UNIVERSITY OF CAPE TOWN

An Exploration of Alternative Features in Micro-Finance Loan Default Prediction Models

Author:
Devon STONE

Supervisor:
Mr S BRITZ

A thesis submitted in the partial fulfilment for a Masters of Science in Data Science from the

DEPARTMENT OF STATISTICAL SCIENCES

June 4, 2020

Declaration of Authorship

I, Devon STONE, declare that this thesis titled, "An Exploration of Alternative Features in Micro-Finance Loan Default Prediction Models" and the work presented in it are my own. I confirm that I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signed: Signed by candidate

Date: 04/06/2020

Abstract

Despite recent developments financial inclusion remains a large issue for the World's unbanked population. Financial institutions - both larger corporations and micro-finance companies - have begun to provide solutions for financial inclusion. The solutions are delivered using a combination of machine learning and alternative data.

This minor dissertation focuses on investigating whether alternative features generated from Short Messaging Service (SMS) data and Android application data contained on borrowers' devices can be used to improve the performance of loan default prediction models. The improvement gained by using alternative features is measured by comparing loan default prediction models trained using only traditional credit scoring data to models developed using a combination of traditional and alternative features. Furthermore, the paper investigates which of 4 machine learning techniques is best suited for loan default prediction. The 4 techniques investigated are logistic regression, random forests, extreme gradient boosting, and neural networks. Finally the paper identifies whether or not accurate loan default prediction models can be trained using only the alternative features developed throughout this minor dissertation.

The results of the research show that alternative features improve the performance of loan default prediction across 5 performance indicators, namely overall prediction accuracy, repaid prediction accuracy, default prediction accuracy, F1 score, and AUC. Furthermore, extreme gradient boosting is identified as the most appropriate technique for loan default prediction. Finally, the research identifies that models trained using the alternative features developed throughout this project can accurately predict loan that have been repaid, the models do not accurately predict loans that have not been repaid.

Acknowledgements

My deepest thanks are extended to my supervisor, Mr Stefan Britz, for his guidance and insights throughout this project. I would like to thank my family for their constant support. Finally, to my friend James Leslie, thank you for your input and support throughout the project.

Contents

List of Figures

List of Tables

List of Abbreviations

SMS	**S**hort **M**essaging **S**ervice
LDA	**L**inear **D**iscriminant **A**nalysis
CART	**C**lassification and **R**egression **T**rees
NN	**N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine
XGB	**XGB**oost
ROC	**R**eceiver **O**perating **C**haracteristic
AUC	**A**rea **U**nder **C**urve
SMOTE	**S**ynthetic **M**inority **O**ver-**S**ampling **T**echnique
MFIs	**M**icro **F**inance **I**nstitutions
SVD	**S**ingular **V**alue **D**ecomposition
KNN	**K**- **N**earest **N**eighbours
REF	**R**ecursive **F**eature **E**limination
XGBoost	**X**treme **G**radient **B**oosting
MLP	**M**ulti **L**ayer **P**erceptron
MOE	**M**ixture **O**f **E**xperts
RAF	**F**uzzy **A**daptive **R**esonance
LVQ	**L**earning **V**ector **Q**uantization
ReLU	**R**ectified **L**inear **U**nit
SD	**S**ociodemographic **D**ata
CB	**C**redit **B**ureau
ALD	**A**lternative **D**ata

Chapter 1

Introduction

1.1 Problem Description

More than 1.7 billion adults around the world do not have access to basic financial services; even more do not have access to a source of safe credit. Mobile banking platforms have been a driving force in worldwide financial inclusion. Since 2011, more than 1 billion adults have gained access to a bank account for the first time (The World Bank, 2018). Despite the rapid developments in financial inclusion, providing credit to the recently banked [1] population remains an issue.

The recently banked population does not have a financial history and are required to develop their financial history with an institution before they can be deemed creditworthy. This can often be a time-consuming process and can cause financial strain. This problem most occurs within the recently banked population in developing countries. However, the issue does relate to young adults entering the financial market within developed countries.

Micro-Finance companies and larger corporate institutions are starting to provide solutions to this issue. The solution is being derived through the use of alternative data in conjunction with machine learning algorithms. Alternative data - which includes data sourced from an individuals personal cellular device such as call and sms data, contact information, social media and other application data etc - is being used to develop the models that drive credit scoring systems, which grant or deny credit to consumers (Óskarsdóttir et al., 2019).

1.2 Background

Credit scoring is the set of modelling and decision techniques associated with autonomously adjudicating whether or not a potential borrower should be granted credit (Zhao et al., 2015). At the heart of these systems are loan default prediction models. The techniques involved are used to drive the strategies for determining the amount of credit a borrower should receive, the period of repayment, and the interest rate due on the amount borrowed (Christl and Pribil, 2005).

Credit scoring systems range in scale from the rating of countries and global international companies to rating personal credits. The systems measure a potential borrower's ability to repay a financial obligation. The systems do not forecast loan profitability. Rather, they are used to reduce credit risk and limit the number of loans that are not repaid, which in turn increases profitability (Jensen, 1992).

[1] People that have recently gained access to a bank account.

Traditional credit scoring involves considering a borrower's previous loan history when determining their credit score. Loan history data can comprise of loans that were taken from the institution granting the credit, or can be acquired from external credit bureaus. A consumer's previous loan performance directly influences the credit score they receive. If a consumer did not fully repay a previous loan, credit scoring systems will take this into consideration and assign the consumer a lower score (Thomas et al., 2001).

Since the beginning of the big data era, financial institutions have been able to access rapidly increasing volumes of data. Beyond the volume of data, financial intuitions have been able to access various types of data, from varying sources. Companies have been able to extract data and create features from; clients' short message service history, their call history, and data from clients' social media platforms. Data acquired from these sources is referred to as alternative data.

Credit scoring is one of the oldest applications of data analytics (Jones and Hensher, 2008). Prior to the rise of machine learning, more traditional modelling techniques such as logistic regression, linear discriminant analysis and naive Bayes classifiers were used to drive credit scoring systems.

Since the rise of big data and machine learning, financial institutions have been able to train and use non-parametric models such as decision trees, support vector machines and neural networks to drive their credit decision systems (Jones and Hensher, 2008).

The use of modern machine learning algorithms within credit scoring systems has not been fully supported within the financial industry. Machine learning algorithms are often complex and the predictions they produce can be difficult to explain.

Beyond their complexity, modern machine learning models dynamically evolve and are required to be regularly retrained on different data flows. Tracing the evolution of machine learning models and the data flows they are retrained on poses major issues for financial regulators (Guégana and Hassan, 2018).

1.3 Aims of Research

Despite recent advancements in the use of alternative data in credit granting models, the issue is still widely felt (Óskarsdóttir et al., 2019). Research into a solution is limited and has been mainly focused on using contact data on potential borrowers' cellular device and network analysis techniques.

This minor dissertation (m.d.) - using data provided by a Nigerian micro-finance company and the Nigerian credit bureaus - seeks to address the following aims:

- Assess if augmenting sociodemographic and credit bureau data with the alternative features used in this project improves the overall performance of loan default prediction models.
- Determine if the alternative features used through this dissertation can be used to train accurate loan default prediction models.

- Identify the optimal technique for developing loan default prediction models out of logistic regression, random forests, extreme gradient boosting, and a multi-perceptron neural networks.

1.4 Scope of Project

Only first time loan applicants that had existing Nigerian credit bureau data were considered for this project. The applicants were required to have a credit history in order to measure the impact of augmenting sociodemographic and credit bureau data with alternative data. The models developed only using alternative feature aim to provide an indication as to how well the models would perform on the unbanked population. However, the impacts on financial inclusion are not measured.

The regulatory and financial implications of each data category and modelling technique used throughout this project are not investigated. The project only investigates the modelling performance of each technique across the various datasets containing all combinations of the various data categories.

The alternative features used throughout this project are generated generated from Short Messaging Service (SMS) data (only messages received from banking intuitions) and Android application data on borrowers' devices. Contact data or other data types on borrowers' devices are not considered.

1.5 Layout of the Paper

Chapter 2 presents a review of literature related to assigning first-time loan customers with a credit score and how a credit score relates to loan default prediction. Particular attention is given to the modelling techniques that have been used to drive credit scoring, and how those techniques have developed over time. Finally the chapter explores the various alternative data sources used in credit scoring and the techniques applied to make use of the sources.

First, Chapter 3 details the data sources used throughout this project. Secondly, it summarises the data wrangling and feature engineering processes completed to generate the alternative features created during this m.d. Chapter 3 then details the preprocessing techniques used to created handle missing values, scale the variables, and handle outliers contained within the data used to train and test the models developed throughout this project.

Chapter 4 summarises the various datasets - and the data categories contained in each dataset - that are used to train the loan default prediction models of this project. The chapter further details the feature selection process for each dataset. Then, the various techniques used to train the loan default prediction models are detailed, which includes the hyperparameters that are tuned for each technique. Finally, the measures used to test whether the alternative features improved model performance and the test used to compare modelling techniques are detailed.

The results chapter of this project - Chapter 5 - displays the results of the feature selection process completed for each dataset. The chapter then displays the performance of each modelling technique across each dataset. Finally, the chapter summarises the findings of the research.

The concluding chapter of this project - Chapter 6 - presents the findings and conclusions of the project and indicates how the research conducted could be furthered.

Chapter 2

Literature Review

This chapter provides an insight into the previous academic work completed in the field of credit scoring and loan default prediction models. The chapter further describes the statistical and machine learning techniques used to train and develop loan default prediction models. Finally, the chapter describes the work related to using alternative features in credit scoring models.

2.1 Loan Default Prediction

The first recorded case of consumer lending dates back to the Babylonian empire, over 4000 years ago (Lewis, 1992). Since then the creditworthiness of each credit applicant has been assessed. The birth of computers allowed for this process to be automated and allowed for credit models to be developed (Lewis, 1992).

Credit scoring is defined as the scientific approach for assessing the creditworthiness of a potential borrower. This approach was first automated by D Durand in 1941. Durand developed a discriminant analysis model that classified potential borrowers into two distinct categories: those unlikely to repay and those likely to repay. This form of classification model, now referred to as loan default prediction model, forms the heart of a credit scoring system. Durand's model not only sped up the process of loan default prediction but removed the need for subjective rules in the creditworthiness assessment process (Thomas, 2000).

Durand's automated approach for assessing creditworthiness was initially met with scepticism by the majority of the banking sector. It took until the late 1960s for credit scoring to be deemed the accepted method for assessing consumer creditworthiness. This was driven by two major developments, namely the introduction of credit cards and advances in the processing power of the era's computers. The introduction of credit cards led to a rapid rise in the number of consumers seeking credit, which meant that manual creditworthiness checks were no longer a valid option. The financial industry turned to automated models (Marquez, 2008).

Credit scoring models were rapidly developed throughout the 1970s and 1980s. Due to the secretive nature of the credit scoring systems within lending companies, very little was released about the specific content used in the models and how the models were developed. However, a number of models developed by academics were designed to represent the models and systems used within the industry during this era. Each type of model had its own statistical strengths and weaknesses (Thomas et al., 2004).

By the 1990s credit scoring models were regularly used to assess potential personal loans, business loans, and small loans. The 1990s also saw the introduction of credit score cards. In more recent times machine learning and artificial intelligence have been used in credit scoring. These models are generally more sophisticated and less interpret-able than than their predecessors. As a result they have not yet been fully accepted by regulatory bodies (Li and Zhong, 2012).

The next three sections of this chapter will detail works related to the various modelling techniques used in credit scoring. For the sake of flow the techniques have been broken into three categories. These categories are statistical learning, machine learning techniques, and ensemble models. This is not an absolute distinction as there is overlap between these fields. Sections 2.3, 2.3, and 2.4 will detail the statistical learning techniques, machine learning techniques, and ensemble techniques widely used in loan default prediction modelling.

2.2 Statistical Credit Models

A wide variety of statistical techniques have been used to develop effective predictive credit scoring models. These techniques include weight of evidence measure, regression analysis, discriminant analysis, probit analysis, logistic regression, linear programming and decision trees. These techniques produce results than can be easily understood by regulators and communicated to other members of a business (Pointon, 2011).

2.2.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a simple parametric statistical technique that is used to distinguish between two classes. In terms of credit scoring, LDA is used to classify potential borrowers into one of two classes, a class containing borrowers that are likely to repay or a class containing borrowers that are unlikely to repay. LDA is still one of the most widely used techniques in credit scoring and loan default prediction. It was first used as a credit scoring approach by Durand (1941).

Durand developed an LDA model with the following linear discriminate function.

$$LDF = a_0 + a_1x_1 + a_2x_2 + ... + a_nx_n \tag{2.1}$$

Where x_1 ... x_n represent the variables used to classify potential borrowers and a_1 ... a_n indicate the discrimination coefficients for the variables. Equation 2.1 returns a a single numeric value. A potential borrower is classified as likely to pay or not based upon the value being above our below a predefined cut-off value.

The variables used in Durand's model were an applicant's age, their sex, their residential status, their occupation, the field of industry the applicant worked in, the number of years the applicant had worked at their current employer, and Boolean variables indicating whether or not the applicant had a bank account, real estate and life insurance. Each variable was bucketed into different categories and each category was assigned a value. The minimum and maximum values of Durand's formula where 0 and 3.46 respectively. Applicants that had a score less than 1.25 were denied credit (Durand, 1941).

LDA models have been scrutinised as they assume linear relationships between dependent variables and independent variables. Furthermore, LDA models require the assumption to be made that all predictor variables; must follow a normal distribution, are homoscedastic, and are multicollinear (Li and Zhong, 2012).

2.2.2 Logistic Regression

Logistic regression was developed by David Cox (1958). Like LDA, logistic regression is an adaptation of linear regression. However, logistic regression does not require the same assumptions to be made about the independent variables used.

The technique is used to describe the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Logistic regression models always produce dichotomous results (values are either 0 or 1) and have been widely used to solve binary classification problems.

Wiginton (1980) published one of the first works relating to using a binary classification logistic regression model in credit scoring. The model he developed was based on the following cumulative logistic probability function.

$$\ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n \tag{2.2}$$

where p_i is the probability of customer defaulting on a potential credit, β_i are the coefficients of the input variables and xi are the input variables.

Wiginton developed an optimal cut-off probability that was used to assign potential borrowers to either a "bad creditors" class or a "good creditors" class. Those assigned to the "bad" class were not granted credit.

A visual representation of Wiginton's model can be seen in figure 2.1. The summation symbol represents equation 2.2, while the sigmoid symbol represents the sigmoid activation function that maps the value produced from 2.2 between 0 and 1.

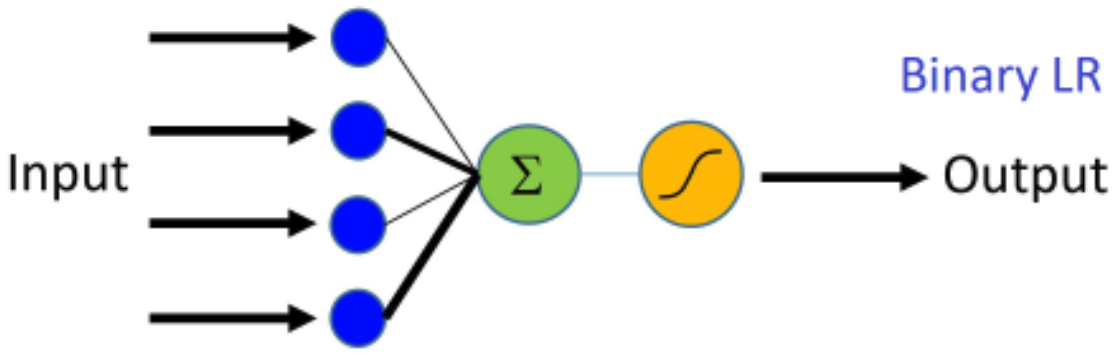

FIGURE 2.1: Wiginton's Logistic Regression Model
(Pant, 2018)

Wiginton deemed that a logistic regression model gave superior classification results when compared to a LDA model. His model was able to achieve an out-of-sample classification accuracy of over 58%. However, Hand and Henley (1997) compared using logistic regression approach to simple linear regression and found that both approaches had very similar classification accuracies.

2.2.3 Classification Tress

Classification and Regression Trees (CART) are another statistical technique that have been commonly used for credit scoring. Like LDA and logistic regression, classification trees have been used to classify potential borrowers into either a "likely to repay a financial obligation" or "unlikely to repay a financial obligation" class. They are non-parametric models used to predict a dependent variable as a function of continuous, discrete, or categorical independent variables. Decision trees are dichotomous models that are developed by splitting the records at each node based on a function of a single input. They consider all possible splits and identify the best sub-tree based on its overall error rate (Zekic-Susac et al., 2004).

The CART algorithm was developed by Thomas et al. (1984). They found that CART models are invariant under transformations in the predictor space and that Multi-factor response is easily dealt with. Furthermore, they found that modelling results could be easily to explained to non-statisticians due to their CART's inherent visual properties. Figure 2.2 displays the algorithm developed by Thomas et al. (1984).

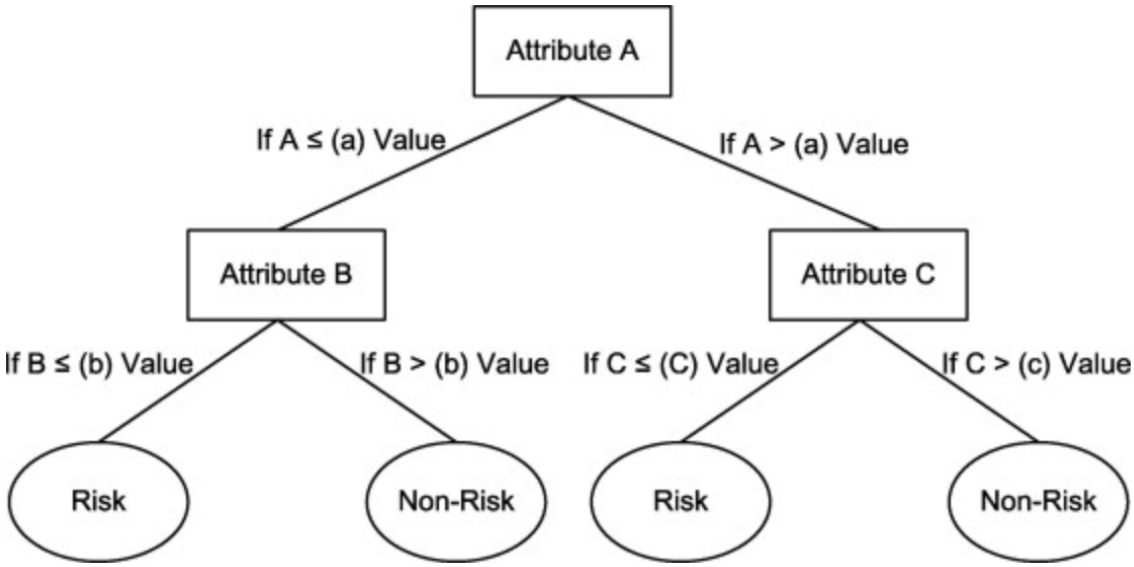

FIGURE 2.2: CART Model
(Thomas et al., 1984)

In figure 2.2 subsets created by each split in the tree are referred to as nodes. The subsets which result in the end of branch, no further splits are made, are termed terminal nodes. Terminal nodes get assigned to one of the predefined classes. In figure 2.2 there are 3 classes. A predicted classes for an input vector is found by passing through each binary node in the tree until a terminal node is reached.

Zekic-Susac et al. (2004) compared the performance of a CART, a neural network (NN) and logistic regression model in scoring a sample of small business loans provided by a Croatian bank. They used pruning to avoid over-fitting when training tree based models. Pruning involves growing a tree and then removing branches and terminal nodes that do not contain a predefined number of data points. They used Gini index as the evaluation function used for splitting.

Gini index is a measure of the likelihood that a randomly sampled data point passed through a model would be incorrectly labelled. It is calculated as shown in equation 2.3.

$$G(p) = 1 - \sum_{i=1}^{J} p_i^2 \tag{2.3}$$

Where J is the number of potential classes and p_i is the set of data points belonging to the class i.

The results of Zekic-Susac et al. (2004) research can be seen in table 2.1.

Model	Total Accuracy (%)	Default Accuracy (%)	Repaid Accuracy (%)
Probabilistic NN	83.30	80.00	85.19
Logistic regression	57.14	66.67	51.85
CART	66.67	66.67	66.67

TABLE 2.1: Results of Research Conducted by Zekic-Susac et al. (2004)

The models displayed in Table 2.1 were trained on the same training data and the same 20 independent variables were used to train each model. The results shown are test sample accuracies. It can be seen from Table 2.1 that the CART model produced by Zekic-Susac et al. (2004) outperformed their logistic regression model in terms of total accuracy and in terms of predicting loans that were repaid (Repaid Accuracy). However, the CART model was outperformed in predicting both loans that were repaid and loans there were not repaid by their probabilistic NN model.

2.3 Machine Learning Models

After the rapid expansion of consumer credit numerous statistical methods were successfully used for credit risk assessment. However, these models often had difficulty in modelling complex financial scenarios due to their use of fixed functions and statistical assumptions (Luo et al., 2017). Studies have shown that machine learning techniques such as Support Vector Machines (SVM's), Random Forests , and Neural Networks are superior to that of statistical techniques in terms of predicting whether consumers are likely to repay a loan when the training sample is large (Bellotti and Crook, 2009).

2.3.1 Support Vector Machines

SVM models define a hyper-plane that best separates two data classes so that the margin width between the hyper-plane and the data points is maximised. The hyper-plane can be linear or non-linear. The wider the margin width, the less complex the model is and the more likely it is to generalise well.

The hyper-plane can often be difficult to define if the classes are not well separated. Figure 2.3 displays such a case.

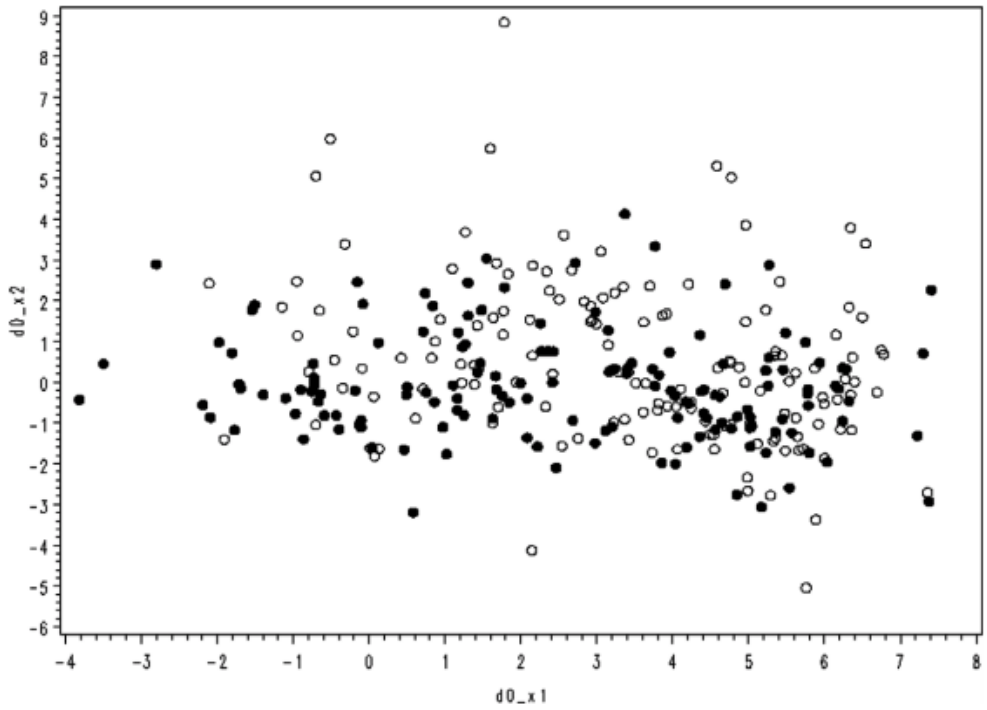

FIGURE 2.3: Poorly Separated Credit Data
(Bellotti and Crook, 2009)

Bellotti and Crook (2009) developed a SVM model that predicted whether credit card users would default on their repayments. A consumer was deemed to have defaulted if they fell more than 3 months behind on their repayments within the first 12 months of their account opening. They used a training sample of 25,000 consumers. The model developed used a non-linear kernel and its parameters were tuned using a grid-search in order to maximise the model's area (AUC) under the ROC curve, which is a single summary statistic used to measure a binary classification model's specificity (true negative rate) and sensitivity (true positive rate). These measures will be expanded upon in chapter 5.

On top using the SVM model to classify loans, Bellotti and Crook (2009) used the magnitude of the weight of each feature as a feature selection criterion. Only included features with a weight of more than 0.1 in their final model. They compared their SVM model to a logistic regression and k-nearest neighbours (KNN) model. Each model's AUC, sensitivity and specificity was compared.

Figure 2.4 compares the performance of SVM model to the performance of the logistic regression model, while figure 2.5 compares the performance of SVM model to the performance of the KNN model. In both figures the solid line is the SVM model's performance and the dashed line is the other model.

We can see from figures 2.4 and 2.5 that the SVM model outperformed the other model. Furthermore, the SVM model had a better training and test AUC, specificity , and sensitivity than the other two models.

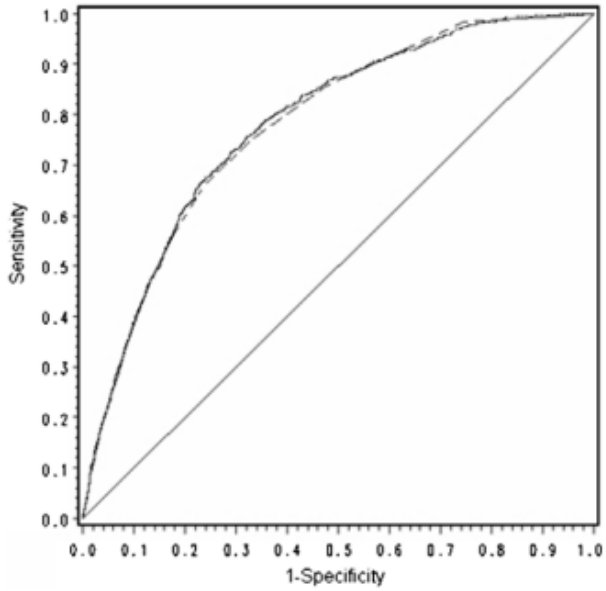

FIGURE 2.4: ROC Curve Comparison between SVM Model and Logistic Regression Model

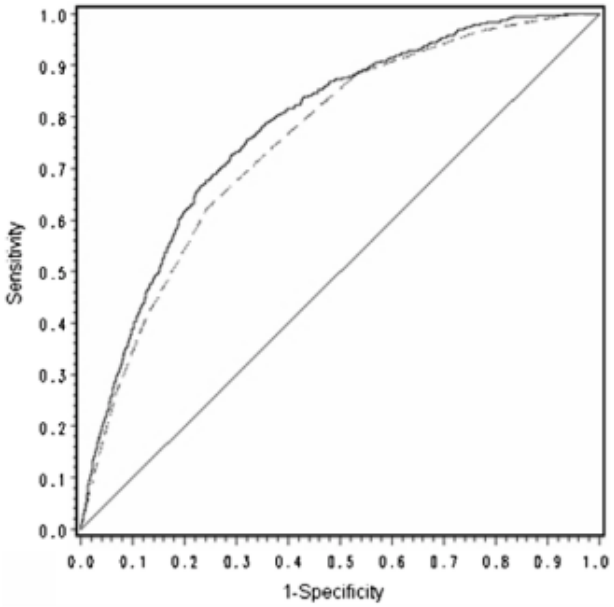

FIGURE 2.5: ROC Curve Comparison between SVM Model and KNN Model

2.3.2 Neural Networks

The research conducted by Zekic-Susac et al. (2004) displayed that neural networks improved on classification accuracy of more traditional statistical techniques such as decision trees and logistic regression when classifying whether or not consumers would repay a financial obligation. The research conducted by West (2000) compared the performance of five different neural network algorithms when applied to loan default prediction. The models developed varied in architecture, loss functions, and learning rates.

West developed models with the following features: a multi-layer perceptron (MLP) model, a model that used the mixture-of-experts (MOE) approach, a model that used a radial basis function (RBF), learning vector quantization (LQV) model, and fuzzy adaptive resonance (RAF) model. The architectures of each model were made similar. The input and output layers of each model were identical. However, the hidden layers of each model varied. West determined the optimal number of nodes in the hidden layers of the MLP and MOE models using a cascade learning approach. The hidden layers of the LQV model was determined by setting the number of neurons in each layer equal to 10% of the size of the training data. The hidden layers of the RBF and RAF models were determined experimentally.

West used a sample of 1000 loans, provided by a German credit provider, to train and test the models he developed. The sample consisted of 700 loans that were repaid and 300 that were not. He used the same features to train each model and used 10-fold cross validation to test the accuracy of each model.

Table 2.2 displays the results of West (2000)'s research. Like Table 2.1, Table 2.2 displays each model's accuracy in terms of predicting loans that were repaid, loans that were not repaid ,and overall prediction accuracy. The accuracies shown are the average of the best three results from the 10-fold cross validation conducted for each model.

Model	Total Accuracy (%)	Default Accuracy (%)	Repaid Accuracy (%)
MLP	87.09	46.92	75.04
MOE	86.99	55.43	77.57
RBF	85.76	51.79	75.63
LQV	79.15	55.20	72.20
FAR	70.92	58.14	62.29

TABLE 2.2: Results of Research Conducted by West (2000)

Table 2.2 displays that the MOE model was the best overall performing model. West (2000) used a chi-square test to assess whether or not there were significant differences between the models he developed in terms of predicting loan default. The chi-square test indicated that MOE, RBF, and MLP are superior models for predicting loan repayment when compared to RAF and LQV models.

Table 2.2 does not show that West developed logistic regression and CART models as reference models. These models outperformed all neural network models, but were deemed to not perform significantly better than the MOE, RBF, and MLP models.

The reason for the strong performance of the statistical approaches was deemed to be due to the smaller size of training set used to develop the models. Machine learning algorithms are data hungry, meaning they require large training datasets in order to detect patterns in the training data and produce accurate modelling results (Obermeyer and Emanuel, 2016).

Shen et al. (2019) used the same dataset as West (2000) to train and test a neural network. Their model used a particle swarm optimisation (PSO) algorithm to search for the optimal weights and deviations.

Furthermore, they used synthetic minority over-sampling technique (SMOTE) to balance the training dataset before training the model. This was done as the majority of credit history datasets are imbalanced towards the class containing customers that repaid their financial obligations. SMOTE involves generating synthetic examples of data points that belong to the minority class in a training dataset. Shen et al. (2019) generated synthetic examples by identifying the k nearest neighbours of a randomly selected minority data point. A variable difference vector between the minority instance under consideration and its corresponding nearest neighbours was then calculated and multiplied by a random value between 0 and 1. The feature vector was then added to the original minority data point. This process was completed until the ratio of credit defaulters and credit re-payers matched. Figure 2.6 visually displays the steps carried out in the class balancing process.

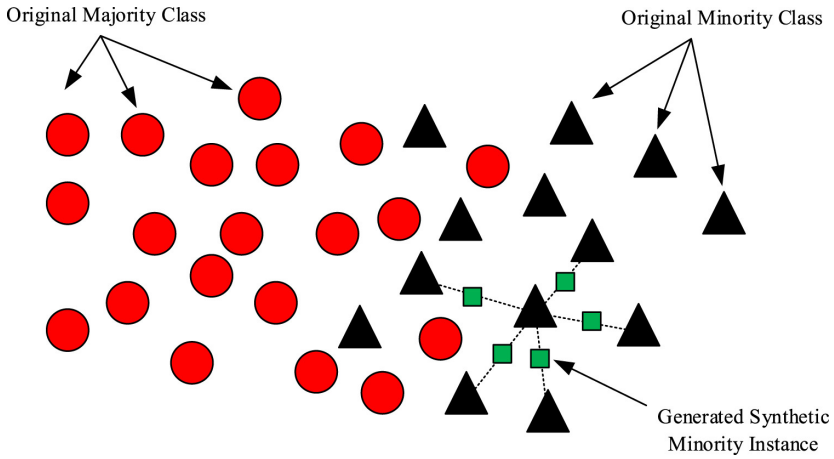

FIGURE 2.6: Synthetic Minority Over-Sampling Technique
(Shen et al., 2019)

Shen et al. (2019) compared their back propagation NN, with PSO weight determination, model against 7 other techniques. These techniques included LDA, logistic regression (Log R in Figure 2.7), SVM, a back propagation NN that did not use the PSO weight determination method (BP in Figure 2.7), KNN, classification trees (CT in Figure 2.7), and a Naive Bayes classifier (NB in Figure 2.7). Each model was trained on the same balanced dataset. Like West (2000), Shen et al. (2019) used 10-fold cross validation to test each developed model. Figure 2.7 displays that their model outperformed the other models in AUC, total accuracy, F1-score, and repaid accuracy (Type I Accuracy in the figure). The model did however not outperform all models in detecting defaulted credits (Type II Accuracy in the figure).

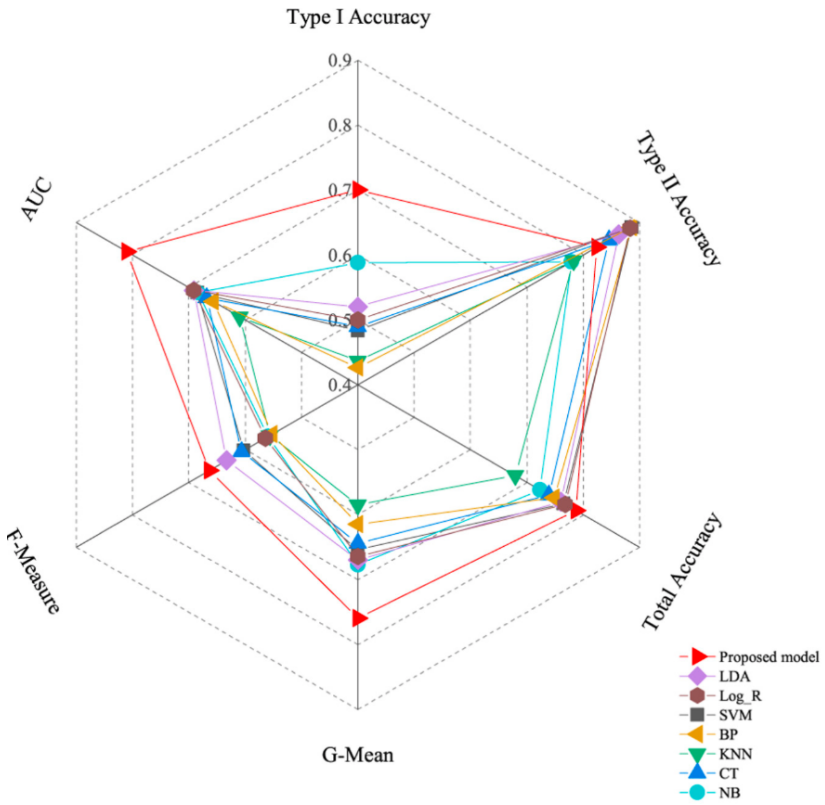

FIGURE 2.7: Shen et al. (2019) Visual Model Performance Comparison

2.4 Ensemble Models

There are two main methods used for ensemble modelling. The first is a parallel structure, which involves developing more than one model from training data and combining their outputs based on an ensemble strategy to produce a final prediction. The second method is a consequential structure, which involves feeding the output of one model into the next until a final outcome is produced. Figure 2.8 visually displays the two main methods.

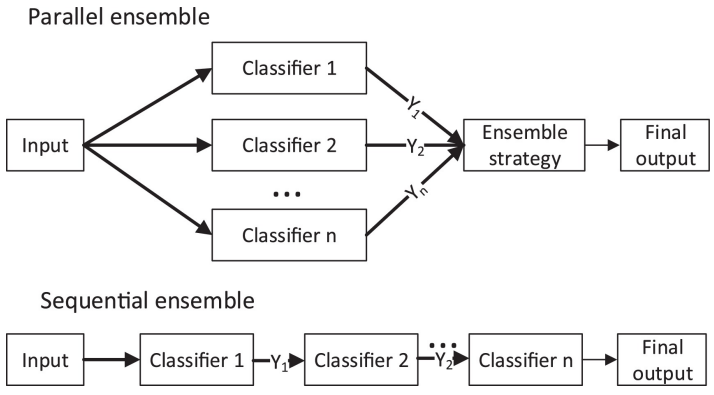

FIGURE 2.8: Parallel and Consequential Ensemble Models (Yufei et al., 2017)

2.4.1 Bagging

Bootstrap aggregating - otherwise known as bagging - is one of the earliest parallel ensemble machine learning methods. It was first developed by Breiman (1996). The method involves developing numerous base models of the same underlying structure. Each model is trained on a separate sub-dataset that is randomly drawn—with replacement—from the entire dataset. The models are combined using majority vote.

Wang et al. (2012) developed a bagging credit scoring model that used decision trees as the underlying base models. In order to further improve the performance of their model and to avoid redundant features impacting the model, when each new decision tree was trained on a sub-dataset not all available features were used. Features were randomly sampled. This feature sampling technique is referred to as random subspace sampling.

Wang et al. (2012) used the same German credit provider dataset used by West (2000) and Shen et al. (2019) to train their model. Wang et al. (2012) further developed a decision tree model, a random forest model, and a bagging model that did not make use of random subspace sampling for comparative purposes. The results from their research can be seen in Table 2.3.

Model	Total Accuracy (%)	Default Accuracy (%)	Repaid Accuracy (%)
DT	72.10	46.80	72.94
Random Forest	77.05	43.72	90.48
Bagging	78.36	41.44	94.02
Bagging with RS	78.52	44.66	92.81

TABLE 2.3: Results of Research Conducted by Wang et al. (2012)

Table 2.3 displays that the bagging model that used random subspace sampling (RS) - for every tree developed - outperformed the other models in overall classification accuracy. It is interesting to note that although the decision tree model had the lowest overall accuracy, it performed best in terms of classifying loans that were not repaid (true negatives). Table 2.3 further displays that all models developed by Wang et al. (2012) had low specificity (classifying loans that were not repaid as bad loans) rates.

2.4.2 Boosting

Boosting is a sequential ensemble machine learning technique that involves altering the weights of samples in training datasets based on the errors of previously created classifiers. Misclassified samples in the training set are assigned with higher weights. A weighted voting scheme is then applied to produce a final model (Freund and Schapire, 1996).

Yufei et al. (2017) used extreme gradient boosting (XGBoost) to develop a credit repayment classification model. New base models in the XGBoost algorithm predict the residuals of previous base models in the sequence. The outputs of each model are then added together to produce a final prediction. The algorithm uses gradient descent to minimise a defined loss function (Cowan et al., 2015).

The base models used in Yufei et al. (2017) were decision trees. The hyper-parameters of their XGBoost model were adaptively tuned using Bayesian optimisation, which involved mapping each hyper-parameter to the loss function and iteratively finding the local hyper-parameter function which minimised the loss function of the XGBoost model.

They further developed baseline models and used other hyper-parameter tuning methods to assess the performance of the model developed. The XGBoost model outperformed the bagging, decision tree, logistic regression, neural network, random forest and support vector machine models in overall prediction accuracy, area under the curve, and Brier score. Furthermore, the XGBoost model developed using Bayesian hyper-parameter optimisation outperformed 4 XGBoost models in overall prediction accuracy, area under the curve, and Brier score. All models developed by Yufei et al. (2017) were trained on five separate credit datasets. The metrics presented were an aggregation of each model's performance across all datasets.

Now that we have considered several types of analyses that have been discuss alternative sources of data that can be incorporated to improve model performance.

2.5 Alternative Data in Credit Scoring

In the decade between 1998 and 2008, the number of micro-finance institutions (MFIs) grew by 474% and their number of customers increased by over 1000%. MFIs generally provide a low amount, short term loans to lower income individuals. In less developed countries many first time customers for MFIs belong to the world's unbanked population. Furthermore, the credit bureaus in less developed countries do not necessarily store and release accurate and reliable data on the banked population. Therefore, traditional credit scoring models can not always be used to predict the creditworthiness of applicants in these regions (Serrano-Cinca et al., 2016).

Blanco et al. (2013) developed a multi-layer perceptron (MLP) network that was trained on over 5,400 loans provided by Peruvian MFIs. The model used features that related to the personal characteristics of the loanees, the economic and financial ratios of the MFI the loan was provided by, the characteristics of the financial obligation (interest rate of loan, loan amount etc.), and variables related to the macroeconomic climate of Peru during the time period of the loan.

Blanco et al. (2013) developed a LDA model, logistic regression model, and multiple MLP models with varying architectures for comparative purposes. The architecture which lead to the most accurate MLP model was a 3 layer perceptron with 20 input nodes, 3 hidden nodes and a single output node. Each model was trained and its parameters tuned using 10-fold cross-validation. Table 2.4 displays the results of the LDA model, logistic regression and most accurate MLP models.

Model	AUC	Default Accuracy (%)	Repaid Accuracy (%)	Misclassification Costs
LDA	0.9303	81.73	93.48	0.5143
LR	0.9322	79.04	94.06	0.5715
MLP	0.9543	84.70	92.24	0.4337

TABLE 2.4: Results of Research Conducted by Blanco et al. (2013)

It can been seen in Table 2.4 that the MLP model has the highest AUC, the lowest misclassification cost and has the highest accuracy in terms of identifying loans that were not repaid.

Óskarsdóttir et al. (2019) investigated the use of alternative data sources to enhance the statistical and economic performance of credit scoring models. They measured the impact of augmenting typical scoring features with features generated from cell phone data. The credit data was provided by a banking institution and the cell phone data was provided by a telecommunications provider. The data provided by the bank contained sociodemographic, account data and credit card repayment data for over 2 million customers. The data provided by the telecommunications company consisted of call data for over 90 million unique cell phone users.

Both data sources were used to generate a connected network between customers of both companies. Figure 2.9 displays an overall view of the network. Creditors were deemed to be a defaulter if they had missed 3 or more credit card repayments.

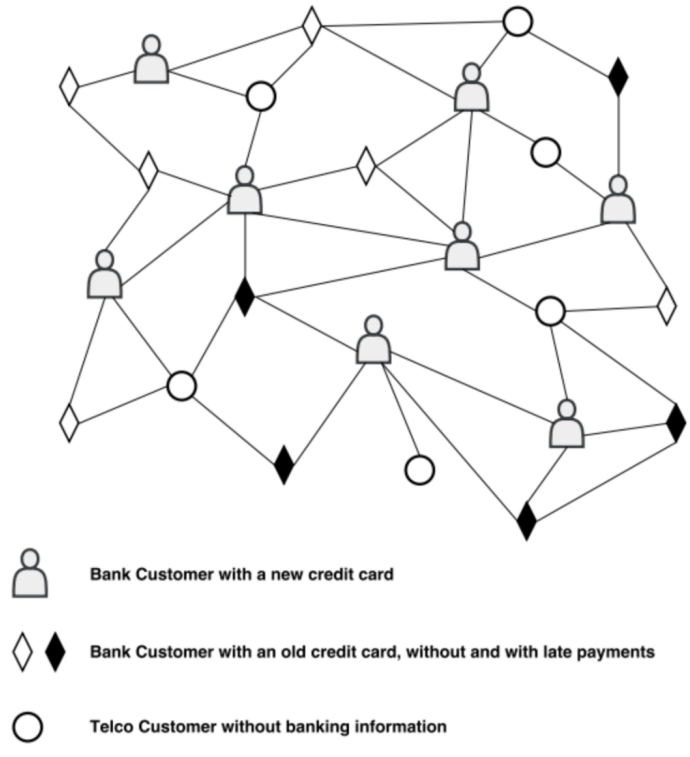

FIGURE 2.9: Network of Bank and Telecommunications Customers
(Óskarsdóttir et al., 2019)

There were a total of 3 networks developed. One for each month that credit cards were disbursed to bank's customers. Bank customers and telecommunications customers that shared a phone call within the three month period prior to card holder's acquisition month were connected. For each network Óskarsdóttir et al. (2019) used network analytics techniques to propagate the influence of related defaulters throughout the network to produce influence scores.

The call data features extracted from the network were as follows: the number and duration of incoming, outgoing and undirected phone calls taking place during the day and night and on different days of the week were computed. Furthermore, exposure scores to defaulted clients were calculated for each customer of each network using Personalised PageRank (PR) and Spreading Activation (SPA).

The features listed above were combined with demographic and account data provided by the bank to build a credit scoring model for a sample of the bank's customers by Óskarsdóttir et al. (2019). This sample included only 22,000 of the bank's 2 million customers. A logistic regression model, a decision tree model and a random forest model were developed for different feature samples. A sample was used that only contained sociodemographic (SD) features, another that only used credit based (CB) features, one that used both SD and CB features, one that used CB features and alternative features generated from the call data and network analysis, and finally one that used SD, CB, and alternative features generated from the call data and network analysis. The AUC of the different feature samples and models can be seen in Table 2.5.

Features	Logistic regression	Decision Trees	Random Forest
SD only	0.5869	0.7004	0.8993
CB only	0.5351	0.7043	0.8700
SD and CB	0.6115	0.7127	0.9227
CB and ALD	0.5182	0.7307	0.9154
SD,CB and ALD	0.6121	0.7263	0.9224

TABLE 2.5: Results of Research Conducted by Óskarsdóttir et al. (2019)

It can been seen in Table 2.5 that the performance of each model type varies substantially. Furthermore, the logistic regression models did not perform better when the cellular-network related features were used. This was believed to be due to linear regression models not being able to capture the non-linear behaviour of the network-related features. The best performing models were the random forest models.

The AUC test devised by DeLong et al. (1988) was used by Óskarsdóttir et al. (2019) to compare the performance of the random forest models. It was discovered - at a 95% confidence level - that the models that used features from multiple of sources (SD, CB, and cellular-network) outperformed the model that used only SD features and only CB features.

However, there was no statistical difference between the model that used SD and CB features, the model produced using CB features and alternative features generated from the call data and network analysis, and the model produce using SD, CB, and alternative features.

2.6 Summary of Literature

This chapter details the origins of credit scoring and how the field has progressed throughout time. The chapter then details the literature behind the popular techniques used to develop loan default prediction models, namely linear discriminant analysis, logistic regression, support vector machines, random forests, XGBoosting, and Neural networks. Then the chapter details the work done in terms of improving loan default prediction models using alternative data sources.

The next chapter will summarise the data sources used to create the datasets used throughout this project, the feature creation techniques used in this project, and the pre-processing completed on the datasets developed for the project before they were used to train various models.

Chapter 3

Data Extraction and Preprocessing

This chapter details the data sources used to train the various loan default prediction models developed throughout this project, the various data extraction techniques used to extract the alternative features within the training sets, and the pre-processing and feature engineering techniques deployed before the modelling phase.

3.1 Data Used

3.1.1 Providers

The models in this project are trained using data from two main sources

The first data provider is a Nigerian micro-finance institution that has disbursed loans to more than 250,000 consumers. The institution is an application (app) based lender and currently only provides credit to android users. The institution, with its customers' consent, gains access to the data on customers' devices. This data includes SMS data, contact data and location data. On top of the alternative data collected, sociodemographic data is collected via customer input on the institution's mobile app.

The second source of data for this project was the Nigerian credit bureaus CRC, CRS and XDS. It is mandatory for credit providing institutions in Nigeria to submit their customers' credit performance data to these credit bureaus.

3.1.2 Personal Data

It is key to note that no personal data - data that relates to an individual or could be used to identify a living individual - was made publicly available, used within the creation of features, or used to train the models developed throughout this project.

3.1.3 Dataset

A final dataset was created for first time loan customers of the micro-finance institution that had existing credit bureau data prior to their first loan application. The customers required existing credit data as it was needed in order to compare the performance of first time credit credit scoring models that use only alternative data or alternative data in conjunction with sociodemographic data, against first time credit scoring models that make use of existing credit data. The final dataset consisted of 62,935 customers/loans.

Both input csv files used to generate the final data set mentioned above - as well as the Python code used to merge the inputs to form the final dataset - can be found in the Github repository mentioned in the appendix attached to this paper.

3.1.4 Data Categories

Three major data categories can be drawn from the data sources. These categories are sociodemographic data, credit bureau data and alternative data. The main aim of this thesis is to assess how alternative data can augment traditional credit scoring data. To complete this aim various combinations of these data categories are used to develop various credit scoring models. These models are a logistic regression model, a random forest model, an extreme gradient boosted model, and a neural network. The statistical performance of the models is assessed in order to test whether using the various data categories resulted in a significant difference in model performance.

3.1.5 Variables Used

Table A.1, which can be found in the appendix attached to this report, displays all variables used to train the loan default prediction models. All sociodemographic variables used are stated by the loan applicants. The variables derived from app and SMS data are scraped from the loan applicants' devices. The variables relating to each applicant's credit history are provided by the Nigerian credit bureaus.

The following figures display the traits of specific independent variables used in the models of this project. The figures show the relationship between the independent variables and the dependent variable (whether a loanee repaid their loan).

Repayment Breakdown

Figure 3.1 displays the breakdown of the first time loans used to train this project's models. It can be seen from the figure that 49,596 of the total 62,935 loans considered for this project were repaid. That means that the default rate of loans considered was 21.19%. This is a high default rate when compared to default rates in Western countries, but in underdeveloped countries - like Nigeria - this aligns with industry standard (Siaw et al., 2014).

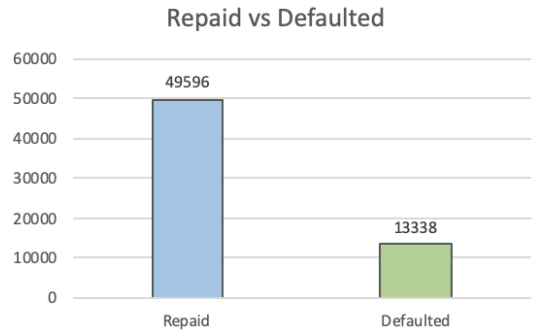

FIGURE 3.1: Repaid vs Defaulted

Age of Loanees

Figures 3.2 and 3.3 display a comparison between the age of the clients that repaid their loan and the age of clients that defaulted on their loan. It can be seen from the figures that the percentage of clients 30 or younger is considerably higher for clients that defaulted compared to clients that repaid. Figure 3.3 is skewed to the right when compared to Figure 3.2, which means that on the whole the sample of clients that defaulted is younger than the sample of clients that repaid.

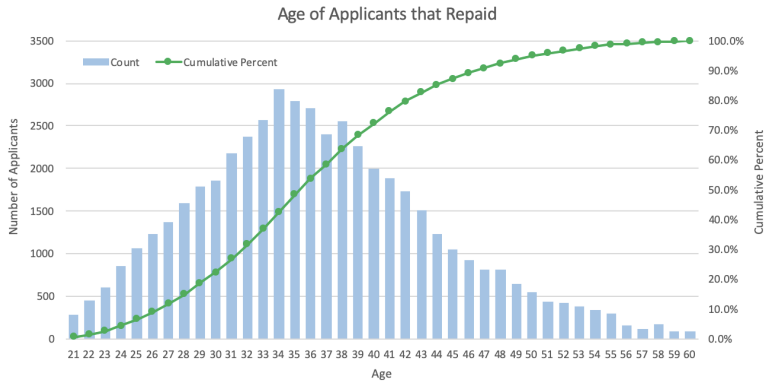

FIGURE 3.2: Age of Clients that Repaid

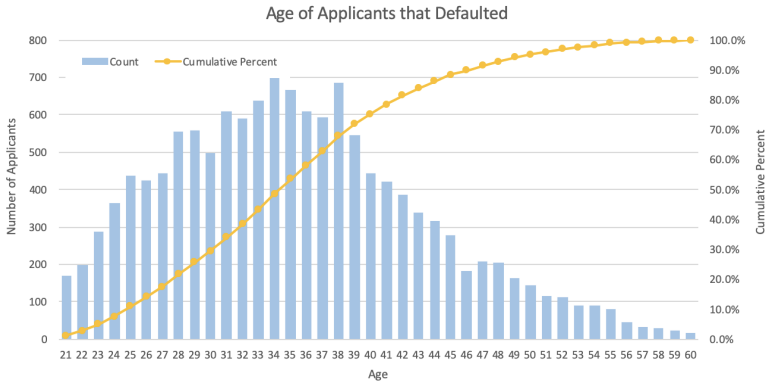

FIGURE 3.3: Age of Clients that Defaulted

Income of Loanees

Figures 3.4 and 3.5 display a comparison between the incomes of the clients that repaid their loan and the incomes of clients that defaulted on their loan. The a larger percentage of clients that defaulted on their loan have an income of 700,000 Naira or less than clients that repaid their loan.

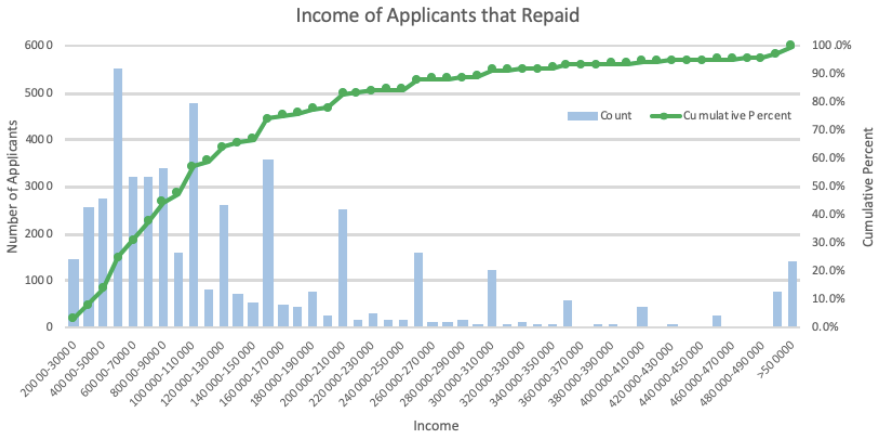

FIGURE 3.4: Income of Clients that Repaid

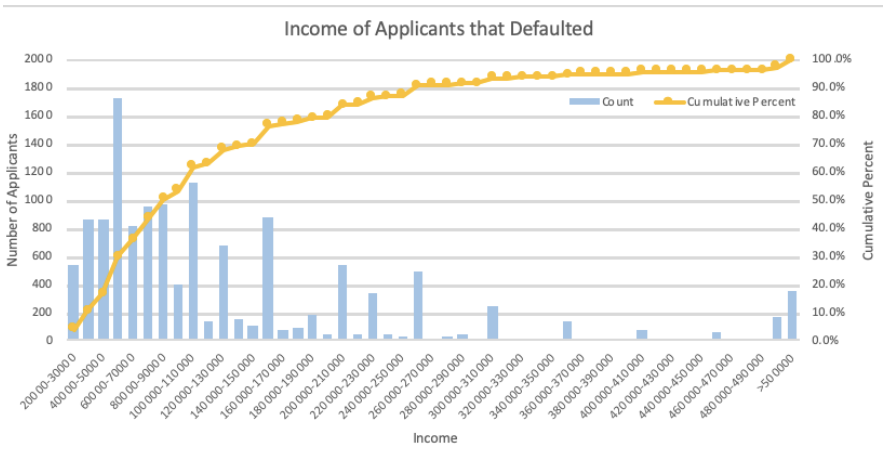

FIGURE 3.5: Income of Clients that Defaulted

Employment Status and Gender

Figure 3.6 displays the various employment statuses of the considered clients. The figure further displays the number of male and female clients contained in employment status and their default rates for their loans.

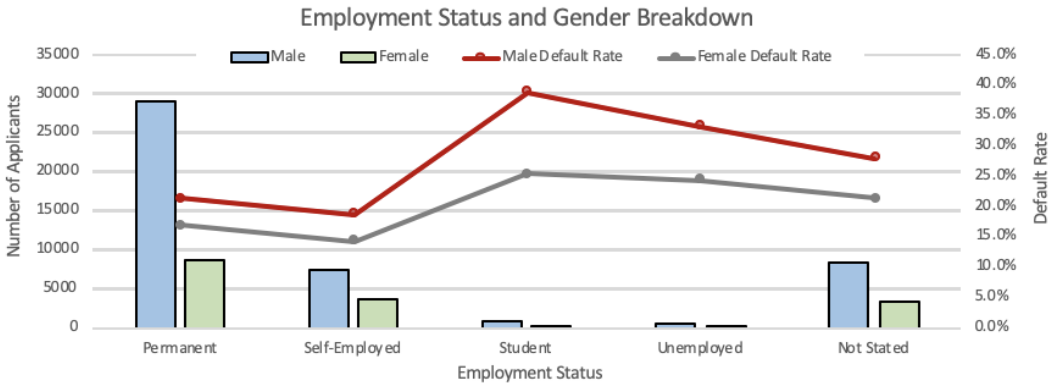

FIGURE 3.6: Employment Status and Gender of Clients

Figure 3.6 shows that the majority of loanees claimed that they are permanently employed and that there were more male loanees contained in the sample than female loanees. The figure further shows that generally students and unemployed loanees are more likely to default. Finally the figure displays that female loanees generally perform better than male loanees.

Credit Bureau Accounts and Gender

Figure 3.7 displays the number of female and male clients that have a specific number of registered accounts with the Nigerian credit bureaus. The figure also displays the default rate by the number of registered accounts. Figure 3.7 - like Figure 3.6 - displays that there are more male loanees in the sample than female and that female loanees tend to repay better than male loanees. Figure 3.7 further displays that the most clients only had 1 account registered with the credit bureaus and that in general the more accounts a client has registered with the credit bureaus, the more likely they are to repay their loan.

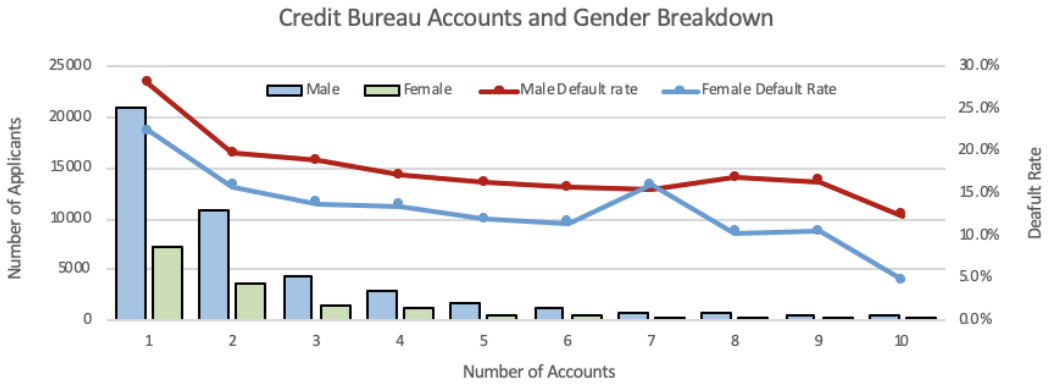

FIGURE 3.7: Number of Credit Bureau Accounts and Gender of Clients

The sources of data and the data itself used throughout the project has now been described. Section 3.2 details how the data is gathered and the processes used to generate the variables used in the models of this project.

3.2 Data Extraction and Feature Engineering

This section details the data extractions and feature engineering techniques used throughout this project. The processes to gather the sociodemographic and credit bureau data require are simpler than those used to gather the alternative data.

3.2.1 Sociodemographic and Credit Bureau Data

The more traditional credit scoring features, developed from sociodemographic and credit bureau data attached to each first-time borrower, were created and extracted using SQL (Structured Query Language). The data was extracted from the company's relational database.

The query was written in a manner that ensured that no data leakage would occur when the credit scoring models were being trained. This means that only data that would be known at the point in time when a particular client applied for their loan could be used to develop features. The only case where data was used that would not be known at the point in time of application was repayment data, as this was used to develop the default (whether the client repaid their loan or not) target variable.

The overall query used to extract the sociodemographic and credit bureau data for each loan was a collection of sub-queries joined on a unique key attached to each loan. The query used to extract the credit bureau required an aggregation in order to generate features that represented the total number of loans each client had prior to their loan application with the micro-finance institution used in this study. The credit bureau features used in the modelling process and their definitions can be found in the appendix of this paper.

3.2.2 Alternative Data

The three main sources of alternative data used to develop features are the app-based, SMS, and device data stored on each customers' cellphone. The data is extracted from one of the micro-finance institution's databases using PyMongo, a Python package that allows a user to query data from a Mongo database from within a Python script.

The app-based and SMS data is extracted from a different Mongo databases, however regular expressions (regex) are used to filter both data types and to develop features. Regex functions are sets of sequences of characters that define a particular search pattern. The functions are then used to identify cases of the defined pattern in strings (Aho, 1990).

The device data is extracted from a separate database than the app and SMS data and an entirely different technique is used to collect the data. The technique used in this case is web scraping, which is a method of extracting data from websites (Waddell and Boeing, 2016).

App-Based Features

The app-based features engineered for this project are counts of particular apps present on a client's device at the point in time of their loan application. The features included a count of the financial, competing micro-finance, news, gambling and virtual private network (VPN) apps. The counts are generated by first compiling a list of all unique apps on a client's device. Then the name of each app is passed through a series of regular expression key word searches. Each expression is designed to detect a specific app type. If a particular app type search results in a match, the count associated with that search is updated.

The process developed to pass an app through the app extracting regular expressions and how the count features are generated throughout this process is represented in Figure 3.8. The process is completed for every unique app recorded on the client's device.

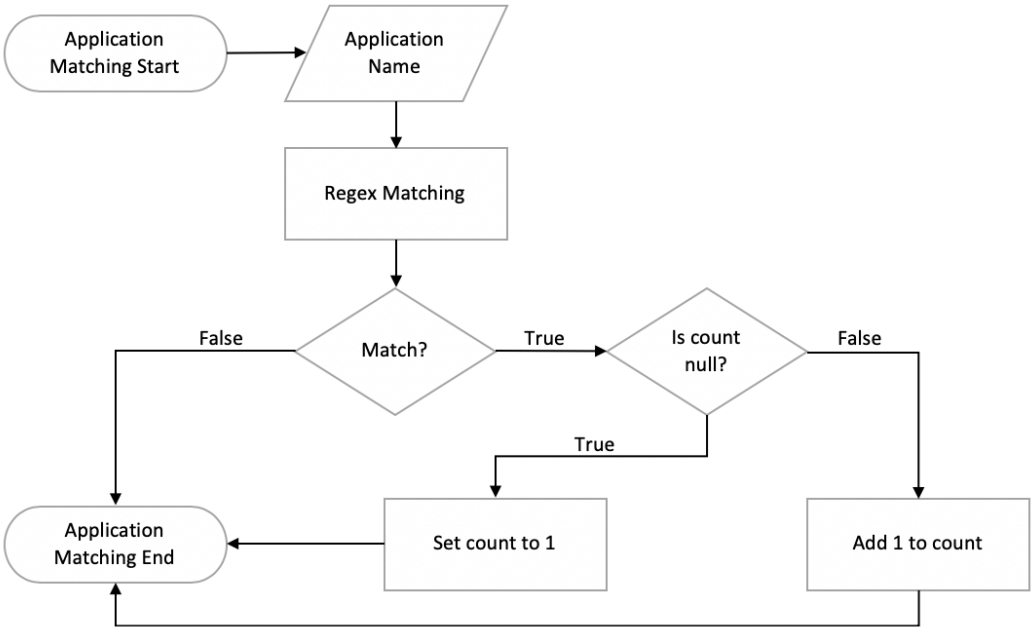

FIGURE 3.8: App-Based Feature Generation

SMS-Based Features

The SMS data consists of messages received by the clients in the 90 days prior to their loan application: by nature this data is more sensitive than the other data used throughout this research. Similar to the process developed to generate the app-based features, each message received by a client is compiled into a list. Each individual message in their list is then passed through a series of regular expressions in order to generate features.

In order to avoid exposing personal messages, each message was passed through two filtering regular expressions. The first expression returned only messages received from Nigerian banks, while the second ensured that only messages returned by competitor micro-finance institutions were returned. The regular expressions had a dual purpose: they prevented exposure to sensitive content and they acted as the first step in the SMS-based feature generation process.

If a message passes through the regular expression for banking messages it is exposed to the banking feature creation process. Typically, messages from Nigerian banks have a similar structure. They display a transaction amount, the date of the transaction, the type of transaction (credit or debit to the account), and finally the balance in the account after the transaction. Regular expressions are used to extract these features and store them as either numeric variables or lists.

If a message did not pass through the first filter regular expression - searching for bank messages - it is then passed through the competitor expression. If the message passes through this expression it is then further screened by another set of regex functions. These functions search for key words in order to determine if a client had another loan with a competitor and if that loan had been repaid successfully or not. The actual loan amount is extracted using regex, as is the loan repayment (instalment amount). These amounts were appended to lists.

After passing every message associated to a particular client through the regex functions the lists created throughout the process are used to generate the SMS-based features for that particular client.

The banking related features generated were:

- The number of unique banks that sent the client a message
- The minimum and maximum debit transaction, credit transactions and account balance values
- The total number of debit and credit transactions
- The number of times the term 'insufficient funds' is found in the client's messages

The process of passing an SMS through the bank related regular expressions and how the banking features are generated throughout that process is represented in Figure 3.9. The process is completed for every message received by a client within the 90-day period prior to their loan application.

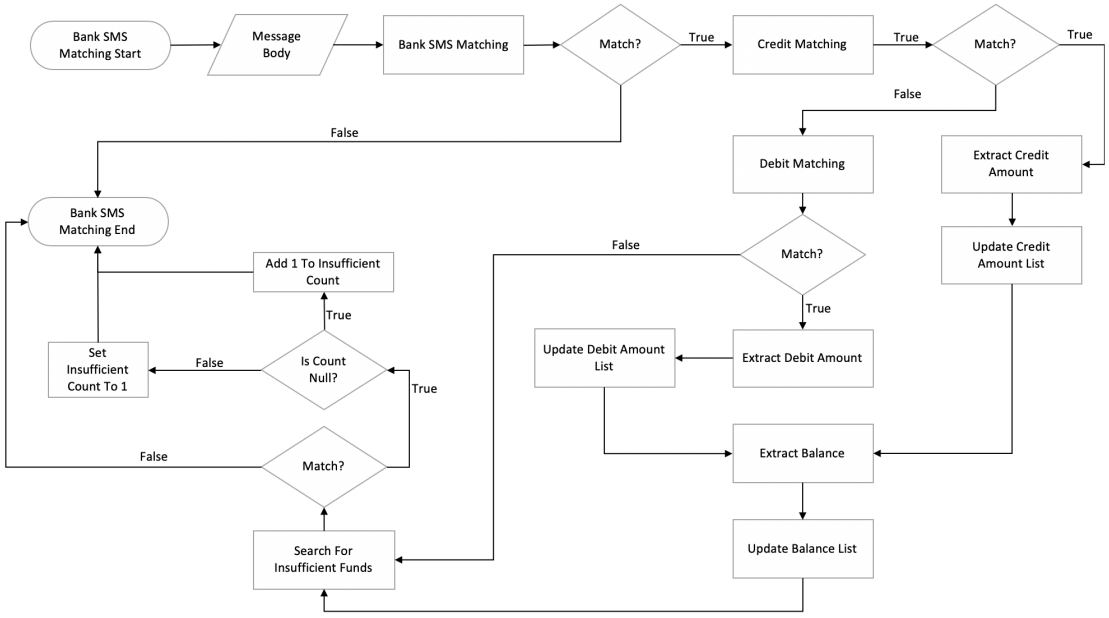

FIGURE 3.9: Bank Based Feature Generation

The competitor related features generated were:

- The number of competitors that sent the client a message
- The number of competitors that approved a loan for the client
- The minimum and maximum loan amount received by, successful loan repayment made by, and unsuccessful loan repayment made by the client
- The number of loans received by, successful loan repayments made by, and unsuccessful loan repayments made by the client
- The number of rejected loan applications made by the client.

The process of passing an SMS through the competitor related regular expressions and how features are generated throughout that process is represented in Figure 3.10. The process was completed for every SMS message received by a client within the 90-day period prior to their application.

Web Scraping

The unique Android ID attached to each customer's cellular device - used when applying for their loan - is used to ascertain the brand and model of the device as well as the device's operating system version. The brand of device and operating system are used directly as features while the brand name and device model were used in conjunction to scrape the price of the device.

The script written to scrape and derive the device price was written in Python and made use of the Beautiful Soup web scraping package. The price of each device was scraped from Jumia and Kara, two of the biggest Nigerian e-commerce platforms.

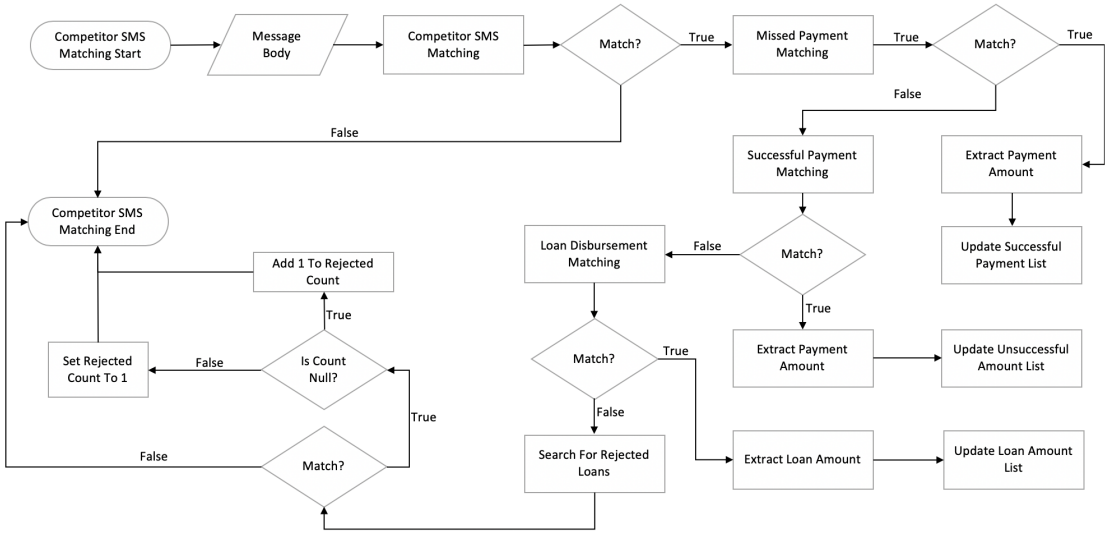

FIGURE 3.10: Competitor Based Feature Generation

The logic used to derive a price for each customer's cellular device is shown in Figure 3.11.

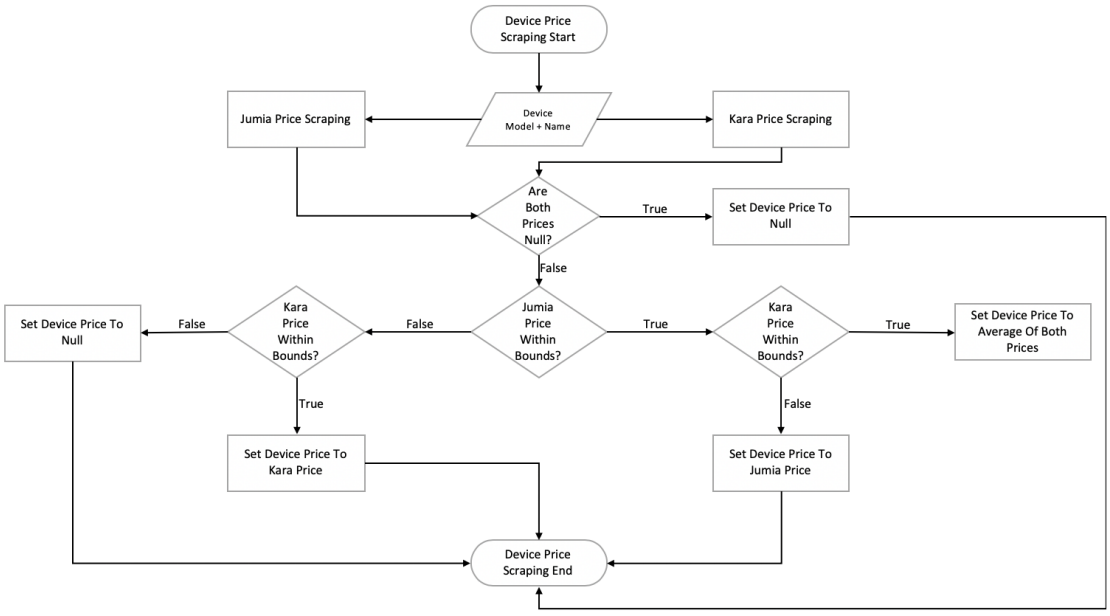

FIGURE 3.11: Device Price Logic

Figure 3.11 displays the possible ways in which a device price could be determined. The possibilities are as follows: a price could not be scraped from either site, therefore, the price was set to null; a price could be scraped from one site but not the other, the scraped price was within the price bounds, then the one price was used; a price could be scraped from both sites, both prices were within the price bounds, then the price was set to be the mean of both prices. Lower and upper price bounds were introduced to reduce the number of scraping miss-classifications and as a result improve data integrity. Unreasonably low prices were often device accessories such as phone cases or screen protectors, while unreasonably high prices were often laptops or other more expensive electronic devices.

This section detailed the gathering of data and the processes used to develop features. Section 3.3 details how the features are processed before being used in the loan default prediction models.

3.3 Preprocessing

This section details how the various types of dependent variables are scaled, how missing values are handled, and how outlier data points are identified.

3.3.1 Scaling and Encoding

In machine learning projects, numeric scaling and categorical encoding is often conducted after imputing missing values. In the case of this project a K-Nearest Neighbours (KNN) model was developed to impute the missing values. The KNN algorithm involves calculating the Euclidean distance between each data point in the dataset in consideration and every other point in that set. If features are not scaled prior to calculating the distances between points, certain features may skew the calculated distances (Deng et al., 2016). Therefore, numeric variables were normalised and categorical variables scaled prior to imputation.

Numeric Variables

The numeric variables used throughout this project were standardised before the modelling process. Standardising numeric features involves transforming the values of each variable so that the values of variable so that the mean is 0 and the standard deviation is 1. This is done for a single variable by first calculating the mean and standard deviation of the variable and then replacing each value by its respective z-score (Cheadle et al., 2003).

The Z-score of each value is show in Equation 3.1, where x each value.

$$z_i = \frac{x_i - \mu}{\sigma} \tag{3.1}$$

The above transformation was done for every populated value of each variable in the dataset. It is key to note that missing values remained missing.

Categorical Variables

The categorical variables contained in the dataset of this project were encoded using weight of evidence (WoE) encoding. This is a common approach for handling categorical variables within the credit risk and financial industries (Siddiqi, 2006). WoE encoding scales the levels of a categorical predictor variables based on their relationship with the target variable. In terms of loan default prediction models, WoE scales the levels of each categorical variable with respect to loan default (Siddiqi, 2006).

WoE encoding handles missing values for categorical variables. In the case of the default prediction models, missing values are consider to be missing not at random, this is because applicants may withhold information while completing loan applications to increase their chance of being granted a loan. This is further explained in sub-section 3.3.2. WoE encoding places missing values into a category and assigns a scaled value to them.

The method for calculating the WoE of each level is shown in Equation 3.2, where probability of repaid (POR) and probability of defaulted (POD) are the proportion of customers per level that repaid and defaulted respectively.

$$WoE = \ln\left(\frac{POR}{POD}\right) \tag{3.2}$$

An example of the WoE scores for one of the features can be seen in Table 3.1. It is key to note that all WoE scores are scaled using the standardisation method explained in Equation 3.1 before being used in the KNN imputation model.

Level	WoE
Single	-13.16
Married	16.15
Widowed	6.54
Separated	13.82
Missing	93.80

TABLE 3.1: Weight of Evidence Values for Marital Status Variable

3.3.2 Missing Values

Missing values are an issue that need to be addressed during any data science project, however missing data is especially significant in credit risk related modelling. Gathering complete credit repayment data is the most important factor when developing credit risk models (Soley-Bori, 2013).

Target Variable

Repayment data is often sparse and complex. Many consumers have missing values based on incompleteness but others have missing values based on the fact that credit term has not been reached. These challenges make it difficult to develop statistically significant datasets required for credit repayment prediction models (Florez-Lopez, 2010).

The loan repayment data used in this minor dissertation is complete. Only loanees that had completed their entire loan tenor are used in the dataset. This ensures that the target variable - loan default - does not contain any missing values.

Predictor Variables

The features used to predict repayment do contain missing values. The missing values need to either be removed or imputed. Firstly, the percent of missing values per predictor variable is assessed to ensure that no more than 50 percent of the values within each variable are missing. This is also done for each row (customer/loan). No more than 50 percent of the values contained in a variable or in a particular row were missing.

Figure 3.12 displays a nullity correlation heat-map of the variables within the dataset used throughout this project. A nullity correlation between two variables ranges from -1 to 1. A value of -1 indicates that if the one variable appears then the other will definitely not appear. A value of 0 indicates that the appearance of the one variable does not influence the appearance of the other variable. A value of 1 indicates that the one variable is always present when the other one is (Bilogur, 2018).

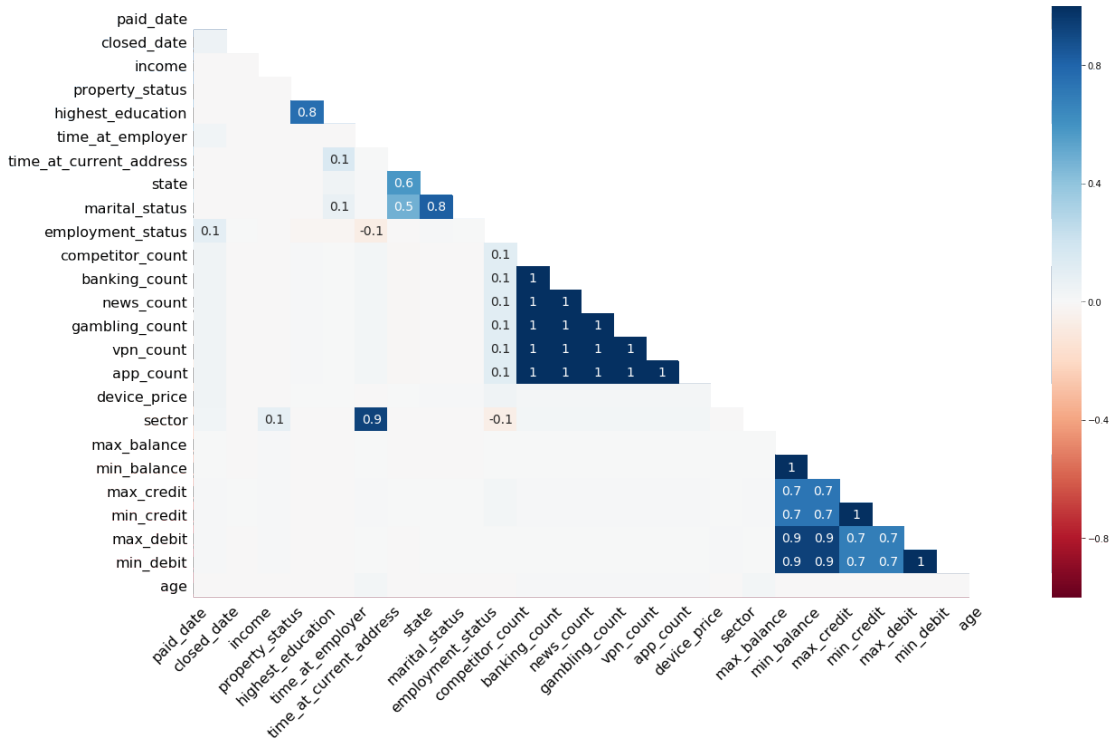

FIGURE 3.12: Nullity Correlation

It can be seen from Figure 3.12 that no particular feature disparages the presence of another. However, the presence of certain features strongly correlates with the presence of other features. This is expected as no feature is sparsely populated.

In data science projects, rows containing missing values are often removed from the dataset (Horton and Kleinman, 2007). In the case of this project excluding all cases containing a missing value was not feasible as too few samples would have remained to train and test a valid model. This is because more than 50 percent of rows within the dataset contained at least one missing value.

Table 3.2 displays the count of applicants, the cumulative count of applicants, and the percentage of total applicants that have the number of missing values shown in the first column of the table. Table 3.2 displays that 2.81% of applicants have more than 10 missing values, while no loan applicant had more than 14 missing values. The total number of dependent variables used is 53.

Missing Values	Count	Cumulative Count	Percentage of Total
0	28,808	28,808	45.77
1	6,276	35,084	55.75
2	6,275	41,359	65.72
3	1,495	42,854	68.09
4	577	43,431	69.01
5	97	43,528	69.16
6	12,500	56,028	89.03
7	3,516	59,544	94.61
8	1,168	60,712	96.47
9	376	61,088	97.07
10	75	61,163	97.19
11	21	61,184	97.22
12	1,046	62,230	98.88
13	623	62,853	99.87
14	81	62,935	100.00

TABLE 3.2: Number of Missing Values by Observation/Applicant

Another common missing value imputation technique involves replacing missing values with the mean (continuous variables) or mode (categorical variables) of their respective variable. This is a successful technique if the variables are considered to be missing at random. In the case of this project missing values were consider to be missing not at random. This is due to the fact that the loanees manually filled certain variables during their loan applications. Loanees may have withheld or altered variables based on how they thought it would affect the outcome of their credit application (Soley-Bori, 2013).

There are many methods for imputing missing values where the values are missing not at random. The two methods explored were SVD (singular value decomposition) and k-nearest neighbours.

SVD involves calculating a matrix's mutually orthogonal eigenvectors. The most important eigenvectors are then linearly combined in order to best predict the missing values of the matrix. In the case of the dataset used in this project, each loan application would be considered as a matrix row and the predictor variables would be the respective columns. An issue with SVD imputation is that the predictions for missing values are calculated using the most important eigenvectors and not all eigenvectors. Therefore, in terms of this project, unusual loan cases would not be well represented by the leading eigenvectors and as a result their missing values may not be accurately filled. This lead to the KNN approach being used (Troyanskaya et al., 2001).

The KNN approach involves filling the missing values of a particular row with the average value of the equivalent variable from the row's K "most similar" neighbours. Similarity can be calculated using various distance metrics, for example Euclidean, Minkowski, and Manhattan distances. Equation 3.3 shows the Euclidean distance between points x and y. The distance is calculated by taking the square root of the sum of the squared differences between the respective variables of each point (Howarde, 1994).

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} \tag{3.3}$$

Before the KNN model was developed to replace the missing values in the dataset, each variable was normalised and scaled. This was done for both categorical and numeric variables. The methods used to do this are explained in sub-section 3.3.1.

After normalisation and scaling, the following steps were completed in order to fill missing values for each loan in the dataset using the KNN approach:

- Set the number of nearest neighbours to be considered for each loan to 3 (arbitrary selection).
- Check if the loan had any missing values. If the loan did not have missing values move to the next loan, if it did then continue to the steps below.
- Calculate the Euclidean distance between the loan under consideration and every other loan.
- Identify the 3 closest loan applications based upon Euclidean distance.
- Fill each missing value with the mean value of the respective variable taken from the loan's 3 nearest neighbours.

3.3.3 Outlier Detection

Outlier detection involves identifying observations that have features that do not conform to the typical patterns of the features of other observations (Khan et al., 2019). In this project, outliers are detected using the isolation forest algorithm, which is an unsupervised extension of the decision tree algorithm. The isolation forest algorithm does not require distances or densities to be calculated between data points to identify outliers, which leads the algorithm to have a low computational cost (Liu et al., 2008).

The isolation forest algorithm involves training a decision tree - in a unsupervised manner - by recursively splitting a dataset until each observation becomes terminal node on the tree. This process is displayed in Figure 3.13. We can see from the figure that the furthest right observation was isolated after only two splits. While, the highlighted observation towards the bottom of the diagram passed through 5 feature splits before becoming isolated.

The depth of each observation - the number of iterations before it is isolated is recorded. This process is repeated multiple times and the average depth for each observation is calculated. Observations with a very small tree depth - relative to the depth of the other observations in the dataset - are considered to be anomalies (Liu et al., 2008).

In the case of this project, loan applicants with features that greatly vary from the features of the other applicants are identified by training an isolation model. Applicants that are deemed as outliers - 2,261 of the total 62,935 - by the isolation forest model are removed from the final dataset used to train the loan default prediction models. A default contamination ratio (suspected ratio of outliers in the dataset) of 0.1 is used to remove the outliers. The default ratio is used as no expected ratio could be attained.

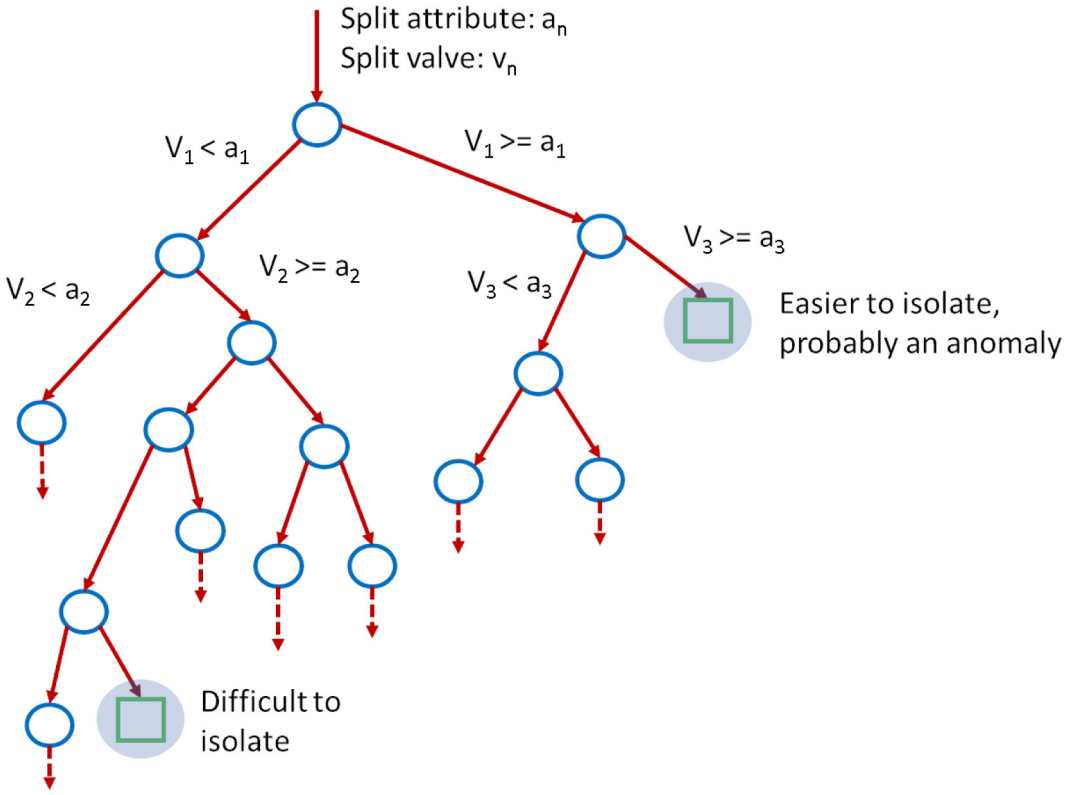

FIGURE 3.13: Isolation Forest Principle
(Khan et al., 2019)

3.4 Summary of Data Extraction and Preprocessing

Firstly, this chapter details the three data sources that are used in this project, namely sociodemographic data provided by the Nigerian micro-fiance company that provide the loan data, Nigerian credit bureau data, and the alternative features developed through web-scraping and regular expressions. Finally, the chapter details the pre-processing techniques used prior to modelling, which includes handling missing values, scaling variables, and handling outlier observations.

The next chapter details the 4 modelling techniques used throughout this project, namely logistic regression, random forest, XGBoost, and neural networks. Finally, the chapter discusses the feature selection methods used and how the various models are tested.

Chapter 4

Modelling Methods

After all features are created, missing values handled, and outliers removed, the next step in this project is to train the various loan default models. There are 4 modelling techniques used in this project, namely logistic regression, random forests, extreme gradient boosting and artificial neural networks. This chapter details the datasets used to train the various models, how the features used in each model are selected, how the parameters of each model are tuned using grid search, and how each model is validated.

It is key to note that the Python code containing all the models trained and testing throughout this paper, their hyper-parameter tuning, and their validation can be found in the Github repository attached to the appendix of this paper.

4.1 Datasets Used

The main focus of this project is to assess if alternative data improves loan default prediction models when it is used to augment traditional credit scoring data. A secondary aim of the project is to test if accurate loan default prediction models can be developed using the alternative data features developed throughout this project. These aims are tested by training models - belonging to each of the 4 mentioned techniques - on all possible combinations of the 3 data categories listed in Section 3.1. The possible combinations can be seen below:

- sociodemographic (SD) only
- credit bureau (CB) only
- alternative data (ALD) only
- SD and CB
- SD and ALD
- CB and ALD
- SD, CB and ALD

For each data category combination, the same dataset is used to train a model belonging to each of the 4 techniques mentioned. This means a total of twenty eight models are trained. Every dataset created contains the same 60,674 loans/applicants. The number of applicants that repaid and defaulted on their loans are 47,960 and 12,714 respectively. To avoid introducing a bias towards the majority class (repaid clients) in the loan default prediction models, the classes need to be balanced before training the various models.

4.2 Class Balancing

Imbalanced classes are a common problem in many classification projects. They occur when the number of observations representing one class is much lower than the number of observations representing the other classes. Imbalances pose a major issue when the "cost" of misclassifying the minority class - the class with far fewer observations - outweighs the cost of misclassifying the majority class (or classes), for example the classification of cancerous cells in medical images (Galar et al., 2012).

In the case of this project, the cost of misclassifying an applicant that is likely to default on their loan outweighs the cost of misclassifying an applicant that is likely to repay their loan. If an applicant defaults on a loan the micro-finance company granting the loan loses the loan amount lent and the potential interest that would have been gained on the loan (minus any repayments made). While, if an applicant is simply not granted a loan then the company will only lose the potential interest that would have been gained if the applicant repaid their loan.

There are a variety of class balancing techniques that can be used in machine learning projects. The techniques can be broken down into two categories, namely algorithmic solutions and data level solutions. Algorithm solutions directly modify the weight that each observation has on the loss function of the model being trained whilst data level solutions

alter the dataset used to train models.

Loss functions are used to evaluate the deviation between a model's predictions and the actual values in the data. Equation 4.1 represents a loss function where L(X,y,β) is a loss function that measures how well a model, parameterised by β, fits the data X. γP(β) is a penalty function on the parameter vector β and its impact on the model is controlled by its γ tuning parameter (James et al., 2013).

$$L(X, y, \beta) + \gamma P(\beta) \tag{4.1}$$

Loss functions are minimised throughout training so that prediction error is decreased. Data level methods either involve over-sampling (copies of the minority class observations are generated) or under-sampling (only a certain percentage of the majority class observations are selected used to train the model) (Raghuwanshi and Shukla, 2019).

Class balancing is completed for each of the variable combinations (datasets) detailed in Section 4.1. However, balancing is only conducted after each dataset is separated into a training and a test set, with each training set containing 80% of the total observations and each test set containing 20% of the total observations. Balancing is only completed after performing a train/test split so that it is possible to test for the occurrence of over-fitting due to the balancing (Galar et al., 2012).

For each of the data category combinations, SMOTE - used by Shen et al. (2019) as detailed in Chapter 2 - is used to over-sample the minority class. This involves generating fictitious applicants by identifying the 5 (arbitrary selection) nearest neighbours of a randomly selected applicant that defaulted. After, the variable difference vector between the applicant's features and the features of the 5 nearest neighbours is calculated.

The variable difference vector is then multiplied by a random value between 0 and 1 and then added to features of the randomly selected applicant. This process is completed until the number of applicants that defaulted matches the number of applicants that repaid in each dataset.

After each dataset is balanced, the next step is to select the most relevant features in each dataset to be used in the modelling process.

4.3 Feature Selection

Feature selection is the process of determining the most relevant predictor variables for modelling purposes. Guyon and Elisseeff (2003) found that feature selection can increase the overall accuracy of a model, while decreasing the training and prediction times of a model (particularly when the training data is large). Furthermore, they found that subset and correlation coefficient selection methods outperformed other methods.

The following sections detail the feature selection methods used throughout this project. Initially, correlation coefficients are used to remove variables with very strong relationships to other variables. Secondly, recursive feature selection - a subset feature selection method -

is used to identify the most relevant features in each of the 7 datasets used throughout this project.

4.3.1 Correlation

Firstly, the Pearson's correlation coefficient between all independent variables is calculated. Pearson's correlations is a measure of a linear relationship between two variables (Schober et al., 2018). Figures 4.1, 4.2, and 4.3 display the correlation between the alternative, sociodemographic, and credit bureau variables respectively.

Figure 4.1 displays that there is a very strong positive correlation between the minimum debit and minimum balance variables. Figure 4.2 shows that there are no strong relationships between the sociodemographic variables. In Figure 4.3 there is a very strong positive relationship between the total number of accounts registered with credit bureaus and the total number of performing loans registered with credit bureaus.

To avoid capriciously choosing feature selection cut-offs based on correlation values, only very strong correlation values, higher than 0.9, are used for feature selection purposes (Schober et al., 2018). This process resulted in a single variable being removed.

Both the variations in variables that have strong relationships with other variables and the variations in variables that do not, cause variations within a model (Xie et al., 2018). Therefore, Recursive Feature Elimination (RFE) is used to select only the relevant features before each of the models are trained.

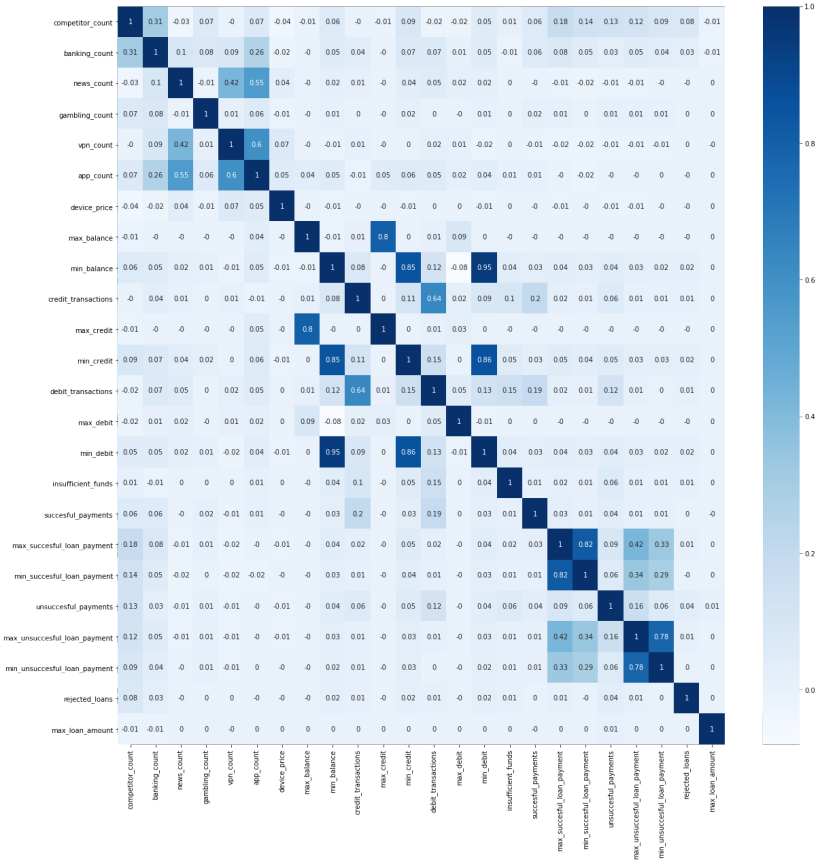

FIGURE 4.1: Correlation Between Alternative Data Variables

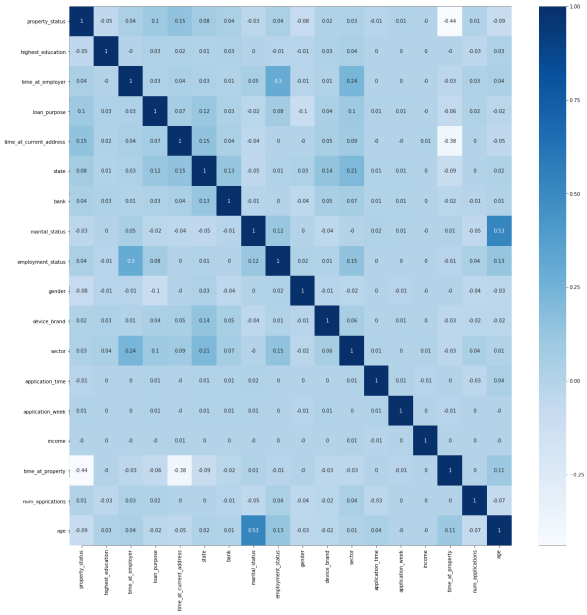

FIGURE 4.2: Correlation Between Sociodemographic Variables

FIGURE 4.3: Correlation Between Credit Bureau Variables

4.3.2 Recursive Feature Elimination

Feature selection is completed for each of the 7 datasets using the same technique. This is done so the effect of adding the alternative data features can be directly measured for each modelling technique.

RFE is an form of subset feature selection. The technique involves recursively training a model and removing the uninformative features. For each training iteration the features used are ranked based on their importance. The weakest feature is removed from the training set and the model is retrained. The optimal number of features is determined by the accuracies of the models produced. This process is often validated using cross validation (Bahl et al., 2019).

RFE reduces the dimensionality of a feature space by removing uninformative features. This decreases the training time of models, improves model performance, and eliminates dependencies and collinearity that may exist in the training data (Bahl et al., 2019)

The RFE feature selection process completed in this project uses a linear support vector machine model as the underlying model. SVM-RFE has been successfully applied to many classification models. The technique is not prone to over-fitting and has been proven to be an accurate and fast feature selection method (Yan and Zhang, 2015). The optimal number of features for each dataset is determined based upon the accuracies of the models produced and is validated using k-fold cross validation.

4.3.3 Cross Validation

K-Fold cross validation involves partitioning the training sample of a model into k partitions. One partition serves as an independent holdout test set for the credit model being trained while the remaining k-1 partitions are used to train the model. This process is then iterated over for all k partitions. This technique minimises the effects of data dependencies and improves the reliability of the estimates (Ling et al., 2019). Figure 4.4 displays the principle of k-fold cross validation.

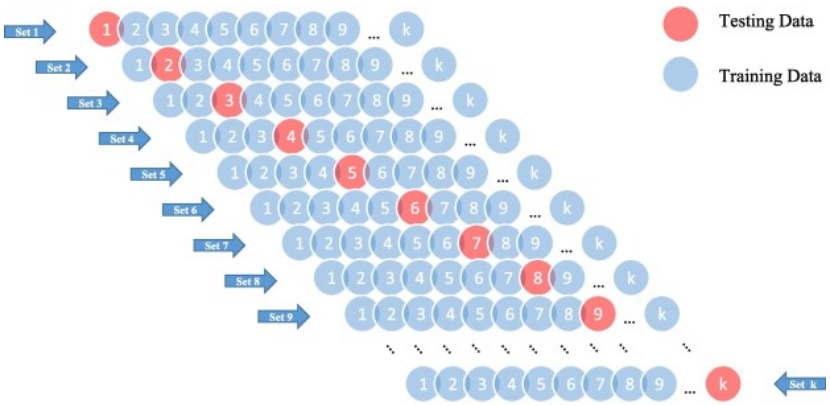

FIGURE 4.4: K-Fold Cross Validation
(Ling et al., 2019)

In the case of this project, each feature selection process is validated using 5-fold cross validation. The k of the cross validation is set to 5 due to computational limitations.

After the most relevant features for each of the 7 datasets are selected, the optimal hyper-parameters for each models need to be determined. This is referred to as parameter tuning.

4.4 Hyper-Parameter Tuning

Random-search and grid-search are two of the most widely used strategies for hyper-parameter tuning in machine learning projects. Random-search is a more efficient method, but often does not lead to the most accurate models being developed (Bergstra and Bengio, 2013).

Grid-search is often considered the brute force method for hyper-parameter optimisation. It involves defining possible values for each hyper-parameter of a model, then training a model for every possible combination of the defined values.

Throughout this project, grid-search, in conjunction with 5-fold cross validation, is used to tune the hyper-parameters for each of the 28 models developed.

4.5 Modelling Techniques

Once feature selection and hyper-parameter optimisation has been completed, all models are trained using only those features and parameters. A total of 7 models - one for each dataset - for each of the techniques used are trained. After training, each of the models developed are tested using a withheld 20% validation sample (created prior to class balancing). This section details each technique used and the parameters tuned for each technique.

4.5.1 Logistic Regression

Wiginton (1980) first used logistic regression to develop a loan default prediction model. Since then the technique has become one of the most widely used in loan default prediction models. This is due to the technique's robustness and transparency (Dong et al., 2010).

As shown in Equation 2.2, logistic regression models - when used for loan default prediction - output the probability of an applicant defaulting on their loan. Logistic regression models are trained using maximum likelihood and the coefficients of each independent variable are estimated in such a way as to minimise the loss function. The impact of each independent variable on predictions can be directly measured through its coefficient and the importance of the variable can be determined using the variables Z score - its coefficient divided by its standard deviation. The clarity of each coefficient and its impact is what makes logistic regression models so transparent (Hastie et al., 2008).

The first phase in training each of the 7 models is to tune their hyper-parameters using grid-search and cross validation. The following hyper-parameters are tuned for each logistic regression model; the number of iterations completed during training, the regularisation method used, and the size of the lambda factor in the regularisation penalty.

A regularisation penalty is added to the loss function of a model to shrink the size of the coefficients of the variables used. This is done to avoid over-fitting and ensure that the model generalises well (Albon, 2018). In the case of the logistic regression models developed, either L1 or L2 regularisation is used. The penalty terms of L1 and L2 regularisation when used in logistic regression can be seen in 4.2 and 4.3 respectively.

$$\max_{\beta_0,\beta} \left\{ (\sum_{i=1}^{N} [y_i(\beta_o + \beta^T x_i) - log(1 + e^{\beta_o + \beta^T x_i})] - \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{4.2}$$

$$\max_{\beta_0,\beta} \left\{ (\sum_{i=1}^{N} [y_i(\beta_o + \beta^T x_i) - log(1 + e^{\beta_o + \beta^T x_i})] - \lambda \sum_{j=1}^{p} \beta_j^2 \right\} \tag{4.3}$$

The major difference between L1 and L2 regularisation is that L1 regularisation shrinks the coefficients of unimportant variables to zero, while L2 regularisation only shrinks the coefficients of unimportant variables towards zero (Hastie et al., 2008). L2 regularisation adds the squared magnitude of a coefficient as penalty term to the loss function, while L2 regularisation adds the absolute magnitude (Hastie et al., 2008).

After identifying the optimal hyper-parameters for each logistic regression model, each model is retrained using its optimal features and hyper-parameters. The models are then tested using a withheld validation set.

The next technique used is random forests. The process followed to train the random forest models is discussed in the next section.

4.5.2 Random Forest

When applied to a classification problem, the random forest algorithm involves training multiple decision trees - on independently sampled training sets (bootstrapped from the same overall sample) and then combining the results of the various classifications of the trees using a voting process (Breiman, 2001).

Each tree in the forest is grown while meeting the following conditions:

- If there are N observations in the overall training sample, a training set is created by sampling N observations - with replacement - from the overall sample.
- If there are M dependent predictor variables in the overall training set, the sample training set consists of m (where $m <= M$) randomly sampled predictor variables.
- Each tree trained is grown to its largest possible extent (no pruning is completed) (Breiman, 2001).

Wang et al. (2012) and Óskarsdóttir et al. (2019) have shown that the random forest algorithm can successfully be applied to loan default prediction. In the case of this project a random forest model is trained for each of the 7 data category combinations. Each model is tested using an unseen validation set. During training the hyper-parameters of each model are tuned using a grid-search.

Parameter Tuning

Each random forest model has the following hyper-parameters tuned during training:

- The number of trees grown in the forest.
- The maximum number of features used within a particular forest.
- The maximum depth (the maximum number of splits) of each tree in the forest.

Tuning the above hyper-parameters of each random forest model allows for the most accurate and generalizable models to be developed. The process ensures that the trees contained within each random forest model are decorrelated, thus reducing the likelihood of overfitting to the training data (Probst et al., 2019).

The criterion used to measure the quality of splits within each tree is Gini impurity, this is the same measure used by Zekic-Susac et al. (2004). Another criterion option is Shannon Entropy, however this is more computationally intensive than Gini impurity (Khaidem et al., 2016).

After each random forest model's optimal hyper-parameters have been identified and validated, the models are retrained using only those parameters. The models are then tested using a holdout validation set.

The next machine learning technique used is Extreme Gradient Boosting (XGBoost), this technique will be discussed in the following section.

4.5.3 Extreme Gradient Boosting

Gradient boosting is a machine learning technique that involves developing a model which is an ensemble of weak prediction models. Typically, the models in the ensemble are decision trees. A gradient boosted model is trained in a stage-wise fashion, the model is generalised by minimising a defined loss function throughout training (Friedman et al., 2000).

The final output of a tree ensemble model is calculated by summing the predictions of each tree in the ensemble, varying weights can be attached to the predictions of specific trees. This technique is shown in Figure 4.5.

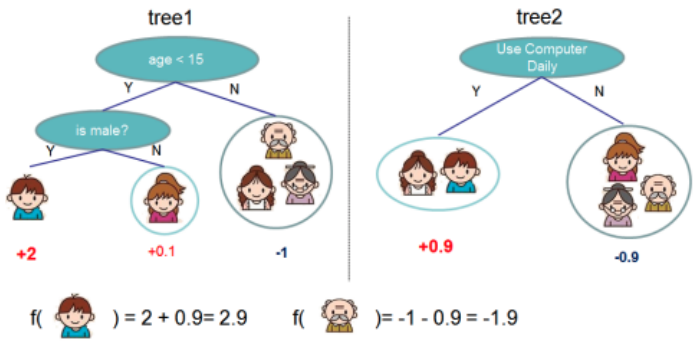

FIGURE 4.5: Example of a Tree Ensemble
(Chen and Guestrin, 2014)

The XGBoost technique was developed by Chen and Guestrin (2014). Its derivation spawned from second order method developed by Friedman et al. (2000). Chen and Guestrin (2014) altered the regularised learning objective developed by Friedman et al. (2000) to improve the performance of XGBoost models. They proposed that for any given dataset, with n observations and m predictor variables, a tree ensemble model can be represented by using K additive functions shown in Equation 4.4.

$$\hat{y}_i = \sum_{k=1}^{k} f_k(x_i), f_k \varepsilon F, \tag{4.4}$$

Where $\hat{y}_i$ is a predicted valued and $F = f(x) = w_{q(x)}(q : R^m \rightarrow T, w \varepsilon R^T)$ is the space of ensemble decision trees. While q represents the structure of each tree and T is the number of leaves (terminal nodes) in the tree. Each f_k corresponds to an independent tree structure (q) and leaf weights (w). The weight on the i-th leaf is represented by w_i (Chen and Guestrin, 2014).

In gradient tree boosting, the set of regularised functions that represent the ensemble of trees are learnt in an additive manner by minimising the loss functions associated to each predicted value. A general version of a loss function used in XGBoost is shown in Equation 4.5. This function is the objective function to be minimised.

$$\tilde{L}^k = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i^{k-1} + f_k(x_i)) + \Omega(f_k) \tag{4.5}$$

Where $\Omega(f_k)$ is a regularisation term penalising model complexity (in this case tree structure). $\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$ is a regularisation term penalising model complexity (in this case tree structure). Note that γ and λ are regularisation parameters. Second-order approximation is used to simplify the optimisation of each loss function. The simplified version can be seen in Equation 4.6.

$$\tilde{L}^k = \sum_{i=1}^{n} \left[g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i) \right] + \Omega(f_k) \tag{4.6}$$

Where g_i and h_i are first and second order gradient statistics on the loss function. If we define an instance of leaf j as $I_j = i|q(x_i) = j$ then we can expand the Ω in Equation 4.6 so that it now appears as below in Equation 4.7.

$$\tilde{L}^k = \sum_{i=1}^{n} [g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{4.7}$$

We can then compute the optimal weight and corresponding value of a fixed structure as shown in Equation 4.8.

$$\tilde{L}^k(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{\sum_{i\varepsilon I_j} g_i}{\sum_{i\varepsilon I_j} h_i + \lambda} + \gamma T \tag{4.8}$$

Figure 4.6 displays how a score is calculated for each tree structure q, based on 4.8. This is done in a greedy manner. The algorithm starts with a single node and iteratively adds branches (Chen and Guestrin, 2014).

Beyond the regularised learning of loss function, the XGBoost algorithm uses two techniques to avoid over-fitting. These techniques are shrinkage and feature sub-sampling (this technique is used in random forests). Shrinkage scales the weights of trees by weights by a factor of η. This reduces the impact a single tree has on the final output of a model (Friedman, 2002).

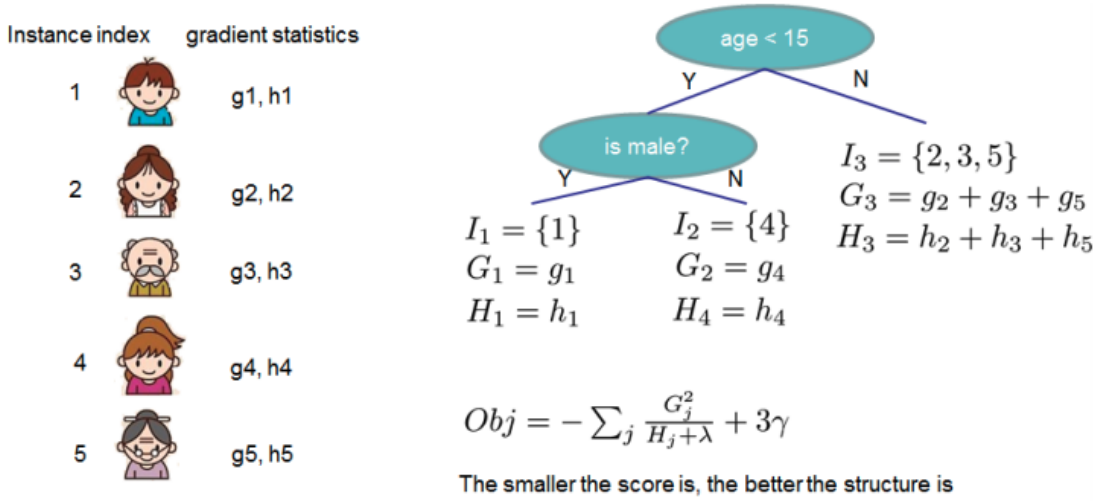

FIGURE 4.6: How Leaves are Scored in XGB
(Chen and Guestrin, 2014)

Yufei et al. (2017) showed that the XGBoost technique can effectively be applied to credit scoring.Their XGBoost model outperformed the logistic regression, random forest, and neural network models they developed for comparative purposes in overall prediction accuracy, area under the curve, and Brier score.

Yufei et al. (2017) used Bayesian optimisation to tune the hyper-parameters of their XGBoost model. In the case of this project grid-search is used to train the hyper-parameters of the XGBoost models developed. The following subsection details the parameters tuned.

Parameter Tuning

The hyper-parameters tuned for each XGBoost model are as follows:

- η, which is the factor by which new weights are shrunk. This prevents over-fitting.
- γ, which is the minimum loss required before a split should be made in a tree. A large Gamma value will result in fewer splits and as a result a more conservative model.
- Maximum depth, the maximum number of splits in a tree.
- Minimum child weight, the minimum sum of weights required in a tree. The larger the minimum child weight the more conservative the model.
- Sub-sample, which is the ratio of the training data that is sub-sampled when a tree is grown.
- Column sample by tree is the proportion of predictor variables used in each tree.

Similarly to hyper-parameter tuning in random forest model, parameter tuning in XGBoost models have a two-fold purpose. It aids in the most accurate models being developed but it also prevents over-fitting from occurring (XGBoost Developers, 2019). Furthermore, the learning objective of the XGBoost models is set to be a binary classification and the evaluation metric used is area under the curve (AUC). AUC is used so that the XGB models generalise well.

After hyper-parameter tuning the models are retrained using the optimal parameters and are then tested using a holdout set. After the XGBoost models are trained and tested, the same process is completed for the 7 neural network models developed.

4.5.4 Neural Networks

Artificial neural networks (NN's) consist of a network of interconnected processors termed neurons. Each neuron receives an input and processes it by passing it through its activation function. Input neurons receive raw input, while other neurons in the network receive a weighted input from previously activated neurons. The learning aspect of a neural network involves adjusting the weights of connections so that the network outputs more desirable results. This process is carried out by back propagation and often involves minimising a loss function so that the predicted values produced by a network are as close as possible to the true values of data points used to train the network (Schmidhuber, 2014).

Neural networks have a variety of architectural structures. Recently recurrent neural network (RNN) architectures (networks that contain cyclic connections between neurons) such as long short-term memory (LSTM) networks have become prominent. These networks are well suited to modelling temporal or sequence behaviour (Sak et al., 2014).

Zekic-Susac et al. (2004) have shown that neural networks can be successfully applied to loan default prediction. Whilst, West (2000) explored the various architectures, activation functions, and loss functions that can be used when training a NN used for loan default prediction.

In the case of loan default prediction: NNs developed throughout this project, no temporal or sequential behaviour is modelled and therefore an RNN architecture is not required. West (2000) displayed that multi-layer perceptrons (MLPs) are effective models when used

for loan default prediction. MLPs are an example of feed-forward neural networks, which means that their neurons are connected in a one-directional fashion (Schmidhuber, 2014). An example of a 3 layer MLP can be seen in Figure 4.7.

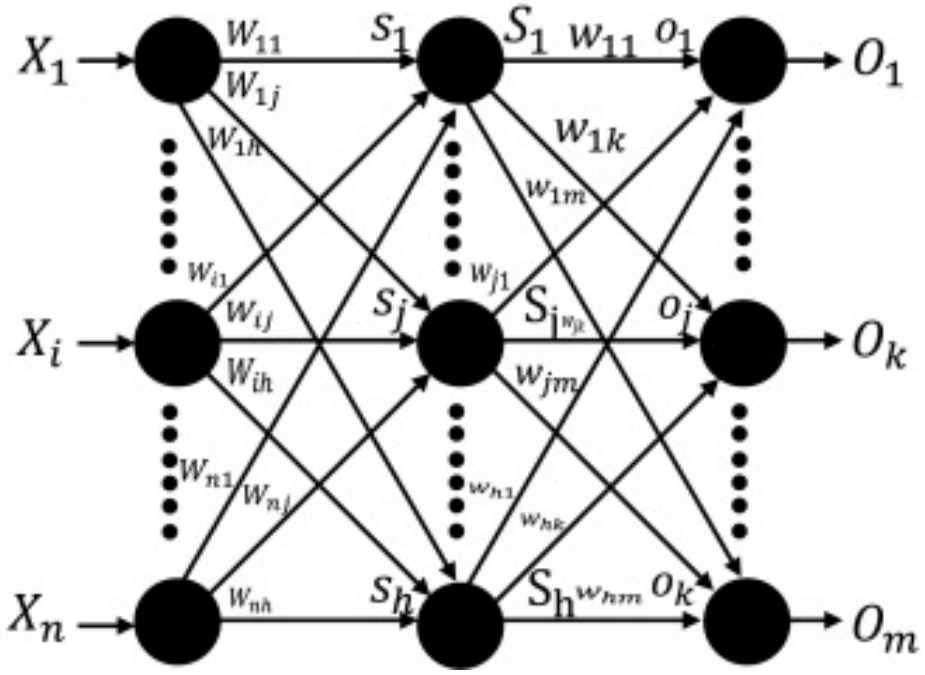

FIGURE 4.7: Architecture of a Multi-Layer Perceptron
(Khishe and Mohammadi, 2019)

The neurons on the left of the Figure represent the input layer, the central neurons form the hidden layer, while the neurons on the right represent the output layer. MLPs can have numerous hidden layers but always have only one input and output layer. The number of neurons in the input layer always matches the number of variables used in the model. (Khishe and Mohammadi, 2019).

Predictions in an MLP model are calculated using Equation 4.9, which illustrates the weighted input in the hidden and output layers.

$$y_j = \sum_{i=1}^{n} W_{ij} X_i - \theta_j, \quad j = 1, 2, ..., h \tag{4.9}$$

Where W_{ij} represents the weight connecting the i-th neuron to j-th-neuron, θ_j represents the bias of the j-th-neuron, and X_i represents the input data to the i-th neuron (Khishe and Mohammadi, 2019).

Each neuron in a neural network has an activation function. Activation functions map the weighted input a neuron receives to the output signal the same neuron produces. Activation functions can be linear or non-linear.

They are often used to limit or smooth the output of a particular neuron. Activation functions commonly used in MLPs are sigmoid, hyperbolic tangent, radial basis, rectified linear unit (ReLU), and softmax (Karlik and Olgac, 2019).

In the case of the NNs developed throughout this project, the input layer of each NN developed contains the same number of neurons as the number optimal features identified during feature selection, while the output layer contains a single neuron with a sigmoid activation function. If the weight inputted into the final neuron is below a certain value then the applicant is deemed to be likely to default. The optimal cut-off value is learnt through back propagation.

In the case of the NNs, the optimisation serves a two-fold purpose. The first is to identify the hyper-parameters which result in the most accurate and generalizable model, while the second is to find the architecture which leads to the most accurate model and generalizable model. In the case of this project, the optimal number of hidden layers and the optimal number of neurons in each hidden layer is determined using grid-search and cross-validation.

Parameter Tuning

The hyper-parameters tuned for each NN model are as follows:

- Learning rate, which determines how quickly a network updates its parameters. The smaller the learning the slower a model trains. A larger learning rate results in a model training faster but can result in weights not converging (Yoo, 2019). This principle is displayed in Figure 4.8.

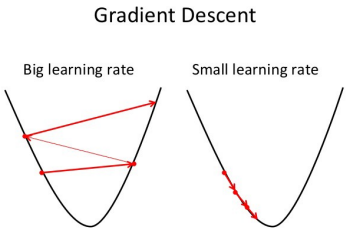

FIGURE 4.8: Varying Learning Rates
(Radhakrishnan, 2017)

- Dropout rate, which is the percentage of neurons that are randomly removed from the network during training. Dropout aims to prevent over-fitting and increase a model's generalizing power (Radhakrishnan, 2017).
- The number of training iterations (epochs). The number of forward and back passes used to train the network (Yoo, 2019).
- The batch size, which is the number of training observations used in a forward/back training pass (Yoo, 2019).
- The architecture of the network: the number of hidden layers in the network and the number of neurons in each hidden layer.

Hyper-parameters that were set but not optimised were the loss function and the activation function. A sigmoid cross-entropy loss function is used in the MLP models. This is not an arbitrary selection, the Python package used to developed the MLP models only supports this loss function for binary classification problems. As previously stated the activation function used a sigmoid function.

After the hyper-parameters of each NN model are identified, each model is tested using a holdout set. The next step in the methodology is to assess the performance of the models developed and test for statistical difference between the performance of the various techniques and datasets used.

4.6 Comparing Feature Combinations and Modelling Techniques

After all twenty eight models are developed the aims of the project need to be addressed.The first is to test if alternative data improves loan default prediction models. The second is to assess if accurate loan default prediction models can be developed using only the alternative features used throughout this project. The third and final question to be answered is which machine learning technique is most suited for loan default prediction.

4.6.1 Comparing Feature Combinations

The first aim involves comparing models that were trained on different datasets. It does not involve comparing different modelling techniques, but rather comparing if adding features of a particular type improved a model. This - like the second aim - can be answered by assessing performance indicators of the models. The training indicators of the models are validated using 5-fold cross validation, while the test indicators for each model are produced from an independent holdout set.

The second aim only considers the models that are trained and tested using only alternative features. Model performance is assessed by considering each model's overall prediction accuracy, their ability to predict loans that were repaid (repaid accuracy), their ability to predict loans that were defaulted (default accuracy), their F_1 score, and their AUC.

F1 score is an accuracy measure for binary classifiers that considers both recall and precision. The closer the F1 score of a model is to 1, the better it is at identifying positive cases as positive cases and negative cases as negative cases (Powers, 2011). In the case of this project, the positive class consists of loans that were not repaid while the negative class consists of loans that were repaid. Recall is the proportion between the number of predicted positive cases and the total number of actual positive cases. Precision is the ratio between the number predicted true positive cases that are actual true cases and the total number of predicted true cases (Powers, 2011). Recall and precision can be visually understood using the binary classification matrix shown in Figure 4.9.

Recall and precision are calculated as shown in Equations 4.10 and 4.11 respectively. The two measures are then combined as shown in Equation 4.12 to calculated f1 score.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

FIGURE 4.9: Binary Classification Matrix
(Shung, 2015)

$$recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{4.10}$$

$$precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4.11}$$

$$f1 = 2 * \frac{\text{precision * recall}}{\text{precision + recall}} \tag{4.12}$$

AUC is the area under the receiver operating characteristic curve, which is a plot of the true positive and true negative rates of a model at various thresholds. The higher the AUC, the better a model is at disguising the two classes of a binary classification problem. An AUC measure lies between 0 and 1 (Powers, 2011).

4.6.2 Comparing Modelling Techniques

The various techniques used throughout this project are compared using McNemar's Chi Square test, which has been shown to be the most powerful test of statistically significant differences between supervised learning models (Dietterich, 1998). This technique was used by West (2000) when comparing various neural network architectures.

The test involves training two algorithms on the same training dataset (each technique was trained on the same dataset for each of the data category combinations). The algorithms are then tested on the same holdout set (again, this was done for each technique for each of the data category combinations). If we call one algorithm A and the other B then the test has a null hypothesis that assumes that the number of observations misclassified by A but not by B (n_{01}) will equal the the number of observations misclassified by B but not by A (n_{10}). The test's statistic is calculated as shown in Equation 4.13 (Dietterich, 1998).

$$\frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{4.13}$$

Under the null hypothesis the statistic follows a Chi Squared distribution with 1 degree of freedom. The p-values calculated from the test statistic shown in Equation 4.13 are used to ascertain how unlikely it is that such big a difference would be if the null hypothesis were true. (Dietterich, 1998).

4.7 Summary of Modelling Techniques

This chapter summarises the 7 data category combinations and the 4 modelling techniques that are used in this project. The feature selection process completed is then detailed, followed by a breakdown of each modelling technique and the hyper-parameters tuned during the training of each model. Finally the chapters details the testing techniques used in this project. The next chapter displays the modelling and testing results, and contains the discussion of the results.

Chapter 5

Results and Discussion

This chapter details the results and findings attained throughout this project. First, the feature selection results for each of the datasets used to train the 28 models are displayed. Then, the performance of each modelling technique is displayed and discussed. This chapter then details if the alternative data features used throughout this project improved model performance and if the alternative features developed could be used to produce accurate loan default prediction models. Finally, the chapter displays the results of the McNemar's Chi Square tests used to compare the models developed.

5.1 Features Selection

For each of the data category combinations shown in Section 4.1, RFE is used to identify the most relevant features, while cross validation is used to validate the optimal number of features.

Figures 5.1, 5.2, and 5.3 are plots containing the number of features selected versus the model accuracy for datasets containing only sociodemographic, credit bureau, and alternative features respectively. Figure 5.4 displays the number of features selected against model accuracy for the dataset containing the features from all 3 data categories.

The figures display large variations in model accuracy based on the number of features used, which highlights the importance of feature selection.

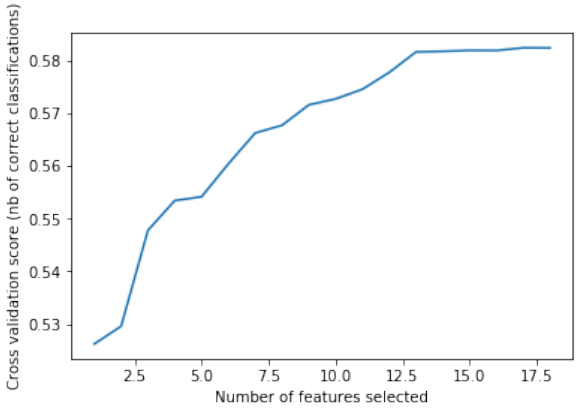

FIGURE 5.1: Sociodemographic Variable Selection

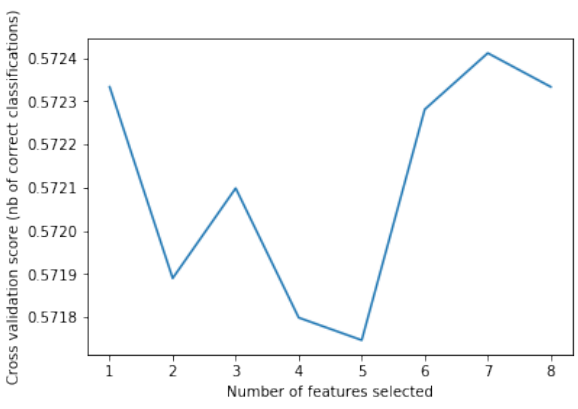

FIGURE 5.2: Credit Bureau Variable Selection

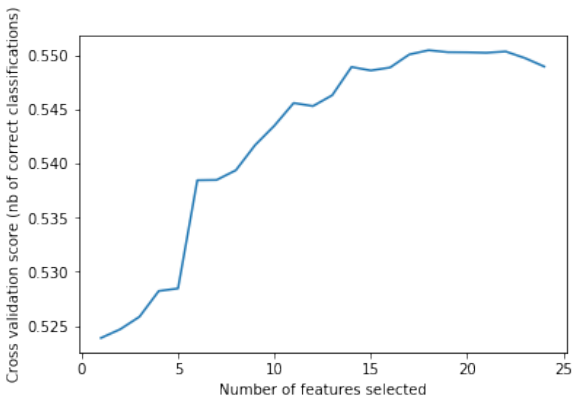

FIGURE 5.3: Alternative Data Variable Selection

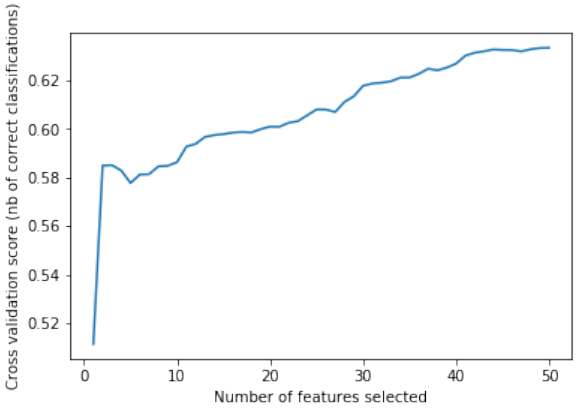

FIGURE 5.4: Variable Selection for the All Datasets

Figure 5.1 displays that when only SD features are used, the best test accuracy is produced when 17 features are used. Figure 5.2 displays that there is a decrease in accuracy when as more CB features are used in the features space, until the number of features moves beyond 5. Figure 5.3 displays that using more alternative features improves model accuracy until the feature space grows larger than 18, after which there is very little improvement in model accuracy or there is actually a decrease in accuracy. Figure 5.4 displays that when using all data categories, model accuracy steadily improves when more features are used in the feature space.

Table 5.1 displays the feature selection results for all 7 data category combinations. The table provides the total number of features contained in each dataset, the number of features selected from the original datasets, and the names of the features that were deemed irrelevant (not selected).

The table highlights a mass drop of features when only CB and ALD features are used. This could indicate dependencies and collinearity between variables in that dataset, however the difference between the lowest and highest validation accuracies is less than 1.

Furthermore, the Table validates the trends displayed in Figures 5.1, 5.2, 5.3, and 5.4. This is particular evident in the ALD row, as it can be seen that a large number of features were deemed irrelevant. It is interesting that many of the ALD features not selected when only using ALD features, are deemed relevant when coupled with SD features.

For each of the data category combinations, the various modelling techniques were trained and tested on the same dataset. The first modelling technique used is logistic regression. The cross-validated training results and the holdout results can be seen for this technique in the following section.

Table 5.1 can be viewed below.

Dataset	Features	Selected Features	Features Not Selected
SD	18	17	application week
CB	8	7	nonperforming loans
ALD	24	18	competitor count, gambling count, app count, successful payments, min successful loan payment, unsuccessful payments, max unsuccessful loan payment, min unsuccessful loan payment
SD and CB	26	24	application time , application week
SD and ALD	42	40	competitor count, gambling count
CB and ALD	32	3	open accounts by date, performing loans, paid loans, nonperforming loans, lost loans, missed payments, competitor count, banking count, news count, gambling count, VPN count, app count, device price, max balance, min balance, credit transactions, max credit, min credit, max debit, min debit, insufficient funds, successful payments, max successful loan payment, min successful loan payment, unsuccessful payments, max unsuccessful loan payment, min unsuccessful loan payment, rejected loans, max loan amount
SD, CB and ALD	50	50	

TABLE 5.1: Feature Selection Results

5.2 Logistic Regression

Tables 5.2 and 5.3 respectively display the training and holdout results of the developed logistic regression models.

The holdout results displayed in Table 5.3 highlight that the logistic regression models improved when the alternative features were added to the sociodemographic and credit bureau datasets. Furthermore, 5.3 displays that the best performing model is trained using features from all three data categories. The model has the highest overall accuracy, repaid accuracy, F1 score, and AUC. The model developed using only credit bureau features and the model developed using the credit bureau and alternative features had high training and holdout default accuracy. However, the same models have low overall accuracy, repaid accuracy, F1 scores, and AUC values.

The logistic regression model developed using only the alternative features does not perform well when classifying loans that were repaid. This can be seen in its low overall and repaid accuracies. The low F1 and AUC scores provide further validation that the model does not accurately predict the outcome of loans were repaid.

The holdout results displayed in Table 5.3 closely align with the logistic regression results produced by Óskarsdóttir et al. (2019).

There is very little discrepancy between the training and holdout model performance indicators for each logistic regression model. This provides a good indication that no over-fitting occurred during training.

Dataset	Accuracy	Repaid Accuracy	Default Accuracy	F1 Score	AUC
SD	0.58	0.62	0.55	0.60	0.58
CB	0.55	0.44	0.65	0.54	0.55
ALD	0.54	0.46	0.62	0.54	0.56
SD and CB	0.60	0.61	0.61	0.60	0.61
SD and ALD	0.62	0.64	0.59	0.61	0.66
CB and ALD	0.58	0.44	0.75	0.59	0.61
SD, CB and ALD	0.63	0.62	0.64	0.63	0.68

TABLE 5.2: Logistic Regression Training Performance

Dataset	Accuracy	Repaid Accuracy	Default Accuracy	F1 Score	AUC
SD	0.61	0.62	0.55	0.61	0.59
CB	0.49	0.44	0.66	0.49	0.56
ALD	0.50	0.47	0.64	0.51	0.57
SD and CB	0.61	0.61	0.62	0.61	0.62
SD and ALD	0.63	0.64	0.59	0.63	0.66
CB and ALD	0.51	0.43	0.73	0.53	0.62
SD, CB and ALD	0.64	0.64	0.63	0.63	0.68

TABLE 5.3: Logistic Regression Holdout Performance

5.3 Random Forest

Tables 5.4 and 5.5 respectively display the cross validation and holdout results of random forests models developed throughout this project.

It can be seen that the random forest models produce better training performance indicators than the logistic regression models - shown in Table 5.2. The holdout default accuarcies of the random forest models are, at times, significantly lower than the holdout default accuarcies of the logistic regression models - shown in Table 5.3. This is due to a combination of effects of SMOTE and that Random forest models, like other ensemble techniques, make use

of classical sub-sampling methods (Feng et al., 2019).

SMOTE is only applied to training datasets of the developed models, meaning that the synthetic obvserabtaions are only generated from the data points in the training data and not from data points on the test data. Variation in the minority class observations leads to differences between the observations in the training and test data. Therefore the RF models struggle to identify minority class observations in the holdout sets (Shen et al., 2019).

The sub-sampling performed when the RF models are trained can result in a common data distribution shared by all base-classifiers. This can result in the loss of important information which in turn results in the trained models poorly identifying the minority class observations contained within the holdout sets (Feng et al., 2019).

Dataset	Accuracy	Repaid Accuracy	Default Accuracy	F1 Score	AUC
SD	0.74	0.69	0.78	0.73	0.74
CB	0.59	0.53	0.74	0.63	0.68
ALD	0.73	0.74	0.72	0.73	0.73
SD and CB	0.77	0.80	0.75	0.78	0.77
SD and ALD	0.76	0.75	0.77	0.76	0.76
CB and ALD	0.65	0.66	0.65	0.65	0.65
SD, CB and ALD	0.80	0.77	0.82	0.80	0.80

TABLE 5.4: Random Forest Training Performance

Dataset	Accuracy	Repaid Accuracy	Default Accuracy	F1 Score	AUC
SD	0.68	0.71	0.44	0.65	0.56
CB	0.56	0.52	0.68	0.56	0.64
ALD	0.70	0.74	0.53	0.69	0.63
SD and CB	0.72	0.81	0.39	0.71	0.59
SD and ALD	0.72	0.77	0.48	0.69	0.66
CB and ALD	0.63	0.65	0.52	0.63	0.63
SD, CB and ALD	0.74	0.79	0.59	0.73	0.69

TABLE 5.5: Random Forest Holdout Performance

The most accurate RF model is trained using features from all three data categories. Furthermore, when alternative features were added to both the datasets containing the sociodemographic and the credit bureau features the performance of the random forest models improve. This can be seen in accuracy, repaid accuracy, F1 score, and AUC values displayed in Table 5.5. Similarly to the logistic regression model containing only credit bureau features, the respective random forest model performs better than other random forest models when identifying loans that were not repaid.

5.4 Extreme Gradient Boosting

The third modelling technique used is XGBoost. Tables 5.6 and 5.7 display the training and holdout results of the 7 XGBoost models developed throughout this project.

Table 5.7 displays that in general, the XGBoost models outperform their respective logistic regression and random forest models. It can be seen from both XGBoost tables that the test default accuracy is often significantly lower than the training default accuracy. This trend is seen in the random forest models and occurs in the XGBoost models for the same reasons that it occurs in the random forest models (the effects of SMOTE and sub-sampling).

Table 5.7 shows that the alternative features improve the overall accuracy, repaid accuracy, F1 score and AUC of the XGBoost models when added to data containing sociodemographic and credit bureau features.

Dataset	Accuracy	Repaid Accuracy	Default Accuracy	F1 Score	AUC
SD	0.83	0.82	0.84	0.83	0.82
CB	0.65	0.56	0.75	0.63	0.72
ALD	0.80	0.79	0.80	0.79	0.79
SD and CB	0.83	0.81	0.84	0.83	0.83
SD and ALD	0.85	0.81	0.89	0.85	0.85
CB and ALD	0.71	0.71	0.72	0.71	0.71
SD, CB and ALD	0.89	0.88	0.91	0.91	0.91

TABLE 5.6: XGBoost Training Performance

Dataset	Accuracy	Repaid Accuracy	Default Accuracy	F1 Score	AUC
SD	0.71	0.81	0.38	0.72	0.59
CB	0.59	0.55	0.64	0.57	0.63
ALD	0.74	0.79	0.55	0.74	0.68
SD and CB	0.74	0.81	0.46	0.72	0.63
SD and ALD	0.78	0.82	0.62	0.76	0.71
CB and ALD	0.68	0.72	0.48	0.66	0.62
SD, CB and ALD	0.81	0.86	0.68	0.81	0.75

TABLE 5.7: XGBoost Holdout Performance

Similarly to both the logistic regression and random forest techniques, the best performing XGBoost model is trained on the dataset containing features from all three data categories.

The XGBoost model trained using only alternative features had an overall test accuracy of 74%, a repaid accuracy of 78%, an f1 score of 0.74, and an AUC of 0.86. These would all indicate the model could be used to relatively accurately predict the outcome of a loan. However, the model only correctly predicted 55% of the applicants in the test set who defaulted on their loans.

The final technique explored during this project is Neural Networks. The following section displays the features of the multi-layer perceptron networks developed.

5.5 Neural Networks

The training and holdout results of the 7 MLP models developed throughout this projected can be seen in Tables 5.8 and 5.9.

The MLP models showed very similar trends to the other modelling techniques. The best performing model based upon overall accuracy, AUC and F1 score is the model trained using features from all three data categories and when the alternative data features are added to the sociodemographic and credit bureau datasets the default accuracy of the models improved.

Dataset	Accuracy	Repaid Accuracy	Default Accuracy	F1 Score	AUC
SD	0.60	0.62	0.59	0.61	0.61
CB	0.58	0.58	0.59	0.57	0.61
ALD	0.51	0.54	0.47	0.50	0.52
SD and CB	0.63	0.55	0.70	0.62	0.63
SD and ALD	0.62	0.61	0.62	0.62	0.62
CB and ALD	0.59	0.42	0.75	0.59	0.62
SD, CB and ALD	0.70	0.69	0.71	0.70	0.69

TABLE 5.8: Neural Network Training Performance

Dataset	Accuracy	Repaid Accuracy	Default Accuracy	F1 Score	AUC
SD	0.60	0.61	0.54	0.60	0.58
CB	0.58	0.58	0.58	0.59	0.59
ALD	0.53	0.54	0.52	0.53	0.51
SD and CB	0.58	0.55	0.66	0.57	0.61
SD and ALD	0.63	0.63	0.62	0.62	0.63
CB and ALD	0.58	0.53	0.47	0.58	0.62
SD, CB and ALD	0.67	0.68	0.66	0.67	0.68

TABLE 5.9: Neural Network Holdout Performance

The MLP models developed throughout this project showed similar patterns to those developed by West (2000). The models generally had a higher repaid accuracy than a default accuracy. However, the overall accuracies achieved by the models developed by West (2000) were significantly higher than those developed throughout this project.

5.6 Best Performing Model

The best performing model developed throughout this project is the XGBoost model trained on all 3 datasets. The optimal hyper-parameters found for the model, its ROC curves, and the importance of its features are displayed in the following sub-sections.

5.6.1 Hyper-Parameters

The optimal hyper-parameters were identified using a grid search. The definition of each hyper-parameter tuned is displayed in Section 4.5.3. The optimal parameters for the best performing model are as follows:

- eta: 0.05.
- $gamma$: 0.5.
- Maximum depth: 25.
- Minimum child weight:5.
- Sub-sample: 0.8.
- Column sample by tree: 0.8.

The maximum depth of each tree trained was limited to 50, this was to avoid over-fitting. The optimal learning rate (eta), sub-sample ratio, and column sample by tree ratio were the maximum values tested for their respective parameters.

5.6.2 ROC Curves

Figures 5.5 and 5.6 display the training and test ROC curves for the best performing model. The decrease in default accuracy from training to testing is shown in the figures. The figures display that the testing default accuracy of the model is significantly lower than the training default accuracy. As mentioned in Section 5.4, this is caused by the effects of SMOTE and sub-sampling.

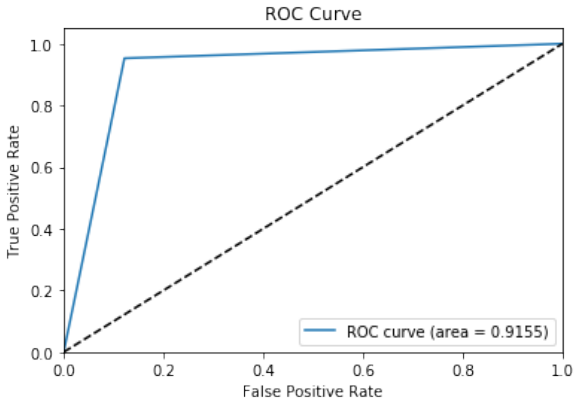

FIGURE 5.5: XGBoost Training ROC

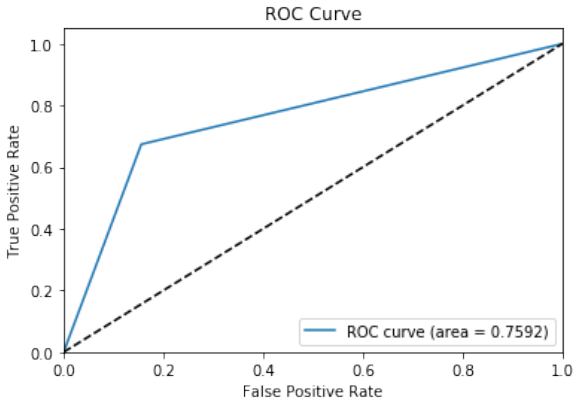

FIGURE 5.6: XGBoost Holdout ROC

Feature Impact

The SHapley Additive exPlanation (SHAP) values for the ten most important features of the best performing XGBoost model can be seen in Figure 5.7. SHAP values show how much each feature contributes, either positively or negatively, to the target variable. Each dot shown in the plot represents a training observation. The plot demonstrates feature importance, the impact of an observation on the final prediction, the distribution of each feature, and the correlation between features and the final prediction (Lundberg and Lee, 2017).

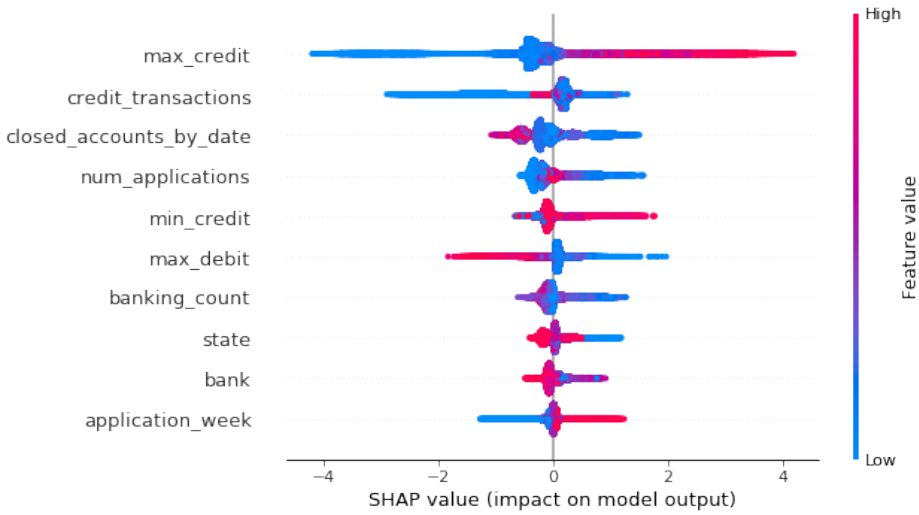

FIGURE 5.7: SHAP Values of Best Performing XGBoost Model

The features are ranked in descending order of importance in Figure 5.7, meaning that the maximum credit transaction extracted from loan applicants' sms messages - an alternative feature - is the most important feature used in the model. Furthermore, we can see the impact the maximum credit feature has on the model. The lower the maximum credit value, the more the prediction is pushed towards 0 (a repaid prediction). Based on the distribution of the maximum credit feature - shown in Figure 5.7 - we can see the majority of maximum credit transactions have a small 0 SHAP value impact on the best performing model.

The final step in the testing process was to statically compare the performance of the models across each of the 7 datasets.

5.7 Model Comparison

The performance indicators displayed in the tables that preceded this section, indicate that the performance of the models improved when alternative data features were added to sociodemographic and credit bureau datasets (the only exception were the NN models). Furthermore, the optimal model developed across all 4 techniques used sociodemographic, credit bureau, and alternative data features. The random forest and XGBoost models developed using only alternative features displayed good overall accuracy, high repaid accuracy, good F1 scores, and good AUC values. However, both models display a relatively low default accuracy. The logistic regression and MLP models developed using only alternative features display a worse overall performance.

The final test performed on the models is McNemar's Chi Square test, which tests if there is a significant difference between two models trained on the same sample. The test is used to compare each technique against all other techniques for each of the 7 data category combinations, which leads to a total of 42 combinations.

The null hypothesis of the McNemar's Chi Square test states that the models compared are not different, and as a result the models will produce the same number of false and true positives and negatives. Of the 42 combinations tested, the null hypothesis was rejected - at a significance level of 0.01 - only 4 times. The null hypothesis was rejected for the following pairs: LR and RF using SD data, LR and MLP using SD data, RF and MLP using SD data, and RF and XGB using a combination of SD and CB data. Other than these pairs all model pairs were found to be significantly different.

For each of the 7 datasets used throughout this project, Table 5.10 displays the modelling technique that attained the highest score - and the score itself - for each of the 5 metrics used to evaluate the models. We can see from Table 5.10 that the XGBoost technique consistently outperforms the other 3 techniques across the majority of the evaluation metrics for all of the datasets other than dataset that uses only credit bureau variables.

Dataset	Accuracy	Repaid Acc	Default Acc	F1 Score	AUC
SD	XGB (0.71)	XGB (0.81)	LR (0.55)	XGB (0.72)	XGB (0.59)
CB	XGB (0.59)	NN (0.58)	RF (0.68)	NN (0.59)	RF (0.64)
ALD	XGB (0.74)	XGB (0.79)	LR (0.64)	XGB (0.74)	XGB (0.68)
SD and CB	XGB (0.74)	XGB/RF (0.81)	NN (0.66)	XGB (0.72)	NN (0.63)
SD, ALD	XGB (0.78)	XGB (0.82)	XGB/NN (0.62)	XGB (0.72)	XGB (0.63)
CB, ALD	XGB (0.68)	XGB (0.72)	LR (0.73)	XGB (0.66)	RF (0.63)
SD, CB, ALD	XGB (0.81)	XGB (0.86)	XGB (0.68)	XGB (0.81)	RF (0.75)

TABLE 5.10: Best Performing Modelling Technique

5.8 Summary of Modelling Results

This chapters displays the cross-validated training results and the holdout results of each of the models developed throughout this project. The alternative features developed during this project are found to improve the performance of loan default prediction models when added to sociodemographic and and credit bureau data: for the logistic regression, random forest, XGBoost, and neural network techniques. For all 4 techniques used the most accurate model is trained on a dataset contain sociodemographic, credit bureau, and alternative features. Models developed only using the alternative features did not perform well when predicting loans that were not repaid. The best performing model developed using only alternative features used XGBoosting, however even that model has a relatively low default accuracy. This indicates that the alternative features used throughout this project cannot solely be used to develop a loan prediction model. Finally, the models were proven to be significantly different using McNemar's Chi Squared Test with 1 degree of freedom. The consistently best performing technique used for loan default prediction in this project is XGBoosting.

The final chapter of this project summarises the conclusions and findings of the research.

Chapter 6

Conclusions and Recommendations

This chapter summarises the findings of the research aims of this masters dissertation displayed in Section 1.3. This chapter concludes by briefly describing the implications of the research conducted in this m.d. and the possible future research opportunities that could extend from the project.

6.1 Research Questions

The research aims of this project are are shown in Section 1.3, but for ease they are listed below:

- Assess if augmenting sociodemographic and credit bureau data with the alternative features used in this project improves the overall performance of loan default prediction models.
- Determine if the alternative features used throughout this dissertation can be used to train accurate loan default prediction models.
- Identify the most appropriate technique for developing loan default prediction models out of logistic regression, random forests, extreme gradient boosting, and artificial neural networks.

The first aim is answered by comparing the five holdout performance indicators of the models trained using the alternative features in conjunction with sociodemographic features, credit bureau features, and both the sociodemographic and credit bureau features against the performance indicators of the models trained using only sociodemographic, only credit bureau, and sociodemographic and credit bureau respectively for each of the 4 modelling techniques.

The performance indicators of the logistic regression, random forest, XGBoost, and multilayer perceptron models developed using only sociodemographic or only credit bureau features improve when the datasets are augmented with alternative features. Therefore, all models using sociodemographic and only credit bureau features improve when the datasets used to train them are augmented with alternative credit bureau features.

Furthermore, the best performing model for each respective modelling techniques used all three data category combinations. Additionally, 6 of the 10 most important features of the best overall performing model are alternative features.

The second aim is addressed by assessing the performance indicators of all 4 models trained using only alternatives features. The indicators can be seen in the holdout results displayed in Chapter 5. The most accurate model trained using only alternative features is an XGBoost model. The model has an overall accuracy of 0.74, a repaid accuracy of 0.81, an F1 score of 0.72, and an AUC measure of 0.68. These indicate that the model accurately predicts whether a loan will be repaid. However, the default accuracy of the model is 0.55. Therefore, the model does not accurately detect when a loan is likely to not be repaid. This is costly within the lending sector.

The third and final aim is assessed using a combination of model performance indicators and McNemar's Chi Squared test. The model performance indicators are used to infer the technique with the best performing indicators, while McNemar's Chi Squared test is used to determine if the models are significantly different.

The most suitable modelling technique - explored within this project - for loan default prediction is found to be XGBoost. This technique consistently produces the best performing model across all 7 data category combinations. The XGBoost models were proven to be significantly different from models of the other 3 techniques using McNemar's Chi Squared test.

6.2 Implications of This Research

The research conducted throughout this projects answers the three research aims stated in Section 1.3. However, there are a number of ways in which the research into each of the aims could extended.

The alternative features used throughout the project did not include the call log or contact data contained on each loan applicant's device. The research completed by Óskarsdóttir et al. (2019) showed that features developed from contact and call log data improved loan default prediction. Features similar to those used by Óskarsdóttir et al. (2019) could be added to the features used throughout this project with the aim of further improving loan default prediction.

Only multi-layer perceptron models were used throughout this project. West (2000) displayed other NN architectures that were found to be suitable for loan default prediction. These NN architectures, as well as others, could be explored.

The hyper-parameters of every model trained and tested during this project are tuned using a grid-search. This method of parameter tuning requires manual input and does not necessarily lead to the most optimal models (West, 2000). The impact of genetic algorithms - such as the one explored by Shen et al. (2019) - on loan default prediction performance could be explored.

Recursive feature elimination is the only feature selection method considered throughout this project. Furthermore, RFE is only used in conjunction with logistic regression to perform feature selection. Other selection methods and other base model types could be explored.

Beyond how the research methods used for this project could be strengthened, there are certain aspects of loan default prediction not explored by this minor dissertation. Firstly, only

first time loan applicants were considered for this project. The performance of each modelling technique on repeat lenders is not explored.

This project focuses on improving prediction in terms of overall accuracy, repaid accuracy, default accuracy, f1 score, and AUC. The financial implications of the loan default prediction models are not considered in the scope of the project.

The regulatory implications of the various data categories and modelling techniques used throughout this project are not explored.

Finally, the impact of improving loan default prediction on financial inclusion was not measured during this project.

Bibliography

Aho, Alfred (1990). "Algorithms for finding patterns in strings. Handbook of Theoretical Computer Science, volume A: Algorithms and Complexity". In: *The MIT Press*, 255–300.

Albon, Chris (2018). *Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning*. O'Reilly, pp. 262–263.

Bahl, Aileen et al. (2019). "Recursive feature elimination in random forest classification supports nanomaterial grouping". In: *NanoImpact* 15.

Bellotti, Tony and Jonathan Crook (2009). "Support Vector Machines for Credit Scoring and Discovery of Significant Features". In: *Expert Systems with Applications* 36.

Bergstra, James and Yoshua Bengio (2013). "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 12, pp. 281–305.

Bilogur, Aleksey (2018). "Missingno: a missing data visualization suite". In: *Journal of Open Source Software* 3, p. 547.

Blanco, Antonio et al. (2013). "Credit scoring models for the microfinance industry using neural networks: Evidence from Peru". In: *Expert Systems with Applications* 40, pp. 356–364.

Breiman, Leo (1996). "Bagging Predictors". In: *Machine Learning* 24, 123–140.

Breiman, LEO (2001). "Credit scorecard based on logistic regression with random coefficients". In: *Machine Learning* 45, pp. 5–32.

Cheadle, Chris et al. (2003). "Analysis of Microarray Data Using Z Score Transformation". In: *The Journal of Molecular Diagnostics* 5, pp. 73–81.

Chen, Tianqi and Carlos Guestrin (2014). "XGBoost: A Scalable Tree Boosting System". In: *University of Washington*.

Christl, J and K Pribil (2005). "Credit Approval Process and Credit Risk Management". In: *Guidelines on Credit Risk Management*.

Cowan, Glen et al. (2015). "Higgs Boson Discovery with Boosted Trees". In: *Journal of Machine Learning Research* 42, pp. 69–80.

Cox, D (1958). "The Regression Analysis of Binary Sequences". In: *Journal of the Royal Statistical Society* 20, pp. 215–242.

DeLong, Elizabeth, David DeLong, and Daniel Clarke-Pearson (1988). "Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach". In: *Journal of Cleaner Production* 44, pp. 837–845.

Deng, Zhenyun et al. (2016). "Efficient kNN classification algorithm for big data". In: *Neurocomputing* 195, pp. 143–148.

Dietterich, T (1998). "Approximate statistical tests for comparing supervised classifications learning algorithms". In: *Neural Computation* 10, pp. 1895–1923.

Dong, G, K Lai, and J Yen (2010). "Credit scorecard based on logistic regression with random coefficients". In: *Procedia Computer Science* 1, pp. 2463–2468.

Durand, D (1941). "Risk Elements in Consumer Instalment Financing, Studies in Consumer Instalment Financing". In: *National Bureau of Economic Research*.

Feng, Wei et al. (2019). "New margin-based subsampling iterative technique in modified random forests for classification". In: *Knowledge-Based Systems* 182.

Florez-Lopez, R (2010). "Effects of missing data in credit risk scoring. A comparative analysis of methods to achieve robustness in the absence of sufficient data". In: *Journal of the Operational Research Society* 61, 486–501.

Freund, Yoav and Robert Schapire (1996). "A desicion-theoretic generalization of on-line learning and an application to boosting". In: *Journal of Computer and System Sciences* 55, pp. 119–139.

Friedman, J (2002). "Stochastic gradient boosting". In: *Computational Statistics Data Analysis* 38, 367–378.

Friedman, J et al. (2000). "Additive logistic regression: a statistical view of boosting". In: *Annals of Statistics* 28.

Galar, Mikel et al. (2012). "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 42, pp. 463–484.

Guyon, Isabelle and Andre Elisseeff (2003). "An Introduction to Variable and Feature Selection". In: *Machine Learning Research* 3, pp. 1157–1182.

Guégana, D and B Hassan (2018). "Regulatory learning: How to supervise machine learning models? An application to credit scoring". In: *The Journal of Finance and Data Science* 4, pp. 157–171.

Hand, D and W Henley (1997). "Statistical Classification Methods in Consumer Credit Scoring: a Review". In: *Journal of the Royal Statistical Society* 160, pp. 523–541.

Hastie, T, Robert Tibshirani, and Jerome Friedman (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, pp. 119–128.

Horton, Nicholas and Ken Kleinman (2007). "Much Ado About Nothing: A Comparison of Missing Data Methods and Software to Fit Incomplete Data Regression Models". In: *JThe American Statistician* 61, 78–90.

Howarde, Anton (1994). "Elementary Linear Algebra". In: *John Wiley Sons* 7, pp. 170–171.

James, Gareth et al. (2013). *An Introduction to Statistical Learning*. Springer, pp. 357–358.

Jensen, H (1992). "Using Neural Networks for Credit Scoring". In: *Managerial Finance* 18, pp. 15–26.

Jones, Stewart and David Hensher, eds. (2008). *Advances in Credit Risk Modelling and Corporate Bankruptcy Prediction*. Cambridge University Press.

Karlik, Bekir and Vehbi Olgac (2019). "Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches". In: *International Journal of Artificial Intelligence And Expert Systems* 1, pp. 112–122.

Khaidem, Luckyson, Snehanshu Saha, and Sudeepa Dey (2016). "Predicting the direction of stock market prices using random forest". In: *Applied Mathematical Finance*.

Khan, Samir et al. (2019). "Unsupervised anomaly detection in unmanned aerial vehicles". In: *Applied Soft Computing* 83.

Khishe, Mohammad and Hassan Mohammadi (2019). "Passive sonar target classification using multi-layer perceptron trained by salp swarm algorithm". In: *Ocean Engineering* 181, pp. 98–108.

Lewis, E M (1992). *An introduction to credit scoring*. Athena Press.

Li, X and Y Zhong (2012). "An Overview of Personal Credit Scoring: Techniques and Future Work". In: *International Journal of Intelligence Science* 2, pp. 181–189.

Ling, H et al. (2019). "Combination of Support Vector Machine and K-Fold cross validation to predict compressive strength of concrete in marine environment". In: *Construction and Buidling Materials* 206, pp. 355–363.

Liu, Fei, Kai Ting, and Zhi-Hua Zhou (2008). "Isolation Forest". In: *IEEE International Conference on Data Mining* 18.

Lundberg, Scott and Su-In Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Paul G. Allen School of Computer Science University of Washington*.

Luo, Cuicui, Desheng Wu, and Dexiang Wu (2017). "A Deep Learning Approach for Credit Scoring Using Credit Default Swap". In: *Engineering Applications of Artificial Intelligence* 65.

Marquez, J (2008). *An Introduction to Credit Scoring For mall and Medium Size Enterprises*. https://siteresources.worldbank.org/EXTLACOFFICEOFCE/Resources/870892-1206537144004/MarquezIntroductionCreditScoring.pdf.

Obermeyer, Ziad and Ezekiel Emanuel (2016). "Predicting the Future — Big Data, Machine Learning, and Clinical Medicine". In: *The New England Journal of Medicine* 375, 1216–1219.

Pant, Ayush (2018). *Introduction to Logistic Regression*. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148.

Pointon, H Abdou J (2011). "Credit Scoring, Statistical Techniques and Evaluation Criteria: A Review of the Literature". In: *Intelligent Systems in Accounting, Finance Management* 18, pp. 59–88.

Powers, D (2011). "EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS CORRELATION". In: *Journal of Machine Learning Technologies* 2, pp. 37–63.

Probst, Philipp, Marvin Wright, and Anne-Laure Boulesteix (2019). "Hyperparameters and Tuning Strategies for Random Forest". In: *Wiley Interdisciplinary Reviews*.

Radhakrishnan, Pranoy (2017). *What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network?* https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a.

Raghuwanshi, Bhagat and Sanyam Shukla (2019). "Generalized class-specific kernelized extreme learning machine for multiclass imbalanced learning". In: *Expert Systems with Applications* 121, pp. 244–255.

Sak, Hasim, Andrew Senior, and Francoise Beaufays (2014). "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling". In: *International Speech Communication Association*.

Schmidhuber, Jürgen (2014). "Deep learning in neural networks: An overview". In: *Neural Networks* 61, 85–117.

Schober, Patrick, Christa Boer, and Lothar Schwarte (2018). "Combination of Support Vector Machine and K-Fold cross validation to predict compressive strength of concrete in marine environment". In: *Anesthesia Analgesia* 126, pp. 1763–1768.

Serrano-Cinca, Carlos, Begona Gutierrez-Nieto, and Nydia Reyes (2016). "Credit scoring models for the microfinance industry using neural networks: Evidence from Peru". In: *Journal of Cleaner Production* 112, pp. 3504–3513.

Shen, Feng et al. (2019). "A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation". In: *Statistical Mechanics and its Applications* 526.

Shung, Koo Ping (2015). *Accuracy, Precision, Recall or F1?* https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9.

Siaw, Anthony et al. (2014). "Pan Empirical Analysis of the Loan Default Rate of Microfinance Institutions". In: *European Journal of Business and Management* 6, pp. 2222–2839.

Siddiqi, Naeem (2006). "Credit Risk Scorecards Developing and Implementing Intelligent redit Scoring". In: *SAS Institute*, 80–81.

Soley-Bori, Marina (2013). "Dealing with missing data: Key assumptions and methods for applied analysis". In: *Boston University School of Public Health*.

The World Bank (2018). *Financial Inclusion*. http://www.worldbank.org/en/topic/financialinclusion/overview.

Thomas, L (2000). "A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers". In: *International Journal of Forecasting* 2, pp. 149–172.

Thomas, L, D Edelman, and J Crook (1984). *Leo Breiman and Jerome Friedman and Charles Stone and R Olshen*. Taylor Francis.

Thomas, L, J Ho, and W Scherer (2001). "Time will tell: behavioural scoring and the dynamics of consumer credit assessment". In: *IMA Journal of Management Mathematics* 12, 89–103.

Thomas, L, D Edelman, and J Crook (2004). *Readings in Credit Scoring: Foundations, Developments and Aims*. Oxford University Press.

Troyanskaya, Olga et al. (2001). "Missing value estimation methods for DNA microarrays". In: *Stanford Medical Informatics* 17, 520–525.

Waddell, P and G Boeing (2016). "New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings". In: *Journal of Planning Education and Research*.

Wang, Gang et al. (2012). "Two credit scoring models based on dual strategy ensemble trees". In: *Knowledge-Based Systems* 26, 61–68.

West, David (2000). "Neural Network Credit Scoring Models". In: *Computers Operations Research* 27, pp. 1131–1152.

Wiginton, J (1980). "A Note on the Comparison of Logit and Discriminant Models of Consumer Credit Behavior". In: *Journal of Financial and Quantitative Analysis* 15, pp. 757–770.

XGBoost Developers (2019). *XGBoost Parameters*. https://xgboost.readthedocs.io/en/latest/parameter.html.

Xie, X, U Krewer, and R Schenkendorf (2018). "Robust Optimization of Dynamical Systems with Correlated Random Variables using the Point Estimate Method". In: *IFAC-Papers* 51, pp. 427–432.

Yan, Ke and David Zhang (2015). "Feature selection and analysis on correlated gas sensor data with recursive feature elimination". In: *Sensors and Actuators B: Chemical* 212, pp. 353–363.

Yoo, Youngjun (2019). "Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches". In: *Knowledge-Based Systems* 178, pp. 74–83.

Yufei, Xia et al. (2017). "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring". In: *Expert Systems with Applications* 78, pp. 225–241.

Zekic-Susac, M, N Sarlija, and M Bensic (2004). "Small business credit scoring: a comparison of logistic regression, neural network, and decision tree models". In: *Information Technology Interfaces* 26.

Zhao, Z et al. (2015). "Investigation and improvement of multi-layer perceptron neural networks for credit scoring". In: *Expert Systems with Applications* 42, pp. 3508–3516.

Óskarsdóttir, M et al. (2019). "The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics". In: *Applied Soft Computing* 74, pp. 26–39.

Appendix A

Variable Definitions

Table A.1 displays all variables used throughout this minor dissertation. The Table further displays a brief description of each variable, each variable's data type, and data category to which the feature belonged.

Variable	Description	Data Type	Category
Age	Age of applicant	Numeric (Discrete)	Sociodemographic
App Count	Number of applications on th applicant's cellular device	Numeric (Discrete)	Alternative
Application Time	The time of day the loan application was made	Categorical	Alternative
Application Week	The week within the month the loan application was made	Categorical	Alternative
Bank	The stated bank with which the applicant holds an account	Categorical	Sociodemographic
Banking Count	The number of financial applications on the clients cellular device	Numeric (Discrete)	Alternative
Banks Contacted	The number of banks that sent an SMS messages to the applicant	Numeric (Discrete)	Alternative
Closed Accounts	The number of closed loan accounts the applicant has registered with the credit bureaus	Numeric (Discrete)	Credit Bureau
Competitor Count	The number of competing micro-finance companies that contacted the applicant	Numeric (Discrete)	Alternative
Credit Transactions	The number of credit transactions extracted from SMS sent to the applicant from banks	Numeric (Discrete)	Alternative
Debit Transactions	The number of debit transactions extracted from SMS sent to the applicant from banks	Numeric (Discrete)	Alternative
Defaulted	A flag indicating whether or not a loan was repaid within 30 days of the repayment date	Numeric (Discrete)	Target
Device Brand	The brand of the cellular device the applicant used to when applying foor their loan	Numeric (Discrete)	Alternative

Device Price	The web-scraped price of the cellular device the applicant used when applying for their loan	Numeric (Discrete)	Alternative
Employment Status	The employment status of the applicant	Categorical	Sociodemographic
Gambling Count	The number of gambling applications on the applicant's cellular device	Numeric (Discrete)	Alternative
Gender	The sex of the applicant	Categorical	Sociodemographic
Highest Education	The highest level of education achieved by the applicant	Categorical	Sociodemographic
Income	The stated monthly income of the applicant in Naira	Numeric (Discrete)	Sociodemographic
Loan Purpose	The stated reason the applicant gave for applying for the loan	Categorical	Sociodemographic
Lost Loans	The number of loan accounts deemed lost/unpaid the applicant has registered with the credit bureaus	Numeric (Discrete)	Credit Bureau
Marital Status	The marital status of the applicant	Categorical	Sociodemographic
Max Balance	The maximum balance extracted from the applicant's bank SMS	Numeric (Discrete)	Alternative
Max Credit	The maximum credit transaction extracted from the applicant's bank SMS	Numeric (Discrete)	Alternative
Max Debit	The maximum debit transaction extracted from the applicant's bank SMS	Numeric (Discrete)	Alternative
Max Loan Amount	The maximum loan amount extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
Max Successful Loan Payment	The maximum successful loan repayment amount extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
Max Unsuccessful Loan Payment	The maximum unsuccessful loan repayment amount extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
Min Balance	The minimum balance extracted from the applicant's bank SMS messages	Numeric (Discrete)	Alternative

Min Credit	The minimum credit transaction extracted from the applicant's bank SMS messages	Numeric (Discrete)	Alternative
Min Debit	The minimum debit transaction extracted from the applicant's bank SMS messages	Numeric (Discrete)	Alternative
Min Loan Amount	The minimum loan amount extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
Min Successful Loan Payment	The minimum successful loan repayment amount extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
Min Unsuccessful Loan Payment	The minimum unsuccessful loan repayment amount extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
Missed Payments	The number of instalments missed by the applicant on their loans registered with the credit bureaus	Numeric (Discrete)	Credit Bureau
News Count	The number of news related applications cellular device	Numeric (Discrete)	Alternative
Non-performing Loans	The number of active loans where the applicant has missed an instalment and the loan was registered with the credit bureaus	Numeric (Discrete)	Credit Bureau
Num Applications	The number of rejected applications the applicant had with the micro-finance company prior to their applicant under consideration	Numeric (Discrete)	Sociodemographic
Num Applications	The number of rejected applications the applicant had with the micro-finance company prior to their applicant under consideration	Numeric (Discrete)	Sociodemographic
Paid Loans	The number of fully repaid loans the applicant has registered with the credit bureaus	Numeric (Discrete)	Credit Bureau
Performing Loans	The number of active loans where the applicant has not missed an instalment and loan was registered with the credit bureaus	Numeric (Discrete)	Credit Bureau

Property Status	The current ownership status of the property where the applicant resides	Categorical	Sociodemographic
Rejected Loans	The number of rejected loan applications extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
Sector	The sector the applicant's occupation falls under	Categorical	Sociodemographic
State	The Nigerian state in which the applicant resides	Categorical	Sociodemographic
Successful Payments	The number of successful loan repayments extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
Time at Current Address	The time in months the client has spent at their current residence	Numeric (Discrete)	Sociodemographic
Time at Employer	The time in months the client has spent at their current employer	Numeric (Discrete)	Sociodemographic
Unsuccessful Payments	The number of unsuccessful loan repayments extracted from the SMS messages received by the applicant from other micro-finance companies	Numeric (Discrete)	Alternative
VPN Count	The number of virtual private network applications on the applicant's cellular device	Numeric (Discrete)	Alternative

TABLE A.1: Variables Used in Models

Appendix B

GitHub Repository

The Github repository containing all data, all data preprocessing steps, and all modelling steps used throughout this project can be accessed using the following link:

https://github.com/devon12stone/masters-thesis-code