START	-0.01	0.01	-0.96	-0.02	0.00	0.17

Table 17 presents the measures of statistical significance and confidence intervals of the LDA parameters indicate that the top 22 features contribute significantly to the model, since the p-values are less than 0.05. This provides an indication of feature importance and the contribution of each feature towards predicting default risk.

The p-values in Table 17 are less than 0.05 indicating that the features are meaningful additions to the model and are associated with the target. This, like the LR, was supported by the relatively low standard error values. It is also observed that the sequence of the importance of features for LDA is similar to that of LR.

The group means for each feature and each class are provided in Table 18. The differences in mean values for each feature per class imply that these features have an impact on the predictions of classes. Furthermore, the low standard errors and confidence intervals indicate that the mean values are expected to fall within the range of given values at a 95% confidence level. Furthermore, the measures of statistical significance of the LDA parameters for default class indicate that the top 22 features contribute significantly to the model since the p-values are less than 0.05.

Table 17: Feature importance and impacts for LDA classifier for case study 2.

Features	Coefficients	std error	z	[.025	.975]	$P \geq \lvert Z \rvert$
INTERCEPT	-2.83	0.01	-240.71	-2.84	-2.82	0.00
EXT_SOURCE_3	-0.46	0.01	-34.88	-0.47	-0.44	0.00
EXT_SOURCE_2	-0.39	0.01	-33.22	-0.41	-0.38	0.00
EXT_SOURCE_1	-0.35	0.02	-19.56	-0.37	-0.33	0.00
DAYS_BIRTH	0.28	0.02	17.80	0.26	0.29	0.00
CREDIT_GOODS_RATIO	0.18	0.01	17.59	0.17	0.19	0.00
CODE_GENDER_1	0.16	0.01	15.11	0.14	0.17	0.00
FLAG_OWN_CAR_1	0.14	0.01	14.41	0.13	0.15	0.00
ORGANIZATION_TYPE_1	0.14	0.01	13.04	0.13	0.15	0.00
EXT_SOURCE_MAX	-0.12	0.02	-6.14	-0.14	-0.10	0.00
AMT_INCOME_TOTAL	0.10	0.04	2.47	0.06	0.15	0.01
ANNUITY_INCOME_RATIO	0.10	0.02	6.73	0.09	0.12	0.00
DAYS_EMPLOYED	0.10	0.01	10.75	0.09	0.11	0.00
DEF_30_CNT_SOCIAL_CIRCLE	0.09	0.01	8.59	0.08	0.10	0.00
NAME_EDUCATION_TYPE_0	-0.08	0.01	-9.62	-0.09	-0.07	0.00
REGION_RATING_CLIENT_W_CITY_0	-0.08	0.01	-6.91	-0.09	-0.07	0.00
OCCUPATION_TYPE_1	0.06	0.01	5.90	0.05	0.07	0.00
CREDIT_ANNUITY_RATIO	-0.06	0.01	-6.88	-0.06	-0.05	0.00
REGION_POPULATION_RELATIVE	0.05	0.01	5.02	0.04	0.06	0.00
DAYS_ID_PUBLISH	0.04	0.01	4.19	0.03	0.05	0.00
DAYS_LAST_PHONE_CHANGE	0.03	0.01	3.58	0.02	0.04	0.00
DAYS_REGISTRATION	0.03	0.01	3.14	0.02	0.04	0.00
NAME_CONTRACT_TYPE_0	-0.03	0.01	-2.65	-0.04	-0.02	0.00
AMT_ANNUITY	0.03	0.02	1.56	0.01	0.04	0.06
HOUR_APPR_PROCESS_START	0.01	0.01	1.26	0.00	0.02	0.10

Table 18: Analysis of group mean estimates for LDA classifier for case study 2.

Features	**Non-default class mean**						**Default class mean**					
	Means	**std error**	**z**	**[.025**	**.975]**	**P ≥ |Z|**	**Means**	**std error**	**z**	**[.025**	**.975]**	**P ≥ |Z|**
EXT_SOURCE_MAX	0.06	0.00	23.98	0.05	0.06	0.00	-0.68	0.01	-74.89	-0.69	-0.67	0.00
EXT_SOURCE_3	0.05	0.00	20	0.05	0.05	0.00	-0.56	0.01	-74.15	-0.57	-0.56	0.00
EXT_SOURCE_2	0.05	0.00	18.42	0.04	0.05	0.00	-0.56	0.01	-60.53	-0.57	-0.55	0.00
EXT_SOURCE_1	0.04	0.00	17.31	0.04	0.04	0.00	-0.49	0.01	-64.35	-0.49	-0.48	0.00
OCCUPATION_TYPE_1	-0.03	0.00	-9.87	-0.03	-0.02	0.00	0.28	0.01	34.89	0.27	0.29	0.00
ORGANIZATION_TYPE_1	-0.02	0.00	-8.36	-0.03	-0.02	0.00	0.24	0.01	31.83	0.23	0.25	0.00
REGION_RATING_CLIENT_W_CITY_0	0.02	0.00	8.84	0.02	0.02	0.00	-0.20	0.01	-26.16	-0.21	-0.19	0.00
DAYS_BIRTH	0.02	0.00	7.75	0.02	0.02	0.00	-0.26	0.01	-32.11	-0.27	-0.25	0.00
CREDIT_GOODS_RATIO	-0.02	0.00	-7.03	-0.02	-0.02	0.00	0.25	0.01	25.81	0.24	0.25	0.00
CODE_GENDER_1	-0.02	0.00	-7.99	-0.02	-0.02	0.00	0.19	0.01	24.98	0.18	0.19	0.00
DAYS_EMPLOYED	-0.02	0.00	-6.28	-0.02	-0.01	0.00	0.23	0.01	42.04	0.22	0.23	0.00
DAYS_LAST_PHONE_CHANGE	-0.02	0.00	-6.28	-0.02	-0.01	0.00	0.19	0.01	25.47	0.18	0.20	0.00
NAME_EDUCATION_TYPE_0	0.01	0.00	5.63	0.01	0.02	0.00	-0.20	0.01	-30.11	-0.21	-0.19	0.00
DAYS_ID_PUBLISH	-0.01	0.00	-6.26	-0.02	-0.01	0.00	0.18	0.01	24.91	0.18	0.19	0.00
DAYS_REGISTRATION	-0.01	0.00	-5.57	-0.01	-0.01	0.00	0.13	0.01	16.06	0.12	0.14	0.00
CREDIT_ANNUITY_RATIO	0.01	0.00	4.20	0.01	0.01	0.00	-0.10	0.01	-13.06	-0.10	-0.09	0.00
REGION_POPULATION_RELATIVE	0.01	0.00	4.46	0.01	0.01	0.00	-0.10	0.01	-15.10	-0.11	-0.09	0.00
DEF_30_CNT_SOCIAL_CIRCLE	-0.01	0.00	-4.29	-0.01	-0.01	0.00	0.11	0.01	12.39	0.10	0.12	0.00
FLAG_OWN_CAR_1	-0.01	0.00	-3.18	-0.01	-0.01	0.00	0.07	0.01	9.29	0.06	0.08	0.00
NAME_CONTRACT_TYPE_0	0.01	0.00	3.19	0.01	0.01	0.00	-0.10	0.01	-15.47	-0.10	-0.09	0.00
HOUR_APPR_PROCESS_START	0.01	0.00	2.34	0.00	0.01	0.01	-0.09	0.01	-11.34	-0.09	-0.08	0.00
AMT_ANNUITY	0.00	0.00	1.79	0.00	0.01	0.04	-0.06	0.01	-8.85	-0.06	-0.05	0.00
ANNUITY_INCOME_RATIO	0.00	0.00	-0.64	0.00	0.00	0.26	0.03	0.01	3.78	0.02	0.04	0.00
AMT_INCOME_TOTAL	0.00	0.00	0.52	0.00	0.00	0.30	0.02	0.03	0.89	0.00	0.05	0.19

6.3.5 Post-modelling explainability using SHAP

SHAP is used to provide insights into feature importance and explanations for the predictions of black box models. In this study, it was also applied to the transparent models to compare the feature importance results presented in Tables 16 and 17. Figure 19 shows the ranking of features and the relative magnitudes of the mean absolute SHAP values, which can be interpreted as measures of feature importance for each model. The EXT_SOURCE_1, EXT_SOURCE_2 and EXT_SOURCE_3 are the most influential features as they rank high for most of the classification models except for the bagging model. The DAYS_BIRTH is the most predictive factor for the bagging classifier. Furthermore, the rankings of all features using SHAP does not produce the same rankings of features for LR and LDA as presented in Tables 16 and 17. This can be attributed to the fact that mean absolute values can be easily influenced by extreme values which can also influence how features rank.

The mean absolute SHAP value shows the relative measure of importance of each feature towards making predictions. This means SHAP is also useful for feature selection, since it quantifies the importance of each feature. It was that observed some classification models had features with negligibly small mean absolute SHAP values, which suggests that further feature selection or reduction could have been applied. In this study, the DT and SVM had features with mean SHAP values of zero. This implies that the predictions of default were not influenced by these features.

Figures 17a and 17b exhibit feature dependence plots for the top five features for each classification technique. The y-axis has two coordinates, left and right. The right coordinate indicates the feature with the highest interaction. The left coordinate shows the SHAP values. SHAP values that are less than zero contribute negatively towards the predictions. A value of zero indicates no contribution. Whereas values greater than zero contribute positively towards predictions. In the case of predicting default, negative values reduce the expected probability of default and positive values increase the expected probability of default. The x-axis shows the range of feature values. In Figure 17a, from plots 1 - 5 in the second row, it can be observed that almost all SHAP values for the top 5 features are close to zero for bagging. This suggests that this particular range of feature values has a minor impact on the SHAP values and, consequently, on the predictions.

The dependence plots illustrate the relationship between a feature's values and the predictions of the model. The dependence plots also show that the relationship between SHAP values, feature values and feature interaction are different for each classification model. The feature interaction effects are analysed between the feature of interest and the most influential feature, i.e., limiting the interaction effects to the most influential feature. A feature that has a strong interaction effect with another feature tends to have a longer range of SHAP values at a constant feature value.

Table 19: Feature importance computed using mean absolute SHAP values for case study 2.

Features	ANN	Bagging	DT	LDA	LGBM	LR	SVM	RF
EXT_SOURCE_3	$7{,}83\times10^{-4}$	1.50×10^{-6}	1.05×10^{-1}	2.93×10^{-3}	2.66×10^{-1}	2.13×10^{-3}	1.00×10^{-2}	1.76×10^{-3}
EXT_SOURCE_2	$6{,}30\times10^{-4}$	1.49×10^{-6}	8.52×10^{-2}	2.18×10^{-3}	2.18×10^{-1}	1.43×10^{-3}	1.01×10^{-2}	1.16×10^{-3}
EXT_SOURCE_1	$4{,}13\times10^{-4}$	1.95×10^{-6}	6.05×10^{-2}	1.78×10^{-3}	1.86×10^{-1}	1.44×10^{-3}	7.59×10^{-3}	1.23×10^{-3}
DAYS_EMPLOYED	$9{,}25\times10^{-5}$	6.64×10^{-7}	5.42×10^{-3}	3.23×10^{-4}	1.01×10^{-1}	3.06×10^{-4}	3.48×10^{-3}	5.48×10^{-5}
DAYS_BIRTH	$4{,}56\times10^{-4}$	8.66×10^{-6}	0.00	2.38×10^{-3}	1.29×10^{-1}	1.60×10^{-3}	1.73×10^{-3}	9.50×10^{-4}
AMT_ANNUITY	$7{,}62\times10^{-5}$	2.85×10^{-7}	1.68×10^{-3}	2.18×10^{-4}	8.40×10^{-2}	6.72×10^{-5}	1.25×10^{-4}	8.02×10^{-5}
ORGANIZATION_TYPE_1	$5{,}64\times10^{-5}$	9.02×10^{-7}	1.64×10^{-3}	6.20×10^{-4}	7.10×10^{-2}	4.71×10^{-4}	2.09×10^{-3}	3.29×10^{-4}
OCCUPATION_TYPE_1	$1{,}88\times10^{-4}$	3.67×10^{-7}	6.83×10^{-3}	4.39×10^{-4}	5.81×10^{-2}	3.08×10^{-4}	1.97×10^{-3}	2.37×10^{-4}
NAME_EDUCATION_TYPE_0	$4{,}39\times10^{-5}$	2.42×10^{-7}	1.51×10^{-2}	2.42×10^{-4}	9.83×10^{-2}	2.24×10^{-4}	2.66×10^{-3}	1.69×10^{-4}
FLAG_OWN_CAR_1	$9{,}23\times10^{-5}$	6.54×10^{-7}	0.00	7.85×10^{-4}	1.05×10^{-1}	4.64×10^{-4}	2.62×10^{-4}	4.97×10^{-4}
CREDIT_GOODS_RATIO	$3{,}44\times10^{-4}$	2.03×10^{-7}	1.18×10^{-2}	1.28×10^{-3}	1.19×10^{-1}	6.95×10^{-4}	3.14×10^{-3}	3.31×10^{-4}
CODE_GENDER_1	$1{,}11\times10^{-4}$	1.08×10^{-6}	6.18×10^{-3}	9.05×10^{-4}	1.10×10^{-1}	6.18×10^{-4}	2.53×10^{-3}	5.34×10^{-4}
REGION_RATING_CLIENT_W_CITY_0	$6{,}26\times10^{-5}$	1.03×10^{-6}	NaN	3.91×10^{-4}	3.70×10^{-2}	3.10×10^{-4}	2.05×10^{-4}	2.29×10^{-4}
NAME_CONTRACT_TYPE_0	$5{,}06\times10^{-5}$	2.49×10^{-7}	NaN	9.05×10^{-5}	3.34×10^{-2}	7.07×10^{-5}	3.63×10^{-5}	6.66×10^{-5}
EXT_SOURCE_MAX	$1{,}17\times10^{-4}$	1.29×10^{-6}	NaN	1.16×10^{-3}	3.02×10^{-1}	2.01×10^{-4}	9.68×10^{-3}	2.00×10^{-4}
DEF_30_CNT_SOCIAL_CIRCLE	$5{,}81\times10^{-5}$	4.24×10^{-7}	NaN	4.84×10^{-4}	3.47×10^{-2}	2.71×10^{-4}		1.97×10^{-4}
DAYS_ID_PUBLISH	$8{,}35\times10^{-5}$	2.36×10^{-7}	NaN	2.67×10^{-4}	7.26×10^{-2}	1.90×10^{-4}	7.99×10^{-4}	1.50×10^{-4}
CREDIT_ANNUITY_RATIO	$1{,}30\times10^{-4}$	1.24×10^{-6}	NaN	2.66×10^{-4}	1.76×10^{-1}	1.29×10^{-4}	9.11×10^{-4}	1.13×10^{-4}
REGION_POPULATION_RELATIVE	$3{,}20\times10^{-5}$	1.66×10^{-7}	NaN	1.70×10^{-4}	4.22×10^{-2}	6.42×10^{-5}	1.31×10^{-4}	4.67×10^{-5}
HOUR_APPR_PROCESS_START	$6{,}18\times10^{-5}$	7.64×10^{-7}	NaN	9.22×10^{-5}	2.74×10^{-2}	6.14×10^{-5}	0.00	9.45×10^{-5}
DAYS_REGISTRATION	$1{,}88\times10^{-4}$	1.05×10^{-7}	NaN	1.18×10^{-4}	2.46×10^{-2}	1.10×10^{-4}	0.00	6.05×10^{-5}
DAYS_LAST_PHONE_CHANGE	$7{,}78\times10^{-5}$	1.59×10^{-7}	NaN	1.59×10^{-4}	4.89×10^{-2}	1.21×10^{-4}	6.84×10^{-4}	1.64×10^{-4}
ANNUITY_INCOME_RATIO	$3{,}11\times10^{-4}$	3.39×10^{-7}	NaN	4.03×10^{-4}	4.48×10^{-2}	3.15×10^{-4}	3.43×10^{-5}	1.84×10^{-4}
AMT_INCOME_TOTAL	$3{,}99\times10^{-5}$	3.02×10^{-7}	NaN	9.02×10^{-5}	2.45×10^{-2}	7.23×10^{-5}	0.00	4.57×10^{-5}

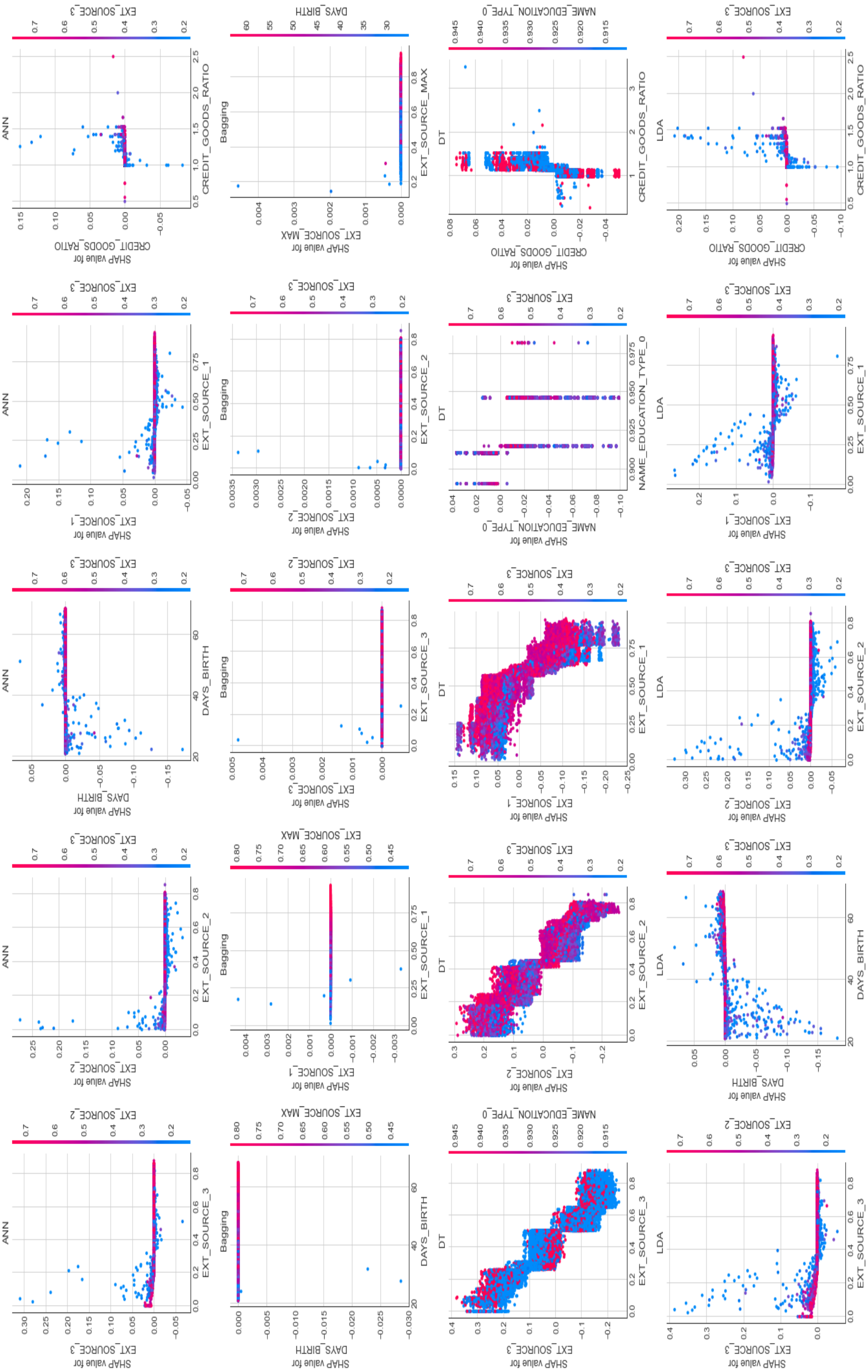

Figure 17a: SHAP feature dependence for classifiers for case study 2.

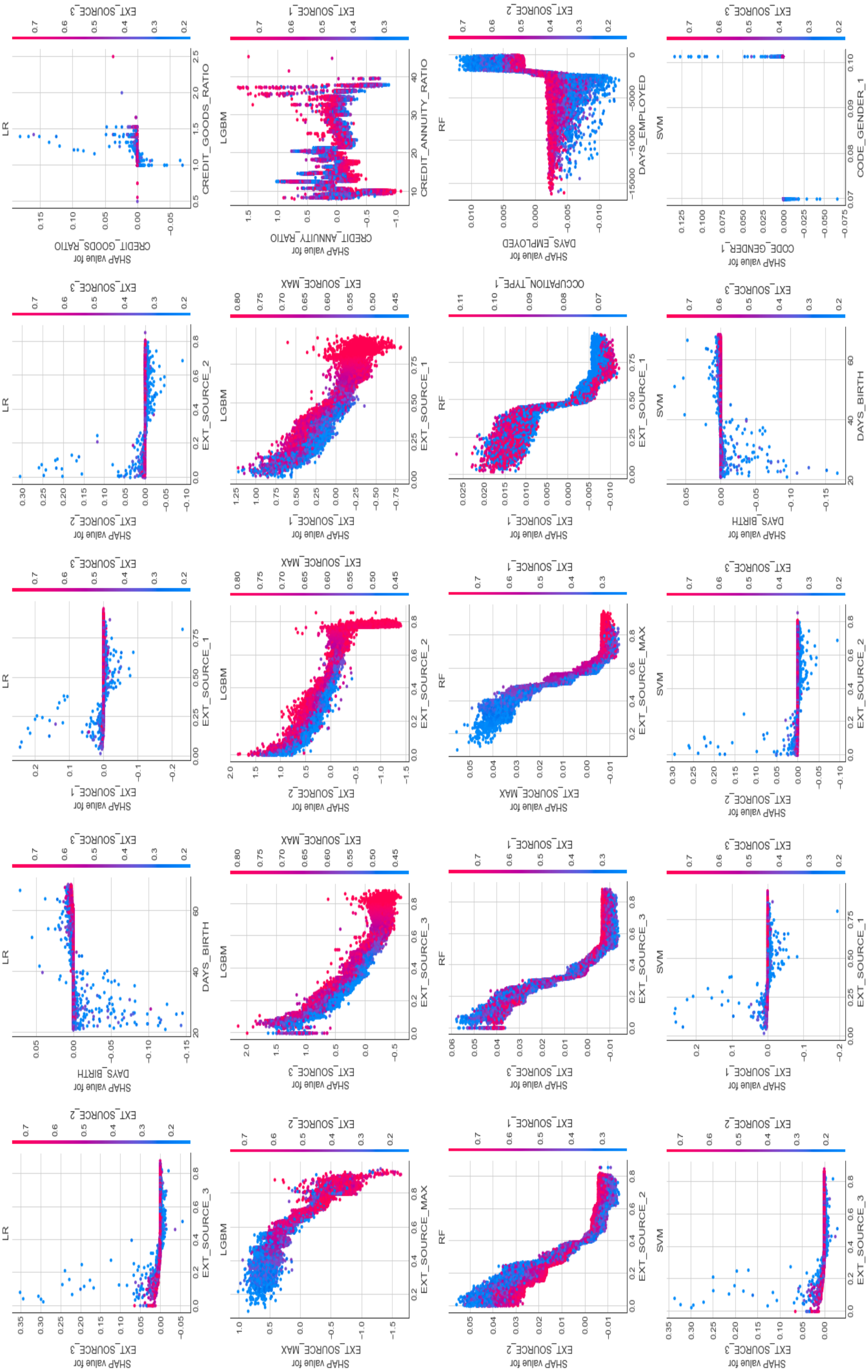

Figure 17b: SHAP feature dependence for classifiers for case study 2.

For example, a long range of SHAP values is observed at a CREDIT_GOODS_RATIO value of 1 for the DT classifier (see the last plot in the third row in Figure 17a). This means that the CREDIT_GOODS_RATIO has a strong interaction effect with NAME_EDUCATION_TYPE_0.

Figure 18 shows the instance level explanation provided by the LIME framework as predicted by LGBM classification model. In this example, the predicted class is non-default (encoded as zero) with a 98% probability. LIME shows the top 9 factors, which include DAYS_EMPLOYED, EXT_SOURCE_3, CREDIT_GOODS_-RATIO, EXT_SOURCE_MAX, CODE_GENDER_0, ORGANIZATION_TYPE_1, NAME_EDUCATION_TYPE_0, EXT_SOURCE_1 and DAY_BIRTH, contributing towards the non-default prediction. The features highlighted in blue are pushing the prediction toward non-default. The total tally of all the features combined are in favour of the non-default class.

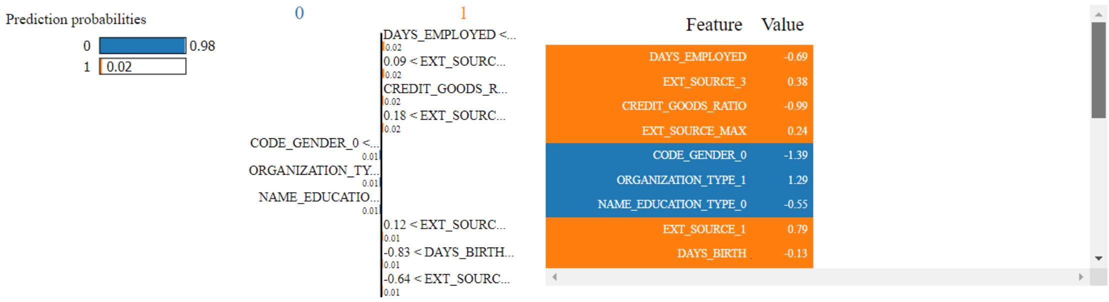

Figure 18: LIME interpretation for LGBM classifier for case study 2.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The main objectives of this project were to explore the advantages and effectiveness of alternative approaches in the context of credit applications and to apply XAI methods to classification models that are deemed as black box models, i.e., where outcomes are not explainable. These objectives as stated in Section 1.3 have been met and are discussed in Chapter 6.

To achieve the objectives of the research, eight classification models were constructed and tested against two credit datasets that are publicly available. Figure 19 highlights the accuracy-explainability for some classifiers. The ranking of accuracy, shown on the y-axis, of the classifiers was based on the average AUC and the Dunn's multi-comparison test presented in Sections 6.2.3 and 6.3.3. The LGBM, ANN and RF outperformed the other classifiers for case study 1. However, LGBM, LR, LDA and SVM outperformed the other classifiers for case study 2. Furthermore, the AUCs of the top performing classifiers for case one are on average higher than those of case study 2. The degree of explainability, shown on the x-axis, was determined by two factors: the intrinsic explainability and the ease of interpretation of the SHAP dependence plots. The DT, LR and LDA rank highest in terms of explainability, with the DT ranking highest because the feature importance, interactions and predictions can be depicted using a diagram. The bagging classifier ranked lowest in terms of explainability for case study 2. The is because the trends in the SHAP dependence plots are not clear.

The outcomes of the applications indicate that there is no single credit classifier that outperforms the others and the outcome depends on the datasets in question. The results also suggest that SHAP outputs are intuitive and enhance understanding and trust in black box models. Furthermore, SHAP outcomes are fairly consistent with outputs of transparent models. The local explanations provided by LIME provide a way to explain reasons behind predictions for individual credit applicants. The latter is imperative for regulatory and legal requirements. LIME computation is more efficient for instance level explanations compared to SHAP, since the computation time of LIME using Python is significantly lower than that of SHAP. LIME produces local explanations almost instantly, making it ideal for practical purposes.

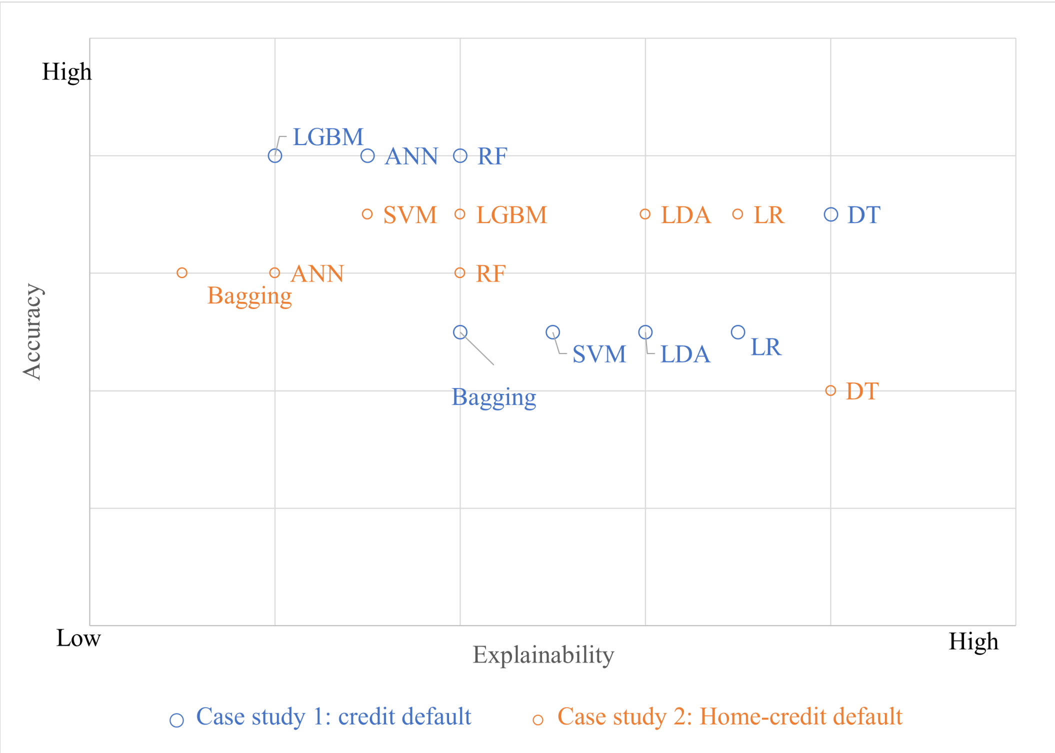

Figure 19: Accuracy-explainability trade-off of credit scoring classifiers applied in case study 1 and 2.

This research compliments previous research on the accuracy of various classification models used in credit and the explainability of these models. The difficulty in explainability and legal requirements or black box perception of classifiers has resulted in the reluctance to adopt and utilise these models in practice. Therefore, the contribution of this research project is to instil confidence in the use of best performing classifiers irrespective of whether the classifier is deemed as a black box or not.

7.2 Recommendations for future work

This research has also demonstrated the advantages and effectiveness of alternative approaches to credit risk scoring. The classification techniques, namely, ANN, bagging, LGBM, SVM and, RF were tested and outcomes were compared against the popular transparent methods DT, LDA and LR.

Literature shows that MCSs are a growing area of research and show promising results. The MCS used by Nalić et al. [2020] is both robust and interpretable, making it ideal to be used for credit scoring. This paper focused on certain MCSs, such as bagging and boosting. There is more future work on other MCSs, such as blending and stacking used by Wang et al. [2011], which can be extended to use interpretable base classifiers.

This research has also demonstrated the effectiveness of SHAP and LIME to explain predictions of black box classifiers. This approach has shown to be useful for both global and local explanations. The areas that have been identified for future research on SHAP include the following:

Current approaches use the mean absolute SHAP values of features to rank the importance. A limitation with this approach is that outliers may have an impact on the mean absolute value and this can in turn have an impact of feature importance. Furthermore, there are cases where the mean absolute values are close to each other which makes it difficult to determine which feature is more important. Although this approach is widely accepted, much work is required to ensure that conclusions are not incorrectly interpreted. An extension of the work on feature importance is to include significance tests, confidence intervals, error measures and pairwise comparisons of the features importance values.

Two approaches were employed to determine SHAP values. Kernel SHAP was used for ANN, bagging, SVM, LDA and LR. Whereas, tree SHAP was used for LGBM, DT and RF. While the kernel SHAP is an improvement to the classic methods of calculating SHAP values, it is still inefficient in terms of the time it takes to compute SHAP values [Misheva et al., 2021]. Tree SHAP is very efficient as it computes SHAP values quickly, however the algorithm is only applicable to decision tree based algorithms. Further work is required to enhance the efficiency of calculating SHAP values for linear classifiers and some ensembles.

The visualisations of SHAP values computed using kernel SHAP sometimes lack useful insights. The dependence plots sometimes fail to show trends that are easily interpretable and therefore defeat the purpose of interpretability. This is possibly due to outliers in SHAP values or the internal computational process. To obtain SHAP values with kernel SHAP, a reasonable sample must be used, which can impact the clarity of the resulting visualisations. Further research is necessary to enhance the quality of plots derived from kernel SHAP calculations.

Bibliography

H.A. Abdou and J. Pointon. Credit scoring, statistical techniques and evaluation criteria: a review of the literature. *Intelligent Systems in Accounting, Finance & Management*, 18(2-3), 2011. doi:https://doi.org/10.1002/isaf.325.

J. Abellán and J.G. Castellano. A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications*, 73:1–10, 2017. doi:https://doi.org/10.1016/j.eswa.2016.12.020.

H.A Alaka, L.O. Oyedele, H.A. Owolabi, V. Kumar, S.O. Ajayi, O.O. Akinade, and M. Bilal. Systematic review of bankruptcy prediction models: Towards a framework for tool selection. *Expert Systems with Applications*, 94:164–184, 2018. doi:https://doi.org/10.1016/j.eswa.2017.10.040.

M. Ala'raj and M.F. Abbod. Classifiers consensus system approach for credit scoring. *Knowledge-Based Systems*, 104:89–105, 2016. doi:https://doi.org/10.1016/j.knosys.2016.04.013.

C.J. Anil Kumar, B.K. Raghavendra, and S. Raghavendra. A credit scoring heterogeneous ensemble model using stacking and voting. *Indian Journal of Science and Technology*, 15(7):300–308, 2022. doi:https://doi.org/10.17485/IJST/v15i7.1715.

A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. doi:https://doi.org/10.1016/j.inffus.2019.12.012.

Board of Governors of the Federal Reserve System. Guidance on Model Risk Management. SR Letter 11-7, 2011. URL https://www.federalreserve.gov/boarddocs/srletters/2011/sr1107a1.pdf.

S.D. Brown and A.J. Myles. Decision tree modeling in classification. *Chemical and Biochemical Data Analysis*, 3:541–569, 2013. doi:https://doi.org/10.1016/B978-0-12-409547-2.00653-3.

M. Bücker, G. Szepannek, A. Gosiewska, and P. Biecek. Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society*, 73(1):70–90, 2022. doi:https://doi.org/10.1080/01605682.2021.1922098.

Y.-C. Chang, K.-H. Chang, and G.-J. Wu. Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Applied Soft Computing Journal*, 73:914–920, 2018. doi:https://doi.org/10.1016/j.asoc.2018.09.029.

A. Chopra and P. Bhilare. Application of ensemble models in credit scoring models. *Business Perspectives and Research*, 6(2):129–141, 2018. doi:https://doi.org/10.1177/2278533718765531.

C. Croux and C. Dehon. Influence functions of the spearman and kendall correlation measures. *tatistical Methods & Applications*, 19:497–515, 2010. URL `https://doi.org/10.1007/s10260-010-0142-z`.

P. Dangeti. *Statistics for Machine Learning*. Packt Publishing, 2017.

J.I. Daoud. Multicollinearity and regression analysis. *Journal of Physics: Conference Series*, 949(1):012009, 2017. doi:https://doi.org/10.1088/1742-6596/949/1/012009.

X. Dastile, T. Celik, and M. Potsane. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, 91:106263, 2020. doi:https://doi.org/10.1016/j.asoc.2020.106263.

A. de Servigny and O. Renault. *Measuring and Managing Credit Risk*. McGraw-Hill, 2004.

K. Demertzis, K. Kostinakis, K. Morfidis, and L. Iliadis. An interpretable machine learning method for the prediction of R/C buildings' seismic response. *Journal of Building Engineering*, 63:105493, 2023. doi:https://doi.org/10.1016/j.jobe.2022.105493.

R. ElShawi, Y. Sherif, M. Al-Mallah, and S. Sakr. Ilime: Local and global interpretable model-agnostic explainer of black-box decision. In *European Conference on Advances in Databases and Information Systems*, pages 53–68, 2019.

S. Finlay. *Credit Scoring, Response Modelling and Insurance Rating: A Practical Guide to Forecasting Consumer Behaviour*. Palgrave Macmillan, 2010.

Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1): 119–139, 1997. doi:https://doi.org/10.1006/jcss.1997.1504.

J.H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi:https://doi.org/10.1214/aos/1013203451.

FSB. Artificial intelligence and machine learning in financial services: Market developments and financial stability implications. *Financial Stability Board*, 2017. URL https://www.fsb.org/wp-content/uploads/P011117.pdf.

M.A. Ganaie, M. Hu, A.K. Malik, M. Tanveer, and P.N. Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022. doi:https://doi.org/10.1016/j.engappai.2022.105151.

A. Ghodselahi. A hybrid support vector machine ensemble model for credit scoring. *International Journal of Computer Applications*, 17(5):1–5, 2011. doi:https://doi.org/10.5120/2220-2829.

L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018. doi:https://doi.org/10.1109/DSAA.2018.00018.

R. Y. Goh and L. S. Lee. Credit scoring: A review on support vector machines and metaheuristic approaches. *Advances in Operations Research*, 2019. doi:https://doi.org/10.1155/2019/1974794.

A. Gramegna and P. Giudici. SHAP and LIME: An evaluation of discriminative power in credit risk. *Frontiers in Artificial Intelligence*, 4, 2021. doi:https://doi.org/10.3389/frai.2021.752558.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), 2019. doi:10.1145/3236009.

S. Han, H. Kim, and Y-S. Lee. Double random forest. *Machine Learning*, 109: 1569–1586, 2020. URL https://doi.org/10.1007/s10994-020-05889-1.

J.T. Hancock and T.M. Khoshgoftaar. Survey on categorical data for neural networks. *Journal of Big Data*, 7, 2020. doi:https://doi.org/10.1186/s40537-020-00305-w.

Home Credit Group. Home Credit Default Risk, 2018a. URL https://www.kaggle.com/competitions/home-credit-default-risk/data.

Home Credit Group. Home Credit Default Risk, 2018b. URL https://www.kaggle.com/c/home-credit-default-risk/rules.

G. James, D. Witten, T Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.

A. Jović, K. Brkić, and N. Bogunović. A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205, 2015. doi:https://doi.org/10.1109/MIPRO.2015.7160458.

M.R. Karim. Interpreting black-box machine learning models with decision rules and knowledge graph reasoning, 2022. URL `https://publications.rwth-aachen.de/record/850613`.

B. Kollár, I. Weissová, and A. Siekelová. Comparative analysis of theoretical aspects in credit risk models. *Procedia Economics and Finance*, 24:331–338, 2015. doi:https://doi.org/10.1016/S2212-5671(15)00673-5.

M. Kuhn and K. Johnson. *Applied Predictive Modelling*. Springer, New York, 2013. doi:https://doi.org/10.1007/978-1-4614-6849-3.

S.K. Kwak and J.H. Kim. Statistical data preparation: management of missing values and outliers. *Korean Journal of Anesthesiology*, 70(4), 2017. doi:https://doi.org/10.4097/kjae.2017.70.4.407.

S. Lessmann, B. Baesens, H.-V. Seow, and L.C. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015. doi:https://doi.org/10.1016/j.ejor.2015.05.030.

W.-Y. Loh. Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, 1(1):14–23, 2011. doi:https://doi.org/10.1002/widm.8.

S. Lundberg and S. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 4768–4777. Curran Associates Inc., 2017. ISBN 9781510860964.

A.F. Markus, J.A. Kors, and P.R. Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, 113:103655, 2021. doi:https://doi.org/10.1016/j.jbi.2020.103655.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. doi:https://doi.org/10.1016/j.artint.2018.07.007.

B.H. Misheva, J. Osterrieder, A. Hirsa, O. Kulkarni, and S. Fung Lin. Explainable AI in credit risk management, 2021. URL `https://doi.org/10.48550/arXiv.2103.00949`.

C. Molnar. *Interpretable Machine Learning*. 2 edition, 2022. URL https://christophm.github.io/interpretable-ml-book.

J. Nalić, G. Martinović, and D. Žagar. New hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers. *Advanced Engineering Informatics*, 45:101130, 2020. doi:https://doi.org/10.1016/j.aei.2020.101130.

A. Natekin and A. Knoll. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7, 2013. doi:https://doi.org/10.3389/fnbot.2013.00021.

R. Nisbet, J. Elder, and G. Miner. Chapter 11 - Classification. In *Handbook of Statistical Analysis and Data Mining Applications*. Academic Press, 2009.

S. Oreski and G. Oreski. Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert Systems with Applications*, 41(4):2052–2064, 2014. doi:https://doi.org/10.1016/j.eswa.2013.09.004.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

G. Plumb, D. Molitor, and A. Talwalkar. Model agnostic supervised local explanations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 2520–2529. Curran Associates Inc., 2018.

E. Rendón, R. Alejo, C. Castorena, F.J. Isidro-Ortega, and E.E. Granda-Gutiérrez. Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences*, 10(4), 2020. doi:https://doi.org/10.3390/app10041276.

M.T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1135–1144. Association for Computing Machinery, 2016. doi:https://doi.org/10.1145/2939672.2939778.

M.T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. doi:10.1609/aaai.v32i1.11491. URL https://ojs.aaai.org/index.php/AAAI/article/view/11491.

V.F. Rodriguez-Galiano, J.A. Luque-Espinar, M. Chica-Olmo, and M.P. Mendes. Feature selection approaches for predictive modelling of groundwater nitrate pollution: An evaluation of filters, embedded and wrapper methods. *Science of The Total Environment*, 624:661–672, 2018. doi:https://doi.org/10.1016/j.scitotenv.2017.12.152.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, (1):206–215, 2019. doi:https://doi.org/10.1038/s42256-019-0048-x.

C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022. doi:https://doi.org/10.1214/21-SS133.

Z. Runchi, X. Liguo, and W. Qin. An ensemble credit scoring model based on logistic regression with heterogeneous balancing and weighting effects. *Expert Systems with Applications*, 212:118732, 2023. doi:https://doi.org/10.1016/j.eswa.2022.118732.

H. Sadok, F. Sakka, and M. El Maknouzi. Artificial intelligence and bank credit analysis: A review. *Cogent Economics & Finance*, 10(1):2023262, 2022. doi:https://doi.org/10.1080/23322039.2021.2023262.

W. Saeed and C. Omlin. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273, 2023. doi:https://doi.org/10.1016/j.knosys.2023.110273.

A. Saranya and R. Subhashini. A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends. *Decision Analytics Journal*, 7:100230, 2023. doi:https://doi.org/10.1016/j.dajour.2023.100230.

R.E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 2009.

L.C. Thomas. A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting*, 16(2):149–172, 2000. doi:https://doi.org/10.1016/S0169-2070(00)00034-0.

L.C. Thomas, D.B. Edelman, and J.N. Crook. *Credit Scoring and Its Applications*. Society for Industrial And Applied Mathematics, Philadelphia, PA, 2002.

S.K. Trivedi. A study on credit scoring modeling with different feature selection and machine learning approaches. *Technology in Society*, 63:101413, 2020. doi:https://doi.org/10.1016/j.techsoc.2020.101413.

T. van Gestel and B. Baesens. *Credit Risk Management Basic Concepts: financial risk components, rating analysis, models, economic and regulatory capital*. Oxford University Press, 2008. doi:https://doi.org/10.1093/acprof:oso/9780199545117.001.0001.

T. van Gestel, B. Baesens, P. van Dijcke, J.A.K. Suykens, J. Garcia, and T. Alderweireld. Linear and non-linear credit scoring by combining logistic regression and support vector machines. *Journal of Credit Risk*, 1(4), 2005. doi:https://doi.org/10.21314/JCR.2005.025.

B. van Giffen, D. Herhausen, and D. Fahse. Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods. *Journal of Business Research*, 144:93–106, 2022. doi:https://doi.org/10.1016/j.jbusres.2022.01.076.

G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo. Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society*, 73(1):91–101, 2022. doi:https://doi.org/10.1080/01605682.2020.1865846.

G. Wang, J. Hao, J. Ma, and H. Jiang. A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, 38(1):223–230, 2011. doi:https://doi.org/10.1016/j.eswa.2010.06.048.

J. Witzany. *Credit Risk Managment: Pricing, Measurement, and Modelling*. Springer, 2017.

J. Yao, Z. Wang, L. Wang, M. Liu, H. Jiang, and Y. Chen. Novel hybrid ensemble credit scoring model with stacking-based noise detection and weight assignment. *Expert Systems with Applications*, 198:116913, 2022. doi:https://doi.org/10.1016/j.eswa.2022.116913.

I.-C. Yeh. Default of credit card clients. UCI Machine Learning Repository, 2016. URL `https://doi.org/10.24432/C55S3H`.

W. Zang, P. Zhang, C. Zhou, and L. Guo. Comparative study between incremental and ensemble learning on data streams: Case study. *Journal of Big Data*, 1(5), 2014. doi:https://doi.org/10.1186/2196-1115-1-5.

B. Zhu, W. Yang, H. Wang, and Y. Yuan. A hybrid deep learning model for consumer credit scoring. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 205–208, 2018. doi:https://doi.org/10.1109/ICAIBD.2018.8396195.

H. Zhu, P.A. Beling, and G.A. Overstreet. A study in the combination of two consumer credit scores. *Journal of the Operational Research Society*, 52(9):974–980, 2001. doi:https://doi.org/10.1057/palgrave.jors.2601225.

APPENDIX A

Univariate analysis for case study 1

Table A.1: Univariate analysis of limit of applicants.

Attribute	**Goods**	**Bads**	**Total**	**%total**	**Bad Rate**	**G:B odds**	**WoE**	**IV**
(9999, 30000]	1463	2618	4081	13.60	35.85	1.79	-0.68	0.07
(30000, 50000]	977	2618	3595	11.98	27.18	2.68	-0.27	0.01
(50000, 70000]	443	1113	1556	5.19	28.47	2.51	-0.34	0.01
(70000, 100000]	801	2465	3266	10.89	24.53	3.08	-0.13	0.00
(100000, 140000]	638	2154	2792	9.31	22.85	3.38	-0.04	0.00
(140000, 180000]	578	2753	3331	11.10	17.35	4.76	0.30	0.01
(180000, 210000]	436	2051	2487	8.29	17.53	4.70	0.29	0.01
(210000, 270000]	478	2456	2934	9.78	16.29	5.14	0.38	0.01
(270000, 360000]	528	2954	3482	11.61	15.16	5.59	0.46	0.02
(360000, 1000000]	294	2182	2476	8.25	11.87	7.42	0.75	0.04
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.18**

Table A.2: Univariate analysis of education of applicants.

Attribute	**Goods**	**Bads**	**Total**	**%total**	**Bad Rate**	**G:B odds**	**WoE**	**IV**
(-1, 1]	2036	8563	10599	35.33	19.21	4.21	0.18	0.01
(1, 2]	3330	10700	14030	46.77	23.73	3.21	-0.09	0.00
(2, 3]	1237	3680	4917	16.39	25.16	2.97	-0.17	0.00
(3, 6]	33	421	454	1.51	7.27	12.76	1.29	0.02
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.04**

Table A.3: Univariate analysis of marital status of applicants.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-1, 1]	3211	10502	13713	45.71	23.42	3.27	-0.07	0.00
(1, 2]	3341	12623	15964	53.21	20.93	3.78	0.07	0.00
(2, 3]	84	239	323	1.08	26.01	2.85	-0.21	0.00
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.01**

Table A.4: Univariate a analysis of age of applicants.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(20, 25]	1032	2839	3871	12.90	26.66	2.75	-0.25	0.01
(25, 27]	566	2167	2733	9.11	20.71	3.83	0.08	0.00
(27, 29]	599	2415	3014	10.05	19.87	4.03	0.14	0.00
(29, 31]	503	2109	2612	8.71	19.26	4.19	0.17	0.00
(31, 34]	671	2795	3466	11.55	19.36	4.17	0.17	0.00
(34, 37]	709	2553	3262	10.87	21.74	3.60	0.02	0.00
(37, 40]	580	2188	2768	9.23	20.95	3.77	0.07	0.00
(40, 43]	520	1768	2288	7.63	22.73	3.40	-0.03	0.00
(43, 49]	778	2528	3306	11.02	23.53	3.25	-0.08	0.00
(49, 79]	678	2002	2680	8.93	25.30	2.95	-0.18	0.00
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.02**

Table A.5: Univariate analysis of repayment status in September 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-3, -1]	1319	7126	8445	28.15	15.62	5.40	0.43	0.05
(-1, 0]	1888	12849	14737	49.12	12.81	6.81	0.66	0.17
(0, 1]	1252	2436	3688	12.29	33.95	1.95	-0.59	0.05
(1, 2]	1844	823	2667	8.89	69.14	0.45	-2.07	0.50
(2, 8]	333	130	463	1.54	71.92	0.39	-2.20	0.10
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.87**

Table A.6: Univariate analysis of repayment status in August 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-3, -1]	1657	8175	9832	32.77	16.85	4.93	0.34	0.03
(-1, 0]	2503	13227	15730	52.43	15.91	5.28	0.41	0.08
(0, 2]	2189	1766	3955	13.18	55.35	0.81	-1.47	0.37
(2, 8]	287	196	483	1.61	59.42	0.68	-1.64	0.06
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.54**

Table A.7: Univariate analysis of repayment status in July 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-3, -1]	1683	8340	10023	33.41	16.79	4.96	0.34	0.04
(-1, 0]	2751	13013	15764	52.55	17.45	4.73	0.30	0.04
(0, 2]	1970	1853	3823	12.74	51.53	0.94	-1.32	0.29
(2, 8]	232	158	390	1.30	59.49	0.68	-1.64	0.05
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.41**

Table A.8: Univariate analysis of repayment status in June 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-3, -1]	1741	8294	10035	33.45	17.35	4.76	0.30	0.03
(-1, 0]	3016	13439	16455	54.85	18.33	4.46	0.24	0.03
(0, 2]	1654	1507	3161	10.54	52.33	0.91	-1.35	0.25
(2, 8]	225	124	349	1.16	64.47	0.55	-1.85	0.05
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.36**

Table A.9: Univariate analysis of repayment status in May 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-3, -1]	1792	8293	10085	33.62	17.77	4.63	0.27	0.02
(-1, 0]	3195	13752	16947	56.49	18.85	4.30	0.20	0.02
(0, 8]	1649	1319	2968	9.89	55.56	0.80	-1.48	0.28
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.33**

Table A.10: Univariate analysis of Repayment status in April 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-3, -1]	1956	8679	10635	35.45	18.39	4.44	0.23	0.02
(-1, 0]	3069	13217	16286	54.29	18.84	4.31	0.20	0.02
(0, 2]	1401	1365	2766	9.22	50.65	0.97	-1.28	0.20
(2, 8]	210	103	313	1.04	67.09	0.49	-1.97	0.05
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.29**

Table A.11: Univariate analysis of amount of bill statement in September 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-165581, 279]	733	2267	3000	10.00	24.43	3.09	-0.13	0.00
(279, 1893]	665	2335	3000	10.00	22.17	3.51	-0.00	0.00
(1893, 6050]	618	2382	3000	10.00	20.60	3.85	0.09	0.00
(6050, 13469]	663	2337	3000	10.00	22.10	3.52	0.00	0.00
(13469, 22382]	766	2234	3000	10.00	25.53	2.92	-0.19	0.00
(22382, 37045]	721	2279	3000	10.00	24.03	3.16	-0.11	0.00
(37045, 52205]	659	2341	3000	10.00	21.97	3.55	0.01	0.00
(52205, 83421]	627	2373	3000	10.00	20.90	3.78	0.07	0.00
(83421, 142134]	590	2410	3000	10.00	19.67	4.08	0.15	0.00
(142134, 964511]	594	2406	3000	10.00	19.80	4.05	0.14	0.00
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.01**

Table A.12: Univariate analysis of amount of bill statement in August 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-69778, 0]	744	2431	3175	10.58	23.43	3.27	-0.07	0.00
(0, 1473]	623	2202	2825	9.42	22.05	3.53	0.00	0.00
(1473, 5500]	613	2388	3001	10.00	20.43	3.90	0.10	0.00
(5500, 12800]	642	2357	2999	10.00	21.41	3.67	0.04	0.00
(12800, 21200]	772	2228	3000	10.00	25.73	2.89	-0.20	0.00
(21200, 34774]	738	2262	3000	10.00	24.60	3.07	-0.14	0.00
(34774, 50690]	664	2337	3001	10.00	22.13	3.52	-0.00	0.00
(50690, 80292]	635	2364	2999	10.00	21.17	3.72	0.06	0.00
(80292, 136906]	600	2400	3000	10.00	20.00	4.00	0.13	0.00
(136906, 983931]	605	2395	3000	10.00	20.17	3.96	0.12	0.00
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.01**

Table A.13: Univariate analysis of amount of bill statement in July 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-157265, 0]	819	2706	3525	11.75	23.23	3.30	-0.06	0.00
(0, 1188]	552	1923	2475	8.25	22.30	3.48	-0.01	0.00
(1188, 5219]	609	2391	3000	10.00	20.30	3.93	0.11	0.00
(5219, 12197]	625	2375	3000	10.00	20.83	3.80	0.08	0.00
(12197, 20088]	749	2251	3000	10.00	24.97	3.01	-0.16	0.00
(20088, 31401]	732	2269	3001	10.00	24.39	3.10	-0.13	0.00
(31401, 49217]	703	2296	2999	10.00	23.44	3.27	-0.08	0.00
(49217, 76777]	647	2353	3000	10.00	21.57	3.64	0.03	0.00
(76777, 132051]	603	2397	3000	10.00	20.10	3.98	0.12	0.00
(132051, 1664089]	597	2403	3000	10.00	19.90	4.03	0.13	0.00
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.01**

Table A.14: Univariate analysis of amount of bill statement in June 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-170001, 0]	899	2971	3870	12.90	23.23	3.30	-0.06	0.00
(0, 988]	496	1635	2131	7.10	23.28	3.30	-0.07	0.00
(988, 4644]	589	2410	2999	10.00	19.64	4.09	0.15	0.00
(4644, 11145]	594	2407	3001	10.00	19.79	4.05	0.14	0.00
(11145, 19052]	721	2280	3001	10.00	24.03	3.16	-0.11	0.00
(19052, 28604]	743	2255	2998	9.99	24.78	3.03	-0.15	0.00
(28604, 45457]	710	2290	3000	10.00	23.67	3.23	-0.09	0.00
(45457, 70579]	652	2349	3001	10.00	21.73	3.60	0.02	0.00
(70579, 122419]	620	2379	2999	10.00	20.67	3.84	0.09	0.00
(122419, 891586]	612	2388	3000	10.00	20.40	3.90	0.10	0.00
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.01**

Table A.15: Univariate analysis of amount of bill statement in May 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-81335, 0]	995	3166	4161	13.87	23.91	3.18	-0.10	0.00
(0, 763]	412	1428	1840	6.13	22.39	3.47	-0.02	0.00
(763, 3637]	585	2415	3000	10.00	19.50	4.13	0.16	0.00
(3637, 9809]	570	2429	2999	10.00	19.01	4.26	0.19	0.00
(9809, 18104]	702	2298	3000	10.00	23.40	3.27	-0.07	0.00
(18104, 26690]	758	2242	3000	10.00	25.27	2.96	-0.17	0.00
(26690, 40943]	721	2279	3000	10.00	24.03	3.16	-0.11	0.00
(40943, 65823]	662	2338	3000	10.00	22.07	3.53	0.00	0.00
(65823, 115883]	613	2387	3000	10.00	20.43	3.89	0.10	0.00
(115883, 927171]	618	2382	3000	10.00	20.60	3.85	0.09	0.00
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.01**

Table A.16: Univariate analysis of amount of bill statement in April 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-339604, 0]	1086	3622	4708	15.69	23.07	3.34	-0.05	0.00
(0, 476]	300	1001	1301	4.34	23.06	3.34	-0.05	0.00
(476, 2702]	616	2375	2991	9.97	20.60	3.86	0.09	0.00
(2702, 8770]	515	2485	3000	10.00	17.17	4.83	0.32	0.01
(8770, 17071]	684	2316	3000	10.00	22.80	3.39	-0.04	0.00
(17071, 25508]	781	2219	3000	10.00	26.03	2.84	-0.21	0.00
(25508, 39252]	742	2258	3000	10.00	24.73	3.04	-0.15	0.00
(39252, 63151]	656	2344	3000	10.00	21.87	3.57	0.01	0.00
(63151, 112110]	654	2346	3000	10.00	21.80	3.59	0.02	0.00
(112110, 961664]	602	2398	3000	10.00	20.07	3.98	0.12	0.00
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.02**

Table A.17: Univariate analysis of amount of previous payment in September 2005.

Attribute	**Goods**	**Bads**	**Total**	**%total**	**Bad Rate**	**G:B odds**	**WoE**	**IV**
(-1, 316]	2054	3948	6002	20.01	34.22	1.92	-0.61	0.09
(316, 1264]	679	2319	2998	9.99	22.65	3.42	-0.03	0.00
(1264, 1724]	684	2319	3003	10.01	22.78	3.39	-0.04	0.00
(1724, 2100]	652	2358	3010	10.03	21.66	3.62	0.03	0.00
(2100, 3000]	680	2423	3103	10.34	21.91	3.56	0.01	0.00
(3000, 4309]	601	2283	2884	9.61	20.84	3.80	0.08	0.00
(4309, 6192]	471	2529	3000	10.00	15.70	5.37	0.42	0.02
(6192, 10300]	432	2571	3003	10.01	14.39	5.95	0.52	0.02
(10300, 873552]	383	2614	2997	9.99	12.78	6.83	0.66	0.04
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.16**

Table A.18: Univariate analysis of amount of previous payment in August 2005.

Attribute	**Goods**	**Bads**	**Total**	**%total**	**Bad Rate**	**G:B odds**	**WoE**	**IV**
(-1, 269]	1960	4040	6000	20.00	32.67	2.06	-0.54	0.07
(269, 1165]	684	2319	3003	10.01	22.78	3.39	-0.04	0.00
(1165, 1600]	785	2320	3105	10.35	25.28	2.96	-0.18	0.00
(1600, 2009]	634	2263	2897	9.66	21.88	3.57	0.01	0.00
(2009, 3000]	743	2800	3543	11.81	20.97	3.77	0.07	0.00
(3000, 4045]	505	1947	2452	8.17	20.60	3.86	0.09	0.00
(4045, 6000]	544	2518	3062	10.21	17.77	4.63	0.27	0.01
(6000, 10401]	447	2491	2938	9.79	15.21	5.57	0.46	0.02
(10401, 1684259]	334	2666	3000	10.00	11.13	7.98	0.82	0.05
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.15**

Table A.19: Univariate analysis of amount of previous payment in July 2005.

Attribute	**Goods**	**Bads**	**Total**	**%total**	**Bad Rate**	**G:B odds**	**WoE**	**IV**
(-1, 3]	1939	4061	6000	20.00	32.32	2.09	-0.52	0.06
(3, 780]	710	2346	3056	10.19	23.23	3.30	-0.06	0.00
(780, 1206]	659	2286	2945	9.82	22.38	3.47	-0.01	0.00
(1206, 1800]	705	2306	3011	10.04	23.41	3.27	-0.07	0.00
(1800, 2500]	691	2511	3202	10.67	21.58	3.63	0.03	0.00
(2500, 3560]	546	2240	2786	9.29	19.60	4.10	0.15	0.00
(3560, 5284]	511	2489	3000	10.00	17.03	4.87	0.32	0.01
(5284, 10000]	511	2615	3126	10.42	16.35	5.12	0.37	0.01
(10000, 896040]	364	2510	2874	9.58	12.67	6.90	0.67	0.04
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.12**

Table A.20: Univariate analysis of amount of previous payment in June 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-1, 500]	2552	6481	9033	30.11	28.25	2.54	-0.33	0.04
(500, 1000]	906	2843	3749	12.50	24.17	3.14	-0.12	0.00
(1000, 1500]	568	1808	2376	7.92	23.91	3.18	-0.10	0.00
(1500, 2100]	616	2234	2850	9.50	21.61	3.63	0.03	0.00
(2100, 3200]	563	2469	3032	10.11	18.57	4.39	0.22	0.00
(3200, 5000]	558	2602	3160	10.53	17.66	4.66	0.28	0.01
(5000, 9571]	467	2333	2800	9.33	16.68	5.00	0.35	0.01
(9571, 621000]	406	2594	3000	10.00	13.53	6.39	0.60	0.03
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.09**

Table A.21: Univariate analysis of amount of previous payment in May 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-1, 500]	2525	6591	9116	30.39	27.70	2.61	-0.30	0.03
(500, 1000]	848	2746	3594	11.98	23.59	3.24	-0.08	0.00
(1000, 1500]	548	1788	2336	7.79	23.46	3.26	-0.08	0.00
(1500, 2123]	655	2299	2954	9.85	22.17	3.51	-0.00	0.00
(2123, 3200]	603	2415	3018	10.06	19.98	4.00	0.13	0.00
(3200, 5000]	571	2604	3175	10.58	17.98	4.56	0.26	0.01
(5000, 9500]	504	2306	2810	9.37	17.94	4.58	0.26	0.01
(9500, 426529]	382	2615	2997	9.99	12.75	6.85	0.66	0.04
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.08**

Table A.22: Univariate analysis of amount of previous payment in April 2005.

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-1, 426]	2504	6498	9002	30.01	27.82	2.60	-0.31	0.03
(426, 1000]	968	3054	4022	13.41	24.07	3.15	-0.11	0.00
(1000, 1500]	547	1671	2218	7.39	24.66	3.05	-0.14	0.00
(1500, 2100]	615	2204	2819	9.40	21.82	3.58	0.02	0.00
(2100, 3200]	590	2392	2982	9.94	19.79	4.05	0.14	0.00
(3200, 5000]	586	2652	3238	10.79	18.10	4.53	0.25	0.01
(5000, 9600]	450	2270	2720	9.07	16.54	5.04	0.36	0.01
(9600, 528666]	376	2623	2999	10.00	12.54	6.98	0.68	0.04
Total	**6636**	**23364**	**30000**	**100.00**	**22.12**	**3.52**	**0.00**	**0.09**

APPENDIX B

Description of data for case study 2

Table B.1: Definition of features provided by Home Credit Group [2018a].

Table	Row	Description
application	SK_ID_CURR	ID of loan in our sample
application	TARGET	Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)
application	NAME_CONTRACT_TYPE	Identification if loan is cash or revolving
application	CODE_GENDER	Gender of the client
application	FLAG_OWN_CAR	Flag if the client owns a car
application	FLAG_OWN_REALTY	Flag if client owns a house or flat
application	CNT_CHILDREN	Number of children the client has
application	AMT_INCOME_TOTAL	Income of the client
application	AMT_CREDIT	Credit amount of the loan
application	AMT_ANNUITY	Loan annuity
application	AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given
application	NAME_TYPE_SUITE	Who was accompanying client when he was applying for the loan
application	NAME_INCOME_TYPE	Clients income type (businessman, working, maternity leave,. . .)
application	NAME_EDUCATION_TYPE	Level of highest education the client achieved
application	NAME_FAMILY_STATUS	Family status of the client
application	NAME_HOUSING_TYPE	What is the housing situation of the client (renting, living with parents, ...)

application	REGION_POPULATION_-RELATIVE	Normalized population of region where client lives (higher number means the client lives in more populated region)
application	DAYS_BIRTH	Client's age in days at the time of application
application	DAYS_EMPLOYED	How many days before the application the person started current employment
application	DAYS_REGISTRATION	How many days before the application did client change his registration
application	DAYS_ID_PUBLISH	How many days before the application did client change the identity document with which he applied for the loan
application	OWN_CAR_AGE	Age of client's car
application	FLAG_MOBIL	Did client provide mobile phone (1=YES, 0=NO)
application	FLAG_EMP_PHONE	Did client provide work phone (1=YES, 0=NO)
application	FLAG_WORK_PHONE	Did client provide home phone (1=YES, 0=NO)
application	FLAG_CONT_MOBILE	Was mobile phone reachable (1=YES, 0=NO)
application	FLAG_PHONE	Did client provide home phone (1=YES, 0=NO)
application	FLAG_EMAIL	Did client provide email (1=YES, 0=NO)
application	OCCUPATION_TYPE	What kind of occupation does the client have
application	CNT_FAM_MEMBERS	How many family members does client have
application	REGION_RATING_CLIENT	Our rating of the region where client lives (1,2,3)
application	REGION_RATING_-CLIENT_W_CITY	Our rating of the region where client lives with taking city into account (1,2,3)
application	WEEKDAY_APPR_PRO-CESS_START	On which day of the week did the client apply for the loan
application	HOUR_APPR_PROCESS_-START	Approximately at what hour did the client apply for the loan
application	REG_REGION_NOT_LIVE_-REGION	Flag if client's permanent address does not match contact address (1=different, 0=same, at region level)
application	REG_REGION_NOT_-WORK_REGION	Flag if client's permanent address does not match work address (1=different, 0=same, at region level)

application	LIVE_REGION_NOT_-WORK_REGION	Flag if client's contact address does not match work address (1=different, 0=same, at region level)
application	REG_CITY_NOT_LIVE_-CITY	Flag if client's permanent address does not match contact address (1=different, 0=same, at city level)
application	REG_CITY_NOT_WORK_-CITY	Flag if client's permanent address does not match work address (1=different, 0=same, at city level)
application	LIVE_CITY_NOT_WORK_-CITY	Flag if client's contact address does not match work address (1=different, 0=same, at city level)
application	ORGANIZATION_TYPE	Type of organization where client works
application	EXT_SOURCE_1	Normalized score from external data source
application	EXT_SOURCE_2	Normalized score from external data source
application	EXT_SOURCE_3	Normalized score from external data source
application	APARTMENTS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	BASEMENTAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	YEARS_BEGINEXPLUATA-TION_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

application	YEARS_BUILD_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	COMMONAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	ELEVATORS_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	ENTRANCES_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	FLOORSMAX_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	FLOORSMIN_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

application	LANDAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	LIVINGAPARTMENTS_-AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	LIVINGAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	NONLIVINGAPARTMENTS_-AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	NONLIVINGAREA_AVG	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	APARTMENTS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

application	BASEMENTAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	YEARS_BEGINEXPLUATATION_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	YEARS_BUILD_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	COMMONAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	ELEVATORS_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	ENTRANCES_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

application	FLOORSMAX_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	FLOORSMIN_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	LANDAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	LIVINGAPARTMENTS_-MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	LIVINGAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	NONLIVINGAPARTMENTS_-MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

application	NONLIVINGAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	APARTMENTS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	BASEMENTAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	YEARS_BEGINEXPLUATATION_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	YEARS_BUILD_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	COMMONAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

application	ELEVATORS_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	ENTRANCES_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	FLOORSMAX_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	FLOORSMIN_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	LANDAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	LIVINGAPARTMENTS_-MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

application	LIVINGAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	NONLIVINGAPARTMENTS_-MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	NONLIVINGAREA_MEDI	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	FONDKAPREMONT_-MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	HOUSETYPE_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	TOTALAREA_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

application	WALLSMATERIAL_MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	EMERGENCYSTATE_-MODE	Normalized information about building where the client lives, What is average (_AVG suffix), modus (_MODE suffix), median (_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor
application	OBS_30_CNT_SOCIAL_CIR-CLE	How many observation of client's social surroundings with observable 30 DPD (days past due) default
application	DEF_30_CNT_SOCIAL_CIR-CLE	How many observation of client's social surroundings defaulted on 30 DPD (days past due)
application	OBS_60_CNT_SOCIAL_CIR-CLE	How many observation of client's social surroundings with observable 60 DPD (days past due) default
application	DEF_60_CNT_SOCIAL_CIR-CLE	How many observation of client's social surroundings defaulted on 60 (days past due) DPD
application	DAYS_LAST_PHONE_-CHANGE	How many days before application did client change phone
application	FLAG_DOCUMENT_2	Did client provide document 2
application	FLAG_DOCUMENT_3	Did client provide document 3
application	FLAG_DOCUMENT_4	Did client provide document 4
application	FLAG_DOCUMENT_5	Did client provide document 5
application	FLAG_DOCUMENT_6	Did client provide document 6
application	FLAG_DOCUMENT_7	Did client provide document 7
application	FLAG_DOCUMENT_8	Did client provide document 8
application	FLAG_DOCUMENT_9	Did client provide document 9
application	FLAG_DOCUMENT_10	Did client provide document 10
application	FLAG_DOCUMENT_11	Did client provide document 11
application	FLAG_DOCUMENT_12	Did client provide document 12
application	FLAG_DOCUMENT_13	Did client provide document 13
application	FLAG_DOCUMENT_14	Did client provide document 14
application	FLAG_DOCUMENT_15	Did client provide document 15
application	FLAG_DOCUMENT_16	Did client provide document 16
application	FLAG_DOCUMENT_17	Did client provide document 17

application	FLAG_DOCUMENT_18	Did client provide document 18
application	FLAG_DOCUMENT_19	Did client provide document 19
application	FLAG_DOCUMENT_20	Did client provide document 20
application	FLAG_DOCUMENT_21	Did client provide document 21
application	AMT_REQ_CREDIT_BUREAU_HOUR	Number of enquiries to Credit Bureau about the client one hour before application
application	AMT_REQ_CREDIT_BUREAU_DAY	Number of enquiries to Credit Bureau about the client one day before application (excluding one hour before application)
application	AMT_REQ_CREDIT_BUREAU_WEEK	Number of enquiries to Credit Bureau about the client one week before application (excluding one day before application)
application	AMT_REQ_CREDIT_BUREAU_MON	Number of enquiries to Credit Bureau about the client one month before application (excluding one week before application)
application	AMT_REQ_CREDIT_BUREAU_QRT	Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application)
application	AMT_REQ_CREDIT_BUREAU_YEAR	Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application)
bureau	SK_ID_CURR	ID of loan in our sample - one loan in our sample can have 0,1,2 or more related previous credits in credit bureau
bureau	SK_BUREAU_ID	Recoded ID of previous Credit Bureau credit related to our loan (unique coding for each loan application)
bureau	CREDIT_ACTIVE	Status of the Credit Bureau (CB) reported credits
bureau	CREDIT_CURRENCY	Recoded currency of the Credit Bureau credit
bureau	DAYS_CREDIT	How many days before current application did client apply for Credit Bureau credit
bureau	CREDIT_DAY_OVERDUE	Number of days past due on CB credit at the time of application for related loan in our sample
bureau	DAYS_CREDIT_ENDDATE	Remaining duration of CB credit (in days) at the time of application in Home Credit

bureau	DAYS_ENDDATE_FACT	Days since CB credit ended at the time of application in Home Credit (only for closed credit)
bureau	AMT_CREDIT_MAX_-OVERDUE	Maximal amount overdue on the Credit Bureau credit so far (at application date of loan in our sample)
bureau	CNT_CREDIT_PROLONG	How many times was the Credit Bureau credit prolonged
bureau	AMT_CREDIT_SUM	Current credit amount for the Credit Bureau credit
bureau	AMT_CREDIT_SUM_DEBT	Current debt on Credit Bureau credit
bureau	AMT_CREDIT_SUM_LIMIT	Current credit limit of credit card reported in Credit Bureau
bureau	AMT_CREDIT_SUM_OVERDUE	Current amount overdue on Credit Bureau credit
bureau	CREDIT_TYPE	Type of Credit Bureau credit (Car, cash,...)
bureau	DAYS_CREDIT_UPDATE	How many days before loan application did last information about the Credit Bureau credit come
bureau	AMT_ANNUITY	Annuity of the Credit Bureau credit
bureau_balance	SK_BUREAU_ID	Recoded ID of Credit Bureau credit (unique coding for each application) - use this to join to CREDIT_BUREAU table
bureau_balance	MONTHS_BALANCE	Month of balance relative to application date (-1 means the freshest balance date)
bureau_balance	STATUS	Status of Credit Bureau loan during the month (active, closed, DPD0-30,... [C means closed, X means status unknown, 0 means no DPD, 1 means maximal did during month between 1-30, 2 means DPD 31-60,... 5 means DPD 120+ or sold or written off])
POS_CASH_balance	SK_ID_PREV	ID of previous credit in Home Credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit)
POS_CASH_balance	SK_ID_CURR	ID of loan in our sample

POS_CASH_balance	MONTHS_BALANCE	Month of balance relative to application date (-1 means the information to the freshest monthly snapshot, 0 means the information at application - often it will be the same as -1 as many banks are not updating the information to Credit Bureau regularly)
POS_CASH_balance	CNT_INSTALMENT	Term of previous credit (can change over time)
POS_CASH_balance	CNT_INSTALMENT_FUTURE	Installments left to pay on the previous credit
POS_CASH_balance	NAME_CONTRACT_STATUS	Contract status during the month
POS_CASH_balance	SK_DPD	DPD (days past due) during the month of previous credit
POS_CASH_balance	SK_DPD_DEF	DPD during the month with tolerance (debts with low loan amounts are ignored) of the previous credit
credit_card_balance	SK_ID_PREV	ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit)
credit_card_balance	SK_ID_CURR	ID of loan in our sample
credit_card_balance	MONTHS_BALANCE	Month of balance relative to application date (-1 means the freshest balance date)
credit_card_balance	AMT_BALANCE	Balance during the month of previous credit
credit_card_balance	AMT_CREDIT_LIMIT_ACTUAL	Credit card limit during the month of the previous credit
credit_card_balance	AMT_DRAWINGS_ATM_CURRENT	Amount drawing at ATM during the month of the previous credit
credit_card_balance	AMT_DRAWINGS_CURRENT	Amount drawing during the month of the previous credit
credit_card_balance	AMT_DRAWINGS_OTHER_CURRENT	Amount of other drawings during the month of the previous credit
credit_card_balance	AMT_DRAWINGS_POS_CURRENT	Amount drawing or buying goods during the month of the previous credit
credit_card_balance	AMT_INST_MIN_REGULARITY	Minimal installment for this month of the previous credit
credit_card_balance	AMT_PAYMENT_CURRENT	How much did the client pay during the month on the previous credit
credit_card_balance	AMT_PAYMENT_TOTAL_CURRENT	How much did the client pay during the month in total on the previous credit

credit_card_balance	AMT_RECEIVABLE_PRINCIPAL	Amount receivable for principal on the previous credit
credit_card_balance	AMT_RECIVABLE	Amount receivable on the previous credit
credit_card_balance	AMT_TOTAL_RECEIVABLE	Total amount receivable on the previous credit
credit_card_balance	CNT_DRAWINGS_ATM_CURRENT	Number of drawings at ATM during this month on the previous credit
credit_card_balance	CNT_DRAWINGS_CURRENT	Number of drawings during this month on the previous credit
credit_card_balance	CNT_DRAWINGS_OTHER_CURRENT	Number of other drawings during this month on the previous credit
credit_card_balance	CNT_DRAWINGS_POS_CURRENT	Number of drawings for goods during this month on the previous credit
credit_card_balance	CNT_INSTALMENT_MATURE_CUM	Number of paid installments on the previous credit
credit_card_balance	NAME_CONTRACT_STATUS	Contract status (active signed,...) on the previous credit
credit_card_balance	SK_DPD	DPD (Days past due) during the month on the previous credit
credit_card_balance	SK_DPD_DEF	DPD (Days past due) during the month with tolerance (debts with low loan amounts are ignored) of the previous credit
previous_application	SK_ID_PREV	ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loan applications in Home Credit, previous application could, but not necessarily have to lead to credit)
previous_application	SK_ID_CURR	ID of loan in our sample
previous_application	NAME_CONTRACT_TYPE	Contract product type (Cash loan, consumer loan [POS] ,...) of the previous application
previous_application	AMT_ANNUITY	Annuity of previous application
previous_application	AMT_APPLICATION	For how much credit did client ask on the previous application
previous_application	AMT_CREDIT	Final credit amount on the previous application. This differs from AMT_APPLICATION in a way that the AMT_APPLICATION is the amount for which the client initially applied for, but during our approval process he could have received different amount - AMT_CREDIT

previous_application	AMT_DOWN_PAYMENT	Down payment on the previous application
previous_application	AMT_GOODS_PRICE	Goods price of good that client asked for (if applicable) on the previous application
previous_application	WEEKDAY_APPR_PROCESS_START	On which day of the week did the client apply for previous application
previous_application	HOUR_APPR_PROCESS_START	Approximately at what day hour did the client apply for the previous application
previous_application	FLAG_LAST_APPL_PER_CONTRACT	Flag if it was last application for the previous contract. Sometimes by mistake of client or our clerk there could be more applications for one single contract
previous_application	NFLAG_LAST_APPL_IN_DAY	Flag if the application was the last application per day of the client. Sometimes clients apply for more applications a day. Rarely it could also be error in our system that one application is in the database twice
previous_application	NFLAG_MICRO_CASH	Flag Micro finance loan
previous_application	RATE_DOWN_PAYMENT	Down payment rate normalized on previous credit
previous_application	RATE_INTEREST_PRIMARY	Interest rate normalized on previous credit
previous_application	RATE_INTEREST_PRIVILEGED	Interest rate normalized on previous credit
previous_application	NAME_CASH_LOAN_PURPOSE	Purpose of the cash loan
previous_application	NAME_CONTRACT_STATUS	Contract status (approved, cancelled, ...) of previous application
previous_application	DAYS_DECISION	Relative to current application when was the decision about previous application made
previous_application	NAME_PAYMENT_TYPE	Payment method that client chose to pay for the previous application
previous_application	CODE_REJECT_REASON	Why was the previous application rejected
previous_application	NAME_TYPE_SUITE	Who accompanied client when applying for the previous application
previous_application	NAME_CLIENT_TYPE	Was the client old or new client when applying for the previous application
previous_application	NAME_GOODS_CATEGORY	What kind of goods did the client apply for in the previous application

previous_application	NAME_PORTFOLIO	Was the previous application for CASH, POS, CAR, ...
previous_application	NAME_PRODUCT_TYPE	Was the previous application x-sell o walk-in
previous_application	CHANNEL_TYPE	Through which channel we acquired the client on the previous application
previous_application	SELLERPLACE_AREA	Selling area of seller place of the previous application
previous_application	NAME_SELLER_INDUSTRY	The industry of the seller
previous_application	CNT_PAYMENT	Term of previous credit at application of the previous application
previous_application	NAME_YIELD_GROUP	Grouped interest rate into small medium and high of the previous application
previous_application	PRODUCT_COMBINATION	Detailed product combination of the previous application
previous_application	DAYS_FIRST_DRAWING	Relative to application date of current application when was the first disbursement of the previous application
previous_application	DAYS_FIRST_DUE	Relative to application date of current application when was the first due supposed to be of the previous application
previous_application	DAYS_LAST_DUE_1ST_VERSION	Relative to application date of current application when was the first due of the previous application
previous_application	DAYS_LAST_DUE	Relative to application date of current application when was the last due date of the previous application
previous_application	DAYS_TERMINATION	Relative to application date of current application when was the expected termination of the previous application
previous_application	NFLAG_INSURED_ON_APPROVAL	Did the client requested insurance during the previous application
installments_payments	SK_ID_PREV	ID of previous credit in Home credit related to loan in our sample. (One loan in our sample can have 0,1,2 or more previous loans in Home Credit)
installments_payments	SK_ID_CURR	ID of loan in our sample
installments_payments	NUM_INSTALMENT_VERSION	Version of installment calendar (0 is for credit card) of previous credit. Change of installment version from month to month signifies that some parameter of payment calendar has changed

installments_pay-ments	NUM_INSTALMENT_NUM-BER	On which installment we observe payment
installments_pay-ments	DAYS_INSTALMENT	When the installment of previous credit was supposed to be paid (relative to application date of current loan)
installments_pay-ments	DAYS_ENTRY_PAYMENT	When was the installments of previous credit paid actually (relative to application date of current loan)
installments_pay-ments	AMT_INSTALMENT	What was the prescribed installment amount of previous credit on this installment
installments_pay-ments	AMT_PAYMENT	What the client actually paid on previous credit on this installment

APPENDIX C

Overview of data for case study 2

Table C.1 shows a descriptive overview of the datasets provided by Home Credit. It shows a description in terms of the means, standard deviations, minimum and maximum values, the 25th, 50th and 75th percentiles as well as data types and proportion of missing values. The table also shows the distribution of the data in terms of kurtosis and skewness.

Table C.1: A descriptive summary of the home credit default datasets

No.	Dataset	Feature	Data Type	count	Missing%	average	σ	minimum	25%	50%	75%	maximum	Skewness	kurtosis	outlier
1	application_train	NAME_CONTRACT_TYPE	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
2	application_train	CODE_GENDER	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
3	application_train	FLAG_OWN_CAR	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
4	application_train	FLAG_OWN_REALTY	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
5	application_train	CNT_CHILDREN	Integer	307 511	0.00	0.42	0.72	0.00	0.00	0.00	1.00	19.00	1.97	7.90	✓
6	application_train	AMT_INCOME_TOTAL	Float	307 511	0.00	168797.92	237123.15	25650.00	112500.00	147150.00	202500.00	117000000.00	391.56	191786.55	✓
7	application_train	AMT_CREDIT	Float	307 511	0.00	599026.00	402490.78	45000.00	270000.00	513531.00	808650.00	4050000.00	1.23	1.93	✓
8	application_train	AMT_ANNUITY	Float	307 499	0.00	27108.57	14493.74	1615.50	16524.00	24903.00	34596.00	258025.50	1.58	7.71	✓
9	application_train	AMT_GOODS_PRICE	Float	307 233	0.09	538396.21	369446.46	40500.00	238500.00	450000.00	679500.00	4050000.00	1.35	2.43	✓
10	application_train	NAME_TYPE_SUITE	String	306 219	0.42	-	-	-	-	-	-	-	-	-	-
11	application_train	NAME_INCOME_TYPE	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
12	application_train	NAME_EDUCATION_TYPE	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
13	application_train	NAME_FAMILY_STATUS	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
14	application_train	NAME_HOUSING_TYPE	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-

No.	Dataset	Feature	Data Type	count	Missing%	average	σ	minimum	25%	50%	75%	maximum	Skewness	kurtosis	outlier
15	application_train	REGION_POPULATION_RELATIVE	Float	307 511	0.00	0.02	0.01	0.00	0.01	0.02	0.03	0.07	1.49	3.26	✓
16	application_train	DAYS_BIRTH	Integer	307 511	0.00	-16037.00	4363.99	-25229.00	-19682.00	-15750.00	-12413.00	-7489.00	-0.12	-1.05	-
17	application_train	DAYS_EMPLOYED	Integer	307 511	0.00	63815.05	141275.77	-17912.00	-2760.00	-1213.00	-289.00	365243.00	1.66	0.77	✓
18	application_train	DAYS_REGISTRATION	Float	307 511	0.00	-4986.12	3522.89	-24672.00	-7479.50	-4504.00	-2010.00	0.00	-0.59	-0.32	✓
19	application_train	DAYS_ID_PUBLISH	Integer	307 511	0.00	-2994.20	1509.45	-7197.00	-4299.00	-3254.00	-1720.00	0.00	0.35	-1.11	-
20	application_train	OWN_CAR_AGE	Float	104582	65.99	12.06	11.94	0.00	5.00	9.00	15.00	91.00	2.75	9.21	✓
21	application_train	FLAG_MOBIL	Integer	307 511	0.00	-	-	-	-	-	-	-	-554.54	307 511.00	-
22	application_train	FLAG_EMP_PHONE	Integer	307 511	0.00	-	-	-	-	-	-	-	-1.66	0.77	-
23	application_train	FLAG_WORK_PHONE	Integer	307 511	0.00	-	-	-	-	-	-	-	1.50	0.26	-
24	application_train	FLAG_CONT_MOBILE	Integer	307 511	0.00	-	-	-	-	-	-	-	-23.08	530.74	-
25	application_train	FLAG_PHONE	Integer	307 511	0.00	-	-	-	-	-	-	-	0.97	-1.05	-
26	application_train	FLAG_EMAIL	Integer	307 511	0.00	-	-	-	-	-	-	-	3.83	12.69	-
27	application_train	OCCUPATION_TYPE	String	211 120	31.35	-	-	-	-	-	-	-	-	-	-
28	application_train	CNT_FAM_MEMBERS	Float	307 509	0.00	2.15	0.91	1.00	2.00	2.00	3.00	20.00	0.99	2.80	✓
29	application_train	REGION_RATING_CLIENT	Integer	307 511	0.00	2.05	0.51	1.00	2.00	2.00	2.00	3.00	0.09	0.80	✓
30	application_train	REGION_RATING_CLIENT_W_CITY	Integer	307 511	0.00	2.03	0.50	1.00	2.00	2.00	2.00	3.00	0.06	0.93	✓
31	application_train	WEEKDAY_APPR_PROCESS_START	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
32	application_train	HOUR_APPR_PROCESS_START	Integer	307 511	0.00	12.06	3.27	0.00	10.00	12.00	14.00	23.00	-0.03	-0.19	✓
33	application_train	REG_REGION_NOT_LIVE_REGION	Integer	307 511	0.00	-	-	-	-	-	-	-	7.94	61.05	-
34	application_train	REG_REGION_NOT_WORK_REGION	Integer	307 511	0.00	-	-	-	-	-	-	-	4.09	14.75	-
35	application_train	LIVE_REGION_NOT_WORK_REGION	Integer	307 511	0.00	-	-	-	-	-	-	-	4.65	19.64	-
36	application_train	REG_CITY_NOT_LIVE_CITY	Integer	307 511	0.00	-	-	-	-	-	-	-	3.14	7.88	-
37	application_train	REG_CITY_NOT_WORK_CITY	Integer	307 511	0.00	-	-	-	-	-	-	-	1.28	-0.36	-
38	application_train	LIVE_CITY_NOT_WORK_CITY	Integer	307 511	0.00	-	-	-	-	-	-	-	1.67	0.79	-
39	application_train	ORGANIZATION_TYPE	String	307 511	0.00	-	-	-	-	-	-	-	-	-	-
40	application_train	EXT_SOURCE_1	Float	134 133	56.38	0.50	0.21	0.01	0.33	0.51	0.68	0.96	-0.07	-0.97	-
41	application_train	EXT_SOURCE_2	Float	306 851	0.21	0.51	0.19	0.00	0.39	0.57	0.66	0.85	-0.79	-0.27	-
42	application_train	EXT_SOURCE_3	Float	246 546	19.83	0.51	0.19	0.00	0.37	0.54	0.67	0.90	-0.41	-0.66	-
43	application_train	APARTMENTS_AVG	Float	151 450	50.75	0.12	0.11	0.00	0.06	0.09	0.15	1.00	2.64	11.39	✓
44	application_train	BASEMENTAREA_AVG	Float	127 568	58.52	0.09	0.08	0.00	0.04	0.08	0.11	1.00	3.57	25.93	✓
45	application_train	YEARS_BEGINEXPLUATATION_AVG	Float	157 504	48.78	0.98	0.06	0.00	0.98	0.98	0.99	1.00	-15.52	248.18	✓
46	application_train	YEARS_BUILD_AVG	Float	103 023	66.50	0.75	0.11	0.00	0.69	0.76	0.82	1.00	-0.96	4.40	✓
47	application_train	COMMONAREA_AVG	Float	92 646	69.87	0.04	0.08	0.00	0.01	0.02	0.05	1.00	5.46	45.99	✓
48	application_train	ELEVATORS_AVG	Float	143 620	53.30	0.08	0.13	0.00	0.00	0.00	0.12	1.00	2.44	7.87	✓
49	application_train	ENTRANCES_AVG	Float	152 683	50.35	0.15	0.10	0.00	0.07	0.14	0.21	1.00	2.40	11.59	✓
50	application_train	FLOORSMAX_AVG	Float	154 491	49.76	0.23	0.14	0.00	0.17	0.17	0.33	1.00	1.23	2.43	✓
51	application_train	FLOORSMIN_AVG	Float	98 869	67.85	0.23	0.16	0.00	0.08	0.21	0.38	1.00	0.95	1.34	✓
52	application_train	LANDAREA_AVG	Float	124 921	59.38	0.07	0.08	0.00	0.02	0.05	0.09	1.00	4.46	34.74	✓
53	application_train	LIVINGAPARTMENTS_AVG	Float	97 312	68.35	0.10	0.09	0.00	0.05	0.08	0.12	1.00	3.04	16.49	✓
54	application_train	LIVINGAREA_AVG	Float	153 161	50.19	0.11	0.11	0.00	0.05	0.07	0.13	1.00	2.85	12.33	✓
55	application_train	NONLIVINGAPARTMENTS_AVG	Float	93 997	69.43	0.01	0.05	0.00	0.00	0.00	0.00	1.00	15.54	284.73	✓
56	application_train	NONLIVINGAREA_AVG	Float	137 829	55.18	0.03	0.07	0.00	0.00	0.00	0.03	1.00	6.56	64.91	✓
57	application_train	APARTMENTS_MODE	Float	151 450	50.75	0.11	0.11	0.00	0.05	0.08	0.14	1.00	2.70	11.75	✓
58	application_train	BASEMENTAREA_MODE	Float	127 568	58.52	0.09	0.08	0.00	0.04	0.07	0.11	1.00	3.48	24.43	✓

No.	Dataset	Feature	Data Type	count	Missing%	average	σ	minimum	25%	50%	75%	maximum	Skewness	kurtosis	outlier
59	application_train	YEARS_BEGINEXPLUATATION_MODE	Float	157 504	48.78	0.98	0.06	0.00	0.98	0.98	0.99	1.00	-14.76	219.96	✓
60	application_train	YEARS_BUILD_MODE	Float	103 023	66.50	0.76	0.11	0.00	0.70	0.76	0.82	1.00	-1.00	4.76	✓
61	application_train	COMMONAREA_MODE	Float	92 646	69.87	0.04	0.07	0.00	0.01	0.02	0.05	1.00	5.62	48.86	✓
62	application_train	ELEVATORS_MODE	Float	143 620	53.30	0.07	0.13	0.00	0.00	0.00	0.12	1.00	2.55	8.60	✓
63	application_train	ENTRANCES_MODE	Float	152 683	50.35	0.15	0.10	0.00	0.07	0.14	0.21	1.00	2.39	11.42	✓
64	application_train	FLOORSMAX_MODE	Float	154 491	49.76	0.22	0.14	0.00	0.17	0.17	0.33	1.00	1.24	2.54	✓
65	application_train	FLOORSMIN_MODE	Float	98 869	67.85	0.23	0.16	0.00	0.08	0.21	0.38	1.00	0.96	1.35	✓
66	application_train	LANDAREA_MODE	Float	124 921	59.38	0.06	0.08	0.00	0.02	0.05	0.08	1.00	4.38	33.27	✓
67	application_train	LIVINGAPARTMENTS_MODE	Float	97 312	68.35	0.11	0.10	0.00	0.05	0.08	0.13	1.00	2.90	14.22	✓
68	application_train	LIVINGAREA_MODE	Float	153 161	50.19	0.11	0.11	0.00	0.04	0.07	0.13	1.00	2.90	12.46	✓
69	application_train	NONLIVINGAPARTMENTS_MODE	Float	93 997	69.43	0.01	0.05	0.00	0.00	0.00	0.00	1.00	16.25	309.73	✓
70	application_train	NONLIVINGAREA_MODE	Float	137 829	55.18	0.03	0.07	0.00	0.00	0.00	0.02	1.00	6.52	63.36	✓
71	application_train	APARTMENTS_MEDI	Float	151 450	50.75	0.12	0.11	0.00	0.06	0.09	0.15	1.00	2.64	11.24	✓
72	application_train	BASEMENTAREA_MEDI	Float	127 568	58.52	0.09	0.08	0.00	0.04	0.08	0.11	1.00	3.55	25.83	✓
73	application_train	YEARS_BEGINEXPLUATATION_MEDI	Float	157 504	48.78	0.98	0.06	0.00	0.98	0.98	0.99	1.00	-15.57	248.40	✓
74	application_train	YEARS_BUILD_MEDI	Float	103 023	66.50	0.76	0.11	0.00	0.69	0.76	0.83	1.00	-0.96	4.47	✓
75	application_train	COMMONAREA_MEDI	Float	92 646	69.87	0.04	0.08	0.00	0.01	0.02	0.05	1.00	5.42	45.26	✓
76	application_train	ELEVATORS_MEDI	Float	143 620	53.30	0.08	0.13	0.00	0.00	0.00	0.12	1.00	2.46	7.96	✓
77	application_train	ENTRANCES_MEDI	Float	152 683	50.35	0.15	0.10	0.00	0.07	0.14	0.21	1.00	2.39	11.47	✓
78	application_train	FLOORSMAX_MEDI	Float	154 491	49.76	0.23	0.15	0.00	0.17	0.17	0.33	1.00	1.24	2.47	✓
79	application_train	FLOORSMIN_MEDI	Float	98 869	67.85	0.23	0.16	0.00	0.08	0.21	0.38	1.00	0.96	1.35	✓
80	application_train	LANDAREA_MEDI	Float	124 921	59.38	0.07	0.08	0.00	0.02	0.05	0.09	1.00	4.37	33.24	✓
81	application_train	LIVINGAPARTMENTS_MEDI	Float	97 312	68.35	0.10	0.09	0.00	0.05	0.08	0.12	1.00	2.99	15.70	✓
82	application_train	LIVINGAREA_MEDI	Float	153 161	50.19	0.11	0.11	0.00	0.05	0.07	0.13	1.00	2.85	12.14	✓
83	application_train	NONLIVINGAPARTMENTS_MEDI	Float	93 997	69.43	0.01	0.05	0.00	0.00	0.00	0.00	1.00	15.67	289.49	✓
84	application_train	NONLIVINGAREA_MEDI	Float	137 829	55.18	0.03	0.07	0.00	0.00	0.00	0.03	1.00	6.51	63.65	✓
85	application_train	FONDKAPREMONT_MODE	String	97 216	68.39	-	-	-	-	-	-	-	-	-	-
86	application_train	HOUSETYPE_MODE	String	153 214	50.18	-	-	-	-	-	-	-	-	-	-
87	application_train	TOTALAREA_MODE	Float	159 080	48.27	0.10	0.11	0.00	0.04	0.07	0.13	1.00	2.80	12.17	✓
88	application_train	WALLSMATERIAL_MODE	String	151 170	50.84	-	-	-	-	-	-	-	-	-	-
89	application_train	EMERGENCYSTATE_MODE	String	161 756	47.40	-	-	-	-	-	-	-	-	-	-
90	application_train	OBS_30_CNT_SOCIAL_CIRCLE	Float	306 490	0.33	1.42	2.40	0.00	0.00	0.00	2.00	348.00	12.14	1424.82	✓
91	application_train	DEF_30_CNT_SOCIAL_CIRCLE	Float	306 490	0.33	0.14	0.45	0.00	0.00	0.00	0.00	34.00	5.18	126.31	✓
92	application_train	OBS_60_CNT_SOCIAL_CIRCLE	Float	306 490	0.33	1.41	2.38	0.00	0.00	0.00	2.00	344.00	12.07	1409.70	✓
93	application_train	DEF_60_CNT_SOCIAL_CIRCLE	Float	306 490	0.33	0.10	0.36	0.00	0.00	0.00	0.00	24.00	5.28	86.56	✓
94	application_train	DAYS_LAST_PHONE_CHANGE	Float	307 510	0.00	-962.86	826.81	-4292.00	-1570.00	-757.00	-274.00	0.00	-0.71	-0.31	✓
95	application_train	FLAG_DOCUMENT_2	Integer	307 511	0.00	-	-	-	-	-	-	-	153.79	23650.08	-
96	application_train	FLAG_DOCUMENT_3	Integer	307 511	0.00	-	-	-	-	-	-	-	-0.93	-1.14	-
97	application_train	FLAG_DOCUMENT_4	Integer	307 511	0.00	-	-	-	-	-	-	-	110.89	12295.64	-
98	application_train	FLAG_DOCUMENT_5	Integer	307 511	0.00	-	-	-	-	-	-	-	7.95	61.18	-
99	application_train	FLAG_DOCUMENT_6	Integer	307 511	0.00	-	-	-	-	-	-	-	2.91	6.45	-
100	application_train	FLAG_DOCUMENT_7	Integer	307 511	0.00	-	-	-	-	-	-	-	72.17	5207.14	-
101	application_train	FLAG_DOCUMENT_8	Integer	307 511	0.00	-	-	-	-	-	-	-	3.06	7.38	-
102	application_train	FLAG_DOCUMENT_9	Integer	307 511	0.00	-	-	-	-	-	-	-	15.93	251.70	-

No.	Dataset	Feature	Data Type	count	Missing%	average	σ	minimum	25%	50%	75%	maximum	Skewness	kurtosis	outlier
103	application_train	FLAG_DOCUMENT_10	Integer	307 511	0.00	-	-	-	-	-	-	-	209.59	43925.86	-
104	application_train	FLAG_DOCUMENT_11	Integer	307 511	0.00	-	-	-	-	-	-	-	15.89	250.63	-
105	application_train	FLAG_DOCUMENT_12	Integer	307 511	0.00	-	-	-	-	-	-	-	392.11	153753.00	-
106	application_train	FLAG_DOCUMENT_13	Integer	307 511	0.00	-	-	-	-	-	-	-	16.75	278.69	-
107	application_train	FLAG_DOCUMENT_14	Integer	307 511	0.00	-	-	-	-	-	-	-	18.37	335.55	-
108	application_train	FLAG_DOCUMENT_15	Integer	307 511	0.00	-	-	-	-	-	-	-	28.70	821.66	-
109	application_train	FLAG_DOCUMENT_16	Integer	307 511	0.00	-	-	-	-	-	-	-	9.89	95.74	-
110	application_train	FLAG_DOCUMENT_17	Integer	307 511	0.00	-	-	-	-	-	-	-	61.21	3745.20	-
111	application_train	FLAG_DOCUMENT_18	Integer	307 511	0.00	-	-	-	-	-	-	-	10.96	118.01	-
112	application_train	FLAG_DOCUMENT_19	Integer	307 511	0.00	-	-	-	-	-	-	-	40.96	1675.42	-
113	application_train	FLAG_DOCUMENT_20	Integer	307 511	0.00	-	-	-	-	-	-	-	44.36	1966.26	-
114	application_train	FLAG_DOCUMENT_21	Integer	307 511	0.00	-	-	-	-	-	-	-	54.61	2980.59	-
115	application_train	AMT_REQ_CREDIT_BUREAU_HOUR	Float	265 992	13.50	0.01	0.08	0.00	0.00	0.00	0.00	4.00	14.53	254.24	✓
116	application_train	AMT_REQ_CREDIT_BUREAU_DAY	Float	265 992	13.50	0.01	0.11	0.00	0.00	0.00	0.00	9.00	27.04	1151.87	✓
117	application_train	AMT_REQ_CREDIT_BUREAU_WEEK	Float	265 992	13.50	0.03	0.20	0.00	0.00	0.00	0.00	8.00	9.29	166.75	✓
118	application_train	AMT_REQ_CREDIT_BUREAU_MON	Float	265 992	13.50	0.27	0.92	0.00	0.00	0.00	0.00	27.00	7.80	90.43	✓
119	application_train	AMT_REQ_CREDIT_BUREAU_QRT	Float	265 992	13.50	0.27	0.79	0.00	0.00	0.00	0.00	261.00	134.37	43707.46	✓
120	application_train	AMT_REQ_CREDIT_BUREAU_YEAR	Float	265 992	13.50	1.90	1.87	0.00	0.00	1.00	3.00	25.00	1.24	1.97	✓
121	Bureau	CREDIT_ACTIVE	String	1716428	0.00	-	-	-	-	-	-	-	-	-	-
122	Bureau	CREDIT_CURRENCY	String	1716428	0.00	-	-	-	-	-	-	-	-	-	-
123	Bureau	DAYS_CREDIT	Integer	1716428	0.00	-1142.11	795.16	-2922.00	-1666.00	-987.00	-474.00	0.00	-0.58	-0.74	-
124	Bureau	CREDIT_DAY_OVERDUE	Integer	1716428	0.00	0.82	36.54	0.00	0.00	0.00	0.00	2792.00	55.93	3374.48	✓
125	Bureau	DAYS_CREDIT_ENDDATE	Float	1610875	6.15	510.52	4994.22	-42060.00	-1138.00	-330.00	474.00	31199.00	5.13	28.18	✓
126	Bureau	DAYS_ENDDATE_FACT	Float	1082775	36.92	-1017.44	714.01	-42023.00	-1489.00	-897.00	-425.00	0.00	-0.77	9.41	✓
127	Bureau	AMT_CREDIT_MAX_OVERDUE	Float	591940	65.51	3825.42	206031.61	0.00	0.00	0.00	0.00	115987185.00	470.91	245696.92	✓
128	Bureau	CNT_CREDIT_PROLONG	Integer	1716428	0.00	0.01	0.10	0.00	0.00	0.00	0.00	9.00	20.32	615.44	✓
129	Bureau	AMT_CREDIT_SUM	Float	1716415	0.00	354994.59	1149811.34	0.00	51300.00	125518.50	315000.00	585000000.00	124.59	49315.97	✓
130	Bureau	AMT_CREDIT_SUM_DEBT	Float	1458759	15.01	137085.12	677401.13	-4705600.32	0.00	0.00	40153.50	170100000.00	36.41	5673.43	✓
131	Bureau	AMT_CREDIT_SUM_LIMIT	Float	1124648	34.48	6229.51	45032.03	-586406.11	0.00	0.00	0.00	4705600.32	18.03	796.10	✓
132	Bureau	AMT_CREDIT_SUM_OVERDUE	Float	1716428	0.00	37.91	5937.65	0.00	0.00	0.00	0.00	3756681.00	403.24	211836.85	✓
133	Bureau	CREDIT_TYPE	String	1716428	0.00	-	-	-	-	-	-	-	-	-	-
134	Bureau	DAYS_CREDIT_UPDATE	Integer	1716428	0.00	-593.75	720.75	-41947.00	-908.00	-395.00	-33.00	372.00	-11.33	596.37	✓
135	Bureau	AMT_ANNUITY	Float	489637	71.47	15712.76	325826.95	0.00	0.00	0.00	13500.00	118453423.50	212.54	58560.69	✓
136	bureau_balance	MONTHS_BALANCE	Integer	27299925	0.00	-30.74	23.86	-96.00	-46.00	-25.00	-11.00	0.00	-0.76	-0.32	-
137	bureau_balance	STATUS	String	27299925	0.00	-	-	-	-	-	-	-	-	-	-
138	credit_card_balance	MONTHS_BALANCE	Integer	3840312	0.00	-34.52	26.67	-96.00	-55.00	-28.00	-11.00	-1.00	-0.60	-0.86	-
139	credit_card_balance	AMT_BALANCE	Float	3840312	0.00	58300.16	106307.03	-420250.18	0.00	0.00	89046.69	1505902.19	2.92	11.78	✓
140	credit_card_balance	AMT_CREDIT_LIMIT_ACTUAL	Integer	3840312	0.00	153807.96	165145.70	0.00	45000.00	112500.00	180000.00	1350000.00	2.06	5.18	✓
141	credit_card_balance	AMT_DRAWINGS_ATM_CURRENT	Float	3090496	19.52	5961.32	28225.69	-6827.31	0.00	0.00	0.00	2115000.00	9.66	164.93	✓
142	credit_card_balance	AMT_DRAWINGS_CURRENT	Float	3840312	0.00	7433.39	33846.08	-6211.62	0.00	0.00	0.00	2287098.31	10.07	184.27	✓
143	credit_card_balance	AMT_DRAWINGS_OTHER_CURRENT	Float	3090496	19.52	288.17	8201.99	0.00	0.00	0.00	0.00	1529847.00	50.57	3628.01	✓
144	credit_card_balance	AMT_DRAWINGS_POS_CURRENT	Float	3090496	19.52	2968.80	20796.89	0.00	0.00	0.00	0.00	2239274.16	19.42	713.99	✓
145	credit_card_balance	AMT_INST_MIN_REGULARITY	Float	3535076	7.95	3540.20	5600.15	0.00	0.00	0.00	6633.91	202882.01	2.49	10.18	✓
146	credit_card_balance	AMT_PAYMENT_CURRENT	Float	3072324	20.00	10280.54	36078.08	0.00	152.37	2702.70	9000.00	4289207.45	12.99	315.76	✓

No.	Dataset	Feature	Data Type	count	Missing%	average	σ	minimum	25%	50%	75%	maximum	Skewness	kurtosis	outlier
147	credit_card_balance	AMT_PAYMENT_TOTAL_CURRENT	Float	3840312	0.00	7588.86	32005.99	0.00	0.00	0.00	6750.00	4278315.69	14.48	393.26	✓
148	credit_card_balance	AMT_RECEIVABLE_PRINCIPAL	Float	3840312	0.00	55965.88	102533.62	-423305.82	0.00	0.00	85359.24	1472316.79	2.94	11.96	✓
149	credit_card_balance	AMT_RECIVABLE	Float	3840312	0.00	58088.81	105965.37	-420250.18	0.00	0.00	88899.49	1493338.19	2.91	11.72	✓
150	credit_card_balance	AMT_TOTAL_RECEIVABLE	Float	3840312	0.00	58098.29	105971.80	-420250.18	0.00	0.00	88914.51	1493338.19	2.91	11.72	✓
151	credit_card_balance	CNT_DRAWINGS_ATM_CURRENT	Float	3090496	19.52	0.31	1.10	0.00	0.00	0.00	0.00	51.00	6.91	81.55	✓
152	credit_card_balance	CNT_DRAWINGS_CURRENT	Integer	3840312	0.00	0.70	3.19	0.00	0.00	0.00	0.00	165.00	10.64	177.93	✓
153	credit_card_balance	CNT_DRAWINGS_OTHER_CURRENT	Float	3090496	19.52	0.00	0.08	0.00	0.00	0.00	0.00	12.00	26.32	1253.26	✓
154	credit_card_balance	CNT_DRAWINGS_POS_CURRENT	Float	3090496	19.52	0.56	3.24	0.00	0.00	0.00	0.00	165.00	11.35	192.55	✓
155	credit_card_balance	CNT_INSTALMENT_MATURE_CUM	Float	3535076	7.95	20.83	20.05	0.00	4.00	15.00	32.00	120.00	1.08	0.64	✓
156	credit_card_balance	NAME_CONTRACT_STATUS	String	3840312	0.00	-	-	-	-	-	-	-	-	-	-
157	credit_card_balance	SK_DPD	Integer	3840312	0.00	9.28	97.52	0.00	0.00	0.00	0.00	3260.00	12.95	190.37	✓
158	credit_card_balance	SK_DPD_DEF	Integer	3840312	0.00	0.33	21.48	0.00	0.00	0.00	0.00	3260.00	89.83	9007.74	✓
159	installment_payments	NUM_INSTALMENT_VERSION	Float	13605401	0.00	0.86	1.04	0.00	0.00	1.00	1.00	178.00	9.59	259.61	✓
160	installment_payments	NUM_INSTALMENT_NUMBER	Integer	13605401	0.00	18.87	26.66	1.00	4.00	8.00	19.00	277.00	2.50	6.71	✓
161	installment_payments	DAYS_INSTALMENT	Float	13605401	0.00	-1042.27	800.95	-2922.00	-1654.00	-818.00	-361.00	-1.00	-0.63	-0.80	-
162	installment_payments	DAYS_ENTRY_PAYMENT	Float	13602496	0.02	-1051.11	800.59	-4921.00	-1662.00	-827.00	-370.00	-1.00	-0.63	-0.80	✓
163	installment_payments	AMT_INSTALMENT	Float	13605401	0.00	17050.91	50570.25	0.00	4226.09	8884.08	16710.21	3771487.85	16.24	388.84	✓
164	installment_payments	AMT_PAYMENT	Float	13602496	0.02	17238.22	54735.78	0.00	3398.26	8125.52	16108.42	3771487.85	14.95	324.60	✓
165	previous_application	NAME_CONTRACT_TYPE	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
166	previous_application	AMT_ANNUITY	Float	1297979	22.29	15955.12	14782.14	0.00	6321.78	11250.00	20658.42	418058.15	2.69	15.07	✓
167	previous_application	AMT_APPLICATION	Float	1670214	0.00	175233.86	292779.76	0.00	18720.00	71046.00	180360.00	6905160.00	3.39	15.76	✓
168	previous_application	AMT_CREDIT	Float	1670213	0.00	196114.02	318574.62	0.00	24160.50	80541.00	216418.50	6905160.00	3.25	14.24	✓
169	previous_application	AMT_DOWN_PAYMENT	Float	774370	53.64	6697.40	20921.50	-0.90	0.00	1638.00	7740.00	3060045.00	36.48	2901.84	✓
170	previous_application	AMT_GOODS_PRICE	Float	1284699	23.08	227847.28	315396.56	0.00	50841.00	112320.00	234000.00	6905160.00	3.07	12.87	✓
171	previous_application	WEEKDAY_APPR_PROCESS_START	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
172	previous_application	HOUR_APPR_PROCESS_START	Integer	1670214	0.00	12.48	3.33	0.00	10.00	12.00	15.00	23.00	-0.03	-0.28	✓
173	previous_application	FLAG_LAST_APPL_PER_CONTRACT	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
174	previous_application	NFLAG_LAST_APPL_IN_DAY	Integer	1670214	0.00	-	-	-	-	-	-	-	-16.74	278.09	-
175	previous_application	RATE_DOWN_PAYMENT	Float	774370	53.64	0.08	0.11	-0.00	0.00	0.05	0.11	1.00	2.11	6.20	✓
176	previous_application	RATE_INTEREST_PRIMARY	Float	5951	99.64	0.19	0.09	0.03	0.16	0.19	0.19	1.00	5.20	28.20	✓
177	previous_application	RATE_INTEREST_PRIVILEGED	Float	5951	99.64	0.77	0.10	0.37	0.72	0.84	0.85	1.00	-1.01	0.26	✓
178	previous_application	NAME_CASH_LOAN_PURPOSE	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
179	previous_application	NAME_CONTRACT_STATUS	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
180	previous_application	DAYS_DECISION	Integer	1670214	0.00	-880.68	779.10	-2922.00	-1300.00	-581.00	-280.00	-1.00	-1.05	-0.04	✓
181	previous_application	NAME_PAYMENT_TYPE	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
182	previous_application	CODE_REJECT_REASON	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
183	previous_application	NAME_TYPE_SUITE	String	849809	49.12	-	-	-	-	-	-	-	-	-	-
184	previous_application	NAME_CLIENT_TYPE	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
185	previous_application	NAME_GOODS_CATEGORY	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
186	previous_application	NAME_PORTFOLIO	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
187	previous_application	NAME_PRODUCT_TYPE	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
188	previous_application	CHANNEL_TYPE	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
189	previous_application	SELLERPLACE_AREA	Integer	1670214	0.00	313.95	7127.44	-1.00	-1.00	3.00	82.00	4000000.00	529.62	296880.64	✓
190	previous_application	NAME_SELLER_INDUSTRY	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-

No.	Dataset	Feature	Data Type	count	Missing%	average	σ	minimum	25%	50%	75%	maximum	Skewness	kurtosis	outlier
191	previous_application	CNT_PAYMENT	Float	1297984	22.29	16.05	14.57	0.00	6.00	12.00	24.00	84.00	1.53	1.87	✓
192	previous_application	NAME_YIELD_GROUP	String	1670214	0.00	-	-	-	-	-	-	-	-	-	-
193	previous_application	PRODUCT_COMBINATION	String	1669868	0.02	-	-	-	-	-	-	-	-	-	-
194	previous_application	DAYS_FIRST_DRAWING	Float	997149	40.30	342209.86	88916.12	-2922.00	365243.00	365243.00	365243.00	365243.00	-3.60	10.97	✓
195	previous_application	DAYS_FIRST_DUE	Float	997149	40.30	13826.27	72444.87	-2892.00	-1628.00	-831.00	-411.00	365243.00	4.64	19.57	✓
196	previous_application	DAYS_LAST_DUE_1ST_VERSION	Float	997149	40.30	33767.77	106857.03	-2801.00	-1242.00	-361.00	129.00	365243.00	2.78	5.73	✓
197	previous_application	DAYS_LAST_DUE	Float	997149	40.30	76582.40	149647.42	-2889.00	-1314.00	-537.00	-74.00	365243.00	1.41	-0.01	✓
198	previous_application	DAYS_TERMINATION	Float	997149	40.30	81992.34	153303.52	-2874.00	-1270.00	-499.00	-44.00	365243.00	1.31	-0.29	✓
199	previous_application	NFLAG_INSURED_ON_APPROVAL	Float	997149	40.30	0.33	0.47	0.00	0.00	0.00	1.00	1.00	0.71	-1.49	-
200	POS_CASH_balance	MONTHS_BALANCE	Integer	10001358	0.00	-35.01	26.07	-96.00	-54.00	-28.00	-13.00	-1.00	-0.67	-0.71	-
201	POS_CASH_balance	CNT_INSTALMENT	Float	9975287	0.26	17.09	12.00	1.00	10.00	12.00	24.00	92.00	1.60	2.45	✓
202	POS_CASH_balance	CNT_INSTALMENT_FUTURE	Float	9975271	0.26	10.48	11.11	0.00	3.00	7.00	14.00	85.00	1.85	3.71	✓
203	POS_CASH_balance	NAME_CONTRACT_STATUS	String	10001358	0.00	-	-	-	-	-	-	-	-	-	-
204	POS_CASH_balance	SK_DPD	Integer	10001358	0.00	11.61	132.71	0.00	0.00	0.00	0.00	4231.00	14.90	255.32	✓
205	POS_CASH_balance	SK_DPD_DEF	Integer	10001358	0.00	0.65	32.76	0.00	0.00	0.00	0.00	3595.00	66.34	4836.55	✓

APPENDIX D

Univariate analysis for case study 2

Table D.1: Univariate analysis of education of applicants

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
Lower sec.	417	3 399	3 816	1.24	10.93	8.15	-0.33	0.00
Sec. special	19 524	198 867	218 391	71.02	8.94	10.19	-0.11	0.01
Incom. higher	872	9 405	10 277	3.34	8.48	10.79	-0.05	0.00
Higher edu.	4009	70 854	74 863	24.34	5.36	17.67	0.44	0.04
Academic deg.	3	161	164	0.05	1.83	53.67	1.55	0.00
Total	**24 825**	**282 686**	**307 511**	**100.00**	**8.07**	**11.39**	**0.00**	**0.05**

Table D.2: Univariate analysis of sources of income of applicants

Attribute	Goods	Bads	Total	%Total	Bad Rate	G:B odds	WoE	IV
Unemployed	10	45	55	0.02	18.18	4.50	-0.93	0.00
Working	15 224	143 550	158 774	51.63	9.59	9.43	-0.19	0.02
Com. associate	5 360	66 257	71 617	23.29	7.48	12.36	0.08	0.00
State servant	1 249	20 454	21 703	7.06	5.75	16.38	0.36	0.01
Pensioner	2 982	52 380	55 362	18.00	5.39	17.57	0.43	0.03
Total	**24 825**	**282 686**	**307 511**	**100.00**	**8.07**	**11.39**	**0.00**	**0.06**

Table D.3: Univariate analysis of occupation of applicants

Attribute	Goods	Bads	Total	%total	Bad Ra te	G:B odds	WoE	IV
Occupation 1	359	1 734	2 093	0.99	17.15	4.83	-0.77	0.01
Occupation 2	2259	17 692	19 951	9.45	11.32	7.83	-0.28	0.01
Occupation 3	7181	60 672	67 853	32.14	10.58	8.45	-0.21	0.01
Occupation 4	3 539	33 216	36 755	17.41	9.63	9.39	-0.10	0.00
Occupation 5	59	692	751	0.36	7.86	11.73	0.12	0.00
Occupation 6	92	1 213	1305	0.62	7.05	13.18	0.24	0.00
Occupation 7	4 584	68 015	72 599	34.39	6.31	14.84	0.36	0.04
Occupation 8	474	9 339	9 813	4.65	4.83	19.70	0.64	0.01
Total	**24 825**	**282 686**	**307 511**	**100.00**	**8.07**	**11.39**	**0.00**	**0.09**

Table D.4: Univariate analysis of organisation of applicants

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
Organization 1	199	1 079	1 278	0.42	15.57	5.42	-0.74	0.00
Organization 2	997	7 535	8 532	2.77	11.69	7.56	-0.41	0.01
Organization 3	144	1 155	1 299	0.42	11.09	8.02	-0.35	0.00
Organization 4	5 329	46 827	52 156	16.96	10.22	8.79	-0.26	0.01
Organization 5	7 624	74 262	81 886	26.63	9.31	9.74	-0.16	0.01
Organization 6	1 838	19 989	21 827	7.10	8.42	10.88	-0.05	0.00
Organization 7	1 855	22 179	24 034	7.82	7.72	11.96	0.05	0.00
Organization 8	1 465	19 448	20 913	6.80	7.01	13.28	0.15	0.00
Organization 9	1 198	16 963	18 161	5.91	6.60	14.16	0.22	0.00
Organization 10	534	8 493	9 027	2.94	5.92	15.90	0.33	0.00
Organization 11	3 045	53 305	56 350	18.32	5.40	17.51	0.43	0.03
Organization 12	543	10 240	10 783	3.51	5.04	18.86	0.50	0.01
Organization 13	38	794	832	0.27	4.57	20.89	0.61	0.00
Organization 14	16	417	433	0.14	3.70	26.06	0.83	0.00
Total	**24 825**	**282 686**	**307 511**	**100.00**	**8.07**	**11.39**	**0.00**	**0.07**

Table D.5: Univariate analysis of age of applicants

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(20.0, 28.0]	3 558	27 194	30 752	10.00	11.57	7.64	-0.40	0.02
(28.0, 32.0]	3 382	27 378	30 760	10.00	10.99	8.10	-0.34	0.01
(32.0, 36.0]	3 015	27 730	30 745	10.00	9.81	9.20	-0.21	0.00
(36.0, 39.0]	2 723	28 036	30 759	10.00	8.85	10.30	-0.10	0.00
(39.0, 43.0]	2 430	28 315	30 745	10.00	7.90	11.65	0.02	0.00
(43.0, 47.0]	2 398	28 366	30 764	10.00	7.79	11.83	0.04	0.00
(47.0, 52.0]	2 193	28 540	30 733	9.99	7.14	13.01	0.13	0.00
(52.0, 56.0]	1 951	28 807	30 758	10.00	6.34	14.77	0.26	0.01
(56.0, 61.0]	1 668	29 089	30 757	10.00	5.42	17.44	0.43	0.02
(61.0, 69.0]	1 507	29 231	30 738	10.00	4.90	19.40	0.53	0.02
Total	**24 825**	**282 686**	**307 511**	**100.00**	**8.07**	**11.39**	**0.00**	**0.08**

Table D.6: Univariate analysis of external source 1

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(0.005, 0.21]	2 356	11 058	13 414	4.36	17.56	4.69	-0.89	0.05
(0.21, 0.3]	1 555	11 858	13 413	4.36	11.59	7.63	-0.40	0.01
(0.3, 0.37]	1 220	12 194	13 414	4.36	9.09	10.00	-0.13	0.00
Missing Values	14 771	15 8607	173 378	56.38	8.52	10.74	-0.06	0.00
(0.37, 0.44]	1 124	12 288	13 412	4.36	8.38	10.93	-0.04	0.00
(0.44, 0.51]	898	12 517	13 415	4.36	6.69	13.94	0.20	0.00
(0.51, 0.57]	808	12 604	13 412	4.36	6.02	15.60	0.31	0.00
(0.57, 0.64]	689	12 724	13 413	4.36	5.14	18.47	0.48	0.01
(0.64, 0.71]	588	12 825	13 413	4.36	4.38	21.81	0.65	0.01
(0.71, 0.79]	471	12 942	13 413	4.36	3.51	27.48	0.88	0.02
(0.79, 0.96]	345	13 069	13 414	4.36	2.57	37.88	1.20	0.04
Total	**24 825**	**282 686**	**307 511**	**100.00**	**8.07**	**11.39**	**0.00**	**0.15**

Table D.7: Univariate analysis of external source 2

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-0.01, 0.22]	5 631	25 055	30 686	9.98	18.35	4.45	-0.94	0.13
(0.22, 0.34]	3 706	26 979	30 685	9.98	12.08	7.28	-0.45	0.02
(0.34, 0.44]	3 056	27 631	30 687	9.98	9.96	9.04	-0.23	0.01
(0.44, 0.51]	2 566	28 118	30 684	9.98	8.36	10.96	-0.04	0.00
Missing Values	52	608	660	0.21	7.88	11.69	0.03	0.00
(0.51, 0.57]	2 278	28 406	30 684	9.98	7.42	12.47	0.09	0.00
(0.57, 0.61]	2 042	28 645	30 687	9.98	6.65	14.03	0.21	0.00
(0.61, 0.65]	1 794	28 889	30 683	9.98	5.85	16.10	0.35	0.01
(0.65, 0.68]	1 499	29 195	30 694	9.98	4.88	19.48	0.54	0.02
(0.68, 0.72]	1 289	29 387	30 676	9.98	4.20	22.80	0.69	0.04
(0.72, 0.85]	912	29 773	30 685	9.98	2.97	32.65	1.05	0.07
Total	**24825**	**282686**	**307511**	**100.00**	**8.07**	**11.39**	**0.00**	**0.31**

Table D.8: Univariate analysis of external source 3

Attribute	Goods	Bads	Total	%total	Bad Rate	G:B odds	WoE	IV
(-0.009, 0.23]	4 941	19 760	24 701	8.03	20.00	4.00	-1.05	0.14
(0.23, 0.33]	3 156	21 588	24 744	8.05	12.75	6.84	-0.51	0.03
(0.33, 0.41]	2 383	22 674	25 057	8.15	9.51	9.51	-0.18	0.00
Missing Values	5 677	55 288	60 965	19.83	9.31	9.74	-0.16	0.01
(0.41, 0.48]	1 970	22 719	24 689	8.03	7.98	11.53	0.01	0.00
(0.48, 0.54]	1 494	22 692	24 186	7.87	6.18	15.19	0.29	0.01
(0.54, 0.59]	1 357	24 035	25 392	8.26	5.34	17.71	0.44	0.01
(0.59, 0.64]	1 173	23 552	24 725	8.04	4.74	20.08	0.57	0.02
(0.64, 0.69]	1 043	23 702	24 745	8.05	4.21	22.72	0.69	0.03
(0.69, 0.75]	836	22 839	23 675	7.70	3.53	27.32	0.88	0.04
(0.75, 0.9]	795	23 837	24 632	8.01	3.23	29.98	0.97	0.05
Total	**24 825**	**282 686**	**307 511**	**100.00**	**8.07**	**11.39**	**0.00**	**0.33**