Strathmore
UNIVERSITY

Credit Risk Modelling in Peer-to-Peer Lending: A Comparative Analysis of Neural Networks and XGBoost

Wachira Njomo
100083

Submitted in partial fulfilment of the requirements for the Degree of Bachelor of Business Science in Actuarial Science at Strathmore University

Strathmore Institute of Mathematical Sciences
Strathmore University
Nairobi, Kenya

February 2021

DECLARATION

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the Research Project contains no material previously published or written by another person except where due reference is made in the Research Project itself.

Wachira Njomo ..... [*Name of Candidate*]

..... [*Signature*]

10/02/2021 ..... [*Date*]

This Research Project has been submitted for examination with my approval as the Supervisor.

..... [*Name of Supervisor*]

..... [*Signature*]

..... [*Date*]

Strathmore Institute of Mathematical Sciences

Strathmore University

Abstract

Consumer credit risk modelling involves coming up with the probability that a borrower will default thus classifying borrowers as either defaulters or non-defaulters. This is important for lending firms because they are then able to lend out cash to borrowers who are most likely to repay on time. This protects their profits. This study aims to use machine learning techniques in consumer credit risk modelling in a bid to find out which technique is more effective. In addition, the study aims at investigating the effect of new credit customers on default experience.

The data used was from Bondora, an online P2P lending firm based in Europe. Two models are used to model credit risk, they are XGBoost and Deep Neural Networks. The default experience of new credit customers was 43.34% compared to that of existing customers (29.57%). Their performance is evaluated using the following performance metrics: Accuracy, Precision, Recall, F1-score and Precision-Recall AUC. Both models outperformed the benchmark logistic model in all the performance metrics except Recall. Because the data is imbalanced the Precision-Recall AUC is used to determine the overall most effective model for credit risk modelling.

The XGBoost model performed the best with a Precision-Recall AUC of 0.6716. The Deep Neural Network model had a Precision-Recall AUC of 0.6701. The difference between the two models was very small.

Table of Contents

List of Abbreviations

ANN - Artificial Neural Network

AUC - Area Under the Curve

CSVM - Clustered Support Vector Machine

DNN - Deep Neural Network

GBM - Gradient Boosting Machine

LDA - Linear Discriminant Analysis

P2P - Peer-to-Peer

PR - Precision-Recall

ReLU - Rectified Linear Unit

SVM - Support Vector Machine

XGBoost - eXtreme Gradient Boosting

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Background of the study

Credit risk is the likelihood of default on debt payment. It arises as a result of a borrower failing to fulfil his/her debt obligations as required by the terms agreed with the lending institution.

Lending institutions make money from advancing loans to individuals. The downside of lending out money is that it exposes the institutions to credit risk. For lending institutions to become profitable and competitive they need to manage their credit risk. This makes credit risk modelling important for lending institutions.

Peer-to-Peer(P2P) lending is an online form of financing where individuals lend or borrow from each other through an online platform without the involvement of traditional financial intermediaries. Individuals place a loan request on the lending platform and investors (lenders) bid to fund the loan. Loan requests span from small amounts to medium amounts. The borrowers provide their personal information as well as financial information which includes the purpose of the loan. Lenders use the information provided to make an investment decision.

One characteristic feature of P2P lending is the lack of collateral required. This coupled with the absence of traditional intermediaries attracts borrowers that find it difficult to secure loans from traditional banks. P2P lending also suffers from asymmetric information. This arises when the borrowers' have more information on their financial state than the lenders. It leads to the trustworthiness of the borrowers' being unknown to lenders. Information asymmetry is exacerbated when the P2P lending platforms are used for microfinance where the targeted customers are mostly economically under-privileged people (Yum et al., 2012). This leads to adverse selection.

To help investors manage credit risk, P2P platforms provide a credit rating for each loan (Guo et al., 2016). The ratings hinge on the information provided in the underwriting stage by the borrowers. The ratings consider the features of the loan such as the loan amount and the financial information of the applicant such as the the borrowers' assets and debts. It also takes into account the applicant's personal information such as type of employment. Loans are

categorized into risk groups according to the level of risk. Investors can diversify their portfolios by funding loans in different risk groups. Investors in P2P lending marketplaces are thus faced with the decision of what loans to fund and how much money to apportion to the loans.

1.2 Problem Statement

Traditional credit scoring techniques are modelled using linear techniques that include linear discriminant analysis (LDA) and logistic regression. The advancement in technology has increased the computing power of computers which has made it possible to apply non-parametric techniques for credit scoring such as neural networks, support vector machines (SVM), gradient boosting among others. These techniques are collectively referred to as machine learning techniques. Khandani et al. (2010) constructed forecasting models of consumer credit risk among credit card holders of a major commercial bank using machine learning methods. Harris (2015) applied the clustered suppport vector machine (CSVM) on a German credit scoring dataset and data collected from a Barbados based credit union.

According to Statista (2020) the transaction value of P2P consumer lending in Europe had grown to 2.9 billion US dollars as of 2018. However, Adhami et al., (2019) assessed more than 3000 loans executed on 68 European P2P platforms from 2013 and 2017 and discovered that on average credit risk is inversely related to the returns of the loans which implies that P2P loans are mispriced. This poses a problem to P2P lenders who invest their money and expect a return consistent with the risk they are undertaking.

Machine learning techniques can be implemented in P2P consumer credit risk modelling to bridge the mispricing gap. Moreover, it is possible to do feature selection with these non-parametric techniques which helps determine the features of the borrowers that have the highest influence on the probability of default.

This paper compares the predictive performance of a deep neural network model and XGBoost model in consumer credit risk modelling. This study will use data from a European P2P lending platform Bondora.

CHAPTER 2: LITERATURE REVIEW

2.1 Theoretical Framework

2.1.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are an interconnected group of nodes that perform information processing and are inspired by the biological neural network in the brain. The motivation for ANNs is from McCulloch & Pitts (1943) whose seminal work was a model of neuron as a binary thresholding device in discrete time. Most ANNs are comprised of 3 layers: the input layer, the hidden layer(s) and the output layer. If the number of hidden layers is greater than one, the network is called a deep neural network (DNN) (Hamori et al., 2018). The input layer receives the input data and the neurons in the input layer get activated. The output from the input layer is then used as an input in the second layer of neurons. This process continues across the layers in the network until the final layer outputs the result.

ANN classifiers are based on the perceptron by Rosenblatt (1958) which was inspired by (McCulloch & Pitts, 1943). A perceptron takes several binary inputs $x_1, x_2, \ldots, x_n$ and produces a single binary output (Nielsen, 2015). The perceptron has weights $w_1, w_2, \ldots, w_n$ which are real numbers that reflect the importance of the respective inputs. The perceptron also has a bias term, b.

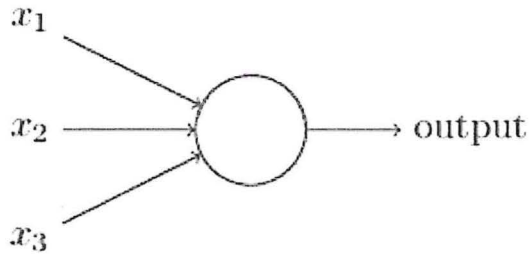

Figure 1: Structure of a perceptron

Source: Nielsen (2015)

The output of the perceptron is either 0 or 1 and it depends on whether the weighted sum of the inputs and bias is greater than a threshold.

$$output = \begin{cases} 0 \ \ if \ \sum_n w_n x_n + b < threshold \\ 1 \ \ if \ \sum_n w_n x_n + b \geq threshold \end{cases} \quad \ldots (2.1)$$

There have been improvements to the perceptron such as activation functions. There are various activation functions such as ReLU function, Sigmoid function, Tanh function and Softmax function (Goodfellow et al., 2016). Activation functions are mathematical expressions that establish the output of a neuron. They show the relationship between the output and input in each neuron.

Multi-layer neural networks are formed by organizing multiple neurons into several layers.

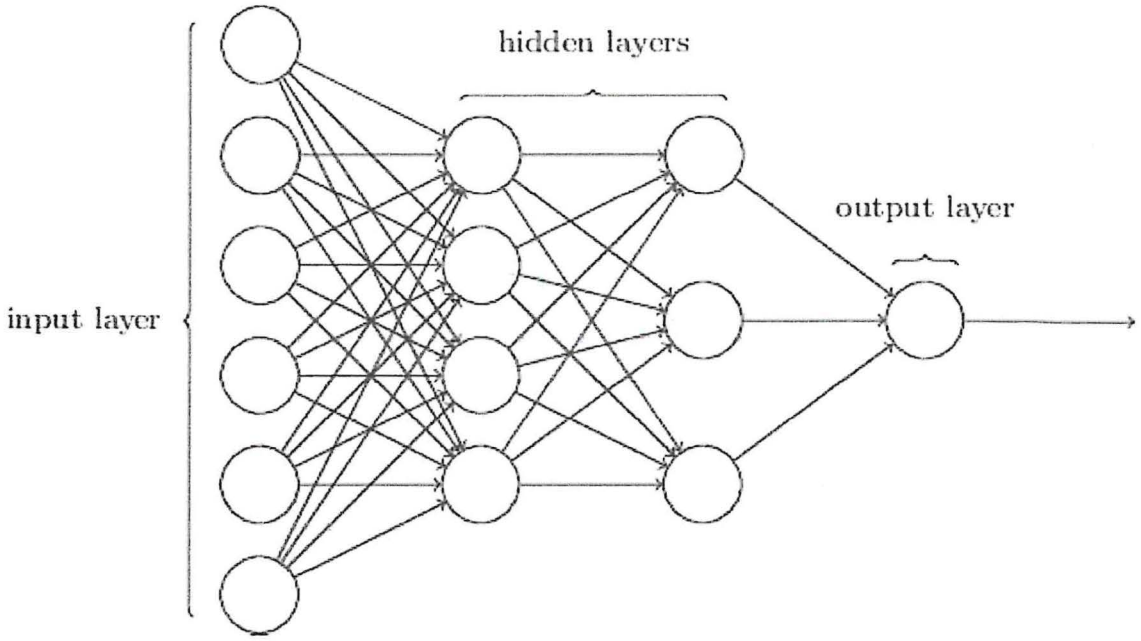

Figure 2: Structure of a neural network

Source: Nielsen (2015)

The predicted value which is the output of the neural network is compared to the actual value using a cost function, C. There are a number of cost functions such as quadratic loss and cross-entropy. The cross-entropy cost function is mostly used for classification and will be used in this study. It is defined as: (Nielsen, 2015)

$$C = -\frac{1}{n}\sum_x [y \ln a + (1-y)\ln(1-a)] \quad \ldots (2.2)$$

In equation (2): n is the number of training inputs x, y the desired output for each training input and a is the activated weighted inputs.

When training a neural network random values are initialized for the weights and biases. The aim of training is to get values for the neural network weights and biases that minimise the

cost function. In order to do this a concept called gradient descent is applied. Gradient descent is an optimization algorithm that minimizes a function by repeatedly going in the direction that reduces the gradient (El-Amir & Hamdy, 2019). This is done by implementing an update rule which changes the values of the weights w_k and the biases b_l (Nielsen, 2015)

$$w_k \rightarrow w'_k = w_k - \eta \frac{\partial C}{\partial w_k} \ldots (2.3)$$

$$b_l \rightarrow b'_l = b_l - \eta \frac{\partial C}{\partial b_l} \ldots (2.4)$$

η is the learning rate. The backpropagation algorithm is used to get the partial derivatives $\frac{\partial C}{\partial w_l}$ and $\frac{\partial C}{\partial b_l}$ of the cost function C (Rumelhart et al.,1986).

2.1.2 XGBoost

Friedman (2001) developed the gradient boosting machine (GBM). Gradient boosting aims to reduce the loss of the model by improving on the weak learners (decision trees). Pseudo residuals of the observations are derived by differentiating the loss function of the model. The pseudo residuals of the model are fit to a decision tree. Whenever a new decision tree is created the existing decision trees are left unchanged. A new tree's output is added to the output of the preceding series of trees in order to refine the model's final output. Afterwards pseudo residuals of the observations are calculated using the new output. This process continues until the loss of the model reaches an acceptable level or a fixed number of trees have been created.

Chen & Guestrin (2016) came up with XGBoost which is 'a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework'. XGBoost improves on the base GBM framework through:

- Regularization to avoid overfitting
- Parallelized tree building
- Efficient handling of sparse data
- Cache awareness and out-of-core computing
- Weighted quantile sketch

A more detailed description of the technique can be found at (Chen & Guestrin, XGBoost: A Scalable Tree Boosting System, 2016).

2.1.3 Logistic Regression

Logistic regression is commonly used in classification tasks owing to the fact that it is easy to understand and quick to implement. It assumes there exists a linear relationship between instance variables and the instance class. Logistic regression is used to predict the probabilities of a certain instance belonging to a particular class. A threshold is used to determine whether the probability of a given instance belonging to a particular class is sufficient to warrant classifying it as that particular class.

2.2 Empirical Literature

One of the first studies on modelling corporate risk was by Altman et al., (1994) who used linear discriminant analysis (LDA) and neural networks to spot financial distress among a thousand Italian firms. Both techniques had over 90% classification and holdout accuracy. The authors were quick to note that neural networks are black-box systems that have illogical weights and suffer from overfitting in the training stageand that these issues negatively impact predictive accuracy.

West (2000) investigated the accuracy of five neural network. A German dataset with 700 creditworthy applicants and 300 uncreditworthy applicants and an Australian dataset with 307 creditworthy applicants and 383 uncreditworthy applicants were used in the study. The results were benchmarked against the traditional models that include LDA, logistic regression, K nearest neighbor (KNN), kernel density estimation and decision trees. The neural network model was the best performing credit scoring model.

Angelini et.al.,(2008) developed credit scoring models using two neural network models. The data was of small businesses from a bank in Italy. One system was a classic feedforward neural network while the other had a special purpose feedforward architecture. The outcomes from the models imply that neural networks are successful in credit risk modelling and result in low errors.

Khashman (2010) used neural network models based on the backpropagation algorithm to create a credit risk evaluation system. Data used to train the model was from a German credit dataset. The results demonstrate that neural networks can be used in processing credit applications.

Othieno & Wagacha (2014) used semi-markov models as a proxy for internal credit risk models for a portfolio of 1000 consumer loans in the Kenyan market over a period of 5 years to determine the transition probabilities to different credit status. The results showed that semi-markov models are effective in forecasting probabilities of default.

Byanjankar et al.,(2015) used neural networks to develop a credit risk assessment model. The neural network was trained on data from a European P2P lending platform, Bondora. The results of the neural network model were compared to a logistic regression model. The neural network model outperformed the logistic regression model in screening default loans.

Beque & Lessmann (2017) explored the prospects of extreme learning machines (ELM) in consumer credit risk modelling. The model was trained on three real world retail credit scoring datasets: Australian Credit, German Credit, and Thomas. The ELM model was compared against established benchmarks: KNN, ANN, SVM and regularized logistic regression. According to the researchers, the previous studies done using ELM for consumer credit risk modelling focused on the model's predictive performance but ignored other relevant performance facets. This is why in Beque & Lessmann (2017) the ELM model was assessed according to the following performance facets: ease of use, computational complexity and predictive performance. In every aspect ELM showed better or competitive results to the benchmarks.

Chang et al., (2018) came up with a credit risk assessment tool using XGBoost. The model was trained using loan data from a Taiwanese financial institution. The results were compared to those from a logistic regression model, a neural network model and an SVM. The XGBoost classifier yielded the highest AUC curve thus exhibiting superior accuracy.

Hamori et al., (2018) came up with credit risk assessment tools that were based on ensemble methods – bagging, random forest, boosting – and neural networks that had different activation functions. Data used was from payment data of a Taiwan bank and it involved credit card holders of the bank. The paper contrasted prediction accuracy and classsification accuracy of the various credit risk models. The resuts obtained indicated that boosting has a superior classification ability to other machine learning models including neural networks.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This chapter describes the methodology used in achieving the laid-out research objectives.

3.2 Research Design

This study fits a neural network model alongside an XGBoost model to a European P2P loan dataset to predict loan defaulters. The implementation of the models will be done using the Python programming language.

3.3 Data Processing

The data will be split in three groups: a training, validation and testing dataset. The hyperparameters for each model (such as the number of layers in a neural network and learning rate in XGBoost) will be tuned using the validation dataset. The best performing model on the validation dataset will be used on the testing dataset. This prevents overfitting.

3.4 Data Analysis

3.4.1 Artificial Neural Networks

A feedforward multilayer neural network will be fit to a loan dataset in this study. The number of hidden layers and neurons will be selected during training. The activation function used for the neurons in the hidden layer will be rectified linear unit (ReLU). The output of a ReLU neuron with input x, weight vector w, and bias b is:

$$max\ (0, w \cdot x + b) \ldots \quad (3.1)$$

One major benefit of the ReLU activation function over other activation functions such as tanh and sigmoid is faster supervised training of deep neural networks whose hidden layers are composed of ReLU (Glorot, Bordes, & Bengio, 2011).

The sigmoid activation function will be used for the output neuron. The output of a sigmoid neuron with input x, weight vector w and bias b is:

$$\sigma(w \cdot x + b) \ldots (3.2)$$

σ is the sigmoid function and is defined as:

$$\sigma(z) \equiv \frac{1}{1+e^{-z}} \ldots (3.3)$$

The cross-entropy cost function will be used to compute the loss. The loss will be minimized using the Adam optimization algorithm that is an extension of stochastic gradient descent. The Adam algorithm starts by computing the gradient of the cost function with respect to parameter θ. The gradient will be calculated using the backpropagation algorithm:

$$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1}) \ldots (3.4)$$

Adam updates exponential moving averages of the gradient (m_t) and the squared gradient (v_t). The moving averages are estimates of the 1st moment and the 2nd raw moment. The hyper-parameters $\beta_1, \beta_2 \in [0,1)$ manage the exponential decay rates of the moving averages. Adam updates the biased first moment estimate and second raw moment estimate (Kingma & Ba, 2015)

$$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \ldots (3.5)$$

$$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g^2{}_t \ldots (3.6)$$

The algorithm then computes the bias-corrected first and second raw moment estimate (Kingma & Ba, 2015)

$$\hat{m}_t \leftarrow m_t / (1 - \beta^t{}_1) \ldots (3.7)$$

$$\hat{v}_t \leftarrow v_t / (1 - \beta^t{}_2) \ldots (3.8)$$

Finally the algorithm updates the parameter (Kingma & Ba, 2015)

$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon) \ldots (3.9)$$

α is the stepsize.

The dropout technique Hinton et al., (2014) will be used to prevent overfitting in the neural network. The idea behind dropout is to temporarily delete half of the hidden neurons randomly from the neural network during training. This reduces the co-adaptation of neurons.

3.4.2 XGBoost

XGBoost was developed by Chen & Guestrin (2016). Assuming we have K trees:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F} \ldots (3.10)$$

where $\mathcal{F} = \{f(x) = w_{q(x)}\}(q : \mathbb{R}^m \to T, \mathrm{w} \in \mathbb{R}^T)$ is the space of regression trees. q is the structure of each tree that maps an example to the corresponding leaf index. T is the number of leaves in the tree. Each f_k corresponds to an independent tree structure q and leaf weights w. w_i represents the score on the i-th leaf. Given an example the decision rules in the trees (given by q) are used to classify it into leaves and calculate the final prediction by summing up the score in the corresponding leaves (given by w). To learn the set of functions used in the model, the following regularized objective is minimized (Chen & Guestrin, XGBoost: A Scaleable Tree Boosting System, 2016):

$$L(\phi) = \sum_i l(\hat{y}_i y_i) + \sum_k \Omega(f_k) \ldots (3.11)$$

Where:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2 \ldots (3.12)$$

l is a convex differentiable loss function. The Ω term penalizes the complexity of the model. This additional regularization term smooths the final learnt weights to avoid overfitting.

The model is trained in an additive manner. Assume $\hat{y}^t$ to be the prediction of the i-th instance at the t-th iteration (Chen & Guestrin, XGBoost: A Scaleable Tree Boosting System, 2016),

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \ldots (3.13)$$

In order to decide on which function to add the objective will be optimized (Chen & Guestrin, XGBoost: A Scaleable Tree Boosting System, 2016):

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \ldots (3.14)$$

The goal will be to find f_t that minimises the above objective. Taking a Taylor expansion of the objective (Chen & Guestrin, XGBoost: A Scaleable Tree Boosting System, 2016)

$$L^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)} + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \ldots (3.15)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the 1st and 2nd order gradient of the loss function.

Define $I_j = \{i|q(x_i) = j\}$ as the instance set of leaf j. Define $G_j = \sum_{i\in I_j} g_i$, $H_j = \sum_{i=I_j} h_i$. Equation (15) can be rewritten by expanding Ω (Chen & Guestrin, XGBoost: A Scaleable Tree Boosting System, 2016):

$$\tilde{L}^{(t)} = \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \ldots(3.14)$$

$$= \sum_{j=1}^{T}[G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2] + \gamma T \ldots(3.15)$$

The optimal weight w_j^* of leaf j can be computed by (Chen & Guestrin, XGBoost: A Scaleable Tree Boosting System, 2016):

$$w_j^* = -\frac{G_j}{H_j+\lambda} \ldots(3.16)$$

and the corresponding optimal value can be calculated by (Chen & Guestrin, XGBoost: A Scaleable Tree Boosting System, 2016):

$$\hat{L}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T \ldots(3.17)$$

Equation (17) can be used to measure how good a tree structure q is. A greedy algorithm that starts from a single leaf and iteratively adds branches to the tree is used to come up with the tree structure q. Let I_L and I_R be the instance sets of the left and right nodes after the split. Define $G_L = \sum_{i\in I_L} g_i$, $G_R = \sum_{i\in I_R} g_i$, $H_L = \sum_{i=I_L} h_i$ and

$H_R = \sum_{i=I_R} h_i$. The change of objective after adding the split is given by (Chen & Guestrin, XGBoost: A Scaleable Tree Boosting System, 2016):

$$Gain = \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda} - \lambda \ldots(3.18)$$

3.4.3 Logistic Regression

The output of logistic regression is similar to that of a sigmoid neuron, equation (3.2) above. The probability of a given instance x belonging to a particular class is given by:

$$h_\theta(x) = \sigma(\theta \cdot w + b) \ldots(3.19)$$

Where σ is the sigmoid function, w_s are the instance variables, θ_s the variable coefficients and b is the bias term.

The sigmoid function is given by:

$$\sigma(z) = \frac{1}{1+e^{-z}} \ldots (3.20)$$

The class of the instance is thus found through:

$$Y_{predicted} = \begin{cases} 0 \; if \; h_\theta(x) < threshold \\ 1 \; if \; h_\theta(x) \geq threshold \end{cases} \ldots (3.21)$$

The threshold should be in the following interval (0,1). A high threshold (>0.5) implies that the model wants to minimize the false positives / maximize the true positives while a low threshold (<0.5) implies that the model wants to minimize the false negatives / maximize true negatives.

During training the parameters (θ_s) are defined so that the model has a high probability for instances of class 1 and a low probability for instances of class 0.

3.5 Performance Evaluation

A confusion matrix is a table that describes the prediction performance of a classification model when compared to test data.

		Prediction	
		No Event	Event
Actual	No Event	True Negative (TN)	False Positive (FP)
	Event	False Negative (FN)	True Positive (TP)

Table 1: Confusion Matrix

True Positive (TP) – Represents the correctly predicted positive events among the observations. The observations that were predicted to default and are loan defaulters in the actual data.

False Positive (FP) – Represents the wrongly predicted positive events among the observations . The observations that were predicted to default but are not defaulters in the actual data.

False Negative (FN) – Represents the wrongly predicted negative events among the observations. The observations that were predicted to be non-defaulters but are defaulters in the actual data.

True Negative (TN) – Represents the correctly predicted negative events among the observations. The observations that were predicted to be non-defaulters but are defaulters in the actual data.

The prediction accuracy is given by :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \ldots (3.19)$$

This measure does not take the distribution of the class into account hence it is considered a poor measure for model evaluation used on imbalanced data. In imbalanced data the class of interest is absolutely outnumbered. This study deals with loan data where the number of defaults is outnumbered.

There are other measures that do well with models used on imbalanced datasets. They include:

1. Precision.

This is the ratio of all correctly classified positive observations among all the predicted positive observations.

$$Precision = \frac{TP}{TP+FP} \ldots (3.20)$$

2. Recall.

This is the ratio of all correctly classified positive observations among all the positive observations from actual data.

$$Recall = \frac{TP}{TP+FN} \ldots (3.21)$$

3. F-score.

This is the harmonic mean between Precision and Recall.

$$F - score = \frac{2*Precision*Recall}{Precision+Recall} \ldots (3.22)$$

Next the classification ability of each model will be analysed using the Precision-Recall Curve. It plots precision against recall for various thresholds. The Precision-Recall curve is the most appropriate for imbalanced data (Saito & Rehmsmeiter, 2015). This study involves imbalanced data which makes the Precision-Recall curve most suitable for this study.

CHAPTER 4: DATA ANALYSIS

4.1 Data Overview

The data used in this study was obtained from Bondora an online P2P lending platform based in Europe. The data covers the period 2015 to 2020. The table below shows a summary of that data:

No. of observations	147,514
No. of features	19
No. of non-defaults	92,380
No. of defaults	55,134
Proportion of non-defaults (%)	63%
Proportion of defaults (%)	37%

Table 2: Summary of data

The data has the following features: Age, Amount of previous loan before loan, Applied amount, Country, Debt to income ratio, Education, Employment duration with current employer, Existing liabilities, Free cash, Gender, Home ownership type, Total income, Language code, Number of previous loans before loan, New Credit Customer, Probability of default, Rating, Verification Type and Default Status.

A heatmap that contained the correlation of the various numerical features was plotted to further understand the data.

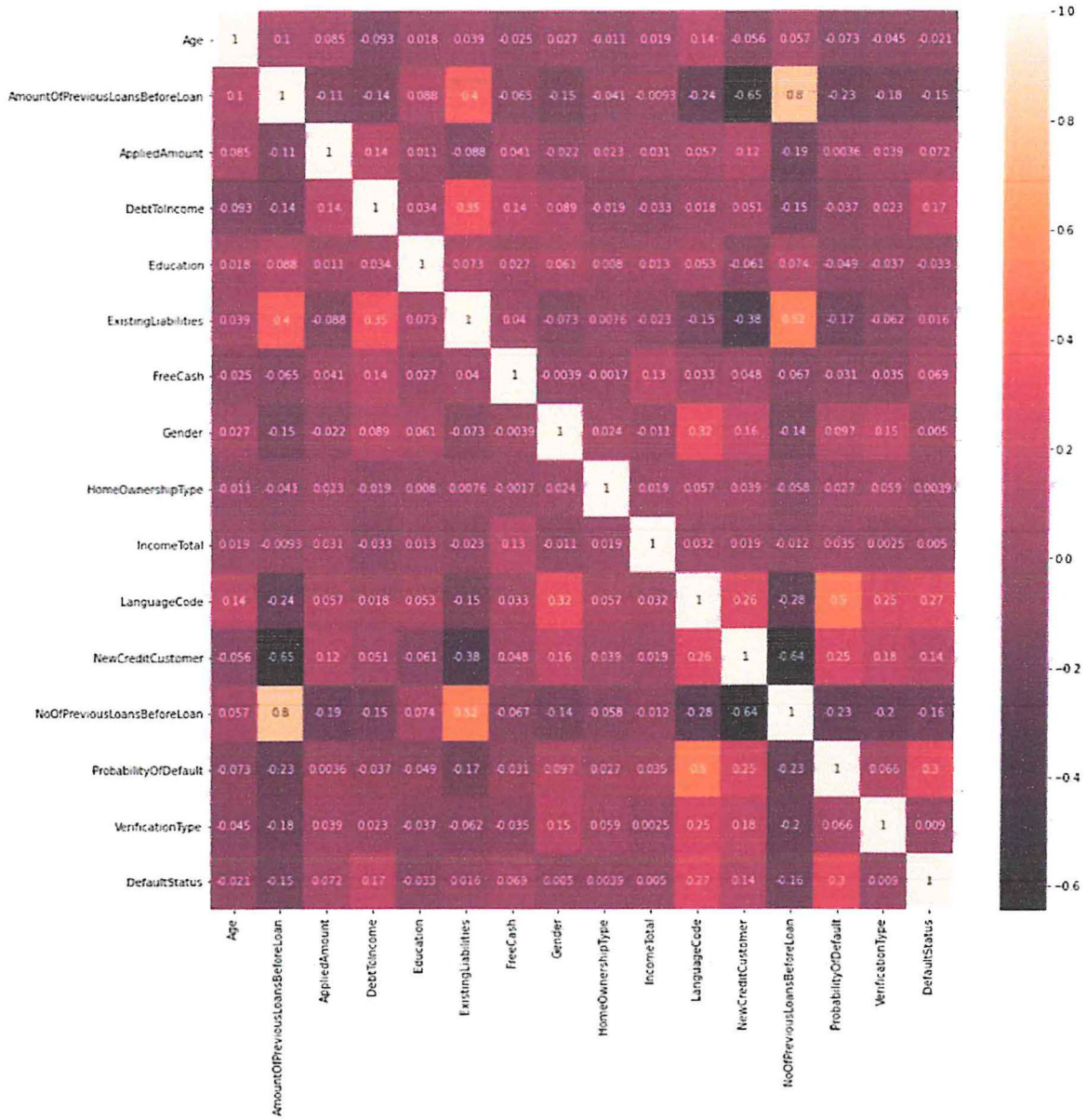

Figure 3: Correlation of numeric features

The correlation values are between 0.3 and -0.0033. The variables with the highest correlation to Default status were Probability of Default, Language code and Debt to income. Only 4 variables are negatively correlated with Default Status. They are: Age, Amount of loan before previous loan, Education and Number of previous loans before loan.

		No Default	Default
Age	<=20	1,528 (69.49%)	671 (30.51%)
	21-40	45,838 (61.33%)	28,901 (38.67%)
	41-60	37,968 (64.10%)	21,264 (35.90%)
	>60	7,046 (62.11%)	4,298 (37.89%)
Gender	0 (Male)	55,678 (60.98%)	35,626 (39.02%)
	1 (Woman)	30,805 (69.95%)	13,234 (30.05%)
	2 (Unidentified)	5,897 (48.45%)	6,274 (51.55%)
Probability of Default	< 0.5	90,458 (64.60%)	49,573 (35.40%)
	> 0.5	1,922 (25.68%)	5,561 (74.32%)
New Credit Customer	0(No)	45,008 (70.43%)	18,893 (29.57%)
	1(Yes)	47,372 (56.67%)	36,241 (43.34%)
Language Code	1 (Estonian)	52,168 (74.40%)	17,946 (25.60%)
	2 (English)	629 (58.68%)	443 (41.32%)
	3 (Russian)	10,329 (70.33%)	4,357 (29.67%)
	4 (Finnish)	18,422 (52.14%)	16,909 (47.86%)
	5 (German)	2 (40%)	3 (60%)
	6 (Spanish)	10,797 (41.52%)	15,208 (58.48%)
	9 (Slovakian)	29 (10.03%)	260 (89.97%)

Table 3: Default experience of specific numeric variables

It can be observed from the table above that higher proportion of males defaulted (39.02%) while 30.05% of females defaulted. This could be partly explained by the fact that majority of borrowers were men (91,304) compared to women (44,039). Majority of borrowers who did not disclose their gender defaulted (51.55%). Majority of borrowers who had a probability of default >0.5 defaulted (74.32%) whereas 35.40% of borrowers with a probability of default <0.5 defaulted. 43.34% of new credit customers defaulted compared to 29.57% of customers with prior credit history with Bondora. 89.97% of borrowers who speak in Slovakian defaulted and 60% of German speaking borrowers defaulted though there were only 5 German speaking borrowers. 58.48% of Spanish speaking borrowers defaulted, 47.86% of Finnish speaking borrowers defaulted and 41.32% of English-speaking borrowers defaulted.

Default experience according to age was relatively the same with default experience being in the range 30%-38%. The age group with the lowest default experience was <=20 though it accounted for only 1.49% of the data.

Before the data could be used it was split into training and testing datasets based on a 70/30 basis. Both training and testing datasets had an almost equal proportion of default entries (37%). The numerical variables were standardized by removing the mean and scaling to unit variance. The categorical variables were then converted into numeric variables via One-Hot encoding so that they could be used in the modelling exercise.

4.2 Model Analysis

The training data was further split into validation data on a 90/10 basis. Validation data was used to tune the models' hyperparameters via Bayesian optimization. A range of values for the different hyperparameters was defined. Different sets of values for the hyperparameters were evaluated for each model and the set of hyperparameters that produced the highest accuracy on the validation data was used as the models' hyperparameters for the test phase. This was a computationally intensive process. For the neural network model two activation functions were used. ReLU was used for the neurons in the hidden layer and logistic sigmoid was used for the output neuron as specified in Chapter 3.

After hyperparameter tuning the models with the new set of hyperparameters were evaluated on the testing data. The performance measures highlighted in Chapter 3 were used to assess the performance of the models on the testing data. They included Accuracy, Precision, Recall, F1-Score and Precision-Recall AUC.

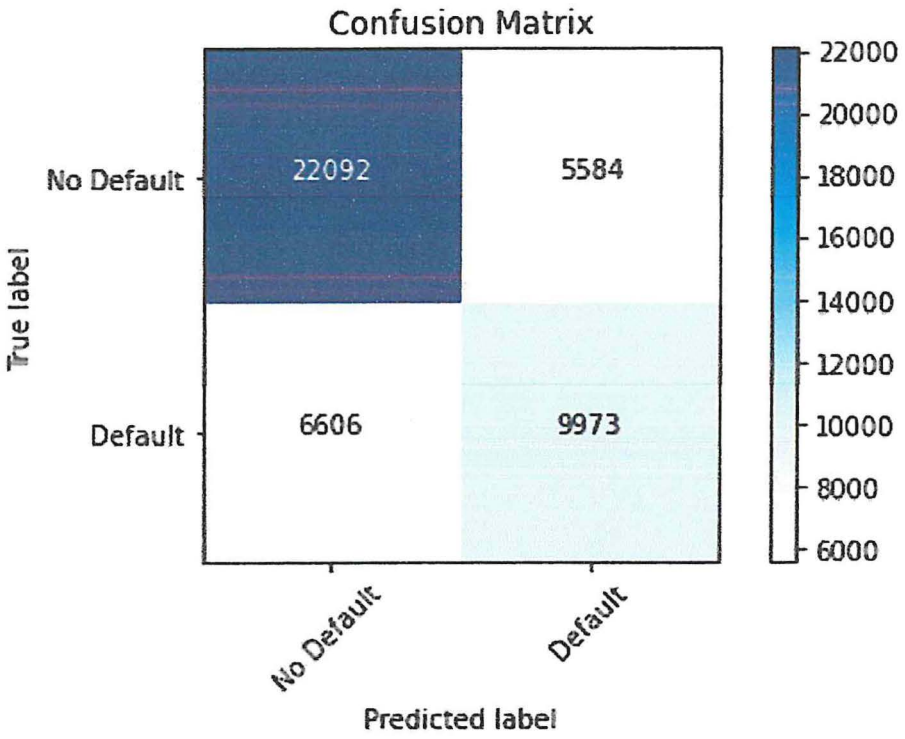

Figure 4: Neural Network Confusion Matrix

The confusion matrix describes the predictive performance of a classifier. The neural network model had 22092 true negatives, 6606 false negatives, 5584 false positives and 9973 true positives. For a lending institution, false negatives will lead to losses and will impact the financial condition of the firm. False positives on the other hand will limit the profits for a lending company because the lender misses out on earning interest by not lending to the false positives. 23.02% of predicted non-defaulters by the neural network ended up defaulting while 64.11% of predicted defaulters ended up defaulting.

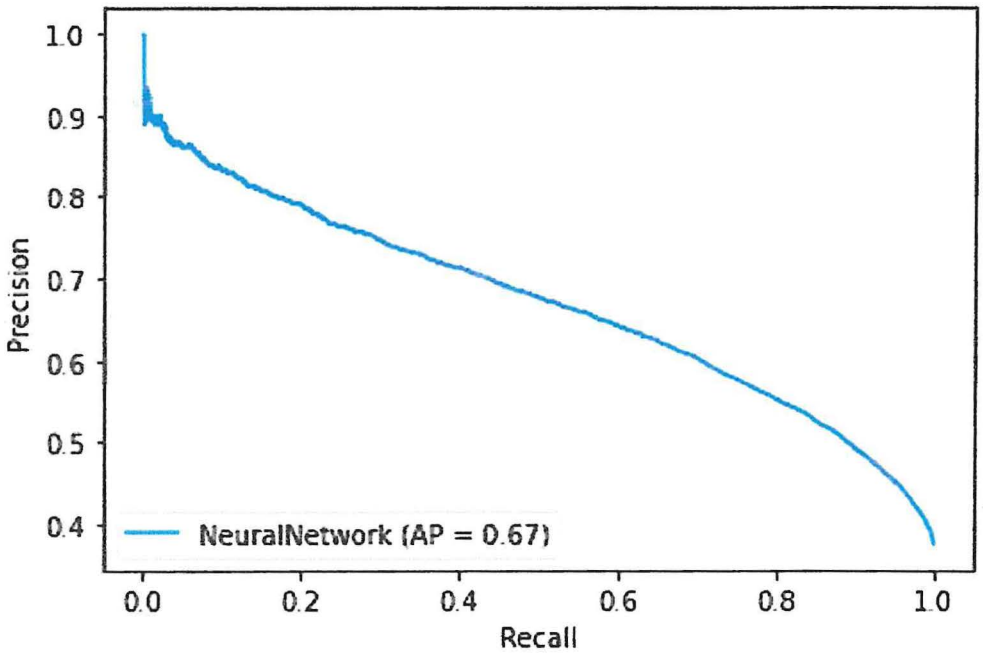

Figure 5: Neural Network Precision-Recall Curve

The Precision-Recall (PR) curve plots precision against recall for various thresholds. A high PR AUC indicates high precision and recall. This implies a low false positive and false negative rate. The Neural Network Model had a PR AUC of 0.6701. This means that the model predicts relatively few defaulters as non-defaulters and relatively few non-defaulters as defaulters.

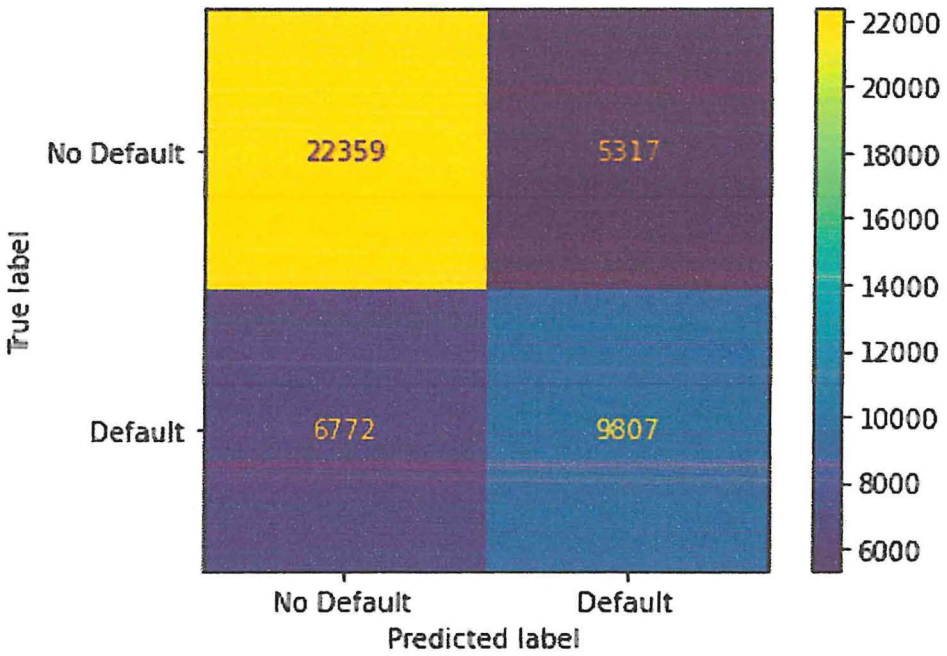

Figure 6: XGBoost Confusion Matrix

The XGBoost model had 22359 true negatives, 6772 false negatives, 5317 false positives and 9807 true positives. 23.25% of predicted non-defaulters ended up defaulting while 64.84% of predicted defaulters ended up defaulting.

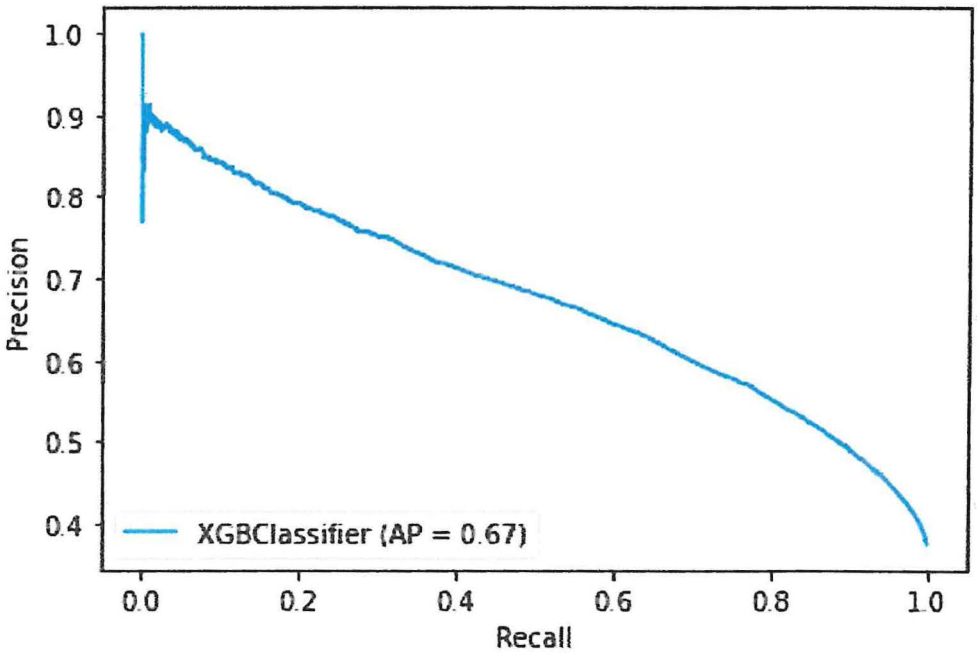

Figure 7: XGBoost Precision Recall-Curve

The XGBoost model had a PR AUC of 0.6716. This implies that the model has a slightly low false positive rate and false negative rate. This means that the model predicts relatively few defaulters as non-defaulters and relatively few non-defaulters as defaulters.

A logistic regression model was also implemented to benchmark its predictive performance with the XGBoost and Neural Network models.

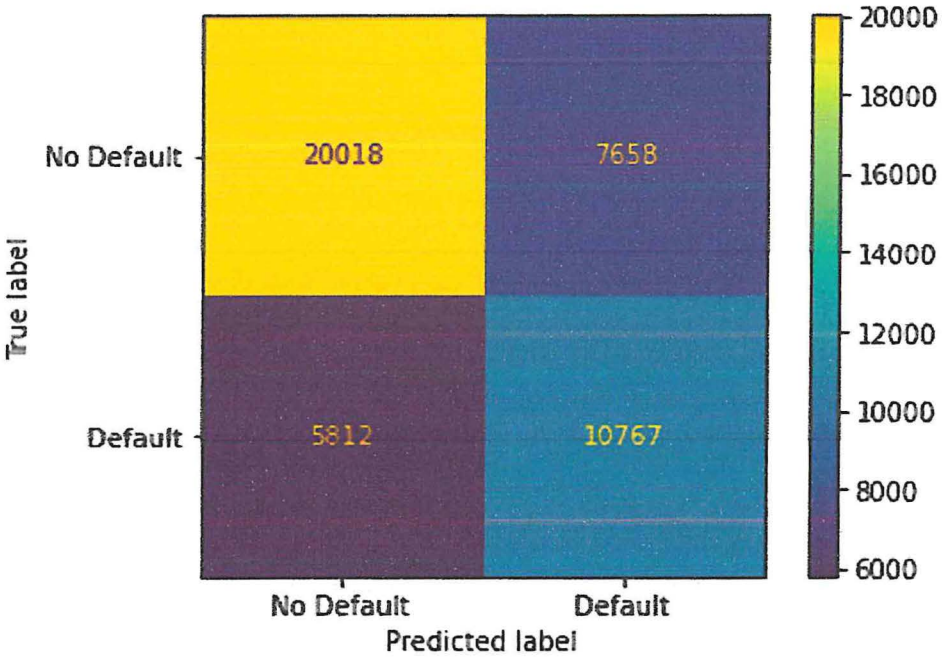

Figure 8: Logistic regression confusion matrix

The logistic regression model had 20018 true negatives, 5812 false negatives, 7658 false positives and 10767 true positives. 22.50% of predicted non-defaulters ended up defaulting while 58.43% of those predicted defaulters ended up defaulting.

Model	Precision Score	Recall Score	F1-Score	Accuracy	PR AUC
Neural Network	0.6411	0.6015	0.6207	0.7246	0.6701
XGBoost	0.6484	0.5915	0.6187	0.7268	0.6716
Logistic Regression	0.5843	0.6494	0.6152	0.6956	0.6300

Table 4: Summary of evaluation metrics results

Recall shows how many defaulters were correctly classified. The Neural Network model correctly classified 60.15% of defaulters while the XGBoost model correctly classified 59.15% of defaulters. So, the neural network model had the highest recall. Precision shows how many defaulters were correctly classified among all predicted defaulters. The neural network model had a precision of 64.11% and XGBoost had a precision of 64.84%. XGBoost had the highest precision. F1-score is the harmonic mean between recall and precision. The neural network model had an F1-score of 0.6207 and XGBoost had an F1-score of 0.6187. Accuracy is the proportion of all correctly predicted defaulters and non-defaulters in the data. The XGBoost model had the highest accuracy (72.68%) compared to the neural network model (72.46%). Both models had a Precision-Recall AUC of 0.67 which means that both models have slightly low false positive and false negative rates. The Precision-Recall (PR) curve plots precision against recall for various thresholds. A high PR AUC indicates high precision and recall which implies a low false positive and false negative rate.

Both models outperformed the benchmark logistic regression model in everything except Recall. The logistic regression model correctly classifies 64.94% of defaulters which is better than the XGBoost and neural network models which have a recall score of 60.15% and 59.15% respectively.

Since the data was imbalanced (63% of non-defaults and 37% of defaults), the PR AUC was used to determine the best model. The Precision-Recall curve is the most appropriate for imbalanced data (Saito & Rehmsmeiter, 2015). The XGBoost model had the highest PR AUC although the difference with the deep neural network model PR AUC was very small.

4.3 Discussion of Findings

A higher proportion of new credit customers (43.34%) default on P2P loans compared to credit customers with past experience with the P2P lending platform (29.57%). This implies that P2P lenders should be more wary when lending out to new credit customers.

The XGBoost model has the superior precision-recall AUC indicating better overall effectiveness in credit risk predictive accuracy when compared to the deep neural network model. Both the XGBoost and Neural network model outperformed the benchmark logistic regression model. This is consistent with results from other studies.

Byanjankar et al.,(2015) which trained a neural network on Bondora P2P loan data and compared its predictive performance with that of a benchmark logistic regression model. In that study the neural network correctly predicted 62.70% of non-default loans and 74.38% of default loans on the testing dataset. The results were obtained from 4811 observations. The neural network model outperformed the benchmark logistic regression model. The neural network model in this study on the other hand has correctly predicted 79.82% of non-default loans and 60.15% of default loans on the testing dataset. These results are based on 44255 observations. The neural network model outperformed the benchmark logistic regression model in similar fashion to the Byanjankar et al.,(2015) study.

Hamori et al., (2018) compared the predictive performance of deep neural networks and ensemble methods that included boosting in credit risk modelling on credit card payment data from a Taiwan bank. In that study the highest AUC obtained was from boosting and the highest F-score obtained was also from boosting. This is different from the results of this study where the XGBoost model had the highest AUC but the deep neural network had the highest F-score.

In Chang et al., (2018) the XGBoost credit risk assesssment model trained on loan data from a Taiwan financial institution yielded the highest AUC when compared to a neural network model thus exhibiting the superior predictive accuracy. The results are consistent with the results of this study.

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This study examined the performance of two machine learning techniques in consumer credit risk modelling. Both machine learning models, deep neural network and XGBoost, were successful in labelling loans as default and non-default. In addition the two models had a higher precision, f1 score, accuracy, and precision-recall AUC when compared to the benchmark logistic regression model. It shows that machine learning algorithms can be used to do consumer credit risk modelling. Being able to predict default loans in advance is important for a lending institution because it can then avoid lending to that borrower or charge a higher-than-normal interest if the lender is a risk taker. One major drawback of machine learning algorithms is that some subjects might be classified as defaulters yet in fact they are not. This may impact the profitability of the lending institutions because they will miss out on the interest from lending to someone who has mistakenly been predicted to be a defaulter.

From the data used in this study it was evident that more males borrow from P2P loan platforms compared to women. This also leads to males having a higher default behaviour (39.02%) compared to females (30.05%). New credit customers were more likely to default (43.34%) compared to customers with previous credit history with the firm (29.57%). Default experience according to age group was fairly constant. This study used imbalanced data (63% non-defaulters and 37% defaulters) which made credit risk modelling challenging.

For this study both the XGBoost model and the deep neural network model had an almost similar performance. The performance metric used to assess the overall performance was the Precision-Recall AUC. The Precision-Recall curve is the most appropriate for imbalanced data (Saito & Rehmsmeiter, 2015). This study involves imbalanced data which makes the Precision-Recall curve most suitable for this study. XGBoost had a Precision-Recall AUC of 0.6716 and the deep neural network model had a Precision-Recall AUC of 0.6701. The XGBoost model performed better than the deep neural network model which is consistent with the results of other studies such as the study by Hamori et al., (2018).

The two credit risk evaluation models in this study can be successfully implemented by P2P consumer lenders to screen their borrowers and depending on the output decide whether or not to invest in them.

5.2 Limitations

The factors that affected how the study was conducted include:

- Hyperparameter tuning was time consuming and computationally intensive. This impacted the extent to which the set of hyperparameters used for each model were optimized.
- Some of the features had a lot of missing data which forced the features to be removed from the data to be used.

5.3 Recommendations

I. A similar study should be conducted on data from Kenyan P2P firms. However a major drawback of this is inadequate data from Kenyan firms.

II. Future studies should go ahead and compute the estimated profit an online P2P firm would have made had it only used machine learning models to classify lenders.

References

Adhami, S., Gianfrate, G., & Johan, S. A. (2019, March 3). Risk and Returns in Crowdlending. doi:http://dx.doi.org/10.2139/ssrn.3345874

Altman, E., Marco, G., & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience). *Journal of Banking and Finance, 18*, 505-529.

Angelini, E., Tollo, G., & Roli, A. (2008). A neural network approach for credit risk evaluation. *The Quarterly Review of Economics and Finance, 48*, 733–755.

Angelini, E., Tollo, G., & Roli, A. (2008). A neural network approach for credit risk evaluation. *The Quarterly Review of Economics and Finance, 48*, 733-755.

Beque, A., & Lessmann, S. (2017). Extreme learning machines for credit scoring: An empirical evaluation. *Expert Systems with Applications, 86*, 42-53.

Byanjankar, A., Heikkila, M., & Mezei, J. (2015). Predicting Credit Risk in Peer-to-Peer Lending: A Neural Network Approach. *IEEE Symposium Series on Computational Intelligence*.

Chang, Y., Chang, K., & Wu, G. (2018). Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Applied Soft Computing, 73*, 914-920.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceeding of the 22nd ACM ACM SIGKDD International Conference*, (pp. 785-794). San Francisco.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scaleable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 785-794). San Francisco.

El-Amir, H., & Hamdy, M. (2019). *Deep Learning Pipeline*. Berkely, CA: Apress.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189-1232.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, (pp. 315-323). Fort Lauderdale, FL, USA.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Guo, Y., Zhou, W., Luo, C., Liu, C., & Xiong, H. (2016). Instance-based credit risk assessment for investment decisions in P2P lending. *European Journal of Operational Research, 249*(2), 417-426.

Hamori, S., Kawai, M., Kume, T., & Murakami, Y. (2018). Ensemble Learning or Deep Learning? Application to Default Risk Analysis. *Journal of Risk and Financial Management, 11*(1), 12.

Harris, T. (2015). Credit scoring using the clustered support vector machine. *Expert Systems with Applications, 42*(2), 741-750.

Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., & Slakhutinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research,* 1929-1958.

Khandani, A., Kim, A., & Lo, A. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance, 34,* 2767–2787.

Khashman, A. (2010). Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert Systems with Applications, 37,* 6233–6239.

Kingma, D., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representations.* San Diego.

McCulloch, S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biophysics.*

Nielsen, M. (2015). *Neural Networks and Deep Learning.* Determination Press.

Othieno, F., & Wagacha, A. (2014). Semi-Markov Credit Risk Modeling for a Portfolio of Consumer Loans in the Kenyan Banking Industry. doi:http://dx.doi.org/10.2139/ssrn.2524826

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6).

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by back-propagating errors. *Nature, 323,* 533-586.

Saito, T., & Rehmsmeiter, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE.*

Statista . (2020). *Alternative finance: P2P lending platforms transaction value in Europe 2013-2018.* Hamburg, Germany: Statista Research Department.

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research, 27,* 1131-1152.

Yum, H., Lee, B., & Chae, M. (2012). From the wisdom of crowds to my own judgment in microfinance through online peer-to-peer lending platforms. *Electronic Commerce Research and Applications, 11*(5), 469-483.