Debt recovery prediction in securitized non-performing loans using machine learning

EDVIN HEDBLOM

RASMUS ÅKERBLOM
Debt recovery prediction in securitized non-performing loans using machine learning

EDVIN HEDBLOM
RASMUS ÅKERBLOM

Degree Projects in Financial Mathematics (30 ECTS credits)
Master's Programme in Industrial Engineering and Management (120 credits)
KTH Royal Institute of Technology year 2019
Supervisor at Collectius: Andreas Granström
Supervisor at KTH: Camilla Landén
Examiner at KTH: Camilla Landén
Abstract

Credit scoring using machine learning has been gaining attention within the research field in recent decades and it is widely used in the financial sector today. Studies covering binary credit scoring of securitized non-performing loans are however very scarce. This paper is using random forest and artificial neural networks to predict debt recovery for such portfolios. As a performance benchmark, logistic regression is used. Due to the nature of high imbalance between the classes, the performance is evaluated mainly on the area under both the receiver operating characteristic curve and the precision-recall curve. This paper shows that random forest, artificial neural networks and logistic regression have similar performance. They all indicate an overall satisfactory ability to predict debt recovery and hold potential to be implemented in day-to-day business related to non-performing loans.
Sammanfattning


Titel: Prediktion av skuldbetalning av ickepresterande lån med maskininlärning
Acknowledgements

We would like to express our great appreciation to everyone at Collectius for their collaboration during the project. We would like to offer special thanks to our supervisor at Collectius, Andreas Granström, for valuable support during the project and for sharing knowledge in the world of machine learning. We would also like to extend our deepest gratitude to Gustav Eriksson and Ivar Björklund at Collectius for sharing their valuable experience with us and for making the project possible.

Furthermore, we would like to thank Camilla Landén, our supervisor at KTH, for exceptional guidance and for sharing her expertise on statistical analysis.

Stockholm, May 2019

Edvin Hedblom and Rasmus Åkerblom
# Table of Contents

Abstract ................................................................. i

Sammanfattning ........................................... ii

Acknowledgements ........................................ iii

1 Introduction ......................................................... 1
   1.1 Background ............................................. 1
   1.2 Credit Scores for Non-Performing Loans ......... 2
   1.3 Research Objectives ................................ 3

2 Theoretical Framework ........................................... 5
   2.1 Logistic Regression ..................................... 5
   2.2 Random Forest ........................................ 7
      2.2.1 Number of Trees ................................ 8
      2.2.2 Number of Variables ........................... 8
      2.2.3 Gini Impurity ................................ 9
      2.2.4 Out-of-Bag Error ................................ 10
   2.3 Artificial Neural Networks ......................... 10
      2.3.1 Framework ...................................... 11
      2.3.2 Activation Function ........................... 13
      2.3.3 Resilient Backpropagation .................. 13
      2.3.4 Number of Layers and Nodes ............... 15
   2.4 Variable Selection ..................................... 15
      2.4.1 Genetic Algorithm ............................ 15
      2.4.2 Mean Decrease Gini .......................... 18
   2.5 Evaluation Methods ................................... 18
      2.5.1 Performance Metrics ......................... 19
         2.5.1.1 Specificity ................................ 19
         2.5.1.2 Sensitivity ................................ 19
         2.5.1.3 Accuracy ................................ 19
         2.5.1.4 Balanced Accuracy ....................... 20
         2.5.1.5 Precision ................................ 20
3 Data and Methodology

3.1 Confidentiality
3.2 Overview
3.3 Portfolio Characteristics
3.4 Response Variable Definition
3.5 Model Assumptions
3.6 Variable Selection and Feature Engineering
3.7 Outlier Treatment
3.8 Sampling of Data
3.9 Feature Selection Procedure
3.10 Scaling
3.11 Programming Packages
3.12 Limitations

4 Results

4.1 Variable Selection
4.1.1 Optimizing ROC AUC
4.1.2 Optimizing PR AUC
4.2 Performance Metrics
4.2.1 ROC Curve
4.2.2 PR Curve
4.2.3 Density
4.2.4 Confusion Matrix ............................................. 38
4.2.5 Ratios ......................................................... 39

5 Discussion ......................................................... 41
  5.1 General ......................................................... 41
  5.2 Non-Performing Loan Portfolio Data ............................ 41
  5.3 Properties of the Models ....................................... 42
  5.4 Comparison of Model Performance .............................. 43
  5.5 Application Fields ........................................... 44
  5.6 Conclusions .................................................. 44
  5.7 Future Studies ............................................... 45

References ......................................................... 46
### Nomenclature

- **ANN**  Artificial Neural Network
- **AUC**  Area Under Curve
- **FN**  False Negative
- **FP**  False Positive
- **FPR**  False Positive Rate
- **GA**  Genetic Algorithm
- **LR**  Logistic Regression
- **MDG**  Mean Decrease Gini
- **m_{try}**  Number of variables available for split at each node in Random Forest
- **NPL**  Non-Performing Loan
- **n_{tree}**  Number of Trees in Random Forest
- **OOB**  Out-Of-Bag
- **PR**  Precision-Recall
- **RF**  Random Forest
- **ROC**  Receiver Operating Characteristic
- **TN**  True Negative
- **TP**  True Positive
- **TPR**  True Positive Rate
1 Introduction

A non-performing loan (NPL) is a loan where the debtor has not made payments within the agreed schedule of 90 or 180 days depending on the loan type. The loan can therefore be considered close to default since the debtor is less likely to repay the full amount. These loans are generally bad for the lender to hold due to the risk exposure. These types of loans are therefore often packaged to securitized debt portfolios which then are sold at a discount to companies specialized in collecting such loans.

One of these specialized companies is Collectius, a fintech company that has been present in the Asian market since 2014. Collectius has given access to one of its portfolios. Through the data collected from these portfolios, it enabled the modelling and analysis to be conducted. This portfolio is originated in the Philippines, one of the largest economies in Southeast Asia.

1.1 Background

Collectius is currently operational in Indonesia, Malaysia, Singapore, Thailand and the Philippines. The NPL ratios in these countries are below the world average NPL ratio of approximately 3.5% (The World Bank, 2017). An approximate NPL portfolio market size and NPL ratio for the operational countries and three comparable countries are presented in Table 1.1 (Edmonds, Orr, Newton, Kong, & Bhatti, 2018).

Table 1.1

<table>
<thead>
<tr>
<th>Country</th>
<th>Approx. Market Size (USD)</th>
<th>NPL Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indonesia</td>
<td>$10 billion</td>
<td>3%</td>
</tr>
<tr>
<td>Malaysia</td>
<td>$6 billion</td>
<td>2%</td>
</tr>
<tr>
<td>Singapore</td>
<td>$7 billion</td>
<td>1%</td>
</tr>
<tr>
<td>Thailand</td>
<td>$12 billion</td>
<td>3%</td>
</tr>
<tr>
<td>The Philippines</td>
<td>$3 billion</td>
<td>2%</td>
</tr>
</tbody>
</table>

According to a survey conducted by FICO (2018), 70% of the responding senior collection managers in Asia Pacific are planning to integrate artificial intelligence
(AI) into their collection systems before the year of 2020. They also found that 48% of them believe the AI implementation can help them optimise collection decisions. Machine learning, an important branch of AI, uses probabilistic modelling to detect patterns and generate classifications which can be used in optimisation tasks and support decision making (Ghahramani, 2015). A comparison in credit scoring prediction performance of human-made predictions (FICO scores) and machine learning predictions showed that the usage of machine learning predictions could reduce the credit losses more than relying on human-made predictions (Munkhdalai, Munkhdalai, Namsrai, Lee, & Ryu, 2019).

1.2 Credit Scores for Non-Performing Loans

Credit scoring is an area that has been gaining attention in the research field since the introduction of the term in the 1950s but is still an area where advanced literature is rather limited (Mester, 1997; Abdou & Pointon, 2011). There is no global definition of a credit score, some variants of a definition are as follows:

"A statistical method used to predict the probability that a loan applicant or existing borrower will default or become delinquent.” (Mester, 1997, p. 3)

"The use of statistical models to transform relevant data into numerical measures that guide credit decisions.” (Anderson, 2007, p. 6)

"The set of decision models and their underlying techniques that aid lenders in the granting of consumer credit. These techniques decide who will get credit, how much credit they should get, and what operational strategies will enhance the profitability of the borrowers to the lenders.” (L. Thomas, Crook, & Edelman, 2017, p. 1)

A contributing reason for the growing interest in credit scoring is likely due to its potential in saving money for lenders in a time efficient way. Through credit scoring, applicants are categorized which allows the company to efficiently decline loan applications from people who are projected to be unable to pay back their debt (Frame, Srinivasan, & Woosley, 2001). In addition to this, methods for classification and credit scoring are continuously getting better and more effective as the computational power increases (L. C. Thomas, 2000).
This study will be investigating a rather peripheral area of the loan industry and credit scoring, an area that is a lot less researched. Credit scoring of an NPL portfolio could be seen as a reversed procedure, where the goal no longer is to score whether a person will become unable to pay but rather whether a person will be able to start paying. The concept is closely related to classification of imbalanced financial data, much like being able to classify fraudulent transactions which is a field with more research conducted (Abdou & Pointon, 2011). Instead of classifying the rather unlikely event of fraud, this study is aiming to classify the rather likeliness of a person not being able to pay their debts for several months instead will start doing so. Hand and Henley (1997) compares this type of situation to a marketing exercise, where the response rates can be as low as 1% or 2%. In this kind of scenario, there are a vast amount of people to contact during a limited period of time with low probability of success for each individual. This also holds true in the debt collection industry, where each failed collection attempt is an indirect cost. Under these circumstances, it is of utmost importance to be able to differentiate high potential debtors and prioritize them.

1.3 Research Objectives

The main research objective of this study is to compare the ability of Random Forest (RF) and Artificial Neural Networks (ANN) to correctly classify people in an NPL portfolio. The more traditional method, Logistic Regression (LR), will be used as a performance benchmark. The classification will be binary, where a person will be categorized as paying or non-paying. In an event where a payment is made within a certain time frame as agreed, puts the person in the paying category. Otherwise, the person will be categorized under non-paying.

To compare the models, the evaluation methods presented in Table 1.2 will be used.
Table 1.2

*Evaluation methods.*

<table>
<thead>
<tr>
<th>Method</th>
<th>Section Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>2.5.1.3</td>
</tr>
<tr>
<td>Area under Precision-Recall Curve</td>
<td>2.5.4</td>
</tr>
<tr>
<td>Area under Receiver Operating Characteristic Curve</td>
<td>2.5.3</td>
</tr>
<tr>
<td>Balanced Accuracy</td>
<td>2.5.1.4</td>
</tr>
<tr>
<td>Cohen’s Kappa</td>
<td>2.5.1.7</td>
</tr>
<tr>
<td>Computational Intensiveness</td>
<td>2.5.9</td>
</tr>
<tr>
<td>Confusion Matrix Distribution</td>
<td>2.5.2</td>
</tr>
<tr>
<td>Cross-Entropy</td>
<td>2.5.5</td>
</tr>
<tr>
<td>Ease of Implementation</td>
<td>2.5.6</td>
</tr>
<tr>
<td>Flexibility of Model</td>
<td>2.5.7</td>
</tr>
<tr>
<td>Negative Predictive Value</td>
<td>2.5.1.6</td>
</tr>
<tr>
<td>Precision</td>
<td>2.5.1.5</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>2.5.1.2</td>
</tr>
<tr>
<td>Specificity</td>
<td>2.5.1.1</td>
</tr>
<tr>
<td>Weighted Cross-Entropy</td>
<td>2.5.5</td>
</tr>
</tbody>
</table>
2 Theoretical Framework

In this section the mathematical theory used in the study is explained. The underlying theory for the models, relevant evaluation methods and variable selection methods are examined.

2.1 Logistic Regression

LR is a part of the regression model family and dates back to the early 19th century (Cramer, 2002). One of the main differences for LR compared to other regression models is that the response variable is discrete. This study has one binary class (1=paying, 0=non-paying) and therefore it will use a binary LR.

Similar to LR the binary LR optimizes the coefficients \( \beta \) in \( Y^{(i)} = \alpha + \beta x^{(i)} \) where \( \alpha \) is the intercept and \( x^{(i)} \) is the input variables corresponding to observation \( i \) (Hosmer Jr, Lemeshow, & Sturdivant, 2013). Since the binary LR predicts values bounded between zero and one it needs to constrain predictions within the same boundary. This is achieved by using the sigmoid function \( \pi(x^{(i)}) = \frac{1}{1+e^{-Y^{(i)}}} \) displayed in Figure 2.1. For each observation \( x^{(i)} \) the prediction \( \pi(x^{(i)}) \) is computed.

\[ l(\beta) = \prod_{i=1}^{n} \pi(x^{(i)})^{y^{(i)}}[1 - \pi(x^{(i)})]^{1-y^{(i)}} \]  

In Equation (1), the likelihood function is presented where \( y^{(i)} \) is the binary response variable corresponding to \( x^{(i)} \). As the logarithm is a monotone function, maximizing the logarithm of the likelihood is equivalent to maximizing the likelihood function.
This is also equivalent to minimizing the cross-entropy (found in Section 2.5.5). Due to this equivalence, this study will use the log likelihood function as the prediction error for LR and it is defined in Equation (2).

\[
L(\beta) = \ln[l(\beta)] = \sum_{i=1}^{n} \left( y^{(i)} \ln[\pi(x^{(i)})] + (1 - y^{(i)}) \ln[1 - \pi(x^{(i)})] \right) \tag{2}
\]

In order for the model to iterate towards an optimal solution the partial derivative with respect to the weight \(\beta_j\) is computed for each weight \(j\). The derivative of the log likelihood function is derived to Equation (3).

\[
\frac{\partial L(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} \left( y^{(i)} - \pi(x^{(i)}) \right) x^{(i)}_j \tag{3}
\]

By letting the derivative be equal to zero, it can be proven that there is no closed form solution to this optimization problem and therefore LR take an iterative approach. This is done by calculating the prediction error, using gradient ascent to update the weights and continue this iteration until a certain threshold is reached. The gradient ascent method is updating the weights according to Equation (4).

\[
\beta^{(t)}_j := \beta^{(t-1)}_j + \eta \frac{\partial L(\beta^{(t-1)})}{\partial \beta_j} \tag{4}
\]

Where \(\eta\) is the step size taken in the gradient’s direction for each iteration \(t\).
2.2 Random Forest

Figure 2.2. Illustration of the RF model where two trees are shown and each circle represents a node where a split is made.

RF is an ensemble learning model introduced by Breiman (2001) which is a modification of earlier methods by using bagging techniques, see Figure 2.2 for an exemplifying illustration of the model. Bagging is the combination of bootstrapping and aggregation, an approach that tends to reduce the error rates compared to similar learners not using the technique (Domingos, 1997). Liaw and Wiener (2002) presents RF modelling in the following way:

- **Bootstrap:** Decide the number of trees ($m_{\text{tree}}$) to be used in the forest and draw $m_{\text{tree}}$ bootstrapped populations (with replacement) of equal size as the total population (see Section 2.2.1 for how $m_{\text{tree}}$ can be decided and Section 2.2.4 for details regarding bootstrapping).

- **Splits:** For each tree, create a classification tree where each node is split from a random subsample of variables ($m_{\text{try}}$) where the optimal split is chosen among these subsampled variables (see Section 2.2.2 on how to decide which $m_{\text{try}}$ to use and Section 2.2.3 for further information on how the optimal split is generated). The splitting procedure continues until node purity is obtained. This indicates that all debtors in a node have same response variable, or until no improving split is possible based on variables available.
- **Voting:** When all $m_{\text{tree}}$ trees have been created, new data can be run through all trees and receive a classification for each tree which then are aggregated to generate a classification by majority vote.

RF has been gaining attention and has been used in a vast number of different fields since it was first introduced. Rodriguez-Galiano, Ghimire, Rogan, Chica-Olmo, and Rigol-Sanchez (2012) mentions some key benefits using RF when working with machine learning:

- Fast and efficient on large data sets.
- Capable of handling large number of variables.
- Generates variable importance when classifying.
- Shows robustness when it comes to outliers and noise.

### 2.2.1 Number of Trees

A broadly accurate guideline would suggest that more trees generates lower generalization error. Due to the Strong Law of Large Numbers, the error almost surely converges as $m_{\text{tree}}$ increases (Breiman, 2001). This is the reason behind why RF has the attractive property of not overfitting data as more trees are added to the model.

The $m_{\text{tree}}$ suitability for a satisfactory model depends on the model parameters, what kind of data that is used and how much of an error is considered acceptable. Liaw and Wiener (2002) states that the number of $m_{\text{tree}}$ required in order to achieve satisfactory results grows with the number of variables. As a rule of thumb, an $m_{\text{tree}}$ between 100 and 500 generally generates satisfactory results and stabilized errors (Breiman, 2001; Oshiro, Perez, & Baranauskas, 2012; Friedman, Hastie, & Tibshirani, 2001; Khoshgoftaar, Golawala, & Van Hulse, 2007).

### 2.2.2 Number of Variables

There is no general rule on how to choose $m_{\text{try}}$ for each node in RF, different studies suggests different approaches (Breiman, 2001; Friedman et al., 2001; Khoshgoftaar
et al., 2007). The optimal \( m_{\text{try}} \) could vary depending on the problem formulation and whether regression or classification is of interest (Friedman et al., 2001). For classification, a study conducted by Breiman (2001) suggests the number of variables to \( \sqrt{M} \), where \( M \) is the total number of variables available (Liaw & Wiener, 2002). Another study evaluated several different data sets by setting \( m_{\text{try}} \) to 1, 2, 3, \( \frac{M}{10} \), \( \frac{M}{3} \), \( \frac{M}{2} \) and \( \log_2 M + 1 \) and concluded that \( \log_2 M + 1 \) generated the best results (Khoshgoftaar et al., 2007). While RF could be fine tuned by modelling with different \( m_{\text{try}} \), Breiman (2001) concludes that the result is rather insensitive to the \( m_{\text{try}} \) chosen.

2.2.3 Gini Impurity

At each node for every tree in RF, an optimal split is calculated among the variables randomly selected at each node. A common splitting criterion in RF is Gini impurity \( i(\tau) \) which is calculated in Equation (5).

\[
i(\tau) = 1 - p_1^2 - p_0^2
\]  

(5)

where \( p_k = \frac{n_k}{n} \) and \( n_k \) is the number of debtors from response variable \( k = \{\text{Non-paying (0), Paying (1)}\} \) from total population \( n \) at node \( \tau \) (Menze et al., 2009). The decrease \( \Delta i \) is calculated by comparing the two nodes generated from the split, that is calculated in Equation (6).

\[
\Delta i(\tau) = i(\tau) - p_l i(\tau_l) - p_r i(\tau_r)
\]  

(6)

where \( p_l = \frac{n_l}{n} \) and \( p_r = \frac{n_r}{n} \) are the fractions of the population arriving in the left node and right node respectively.

The optimal split is thereafter found by searching through all variables \( \theta \) that randomly has been selected at the current node and at all possible thresholds \( t_\theta \) for each of the variables. The variable and threshold that generates the largest \( \Delta i \) is finally chosen and the split is conducted.

The Gini impurity is closely related to variable importance and Mean Decrease Gini (MDG), see Section 2.4.2 for further details.
2.2.4 Out-of-Bag Error

RF is created by bootstrapping and each tree is sampled with replacement. Due to this, an average of approximately 37% of the observations will not be a part of creating a tree. The observations that are not a part of the tree creation, is known as the out-of-bag (OOB) data (Ishwaran, Kogalur, Blackstone, & Lauer, 2008; Liaw & Wiener, 2002). The OOB data can be used to evaluate the error rates. This is done by letting the unused observations be classified by all trees in which they were not a part of creating. Finally the procedure lets the majority vote generate a prediction \( \hat{f}_{\text{ooob}}(x_i) \) as in Equation (7).

\[
\hat{f}_{\text{ooob}}(x_i) = \begin{cases} 
1, & \text{if } \sum_{t=1}^{T} I\{f(t(x_i))=1\}I\{t\in F^-(x_i,T)\} > \sum_{t=1}^{T} I\{f(t(x_i))=0\}I\{t\in F^-(x_i,T)\} \\
0, & \text{otherwise}
\end{cases}
\]  

(7)

Where \( I \) is the indicator function, \( f(t(x_i)) \) is the prediction at tree \( t \) and \( F^-(x_i,T) \) is the set of trees of all trees \( T \) which has not been used for classification of observation \( x_i \) (C. Zhang & Ma, 2012; Ishwaran et al., 2008).

Using this, the OOB error (\( E_{\text{ooob}} \)) can be calculated with an average number of incorrect predicted responses by comparing it to the actual responses \( y_i \) over all observations \( N \):

\[
E_{\text{ooob}} = \frac{1}{N} \sum_{i=1}^{N} I\{y_i \neq \hat{f}_{\text{ooob}}(x_i)\}
\]  

(8)

The OOB error is a suitable estimate when evaluating an RF and deciding the number of trees needed to get a satisfactory outcome accurately. When the OOB error stabilizes there is no significant benefit in adding more trees (Liaw & Wiener, 2002; Friedman et al., 2001).

2.3 Artificial Neural Networks

The earliest models of the deep learning study field is dated back to the 1940s. Together with the connectionism era of the 1980s it became more popular to build statistical models with inspiration from the biological brain. This is when ANN took shape (Marsland, 2011). In the recent years ANN has been widely used in
credit scoring for financial institutions (Ayouche, Abouaich, & Ellaia, 2017; Pacelli & Azzollini, 2011) and have sometimes provided a significantly better classification in relation to traditional methods (Bensic, Sarlija, & Zekic-Susac, 2005; Imtiaz & Brimicombe, 2017; Nwulu & Oroja, 2011; Doori & Beyrouti, 2014). The previous studies on ANN usage in lending risk spans over a wide range of perspectives but the field of credit risks within NPL securities has not been studied thoroughly yet.

2.3.1 Framework

An ANN is a network that can find complex non-linear relations among variables. Its construction is inspired by the brain’s neurons and the connecting synapses between them (Van Gerven & Bohte, 2018). The ANN has an input and output layer connected via layers in between, so called hidden layers. Each layer has its own set of neurons. The neurons in one layer are all connected to the nearest layer’s neurons, hence connecting both the input and output layer indirectly. For all layers but the input layer there will also be a bias node which is only connected to the neurons in the same layer as itself. See Figure 2.3 for an illustrative example of an ANN.
Figure 2.3. An ANN with four nodes in the input layer, two nodes in the hidden layer and one output layer. The hidden layer and the output layer have one unit-valued bias node each.

Each connection in the network has a weight that will iteratively readjust until it reaches an optimized solution. This is done by first randomly choosing weights for all connections, calculating the predictions with the weights (via an activation function, see Section 2.3.2), computing the error and then propagating the error back for each neuron in the network. For all neuron values $s_i$ in the hidden layers and the output layer each value is a summation of all the neuron values in the previous layer (including a bias node) multiplied by a corresponding weight. Mathematically it can be expressed as in Equation (9).

$$s_i = \sum_j w_i^j h^j$$

In Equation (9) $j$ is the neuron number in the previous layer, $w_i^j$ is the connection’s weight (between node $j$ in the previous layer and the node $i$ in the current layer) and $h^j$ is the neuron value for node $j$ in the previous layer (Marsland, 2011).
2.3.2 Activation Function

In the case of binary classification of a response variable it is common to use an activation function. This function will take any real number and transform it to a value between zero and one. Therefore a sigmoid function $y_i(s) = \frac{1}{1+e^{-s}}$ (visualized in Figure 2.1) will be used in this ANN (Marsland, 2011).

2.3.3 Resilient Backpropagation

ANN has developed over the years and the number of learning methods with varying complexity and training speed has increased (LeCun, Touresky, Hinton, & Sejnowski, 1988). A traditionally used method to readjust the network weights was introduced as the gradient descent backpropagation by Werbos (1974). In 1992 this method inspired scientists to develop a faster converging version of this, called resilient backpropagation (Riedmiller, 1994). The procedure starts by computing the loss function’s partial derivative with respect to each weight respectively. If its partial derivative with respect to a certain weight is positive (negative) it means that the same weight should decrease (increase). When the network has one binary output class, the sigmoid function (from Section 2.3.2) is used as the activation function and cross-entropy (from Section 2.5.5) is used as the loss function then the loss functions partial derivative with respect to a certain weight $\frac{\partial CE}{\partial w_{ij}}$ is derived in Equations (10)-(14).

$$\frac{\partial CE}{\partial w_{ij}} = \frac{\partial CE}{\partial y_i} \cdot \frac{\partial y_i}{\partial s_i} \cdot \frac{\partial s_i}{\partial w_{ij}}$$ (10)

$$\frac{\partial CE}{\partial y_i} = \frac{y_i - t_i}{y_i(1 - y_i)}$$ (11)

$$\frac{\partial y_i}{\partial s_i} = y_i(1 - y_i)$$ (12)

$$\frac{\partial s_i}{\partial w_{ij}} = h_j$$ (13)
Where CE is the cross-entropy, \( y \) is the activation function, \( t \) is the observed class and \( s \) is a summation of each neuron in the previous layer multiplied by the connecting weight.

With Equations (11), (12) and (13) in (10), the expression simplifies to Equation (14).

\[
\frac{\partial \text{CE}}{\partial w_{ij}} = (y_i - t_i) h_j
\] (14)

In resilient backpropagation, a weight will update in the opposite direction of the sign on the error functions partial derivative with respect to that certain weight. The weight’s update difference will be multiplied by \( \eta^+ \) (\( \eta^- \)) if the derivative has the same (different) sign as in the previous step. Mathematically it can be expressed as:

\[
\Delta w_{ij}^{(t)} = \begin{cases} 
-\Delta_{ij}^{(t)}, & \text{if } \frac{\partial \text{CE}^{(t)}}{\partial w_{ij}} > 0 \\
+\Delta_{ij}^{(t)}, & \text{if } \frac{\partial \text{CE}^{(t)}}{\partial w_{ij}} < 0 \\
0, & \text{else}
\end{cases}
\] (15)

\[
\Delta_{ij}^{(t)} = \begin{cases} 
\eta^+ \cdot \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial \text{CE}^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial \text{CE}^{(t)}}{\partial w_{ij}} > 0 \\
\eta^- \cdot \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial \text{CE}^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial \text{CE}^{(t)}}{\partial w_{ij}} < 0 \\
\Delta_{ij}^{(t-1)}, & \text{else}
\end{cases}
\] (16)

where \( 0 < \eta^- < 1 < \eta^+ \) and \( \Delta_{ij} \) is the partial derivative with respect to the weight between node \( i \) and \( j \) (from Equation (14)).

The updated weights are then computed in Equation (17).

\[
w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}
\] (17)

This weight update procedure will go on until a given threshold is reached and the network has thereby converged (Riedmiller, 1994). Prechelt (1998) mentions that the threshold has to be chosen with regards to training time, efficiency, effectiveness, robustness, trade-offs and quantification which means that what is a suitable threshold will vary among studies.
2.3.4 Number of Layers and Nodes

The number of hidden layers that should be used in an ANN depends on the specific problem to be solved (Stathakis, 2009). For the majority of practical problems there is no need for more layers (Vujicic, Matijevic, Ljucovic, Balota, & Sevarac, 2016).

The number of nodes used in each hidden layer has not been studied thoroughly in previous research. There are some guidelines but no general optimized set of nodes. The ANN model in this study will use a number of hidden nodes derived from Equation (18) providing the best test error according to Vujicic et al. (2016).

\[ N_{\text{nodes}} = \frac{4N_{\text{vars}}^2 - 3}{N_{\text{vars}}^2 - 8} \]  \hspace{1cm} (18)

2.4 Variable Selection

In order to find an optimal prediction performance for each model only the best set of explanatory variables should be used as the input. An optimization algorithm that can be used for this purpose is the genetic algorithm (GA). It is known for its global search method of solving complex optimization problems and has been used for selection of input variables in ANN (D’heygere, Goethals, & De Pauw, 2002; Panagoulia, Tsekouras, & Kousiouris, 2017; Ahmad, Mat-Isa, Hussain, Boudville, & Osman, 2010). For RF specifically, MDG is a commonly used method for variable selection which enables comparative variable evaluation by ranking its importance (Han, Guo, & Yu, 2016; Rodriguez-Galiano et al., 2012).

2.4.1 Genetic Algorithm

GA is a self-learning method that can be used for solving optimization problems using an approach inspired by the Darwinian evolution theory. The algorithm starts with a population of chromosomes, each represented by a set of genes. Each chromosome is a suggested solution to the optimization problem (Marsland, 2011). In Figure 2.4 the relations are visualized.
To determine the suitability of a gene a fitness function is used. This function is used for each gene in the population to score them. In order to develop a better performing population for the next generation, three operators are used (Marsland, 2011).

- **Selection:** The idea is to select parents based on the chromosomes’ rank in terms of fitness score. Chromosomes with a high fitness score have a higher probability of being selected for reproduction.

- **Crossover:** For each pair of parents, a random crossover point is selected over which the genes are swapped.

- **Mutation:** A randomly chosen range of genes are flipped within a chromosome.

These operators will repeat and create new generations until the last generation is not significantly different from the previous ones. The algorithm is visualized in Figure 2.5 (Scrucca, 2013a; Marsland, 2011).
In order to find the optimal set of variables, GA is used on each model in this study and a performance metric (ROC AUC from Section 2.5.3 and PR AUC from Section 2.5.4) is the fitting function for the algorithm.

Figure 2.5. Algorithm visualization of a general GA process.
2.4.2 Mean Decrease Gini

MDG is closely related to Gini impurity (see Section 2.2.3) and is a method for evaluating the importance of each variable in the RF model (Han et al., 2016; Rodriguez-Galiano et al., 2012). For each variable \( \theta \), \( \text{MDG}_\theta \) sums the total decrease of impurity from all nodes \( \tau \) in each tree \( T \) and averages this by the number of trees used in the model.

For each variable \( \theta \), \( \text{MDG}_\theta \) is calculated as:

\[
\text{MDG}_\theta = \frac{\sum_T \sum_\tau \Delta i_\theta(\tau, T)}{n_{\text{tree}}}
\]

(19)

where \( \Delta i_\theta \) is the decrease of Gini impurity (Louppe, Wehenkel, Sutera, & Geurts, 2013).

A large MDG for a variable in comparison with other variables will give an indication of high importance (Menze et al., 2009). The large value shows that the variable often was selected as the most optimal variable to split in a node and that the split on average yielded a high decrease in Gini impurity. The opposite applies for variables with small MDG.

As mentioned by Nembrini, König, and Wright (2018), a drawback with MDG is relevant to highlight in this study. MDG has a tendency to be biased in favor of variables that are able to be split into many partitions (which usually are continuous variables). While there are other methods to measure variable importance, MDG is chosen due to attractive attributes such as computational effectiveness and robustness to perturbed data (Nembrini et al., 2018).

2.5 Evaluation Methods

This study will evaluate the models and examine the prediction performance based on the evaluation methods presented in this section.
2.5.1 Performance Metrics

A way to evaluate models is by comparing how well the predicted values match the actual values which can be done in several different ways (Lalkhen & McCluskey, 2008). In general, the comparisons consist of some combination of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) for some cut-off point. The cut-off point in this study will be set to balance specificity (Section 2.5.1.1) and sensitivity (Section 2.5.1.2).

2.5.1.1 Specificity

Specificity shows how well the model correctly classifies all negative predictions.

$$\frac{TN}{TN + FP} \quad (20)$$

If the model has 100% specificity that means that all non-paying debtors are classified correctly, whereas a lower percentage would mean that some of the paying debtors are falsely classified as non-paying.

2.5.1.2 Sensitivity

Sensitivity (also known as recall) shows how well the model correctly classifies all positive predictions.

$$\frac{TP}{TP + FN} \quad (21)$$

If the model has 100% sensitivity that means that all paying debtors are classified correctly, whereas a lower percentage would mean that some of the paying debtors are falsely classified as non-paying.

2.5.1.3 Accuracy

Accuracy combines specificity and sensitivity by evaluating how often the classifier is correct overall.

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (22)$$
When having imbalanced data, where a small percentage of the data belongs to one of the categories, accuracy could be misleading. If the model were to classify the entire population to be a part of the majority category the accuracy would still be able to come close to 100%.

2.5.1.4 Balanced Accuracy

Balanced accuracy is an alteration of Accuracy which generally is more suitable when having imbalanced data since the disadvantage with majority class domination is reduced. Balanced accuracy is calculated as:

\[
\frac{TP}{TP+FN} + \frac{TN}{TN+FP}
\]

(23)

2.5.1.5 Precision

Precision (also known as positive predictive value) shows how likely a debtor is truly paying given that the debtor is classified that way.

\[
\frac{TP}{TP+FP}
\]

(24)

2.5.1.6 Negative Predictive Value

Negative Predictive Value shows how likely a debtor is truly non-paying given that the debtor is classified that way.

\[
\frac{TN}{TN+FN}
\]

(25)

2.5.1.7 Cohen’s Kappa

Cohen’s Kappa is a measure of accuracy which shows the proportion of unitary categorization as the random chance of unitary categorization is excluded (Cohen, 1960).

Cohen’s Kappa (\(\kappa\)) is calculated as follows:

\[
\kappa = \frac{\pi_o - \pi_e}{1 - \pi_e}
\]

(26)
where $\pi_o$ is the proportion of correct classification (accuracy) and $\pi_e$ is the proportion of correct classification that could be expected by a chance (Cantor, 1996). Cohen’s Kappa is closely related to confusion matrices (see Section 2.5.2) and this could be expressed as:

$$\kappa = \frac{\sum_{i=1}^{I} P(x_{ii}) - \sum_{i=1}^{I} P(x_i)P(x_i)}{1 - \sum_{i=1}^{I} P(x_i)P(x_i)}$$

(27)

where $I$ is the number of categories, $P(x_{ii})$ are the accuracy probabilities of the main diagonal, $P(x_i)$ are the columns marginal probabilities and $P(x_i)$ are the row marginal probabilities (Ben-David, 2008). A Cohen’s Kappa of 1 would indicate perfect accuracy, 0 shows no accuracy and -1 would indicate perfect inaccuracy.

Cohen (1960) presents three assumptions that needs to be fulfilled in order for the measure to be relevant:

- Each data point should be independent.
- Each category should be independent, mutually exclusive and exhaustive.
- The categorization should be done independently.

An advantage of the way Cohen’s Kappa measures accuracy compared to the measure presented in Section 2.5.1.3 is that it presents a more impartial view when dealing with imbalanced data. This measure is also an important tool in balanced data, Ben-David (2008) found that 15 benchmark datasets showed that an average of one third of correct categorization could be attributed to chance.

### 2.5.2 Confusion Matrix

A confusion matrix is summarizing the prediction performance (separated on a cut-off point mentioned in the introductory part of Section 2.5.1) of classification by comparing predicted classes with actual classes. Since the classification in this case is binary with outcomes Paying (P) and Non-Paying (NP) there will be a 2x2 matrix, see Figure 2.6.
2.5.3 Receiver Operating Characteristic

The Receiver Operating Characteristic (ROC) curve is a visualization of the relation between the percentage of the true positive rate (TPR) and the percentage of the false positive rate (FPR) for different thresholds. The TPR is a fraction between the number of all TP classifications and the total number of true outcomes. The FPR is a fraction between the number of all FP classifications and the number of all negative outcomes. ROC is used both as an evaluation method as well as a fitting function for GA (see Section 2.4.1).

This fits the case of binary classifications due to the nature of pairwise comparison
in the ROC curve. In Figure 2.7 the dotted diagonal line represents random chance and all points above are better than random chance and all points below are worse. In the case of a highly skewed class distribution one will receive a larger number of errors due to poor inverse recall ($\frac{TN}{TN+FP}$) than the number of errors due to poor recall ($\frac{TP}{TP+FN}$), or vice versa (Powers, 2011).

For the ROC curve it is better the further away it is above the random line. To measure this behaviour throughout different thresholds it is common to measure the area under the ROC curve (ROC AUC). The bigger area the better. ROC AUC as a classification performance measure has gained popularity within machine learning research, showing comprehensive evaluation when modelling with binary classifiers (Vanderlooy & Hüllermeier, 2008; Huang & Ling, 2005; Provost, Fawcett, & Kohavi, 2001).

2.5.4 Precision-Recall

A Precision-Recall (PR) curve visualizes the trade-off relation for precision (from Section 2.5.1.5) and recall (from Section 2.5.1.2). Precision shows how likely a debtor is truly paying given that prediction classifies that way. Recall shows how well the model correctly classifies all positive predictions. When classifying imbalanced data, the PR curve can be useful because there is less interest in correctly predicting the high prevalence class (Saito & Rehmsmeier, 2015). As for ROC, PR is used both as an evaluation method as well as a fitting function for GA (see Section 2.4.1). In line with ROC, it can be useful to also evaluate the area under the PR curve (PR AUC), where larger area corresponds to better performance. See Figure 2.8 for visualization of PR curves.
Figure 2.8. Simulated examples of PR curves showing the relation between precision and recall on varying thresholds.

2.5.5 Cross-Entropy

In order for LR and ANN to iterate, they need to know how to validate a prediction. For this, the loss function is implemented. The loss function is in fact the function that the algorithms will minimize. In addition, the loss function needs to measure classification error and also be differentiable in order for the model to decide whether it should increase or decrease a certain weight. A commonly used loss function that fulfills these criteria is the cross-entropy measure. The general formula for the weighted cross-entropy (WCE) is:

\[
WCE = - \sum_{n}^{N} \sum_{k}^{c} v_k t_k \ln y_n
\]  

(28)

where \( n \) represents which data in the population size \( N \) is validated, \( k \) is the output neuron, \( c \) is the number of classes, \( t \) is the true outcome, \( y \) is the predicted outcome and \( v \) is the weight corresponding to the outcome. In the case of a binary one-class
response variable the formula will simplify to Equation (29).

\[
WCE = - \sum_{n}^{N} \left[ v^n t^n \ln y^n + (1 - v^n)(1 - t^n) \ln(1 - y^n) \right] (29)
\]

The weights \( v \) for WCE in this study will penalize the minority class errors with the ratio \( \frac{\text{Nr of majority classes}}{\text{Nr of observations}} \).

Since the models are trained on balanced data, LR and ANN will use a equal-weighted cross-entropy (CE) as the loss function. Therefore, Equation (29) simplifies to Equation (30).

\[
CE = - \sum_{n}^{N} \left[ t^n \ln y^n + (1 - t^n) \ln(1 - y^n) \right] (30)
\]

Both weighted cross-entropy and cross-entropy will be compared among models in this study.

2.5.6 Ease of Implementation

When working with advanced mathematical models, it is important how advanced the models are in theory and how difficult it is to transfer knowledge regarding the algorithms behind the models. This measure in its nature is rather difficult to define and quantify but Chater and Vitányi (2003, p. 19) described it as:

"A long tradition in epistemology, philosophy of science, and mathematical and computational theories of learning argues that patterns 'should' be chosen according how simply they explain the data."

2.5.7 Flexibility of Model

The data will vary depending on which country and what bank the portfolios are purchased from. This calls out for flexibility in the models. A model that easily can be configured to be used with new variables, more data or similar alterations will be considered as an advantage.
2.5.8 Stability of Model

A model that is able to generate overall consistent good results each time it is run would be preferred. Since this study models on one single portfolio, the focus will be to compare each model with regards to consistency between runs on the same data.

2.5.9 Computational Intensiveness

As the amount of data available grows larger, computational intensiveness becomes an increasingly important aspect when modelling (Bottou, 2010). In general, a more precise model is more attractive than a less precise model but as data grows larger there is a need to have balance between accuracy and computational intensiveness, which translates into the time taken to run the models. Webb, Pazzani, and Billsus (2001) provided general research on the subject of machine learning critique as the majority of research try to improve accuracy on small data sets rather than working with improving efficiency on large data sets. They argue that it becomes a gap between research and the real world application.

In this study, computational intensiveness will be taken into regard when evaluating the models which partly overlaps with ease of implementation (see Section 2.5.6).

2.6 Sampling Methodology

When modelling imbalanced data using machine learning, results tend to improve when letting the algorithms train on balanced data (Japkowicz, 2000; X. Zhang & Li, 2011). There are several ways to make the training data balanced. Such as making the majority class (non-paying debtors in this study) of the same size as the minority class (paying debtors in this study). The most frequent sampling methods are:

- **Under-sampling:** Randomly eliminates majority class until the majority class is of the same size as the minority class. Also known as down-sizing.
• **OVER-SAMPLING:** Randomly resamples the minority class until the minority class is of the same size as the majority class

• **COMBINATION:** A combination of under-sampling and over-sampling, where the minority class is randomly over-sampled and the majority class is randomly under-sampled.

In addition to the methods above, several alternatives have been introduced. Chawla, Bowyer, Hall, and Kegelmeyer (2002) presents a method called Synthetic Minority Over-sampling Technique (SMOTE) which over-samples the minority class but with an alteration. Instead of just resampling existing minority data points, the over-sampling is done by taking the difference between the data points’ features that is to be resampled and its $k$ nearest neighbors’ features (of the minority class) and multiplying this difference by a random number $(0,1)$.

There is no general sampling method that is overall superior to others (Barandela, Valdovinos, Sánchez, & Ferri, 2004). A study conducted by Drummond and Holte (2003) shows that under-sampling gave better results in their case while a study by Japkowicz and Stephen (2002) presented results that advocated for over-sampling. The conclusion of the mixed results is that the choice of sampling method should be made with regard to class imbalance severeness, data complexity, data size and quality of variables (Japkowicz & Stephen, 2002).
3 Data and Methodology

3.1 Confidentiality

This study was conducted in collaboration with a company in a competitive industry. Details regarding the data and treatment of the data has therefore been limited due to confidentiality.

3.2 Overview

This study used data from an NPL portfolio with debtors from one of the largest banks in the Philippines, which is one of the largest economies in Southeast Asia. The data consisted of information provided by the bank after the portfolio purchase.

In order to create the response variable (classifying debtors as paying or non-paying) data were collected from Collectius’ daily operations. The model predictions was however exclusively based on data provided by the banks after purchase. The idea behind choosing this distinction was making the company able to predict debt recovery from the date of purchase and thereby enabling instant prioritizing of whom to contact in order to maximize effectiveness and profitability.

3.3 Portfolio Characteristics

The main characteristic to consider when it comes to NPL portfolios is the nature of imbalanced data. Even though these collection companies are specialized in collating these close to default debts, the success ratio of getting in contact, setting up an installment plan and actually receiving payments is rather low.

3.4 Response Variable Definition

The response variable is binary and categorized as paying and non-paying debtors. The definition of the response variable is built on two criteria. The first criterion is that the debtor has made an agreement to transfer money to the collection company.
The second criterion is that the payment has to be made within a certain time frame from that agreed date. If these two criteria are met, the debtor is categorized as paying.

### 3.5 Model Assumptions

It is important that the data and model settings meet the assumptions in order to assure that the model outcomes are appropriately interpreted. For LR, the assumptions mentioned by Stoltzfus (2011) are met on all points except for multicollinearity among a minority of the input variables. However, by letting the GA select optimal variable sets, a big part of the highly correlated variables are removed and only the best ones are kept in the LR setup.

For RF, there are no assumptions to be made regarding the distribution of the data. The only generally applicable rule is the non-independence among explanatory variables (Dormann et al., 2013). In line with LR, GA is suggesting to keep only a few of the highly correlated variables.

When it comes to ANN, it has no thoroughly researched general assumptions. It is said to be fairly insensitive to problems of multicollinearity among the input variables (De Veaux & Ungar, 1994) but because of the few cases of harmful consequences it can be useful to treat it (Sharkey, 1999). In line with the previous models, GA is selecting a minority of the highly correlated variables.

### 3.6 Variable Selection and Feature Engineering

A total of 38 explanatory variables were used as input variables in the models. 17 of these were variables directly available from the data and 21 of these were created by feature engineering, for example treating or combining variables. The feature engineering conducted was for example bucketing of variables and taking quotients or sums of variables. Approximately 35% of the variables are debt related, 25% are person related, 15% are payment related and the remaining 25% are quotas between the different main groups.
3.7 Outlier Treatment

The data are imbalanced by nature. Therefore, the outlier treatment must be conducted with care. Something that might be considered as outlier tendencies could very well be part of explanatory factors to the imbalanced data.

With that in mind, the raw data were cleaned from abnormalities where the data were obviously corrupt and debtors with corrupt data were excluded from the analysis.

3.8 Sampling of Data

The data were initially split into two parts. 80% of the data were sampled as training data from which the models were built. The remaining 20% were sampled as test data which were used to validate the models’ predictability. The split was conducted using stratified random sampling such that the proportion of the response variable was the same in each sample.

Thereafter, the training data were sampled using the different methods presented in Section 2.6. The under-sampling method, where non-paying debtors are randomly eliminated until being equal in numbers to paying debtors, showed the most promising outcomes after the initial evaluation and was therefore chosen to be the sampling method. Since the size of the original data set was rather large, the drawbacks with removing data points are limited.

3.9 Feature Selection Procedure

The data used in the models were treated using the feature engineering presented in Section 3.6 and the outlier treatment mentioned in Section 3.7.

All models take advantage of the benefits of GA which finds a close to optimal combination of variables based on the chosen optimize criterion, see Section 2.4.1 for further information regarding GA.

For RF specifically, MDG (Section 2.4.2) could be used to rank importance among variables (Han et al., 2016). When having a large number of variables this could be
useful to decrease the number of variables used in the model with low risk of losing important information. MDG was used for this purpose and to get an enhanced understanding of the data and the relative importance of the variables. In line with literature mentioned in Sections 2.2.1 and 2.2.2, $n_{\text{tree}}$ and $m_{\text{try}}$ were set to 2000 and 6 respectively.

For ANN specifically, it was stated in Section 2.3.4 that it is efficient to use one layer in the most cases. In addition, with the number of input variables in this study, Equation (18) is computed to approximately four. When taking aspects mentioned in Section 2.3.3 into regard, a suitable stopping threshold was concluded to be 0.1. Due to this, the study used one layer and four nodes with a threshold of 0.1 in the ANN setup.

### 3.10 Scaling

To make the data suitable for LR and ANN it can be useful to have the same scale on the input variables. The input variables in this study was rescaled using standardization:

$$x' = \frac{x - \mu_x}{\sigma_x} \tag{31}$$

where $x'$ is the standardized value of $x$, $\mu_x$ is the mean and $\sigma_x$ is the standard deviation. The scaling gives all variables a mean of zero and unit standard deviation while preserving the relationship to the response variable. RF did not use a standardization since it doesn’t change the model performance since the data splitting in RF only depend on the order of the observed values and not its scale.

### 3.11 Programming Packages

For the models used in this study, version 3.5.2 of the programming language R has been used. The predictions have been modelled with the theory mentioned in Section 2 and by using R packages based on the frameworks contained in that Section.

For LR, the function `glm` from the R package `stats` version 3.5.3 was used where details regarding the algorithm are presented in Section 2.1.
RF was implemented by using the R package *randomForest* version 4.6 which is based on the research conducted by Breiman (2001). The details regarding the model and the settings are presented in detail in Section 2.2.

For ANN, the R package *neuralnet* version 1.33 was used based on the study conducted by Riedmiller (1994) and explained thoroughly in Section 2.3 together with the relevant settings.

The variable selection with GA has used the R package *GA* version 3.2 with its default settings regarding probabilities for its procedure for selection, crossover and mutation (Scrucca, 2013b).

### 3.12 Limitations

A limitation with the data analyzed was that it consisted of just one type of loans (credit card loans) from one bank with debtors from the same country which affects generalizability.

Another limitation was that the portfolio was still active when the models were implemented, meaning that debtors who at the time of analysis was categorized as non-paying could be recategorized in the future. A completely closed portfolio could have the benefit of being final in its categorization. However, with minor alterations to current design of the model, modelling an active portfolio could be beneficial since this enables the model to be used as a continuous tool in the day-to-day business, see Section 5.5 for further elaboration regarding implementation.
4 Results

4.1 Variable Selection

For RF, a visual inspection of the OOB error was used to evaluate how many trees that were needed in the upcoming model evaluations. Figure 4.1 shows that around 100 trees are adequate for the OOB error to approximately converge and therefore the same number of trees was used when performing variable selection.

![Figure 4.1. OOB error progress for RF with all variables used in model.](image)

For ANN, it was found that a suitable stopping criterion for the model was when the partial derivatives of the error function reached as low as 3. This stopping criterion will therefore be used when performing variable selection due to aspects mentioned in Section 2.3.3 but the optimal models are evaluated after a stopping criterion of 0.1.

Each of the methods are ran in GA in order to select the optimal set of variables for the given optimisation objective. In Sections 4.1.1 and 4.1.2 the selection evolutions are explained in more detail. For RF, MDG was used in addition to GA to investigate whether this method could perform superior variable selection. Initial testing showed that GA generated more promising results which is why this method was used onwards.
4.1.1 Optimizing ROC AUC

For all models, GA started with an initial guess of all variables.

LR and RF used a population of 50 chromosomes in the algorithm and ANN used a population of 20 chromosomes. LR, RF and ANN ended up selecting 31, 25 and 21 variables respectively for optimizing its ROC AUC measure.

The stepwise improvement for each model is presented in Figure 4.2.

![Figure 4.2. Selection evolution using GA with ROC AUC as an optimization objective. LR to the left, RF in the middle and ANN to the right.](image)

4.1.2 Optimizing PR AUC

For all models, GA started with an initial guess of all variables.

LR and RF used a population of 50 chromosomes in the algorithm and ANN used a population of 20 chromosomes. LR, RF and ANN ended up selecting 27, 26 and 24 variables respectively for optimizing its PR AUC measure.

The stepwise improvement for each model is presented in Figure 4.3.

![Figure 4.3. Selection evolution using GA with PR AUC as an optimization objective. LR to the left, RF in the middle and ANN to the right.](image)
4.2 Performance Metrics

The set of variables that provide the highest fitness value (from Sections 4.1.1 and 4.1.2) are selected for each model with respect to the different optimization objectives, respectively. The following performance metrics shows results for each optimal set and also the model using all variables.

Overall, the results go in line with what is expected of rather well-performing models. There is however limited comparability with previous studies because of the pioneering mix of NPL and machine learning in combination with the novelty of the performance metrics when it comes to evaluation and comparison of machine learning models (Provost et al., 2001).

4.2.1 ROC Curve

In this section the models’ ROC curves are presented. The colors throughout the curves correspond to a certain threshold stated in the bar to the right of each figure. The results are varying marginally among the models and different variable selections, as seen in Figures 4.4, 4.5 and 4.6 where LR is to the left, RF in the middle and ANN to the right.

*Figure 4.4.* ROC curves for the models with all variables included.
4.2.2 PR Curve

In this section the models’ PR curves are presented. The color bars correspond to a certain threshold stated in the bar to the right of each figure. Just as the ROC curves, the PR curves only vary marginally between the models and different variable selections, as seen in Figures 4.7, 4.8 and 4.9 where LR is to the left, RF in the middle and ANN to the right.

Figure 4.5. ROC curves for the models with variables corresponding to their optimal ROC AUC.

Figure 4.6. ROC curves for the models with variables corresponding to their optimal PR AUC.
Figure 4.8. PR curves for the models with variables corresponding to their optimal ROC AUC.

Figure 4.9. PR curves for the models with variables corresponding to their optimal PR AUC.

4.2.3 Density

The classification performance for each model and variable selection is visualized with a density function, presented in Figures 4.10, 4.11 and 4.12, where LR is to the left, RF in the middle and ANN to the right. It has the constrained real-value predictions on the horizontal axis and the density of each prediction on the vertical axis. The density function is grouped on paying and non-paying debtors and therefore relative to each groups’ absolute size. The dotted line represents the chosen cut-off point (from Section 2.5.1) for each model. Once again, the results are in general rather homogeneous with some minor dissimilarities.

Figure 4.10. Density functions for the models predictions grouped on actual outcome. All variables included.
4.2.4 Confusion Matrix

When applying the chosen cut-off point (from Section 2.5.1), the confusion matrices in the Tables 4.1, 4.2 and 4.3 are generated. The number in the upper left corner represents the percentage of the validation data that are TN, the upper right holds the percentage of FP while the lower left and right corner presents the percentage of FN and TP respectively.

**Table 4.1**

*Confusion matrices for the models with all variables included.*

<table>
<thead>
<tr>
<th></th>
<th>LR</th>
<th>RF</th>
<th>ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Predicted</td>
<td>Predicted</td>
<td>Predicted</td>
</tr>
<tr>
<td></td>
<td>NP  P</td>
<td>NP  P</td>
<td>NP  P</td>
</tr>
<tr>
<td>Actual</td>
<td>NP 67.7 22.1</td>
<td>NP 69.9 19.9</td>
<td>NP 71.1 18.7</td>
</tr>
<tr>
<td></td>
<td>P  2.61 7.60</td>
<td>P  2.65 7.56</td>
<td>P  2.87 7.33</td>
</tr>
</tbody>
</table>
Table 4.2

Confusion matrices for the models with variables corresponding to their optimal ROC AUC.

<table>
<thead>
<tr>
<th></th>
<th>Predicted</th>
<th>Actual</th>
<th></th>
<th>Predicted</th>
<th>Actual</th>
<th></th>
<th>Predicted</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NP</td>
<td></td>
<td></td>
<td>NP</td>
<td></td>
<td></td>
<td>NP</td>
</tr>
<tr>
<td>LR</td>
<td></td>
<td>68.3</td>
<td>21.5</td>
<td></td>
<td>68.9</td>
<td>20.9</td>
<td></td>
<td>71.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.65</td>
<td>7.55</td>
<td></td>
<td>2.48</td>
<td>7.73</td>
<td></td>
<td>2.62</td>
</tr>
<tr>
<td>RF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NP</td>
<td></td>
<td></td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>68.9</td>
<td>20.9</td>
<td></td>
<td>68.8</td>
<td>21.0</td>
<td></td>
<td>72.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.82</td>
<td>7.39</td>
<td></td>
<td>2.54</td>
<td>7.67</td>
<td></td>
<td>2.71</td>
</tr>
<tr>
<td>ANN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P</td>
<td></td>
<td></td>
<td>P</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.62</td>
<td>7.58</td>
<td></td>
<td>2.71</td>
<td>7.50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3

Confusion matrices for the models with variables corresponding to their optimal PR AUC.

<table>
<thead>
<tr>
<th></th>
<th>Predicted</th>
<th>Actual</th>
<th></th>
<th>Predicted</th>
<th>Actual</th>
<th></th>
<th>Predicted</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NP</td>
<td></td>
<td></td>
<td>NP</td>
<td></td>
<td></td>
<td>NP</td>
</tr>
<tr>
<td>LR</td>
<td></td>
<td>68.9</td>
<td>20.9</td>
<td></td>
<td>68.8</td>
<td>21.0</td>
<td></td>
<td>72.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.82</td>
<td>7.39</td>
<td></td>
<td>2.54</td>
<td>7.67</td>
<td></td>
<td>2.71</td>
</tr>
<tr>
<td>RF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NP</td>
<td></td>
<td></td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>68.8</td>
<td>21.0</td>
<td></td>
<td>68.8</td>
<td>21.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.82</td>
<td>7.39</td>
<td></td>
<td>2.54</td>
<td>7.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P</td>
<td></td>
<td></td>
<td>P</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.62</td>
<td>7.58</td>
<td></td>
<td>2.71</td>
<td>7.50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2.5 Ratios

The relevant metrics for performance comparison among the models are presented in Table 4.4. All measures are computed for the models using the optimal set of variables for each optimization objective.
Table 4.4

Performance metrics for the models.

<table>
<thead>
<tr>
<th>Ratios</th>
<th>LR All</th>
<th>LR RA*</th>
<th>LR PA**</th>
<th>RF All</th>
<th>RF RA*</th>
<th>RF PA**</th>
<th>ANN All</th>
<th>ANN RA*</th>
<th>ANN PA**</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROC AUC</td>
<td>.833</td>
<td>.835</td>
<td>.831</td>
<td>.839</td>
<td>.841</td>
<td>.841</td>
<td>.839</td>
<td>.852</td>
<td>.846</td>
</tr>
<tr>
<td>PR AUC</td>
<td>.501</td>
<td>.500</td>
<td>.509</td>
<td>.522</td>
<td>.521</td>
<td>.526</td>
<td>.519</td>
<td>.537</td>
<td>.527</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>.744</td>
<td>.740</td>
<td>.724</td>
<td>.740</td>
<td>.757</td>
<td>.751</td>
<td>.719</td>
<td>.743</td>
<td>.735</td>
</tr>
<tr>
<td>Specificity</td>
<td>.754</td>
<td>.760</td>
<td>.768</td>
<td>.778</td>
<td>.768</td>
<td>.766</td>
<td>.792</td>
<td>.799</td>
<td>.801</td>
</tr>
<tr>
<td>Accuracy</td>
<td>.753</td>
<td>.758</td>
<td>.763</td>
<td>.774</td>
<td>.767</td>
<td>.764</td>
<td>.785</td>
<td>.794</td>
<td>.795</td>
</tr>
<tr>
<td>Precision</td>
<td>.256</td>
<td>.260</td>
<td>.261</td>
<td>.275</td>
<td>.270</td>
<td>.267</td>
<td>.282</td>
<td>.296</td>
<td>.296</td>
</tr>
<tr>
<td>Negative Predictive Value</td>
<td>.963</td>
<td>.963</td>
<td>.961</td>
<td>.963</td>
<td>.965</td>
<td>.964</td>
<td>.961</td>
<td>.965</td>
<td>.964</td>
</tr>
<tr>
<td>Balanced Accuracy</td>
<td>.749</td>
<td>.750</td>
<td>.746</td>
<td>.759</td>
<td>.762</td>
<td>.759</td>
<td>.755</td>
<td>.771</td>
<td>.768</td>
</tr>
<tr>
<td>Cohen’s Kappa</td>
<td>.270</td>
<td>.275</td>
<td>.275</td>
<td>.296</td>
<td>.292</td>
<td>.287</td>
<td>.303</td>
<td>.325</td>
<td>.324</td>
</tr>
<tr>
<td>Cross-Entropy</td>
<td>0.487</td>
<td>0.487</td>
<td>0.486</td>
<td>0.480</td>
<td>0.480</td>
<td>0.480</td>
<td>0.473</td>
<td>0.458</td>
<td>0.462</td>
</tr>
<tr>
<td>Weighted Cross-Entropy</td>
<td>0.091</td>
<td>0.091</td>
<td>0.091</td>
<td>0.089</td>
<td>0.089</td>
<td>0.089</td>
<td>0.089</td>
<td>0.086</td>
<td>0.088</td>
</tr>
</tbody>
</table>

*RA = Optimized on ROC AUC, **PA = Optimized on PR AUC
5 Discussion

5.1 General

The main research objective of this study was to compare the ability of RF and ANN to correctly classify people in an NPL portfolio while using LR as a benchmark. In brief, this study shows that the three models have similar performance while there are pros and cons as well. The overall performance is satisfactory and the models show real potential in being able to be implemented in day-to-day business.

A general aspect to have in mind when evaluating the results is the effect of macroeconomic factors. It is challenging to derive causal conclusions from macroeconomic factors but it is important to illuminate the fact that there has been an economic boom in Southeast Asia during the last decade (Edmonds et al., 2018; The World Bank, 2017). People have been better at repaying loans and since the modelling in this thesis is made on such data from a period of one year, the model might be worse if a recession were to occur.

5.2 Non-Performing Loan Portfolio Data

The portfolio analysed in this study was imbalanced, which is the nature of NPL portfolios in general. This is something to be taken into account when working with data preparation and methodology. Models insensitive to outliers (or extreme values) could be preferable since this enables less cleaning of data. Less cleaning decreases the risk of removing explanatory parts of an imbalanced data set. If working with models that are sensitive to imbalanced data, a recommendation would be to use an insensitive model as a benchmark.

Another aspect is that the portfolio analysed in this study contained many data points, the number of variables accessible were however limited partly due to confidentiality. When working with a limited number of variables, feature engineering becomes extra important. Firstly, feature engineering could help the models finding patterns which otherwise would be missed. Secondly, it enables the researcher to guide the models by giving them more variables related to some certain area that might be of particular
interest.

It is also important to highlight the fact that the validity of the models could be weak on other NPL portfolios. The results in this study shows satisfactory performance when validating on data from the same portfolio and time frame. To improve model validity for another purpose, the researcher should validate them using data fulfilling this purpose.

5.3 Properties of the Models

LR has a fast convergence and stable performance. It does not change much when selecting the optimizing sets of variables which could be a useful property when validating future portfolios lacking certain variables. This model has been used mainly as a performance benchmark, providing insights in whether the other two more sophisticated models generate better results than a traditional one.

RF is known for being easy to use and implement. One of the reasons for this is that the model tends to perform well without extensive data preparation such as feature engineering and feature selection. The results in this study supports this. RF tends to generate low variability in results when varying the number of trees (above a certain lower limit, see Section 2.2.1), number of variables chosen at each node and even when providing the model with data that is diluted with noise.

Even though there is no general rule when choosing the number of variables, randomly chosen at each node, as stated in Section 2.2.2, several studies would suggest $m_{try}$ in this study to be equal to five or six based on the total number of variables in the model, which varies between 25 and 40 (Breiman, 2001; Friedman et al., 2001; Khoshgoftaar et al., 2007). However, slightly better results were generated when choosing $m_{try}$ to be two or three. Worth noting is that the differences in results are small in general, which is in line with the previous statement regarding the overall consistency of RF (Breiman, 2001).

For ANN, the performance is powerful in many aspects. The downside is that it lacks in speed and is also hard to interpret in terms of the weight relations in combination with the activation function in the network. It also performs well when using different numbers of layers and nodes which indicates that a thorough analysis of the network
setup is not very meaningful. For one hidden layer, one of the best performing node settings was suggested by the function given in Equation (18) even though other node settings provided almost the same performance. In terms of variables selected it performs well since it includes a smaller amount of variables than LR and RF when optimizing ROC AUC and PR AUC.

### 5.4 Comparison of Model Performance

In all the density plots in Section 4.2.3, it can be seen that all classifiers are splitting the observations well. All the models generates graphs that could arguably be interpreted as having a relatively strong representation of prediction power for some characteristics of the paying debtors while the rest of the data set has a somewhat weaker prediction power.

An important note for companies in the collection industry could be the precision ratio which is non-overlapping among the models with ANN performing the best and LR the worst. In these circumstances it means that given a prediction threshold that maximizes the sensitivity and specificity, ANN will predict the most of the actual paying debtors as paying debtors. This can be compared with the overall balanced accuracy ratio which does not have a clear over-performer among the models even though the variable selection for optimizing ROC AUC has a better performance than the other sets of variables.

In terms of Cohen’s Kappa the models provided a clear non-overlapping performance range for the three models regardless of which of the given variable sets that were taken into account. ANN places in top followed by RF. The fact that ANN generates highest Cohen’s Kappa suggests that this model is most promising when it comes to predicting debtors correctly, adjusting for the possibility that correct predictions were made by chance.

LR has weaker ROC AUC and PR AUC measures than the other models regardless of which optimized set of variables were taken into account. When all variables were taken into account, RF and ANN performed similarly on ROC AUC, but RF had a better PR AUC measure. After the variable selection ANN performed slightly better on these measures.
Another detail worth mentioning is that by optimizing variables for PR AUC, RF will generate the same ROC AUC as if the variables for optimizing ROC AUC were used. This suggests that a researcher who only wants a model with high ROC AUC and PR AUC could optimize on PR AUC. One should however be aware that other ratios could suffer from this approach.

LR and RF do not change in either cross-entropy or weighted cross-entropy when selecting only a subset of all variables, which is a good sign for the potential problems caused by lack of data in usage of the models later on. Even though cross-entropy is the function that LR minimizes, it is weaker than for the RF and ANN. The other two models’ weighted cross-entropy is the same even though ANN is optimizing on it while RF is not. In terms of speed of running each model, LR and RF outperforms ANN tremendously. This is due to poor convergence for its error function.

ANN provides a relatively high PR AUC when GA selects optimal variables. When ANN uses these variables independently it ends up generating a PR AUC that is lower than when GA was running it. This is in line with the instability properties of the ANN model in this study.

### 5.5 Application Fields

The user of these models could apply them as a guideline when prioritizing who to contact. The ones who are categorized as more likely to pay, should preferably be contacted first. The modelling also enables an overview of how the current portfolio is structured, an approximation of how much is expected to be collected.

Elaborating on the implementation possibilities, this type of modelling could also be used before purchase of a portfolio from a bank. By categorizing debtors before buying, an approximate portfolio value could be generated.

### 5.6 Conclusions

When comparing the results of the three different models, they are all performing well. The differences in results are rather small and none of the models are unequivocally an under or overperformer. In terms of the main measures optimized upon, ANN
performed the best but not with a big marginal to RF. In terms of speed and performance stability LR and RF is performing the best while ANN has slow convergence. RF is more beneficial when it comes to flexibility due to its nature of handling categorical variables and lack of scaling dependency.

5.7 Future Studies

An interesting approach for future studies would be to develop a credit score model resulting in more than a binary classification. This would enable a more flexible categorization and implementation. Another alteration to the modelling approach could be to develop a model that is continuously updated as time passes which allows more data to be collected and potentially lead to a more precise classification.

Furthermore, modelling on more than one NPL portfolio, from one bank operating in one country could yield new findings and make it possible to draw conclusions of another magnitude regarding predictability. This would most likely strengthen the generalization, with the drawback of not being able to predict the specific country, bank or portfolio with the same precision.

Finally, modelling with machine learning methods is a never ending project. Improvements, alterations and tweaking of the models presented in this study could yield further findings. The models could for example be developed focusing on optimization of other performance metrics or using another response variable. In addition to this, as new models enter the field it would be of interest to benchmark the performance of these with the ones used in this study.
References


Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one, 10*(3), e0118432.


