# The Effect of Dataset Size on the Performance of Classification Algorithms for Credit Scoring

By
Ben Gidlow
GDLLIA001

Master's Dissertation

Department of Finance and Tax, Faculty of Commerce

February 2022

Supervisor: Darron West

# COMPULSORY DECLARATION

**1.** This dissertation has been submitted to Turnitin and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

**2.** This work has not previously been submitted, in whole or in part, for the award of any degree at this or any other university. It is my own work. Each significant contribution to, and quotation in, this dissertation from the work, or works, of other people has been attributed, and has been cited and referenced.

| | |
|---|---|
| **Student Number** | GDLLIA001 |
| **Student Name** | Ben Gidlow |
| **Student Signature** | Signed by candidate |
| **Date** | **02/02/2022** |

# Contents

## Abstract

In the body of research on predicting borrower default in the field of credit risk, the relative performance of different classification algorithms has received much attention by researchers. With the rise of machine learning techniques spurred on by fast, cheap computing and the generation and collection of massive datasets, there has been a great deal of research benchmarking the performance of these powerful machine learning algorithms against traditional techniques used in credit scoring.

This dissertation extends the research into the benchmarking of different classification algorithms by looking to establish the effect that the size of a training dataset has on the relative performance of different algorithms. This paper conducts a thorough review of the literature on credit risk prediction to establish the methodologies and findings prevalent in the literature. It finds that most of the research on the subject has been conducted using very small datasets to benchmark algorithm performance on. Furthermore, due to the relatively low number of publicly available credit risk datasets available, most of the research is conducted on the same datasets. This paper uses five publicly available credit risk datasets, on which six algorithms are evaluated. The algorithms evaluated in this paper are logistic regression, random forests, neural networks, gradient boosting machines, extreme gradient boosting, and stacked ensembles. This paper conducts two separate analyses on these datasets and algorithms – firstly, a general analysis, where algorithm performance is benchmarked by average relative performance over all datasets, in line with the conventions established by other researchers prevalent in the literature. Secondly, a learning curve analysis established by researchers in a different field is utilised – this technique, which involves training algorithms on subsamples of different sizes of a large dataset, allows us to investigate the ability of different algorithms to leverage additional data to improve prediction performance.

The general analysis affirms the relative outperformance of the stacked ensemble algorithms relative to the other algorithms evaluated – a finding prevalent in the body of literature. It also establishes evidence for the effect of dataset size on the relative performance of different algorithms. The learning curve analysis shows definitively that the more sophisticated classification algorithms - random forests, neural networks, gradient boosting machines, extreme gradient boosting and stacked ensembles – are able to consistently improve performance with more data available to train on. Logistic regression, by contrast, reaches a performance plateau at

a relatively small training dataset size. Given the prevalence of small datasets in the literature, this finding implies that existing research might overemphasise the performance of logistic regression relative to more sophisticated algorithms.

## List of Abbreviations Used

| | |
|---|---|
| PD | Probability of default |
| EAD | Exposure at default |
| LGD | Loss given default |
| CART | Classification and regression trees |
| ANN | Artificial neural network |
| SVM | Support vector machine |
| LR | Logistic regression |
| ACC | Average correct classification (accuracy) |
| TP | True positive |
| TN | True negative |
| FP | False positive |
| FN | False negative |
| K/S statistic | Kolmogorov-Smirnov statistic |
| ROC curve | Receiver operating characteristic curve |
| AUC | Area under the ROC curve |
| AUC_PR | Area under the precision-recall curve |
| UCI | University of California Irvine |
| CV | Cross-validation |

# Chapter 1
# Introduction

## *1.1 Problem Description*

With the rise and rise of artificial intelligence and machine learning as computers become ever cheaper and more powerful, the adoption of sophisticated modelling techniques in financial services continues to gain momentum. Within the credit industry, there are clear benefits to leveraging technology to improve predictions of borrower behaviour; if lenders are able to predict which borrowers will not repay a loan more accurately, their overall cost of risk decreases. This results, all else equal, in higher profits for lenders. In a competitive environment, a lower cost of risk allows lenders to charge lower interest rates to qualifying customers, making credit cheaper and stimulating economic activity. Additionally, better predictive models that are capable of leveraging big data and alternative data more effectively than traditional modelling techniques allow for the expansion of credit access to previously unserved customers; the so-called "financially excluded". As a result of the clear benefits derived from improving the accuracy of predictions in credit risk, a great deal of research has been conducted into establishing the relative benefits of new classification algorithms on credit model prediction problems.

The body of literature on the benchmarking of different predictive algorithms' performance on credit risk datasets highlights the gains in predictive performance of advanced statistical techniques relative to more traditional techniques. However, relatively little research has been carried out in the field of credit risk on the contexts in which the results of these studies are obtained - in other words, the characteristics of the data sets that are used in the studies, and whether these characteristics are an important factor in the performance of certain algorithms. This is a notable gap in existing research, as the performance of different algorithms have been shown to be dependent on the characteristics of the datasets on which they are trained in different fields of research.

This paper builds on the body of work on the use of modern statistical techniques in credit scoring through an investigation of the performance of various algorithms on credit risk datasets. It shall first establish the existing paradigm of research in the field by conducting a review of the literature on the benchmarking of classification techniques for credit default prediction. It shall provide a brief introduction into the various prediction problems within the field that require predictive modelling, and the specific prediction problem that is the subject of this research. This study shall focus on probability of default (PD) modelling – the estimation of a customer's probability of default at the point of application for a loan – as the modelling problem on which algorithms are applied. It shall outline the various components of the credit default prediction modelling problem that are prominent in the literature: feature selection, performance measures, sampling and datasets, and the various classification algorithms that feature in the literature. It shall look to build on previous benchmarking studies on the performance of different classification techniques on loan default prediction by first benchmarking the performance of different classification algorithms on a set of credit risk datasets according to defined performance metrics. It shall aim to extend previous research, which sought to answer the question as to which modelling technique, on average, generates the best predictions. It shall do so through an investigation of the specific characteristics of a dataset that influence the performance of a classification technique, rather than simply which algorithms perform the best, on average, over a range of datasets.

## *1.2 Background*

Whilst the concept of statistical credit scoring is relatively modern, that of credit itself has been around for a very long time. The earliest recorded mentions of credit can be traced to the time of Hammarubi, the king of Babylonia circa 1726 B.C. The Code of Hammarubi, one of the oldest deciphered artefacts of writing in the world, consists of the code of laws for ancient Mesopotamia at the time of Hammarubi's rule. It contains references to many concepts relevant to ancient banking institutions, including regulations pertaining to loans, guarantees, and interest (MacDonald & Gastmann, 2017). Given the relative complexity of these concepts compared to credit in its simplest sense – of borrowing something now and returning it later – we can reasonably assume that credit itself has existed for far longer. Beginning with the simple concept of farmers borrowing to plant crops and repaying their debt once they harvest, credit has become the lifeblood of the global economy, its importance highlighted by its role in two major crises preceding the Coronavirus pandemic: the European debt crisis, and the global financial crisis of 2007. The global financial crisis, brought on by the collapse of the subprime credit market, is an example of the importance of proper credit risk management in the consumer credit market (Crotty, 2009). In order to manage credit risk effectively, lenders require accurate estimates of customers' creditworthiness. The practice of estimating the creditworthiness of potential credit customers has made great strides from its origins in the individual assessment of applicants by credit agents; modern consumer credit markets utilise advanced statistical methods that take advantage of powerful computers to leverage massive datasets to find good predictors of creditworthiness (Abdou & Pointon, 2011).

### *1.3 Aims of Research*

This paper shall build on existing research that compares the performance of various algorithms on predicting default across a range of credit risk data sets, typically reporting results as average relative performance across all datasets utilised. It shall first establish equivalence with existing research by benchmarking the average performance of different classification algorithms across five credit risk data sets of various sizes. It shall then expand on existing research by seeking to answer the question as to whether the size of the datasets are a factor in the relative performance of different classification algorithms.

This paper shall present two distinct research methods – a benchmarking analysis in line with the methodologies generally used by existing studies in the literature, and a technique called "learning curve analysis" hitherto not utilised by researchers in the field of credit risk prediction to the knowledge of this author.

The benchmarking analysis referred to in this paper as "general analysis" seeks to affirm established findings in the literature on the relative performance of different classification algorithms in probability of default prediction. It does so by training a set of six different classification algorithms, selected for their prominence in recent studies in the literature, on five different credit risk datasets. It shall evaluate the performance of the algorithms on each dataset using a set of performance metrics chosen to represent a range of different scenarios in which a probability of default model is likely to be applied in a commercial setting. The general analysis on model performance is then conducted by comparing the average rank of each algorithm over all five training datasets. The general analysis conducted in this paper shall confirm the findings of Lessmann et al. (2015), that ensemble models outperform sophisticated individual classification techniques on average over a range of training datasets predicting credit risk defaults. It shall illustrate the problem of failing to take training dataset size into account in benchmarking algorithm performance, by showing how average performance changes over sets of smaller and larger datasets. It shall outline how this may lead to results that are potentially biased by a researcher's choice of datasets included in a study.

The learning curve analysis is conducted in order to establish the effect of the size of a training set on the performance of different algorithms. Utilising two large datasets, training datasets of different sizes are sampled, with models being trained on increasingly large training samples in order to determine the marginal impact of larger training samples on model performance. The performance of each model on a training sample is then plotted to create the "learning curves" for each performance metric to illustrate the effectiveness of each algorithm at leveraging more data in order to drive gains in prediction performance. The learning curve analysis, as used in Perlich, Provost and Simonoff (2003), is used to illustrate the relationship between size of training datasets and the performance of different classification algorithms. By showing the importance of dataset size in evaluating an algorithm's performance, it shall argue that the failure of much of the research that makes up the body of literature to date to take the characteristics of the datasets used in their studies into account makes their findings difficult to generalise, particularly in a commercial context in which datasets tend to be orders of magnitude larger than those that feature prominently in the literature.

### *1.4 Paper Layout*

Chapter 2 presents a review of the literature relating to the evaluation of the performance of different classification algorithms in the field of credit risk prediction. It shall establish the paradigm in which this research is conducted, and highlight the gap in existing research relating to the interaction between dataset size and algorithm performance in credit risk prediction that this paper looks to answer. Chapter 3 discusses the methodology of the research. It details the data preprocessing involved, sampling methods, classification techniques and performance measures used in this research. The results of the study, and discussion thereof, is presented in Chapter 4. Chapter 5 concludes this paper.

# Chapter 2

# Literature Review

## *2.1 Background*

Credit scoring is an application of statistically differentiating between different groups in a population, where the characteristic that differentiates the groups (in this case, whether a borrower will default on a loan) is not observable. In 1936, Ronald Fisher first introduced the concept of statistically discriminating between groups, looking to differentiate between varieties of iris, and the origins of skulls using their physical characteristics as explanatory variables (Thomas, 2000). The history of statistical credit scoring, defined as the evaluation of risk associated with lending to individuals or companies begins in 1941 (Crook, Edelman & Thomas, 2007). In 1941, David Durand conducted a study applying Fisher's method of statistical discrimination to the problem of credit risk prediction, examining loans made by 37 lending companies in the United States in order to establish their ability to identify bad credit risk using the techniques pioneered by Fisher. Commissioned by the U.S. National Bureau of Economic research in response to their observation that the consumer credit market experienced relatively small losses during the Great Depression compared to other credit markets, Durand's study established the foundations for the field of credit scoring over the subsequent decades (Durand, 1941).

Prior to the advent of the widespread use of statistical techniques for evaluating creditworthiness, credit was granted solely on the judgement of the lender, or the credit analysts acting on their behalf. Decisions were based on what is referred to as the "five C's" - the character of the applicant, the capital being requested, the collateral offered, the capacity (income available to service debt), and the conditions of the market (Thomas, 2000).

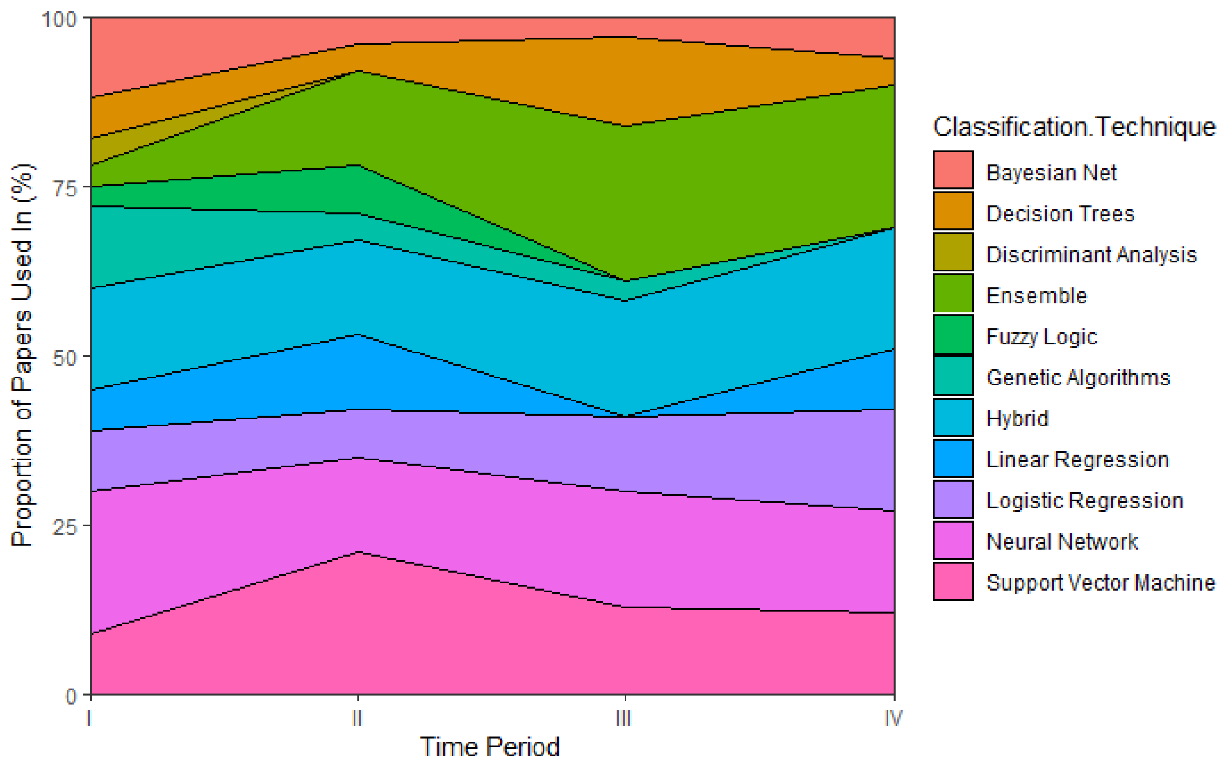
The output of credit scoring, credit scores, are estimates of the probability that a borrower will behave in a certain way if given access to a loan. To be precise, this is referred to as *application scoring*, to distinguish predictions made prior to the granting of credit access from estimates of behaviour of an agent once credit has been granted, referred to as *performance* or *behavioural* *scoring* (Hand & Henley, 1997). A large body of literature exists on the subject of predictive models used in the credit industry - statistical models that estimate the likelihood that some event will occur given a set of explanatory variables. These models are created and evaluated using historical data – i.e., data relating to loans for which the outcome is already known. The most common predictive models used in the field of credit are probability of default (PD), exposure at default (EAD), and loss given default (LGD) models. Of these, PD models are the most extensively researched (Lessmann et al., 2015). PD models can be developed as either classification or survival models: the former estimating the probability of whether or not a borrower will default on a loan, whilst the latter not only estimates if, but also when a borrower might default (Hand & Henley, 1997). This paper focuses on the problem of application scoring for PD classification - estimating the probability of the binary outcome of whether a borrower will default on a loan at a point in time before the loan is granted.

There are three fields of research in the body of literature concerning credit scoring algorithms which have received the most attention by researchers: the predictive accuracy of different algorithms, or classification techniques, feature selection, and performance metrics (Louzada, Ara & Fernandes, 2016). These fields of research are outlined in more detail in the sections to follow.

## *2.2 Classification algorithms*

Researchers in the field of credit risk prediction have evaluated the performance of many classification techniques over the years. Louzada, Ara and Fernandes (2016) conducted a systematic review of the classification techniques utilised by researchers in the literature. Reviewing 187 papers on the topic, they found that the most common techniques used by researchers were (in order from most to least often featured in reviewed papers): neural networks, hybrid methods, combined methods, support vector machines, logistic regression, decision trees, Bayesian net, linear regression, genetic algorithms, fuzzy logic, and discriminant analysis. Furthermore, they observed shifts in the popularity of these techniques over time. They categorise papers into four time periods based on their publication date - papers published before 2006 as classified as “I”, those published between 2006 and 2010 are classified as “II”, between 2010 and 2012 “III”, and after 2012 “IV”.

**Figure 1.1.1  Distribution of Classification Techniques Used in Papers on Credit Scoring by Time Period**




As can be observed in the Figure 2.2.1 above, the popularity of neural networks as a classification algorithm in credit scoring has been substantial – and this popularity has remained consistent over time, as is the case with hybrid methods. Ensemble methods have gained popularity since 2010, and feature in more papers in the most recent time period than any other technique. Support vector machines, whilst remaining prominent in the most recent time period, peaked in the 2010-2012 period, whilst use of logistic regression has been consistent since the earliest time period reviewed, increasing in the more recent time periods. Usage of discriminant analysis, fuzzy and genetic methods decreased significantly over the time periods observed, to the point that they do not feature at all in papers from the most recent time period (Louzada, Ara & Fernandes, 2016).

A short summary of the most popular classifiers featured in the literature is presented below.

### 2.2.1 Logistic regression:

Logistic regression is a statistical technique whereby the linear relationship between the logit of some binary dependent variable $y = \{0,1\}$ and a set of independent variables $x = \{x1, \dots, xn\}$ is estimated. The model may be represented thus:

$$\pi^i = \frac{exp\,\{x_i\beta\}}{1-exp\,\{x_i\beta\}} \qquad (1)$$

Where $i$ is the probability of the $i^{th}$ observation belonging to class y=1, given $x_i$ (Louzada, Ara & Fernandes, 2016). Logistic regression is one of the most widely used classification techniques in credit risk, due to the robustness, stability and accuracy of its estimations. It is also noteworthy that it has a major benefit over more advanced techniques in terms of "explainability" - credit scores generated from logistic regression models can be easily explained in a way more advanced models cannot (Abdou & Pointon, 2011).

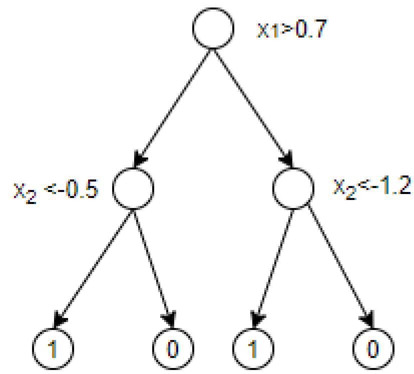
### 2.2.2 Discriminant Analysis:

Linear discriminant analysis is a statistical technique to discriminate between two groups. Introduced by Fisher (1936) and first applied to the field of credit scoring by Durand (1941), discriminant analysis was also employed by Altman (1968) for his famous Z-score to predict corporate bankruptcy (Abdou & Pointon, 2011). Discriminant analysis uses Bayes' theorem to assign an observation to the class to which it most likely belongs (Baesens et al., 2003).

### 2.2.3 Decision trees:

Also referred to as Classification and Regression Trees (CART), decision trees are built of nodes that split the records in a training dataset based on the function of a single input, or independent variable. Different sub-trees are evaluated according to a predetermined performance measure, with the 'winning' sub-tree selected as the classification technique (Abdou, 2009).

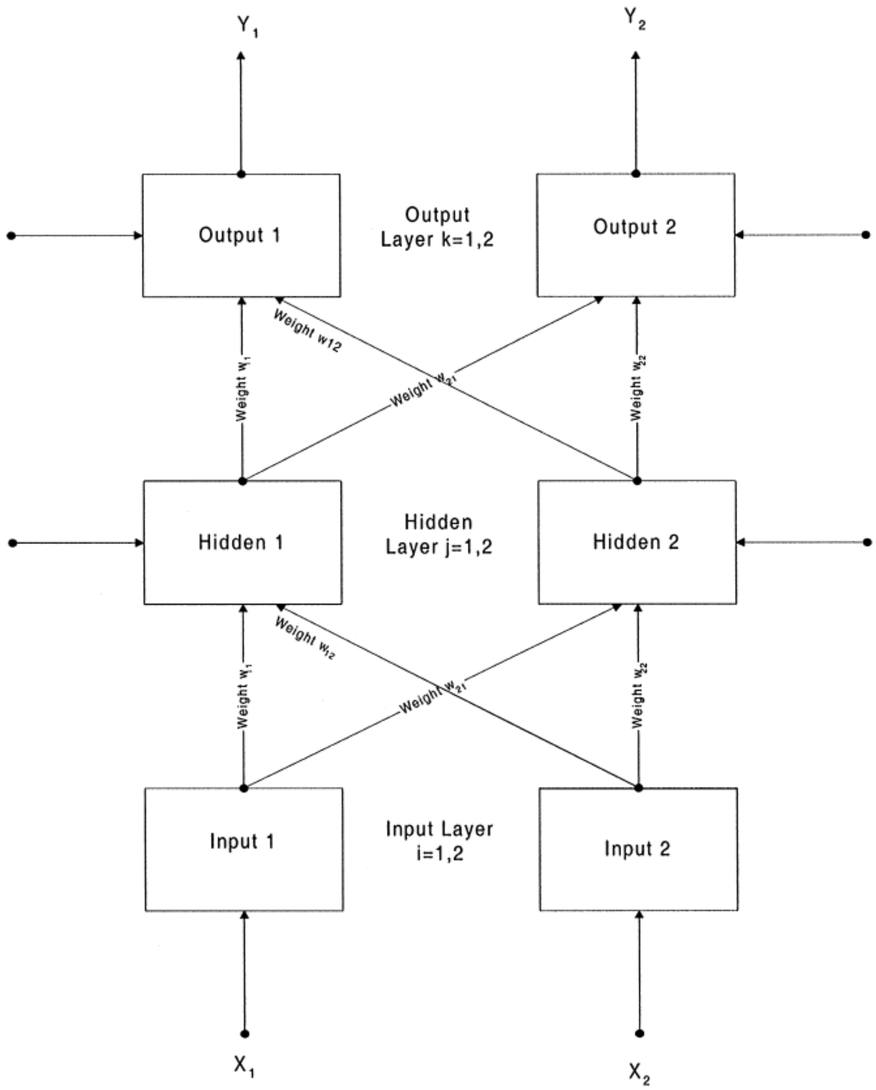
**Figure 2.2.1 Decision Tree**




### 2.2.4 Artificial neural networks (ANN):

ANNs, also referred to as neural networks or deep learning, are a statistical technique loosely modelled on the workings of the human brain, where a set number of 'neurons' work in parallel without any centralised control. Densely interconnected, simple processing nodes referred to as neurons are arranged in layers, with a system of weighted connections determining the flow of information between the network of neurons (Lahsasna, Ainon & Teh, 2010). Explanatory, or input variables are fed through the layers, with the directionality of the information flow determining the type of ANN - for instance, only allowing information to flow in one direction defines "feed-forward" neural networks. A node can be connected to multiple nodes in the layer beneath it - meaning that it receives data from these nodes, and multiple nodes in the layer above it, meaning that it sends data to these nodes. Nodes at the lowest layer - the input layer - take input variables and multiply them by "weights" - a number assigned to a particular input. The sum of the products of weights and inputs is then evaluated against a threshold and is passed on to the connected nodes in the next layer if it exceeds the threshold. If it is lower than the threshold, it does not pass any data to the next layer. When a neural network is being trained, all weights and thresholds are assigned random values. The predictions generated from the random numbers are evaluated against observed outcomes, and the weights and thresholds are adjusted, generating new predictions. This process is repeated until predictions are sufficiently similar to observed outcomes in the training data (West, 2000).

**Figure 2.2.2 Neural Network**




(West, 2000).

### 2.2.5 Support vector machine (SVM):

SVMs are based on the same principle as neural networks, also employing weights, thresholds and layers of nodes, with weights and thresholds being randomly allocated in training. However, SVMs differ from neural networks in that neural networks solve for separating classes in a classification problem, but do not differentiate between two different solutions that separate the training data equally accurately, whilst SVMs solve for the optimal hyperplane that separates the closest points of the different classes (Ren, 2012). This, theoretically, improves the generalisation performance of SVM techniques relative to neural nets (Louzada, Ara & Fernandes, 2016).

### 2.2.6 Hybrid methods:

Hybrid methods, as the name suggests, involve combining different statistical techniques to create a new classifier. For example, Lee and Chen (2005) evaluate a hybrid method that combines neural networks with discriminant analysis to predict credit risk (Louzada, Ara & Fernandes, 2016).

### 2.2.7 Ensemble methods:

Ensemble algorithms involve the combination of predictions from different classifiers to generate more accurate predictions. The three most prominent methods of creating ensemble models are bagging, boosting and stacking.

Bagging: One of the first ensemble machine learning techniques, developed by Breiman (1996a), bagging – short for bootstrap aggregating – involves training a multitude of weak base learners of the same underlying structure, referred to as homogenous learners, in parallel. The base learners are trained on subsamples of the training data, randomly sampled with replacement. In order to create the ensemble model, the base learners are combined using a majority vote methodology. A random forest is an example of a bagging ensemble using multiple decision trees as base learners.

Boosting: As with bagging, boosting is a method of training a model that involves the training of homogenous weak base learners that are combined to create a more powerful model. It differs from bagging in that learners are trained in series, not in parallel, allowing for the adaption of subsequent learners based on the errors of a previous learner. Examples of boosting ensemble algorithms are gradient boosting machines, extreme gradient boosting, and AdaBoost.

Stacking: Also referred to as “Super Learning” or “Stacked Regression”, stacking was first proposed by David Wolpert in 1992. It involves training a “super learner” - i.e., an algorithm, trained not on the variables included in the training set, but on the predictions from a set of algorithms trained on the variables included in the training data. These algorithms, referred to as 'base learners', can consist of any algorithm – stacked ensembles thus are able to consider heterogenous base learners in comparison to bagging and boosting, which can only consider homogenous base learners. Whilst bagging and boosting methods aim to take a set of multiple weak learners to create a single, strong learner, stacking aims to find the optimal combination of strong, diverse algorithms as an ensemble. The point about diversity of learners is important, since stacking allows us to combine the predictions of multiple algorithms, ensembling models with highly correlated predictions is unlikely to yield a super learner that substantially outperforms its base models. However, if we stack a diverse – i.e. less correlated – set of algorithms, the resulting ensemble model is more likely to significantly outperform the predictive performance of any of its base learners (Wang et al., 2011; Wolpert, 1992).

## 2.3 *Findings on Classification Algorithm Performance*

Despite extensive research on the performance of different classifiers in the literature, there is no clear consensus as to the superiority of certain algorithms over others.

**Figure 2.3.1 Crook, Edelman & Thomas (2007) Performance Rank of Classification Techniques in the Literature**

| Author(s) | Year | LR/LDA | LOG | DT | MP | NN | GA | GP | KNN | SVM |
|---|---|---|---|---|---|---|---|---|---|---|
| Srinivisan & Kim | 1987 | 3 | 2 | **1** | 4 | | | | | |
| Boyle et al. | 1992 | **1** | | 2 | 3 | | | | | |
| Henley | 1995 | 2 | 3 | **1** | | | | | | |
| Desai et al. | 1997 | 2 | **1** | | | 3 | | | | |
| Yobas et al. | 2000 | **1** | | 3 | | 4 | 2 | | | |
| West | 2000 | 3 | 2 | 4 | | **1** | | | 5 | |
| Lee et al. | 2002 | 3 | 2 | | | **1** | | | | |
| Malhotra & Malhotra | 2003 | 2 | | | | **1** | | | | |
| Lessmann | 2003 | 3 | 3 | 7 | 5 | 2 | | | 6 | **1** |
| Ong et al. | 2005 | 3 | | 4 | | 2 | | **1** | | |

* LR: Linear regression, LDA: Linear Discriminant Analysis, LOG: Logistic Regression, DT: Decision Trees, MP: Math Programming, NN: Neural Networks, GA: Genetic Algorithms, GP: Genetic Programming, KNN: K-nearest Neighbours, SVM: Support Vector Machines.

** Best-performing algorithm per study indicated in bold

The above table presents the results of Crook, Edelman and Thomas (2007), who summarise the results of the predictive accuracy of different classifiers across ten different studies using credit scoring data. The table presents the relative rank (1 being the most accurate) of different classifiers included in each study. As is evident from these results, none of the algorithms included in any of the studies included in the comparison by Crook et al. stands out as consistently outperforming the others across different studies in the literature (Crook, Edelman & Thomas, 2007).

More recently, Lessmann et al. (2015) conducted a comprehensive benchmarking of classifiers on eight credit risk datasets. The average rank of each type of classifier over a range of performance metrics is shown in the table below.

***Figure 2.3.2*** **Lessmann et al. (2015) Classification Techniques Benchmarking Results**

| Classifier Family | Classifier | AUC | PCC | BS | H | PG | KS | AvgR | High Score |
|---|---|---|---|---|---|---|---|---|---|
| Individual Classifier | ANN | 16.2 | 18.6 | 27.5 | 17.9 | 14.9 | 17.6 | 18.8 | 14 |
| | B-Net | 27.8 | 26.8 | 20.4 | 28.3 | 23.7 | 26.2 | 25.5 | 30 |
| | CART | 36.5 | 32.8 | 35.9 | 36.3 | 25.7 | 34.1 | 33.6 | 38 |
| | ELM | 30.1 | 29.8 | 35.9 | 30.6 | 27 | 27.9 | 30.2 | 36 |
| | ELM-K | 20.6 | 19.9 | 36.8 | 19 | 23 | 20.6 | 23.3 | 26 |
| | J4.8 | 36.9 | 34.2 | 34.3 | 35.4 | 35.7 | 32.5 | 34.8 | 39 |
| | k-NN | 29.3 | 30.1 | 27.2 | 30 | 26.6 | 30.5 | 29 | 34 |
| | LDA | 21.8 | 20.9 | 16.7 | 20.5 | 24.8 | 21.9 | 21.1 | 20 |
| | LR | 20.1 | 19.9 | 13.3 | 19 | 23.1 | 20.4 | 19.3 | 16 |
| | LR-R | 22.5 | 22 | 34.6 | 22.5 | 21.4 | 21.4 | 24.1 | 28 |
| | NB | 30.1 | 29.9 | 23.8 | 299.3 | 22.2 | 29.1 | 27.4 | 33 |
| | RbfNN | 31.4 | 31.7 | 28 | 31.9 | 24.1 | 31.7 | 29.8 | 35 |
| | QDA | 27 | 26.4 | 22.6 | 26.4 | 23.6 | 27.3 | 25.5 | 31 |
| | SVM-L | 21.7 | 23 | 31.8 | 22.6 | 19.7 | 21.7 | 23.4 | 27 |
| | SVM-Rbf | 20.5 | 22.2 | 31.8 | 22 | 21.7 | 21.3 | 23.2 | 25 |
| | VP | 37.8 | 36.4 | 31.4 | 37.8 | 34.6 | 37.6 | 25.9 | 40 |
| Homogenous Ensemble | ADT | 22 | 18.8 | 19 | 21.7 | 19.4 | 20 | 20.2 | 17 |
| | Bag | 25.1 | 22.6 | 18.3 | 23.5 | 25.2 | 24.7 | 23.2 | 24 |
| | BagNN | 15.4 | 17.3 | 12.6 | 16.5 | 15 | 16.6 | 15.6 | 13 |
| | Boost | 16.9 | 16.7 | 25.2 | 18.2 | 19.2 | 18.1 | 19 | 15 |
| | LMT | 22.9 | 23.4 | 15.6 | 25.1 | 20.1 | 22.9 | 22.2 | 23 |
| | RF | 14.7 | 14.3 | 12.6 | 12.8 | 19.4 | 15.3 | 14.8 | 12 |
| | RotFor | 22.8 | 21.9 | 23 | 21.1 | 21.6 | 22.9 | 22.2 | 23 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | SGB | 21 | 19.9 | 20.8 | 21.2 | 22.5 | 20.8 | 21 | 19 |
| Heterogenous Ensemble | AvgS | 8.7 | 10.8 | 6.6 | 9.2 | 12 | 9.2 | 9.4 | 4 |
| | AvgW | 7.3 | 12.6 | 7.9 | 7.3 | 10.2 | 7.9 | 8.9 | 2 |
| | Stack | 30.6 | 26.6 | 37.4 | 29.6 | 30.7 | 29.5 | 30.7 | 37 |
| | CompM | 18.3 | 15.3 | 36.5 | 17.2 | 20 | 18.2 | 20.9 | 18 |
| | EPVRL | 8.2 | 10.8 | 6.8 | 9.3 | 13.7 | 11 | 10 | 5 |
| | GASEN | 8.6 | 10.6 | 6.5 | 9 | 11.4 | 9 | 9.2 | 3 |
| | HCES | 10.9 | 11.7 | 7.5 | 10.2 | 14.8 | 13.1 | 11.4 | 9 |
| | HCES-Bag | 7.7 | 9.7 | 5.8 | 8.2 | 12.5 | 9.2 | 8.8 | 1 |
| | MPOE | 9.9 | 10.1 | 9.4 | 9.9 | 15.1 | 10.9 | 10.9 | 6 |
| | Top-T | 8.7 | 11.3 | 10 | 9.8 | 14.8 | 12.3 | 11.2 | 8 |
| | CuCE | 10 | 12 | 10.1 | 10.8 | 12.1 | 11.2 | 11 | 7 |
| | k-Means | 12.6 | 13.6 | 9.8 | 11.2 | 14.9 | 12 | 12.4 | 10 |
| | KaPru | 27.7 | 25.3 | 15.7 | 28.1 | 25.1 | 25.4 | 24.5 | 29 |
| | MDM | 24.4 | 24 | 11.6 | 23.7 | 21.7 | 23.7 | 21.5 | 21 |
| | UWA | 9.3 | 11.8 | 19.5 | 10.1 | 14.3 | 10.9 | 12.7 | 11 |
| | kNORA | 27.1 | 26.7 | 28.1 | 28.1 | 23.4 | 25.9 | 26.6 | 32 |
| | PMCC | 40.1 | 28.6 | 32.9 | 39.5 | 39.9 | 38.8 | 38.3 | 41 |

* Abbreviations defined as follows: AUC=Area under the Receiver Operating Characteristics Curve, PCC=Percentage Correctly Classified, BS=Brier Score, H=H-measure, PG=Partial Gini, KS=Kolmogorov-Smirnov statistic, AvgR=Average Rank, B-Net=Bayesian Network, CART=Classification and Regression Trees, ELM=Extreme learning machine, ELM-K=Kernalized ELM, kNN=k-nearest neighbor, J4.8=J4.8, LDA=Linear discriminant analysis, SVM-L=Linear support vector machine, LR=Logistic regression, ANN=Multilayer perceptron artificial neural network, NB=Naive Bayes, QDA=Quadratic discriminant analysis1, RbfNN=Radial basis function neural network, LR-R=Regularized logistic regression, Rbf=SVM with radial basis kernel function SVM-, VP=Voted perceptron, ADT=Alternating decision tree, Bag=Bagged decision trees, BagNN=Bagged MLP, Boost=Boosted decision trees, LMT=Logistic model tree, RF=Random forest, RotFor=Rotation forest, SGB=Stochastic gradient boosting, AvgS=Simple average ensemble, AvgW=Weighted average ensemble, Stack=Stacking, CompM=Complementary measure, EPVRL=Ensemble pruning via reinforcement learning, GASEN=GASEN, HCES=Hill-climbing ensemble selection, HCES-Bag=HCES with bootstrap sampling, MPOE=Matching pursuit optimization ensemble, Top-T=Top-T ensemble, CuCE=Clustering using compound error, k-Means=k-Means clustering, KaPru=Kappa pruning, MDM=Margin distance minimization, UWA=Uncertainty weighted accuracy, PMCC=Probabilistic model for classifier competence, kNORA=k-nearest oracle.

As is evident from the performance of individual classifiers presented in Figure 2.3.2, the traditional credit risk modelling techniques of logistic regression (LR) is competitive with more sophisticated algorithms. The average performance rank of logistic regression is 19.3, only bettered by neural networks (ANN) with an average rank of 18.8. The finding that logistic regression is competitive with neural networks and outperforms sophisticated statistical techniques such as support vector machines (SVM), confirms the finding of Crook, Edelman and Thomas (2007); that more sophisticated statistical techniques do not necessarily result in more accurate classifiers.

Looking at the performance of the heterogenous and homogenous ensembles relative to the individual classifiers, there is strong evidence that for the datasets used in this study, ensemble models outperform individual classifiers at predicting default - on average. It is noteworthy that Lessmann et al. (2015) do not break down the performance of the classifiers evaluated on each of the datasets used in their study, rather simply reporting the average rank of each classifier for each performance metric across all the datasets used.

The findings of Crook, Edelman and Thomas (2007) and Lessmann et al. (2015) – that there is no ‘best’ classification technique for credit risk prediction - is echoed by Abdou and Pointon (2011) in their review of some 214 articles of research on credit risk prediction. They find that the predictive performance of an algorithm is dependent on a multitude of factors, including the size of the dataset, the number of variables, the predictive power of these variables, and many others (Abdou & Pointon, 2011).

A relatively recent invention, extreme gradient boosting – an ensemble modelling method using a boosting techniques that has fast become the darling of the machine learning community for its speed and performance – has featured in the credit risk literature, despite not being included in the aforementioned reviews that preceded its creation. Xia et al. (2017) conducted a benchmarking study on a peer-to-peer lending dataset and found that extreme gradient boosting models performed significantly better than neural networks, logistic regression, random forest, and gradient boosting machines over a selection of performance metrics (Xia et al., 2017).

## 2.4 *Feature Selection*

Feature selection is the process of removing partially or completely irrelevant or redundant explanatory variables from the training dataset prior to fitting a model to the data. Irrelevant features can be defined as those explanatory variables that do not affect the dependent variable in any way, whilst redundant features are those that do not add any additional information to the prediction of the target. There are a number of reasons for using feature selection techniques to reduce the number of features in a training dataset, with the most relevant to credit scoring being the improvement of prediction accuracy and efficiency in terms of training time of algorithms (Dash & Liu, 1997).

There are two distinct groups of feature selection algorithms - the filter approach, and the wrapper approach (Chen & Li, 2010). Techniques are classified according to their dependence on the inductive algorithm that will make use of the subset of the training dataset generated by the feature selection process itself. Filter methods do not make use of the inductive algorithm, whilst wrapper methods employ it as the evaluation function for feature importance (Dash & Liu, 1997).

Of the 187 studies on credit scoring reviewed by Louzada, Ara and Fernandes (2016) just over half of all papers involved some form of feature selection. (Louzada, Ara & Fernandes, 2016).

## 2.5 *Performance measures*

### 2.5.1 Confusion matrix/threshold metric:

The confusion matrix provides a comparison of the predicted outcomes of a classifier against the actual or observed outcomes in the test dataset. Classifications are defined as correct when the class assigned to an observation (in the retail credit context, meaning an individual) is equal to their observed class. For a PD model, a correct classification implies a model assigning the "no default", or $\{0\}$ class to a customer who did not default, and a "default" or $\{1\}$ class to a customer who did default. Incorrectly classified observations are referred to as *misclassified.*

**Figure 2.5.1 Confusion Matrix**

| | | Predicted | |
|---|---|---|---|
| | | 1 (default) | 0 (no default) |
| Observed | 1 (default) | *True Positive (TP)* | *False Negative (FN)* |
| | 0 (no default) | *False Positive (FP)* | *True Negative (TN)* |

It is important to note that although the predicted outcomes depicted in the above table are binary, the output of a binary classification model is in actuality a probability. In order to derive predicted class from a probability, a threshold is required. In the context of classification prediction, thresholds refer to the cut-off point where an estimated probability is defined as a "good" or a "bad". For a PD model with a threshold of 0.5, all potential borrowers with an estimated PD greater than or equal to 0.5 would be classified as a predicted "bad" $\{1\}$ whilst all those with 0.5 or less would be classified as $\{0\}$, or "good" (Fawcett, 2006).

There are several metrics based on the confusion matrix that are used to evaluate the performance of a classifier on a test dataset - although these are all applicable to any classification problem, they will be defined here in the context of a PD binary classification model. The terms "default" and "bads" are used interchangeably, as are non-defaults and "goods" - referring to class $\{1\}$ and $\{0\}$ respectively.

### *2.5.2* Average Correct Classification/Accuracy:

The percentage of predictions made by a model that classified customers to the class in which they were observed to belong to, defined as

$$ACC = \frac{TP+TN}{N}$$

(Louzada, Ara & Fernandes, 2016).

### *2.5.3* Sensitivity/True Positive Rate/Recall:

The fraction of all customers that were observed to have defaulted that were assigned to class {1} at a set threshold. Defined as

$$sensitivity = \frac{TP}{TP+FN}$$

(Louzada, Ara & Fernandes, 2016).

### *2.5.4* Specificity/True Negative Rate:

The percentage of cases where cases classified as {0} were observed to have not defaulted. Defined as

$$specificity = \frac{TN}{TN+FP}$$

(Louzada, Ara & Fernandes, 2016).

### *2.5.5* Precision:

The ratio of true positives to all cases classified as positive:

$$precision = \frac{TP}{TP+FP}$$

(Louzada, Ara & Fernandes, 2016).

### *2.5.6* False Negative Rate/Type I Error:

The fraction of customers observed to have defaulted who were predicted to not default. Defined thus:

$$FNR = \frac{FN}{TP+FN}$$

(Louzada, Ara & Fernandes, 2016).

### *2.5.7* False Positive Rate/Type II Error:

The percentage of customers who were observed to have not defaulted who were predicted to default:

$$FPR = \frac{FP}{TN+FP}$$

(Louzada, Ara & Fernandes, 2016).

### 2.5.8 F-measure:

Combines precision and recall into a single metric:

$$F - measure = \frac{(\beta^2+1)*precision*recall}{\beta^2*precision+recall}$$

where setting β=1 results in a F-measure that weights precision and recall equally, whilst values of $\beta > 1$ result in a favouring of precision, and recall where $\beta < 1$ (Sokolova, Japkowicz & Szpakowicz, 2006).

F1, with $\beta = 1$ score weights precision and recall equally and is calculated according to the equation

$$F1 = 2(\frac{precision\ *\ recall}{precision\ +\ recall})$$

(H2O.ai, 2021).

F2 sets $\beta = 2$ and assigns more weight to recall than precision. It penalises a model more for false negatives than false positives according to the formula

$$F2 = 5(\frac{precision * recall}{4*precision + recall})$$

(H2O.ai, 2021).

F0.5 sets $\beta = 0.5$, penalising false positives more than false negatives and prioritising precision over recall according to the formula

$$F0.5 = 1.25(\frac{precision * recall}{0.25 * precision + recall})$$

(H2O.ai, 2021).

### 2.5.9 K/S statistic:

The Kolmogorov-Smirnov statistic measures the maximum distance between the distributions of the two classes – in this case defaulters and non-defaulters (Stephens, 1974). It measures a model’s ability to differentiate between the two classes in a binary classification problem and is often used in conjunction with AUC to evaluate model performance (H2O.ai, 2021) (Adeodato & Melo, 2016). It is calculated according to the equation:

$$KS = sup_x|F_{1,n}(x) - F_{2,m}(x)$$

where $sup_x$ is the supremum function, $F_{1,n}$ is the sum of all events observed so far up to the category i as a fraction of the total number of events and $F_{2,m}$ is the sum of all non-events observed so far up to the category i as a fraction of the total number of non-events (H2O.ai, 2021).

### 2.5.10 Area under the ROC curve

The Receiver Operating Characteristic Curve (ROC Curve) plots the performance of a binary classifier over all possible thresholds. ROC curves plot the *sensitivity* of a binary classifier (bads classified as bads) on the vertical axis against the rate of goods classified as bads (1 - *specificity*) on the horizontal axis. To compare the performance of different classifiers using the ROC curve, the area underneath the ROC curve (AUC) can be calculated for each classifier. The AUC can be understood as the probability that a randomly selected observation of class {1} (i.e., a bad) is correctly classified as a bad. The diagonal, where the rate of correctly classified bads equals the rate of goods classified as bads, has a AUC of 0.5 - hence, the difference in AUC between a classifier and the AUC of the diagonal (0.5) can be understood as the improvement in prediction accuracy from a random classifier (Lessmann et al., 2015).

**Figure 2.5.2 ROC Curve**


(Abdou & Pointon, 2011).

### 2.5.11 GINI Coefficient

The GINI coefficient is closely related to the area under the ROC curve, defined as twice the area between the ROC curve and the diagonal $GINI = 2 * AUC - 1$. Given their close relationship, researchers tend to use one or the other in evaluating the performance of a classifier, but not both (Hand & Till, 2001).

### 2.5.12 Area under the Precision-Recall Curve

Similarly to the Receiver Operating Characteristics (ROC) curve, the precision-recall curve plots performance on two performance metrics – in this case, precision and recall – over every possible threshold of a classifier. Precision is plotted on the vertical axis, with recall plotted on the horizontal axis. Because the precision-recall curve does not take true negatives into account, it is recommended over AUC for cases of class imbalance in binary classification (Goadrich, Oliphant & Shavlik, 2006).

### 2.5.13 H-measure

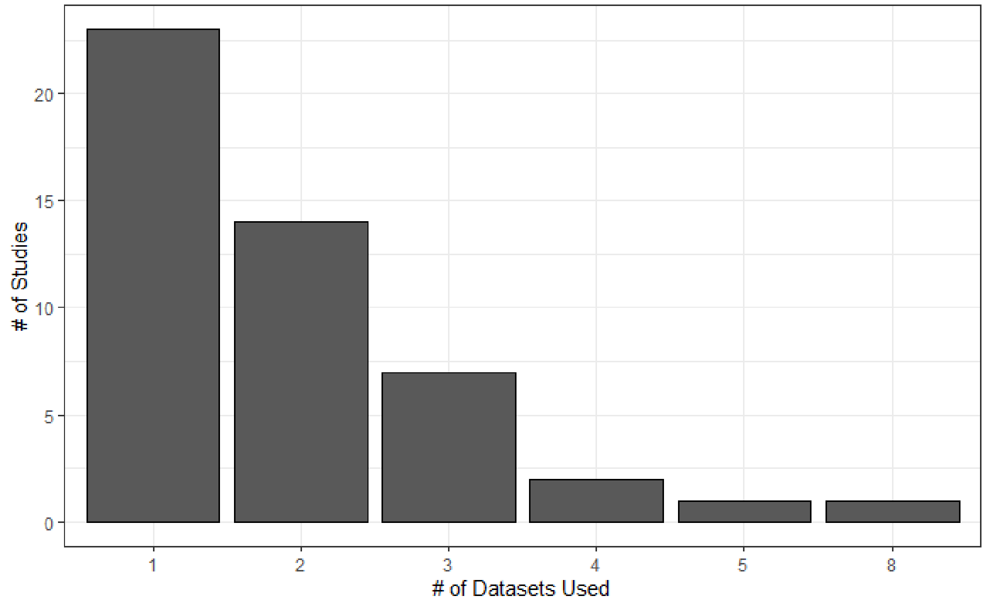
Introduced by Hand (2009), as an alternative to AUC, the h-measure accounts for differences in misclassification costs across different classifiers. Because the AUC weights type I and II errors equally - i.e., the misclassification of a bad (observed 1) as a good (predicted 0) is as costly as the misclassification of a good (observed 0) as a bad (predicted 1) (Hand, 2009). Naturally, the accuracy of this assumption depends on the nature of the classification problem - whilst we can imagine it to be true for some problems, it is far from guaranteed in the retail credit context. For a Type II error in credit, the misclassification cost equals all potential interest and fee revenue foregone from not granting a loan application. For a Type I error, the misclassification cost depends on exposure at default (EAD) and loss given default (LGD). For unsecured credit, where there is no security in place to reduce LGD, the cost of a type I error could be as high as the value of the loan principal, whilst the cost of a type II error would be substantially smaller. For secured credit, the difference would be less. The h-measure uses a *beta*-distribution to specify the relative costs of misclassification errors in a manner that ensures consistency across classifiers (Hand, 2009).

## *2.6 Datasets*

Lessmann et al. (2015) review of studies comparing classifier performance observed that researchers tend to use a small number of data sets, containing low numbers of observations and relatively few independent variables. They note the inappropriateness of this, highlighting the need for multiple data sets to demonstrate the robustness of a predictive model to environmental conditions. Additionally, they highlight the fact that the large size (many observations) and high-dimensionality (many variables) of real-world credit data sets makes findings on small, low-dimensional data sets difficult to justify for real-world applications. Of the studies referenced, an average of 1.9 datasets was utilised, with 48% of studies using only a single dataset to evaluate classifiers. Only 23% of the 48 studies used three or more data sets. 68% of the studies used data sets with 20 or less independent variables, with 15% using data sets containing 30 or more independent variables. 75% of the studies used datasets containing less than 1500 observations per data set, with only 7 of the 48 studies referenced using datasets containing more than 4000 observations (Lessmann et al., 2015).

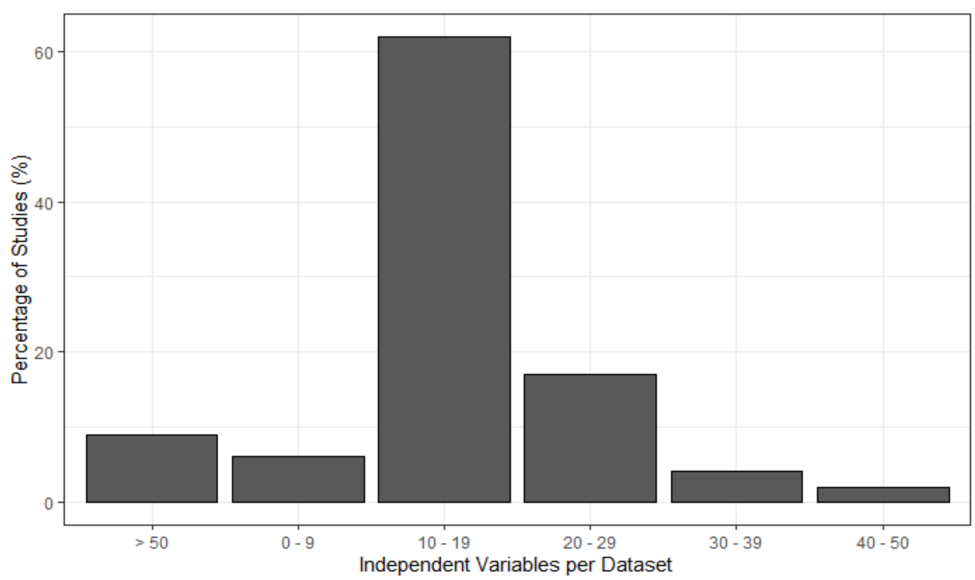
In their review of the literature on classification methods in credit scoring, Louzada, Ara and Fernandes (2016) measured the proportion of public and private datasets used by researchers in the 187 papers they reviewed. They found that approximately 50% of authors used private datasets, whilst approximately 40% used public datasets, with the remainder using some combination of private and public - and that this proportion remained constant over time. Furthermore, they found that some 45% of all reviewed papers considered two famous public datasets from the University of California Irvine (UCI) library in their study - known as the g*erman* and *australian* credit datasets (Louzada, Ara & Fernandes, 2016).

**Figure 2.6.1 Distribution of Number of Datasets used by Studies Referenced by Louzada, Ara & Fernandes (2016)**




(Louzada, Ara & Fernandes, 2016).

**Figure 2.6.2 Distribution of Dimensionality of Datasets used by Studies Referenced by Louzada, Ara & Fernandes (2016).**

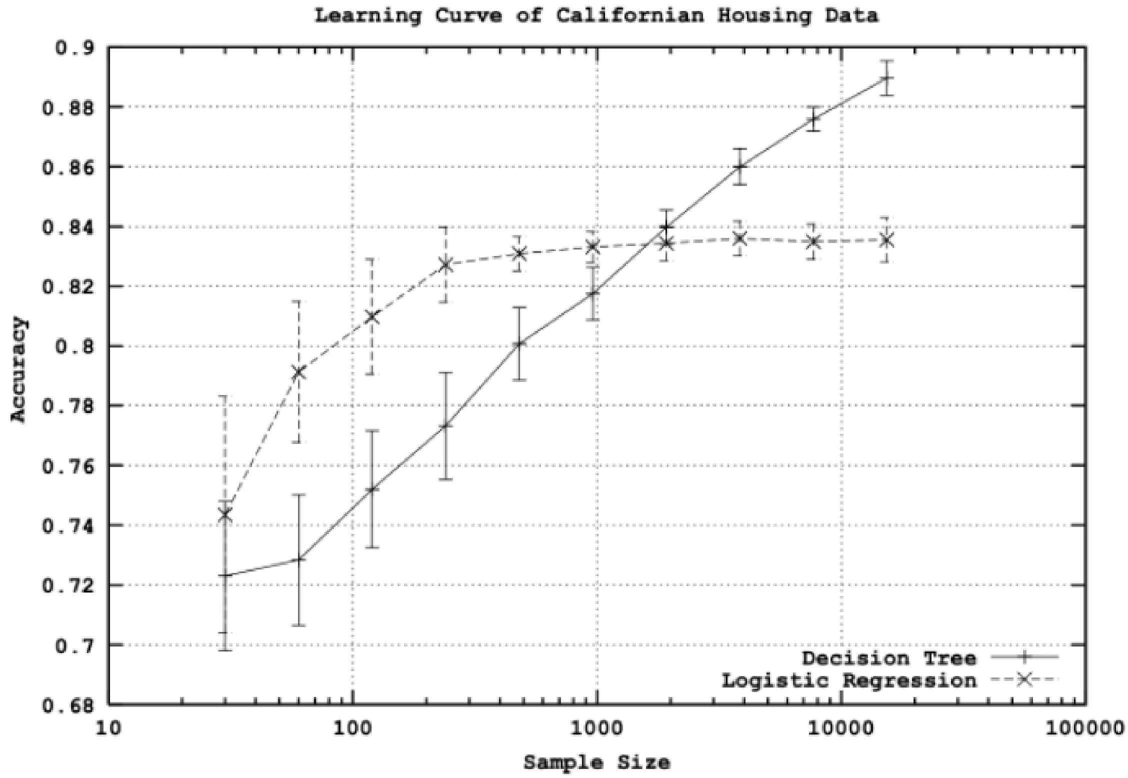



(Louzada, Ara & Fernandes, 2016).

To the knowledge of this author, no research has been conducted on the relationship between the characteristics of the datasets used in studies and the performance of different classification algorithms in the field of credit risk prediction. However, research carried out elsewhere is applicable here. Sun et al. (2017), using deep learning in computer vision for object classification, object detection, semantic segmentation and human pose estimation, found that model performance improves logarithmically based on the volume of the training data size (the dataset they used for this study had more than 300 million images to include in their training set - truly huge data).

In their 2003 paper, Perlich, Provost and Simonoff used learning-curve analysis to examine the relationship between the size of a training set to the performance of the classification accuracy of logistic regression models and tree induction. Using a sample of 36 datasets from the UCL library representing binary classification tasks, they created subsamples of these datasets that varied in the number of observations included from the original datasets. They trained a logistic regression and a decision tree classifier on each of the datasets, evaluating the accuracy of each dataset on each sample. They then plotted the results, in order to evaluate the relationship between model performance and training set size; this method of analysis is called learning-curve analysis (Perlich, Provost & Simonoff, 2003).

**Figure 2.6.3 Learning Curve Analysis from Perlich, Provost & Simonoff (2003)**




(Perlich, Provost & Simonoff, 2003)

Perlich, Provost and Simonoff found that, in general, logistic regression models outperform decision trees for smaller datasets, and that decision trees outperform logistic regression models for large datasets. Looking at the figure above from their paper, this result is evident for the case of the Californian Housing dataset. It is clear how the decision tree classifier is able to leverage larger quantities of data to improve accuracy, whilst the logistic regression algorithm, despite being more accurate on smaller subsamples of the data, plateaus in terms of performance; accuracy does not increase significantly as the size of the subsamples increases beyond 1,000 observations (Perlich, Provost & Simonoff, 2003). This finding has implications for much of the research in the body of literature comparing the performance of different classifiers; given the widespread use of UCI datasets by researchers, and the fact that most of these datasets can be classified as "small" by the methodology outlined by Perlich, Provost and Simonoff, the common finding of the relatively high performance of logistic regression compared to more sophisticated statistical techniques can be argued to be biased by the nature of the training sets that are predominantly used by researchers. Additionally, the tendency of researchers in the literature to present their findings as benchmarking analysis, where the average performance of classifiers over multiple datasets is reported without including the performance on individual datasets, has the potential to lead to skewed conclusions. Attempts to make such analysis more robust by including additional datasets may not solve the problem, due to the high availability of small datasets. This relatively easy access to small datasets, and the equal weighting of results over all datasets in calculating average performance rank, can be argued to have resulted in a misrepresentation of algorithms that perform relatively poorly on smaller training samples but perform well on larger samples in the literature.

# Chapter 3

# Methodology

## *3.1 Datasets*

This paper shall make use of the following publicly available datasets, which have all been used previously by various researchers in the literature:

- The *australian* dataset, provided by the UCI library, concerns a sample of credit approvals in Australia, and contains 690 observations and 14 variables.
- The *german* dataset, also provided by the UCI library, classifies a sample of German customers as good or bad credit risks, and contains 1000 observations and 20 variables.
- The *taiwan* dataset, also provided by the UCI library, contains information on whether or not a customer defaulted on a loan payment due between April and September 2005. It contains 30000 observations and 24 variables.
- The 'Give me some credit' (*gmc)* dataset, from the eponymous Kaggle competition, classifies some form of financial distress in the next two years, and contains 150,000 observations and 12 variables.
- The *lc* dataset, from the Kaggle competition 'Lending Club Loan Data' has information on all loans issued by Lending Club, an American lending company, between 2007-2015. It contains 145 variables and 1,303,607 observations.

In order to test the interaction between model performance and dataset size, the large datasets used in this paper (i.e., the *gmc* and the *lc* datasets) are sampled in order to create artificially smaller samples. Samples of between 1,000 and 100,000 loans with step sizes of 1,000 (i.e., 1,000, 2,000, 3,000, …, 100,000) are randomly drawn from the full dataset in order to create these subsamples. Due to the high time requirements for training complex models on large datasets, no samples with more than 100 thousand observations are used for the purpose of training models in this paper.

The evaluation of classification techniques in this study can be split into two distinct streams for clarity; the benchmarking of classification techniques on the full set of datasets - *australian*, *german*, *taiwan*, *gmc* and *lc* – can be referred to as general analysis. The second part, in which algorithms are benchmarked on smaller training samples created from the *lc* and *gmc* dataset shall be referred to as learning curve analysis.

## *3.2 Data Preprocessing*

For the *lc* dataset, the outcome of the loan in the raw dataset, *loan_status*, was a categorical variable with nine levels. These are presented with their frequencies in the table below.

**Figure 3.2.1 Lending Club Loan Status Frequencies**

| **loan_status** | **n_customers** |
|---|---:|
| Fully Paid | 1,041,952 |
| Current | 919,695 |
| Charged Off | 261,655 |
| Late (31-120 days) | 21,897 |
| In Grace Period | 8,952 |
| Late (16-30 days) | 3,737 |
| Does not meet the credit policy. Status: Fully Paid | 1,988 |
| Does not meet the credit policy. Status: Charged Off | 761 |
| Default | 31 |

At the time of data extraction, only two of these categories are deemed relevant to this modelling problem. Customers with a loan status of “Current”, “Late (31-120 days)”, “In Grace Period”, “Late (16-30 days)” are considered to be either in-loan or in arrears prior to the date at which loans in arrears are written off (this can be inferred at 120 days from the data above). For customers with a loan status of “Does not meet the credit policy. Status: Fully Paid” or “Does not meet the credit policy. Status: Charged Off”, customers were not extended credit facilities due to not meeting prerequisite requirements for a loan. Customers with a loan status of “Default” are excluded due to their low volumes; this loan status category is assumed to correspond to some legacy credit policy that has since been deprecated. This leaves the *loan_status* variable with two categories – “Fully Paid” and “Charged Off”. To convert this to the binary variable for modelling, *loan_status* is recoded to a binary variable set to a value of 1 if a loan is “Charged Off” and 0 if a loan is “Fully Paid”. All observations with any other loan status are dropped.

Variables with more than five percent of their values missing or all of their values constant (i.e., all loans have the same value for said variable) are dropped. Of the 145 variables included in the *lc* dataset, we select 23 variables for inclusion in the sample on which the algorithms are trained. A list of these variables and their meanings are included in Figure 3.2.2 below.

**Figure 3.2.2 Lending Club Variables Included in Modelling**

| variable name | data type | description |
|---|---|---|
| response | categorical | Binary variable set to a value of 1 where a loan has been charged off and 0 if a loan has been fully paid |
| purpose | categorical | Self-reported purpose for loan. 14 possible categories such as "car", "house", and "home_improvement" |
| emp_length | categorical | Length of time borrower has been employed at current job, binned for 10+ years |
| acc_now_delinq | numeric | Number of accounts for which borrower is delinquent |
| home_ownership | categorical | Description of borrower's home ownership status |
| revol_util | numeric | The percentage of total available revolving credit a borrower is utilising |
| desc | numeric | The number of characters included in the loan description provided by the borrower |
| delinq_2yrs | numeric | The number of accounts for which the borrower is categorised as delinquent for a rolling 2-year window from point of application |
| inq_last_6mths | numeric | The number of inquiries made by borrower over past 6 months |
| tax_liens | numeric | Total number of tax liens against customer |
| earliest_cr_line | numeric | The length of time (in years) since borrower first opened a line of credit |
| annual_inc | numeric | The self-reported annual income of a customer in US dollars |
| loan_amnt | numeric | The total value in US dollars of the loan being applied for |
| term | categorical | The term of the loan being applied for - either 36 or 60 months. |
| grade | categorical | Letter grade (A-G) of borrower's estimated creditworthiness |

| | | |
|---|---|---|
| sub_grade | categorical | Number grade (1-5) that ranks creditworthiness within "grade" variable |
| delinq_amnt | numeric | The total value in US dollars of due repayment to existing credit facilities for which borrower is currently delinquent |
| open_acc | numeric | The total number of open credit facilities for the borrower |
| total_acc | numeric | The total number of credit facilities for the borrower |
| dti | numeric | The borrower's debt coverage ratio excluding mortgage repayments, calculated using their self-reported income |
| pub_rec | numeric | Number of derogatory public records associated with borrower |
| pub_rec_bankruptcies | numeric | Number of public record bankruptcies associated with borrower |
| verification_status | categorical | Indicates borrower's verification status with Lending Club |

For the *german* dataset, the target variable, *response*, assigns a value of 2 to "bad" loans, and a value of 1 to "good" loans. For consistency, the *response* variable is recoded such that a value of 1 indicates "bad" (i.e., defaulted) loans, with zero indicating "good", non-defaulted loans. The frequency of each outcome is shown in Figure 3.2.3 below.

**Figure 3.2.3 German Response Frequencies**

| response | n_customers |
|---|---|
| 0 (good) | 700 |
| 1 (bad) | 300 |

All variables from the original dataset are included in the training samples. A list of these variables and their descriptions is provided in the table below.

**Figure 3.2.4 German Variables**

| variable name | data type | description |
|---|---|---|
| response | categorical | Binary variable set to a value of 1 where a loan has defaulted and 0 if a loan has been settled |
| account_status | categorical | Status of existing checking account |
| duration | numeric | Duration of loan in months |
| credit_history | categorical | Categorical variable indicating customer's history of repaying loans |
| purpose | categorical | Purpose for which loan is taken. Includes categories such as "car (new)", "car (used)", "business", and "education" |
| credit_amount | numeric | The currency value of the loan, denominated in Deutsche Mark |
| savings | categorical | Binned values of a customer's total savings, denominated in Deutsche Mark |
| employment_since | categorical | Binned duration of current employment in years |
| instalment_rate | numeric | Instalment amount as a percentage of disposable income |
| status | categorical | Categorical variable indicating customer's sex and relationship status |
| other_debtors | categorical | Indicates whetehr application is joint or single |
| present_residence_duration | numeric | Length of time in years applicant has lived in current residence |
| property | categorical | Categorical variable describing applicant's assets |
| age | numeric | Applicant's age in years |
| other_instalments | categorical | Describes any existing loans of applicant |
| housing | categorical | Describes whether applicant rents or owns their current residence |
| n_credits | numeric | Number of existing loans applicant has with lender |
| job | categorical | Describes applicant's current employment |
| n_people | numeric | Number of dependents |
| telephone | categorical | Indicates whether applicant owns telephone or not |
| foreign | categorical | Indicates whether applicant is a foreign worker |

For the *taiwan* dataset, the response variable is a binary categorical variable set to 1 where a borrower has defaulted on their credit card repayments and 0 otherwise.

**Figure 3.2.5 Taiwan Response Frequencies**

| response | n_customers |
|---|---:|
| 0 (repaid) | 19,471 |
| 1 (default) | 5,529 |

All 23 independent variables from the original training dataset are included in the samples used in this study. Table 3.2.6 below provides a description of these variables.

**Figure 3.2.6 Taiwan Variables**

| variable name | data type | description |
|---|---|---|
| X1 | numeric | Value of the credit facility, denominated in NT dollars. Includes both individual consumer credit and their family supplementary credit |
| X2 | categorical | Indicates sex of borrower |
| X3 | categorical | Categorical variable indicating education level of borrower |
| X4 | categorical | Marital status of borrower |
| X5 | numeric | Age of borrower in years |
| X6 | categorical | Customer repayment status in September 2005 |
| X7 | categorical | Customer repayment status in August 2005 |
| X8 | categorical | Customer repayment status in July 2005 |
| X9 | categorical | Customer repayment status in June 2005 |
| X10 | categorical | Customer repayment status in May 2005 |
| X11 | categorical | Customer repayment status in April 2005 |
| X12 | numeric | Bill statement amount denominated in NT dollars in September 2005 |
| X13 | numeric | Bill statement amount denominated in NT dollars in August 2005 |
| X14 | numeric | Bill statement amount denominated in NT dollars in July 2005 |
| X15 | numeric | Bill statement amount denominated in NT dollars in June 2005 |
| X16 | numeric | Bill statement amount denominated in NT dollars in May 2005 |

| | | |
|---|---|---|
| X17 | numeric | Bill statement amount denominated in NT dollars in April 2005 |
| X18 | numeric | Amount paid in NT dollars in September 2005 |
| X19 | numeric | Amount paid in NT dollars in August 2005 |
| X20 | numeric | Amount paid in NT dollars in July 2005 |
| X21 | numeric | Amount paid in NT dollars in June 2005 |
| X22 | numeric | Amount paid in NT dollars in May 2005 |
| X23 | numeric | Amount paid in NT dollars in April 2005 |

For the *australian* dataset, the response variable is a binary categorical variable that is assigned a value of 1 where a borrower has defaulted on their credit card repayments and 0 if they are in good standing. The frequency of each outcome is described in Figure 3.2.7 below.

**Figure 3.2.7 Australian Response Frequencies**

| response | n_customers |
|---|---|
| 0 | 383 |
| 1 | 307 |

This study used all 14 of the original independent variables for modelling. In order to preserve anonymity, the providers of the dataset changed the names of all variables to meaningless symbols. Therefore, Figure 3.2.8 below provides the data type of the variables included in the *australian* dataset but does not include any descriptions as these are not available.

**Figure 3.2.8 Australian Variables**

| variable name | data type |
|---|---|
| response | categorical |
| A1 | categorical |
| A2 | numeric |
| A3 | numeric |
| A4 | categorical |
| A5 | categorical |
| A6 | categorical |
| A7 | numeric |
| A8 | categorical |
| A9 | categorical |

| | |
|---|---|
| A10 | numeric |
| A11 | categorical |
| A12 | categorical |
| A13 | numeric |
| A14 | numeric |

For the *gmc* dataset, the response variable *SeriousDlqin2yrs* is a binary categorical variable that has a value of 1 where a borrower was delinquent at 90 days past due at any point over the previous two years and 0 otherwise. The frequency of each outcome is presented in Figure 3.2.9 below.

**Figure 3.2.9 GMC Response Frequencies**

| response | n_customers |
|---|---|
| 0 | 139,974 |
| 1 | 10,026 |

**Figure 3.2.10 GMC Variables**

| variable name | data type | description |
|---|---|---|
| SeriousDlqin2yrs | categorical | Borrower was delinquent at 90 days past due in the previous 2 years |
| RevolvingUtilizationOfUnsecuredLines | numeric | Total balance on all unsecured credit facilities as a percentage of total available credit |
| age | numeric | Age of borrower in years |
| NumberOfTime30-59DaysPastDueNotWorse | numeric | Number of times borrower has been 30-59 days past due but no worse in the last 2 years |
| DebtRatio | numeric | Monthly debt payments, alimony and living costs divided by monthly gross income |
| MonthlyIncome | numeric | Monthly income |
| NumberOfOpenCreditLinesAndLoans | numeric | Total number of open loans (e.g., vehicle finance or mortgage) and lines of credit (e.g., credit cards) |
| NumberOfTimes90DaysLate | numeric | Number of times borrower has been in a delinquent state at 90 days past due. |

| | | |
|---|---|---|
| NumberRealEstateLoansOrLines | numeric | Number of mortgage and real estate loans including home equity lines of credit |
| NumberOfTime60-89DaysPastDueNotWorse | numeric | Number of times borrower has been 60-89 days past due but no worse in the last 2 years. |
| NumberOfDependents | numeric | Number of dependents in family excluding themselves (spouse, children etc.) |

For all datasets, missing data is handled on an algorithm-specific basis, according to the ability of the algorithm to handle missing values. For the deep learning and logistic regression algorithms, which are unable to handle missing values, mean imputation is used. Mean imputation involves replacing missing values with the mean value of the missing variable. For the tree-based algorithms – extreme gradient boosting, random forest and gradient boosting machine – missing values are treated in the same manner as categorical variables. This allows for missing values to be treated as "missing by design" rather than missing at random.

Categorical variables are encoded as dummy variables though one-hot encoding, a common technique where each distinct value of a categorical variable is converted to a new binary variable in the dataset. This variable will be set to one if an observation belongs to the group indicated by the column and zero otherwise.

## *3.3 Classification Techniques*

This study utilises the logistic regression (*glm*), neural networks (*deeplearning*), random forest (*randomForest*), gradient boosting machine (*gbm*), extreme gradient boosting (*xgboost*) and stacked ensemble algorithms (*ensemble*) in both the general analysis and learning curve analysis. Logistic regression is included as it is considered to be the industry standard, valued for its accuracy, simplicity and interpretability (Xia et al., 2017). Due to its position as the traditional classification algorithm in credit scoring, logistic regression is used widely in benchmarking studies in the literature. The other algorithms are chosen due to their performance as seen in other studies in the literature. Whilst this relatively small set of algorithms is not exhaustive of the models used in the literature it does represent algorithms which have been shown to perform at the highest level in other studies. (Lessmann et al., 2015) (Xia et al., 2017) (West, 2000).

## 3.4 Performance Measures

This study makes use of six different performance metrics by which model performance can be evaluated – the area under the ROC curve, area under the Precision-Recall curve, F1, F0.5, F2, Kolmogorov-Smirnov statistic, and accuracy. These metrics are chosen to represent a range of scenarios which a credit modeller is likely to have to make trade-offs in applying a model to a problem, taking into account any real-world context.

Area under the ROC curve (*auc*), area under the precision-recall curve (*auc_pr*), F-measure with $\beta = 1$ (*f1)*, F-measure with $\beta = 2$ (*f2*), F-measure with $\beta = 0.5$ (*f0p5*), Kolmogorov-Smirnov statistic (*ks)* and accuracy (*acc*) are used as metrics to evaluate algorithm performance.

Area under the ROC curve is included as one of the most commonly used performance metrics in credit scoring, and a strong single measure of model performance across all possible thresholds (Kraus & Kuechenhoff, 2014).

In order to deal with cases in which the class distribution of the training dataset is imbalanced, area under the precision-recall curve is included as a performance metric, as it has been shown to be more robust in evaluating the performance of a binary classifier in an imbalanced dataset (Davis & Goadrich, 2006).

The three F-measure metrics are chosen to account for scenarios in which the modeller's preference for precision and recall might differ. *F1* weights precision and recall equally and is calculated according to the equation $F1 = 2(\frac{precision * recall}{precision + recall})$ (H2O.ai, 2021).

*F2* assigns more weight to recall than it does to precision, by penalising the model more for false negatives than false positives according to the formula $F2 = 5(\frac{precision * recall}{4{*}precision + recall})$ (H2O.ai, 2021)..

*F0.5* penalises false positives more than it does false negatives, prioritising precision over recall according to the formula $F0.5 = 1.25(\frac{precision * recall}{0.25 * precision + recall})$ (H2O.ai, 2021)..

The Kolmogorov-Smirnov statistic calculates the maximum distance between the true positive rate and the false positive rate of a model. It measures a model's ability to differentiate between the two classes in a binary classification problem and is often used in conjunction with AUC to evaluate model performance (H2O.ai, 2021) (Adeodato & Melo, 2016).

Accuracy is simply the number of correct classifications made by a model as a percentage of the total number of predictions made by the model: $accuracy = \frac{TP+TN}{N}$. Accuracy is the only metric included in this study that is calculated based on a single threshold – all other metrics present a measure of predictive performance over all possible thresholds (Louzada, Ara & Fernandes, 2016). Therefore, it is not included in the learning curve analysis, as differences in optimal threshold between models trained on different sized training samples make comparison difficult using accuracy as an evaluation metric.

## *3.5 Sampling*

When evaluating the performance of advanced classification algorithms, the application of the appropriate sampling methodology is exceptionally important. Machine learning algorithms are able to create very complex models that "overfit" to the data they are trained on, resulting in poor performance when generalising predictions to observations not included in the training sample (Ying, 2019). Therefore, in order to evaluate the performance of such models, predictive performance needs to be assessed on a data sample that has not been "seen" by the model being evaluated. The simplest and most widely used sampling methodology is the training/test dataset split, in which a dataset is randomly split into two subsets, with one being used to train the model (the "training" sample) and the other being used to evaluate prediction accuracy (the "test" sample). There is no overlap between these two samples, meaning no observation is common to both datasets.

When hyperparameter tuning of the model is done to improve model performance, a validation sample is also required - with the train/test/validation approach, the model is trained on the training dataset, and predictive accuracy is tested on the test dataset as in the train/test approach. Model parameters are then tweaked, or “tuned” to improve predictions on the test dataset. Because the parameters have “seen” the test dataset, a final check of the predictive accuracy of the model is required on the validation or “holdout” set which is unseen at any previous stage of model development to ensure valid results.

The ratio between training, test and (if used) validation datasets is determined by the size of the dataset - ideally, you want the training set to include as many observations as possible, with the constraint that the test and validation samples be sufficient to be representative of the population (Paliwal & Kumar, 2009).

Another commonly used sampling method is cross-fold validation, in which a sample is split into k separate sets, or “folds”. The model is then trained and evaluated k times, each time with a different subset being used as the test set, with the performance of the model being evaluated across all of the different folds. The value of k is determined by the amount of data available, with larger values of k only justified by large datasets. Cross-fold validation has the benefit of being more robust than the train/test methodology, as model performance has been evaluated over the entire dataset instead of a small sample which might suffer from some unseen bias (Abdou & Pointon, 2011).

For the purposes of this study, 5-fold cross-validation is used for model training and hyperparameter tuning. Because the model training process includes hyperparameter tuning, a validation sample is utilised for model evaluation, with an 80/20 split – in other words, eighty percent of the dataset is randomly sampled for the training dataset, with twenty percent being utilised for the validation sample.

For the general analysis, twenty percent of each dataset is randomly sampled prior to model training to create the holdout sample, with the remaining eighty percent being the training sample. For the learning curve analysis, a single holdout sample of twenty percent of the full dataset is sampled randomly prior to model training, with the models trained over all training sample sizes evaluated over the same holdout sample.

## 3.6 Software used

All data processing and modelling is done using the R programming language, specifically version 4.0.1. The *tidyverse* package version 1.3.0 provides the libraries *ggplot* (version 3.3.1) *dplyr* (version 1.0.0) which were used for all graphing and data preprocessing respectively. All modelling was done using the open-source library *h2o* (version 3.32.0.1).

# Chapter 4
# Empirical Results and Discussion

## *4.1 Outline*

This chapter presents the empirical results of the research conducted in this paper and details the findings thereof. It is presented in three sections: firstly, the general analysis, comprising of the performance of each algorithm for each performance metric and dataset, and the average ranking of algorithms calculated over three groups of datasets.

For the general analysis, algorithm performance is presented firstly according to the standard methodology of the literature, as the average performance rank according to the performance metrics over all datasets included in this study. In order to illustrate the general impact of dataset size on algorithm performance, this is repeated over two subsets of the datasets categorised by their size – small and large, according to the definitions provided by Lessmann et al. (2015).

The learning curve analysis is then presented, where subsamples between 1,000 and 100,000 observations in step sizes of 1,000 were drawn from the *gmc* and *lc* datasets and models trained on each subsample, in order to ascertain the marginal effects of dataset size on the performance of the various classification algorithms according to the specified performance metrics. All figures are rounded to the third decimal point.

## 4.2 Classifier Performance by Dataset

**Figure 4.2.1 Australia Performance by Classifier**

| algorithm | n_obs | auc | auc_pr | f1 | f2 | f0p5 | accuracy |
|---|---|---|---|---|---|---|---|
| glm | 553 | 0.941 | **0.951** | **0.915** | 0.916 | **0.934** | **0.927** |
| deeplearning | 553 | 0.939 | 0.948 | 0.898 | 0.916 | 0.929 | 0.912 |
| gbm | 553 | 0.938 | 0.945 | 0.902 | **0.924** | 0.907 | 0.912 |
| xgboost | 553 | 0.943 | 0.941 | 0.900 | 0.918 | 0.917 | 0.912 |
| randomForest | 553 | **0.944** | 0.948 | 0.903 | **0.924** | 0.904 | 0.912 |
| ensemble | 553 | 0.943 | 0.948 | 0.905 | 0.922 | 0.912 | 0.912 |

For the *australia* dataset – the smallest dataset at 553 observations – we observe that there is substantial variation in relative algorithm performance depending on performance metric. For AUC, the random forest algorithm has the best performance, whilst the gradient boosting machine algorithm has the worst. For the area under the precision-recall curve metric, logistic regression performs at the highest level, with extreme gradient boosting performing the worst. Logistic regression is also the best classifier according to the F1, F0.5 and accuracy metrics, with the gradient boosting machine and random forest algorithms tied for the top F2 score.

**Figure 4.2.2 German Performance by Classifier**

| algorithm | n_obs | auc | auc_pr | f1 | f2 | f0p5 | accuracy |
|---|---|---|---|---|---|---|---|
| glm | 800 | 0.585 | 0.387 | 0.480 | 0.686 | 0.456 | 0.715 |
| deeplearning | 800 | **0.625** | **0.437** | 0.494 | **0.706** | **0.492** | **0.740** |
| gbm | 800 | 0.573 | 0.373 | 0.468 | 0.682 | 0.417 | 0.710 |
| xgboost | 800 | 0.610 | 0.376 | **0.495** | 0.701 | 0.417 | 0.705 |
| randomForest | 800 | 0.601 | 0.393 | 0.482 | 0.685 | 0.475 | 0.710 |
| ensemble | 800 | 0.611 | 0.388 | 0.494 | 0.699 | 0.456 | 0.710 |

For the second-smallest dataset *german*, with 800 observations, deep learning outperforms the other algorithms on all but one of the evaluation metrics. According to AUC, area under the precision-recall curve, F2, F0.5, and accuracy, deep learning is the best classifier. For the F1 performance metric, extreme gradient boosting slightly outperforms the neural network.

**Figure 4.2.3 Taiwan Performance by Classifier**

| algorithm | n_obs | auc | auc_pr | f1 | f2 | f0p5 | accuracy |
|---|---|---|---|---|---|---|---|
| glm | 24,001 | 0.721 | 0.495 | 0.521 | 0.593 | 0.561 | 0.813 |
| deeplearning | 24,001 | 0.765 | 0.533 | 0.545 | 0.620 | 0.577 | 0.820 |
| gbm | 24,001 | 0.783 | 0.556 | 0.552 | 0.638 | **0.580** | **0.821** |
| xgboost | 24,001 | 0.779 | 0.553 | 0.549 | 0.639 | 0.572 | 0.819 |
| randomForest | 24,001 | 0.778 | 0.552 | 0.547 | 0.633 | 0.577 | 0.819 |
| ensemble | 24,001 | **0.784** | **0.562** | **0.553** | **0.640** | 0.579 | 0.820 |

For the *taiwan* dataset, the third largest at 24,001 observations, the ensemble method provides the highest performance according to the AUC, area under the precision-recall curve, F1 and F2 metrics. For the F0.5 and accuracy metrics, the gradient boosting machine algorithm outperforms the others.

**Figure 4.2.4 GMC Performance by Classifier**

| algorithm | n_obs | auc | auc_pr | f1 | f2 | f0p5 | accuracy |
|---|---|---|---|---|---|---|---|
| glm | 96,000 | 0.697 | 0.247 | 0.316 | 0.347 | 0.356 | 0.936 |
| deeplearning | 96,000 | 0.811 | 0.352 | 0.419 | 0.485 | 0.433 | 0.937 |
| gbm | 96,000 | 0.852 | 0.377 | 0.438 | 0.513 | 0.439 | 0.936 |
| xgboost | 96,000 | **0.867** | 0.402 | **0.454** | **0.537** | 0.461 | **0.938** |
| randomForest | 96,000 | 0.858 | 0.392 | 0.442 | 0.525 | 0.460 | 0.937 |
| ensemble | 96,000 | **0.867** | **0.408** | **0.454** | 0.535 | **0.466** | **0.938** |

For the *gmc* dataset, which is the second largest at 96,000 observations, the extreme gradient boosting and ensemble algorithms dominate across all performance metrics. Extreme gradient boosting achieves the highest F2 score and is tied in performance with the stacked ensemble algorithm for AUC, F1, and accuracy. The stacked ensemble algorithm achieves the highest performance according to the area under the precision-recall curve and the F2 performance metrics.

**Figure 4.2.5 Lending Club Performance by Classifier**

| algorithm | n_obs | auc | auc_pr | f1 | f2 | f0p5 | accuracy |
|---|---|---|---|---|---|---|---|
| glm | 96,001 | 0.639 | 0.298 | 0.378 | 0.566 | 0.324 | **0.800** |
| deeplearning | 96,001 | 0.637 | 0.294 | 0.377 | 0.564 | 0.321 | 0.799 |
| gbm | 96,001 | 0.641 | 0.299 | 0.377 | 0.570 | 0.322 | **0.800** |
| xgboost | 96,001 | 0.641 | 0.299 | 0.378 | 0.569 | 0.324 | **0.800** |
| randomForest | 96,001 | 0.630 | 0.287 | 0.371 | 0.564 | 0.313 | **0.800** |
| ensemble | 96,001 | **0.648** | **0.306** | **0.382** | **0.572** | **0.328** | **0.800** |

For the largest dataset, *lc*, the stacked ensemble method outperforms all the individual classifiers in all of the performance metrics except for accuracy, in which there is a tie between five of the six algorithms tested. Deep learning marginally underperforms on this metric.

Figures 4.2.1 through 4.2.5 illustrate some key findings of this research. Firstly, we can see that the primary premise on which this research is based – that the performance of different classification techniques is influenced by the characteristics of the dataset in which they are applied – is supported by these results. We observe that the relative performance of each algorithm varies over both dataset to which it is applied, and which performance metric is used to rank performance. In addition, we observe a trend in the relationship between the performance of certain algorithms and the size of the dataset on which they were trained and evaluated. For the larger datasets, the relative performance rank of the more complex algorithms such as extreme gradient boosting and stacked ensembles is high, whilst simpler techniques like logistic regression are more competitive for the smaller datasets. We also observe that the difference in performance achieved by the highest-performing classifiers relative to the lowest-performing varies substantially between datasets. Using AUC as an example, we observe that for the *lc* dataset, the difference between the top (0.648) and bottom-performing (0.63) classifiers is 0.018. For *gmc*, the difference is (0.867-0.697) 0.17, with differences of 0.063, 0.052 and 0.006 for the *taiwan*, *german*, and *australia* datasets respectively. For the dataset with the biggest difference between the top and bottom-performing algorithms according to AUC, *gmc*, this equates to a 24.4% improvement in AUC achieved by the extreme gradient boosting and stacked ensemble algorithms over the AUC achieved by logistic regression. By contrast, the dataset with the lowest difference between the top and bottom-ranked algorithm, *australia*, there is a 0.6% difference between the AUC of the random forest and gradient boosting machine classifiers. Despite this, the benchmarking methodology of reporting performance by average performance rank weights the outperformance of the top-ranked algorithm for both the *gmc* and *australia* datasets equally, with a rank of 1.

## 4.3 Average Classifier Performance Across Datasets

In line with the methodologies used in the majority of the literature reviewed in preparation for this study, the average ranking of the classifiers for each performance metric calculated over the datasets is presented next.

In order to calculate the average rank of the different algorithms across the datasets, the values corresponding to all the classifiers' performance on a single metric for each dataset was arranged in descending order (i.e., largest to smallest), with the highest performance being allocated a rank of 1 and the lowest a rank of 6. Standard competition ranking is used, therefore in the event of a tie, all algorithms that achieved the tying score received the highest possible rank. In other words, in the event of two algorithms which achieved the highest performance according to AUC, both would be allocated a rank of 1, with the next highest being allocated a rank of 3. This process is repeated across all specified datasets, with the average calculated as a simple mean over the datasets in question. This process is conducted over three groups of datasets – all five datasets included in this study, the datasets classified as large, and the datasets classified as small. The definition of "small" and "large" is provided by Perlich, Provost and Simonoff (2003) who defined small datasets as containing 24,000 observations or less, with large datasets containing more than 24,000 observations.

**Figure 4.3.1 Average Performance Rank Over All Datasets by Classifier**

| algorithm | auc | auc_pr | f1 | f2 | f0p5 | accuracy | avg_rank |
|---|---|---|---|---|---|---|---|
| glm | 5.0 | 4.2 | 4.0 | 5.0 | 3.6 | 3.0 | 4.1 |
| deeplearning | 4.2 | 3.6 | 4.4 | 4.2 | 3.2 | 2.8 | 3.7 |
| gbm | 4.0 | 3.8 | 4.0 | 3.2 | 3.8 | 2.4 | 3.5 |
| xgboost | 2.2 | 3.6 | 2.4 | 2.4 | 3.4 | 2.8 | 2.8 |
| randomForest | 3.6 | 3.4 | 4.0 | 3.6 | 4.0 | 2.6 | 3.5 |
| ensemble | **1.4** | **1.6** | **1.4** | **2.0** | **2.2** | **1.8** | **1.7** |

We begin by comparing the average rank of the classifiers across all the datasets used in this study, and find that over every performance metric, the stacked ensemble method is on average the best-performing algorithm with an average rank of 1.7. Extreme gradient boosting is the second-best, with an average ranking of 2.8, with random forests and gradient boosting machines tied for third with and average rank of 3.5. Deep learning is second from the bottom with an average rank of 3.7, with logistic regression bringing up the rear with an average rank of 4.1. These results correspond with the finding of Lessmann et al. (2015), who also found that ensemble methodologies performed the best, on average, across a range of different credit risk datasets. However, the studies published by Baesens et al. (2003); Desai et al. (1997); Srinivisan and Kim (1987); West (2000), and Lee et al. (2002) found that logistic regression was competitive with more advanced classification techniques, which is not reflected by its position in the ranking in this study.

This perhaps unexpected underperformance of logistic regression might be explained by referring to the research of Perlich, Provost and Simonoff (2003) showed that logistic regression performs better on smaller datasets than decision trees, but as the size of the dataset increased, the rate at which the decision tree was able to improve far outstripped that of the logistic regression model. Additionally, Louzada, Ara and Fernandes (2016) found that the majority of the studies in the literature on the subject were conducted using publicly available datasets (with four percent of all reviewed papers using the *german* and *australian* datasets used by this study). Furthermore, the research of Lessmann et al. (2015) showed that only seven percent of the studies they reviewed used datasets containing fewer than 1,500 observations, with only 15 percent using datasets containing more than 4,000 observations. The intersections of the findings of these three studies allows us to better understand the discrepancy between the results of this paper and the literature regarding the competitiveness of logistic regression as a classification technique relative to more sophisticated modelling techniques. In this paper, the majority of the datasets used for training models (three of the five, or some 60 percent) comprise of more than 24,000 observations, which makes them large datasets according to the definition provided by Perlich, Provost and Simonoff (2003) – thus, if the size of a dataset is expected to influence the performance of different algorithms, and logistic regression is known to perform better than more sophisticated techniques on small datasets, then we can expect our results to differ somewhat to the findings of the literature in this respect.

Thus, we can compare the average ranks of the competing algorithms over the different definitions of dataset – small and large – according to the definition outlined by Perlich, Provost and Simonoff (2003).

**Figure 4.3.2 Average Performance Rank Over Small Datasets by Classifier**

| **algorithm** | **auc** | **auc_pr** | **f1** | **f2** | **f0p5** | **accuracy** | **avg_rank** |
|---|---|---|---|---|---|---|---|
| glm | 4.5 | 2.5 | 3.0 | 4.5 | 2.0 | 1.5 | 3.0 |
| deeplearning | 3.0 | **1.5** | 4.0 | **3.0** | **1.5** | **1.5** | **2.4** |
| gbm | 6.0 | 5.5 | 5.0 | 3.5 | 5.0 | 2.5 | 4.6 |
| xgboost | 2.5 | 5.5 | 3.0 | **3.0** | 4.0 | 4.0 | 3.7 |
| randomForest | 2.5 | 2.0 | 3.5 | **3.0** | 4.0 | 2.5 | 2.9 |
| ensemble | **2.0** | 2.5 | **2.0** | **3.0** | 3.5 | 2.5 | 2.6 |

**Figure 4.3.3 Average Performance Rank Over Large Datasets by Classifier**

| algorithm | auc | auc_pr | f1 | f2 | f0p5 | accuracy | avg_rank |
|---|---|---|---|---|---|---|---|
| glm | 5.3 | 5.3 | 4.7 | 5.3 | 4.7 | 4.0 | 4.9 |
| deeplearning | 5.0 | 5.0 | 4.7 | 5.0 | 4.3 | 3.7 | 4.6 |
| gbm | 2.7 | 2.7 | 3.3 | 3.0 | 3.0 | 2.3 | 2.8 |
| xgboost | 2.0 | 2.3 | 2.0 | 2.0 | 3.0 | 2.0 | 2.2 |
| randomForest | 4.3 | 4.3 | 4.3 | 4.0 | 4.0 | 2.7 | 3.9 |
| ensemble | **1.0** | **1.0** | **1.0** | **1.3** | **1.3** | **1.3** | **1.2** |

Splitting the performance of the algorithms over the performance metrics by the size of the datasets on which they were trained yields some interesting differences from the results obtained by averaging performance over all datasets. If we focus on the average ranking of the algorithms according to all performance metrics, we observe that the top-ranked algorithm differs between the small and large datasets. For small datasets, deep learning performs the best, with ensembles performing second best and random forests third, closely followed by logistic regression. For large datasets, we observe that the ensemble technique performs the best, followed by extreme gradient boosting and then gradient boosting machines. We also observe that the ranking of the algorithms over the performance metrics is far more consistent for the large datasets than the small ones, with ensemble models performing the best when measured by average rank according to all performance metrics for large datasets. For the small datasets, deep learning, extreme gradient boosting, gradient boosting machines and ensemble models all feature as the best-performing model as measured by average rank for at least one of the performance metrics.

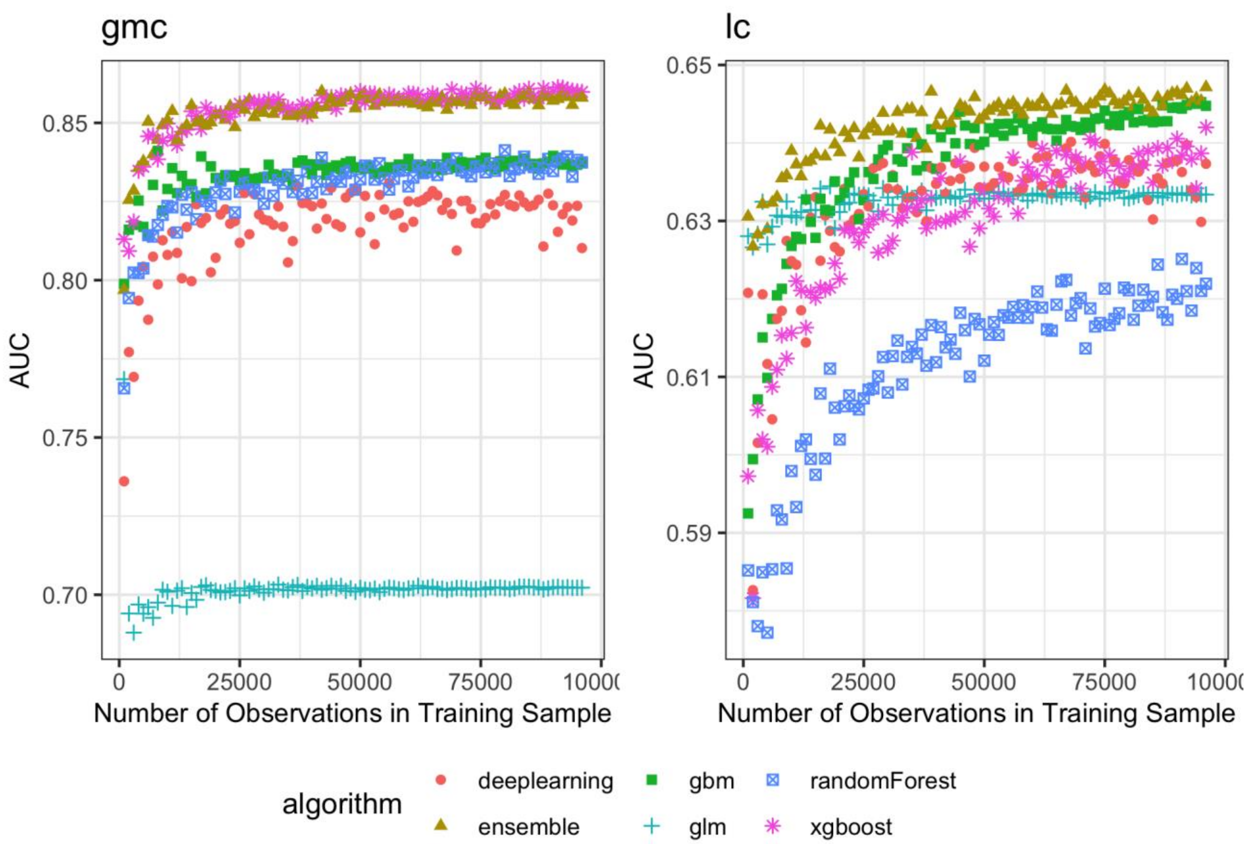
These findings conform to the conclusions drawn by Perlich, Provost and Simonoff (2003); that the size of a dataset has a significant effect on the performance of different types of classification algorithms trained on the dataset in question. This is a relevant finding for the field of research into credit risk prediction, where it has been shown that the datasets used in research tend to be small, and the majority of research is conducted on the same publicly available credit risk datasets (Lessmann et al., 2015; Louzada, Ara & Fernandes, 2016).

## 4.4 Learning Curves of Classification Algorithms by Performance Metric

This section presents the learning curve analysis; results obtained by training models on subsamples of data of different sizes drawn from the two largest datasets – the *gmc* and *lc* datasets. Observations for the subsamples are drawn at random with replacement, with 100 subsamples created ranging from 1,000 observations to 100,000 in increments of 1,000 (i.e., 1,000, 2,000, 3,000, …, 100,000). A model is trained for each classification algorithm on each distinct subsample, and the model performance of each resulting model according to each performance metric is evaluated using the same holdout sample.

The results are presented as charts where the performance of each classification algorithm for a single performance metric is represented by the vertical axis for the two datasets, with the size of the subsample on which the algorithm was trained on the horizontal axis. The vertical axes of the charts, representing the performance of the classifiers for each performance metric for the *gmc* and *lc* datasets, are not synchronised by design. This is due to the fact that the relative changes within each dataset as the size of the training sample increases is more important than the absolute performance for this analysis. We refer to this method of analysis as learning curve analysis in line with the methodology set out by Perlich, Provost and Simonoff (2003).

**Figure 4.4.1 Learning Curves by AUC over Training Sample Size**



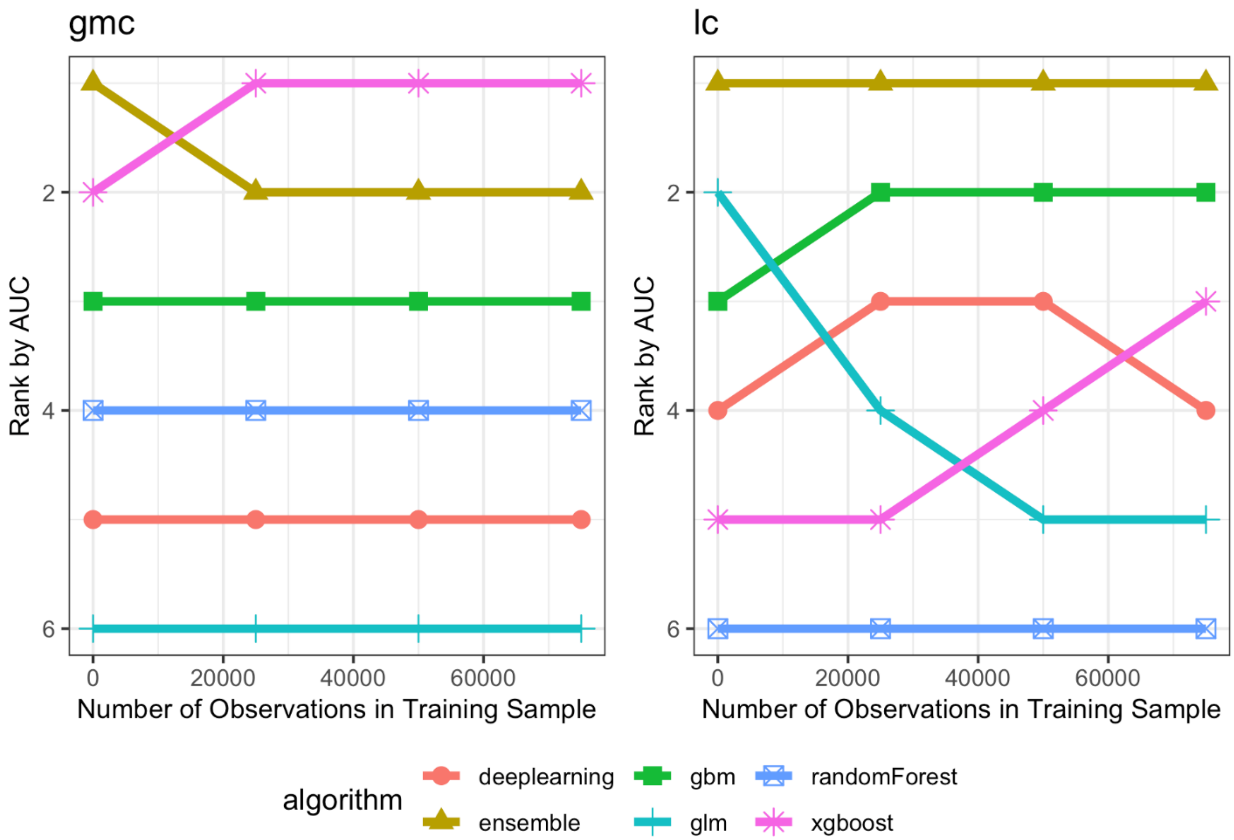

We begin with perhaps the most commonly-used performance metric for credit risk modelling, the AUC – or area under the Receiver Operating Characteristics (ROC) curve. Each point on the above graphs refer to the AUC of a model on the holdout sample, with the holdout sample being identical for all subsamples of the *gmc* and *lc* datasets.

Looking simply at the slopes of the curves implied by the points representing the AUC of each classification algorithm on the holdout samples, we can clearly observe that different algorithms "learn" at different rates. In other words, the marginal increase in AUC observed on the holdout sample as the size of the training dataset increases differs according to algorithm within a dataset. It also differs between datasets, implying that dataset size is not the only dataset characteristic that influences the performance of a classification algorithm. However, this is beyond the scope of this study and is left to future research to elaborate on.

Focussing on the learning curves of the competing algorithms as measured by AUC for the *gmc* dataset, we can see how size of the training sample has a strong effect on the non-linear algorithms – i.e., random forests, deep learning, extreme gradient boosting, gradient boosting machines, and the ensemble method. We observe a strong positive relationship between AUC and training sample size, with a gradient that is steeper at smaller values of training sample size and flattens out as training sample size increases. However, the slope of the implied curves remains positive at the top end of the measured values of training sample size, which suggests that these algorithms can continue to learn; performance on the AUC metric can be expected to improve if more data were available on which to train models. By contrast, for the linear algorithm of logistic regression, there is some performance improvement on the AUC metric for small training sample sizes, but from upwards of approximately 15,000 observations there is little to no improvement in AUC measured on the holdout sample, as shown by the plateauing of the implied curve. The implied curve at the top end of measured training sample sizes has a flat gradient, which implies that logistic regression can "learn" no more on this particular dataset – performance will not improve even if more data were available on which to train the algorithm.

For the *lc* dataset, we observe similar results to those discussed for the *gmc* dataset – the non-linear classification algorithms demonstrate a greater propensity to improve in performance on the AUC measured on the holdout sample, whilst the learning curve of the linear algorithm, logistic regression, plateaus at a training sample size of approximately 25,000 observations.

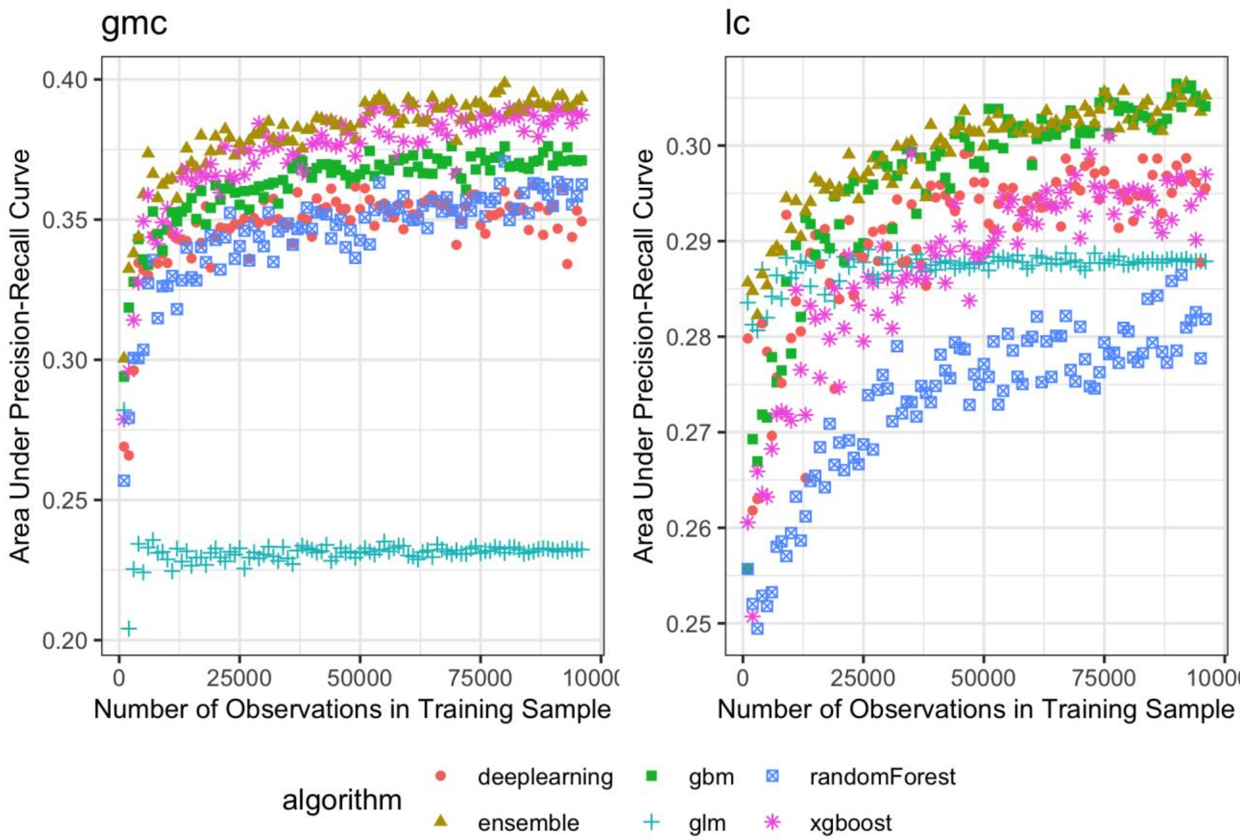
**Figure 4.4.2 Performance Rank by AUC over Training Sample Size**




In order to help evaluate the relative performance of the competing algorithms, we smooth the results presented in the learning rate analysis by binning the evaluated training sample sizes to bins 25,000 observations wide – i.e., four bins corresponding to 1 to 25,000, 25,001 to 50,000, 50,001 to 75,000 and 75,001 to 100,000 observations. The rank of the average performance of each classification algorithm is calculated over a given performance metric within each evaluated training sample set size. We observe that for the *gmc* dataset with performance ranked by AUC, the ensemble algorithm is the highest performing classifier for the smaller training samples of up to 25,000 observations. The extreme gradient boosting algorithm is the second-best performing algorithm for the smaller training samples but improves to be the best-performing algorithm for training samples larger than 25,000 observations. The gradient boosting machine algorithm has the third-best performance on the AUC metric for all training sample size bins, with the random forest classifiers in fourth place for all bins. Both the deep learning and logistic regression classifiers algorithms, ranking fifth and sixth (and last) respectively, maintain this ranking across all sample size bins.

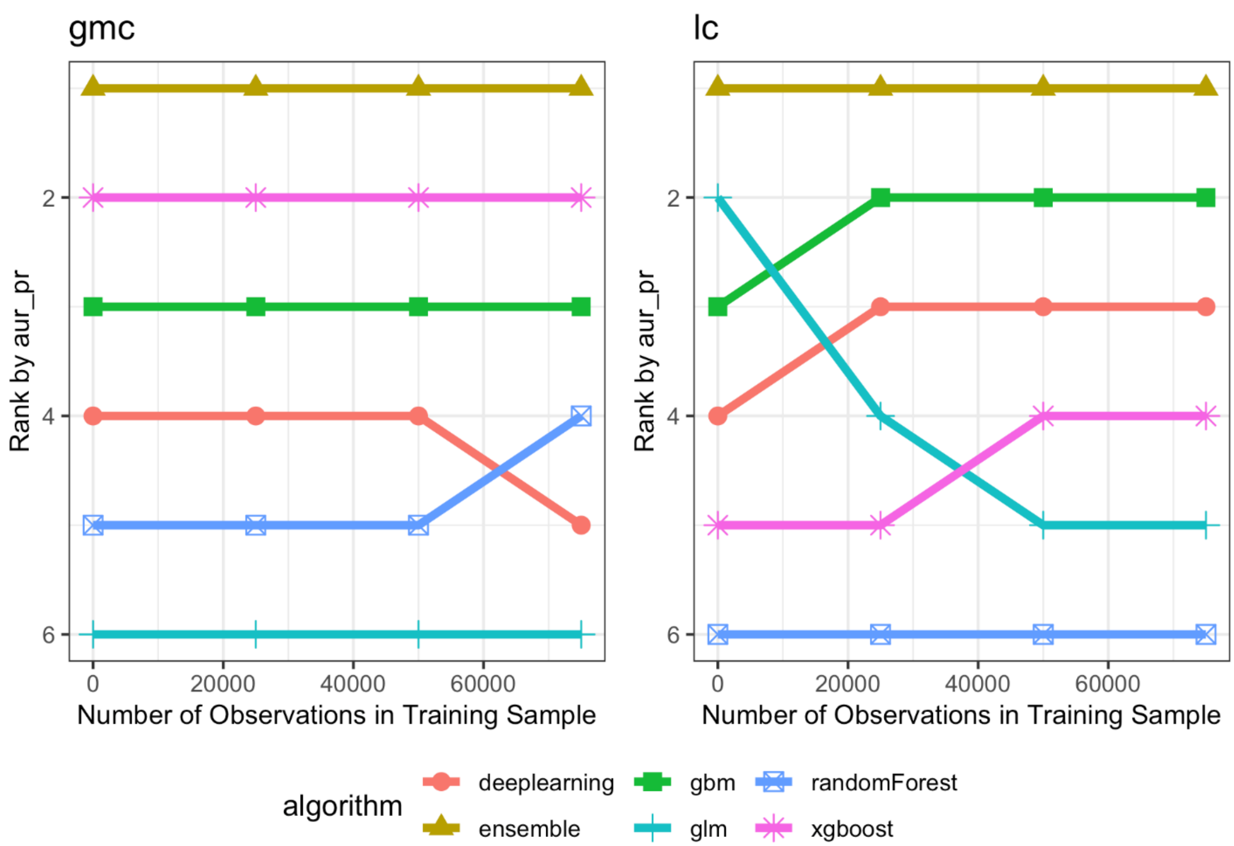
For the *lc* dataset, we observe very different results for the performance ranking by AUC to those observed for the *gmc* dataset. With the ensemble classifiers rank first for all the training sample bins, we see a significant decline in the relative performance of logistic regression as we move from the smaller to the larger training sample bins, falling from a rank of two to one of five from the smallest to the largest bin. The extreme gradient boosting algorithm improves its ranking the most, going from the second-lowest rank of five to a rank of three from the smallest to the largest training sample bin. The gradient boosting machine classifier improves one rank, from third to second, whilst the deep learning algorithm makes gains in the two middle training sample bins, improving from a rank of four to one of three before being overtaken by extreme gradient boosting in the largest training sample bin. The random forest classifiers maintain their rank as the worst-performing by AUC for all training sample bins.

**Figure 4.4.3 Learning Curves by Area Under the Precision-Recall Curve over Training Sample Size**




The learning curves for the Area under the Precision-Recall Curve performance metric for the *gmc* and *lc* datasets are presented in Figure 4.4.3, above. We observe that for both datasets, the logistic regression algorithm is more limited in its ability to leverage larger training datasets to improve predictive performance than the other algorithms being evaluated. Whilst performance improves for relatively small training dataset sizes – up to approximately 12,500 for the *gmc* dataset and 25,000 for the *lc* dataset – there is no observable improvement as the training datasets increase in size past these points. By contrast, the other classification techniques all continue to improve according to the Area under the Precision-Recall curve metric for the full range of training dataset size.

**Figure 4.4.4 Performance Rank by Area Under the Precision-Recall Curve over Training Sample Size**




Whilst we observe a similar inability of logistic regression to improve prediction performance as training dataset size increases across the *gmc* and *lc* datasets, the actual performance of logistic regression differs substantially across the two datasets. For the *gmc* dataset, logistic regression is the worst performing algorithm even at the smallest training dataset size, where its performance is closest to its competitors. As the other algorithms leverage additional data more effectively to improve performance according to the Area under the Precision-Recall Curve metric, its relative performance worsens and thus it remains in its position as the lowest-ranked algorithm. By contrast, logistic regression in the *lc* dataset shows very strong performance for the smallest training sample sizes, second only to the ensemble algorithm. As the training sample size increases, we observe that the gradient boosting machine, deep learning, and extreme gradient boosting algorithms improve their relative performance to improve from third, fourth and fifth-ranked algorithms to second, third and fourth-ranked respectively. The random forest algorithm has the lowest performance for all training sample sizes on the Area under the Precision-Recall Curve metric, despite substantial improvements in actual performance on this metric as more data became available for training.

**Figure 4.4.5 Learning Curves by F1 Score over Training Sample Size**

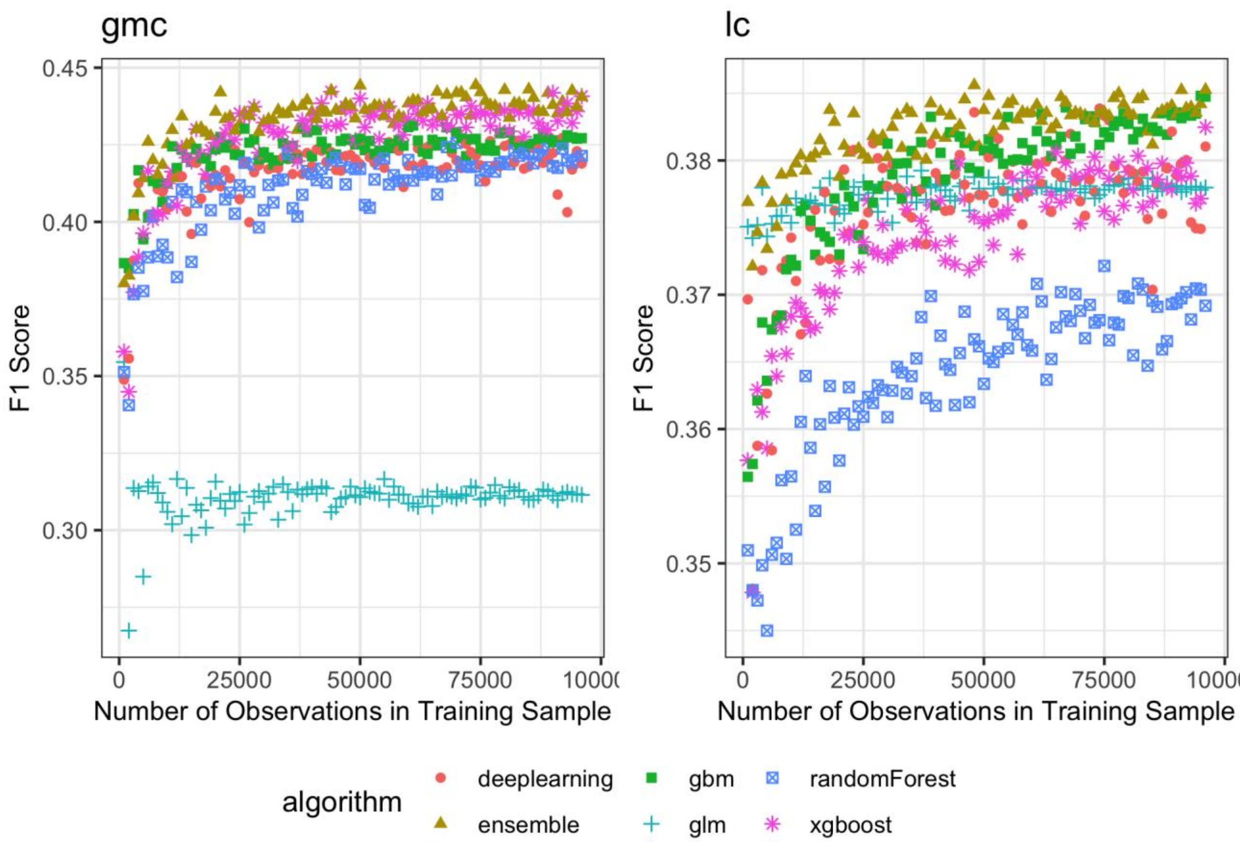
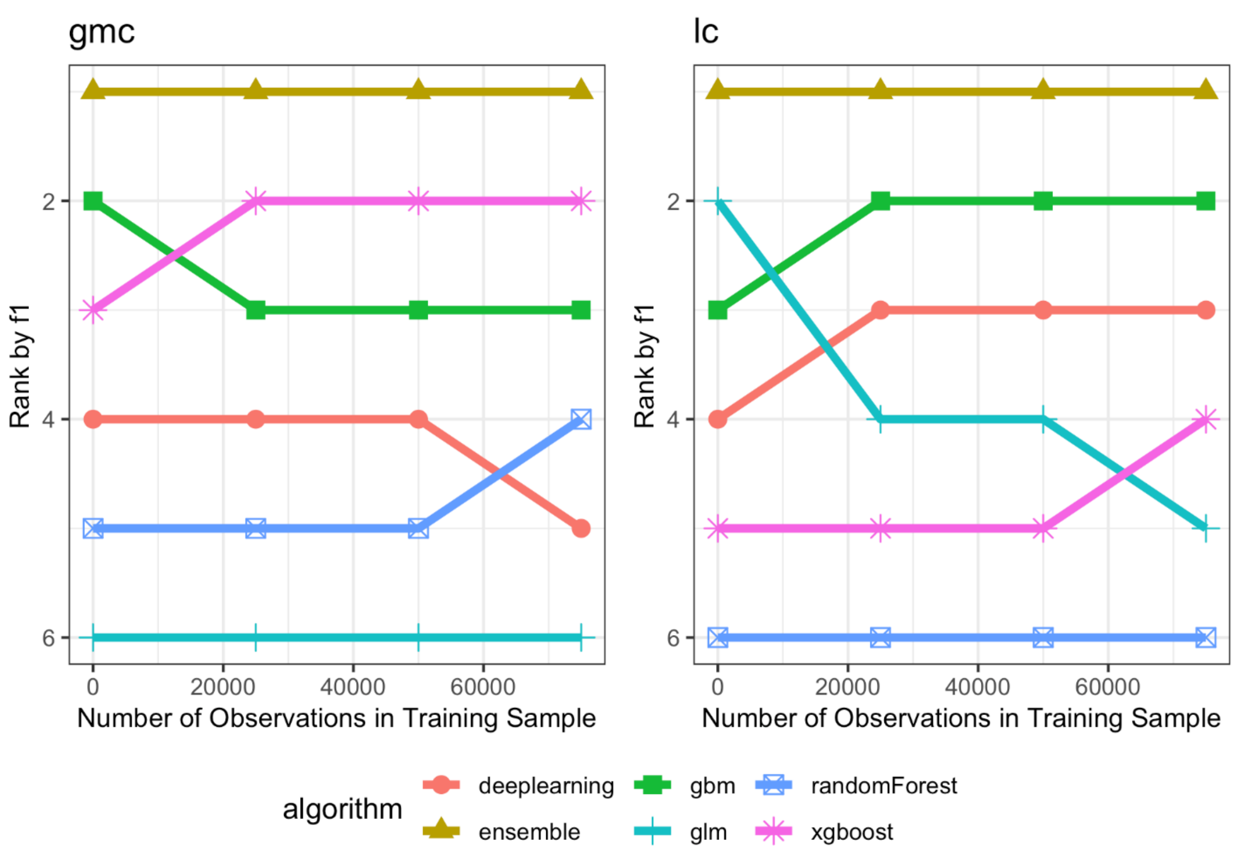



Figure 4.4.5 (above) presents the learning curves of the algorithms being evaluated in this study as measured by the performance of the algorithms according to the F1 performance metric as the sample size of the training dataset increases for the *gmc* and *lc* datasets. As was the case with the learning curves for the AUC and Area under the Precision-Recall Curve performance metrics, we observe that although the learning rates of the logistic regression algorithm is similar for models trained on the *gmc* and *lc* datasets, the relative performance of logistic regression is substantially better for the *lc* dataset than the *lc* dataset. As was the case with the previous two performance metrics, we observe that all the non-linear algorithms continue to improve the F1 score on the holdout dataset as they are trained on ever-increasing training samples. By contrast, logistic

regression, whilst able to make improvements with additional data over smaller training sample sizes, plateaus relatively early and ceases to improve on its performance as measured by F1 score.

**Figure 4.4.6 Performance Rank by F1 Score over Training Sample Size**




For both the *gmc* and *lc* datasets, the ensemble algorithm is the best performing across all training sample sizes. For the *lc* dataset, gradient boosting machines are ranked second for the smallest training samples, but extreme gradient boosting, ranked third for the smallest training samples, improves its relative performance by F1 score to be the second-best algorithm for the larger training dataset sizes, with gradient boosting machines falling to a rank of three. Deep learning has a rank of four for all but the largest training samples, where the random forest algorithm improves from rank five to four, with deep learning falling to fifth-best algorithm as measured by F1 score. Logistic regression is the worst performing algorithm according to F1 score for all training dataset sizes. For the *gmc* dataset, we observe logistic regression as the second-best predictor according to the F1 metric for the smallest training dataset sizes of up to

25,000 observations. Between training samples of 25,001 and 75,000 observations, it falls to fourth place, falling further to a rank of five between training sample sizes of 75,001 and 100,000 observations. The gradient boosting machine algorithm improves from its initial rank of three for training samples up to 25,000 observations, being the second-best performing classifier according to F1 score for training samples between 25,001 and 100,000 observations. Deep learning improves its relative rank of fourth for training samples up to 25,000 observations to third between 25,001 and 75,000 observations, but the improvement of the extreme gradient boosting algorithm from fifth for training samples up to 75,000 observations to third between 75,001 and 100,000 observations pushes deep learning back down to its initial rank of fourth. The random forest algorithm remains the worst-performing algorithm according to F1 score for all training dataset sample sizes.

**Figure 4.4.7 Learning Curves by F2 Score over Training Sample Size**

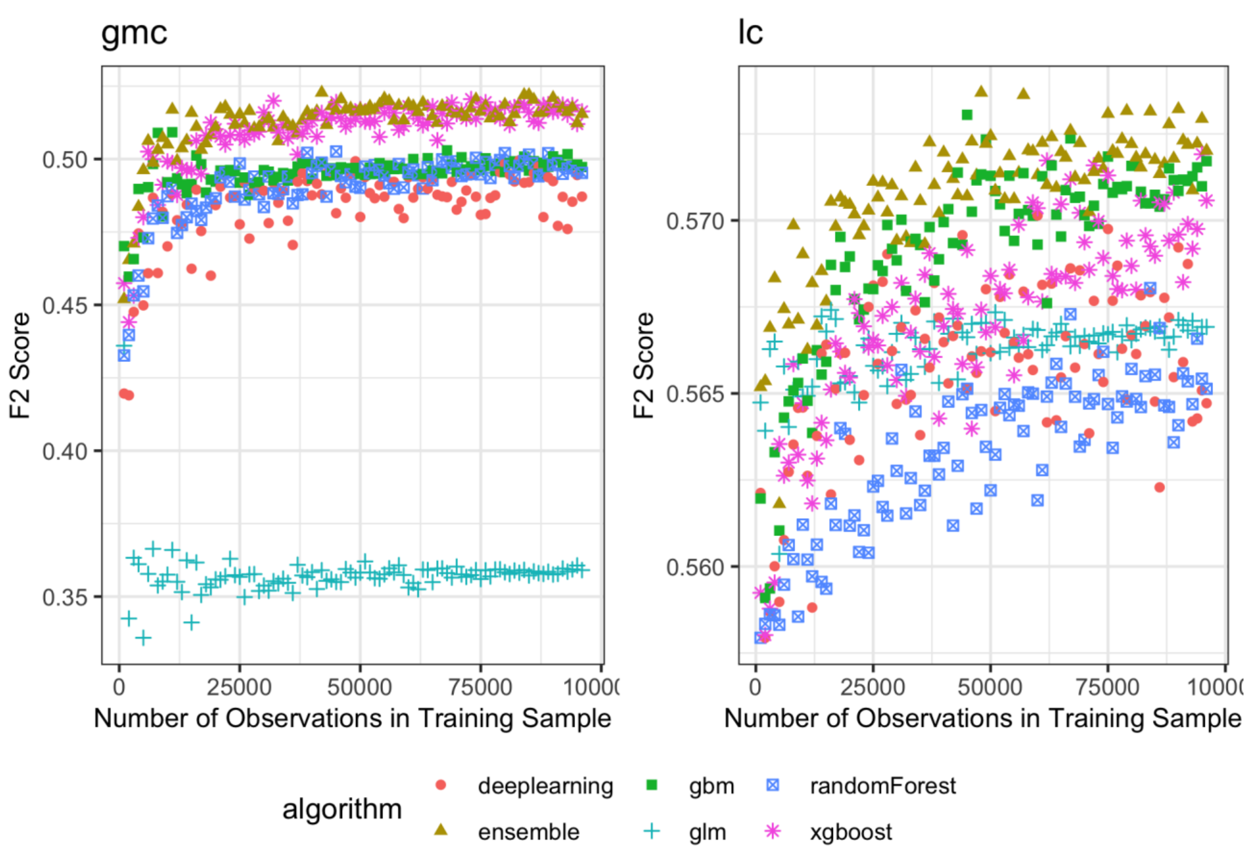
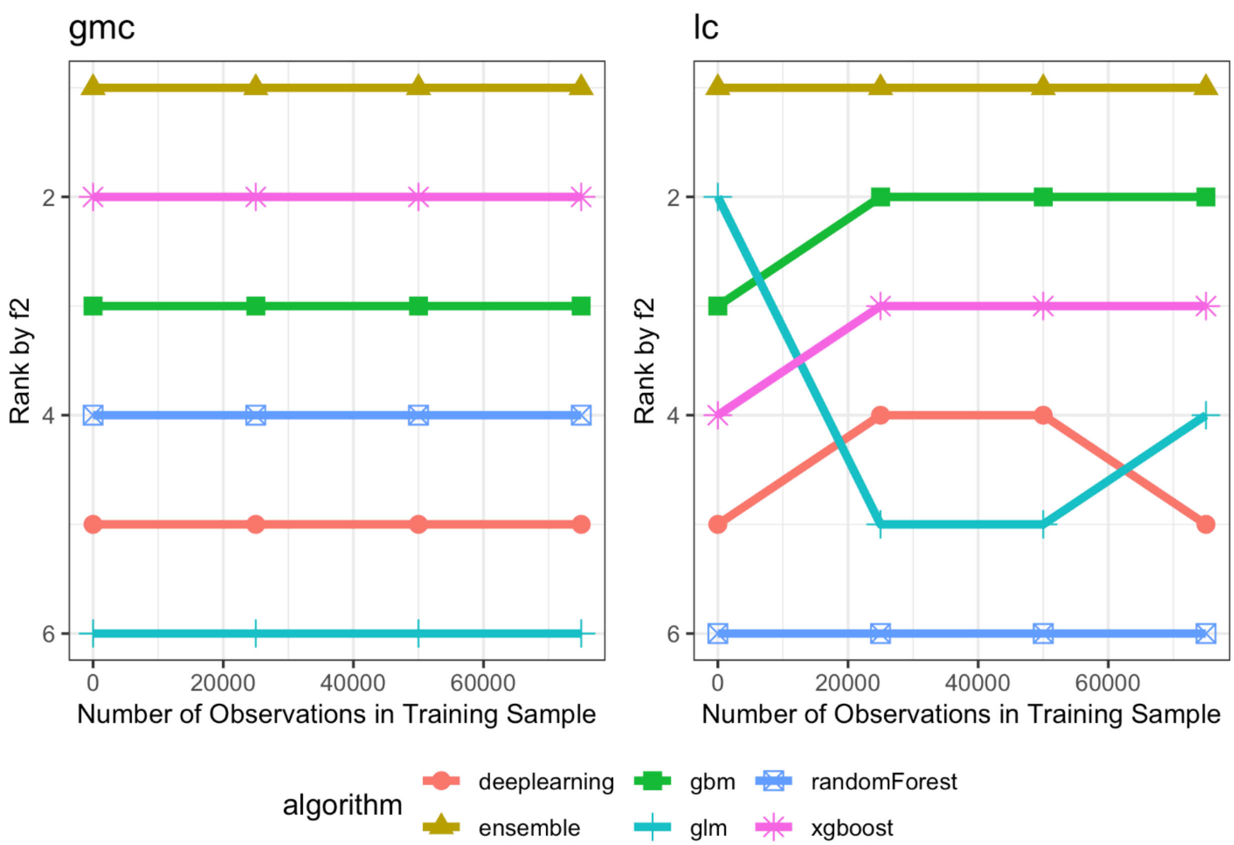



Figure 4.4.7 above presents the learning curves of the competing classifiers as measured by their F2 score performance on the holdout sample for the *gmc* and *lc* datasets. Again, we observe that

the learning curve of the logistic regression algorithm differs from the non-linear techniques, with performance plateauing at relatively low training samples whilst the other algorithms continue to improve their performance as measured by F2 score as more data is available on which to train a model. As was the case for the performance metrics already presented in this study, there is a large discrepancy between the relative performance of logistic regression between the *gmc* and *lc* datasets, with logistic regression returning F2 scores significantly lower than the other algorithms for the *gmc* dataset, whilst remaining competitive for the *lc* dataset.

**Figure 4.4.8 Performance Rank by F2 Score over Training Sample Size**




Looking at the relative performance according to the F2 score of the different classification algorithms at different training sample sizes for the two datasets presented in figure 4.4.8, we observe general stability for the *gmc* dataset. The ensemble algorithm has the highest F2 score at all training sample sizes, with the gradient boosting machine algorithms having the second-best performance. Deep learning has the third highest F2 score for training sample sizes between 1,000 and 25,000 observations, but for samples between 25,001 and 100,000 observations, the performance of the extreme gradient boosting algorithm improves from fourth place for the smallest training sample size bin to third, with deep learning falling to third. The random forest and logistic regression algorithms remain in fifth and sixth rank respectively for all training sample sizes.

**Figure 4.4.9 Learning Curves by F0.5 Score over Training Sample Size**

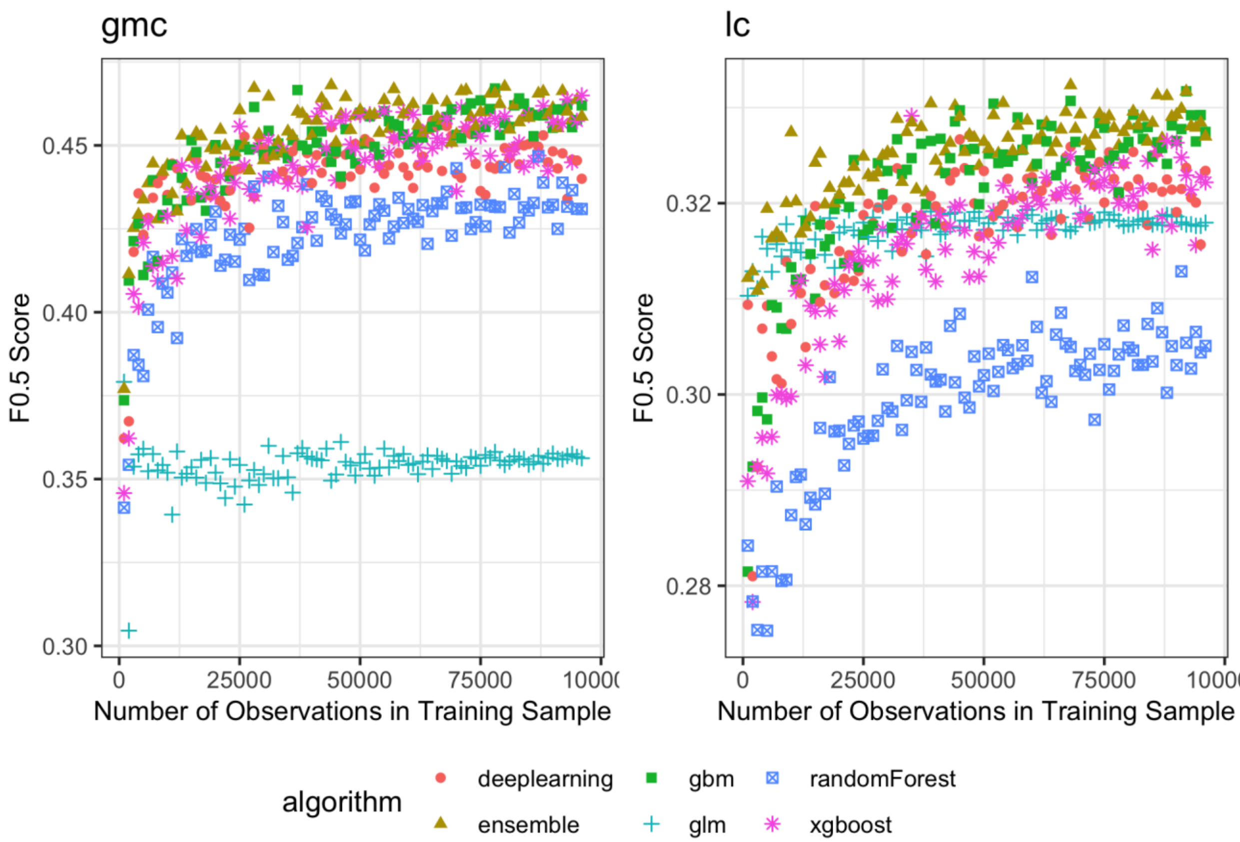
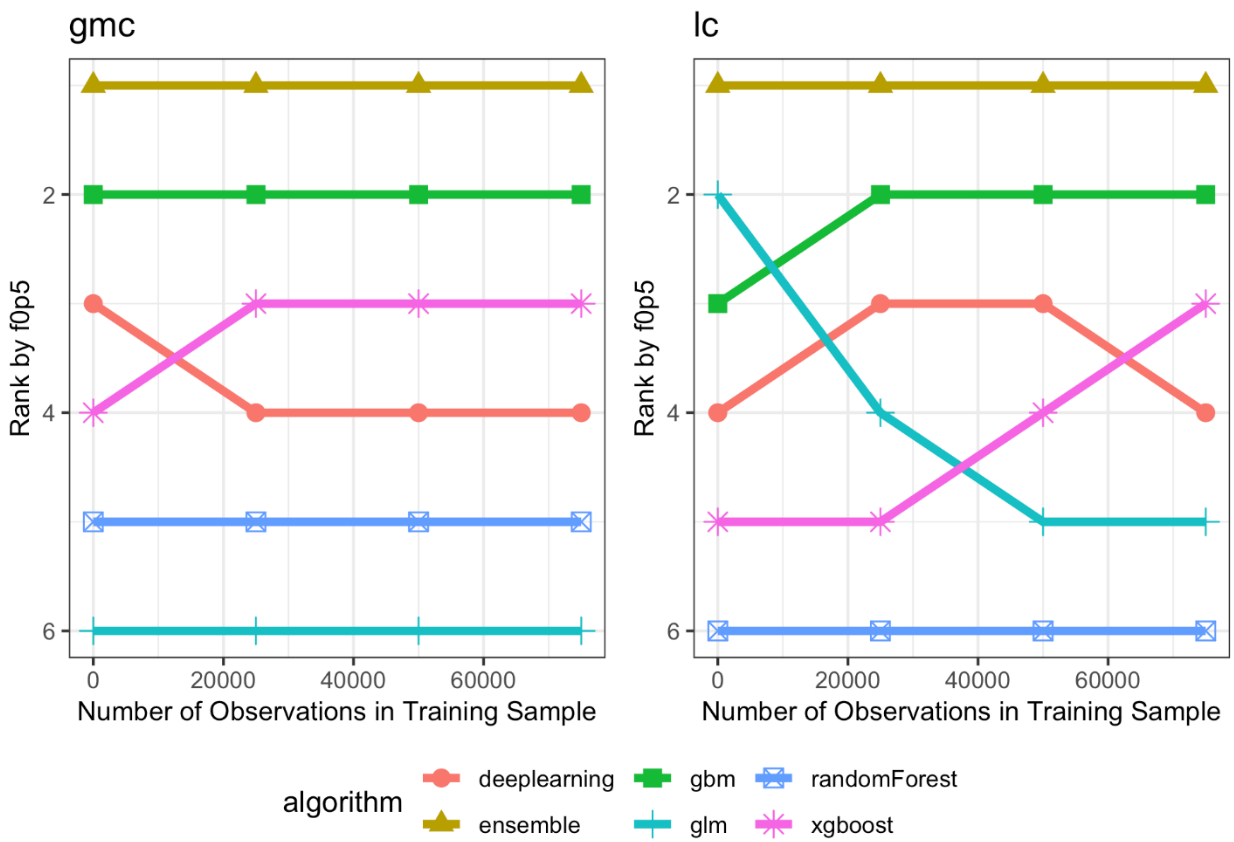



Figure 4.4.9 above presents the performance of the various algorithms evaluated in this study as measured by the F0.5 score over different training sample sizes for both the *gmc* and *lc* datasets. We confirm that the learning rates of the linear and non-linear classifiers – that is, the logistic regression algorithm and the other algorithms respectively – follow the same trends for the F0.5 score as for the previously presented performance metrics. That is, that the ability of the logistic regression algorithm to leverage bigger training samples to improve F0.5 score is lower than that of the other algorithms, as is evident by the plateauing of the logistic regression algorithm's learning curve whilst those of the other algorithms continue to have a positive gradient.

**Figure 4.4.10  Performance Rank by F0.5 Score over Training Sample Size**




The relative performance of the different classifiers on the F0.5 metric, presented in Figure 4.4.10 above, shows the difference in relative performance between the *gmc* and the *lc* datasets. Whilst the ensemble technique performs the best according to F0.5 score for all training sample sizes for both datasets, extreme gradient boosting yields the second-best performance for the *gmc* dataset for all training sample sizes, but for the *lc* dataset, it only has the fourth-highest ranking for training samples between 1,000 and 25,000 observations, improving to third-best between 25,001 and 100,000 observations. Gradient boosting machines perform well across both datasets, with a rank of second and third across all training sample sizes for the *gmc* and *lc* datasets respectively. The random forest algorithm, whilst yielding the fourth-highest F0.5 scores across all training sample sizes for the *gmc* dataset, is sixth-best (and last) for the *lc* dataset across all training sample sizes. Deep learning is ranked fifth for all training sample sizes for the *gmc* dataset, whilst for the *lc* dataset, it is also ranked fifth for training samples between 1,000 and 25,000 observations, improves to fourth between 25,001 and 75,000 observations, then drops back down to fifth between 75,001 and 100,000 observations. Logistic regression is the worst-performing algorithm according to F0.5 score for all training sample sizes for the *gmc* dataset. However, for the *lc* dataset, it is the third-best classifier according to F0.5 score for training samples between 1,000 and 25,000 observations, dropping to fifth ranked between 25,001 and 75,000 observations, and improving to fourth between 75,000 and 100,000 observations.

## 4.5 Discussion

This study comprised of two broad categories of analysis of the performance of different classification algorithms in credit risk prediction. Firstly, a general analysis, in line with the methods most commonly utilised in the literature on the subject, compared the average performance of the different classifiers across a range of datasets with varying characteristics. Secondly, a learning curve analysis was conducted, where subsamples of a range of sizes of the two largest datasets were created on which the different algorithms could be trained. Their performance on these subsamples was then presented as “learning rate” curves, allowing the inference of the effect of training dataset size on the predictive performance of the algorithms. Although learning curve analysis is not a novel technique, to the knowledge of this author there is no precedent for its use in the body of research on credit risk prediction.

The results of the general analysis, where the relative performance of the classification algorithms is evaluated by their average ranking over all the datasets included in this study, are broadly reflective of previously established results obtained by other researchers in the literature. As shown in Figure 4.3.1, the ensemble algorithm is the best performing classifier on average over all the datasets included in this study. This result is consistent with the findings of Lessmann et al. (2015) who also found ensemble techniques to be the highest performing on average over a range of credit risk datasets. The relative lack of competitiveness of logistic regression is a slight surprise, running counter to previous findings in the literature (Crook, Edelman & Thomas, 2007). This may be explained by the relatively high weighting of large datasets used in this study compared to the bulk of studies in the body of research on the subject to date (Louzada, Ara & Fernandes, 2016). However, looking at the performance of the classification algorithms on the individual datasets, presented in Figures 4.2.1 through 4.2.5, the findings of Crook, Edelman and Thomas (2007) can be seen to hold in the results of this study. This study also finds that there is no single "best classifier" for credit risk prediction regardless of circumstance, as is evident by the substantial variance in relative performance of the various classification algorithms between datasets, and according to different performance metrics on the same dataset.

Figures 4.3.2 and 4.3.3 present the relative performance of the classification algorithms, calculated as the average performance rank over a segment of the datasets included in this study, defined by the size of the dataset. Datasets comprising of 24,000 or less observations were classified as "small", whilst those over 24,000 observations were classified as "large". Through segmenting the general analysis according to dataset size, it is apparent that, whilst it remains true that there is no "best" classification algorithm for credit risk prediction problems as established by Crook, Edelman and Thomas (2007), there is a clear relationship between the performance of certain algorithms and the size of the dataset on which they were trained and evaluated. In particular, we see that logistic regression, often regarded as the "traditional" credit modelling algorithm, performs significantly better on smaller datasets relative to the more complex models that this study compares it against than it does on larger datasets. This is important given the context of the literature on credit risk prediction. Lessmann et al. (2015) found that of the 48 studies they reviewed on the subject, some 75 percent used training sets of 1,500 observations or less to train their models. Furthermore, Louzada, Ara and Fernandes (2016) found that nearly half (45 percent) of the 187 studies they reviewed on the subject included the publicly available *german* and a*ustralian* datasets as training datasets (both the *german* and *australian* datasets were included in this study, and were the datasets where logistic regression was more competitive with the more complex algorithms).

To reiterate these two findings from the literature for emphasis: there is substantial homogeneity in the choices of datasets used in the literature on the performance of different algorithms in credit risk prediction, and the datasets utilised by researchers in the literature overwhelmingly skew towards datasets of 1,500 observations or less. This study found in its general analysis that the performance of different algorithms varied greatly between the different datasets used as training samples, and that the size of the training samples had a significant impact on the relative performance of the different classification algorithms. Clearly, these results are of some consequence to the lens through which the results of the body of literature on algorithm performance in credit risk prediction is viewed. For example, if we were to scrutinise the finding that logistic regression is relatively competitive compared to more sophisticated classification techniques in credit risk prediction problems (Xia et al., 2017), it stands to reason to question whether this finding is overstated due to the high prevalence of small training datasets in the literature as stated by Louzada, Ara and Fernandes (2016). This, combined with the finding of this study that logistic regression tends to perform better relative to more advanced algorithms on smaller datasets (remembering that the definition of a small dataset used in this study – 24,000 or fewer observations – is substantially larger than the 1,500 observation threshold that 75% of the studies reviewed by Lessmann et al. (2015) fell under), implies that the findings in the literature might be inflating the relative performance of logistic regression. In a business context, where datasets tend to be far larger than those used by researchers in the literature, the validity of findings prevalent in the literature might be called into question (Lessmann et al., 2015).

This study delves deeper into the relationship between dataset size and the performance of different classification algorithms, employing learning curve analysis to investigate how performance on a range of performance metrics is influenced by the size of the sample on which an algorithm is trained. Using the two largest datasets included in this study – the *lc* and *gmc* datasets each comprising of over 100,000 observations – samples from 1,000 to 100,000 observations were randomly drawn from the full datasets in increments of 1,000. Each of these subsamples of the *lc* and *gmc* datasets were used as training samples for the six algorithms included in this study, with the performance of each model being evaluated on six performance metrics, using the same holdout sample for model evaluation for all subsamples.

The performance of each algorithm on each distinct performance metric is presented on a chart with the horizontal axis corresponding to the size of the training sample, and the values on the vertical axis corresponding to the performance of a classification algorithm according to the performance metric presented by the chart. This allows for the creation of implicit "learning curves", which convey the effectiveness of each algorithm at "learning" – in other words, improving on an evaluation metric as the size of the dataset on which it is trained increases. The effectiveness of an algorithm's ability to leverage additional observations to learn can be ascertained through the slope of the curve – steeper curves imply a greater ability to leverage large datasets to improve performance whilst flatter curves imply that leveraging additional data does not result in improved model performance. Because the relative performance of the algorithms is of interest in this study, the relative ranks of the algorithms are also presented to show how the different classifiers interact with training sample size over the different performance metrics. Due to somewhat noisy data and the performance of many of the algorithms being very close in some cases, the ranks were calculated as the rank of the average performance for each performance metric within bins representing ranges of training sample size. These bins were from 1,000 to 25,000 observations, 25,001 to 50,000 observations, 50,001 to 75,000 observations and 75,001 to 100,000 observations.

The learning curve analysis is clear in demonstrating the difference between traditional and non-traditional credit modelling techniques – i.e., logistic regression relative to the rest of the algorithms – in leveraging additional observations to drive improvement in prediction performance. For both datasets and across all of the performance metrics included, the slope of the learning curve for logistic regression plateaued at relatively low training sample sizes, whilst the other algorithms' performance continued to improve past this point. Continuing the theme of performance of the logistic regression algorithm, we observe a clear discrepancy in the competitiveness of logistic regression versus the other classifiers between the two datasets. For the *gmc* dataset, logistic regression is consistently the worst-performing of the algorithms across all evaluation metrics even at small training sample sizes. As the number of observations in the training samples increase, the relative advantage of the other algorithms in leveraging additional data to improve performance increases the gap between logistic regression and the other algorithms. By contrast, for the *lc* dataset, logistic regression is very competitive at low training sample sizes across all evaluation metrics. However, similarly to the results observed in the *gmc* dataset, the non-traditional algorithms are able to leverage additional observations more effectively than the traditional technique, resulting in it becoming less competitive at larger training sample sizes.

Along with logistic regression, we observe differences between all the algorithms except for ensemble models between the datasets in the learning curve analysis, with the relative performance also varying according to performance metric. This is relevant to the discussion around the tendency of researchers to present the results of studies on the relative performance of different classification algorithms in credit risk modelling as average performance over a number of different training datasets. This practice, used to improve the robustness of results by lessening the potential bias introduced by a single dataset, might actually be masking the bias introduced by the choice of datasets used to train the models. As the results presented in this study show, training sample size has a significant effect on the performance of different classification algorithms. This finding is consistent across different datasets and evaluation metrics. Thus, when presenting the relative prediction performance of a set of algorithms as the average of their performances over multiple training datasets, we can infer that the choice of datasets included in the study will influence the results of the study. If a researcher were to choose predominantly small datasets, as Lessmann et al. (2015) found to be the case for some 75 percent of studies they reviewed, then, if the trends of the results obtained by this study hold, results could differ from those obtained using larger datasets. Logistic regression could be expected to perform relatively well and algorithms that are able to leverage additional data more effectively, like extreme gradient boosting, gradient boosting machines, and deep learning, might underperform relative to expectations set by the success of these algorithms in non-credit risk contexts. Additionally, the preponderance of a relatively small number of publicly available datasets used by researchers in the literature undermines attempts to reinforce the robustness of results by averaging performance over a range of datasets. If these datasets biased algorithm benchmarking results, we might expect this bias to propagate throughout the body of literature due to their prevalence in the studies of which it is comprised.

In addition to the finding that the size of the training sample is a significant factor in algorithm performance, the results of this study indicate that additional characteristics of a dataset influence the performance of algorithms trained on the data in question. From the differing relative performance of algorithms between the *lc* and *gmc* datasets at the same training sample size – e.g., at training samples of 1,000 observations, logistic regression is the worst-performing of the algorithms according to the AUC metric for the *gmc* dataset, but the second best (and best of the individual classifiers) for the *lc* dataset. For the newest algorithm included in this study, extreme gradient boosting, we observe better performance relative to the other algorithms for the *gmc* dataset in comparison to the *lc* dataset across all performance metrics except for the Kolmogorov-Smirnov statistic. The difference in performance of the logistic regression and extreme gradient boosting algorithms between the two datasets – specifically the fact that the ranking of the algorithms differs so much between datasets controlling for training sample size – suggests that other dataset characteristics play a role in model performance. Whilst investigating dataset characteristics aside from training sample size fall outside the scope of this study, there is certainly scope for future research to delve into how such characteristics influence the performance of classification algorithms in a credit risk context. Given the evolution of credit scoring, particularly in the realm of digital lending and big data, and the ever-increasing importance of so-called "alternative data" in credit scoring, it would benefit the literature to establish how the following factors might interact with different algorithms and influence performance:

- Dataset dimensionality (i.e., the number of variables available for inclusion in a model)
- The quality of the "predictiveness" of a dataset (i.e., how useful the variables included in a dataset are for predicting credit default. Whilst the rise of alternative data sources has allowed for great expansion into previously unserved credit markets in the past decade, it remains true that not all data is created equal in predictive modelling)
- The underlying relationship of explanatory variables to the dependant variable, particularly whether this relationship is linear or non-linear (whilst difficult using real data, this can be achieved using simulated data)

Over all of the training sample sizes for both the *gmc* and *lc* datasets, the ensemble algorithm is the highest-performing algorithm for nearly all performance metrics by some distance. The ensemble algorithm has the top performance rank according to all performance metrics except for AUC for training samples over 25,001 observations for the *gmc* dataset and below 50,000 observations for the Kolmogorov-Smirnov statistic for the *lc* dataset and looking at the learning curves for these it is apparent that its performance on these metrics is very similar to the top-ranked algorithm. The finding that the ensemble algorithm is the best-performing classification algorithm is consistent with the findings of Lessmann et al. (2015) and with the results obtained in the general analysis presented earlier in this paper. By extending the analysis to take the size of the training sample into account, the finding of the relative outperformance of the ensemble algorithm can be argued to be robust to questions about the context of the datasets included in a study. However, the context of the ensemble algorithm itself becomes important if we make the claim that, on average, it is the best-performing classification technique in credit risk prediction. For an ensemble model, the individual models that serve as inputs to the ensemble are not fixed, being a choice of whoever is doing the modelling. In this study, the individual models which were “stacked” in order to create the ensemble were simply all the other models included in the study. This explains why the ensemble model was able to perform at such a high level over all training sample sizes, as the weighting of each individual model can be adjusted based on its performance – the stacked ensemble model is able to leverage strong individual algorithm performance selectively.

# Chapter 5

# Conclusion

This study has sought to build on the existing body of research into the use of advanced statistical techniques in credit scoring – specifically, probability of default (PD) scoring at loan application. It has conducted a review of the literature to ascertain the established findings of researchers in the field and highlighted the fact that whilst a great deal of research has been done on benchmarking the performance of different classification algorithms, there has been less focus on researching the datasets themselves and what effect their characteristics might have on algorithm performance. It pointed to a potential gap for research that investigated how the size of training datasets affect algorithm performance, given the research of Lessmann et al. (2015) who found that 75 percent of studies on benchmarking classification algorithms for credit scoring used datasets comprising of 1,500 or fewer observations.

This study was initiated on the hypothesis that if it could be established explicitly that there is a relationship between dataset size and algorithm performance, then we can argue that the choice of datasets included in a study by researchers bias the results of said study. Given the propensity of researchers in the field of algorithm benchmarking in credit risk prediction to choose very small datasets, we can extend this argument to suggest that the caution should be applied in applying the results of studies in the literature in different contexts to which they were obtained.

One common finding that might be called into question is that of the relative competitiveness of logistic regression with more sophisticated algorithms (Xia et al., 2017). This study has shown that logistic regression performs well when compared against more sophisticated techniques at small training sample sizes.

However, it has a comparative disadvantage in its ability to leverage additional training data effectively, showing poor marginal improvements in performance on all performance metrics relative to the other algorithms as training dataset size increases. For the 75 percent of studies reviewed by Lessmann et al. (2015), this poses a difficult question regarding the external validity of their results. In particular, we can highlight the issue of applying results obtained using small training datasets in a commercial setting, where datasets tend to be far larger.

Another phenomenon observed in this study is the effect of other dataset characteristics on algorithm performance. Although it was beyond the scope of this study to explore these in detail, we observed differing relative algorithm performance between different datasets of similar sizes. This can be attributed to characteristics of those datasets aside from size, which has been shown to influence algorithm performance. This observation also has implications for many of the studies that make up the body of literature. Louzada, Ara and Fernandes (2016) found that nearly half of the studies they reviewed contained either the *german* or *australian* datasets (or both), also included in this study. Given the observed effect of dataset characteristics other than size on algorithm performance, we can argue that the disproportionate use of these two datasets to benchmark classifiers has led to biased findings in the literature.

Whilst the findings of this study do not invalidate those obtained by other researchers on the subject, they do call for additional caution in the application of these findings. In most things, context is important, and it remains true for the performance of classification algorithms in credit scoring. In the field of credit risk scoring, the important context of dataset characteristics should not be ignored; taking factors such as dataset size into account when choosing a modelling strategy ensures best results.

# References

Abdou, H.A. 2009. Genetic programming for credit scoring: The case of Egyptian public sector banks. *Expert systems with applications.* 36(9):11402-11417.

Abdou, H.A. & Pointon, J. 2011. Credit scoring, statistical techniques and evaluation criteria: a review of the literature. *Intelligent Systems in Accounting, Finance & Management.* 18(2-3):59-88.

Adeodato, P.J. & Melo, S.B. 2016. On the equivalence between Kolmogorov-Smirnov and ROC curve metrics for binary classification. *arXiv.*

Akkoç, S. 2012. An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. *European Journal of Operational Research.* 222(1):168-178.

Ala'raj, M. & Abbod, M.F. 2016. Classifiers consensus system approach for credit scoring. *Knowledge-Based Systems.* 104:89-105.

Altman, E.I. 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The journal of finance.* 23(4):589-609.

Anagnostopoulos, C. 2012. Measuring classification performance: the hmeasure package. *UFIL.*

Anderson, R. 2007. *The credit scoring toolkit: theory and practice for retail credit risk management and decision automation.* Oxford University Press.

Baesens, B. 2003. Developing intelligent systems for credit scoring using machine learning techniques. Katholieke Universiteit, Leuven.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J. & Vanthienen, J. 2003. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the operational research society.* 54(6):627-635.

Boyle, M., Crook, J., Hamilton, R. & Thomas, L. 1989. *Methods for credit scoring applied to slow payers.* University of Edinburgh Department of Business Studies.

Breiman, L. 1996a. Bagging predictors. *Machine learning.* 24(2):123-140.

Breiman, L. 1996b. Stacked regressions. *Machine learning.* 24(1):49-64.

Chen, F.-L. & Li, F.-C. 2010. Combination of feature selection approaches with SVM in credit scoring. *Expert systems with applications.* 37(7):4902-4909.

Crook, J.N., Edelman, D.B. & Thomas, L.C. 2007. Recent developments in consumer credit risk assessment. *European Journal of Operational Research.* 183(3):1447-1465.

Crotty, J. 2009. Structural causes of the global financial crisis: a critical assessment of the 'new financial architecture'. *Cambridge journal of economics.* 33(4):563-580.

Dash, M. & Liu, H. 1997. Feature selection for classification. *Intelligent data analysis.* 1(1-4):131-156.

Davis, J. & Goadrich, M. Eds. 2006. The relationship between Precision-Recall and ROC curves. 233-240.

Desai, V.S., Conway, D.G., Crook, J.N. & Overstreet Jr, G.A. 1997. Credit-scoring models in the credit-union environment using neural networks and genetic algorithms. *IMA Journal of Management Mathematics.* 8(4):323-346.

Dietterich, T. 1995. Overfitting and undercomputing in machine learning. *ACM computing surveys.* 27(3):326-327.

Durand, D. 1941. *Risk elements in consumer installment financing.* National Bureau of Economic Research, New York.

Fawcett, T. 2006. An introduction to ROC analysis. *Pattern recognition letters.* 27(8):861-874.

Fisher, R.A. 1936. The use of multiple measurements in taxonomic problems. *Annals of eugenics.* 7(2):179-188.

Freund, Y. & Schapire, R.E. Eds. 1996. Experiments with a new boosting algorithm. Citeseer. 148-156.

Goadrich, M., Oliphant, L. & Shavlik, J. 2006. Gleaner: Creating ensembles of first-order clauses to improve recall-precision curves. *Machine learning.* 64(1-3):231-261.

H2O.ai. 2021. *Model Performance*. https://docs.h2o.ai/h2o/latest-stable/h2o-docs/performance-and-prediction.html#classification. [2021, 01 September].

Hand, D.J. 2009. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine learning.* 77(1):103-123.

Hand, D.J. & Henley, W.E. 1997. Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A.* 160(3):523-541.

Hand, D.J. & Till, R.J. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine learning.* 45(2):171-186.

Henley, W.E. 1995. Statistical aspects of credit scoring. The Open University.

Kraus, A. & Kuechenhoff, H. 2014. Credit scoring optimization using the area under the curve. *The Journal of Risk Model Validation.* 8(1):31.

Lahsasna, A., Ainon, R.N. & Teh, Y.W. 2010. Credit Scoring Models Using Soft Computing Methods: A Survey. *Int. Arab J. Inf. Technol.* 7(2):115-123.

Lee, T.-S. & Chen, I.-F. 2005. A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert systems with applications.* 28(4):743-752.

Lee, T.-S., Chiu, C.-C., Lu, C.-J. & Chen, I.-F. 2002. Credit scoring using the hybrid neural discriminant technique. *Expert Systems with applications.* 23(3):245-254.

Lessmann, S., Baesens, B., Seow, H.-V. & Thomas, L.C. 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research.* 247(1):124-136.

Linjordet, T. & Balog, K. Eds. 2019. Impact of Training Dataset Size on Neural Answer Selection Models. Springer. 828-835.

Louzada, F., Ara, A. & Fernandes, G.B. 2016. Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research Management Science.* 21(2):117-134.

MacDonald, S.B. & Gastmann, A.L. 2017. *A history of credit & power in the western world.* Routledge.

Malhotra, R. & Malhotra, D.K. 2003. Evaluating consumer loans using neural networks. *Omega.* 31(2):83-96.

Ong, C.-S., Huang, J.-J. & Tzeng, G.-H. 2005. Building credit scoring models using genetic programming. *Expert systems with applications.* 29(1):41-47.

Paliwal, M. & Kumar, U.A. 2009. Neural networks and statistical techniques: A review of applications. *Expert systems with applications.* 36(1):2-17.

Perlich, C., Provost, F. & Simonoff, J. 2003. Tree induction vs. logistic regression: A learning-curve analysis. *Journal of Machine Learning Research.* 4:211-255.

Ren, J. 2012. ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging. *Knowledge-Based Systems.* 26:144-153.

Schapire, R.E. 1990. The strength of weak learnability. *Machine learning.* 5(2):197-227.

Sokolova, M., Japkowicz, N. & Szpakowicz, S. Eds. 2006. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. Springer. 1015-1021.

Srinivisan, V. & Kim, Y. 1987. Credit granting a comparative analysis of classificatory procedures. *Journal of Finance.* 42(3):655-683.

Stephens, M.A. 1974. EDF statistics for goodness of fit and some comparisons. *Journal of the American statistical Association.* 69(347):730-737.

Sun, C., Shrivastava, A., Singh, S. & Gupta, A. Eds. 2017. Revisiting unreasonable effectiveness of data in deep learning era. 843-852.

Svozil, D., Kvasnicka, V. & Pospichal, J. 1997. Introduction to multi-layer feed-forward neural networks. *Chemometrics intelligent laboratory systems.* 39(1):43-62.

Thomas, L.C. 2000. A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International journal of forecasting.* 16(2):149-172.

Van Gestel, I.T., Baesens, B., Garcia, I.J. & Van Dijcke, P. Eds. 2003. A support vector machine approach to credit scoring. Citeseer. 73-82.

Wang, G., Hao, J., Ma, J. & Jiang, H. 2011. A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications.* 38(1):223-230.

West, D. 2000. Neural network credit scoring models. *Computers Operations Research.* 27(11-12):1131-1152.

Wolpert, D.H. 1992. Stacked generalization. *Neural networks.* 5(2):241-259.

Xia, Y., Liu, C., Li, Y. & Liu, N. 2017. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert systems with applications.* 78:225-241.

Yeh, I.-C. & Lien, C.-h. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications.* 36(2):2473-2480.

Ying, X. Ed. 2019. An overview of overfitting and its solutions. IOP Publishing. 022022.

Yobas, M.B., Crook, J.N. & Ross, P. 2000. Credit scoring using neural and evolutionary techniques. *IMA Journal of Management Mathematics.* 11(2):111-125.