1. **Introduction**

*1.1 The Global Financial Crisis and Risk Management Practices*

The Global Financial Crisis (GFC) was the most severe international economic crisis since the Great Depression. The GFC had far-reaching consequences for the financial system, the housing market, the economy and wider society in general. It resulted in a recession leading to extremely high levels of unemployment in the United States and most of Europe, homelessness crises and general economic turmoil (Consequences of the Global Financial Crisis, Grant, Wyn (editor)). The crisis consisted of wide-ranging issues in credit markets centred in the United States of America, Europe and the United Kingdom. Housing bubbles in the United States, the United Kingdom, Spain and Ireland were caused by banks providing households with loans at very low interest rates. Households were able to borrow at levels that they could not realistically afford and purchase larger properties. Turmoil began to unfold when property prices began to drop and households found themselves in crippling levels of debt, in some cases their debt was larger than the value of the property. It later emerged that complex financial instruments had been built by large banks on these underlying household loans that were ultimately extinct (McKinsey). A quintessential component of the financial system is the prudential measures applied to combat against malpractice and avoid devastating impacts as observed by the fall out of the GFC. Following the GFC greater efforts were made by regulatory bodies to provide frameworks by which financial institutions must comply by to ensure sufficient and adequate risk management practices in lending money. The Basel III framework was published in June 2011 as part of the Basel Committee’s response to the GFC. It addresses several shortcomings in the pre-crisis regulatory framework and provides a foundation for a resilient banking system that will help avoid the build-up of systemic vulnerabilities. Basel III scrutinised the pre-crisis risk management practices of financial institutions. A particularly important component of the Basel III framework was the increase in the level of capital requirements to ensure that banks are sufficiently resilient to withstand losses in the times of stress. This increased scrutiny on the capital requirements for banks provided an impetus for institutions to improve risk management practices, one of which being the credit risk framework.

* 1. *Risk Weighted Assets and Credit Risk Modelling*

According to the CFA Institute, credit risk is the risk of loss resulting from a borrower failing to make full and timely payments of interest and/or principal (CFA). Credit risk is the biggest risk for banks. Other types of risks include market risk, operational and concentration risk. However, poor credit risk management can result in bank failure, and this was accentuated by failures of banks such as Northern Rock and Anglo-Irish bank during the GFC. Therefore, it is worthwhile for banks to implement sound and effective credit risk management systems/frameworks to remain sufficiently capitalised and resilient to economic shocks. Banks have developed sophisticated systems to model the credit risk arising from important aspects of their business lines. These sophisticated models are intended to aid banks in quantifying, aggregating, and managing risk across different segments of their disparate business services. The output produced by these models play a significantly important role in the risk management processes. The Basel Committee on Banking Supervision found that credit risk modelling may prove to result in better internal risk management (Basel paper 1999). The output that is produced by credit risk models are used as parameters to compute a bank’s level of capital requirements. Credit risk modelling practices assist in the computation of regulatory capital. Regulatory capital is a component of ‘Pillar One’ of the Basel capital framework, which sets out an institution’s minimum capital requirements. The quantitative regulatory capital under Pillar One has a core simple equation, to maintain financial resources equal to or greater than a percentage of its risk weighted assets (RWAs). Risk weighted assets (RWAs) are a measure of a bank’s assets adjusted for their perceived risks (Irish RWA paper). Capital requirements are in line with RWAs, for example the minimum risk-based capital requirements as established by the Basel framework is that total capital held by an institution must be at least 8% of RWA. The Basel Framework describes how to calculate RWA for credit risk. RWA for credit risk can be calculated either using the standardised approach or the internal ratings-based (IRB) approach. The IRB approaches are complex and require considerable resources as well as robust systems. Banks aspire to use IRB models as they are a mark of greater sophistication and usually, they produce lower risk calculation than the standardised approach (Irish RWA). There are three parameters in developing IRB models: Probability of Default (PD), Loss given default (LGD) and Exposure at default (EAD). This paper places a greater emphasis on the PD parameter.

* 1. *Probability of Default and Current Methods*

The probability of default quantifies the likelihood of a borrower that he will not be able to meet its contractual obligations and will default. Banks need to estimate rating-wise PD for the calculation of regulatory capital. The prime objective in modelling default risk is to measure credit risk in terms of default probabilities rather than ordinal rankings. PD can be estimated at a borrower, or an overall portfolio level and it depends on borrower-specific characteristics ranging from financial, behavioural, geographical or external market factors (Managing Portfolio Credit Risks in banks). Traditional statistical methods such as Logistic Regression or Decision Tree based methods are utilised by financial institutions for likelihood of default prediction. This paper compares advanced machine learning algorithms versus the typical statistical methods.

* 1. *General Project Aim*
  2. *Objectives*
  3. *Report Overview*

1. **Background Research**

*2.1 Introduction*

There is a plethora of research outlining the use of machine learning concepts for credit risk models. The literature review included as part of this research proposal focusses on the development the Internal Ratings-Based approach to credit risk models, the early beginnings of various machine learning models and the relevant testing measures to compare the economic impact of each predictive model on capital requirements for financial institutions. Inspiration is drawn from various sources in particular to Alonso and Carbó 2020 in which they outline their methodology to understand the performance of machine learning models to predict credit default.

*2.2 Credit Risk & Probability of Default Definition*

Credit is the process of borrowing and lending capital. In order to compete with investment banking activities, commercial banks allow customers to acquire debt (Gande, 2008). There are numerous major risks faced by banks including credit, operational, market and liquidity risks. The most challenging risk associated with banking practices is Credit Risk. Loan performance affects profitability and the stability of bank growth and development. Credit risk is defined as the risk associated with a loan given by a bank, which will not be repaid-either partially or fully-on time (Campbell, 2007). One of the principle reasons of failure in the banking system, as accentuated by the Global Financial Crisis 2007 – 2008, is the persistent occurrence of non-performing loans on a financial institution’s balance sheet. Effective management of credit risk is a critical component of a comprehensive approach to risk management and essential to the long-term success of any banking organization (Basel, 2000). The Basel Committee on Banking Supervision issued its ‘Principles for the Management of Credit Risk’ in October 2000 encouraging banking supervisors to promote the sound practices for managing credit risk. Effective management of credit risk involves sound identification, measuring, monitoring and controlling of credit risk in order to determine the sufficient amount of capital to hold. This paper focusses on measuring credit risk. In January 2001, the Basel Committee on Banking Supervision published a consultative document on the ‘Internal Ratings-Based (IRB) approach’ for measuring credit risk. The paper stressed that this approach relies heavily upon a bank’s internal assessment of its counterparties and exposures. The two key objectives to be secured by the approach was additional risk sensitivity and incentive compatibility. The hope was that the IRB approach will provide a single framework by which a given set of risk components or ‘inputs’ are translated into minimum capital requirements. According to the Basel Committee on Banking Supervision, banks’ internal measures of credit risk are based on assessments of risk characteristics of both the borrower and the specific type of transaction. One such component to provide a measure of potential losses and, as such, credit risk is the Probability of Default of a borrower. The Probability of Default is one of the components that form the basic input to the IRB approach. According to (Grigutis, 2023) the Probability of Default (PD) is one of the most the important metrics in credit risk management. PD is a financial term that describes the likelihood of a borrower not being able to repay the obligations of a loan within a predefined timeframe. A higher PD indicates a higher risk of not repaying loan obligations, while a lower PD indicates a lower risk of not repaying loan obligations (Kenton, 2023). A bank’s estimate of PD to a certain exposure helps to map into a schedule of regulatory capital risk weights. Risk weighted assets (RWA) link the minimum amount of capital that a bank must have, to the risk of the profile of the bank’s asset book. The key role of RWA is to act as the measure of unexpected loss for banks portfolios (Lyons & Rice, 2022). Under the IRB framework, the risk inputs calculated using internal estimates, including PD, could map to different regulatory risk weights and produce a low level of risk allowing less capital held.

*2.3 Probability of Default Calculation Methods*

A classification is a method of predicting similar information from the value of a categorical target or categorical class variable (Reddy & Babu, 2018). Predicting borrowers default is a type of classification problem and, as such, classification methods are incorporated to predict a binary outcome. Logistic regression, linear-probability modelling and discriminant analysis are the techniques that are most commonly used in credit scoring. The considerations to take into account when selecting the techniques that is most appropriate for credit risk modelling are; ease of calculation, transparency, and whether or not it is statistically suitable for the problem. When credit scoring was first introduced in the 1950s and 1960s linear-probability modelling and linear discriminant analysis were the primary tools to measure risk. As time progressed, in tandem with the progression of computing power, maximum likelihood estimation (MLE) techniques became more feasible. Techniques such as logistic and Gaussian regression were adopted, both with less demanding statistical assumptions required compared to linear techniques. Logistic regression is used by between 80 and 90 per cent of scorecard (probability of default) developers (Anderson, 2007). Logistic regression is a well-known econometrics model in the banking industry (Alonso & Carbó, 2020). Research has been extended to model PD using alternative measures. Supriya et al. (2019) used Decision Trees as a machine learning tool to implement their model. Shoumo et al. (2019) used Support Vector Machines. Hamid and Ahmed (2016) used Data Mining techniques to build a model for classifying loan risk using three algorithms, J48, Bayes Net and Naïve Bayes. This paper will draw comparisons between traditional statistical modelling techniques to predict default (Logistic Regression) and more advanced machine learning methods to calculate PD.

*2.4 Logistic Regression*

Linear regression and linear probability modelling dominated early credit scoring techniques despite an obvious shortcoming, in that the target variable is binary. Logistic regression is more appropriate for credit scoring due to the prediction of a binary outcome. Logistic regression has its origins to the early 19th century for the description of the growth of populations and the course of autocatalytic chemical reactions (Cramer, 2003). Today, logistic regression uses a process called maximum likelihood estimation (MLE) to, transform the target variable into a log function, predict the coefficient size for independent variables, and determine the changes to coefficients, in order to maximize the likelihood of the log. It is a largely iterative and calculation-intensive approach, requiring almost six attempts to reach convergence, using one of several convergence criteria. The formulae for logistic regression can be found below:

A math equations with numbers and symbols

Description automatically generated with medium confidence

Logistic regression holds the following assumptions:

1. Categorical target variable;
2. Linear relationship with the log odds function;
3. Independent error terms;
4. Uncorrelated predictors;
5. Use of relevant variables.

Logistic regression is particularly effective due to its inherent ability to limit the predicted outcome between 0 and 1, and with the development of computer processing power has been adopted by major financial institutions in predicting loan outcome (Anderson, 2007). The emergence and development of Big Data, Machine Learning and Advanced Data Analytics methods has allowed for the possibility of more sophisticated tools to be used in the area of credit risk management (Alonso & Carbó, 2020).

*2.5 Machine Learning*

According to IBM, Machine Learning is a branch of artificial intelligence (AI) and computer science which focusses on the use of data and algorithms to imitate the way humans learn, gradually improving its accuracy. Machine learning is an important component of the growing field of data science. Through use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects (IBM, 2023). The origin of machine learning is associated with Frank Rosenblatt from Cornell University, who created a machine that was built for the purpose of recognizing letters of the alphabet (Rosenblatt, 1958). The machine, labelled ‘perceptron’, used both analog and discrete signals and became the prototype of modern artificial neural networks (ANN) (Fradkov, 2020). UC Berkely outline the three basic components of a typical supervised machine learning algorithm:

1. Decision process: calculations that attempt to predict the data pattern the algorithm is to find;
2. Error function: a method of measuring the prediction attempt by comparing to known examples (if available);
3. Optimization process: a method in which the algorithm can analyse the errors made and update how the decision process eventually selects a final decision.

Ayodele, 2010 outlines the main types of algorithms used in machine learning. These include; supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, transduction and learning to learn. Supervised learning entails learning a mapping between a set of input variables and an output variable and applying this mapping to predict the outputs for unseen data. Supervised learning is the most important methodology in machine learning (Cunningham et al. 2008). Unsupervised learning is where the machine simply receives inputs but obtains neither a supervised target, nor rewards from its environment. The goal of unsupervised learning is to build representations of the input variable that can be used for decision making, predicting future inputs or efficiently communicating the inputs to another machine (Ghahramani, 2003). Reinforcement learning is defined as the aim of building a machine that can learn from interacting with the environment (Sutton & Barto, 2018). The algorithm’s behavior is shaped through a sequence of rewards and penalties, which depend on whether its decisions towards a defined goal are correct or incorrect. This paper places emphasis on supervised machine learning algorithms mapping to the outcome of loan payment.

*2.6 Decision Tree Method*

A decision tree is a machine learning algorithm used for supervised learning. A decision tree simulates human thinking in tree structure for making a decision easily. Firstly, a decision tree picks an attribute to be the root node. Secondly, it creates a branch for each possible attribute value and separates the instance into multiple subsets. Thirdly, every subset links to a branch of the root node. Finally, the algorithm repeats the process iteratively on each branch (Kohavi, 2002). Decision trees are widely used in the banking industry due to their high accuracy and their ease of interpretability. Frydman et al. (1985) showed that results obtained through Recursive Partitioning Algorithm outperform discriminant analysis when using financial variables for credit risk measurement. Simha and Satchidananda (2006) applied both decision tree learning and logistic regression to predict default risk for agricultural loan information from India. The main conclusions were that decision trees generated good results and have the benefit of being easy to understand and implement. The figure below outlines a simple decision tree structure used for granting loan decision.

A diagram of a divorce

Description automatically generated

The Classification and Regression Trees (CART) algorithm is utilized as the learning algorithm in order to construct binary trees using the feature and threshold that yield the largest information gain at each node. Lewis (2000) introduces the concept of CART in order to create clinical decision rules found to perform as well or better than rules developed using methods that are more traditional. CART analysis consists of four basic steps. The first step involves building trees where each built tree is assigned a predicted class. The class is assigned irrespective of whether the node has been split into further child nodes. The second step involves stopping the tree building process, where a maximal tree is produced. The third step involves, ‘pruning’ a sequence of simpler trees by cutting off increasingly important nodes. The final step optimally selects trees, where the tree which fits information from the learning dataset, without overfitting, is selected among the new sequence of pruned trees. Luo, Hsu and Xu (2020) utilize this algorithm in the credit in an SME default prediction framework.

*2.7 Support Vector Machine*

Vapnik and coworkers first introduced a Support Vector Machine (SVM) for the purpose of classification. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. SVM is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data. SVMs can be defined as systems which use hypothesis space of a linear function in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory (Jakkula, 2006). Chen et al. (2011) propose using a Support Vector Machine (SVM) to predict the default risk of German firms. In the paper, the performance of the SVM is compared to more traditional methods such as logit and discriminatory analysis. It is found that the nonlinear model classified by the SVM outperforms to logit model using the same predictors. The aim of the SVM algorithm is to find an optimal hyperplane that separates data points belonging to different classes. A kernel function maps input data point into a higher-dimensional space in order to find a hyperplane that can effectively separate non-linearly separable data in the original feature space. The optimal hyperplane is then described as the one that maximizes the distance between the hyperplane and the nearest data point from either class. The support vectors are defined as the data points that lie closest to the hyperplane and largely influence its placement. The figurebelow displays the output of SVM classification.



*2.8 Random Forest*

Random forests are a scheme proposed by Breiman (2001) to build a predictor ensemble using a set of decision trees that grows in randomly-selected subspaces of data. A random forest is a classifier consisting of a collection of tree-structured classifiers where the inputs are independent identically distributed random variables used to determine how the successive cuts are performed when building individual trees. The accuracy of a random forest depends on the strength of the individual tree classifiers and the measure of the dependence between them. For each input, the random forest procedure will be able to predict, with an optimal number of trees, the binary output value. A stopping criteria is used to minimize the prediction error (Addo et al., 2018). According to IBM, the key benefits of random forests are; a reduced risk of overfitting due to the robust number of decision trees in the forest lowers variance and error, providing flexibility as it can handle both classification and regression problems, and, easy to determine feature importance through metrics such as gini importance and mean decrease in impurity. Random forest have been applied in many disparate industries. Madaan et al. (2021) found that the random forest algorithm outperformed the decision tree model when prediction loan outcome for a peer-to-peer financial institution dataset. The figurebelow displays the structure of a random forest algorithm.

A diagram of a tree

Description automatically generated

*2.9 Artificial Neural Network*

Deep learning (DL) is the subfield of artificial intelligence that focuses on creating large neural network models that are capable of making accurate data-driven decisions, particularly suited to contexts where the data available is complex and large (Kelleher, 2019). There have been three waves of developments in DL: cybernetics in the 1940s-1960s, connectionism in the 1980s-1990s and the current resurgence beginning in 2006 (Goodfellow et al., 2016). Haykin (2016) defines an Artificial Neural Network (ANN) as a massively parallel combination of simple processing units, which can acquire knowledge from an environment through a learning process and store the knowledge in its connections. Artificial/Feed-forward Neural Networks (ANNs) consist of an input layer, an output layer and a specific number of hidden layers. Neurons are contained in the input and output layers related to the number of input and output parameters. It is then an objective to determine the number of hidden layers and the number of neurons in each hidden layer. Activation functions are applied by each neuron in the hidden and output layers to the weighted sum of inputs. A characteristic of Artificial Neural Networks is the theory of Back Propagation. Back Propagation is a learning algorithm that is based on the Deepest-Descent technique. If the network is provided with an appropriate number of Hidden units, the network will also be able to minimize the error of nonlinear functions with high complexity. Practically, it is often necessary to provide ANNs with at least two layers of hidden units, when the function to compute is particularly complex, or when the chosen data, in order to train the back propagation are not reliable (Buscema, 1998). Artificial neural networks have been employed in various industries such as weather forecasting, medical diagnosis, aerospace, facial recognition, stock market and social media.

A diagram of a machine learning

Description automatically generated

Although modern software products enable relatively easy handling with artificial neural networks, their creation, optimization, and usage in real-life situations requires an understanding of the supporting theory and output required (Suzuki, 2011). Selection of the appropriate neural network architecture, hyperparameter tuning tuning, selection of the loss function, optimization algorithm and computational resources are a few of the considerations to take into account before developing a neural network. Bayraci and Susuz (2019) applied a Deep Neural Network to Turkish commercial bank data in order to predict loan default. It was found that the Deep Neural Network model significantly improved the performance of the credit scoring system when compared to logistic regression and support vector machine models for the loan application dataset. The performance metrics of choice included balanced accuracy and Type I and Type II error. However, the deep neural network did not outperform its counterparts in the loan performance dataset. It was found that traditional classification models may be preferred as opposed to more sophisticated deep learning models due to ease of implementation and interpretability. Byanjakar et al. (2015) found that a neural network-based credit scoring model performs effectively in screening default applications when applying such a model to peer-to-peer lending data.

*2.10 XGBoost*

Chen and Guestrin (2016) describe eXtreme Gradient Boosting (XGBoost) as a scalable machine learning system for tree boosting. It is a parallel tree boosting, also known as Gradient Boosted Decision Trees (GBDT) and Gradient Boosting Machine (GMB), which solve many data science problems in a fast and accurate way. In XGBoost individual trees are created using multiple cores and data is organized in order to minimize the lookup times. Schapire (2003) defines boosting as the machine learning method based on the observation that finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule. XGBoost has gradient boosting at its core, however, the difference between the simple gradient boosting algorithm and XGBoost algorithm is that, the process of adding weak learners does not happen successively. It takes a multi-thread approach whereby proper utilization of the CPU core of the machine are utilized, leading to greater speed and performance. XGBoost has been applied to credit risk in many academic papers. Wang (2022) applies an XGBoost model to build a personal credit risk evaluation model. It was found that the XGBoost model performs better than a decision tree and K-nearest neighbor model, based on AUC, accuracy and Type II error. Petropoulos et al. produced a paper where they used an XGBoost and deep learning neural network model to quantify the probability of default for a large dataset of corporate credit loans in the Greek banking system. The study’s conclusion found that the XGBoost model seemed to marginally outperform the deep neural network however there was greater flexibility offered by the neural networks, which may optimize the bias variance trade-off. XGBoost was selected as it can perform very well on any dataset size even with little features.

*2.11 Performance Measures*

Multiple models have been proposed as part of this research. Consequently, the performance of the proposed models will have to be compared against each other to formalize conclusions. There are many forms of performance measure in machine learning which depend on the type of problem to be solved. In our case the problem to be solved is loan default outcome which is, by definition, a binary classification problem. Standard performance measures of binary classifications problems include Type I Error, Type II Error, accuracy, precision, recall, AUC, confusion matrix etc. All these performance measures will be explored and reported. The performance metrics will be placed in sub categories based on their purpose. The sections will be accuracy-related metrics, discriminatory metrics and stability metrics.

*2.12 Accuracy-Related Performance Metrics*

Accuracy related metrics in machine learning measure the extent to which your model makes correct predictions of the target variable. The measure explored in this literally review is the Receiver Operating Characteristic (ROC) curve and consequently the Area under the Curve (AUC) metric.

*2.13 Receiver Operating Characteristic (ROC) Curve and Area under the Curve (AUC)*

The receiver operating characteristic (ROC) curve and the area under the curve (AUC) are effective measures of the accuracy of model predictions. ROC curves are used to judge the discrimination ability of various statistical methods (Hanley & McNeill, 1982). Conventionally, a standard way of describing binary classification outcomes is by using a confusion matrix. A confusion matrix visualizes and summarizes the performance of a classification algorithm (Tiwari, 2022).



From the above confusion matrix the sensitivity of specificity of a model’s predictions can be calculated. The sensitivity of a model’s predictions is equal to the true positive fraction (TP/(TP+FN)). The specificity of a model’s predictions is equal to the true negative fraction (TN/(TN+FP)). These are performance measures in their own right but also are factored into the ROC curve measurement. The formal definition of the ROC curve is the plot of sensitivity versus 1-Specificity (Metz, 1978). One way to look at the ROC curve is to view it as a function built using two cumulative distribution functions (CDFs). With regards to credit risk modelling, the two CDFs used are credit scores relating to good and bad customers. Gönen and Haller (2010) summarizes the general formula for a ROC curve as below:



Where FB is a CDF of the scores of the bad customers, FG is a CDF of good customers, and FG-1 is its inverse.

A ROC curve can be summarized with its area under the curve (AUC). The AUC measures the discrimination power of the binary classifier. Generally, the higher the AUC, the better at predicting the outcomes the model is. The ROC curve is considered standard practice by credit-risk professionals (Anderson, 2007). The AUC is used to assess, improve, and compare credit risk models (Xia et al. 2020). With the ROC curve model, the path of the curve can be estimated based on sample results and the confidence intervals for AUC can be computed (Lahiri & Yang, 2018). However, computing the AUC in isolation might not be informative of the discriminatory power of the model. Řezáč and Koláček (2012) showed that some models excel at distinguishing good customers form bad ones dependent on the shape of the ROC curve. Tang and Chi (2005) show that a model with a much lower AUC outperformed a model with a higher AUC.

A diagram of a positive and negative test results

Description automatically generated

*2.14 Discriminatory Metrics*

The discriminatory power of a classification model is the ability to discriminate between the different classes, which serve as the output of the model.

*2.15 Gini Coefficient*

Gini coefficient is one of the methods that have been used to measure inequality in a population. It can be defined as the mean of the absolute differences between all pairs of individuals for some measure. In the context of credit risk, the gini coefficient can be computed by comparing the concentration of ‘bad’ borrowers (defaults) in the higher probabilities and ‘good’ borrowers in the lower probabilities. The gini coefficient is a rescaling of the area under the curve (AUC) metric from the receiver operating characteristic (ROC) curve.

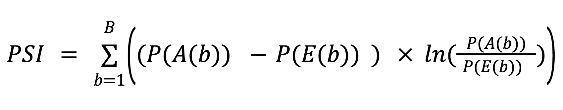
  
The gini coefficient is equivalent to a special case of Somers’ D statistic where one of the variables is binary and the other one is ordinal (Somers, 1962). The gini coefficient is a metric that is used by professionals to validate the credit risk models developed. It is recommended by the Basel Committee to be used. A gini coefficient of 50% indicates that the model outputs each value evenly and does not provide a better discriminatory model than a random guess. A gini coefficient of above 50% is considered useful and a better discriminator than random chance.

*2.16 Stability Metrics*

A crucial component during model development and testing is to assess whether the population experiences major changes.

*2.17 Population Stability Index (PSI)*

The population stability index (PSI) is a statistic that measures how much a variable has shifted over time, and is used to monitor applicability of a statistical model to the current population (Yurdakul & Naranjo, 2020). The formula for PSI can be found below:



The rules of thumb for PSI are; PSI < 0.1 means ‘little shift’, 0.1 < PSI < 0.25 means ‘moderate shift’, and PSI > 0.25 ‘significant shift’. The PSI is typically calculated for each independent variable in the model, however in this research paper it is proposed to apply PSI to the credit grades that are calculated as part of the probability of default model. The PSI is informative of a shift in the population of a model, however it does not inform model developers of the cause of the shift and further investigation is required to obtain this information.

*2.18 Feature Importance Measures*

A useful comparison between models will be to assess which features are important in each and perceive if certain features are better predictors for default than others. Numerous feature importance metrics will be used, dependent on model type. Coefficient size will be used for logistic regression, feature importance for random forest and decision and SHAP values for neural networks and XGBoost. SHAP values will be extended on in this literature review.

*2.19 Shapley Additive explanation values*

A recurring motif with machine learning models is the problems with interpretability of the models. It can be difficult to interpret the results of the model, in particular the key features that contribute to a model’s output. Lundberg and Lee (2017) introduced SHapley Additive exPlanations to address the problem of interpreting the predictions of complex models. With SHAP, an explanation model is used and represented by a linear model to determine an additive feature attribution method. According to C3.ai,the SHapley vlue is the average marginal contribution of a feature considering all possible combinations. SHapley values are computed for each individual prediction to assign which features are the most important in determining a particular output. These values can also be aggregated to a global level for each model and analyse which features are the most prominent. SHapley values are becoming more prominent in the banking industry as accentuated by their presence in the EBA’s response to their machine learning paper where Intesa Sanpaolo and the French Banking Federation mention their use of SHAP methodology (ref).

*2.20 Regulatory Capital Calculation*

For each type of exposure a bank has under its operations, risk weights are derived under a specific function. A risk-weighted asset is defined as the risk weight of a transaction multiplied by a measure f exposure for that transaction. Total risk weighted assets (RWA) are the sum of individual RWA across all transaction (BIS, 2001). As part of this research it is proposed to attach a tangible economic impact for the dataset under examination by calculating the capital requirement under each model. Baena et al. (2005) explains that theoretically statistical models with better predictive power have the potential to yield a better outcome in term of regulatory capital requirements. It is shown that the risk weighted function using the Internal Ratings-Based (IRB) approach is concave in PD. This implies that the capital requirement increases as PD increase. Therefore, if the predictive power of PD improved with more sophisticated models a lower capital requirement could be the outcome. Therefore, the capital requirements of advanced machine learning techniques will be assessed in terms of savings to regulatory capital holdings by banks. The formulas that are used to calculate capital requirements are included below (these may be subject to change dependent on the type of exposures included in the dataset, i.e mortgages vs corporate loans).

A math equation with black text

Description automatically generated with medium confidence

A mathematical equation with numbers and symbols

Description automatically generated



Various components of the above formulae will not be computed as part of this research (EAD, LGD). Loss-given default (LGD) can be benchmarked to a standardized figure published by the EBA dependent on the exposure class. Exposure at default (EAD) cannot be benchmarked and therefore the RWA figure cannot be calculated. As such, the capital requirement function will be used to compare different models.

*2.21 Conclusion*

There is a significant amount of research carried out on the potential applications of machine leaning techniques on predicting the outcome of borrower loan behavior for financial institutions. The above literature review explores multiple sources and methodologies upon which the realm of advanced data analytics has been extended into credit risk modelling. Upon investigation, a particular concept with little research are the actual economic impacts associated with these applications. Little research has been carried out to assess the financial impacts/trade-offs/consequences for banks in advancing their machine learning capabilities. This research paper intends to extend the outcome of machine learning applications from scientific measurement to a tangible measurement, namely capital requirement. In doing so, this paper may help answer the question as to whether it would be in the greater interest for financial institutions to invest time and resources in more sophisticated applications of artificial intelligence.

1. **Methodology**

*3.1 Overview*

The methodology to achieve the aforementioned goals as part of this research paper will be discussed in the following section. This section will describe the data collection process, including the primary research strategy with a presentation of the primary research results and the source information for the secondary data that is collected and used to implement various machine learning models. Furthermore, this section will cover the machine learning modelling implementation techniques, including and not limited to logistic regression, random forests and deep learning techniques such as an artificial neural network.

* 1. *Primary Research*
  2. *Secondary Data*

The secondary data used in this research paper is used for the purpose of the development of machine learning models to predict the likelihood of a borrower defaulting on their credit obligation. The dataset is sourced from the supplementary data available as part of the *Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS* by Bart Baesens, Daniel Roesch and Harald Scheule. The dataset is in panel form and reports origination and performance observations for 50,000 residential U.S. mortgage borrowers over 60 periods. The periods have been deidentified. As in the real world, the loans may originate before the start of the observation period (this is an issue where loans are transferred between institutions as in securitisation). The loan observations may thus be censored as the loans mature or borrowers refinance. The dataset is a randomised selection of mortgage-loan-level data collected from the portfolios underlying U.S. residential mortgage-backed securities (RMBS) securitisation portfolios and provided by International Financial Research ([www.internationalfinancialresearch.org](http://www.internationalfinancialresearch.org)). The key features are listed and described in Table 1**.**

Table 1 Mortgage Dataset Features

| **Feature** | **Description** |
| --- | --- |
| id | Borrower ID |
| time | Time stamp of observation |
| orig\_time | Time stamp for origination |
| first\_time | Time stamp for first observation |
| mat\_time | Time stamp for maturity |
| balance\_time | Outstanding balance at observation time |
| LTV\_time | Loan-to-value ratio at observation time, in % |
| interest\_rate\_time | Interest rate at observation time, in % |
| hpi\_time | House price index at observation time, base year = 100 |
| gdp\_time | Gross domestic product (GDP) growth at observation time, in % |
| uer\_time | Unemployment rate at observation time, in % |
| REtype\_CO\_orig\_time | Real estate type condominium = 1, otherwise = 0 |
| REtype\_PU\_orig\_time | Real estate type planned urban development = 1, otherwise = 0 |
| REtype\_SF\_orig\_time | Single-family home = 1, otherwise = 0 |
| investor\_orig\_time | Investor borrower = 1, otherwise = 0 |
| balance\_orig\_time | Outstanding balance at origination time |
| FICO\_orig\_time | FICO score at origination time, in % |
| LTV\_orig\_time | Loan-to-value ratio at origination time, in % |
| Interest\_Rate\_orig\_time | Interest rate at origination time, in % |
| hpi\_orig\_time | House price index at origination time, base year = 100 |
| default\_time | Default observation at observation time |
| payoff\_time | Payoff observation at observation time |
| status\_time | Default (1), payoff (2), and nondefault/nonpayoff (0) observation at observation time |

* 1. *Exploratory Data Analysis*

An important step in any study leveraging off a dataset is to understand the data you are working with. This section will explore the data to be used for modelling. The dataset contains 622,489 observations and 23 features. The first two explorations in the data were the degree of missingness and the degree of uniqueness per feature.

Table 2 Missing Count per Feature

| **Feature** | **Missing Count** |
| --- | --- |
| id | 0 |
| time | 0 |
| orig\_time | 0 |
| first\_time | 0 |
| mat\_time | 0 |
| balance\_time | 0 |
| LTV\_time | 270 |
| interest\_rate\_time | 0 |
| hpi\_time | 0 |
| gdp\_time | 0 |
| uer\_time | 0 |
| REtype\_CO\_orig\_time | 0 |
| REtype\_PU\_orig\_time | 0 |
| REtype\_SF\_orig\_time | 0 |
| investor\_orig\_time | 0 |
| balance\_orig\_time | 0 |
| FICO\_orig\_time | 0 |
| LTV\_orig\_time | 0 |
| Interest\_Rate\_orig\_time | 0 |
| hpi\_orig\_time | 0 |
| default\_time | 0 |
| payoff\_time | 0 |
| status\_time | 0 |

There does not appear to be a systematic issue with the level of missing data in the mortgages dataset. There is one variable with 270 missing counts however, as there are 622,219 non-missing entries for the LTV\_time feature this is not a worryingly high number and a reasonable imputation can be applied to the missing observations. A mean imputation was applied to the 270 missing observation for LTV\_time.

Table 3 Unique Count per Feature

| **Feature** | **Unique Count** |
| --- | --- |
| id | 50000 |
| time | 60 |
| orig\_time | 98 |
| first\_time | 50 |
| mat\_time | 202 |
| balance\_time | 482832 |
| LTV\_time | 567800 |
| interest\_rate\_time | 5539 |
| hpi\_time | 60 |
| gdp\_time | 60 |
| uer\_time | 36 |
| REtype\_CO\_orig\_time | 2 |
| REtype\_PU\_orig\_time | 2 |
| REtype\_SF\_orig\_time | 2 |
| investor\_orig\_time | 2 |
| balance\_orig\_time | 10694 |
| FICO\_orig\_time | 397 |
| LTV\_orig\_time | 552 |
| Interest\_Rate\_orig\_time | 1825 |
| hpi\_orig\_time | 97 |
| default\_time | 2 |
| payoff\_time | 2 |
| status\_time | 3 |

The number of unique observations per feature informs the data user whether there is a degree of duplication in the features to be potentially used as key features for modelling. Should there be a degree of duplication in modelling features this may lead to invalid results. From Table 3 there is no issue in duplicate observations. The variables REtype\_CO\_orig\_time, REtype\_PU\_orig\_time, REtype\_SF\_orig\_time, investor\_orig\_time, hpi\_orig\_time, default\_time, payoff\_time and status\_time are indicator variables, either binary or multiclass and therefore it is reasonable to expect two or three levels of uniqueness.

Figure 1 Distribution Breakdown of Mortgages Dataset

A screenshot of a graph

Description automatically generated

Figure 1 above details the distributions of the features found in the mortgages dataset. Distribution visualisations inform model developers of key insights of the features to be potentially used in models. The central tendency of the variables can be extracted from the above panel. The LTV\_time feature is relatively normally distributed, whereas the balance\_orig\_time feature is skewed right. The distribution charts provide an initial insight into the spread of the features. It can be observed that the balance\_time feature is spread very widely as the x-axis ranges from values of 0 to 600,000. The gdp\_time feature is not spread across a wide range of values as the x-axis ranges from values of 0 to 4.

Figure 2 Boxplots of Mortgages Dataset

A screenshot of a computer

Description automatically generated

Figure 2 above displays boxplots for all the features included in the mortgages dataset. Boxplots inform model developers as to the degree of outlying information present in the possible features to be included in potential models. Outliers are defined as patterns in data that do not conform to a well-defined notion of normal behaviour (Outlier Detection: Applications And Techniques, Karanjit Singh and Dr. Shuchita Upadhyaya). In this paper Tukey’s (1977) method of constructing a boxplot and using the sample mean and standard variance to detect extreme values. The rules of the method are as follows:

1. The IQR (Inter Quartile Range) is the distance between the lower (Q1) and upper (Q3) quartiles.
2. Inner fences are located at a distance 1.5 IQR below Q1 and above Q3 [Q1-1.5 IQR, Q3+1.5 IQR].
3. Outer fences are located at a distance 3 IQR below Q1 and above Q3 [Q1-3 IQR, Q3+3 IQR].
4. A value between the inner and outer fences is a possible outlier. An extreme value beyond the outer fences is probably an outlier. There is no statistical basis for the reason that Tukey uses 1.5 and 3 regarding the IQR to make inner and outer fences.

For the purpose of this research paper anything located in the outlier region as described above will be treated as an outlier. Outliers are to be removed from the continuous numeric features. Figure 3 below presents the boxplot breakdown for the features of the mortgages dataset following the removal of outliers.

Figure 3 Boxplots of Mortgages Data (Post-Outlier adjustment)

A diagram of a diagram

Description automatically generated with medium confidence

Another EDA technique for model developers is to investigate correlations amongst features of the dataset to be used. This helps developers to adjust features to be included in models in order to avoid issues such as multicollinearity, a problem which is particularly relevant for logistic regression models. Highly correlated features will be removed from the logistic regression model, where the feature with higher information value (IV) will remain in the model.

Figure 4 Feature Correlations

A screenshot of a computer screen

Description automatically generated

Highly correlated feature pairs are defined as having a correlation greater than 40%. The highly correlated pairs are presented in Table 4.

Table 4 Highly correlated feature pairs

|  |  |
| --- | --- |
| **Feature 1** | **Feature 2** |
| balance\_time | balance\_orig\_time |
| LTV\_time | hpi\_time |
| LTV\_time | uer\_time |
| LTV\_time | hpi\_orig\_time |
| interest\_rate\_time | FICO\_orig\_time |
| interest\_rate\_time | Interest\_Rate\_orig\_time |
| hpi\_time | uer\_time |
| REtype\_PU\_orig\_time | REtype\_SF\_orig\_time |

Another key aspect of the model development process was to resample the data for the model to be built on. The mortgages dataset displayed signs of being imbalanced. Imbalanced data typically refers to a problem with classification problems where the classes are not represented equally, including binary classification problems as well as multi-class classification problems (A generalized method to predict the compressive strength of high-performance concrete by improved random forest algorithm). **INPUT THE ACTUAL IMBALANCE.** When experimenting with the machine learning algorithms to be explored as part of this research paper, the results appeared biased towards the majority class in the training datasets, the performing observations. In all the machine learning models the predictions made on the testing data skewed towards the performing observations and no defaulted cases were predicted. Consequently, approaches to class imbalance remediation were explored. Methods explored included resampling the data and generating synthetic samples. It was decided to randomly sample the training datasets to improve the machine learning methods due to computing time and interpretability. Other methods such as using the SMOTE algorithm is more difficult and time consuming in Python. It was decided to randomly over sample the minority class in the data, being the default next twelve months data points. This was done to prevent the machine learning models from being overly biased towards the majority class while also allowing the algorithms to improve the model’s ability to discern relevant patterns and features. The RandomOverSampler function from the imbalance learn package in python was utilised to achieve the target of addressing class imbalance in the training datasets. The objective of the RandomOverSampler function is to over-sample the minority class by picking samples at random with replacement. The sampling strategy selected was equal to 0.5 to have half of the majority cases numbers equal to the minority cases. This resulted in less biased samples and improved testing metrics to accurately compare the performance of each machine learning method in predicting the default behaviour for borrowers over a twelve-month period.

The target variable to be modelled required adjustment. This paper focuses on the development of sophisticated machine learning methods for the application of probability of default models. The target variable in these models is whether a borrower defaults on their credit obligation. However, an adjustment is made to the default indicator feature to predict the behaviour of a borrower over a twelve-month period. The target variable is thus transformed into whether a borrower defaults in the next twelve-months. This is carried out for many reasons. The primary reasons for this adjustment are to allow for the better prediction of future events, removing the potential seasonality effect in the model, alignment with the terms of credit obligation contracts and it is in line with industry standards and regulatory requirements. An arbitrary selection to use December snapshots was made. This, combined with the resampling exercise, resulted in a final modelling dataset of **INPUT THE DIMESNIONS OF THE DATASET TO BE MODELLED ON**.

* + 1. *Feature Engineering*

The Weight of Evidence (WoE) feature engineering technique is implemented for the purpose of the logistic regression model build. The concept of WoE was expanded upon in the background research section. The purpose of the WoE transformation is to recode quantitative and qualitative predictor features into discrete categories. Sidiqi (2006) develops on the process for implementing WoE transformations. The continuous features of the dataset are broken down into 4 discrete categories. The binary features of the dataset are broken down into 2 discrete categories. The WoE values then replace the actual raw values present in the data to be included as features in the logistic regression model. This is a technique that is commonly used in credit scoring and logistic regression and helps to express the predictive power of the feature in the context of the target variable. The Information Value (IV) helps to assess the overall predictive power of features. IV is a key determinant as to which highly correlated feature should be dropped from model development for logistic regression. The formulae for WoE and IV can be found below.

Table 5 lists the IV figures for the features of the mortgages dataset.

Table 5 Information Value per feature

| **Variable** | **IV** |
| --- | --- |
| LTV\_time | 0.32925 |
| interest\_rate\_time | 0.21895 |
| hpi\_orig\_time | 0.18343 |
| Interest\_Rate\_orig\_time | 0.17218 |
| FICO\_orig\_time | 0.13965 |
| gdp\_time | 0.11938 |
| hpi\_time | 0.06754 |
| uer\_time | 0.03463 |
| balance\_time | 0.02544 |
| balance\_orig\_time | 0.02339 |
| LTV\_orig\_time | 0.02181 |
| investor\_orig\_time | 0.00069 |
| REtype\_SF\_orig\_time | 0.00053 |
| REtype\_PU\_orig\_time | 0.00006 |
| REtype\_CO\_orig\_time | 0.00000 |

Prior to the development of the Decision Tree, SVM, Random Forest, ANN and XGBoost models the features are standardised using the StandardScaler function as part of the scikit learn package in Python. This is a common requirement for many machine learning estimators, where they might behave badly if the individual features do not appear normally distributed.

* 1. *Model Development*

This subsection will describe the methodology behind the development of six machine learning methods used as probability of default models for regulatory capital purposes. The methods explored include Logistic Regression, Decision Tree, Support-Vector Machine (SVM), Random Forest, Artificial Neural Network (ANN) and XGBoost. The development process will be discussed, as well as the model optimisation techniques, final model fits and the classification accuracy for each.

* + 1. *Logistic Regression*

The first method explored was a Logistic Regression model. Logistic Regression is a fundamental classification technique as is the most widely used model to predict the likelihood of a borrower defaulting on a credit obligation in the financial services industry. Logistic regression bases itself on the odds of a two-level outcome of interest. The odds of the event being the ratio of the probability of the event happening divided by the probability of the event not happening. The natural logarithm of the odds as a regression function of the predictors are taken. Unlike linear regression, there is no formula for the estimates of the coefficients for logistic regression. The best estimates require repeatedly improving estimates until stability is reached (Logistic Regression, Michael P. LaValley). The optimal weights are assigned once the log-likelihood function is maximised, better known as the maximum likelihood estimates (MLE).

A stepwise method was utilised to find the logistic regression model of best fit. All the potential WoE transformed features in the dataset were included in the original model as a starting point. Variables were initially dropped from the model based off a p-value cut-off point. Any features, when included in the model, that had a p-value greater than 0.05 were dropped immediately. A p-value greater than 0.05 indicates that a feature is not overly predictive for the target variable, at the 5% significance level. There were no features to be dropped due to a high p-value. Therefore, the only reason to drop features from a logistic regression model was due to high correlation amongst regressors. Multicollinearity is a feature among regression fits when there is a degree of correlation amongst predictors. It can lead to problems such as unstable coefficients, model instability and loss of variable importance. One variable in each highly correlated pair as detailed in Table 4 was removed from the logistic regression model. The feature to be removed was determined on their information value figure, as outlined in Table 5. Therefore, the features to be removed due to high correlation were balance\_orig\_time, Interest\_Rate\_orig\_time, LTV\_time, Retype\_PU\_orig\_time, uer\_time and FICO\_orig\_time. The final model fit, including coefficients, p-values, 95% confidence intervals for the coefficients and standard errors for feature are included in the below model summary.

Figure 5 Logistic Regression Model Summary

A screenshot of a computer

Description automatically generated

All features in the model are significant at the 5% level. All the features have positive coefficients, meaning that increases in the value of the features are associated with increases in the log-odds of the borrower defaulting in the next twelve-months. The final functional form of the logistic regression model is as per below. **INSERT FORMULA**

Figure 6 Correlation for Logistic Regression Features

A screenshot of a color chart

Description automatically generated

The above correlation diagram confirms there is no existing degree of multicollinearity in the logistic regression model. The highest level of correlation amongst the regressors being -0.34 between REtype\_SF\_orig\_time and REtype\_CO\_orig\_time.

The below confusion matrix outlines the performance of a classification model. It breaks down the results of the logistic regression model into four categories, true positive, true negative, false positive and false negative. The true positive calculation identifies the number of instances that were correctly predicted as positive by the model. The true negative calculation identifies the number of instances that were correctly predicted as negative by the model. The false positive calculation identifies the number of instances that were predicted as positive by the model that were actually negative. The false negative calculation identifies the number of instances that were predicted as negative by the model that were actually positive.

Table 6 Logistic Regression Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | | |
|  | Predicted 0 | Predicted 1 |
| Actual 0 | 13,005 | 2,093 |
| Actual 1 | 270 | 156 |

Confusion matrices use the true positive rate, true negative, false positive and false negative rates to provide insights into the performance of classification models. The precision, recall, accuracy and F1 score of the model can be computed from a confusion matrix. The precision of a model is a measure of the percentage of the predicted events, in our case borrower default in the next twelve-months, which are actually true positives. The recall of a model is a measure of the percentage of actual occurrences against all predicted occurrences of default. The accuracy of the model is the percentage of correctly predicted observations against all observations. The F1 score is the harmonic mean of precision and recall. It takes both false positives and false negatives into account, meaning it performs well on an imbalanced dataset. The formula for each metric can be found below.

Table 7 Logistic Regression Classification Metrics

|  |  |
| --- | --- |
| Metric | Value |
| Accuracy | 84.78% |
| Precision | 6.94% |
| Recall | 36.62% |
| F1 | 11.66% |

An accuracy metric of c. 85% means that the logistic regression probability of default model is correct in its predicted outcome almost 85% of the time. However, this should not be viewed in isolation and other metrics should be combined to provide useful insight into the overall predictive power of the model. The precision value is very low at c. 7%. This low precision indicates that when the model predicts a positive instance it is very often incorrect. The recall value is low at c. 37%. A low recall suggests that the model is not effectively identifying a significant portion of the actual positive instances. The F1 score is very low at c. 12% meaning that the model is not performing well in terms of both precision and recall. The overall balance between precision and recall is not satisfactory, resulting in a low F1 score. Overall, while the model has a relatively high accuracy it has very low precision, recall and F1 scores. This means that the model highly struggles to identify and classify positive instances. While steps have been taken to address imbalances in the data it seems that the model does not respond well, and further steps would have to be taken to improve overall model performance for logistic regression.

* + 1. *Decision Trees*

The next model explored was a Decision Tree classification method. A Decision Tree is a classification technique that assigns each object in dataset into a predicted class, default in the next twelve-months or non-default in the next twelve-months, based on each objects’ attributes. The decision tree algorithm uses information gain to find the best attribute for classifying the data. For each value defined for the decision values of the best attribute, the algorithm repeats the process with additional attributes (A Comparison of Logistic Regression to Decision Tree Induction in a Medical Domain). A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node (Loan Credibility Prediction System Based on Decision Tree Algorithm). The attraction for financial services to implement decision trees for credit risk modelling purposes is the ease of explainability and interpretability of a decision tree-based machine learning algorithm compared to more advanced techniques such as neural networks. Decision trees can be visually represented and require minimal knowledge of the underlying statistical theory.

The process to fit a decision tree algorithm to the mortgages dataset including splitting the data into training and testing datasets, with a 30% split. Similarly to the logistic regression model build, the training dataset for the purpose of decision tree modelling was imbalanced and thus required a random oversampling of the minority class. All possible features for modelling were included in the training dataset. This is an important characteristic for decision tree machine learning methods. Decision trees have the ability to automatically select features during the learning process, making them less sensitive to irrelevant or redundant features. The algorithm selects features based on their ability to split the data and improve the overall purity or information gain in the nodes. The input data to the decision tree model is transformed using the StandardScaler function from the scikit learn package to standardise the features to unit variance. The DecisionTreeClassifier function from scikit learn is also utilised as the classification model. A required technique when fitting a machine learning model is to fine tune the hyperparameters in order to optimise model performance. The GridSearchCV function from the scikit learn package is used to produce an exhaustive search over the specific potential parameters that could be included in the decision tree classification model. The parameters to be explored for the decision tree were the criterion to be used as the measure to quality of the split, the maximum depth of the tree, the minimum number of samples required to be at a leaf node, and the minimum number of samples required to split an internal node. The hyperparameter grid to be optimised for the decision tree model is found in Table 8.

Table 8 Hyperparameters for Decision Tree Classification

|  |  |
| --- | --- |
| Hyperparameter | Value |
| Criterion | gini, entropy |
| Max\_depth | 5, 10, 15 |
| Min\_samples\_leaf | 2, 5 |
| Min\_samples\_split | 1, 4 |

The GridSearchCV function uses a cross-validation technique to optimise the estimator parameters to search over the predefined grid. 3-fold cross validation was specified to search the parameter grids for the optimal hyperparameters for the decision tree model. The scoring metric to measure the performance of the cross-validation on the test set was set to the accuracy metric. The result of the grid search assigned the gini criteria to measure the quality of the split, a value of 15 for the maximum depth of the tree, a minimum number of 1 sample required to be at a leaf node and a minimum of 5 samples required to split the internal nodes of the tree. Model predictions for the testing dataset were produced using these parameters as inputs, resulting in the following confusion matrix.

Table 9 Decision Tree Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | | |
|  | Predicted 0 | Predicted 1 |
| Actual 0 | 13,655 | 1,455 |
| Actual 1 | 296 | 118 |

Similarly to the logistic regression confusion matrix, the accuracy, precision, recall and F1 score will inform model developers and relevant stakeholders as to the classification performance of the decision tree model.

Table 10 Decision Tree Classification Metrics

| Metric | Value |
| --- | --- |
| Accuracy | 88.72% |
| Precision | 7.50% |
| Recall | 28.50% |
| F1 | 11.88% |

An accuracy metric of c. 89% means that the decision tree probability of default model is correct in its predicted outcome almost 89% of the time. The precision value is very low at 7.5%. This low precision indicates that when the model predicts a positive instance it is very often incorrect. The recall value is low at 28.5%. A low recall suggests that the model is not effectively identifying a significant portion of the actual positive instances. The F1 score is very low at c. 12% meaning that the model is not performing well in terms of both precision and recall. The overall balance between precision and recall is not satisfactory, resulting in a low F1 score. Overall, while the model has a relatively high accuracy it has very low precision, recall and F1 scores. This means that the model highly struggles to identify and classify positive instances. It seems that the imbalance problem is still persistent despite a more advanced machine learning methodology and the more sophisticated algorithm was unable to handle the low event prediction issue.

An important output of the DecisionTreeClassifier from the scikit learn package is the feature importances. Feature importance is an important aspect of machine learning algorithms as it allows developers and stakeholders to gain insight into how complex algorithms work in order to better understand the domain of application. The feature importance metric used by the decision tree classifier is the gini importance metric used by the CART (classification and regression tree) algorithm to optimise the decisions at the nodes of the tree. The importance of the a feature is computed as the (normalised) total reduction of the criterion brought by that feature, in our case the features are measured by their overall reduction of the gini factor when they are removed from the decision tree.

Table 11 Feature Importance Decision Tree

|  |  |
| --- | --- |
| Feature | Importance Scores |
| LTV\_time: | 0.3118 |
| FICO\_orig\_time: | 0.1515 |
| interest\_rate\_time: | 0.0957 |
| balance\_time: | 0.0910 |
| Interest\_Rate\_orig\_time: | 0.0795 |
| balance\_orig\_time: | 0.0663 |
| LTV\_orig\_time: | 0.0638 |
| hpi\_orig\_time: | 0.0599 |
| gdp\_time: | 0.0281 |
| REtype\_SF\_orig\_time: | 0.0132 |
| hpi\_time: | 0.0105 |
| REtype\_CO\_orig\_time: | 0.0095 |
| uer\_time: | 0.0074 |
| REtype\_PU\_orig\_time: | 0.0065 |
| investor\_orig\_time: | 0.0055 |

According to the gini importance metric, LTV\_time is the most importance feature. It is found that the LTV\_time is the biggest contributor to distinguishing between a borrower defaulting in the next twelve-months or not defaulting. The feature with the lowest gini importance is the investor\_orig\_time feature. This feature has little or no effect on correctly distinguishing between good and bad borrowers. The LTV\_time and FICO\_orig\_time are the only features with a great than 10% gini importance metric.

* + 1. *Support Vector Machine (SVM)*

SVM was the next machine learning algorithm that was experimented as part of this research paper. The primary aim of the SVM technique is to project nonlinear separable samples onto another higher dimensional space by using different types of kernel functions (Sandeep Kumar Satapathy, ... Shruti Mishra, in EEG Brain Signal Classification for Epileptic Seizure Disorder Detection, 2019). SVMs distinguish between two classes by finding the optimal hyperplane that maximises the margin between the closest data points of the opposite class. The features of the input data determine if the hyperplane is a line in a 2-D space or a plane in a n-dimensional space. Maximising the margin between points enables the algorithm to find the best decision boundary between classes, allowing for SVMs to generalise well to unseen data and make accurate out of sample classification predictions. The lines that are adjacent to the optimal hyperplane are known as the support vectors (IBM). The advantages of using a SVM for a classification problem are that it is effective in a high-dimensional space, allowing models to perform well in a dataset with many features, it naturally handles outliers as the decision boundaries are limited to the concept of margin and it can be robust to overfitting by generalising well to unseen data. The disadvantage is that it is very computationally heavy and doesn’t work very well to large data.

The process to develop the SVM technique was like that of fitting a decision tree. The in-scope mortgage borrowers were split into training and testing datasets at a split of 30%. To satisfy the imbalanced data a random oversampling was carried out of the training dataset, once again performed at the 50% level. The input data was then scaled to normalise features. A key aspect of fitting a SVM is to tune the hyperparameters that constitute the model fit. The GridSearchCV function was used to tune the key hyperparameters. The hyperparameters to be tuned for the SVM included the ‘C’ regularisation parameter, which determines the strength of the regularisation at an inversely proportional relationship, the kernel type to be used in the algorithm and the gamma parameter which is a tuning parameter that appears in kernel functions to influence the shape of the decision boundary. The grid of potential hyperparameters to be tuned can be found in Table 12.

Table 12 Hyperparameters for SVM Classification

|  |  |
| --- | --- |
| Hyperparameter | Value |
| C | 1, 10 |
| Kernel | Linear, RBF |
| Gamma | Scale, Auto |

The GridSearchCV function uses a cross-validation technique to optimise the estimator parameters to search over the predefined grid. 3-fold cross validation was specified to search the parameter grids for the optimal hyperparameters for the SVM. The scoring metric to measure the performance of the cross-validation on the test set was set to the accuracy metric. The result of the grid search assigned a value of 10 for the regularisation parameter, the ‘RBF’ (Radial Basis Function) kernel that is used typical for non-linear classification and a gamma value of ‘auto’ to determine the width of the RBF kernel and the influence of individual training samples. The functional form of ‘auto’ for SVM is equal to 1/n\_features. Model predictions for the testing dataset were produced using these parameters as inputs, resulting in the following confusion matrix.

Table 13 SVM Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | | |
|  | Predicted 0 | Predicted 1 |
| Actual 0 | 12,302 | 2,828 |
| Actual 1 | 207 | 187 |

The accuracy, precision, recall and F1 score will inform model developers and relevant stakeholders as to the classification performance of the SVM model.

Table 14 SVM Classification Metrics

|  |  |
| --- | --- |
| Metric | Value |
| Accuracy | 80.45% |
| Precision | 6.20% |
| Recall | 47.46% |
| F1 | 10.97% |

An accuracy metric of c. 80% means that the SVM probability of default model is correct in its predicted outcome almost 80% of the time. The precision value is very low at 6.2%. This low precision indicates that when the model predicts a positive instance it is very often incorrect. The recall value is low at c. 47.4%. A low recall suggests that the model is not effectively identifying a significant portion of the actual positive instances. The F1 score is very low at c. 11% meaning that the model is not performing well in terms of both precision and recall. The overall balance between precision and recall is not satisfactory, resulting in a low F1 score. Overall, while the model has a relatively high accuracy it has a very low precision score with a relatively decent recall score. The low precision lowers the F1 score. This means that the model highly struggles to identify and classify positive instances. It appears that the SVM model did not perform as well as the two previous methods, despite a higher degree of complexity in the algorithm and supposed ability to generalise well and more accurate predictions to unseen data.

*3.5.4 Random Forest*

The Random Forest classifier consists of a combination of tree classifiers where each classifier is generated using a random vector sampled independently from the input vector, and each tree casts a unit vote for the most popular class to classify an input vector (Dutta et al., 2021). The RandomForestClassifier function from the scikit-learn package in python is used to build a random forest model to predict a borrower defaulting over the next twelve-months. The algorithm develops trees in an ensemble that is built from a sample drawn with replacement (bootstrapping) from the training set. When each node is split during the construction of a tree, the best split is found through an exhaustive search of the feature values of either all input features or a random subset of a size identified through a predefined hyperparameter. These two sources of randomness allow the random forest estimator to decrease its overall variance, compared to individual decision trees, which exhibit high variance and tend to overfit. The reduced variance is achieved by combining diverse trees, sometimes at the cost of a slight increase in bias. The random forest implementation of classifiers combines individual tree classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class.

The in-scope borrowers were split into training and testing datasets at a split of 30%. To satisfy the imbalanced data a random oversampling was carried out of the training dataset, once again performed at the 50% level. The input data was then scaled to normalise features. Once again, a hyperparameter tuning exercise was completed to optimise the random forest model algorithm. The hyperparameters to be tuned included the number of trees in the forest, the maximum depth of the tree, the minimum number of samples required to split an internal node, and the minimum number of samples required to be at a leaf node. The grid of potential hyperparameters to be tuned can be found in Table 15.

Table 15 Hyperparameters for Random Forest

|  |  |
| --- | --- |
| Hyperparameter | Value |
| N\_estimators | 50, 100, 200 |
| Max\_depth | 5, 10, 15 |
| Min\_samples\_split | 2, 5, 10 |
| Min\_sampled\_leaf | 1, 2, 4 |

The GridSearchCV function was utilised to search through the potential combinations of hyperparameters using 3-fold cross validation, optimised on the accuracy metric. Following 3-fold cross validation, the optimal hyperparameters to be used to fit a random forest model were defined as 100 trees, a maximum depth of 15, 2 samples at a minimum to split an internal node and 1 minimum sample required at a leaf node. These parameters were defined, and the model was fit on the training dataset with output predictions for the unseen training data. The below confusion matrix summarises the classification prediction performance of the random forest.

Table 16 Random Forest Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | | |
|  | Predicted 0 | Predicted 1 |
| Actual 0 | 14,963 | 168 |
| Actual 1 | 378 | 15 |

The accuracy, precision, recall and F1 score will inform model developers and relevant stakeholders as to the classification performance of the random forest model.

Table 17 Random Forest Classification Metrics

|  |  |
| --- | --- |
| Metric | Value |
| Accuracy | 96.48% |
| Precision | 8.20% |
| Recall | 3.82% |
| F1 | 5.21% |

An accuracy metric of c. 96% means that the Random Forest probability of default model is correct in its predicted outcome almost 96% of the time. This is completely biased towards the prediction of non-default events. The precision value is very low at 8.2%. This low precision indicates that when the model predicts a positive instance it is very often incorrect. The recall value is extremely low at c. 4%. A low recall suggests that the model is not effectively identifying a significant portion of the actual positive instances. The F1 score is very low at c. 5.2% meaning that the model is not performing well in terms of both precision and recall. The overall balance between precision and recall is not satisfactory, resulting in a low F1 score. Overall, while the model has a relatively high accuracy it has very low precision, recall and F1 scores. This means that the model highly struggles to identify and classify positive instances. The random forest model is not effective at predicting default outcomes and is significantly biased towards the performing/non-default outcomes.

* + 1. *Artificial Neural Network*

Deep learning methods are explored as part of this research paper through the development of an Artificial Neural Network (ANN). ANN models use similar input and output parameters as linear models. These models have three primary components, an input layer, a hidden layer and an output layer. The input layer received the input value, the hidden layer connects a set of neurons between input and output layers and the output layer contains one neuron and its output ranges from 0 to 1. Each of the layers contains nodes which are processing units called neurons to connect to each adjacent layer to each other (Measuring Credit Risk of Bank Customers Using Artificial Neural Network). The neurons have a function that determines the activation of the neuron. A bias is also added to the neurons along with inputs. A weight is the connection of the signal of the neuron. The product of the weight and input gives the strength of the signal. A neuron receives multiple inputs from different sources and has a single output. There are various functions that can be used as activation functions, depending on the type of problem the model is fit to. The sigmoid function is the most used function, used in the output layer for binary classification problem. Other activation functions include the step function, linear function, ramp function, hyperbolic tan and the rectified linear unit function. The sum is the weighted sum of the inputs multiplied by the weights between one layer and the next. The form of the ANN developed in this paper is multilayer perceptions (MLP), which contain one input layer, one output layer and one or more hidden layers. Back propagation is used to train the neural network. The training method uses the difference in the target output and the output predicted by the network and propagates that back between layers and adjusts the weights.

The in-scope borrowers were split into training and testing datasets at a split of 30%. To satisfy the imbalanced data a random oversampling was carried out of the training dataset, once again performed at the 50% level. The input data was then scaled to normalise features. The architecture of the ANN had to be defined. The ANN contained an input layer, a hidden layer and an output layer. The hidden layer contained an activation function to be hyperparameter tuned and the input dimensions are set to the number of features in the input dataset. The output layer contains one neuron, and the activation function is the sigmoid function, used for binary classification. The network is then compiled by a loss function and an optimiser algorithm. The loss function is set to binary crossentropy and the optimiser algorithm is set for hyperparameter tuning. The other hyperparameters to be tuned included the batch size which is the number of samples per gradient update, the number of epochs to update the model based off error and loss function, the optimisation algorithm, the number of neurons in the hidden layer and the activation function the be used in the hidden layer. The hyperparameters and the possible outcomes can be found in Table 18.

Table 18 Hyperparameters for ANN

|  |  |
| --- | --- |
| Hyperparameters | Value |
| Batch\_size | 16, 32 |
| Epochs | 10, 20 |
| Optimizer | adam, rmsprop |
| Units | 10, 20 |
| Activation | relu, sigmoid |

The GridSearchCV function was utilised to search through the potential combinations of hyperparameters using 3-fold cross validation, optimised on the accuracy metric. Following 3-fold cross validation, the optimal hyperparameters to be used to fit the ANN were defined as a batch size of 16, the number of epochs set at 20, a relu activation function in the hidden layer, the adam optimisation algorithm and 20 neurons in the hidden layer. These parameters were defined, and the model was fit on the training dataset with output predictions for the unseen training data. The below confusion matrix summarises the classification prediction performance of the ANN.

Table 19 ANN Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | | |
|  | Predicted 0 | Predicted 1 |
| Actual 0 | 12,800 | 2,336 |
| Actual 1 | 223 | 165 |

The accuracy, precision, recall and F1 score will inform model developers and relevant stakeholders as to the classification performance of the ANN.

Table 20 ANN Classification Metrics

|  |  |
| --- | --- |
| Metric | Value |
| Accuracy | 83.52% |
| Precision | 6.60% |
| Recall | 42.53% |
| F1 | 11.42% |

An accuracy metric of c. 83.5% means that the ANN probability of default model is correct in its predicted outcome almost 84% of the time. The precision value is very low at 6.6%. This low precision indicates that when the model predicts a positive instance it is very often incorrect. The recall value is better than previous models at c. 42.5%. A low recall suggests that the model is not effectively identifying a significant portion of the actual positive instances. The F1 score is low at c. 11.4% meaning that the model is not performing well in terms of both precision and recall. The overall balance between precision and recall is not satisfactory, resulting in a low F1 score. Overall, while the model has a relatively high accuracy it has very low precision, with an improved recall and F1 scores. The low precision means that the model is not very effective in predicting a default event occurring. The recall value has improved compared to previous models meaning that of all the actual positive values the number of positives predicted by the model has improved.