Credit risk prediction using artificial neural network algorithm

Gulfornia Thakana Phahlane (gulfornia@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Prof Phillip Mashele
North-West University, South Africa

22 September 2020

Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

Abstract

Credit risk is one of the biggest risks that banks face on a daily basis. Banks match lenders with borrowers, however, this may lead to credit risk when borrowers fail to pay back their loans. In this project, we will use artificial neural networks, specifically the multi-layer feed-forward network to predict credit risk. However, the dataset is severely imbalanced and to handle the imbalance we will use re-sampling techniques, such as Nearmiss, Synthetic Minority Oversampling Technique (SMOTE) combined with Tomek links, Edited Nearest Neighbor (ENN) combined with Condensed Nearest Neighbor (CNN), and Adaptive Synthetic approach (ADASYN). All of these re-sampling techniques will also be used together with machine learning classifiers, namely K-Nearest Neighbor (KNN), Naive Bayes (NB) and Random Forest (RF) for experimentation. We will evaluate the performance of each model based on the precision score, recall, and F1-score.

Keywords: SMOTE, Tomek links, ENN, CNN, ADASYN, RF, KNN, NB, precision score, recall, F1-score.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Gulfornia Thakana Phahlane, 22 September 2020

Contents

1. Introduction

Credit is the contractual agreement between a borrower and a lender in which a borrower is granted a loan that is repaid on an agreed date with interest. There are many different forms of credit with financial credit being the most popular. This kind of credit includes car loans, mortgages, and signature loans. Defaulting is the inability or unwillingness of a borrower to pay back the principal amount and interest of their loan on time (Spuchl'áková et al., 2015). Credit risk is the risk of lenders losing money due to default on loans.

1.1 Background

According to Ogboi and Unuafe (2013), credit risk is one of the causes of the global financial crisis from 2007 to 2009 and the follow-up economic meltdown. This crisis negatively impacted global economy particularly South Africa. In 2008 and 2009 the economy went into recession, where 1 million jobs were lost, and the unemployment rate raised to 25% (Rena and Msoni, 2014). Since then, South Africa's economic growth is very slow with the possibility of another recession. Currently, with the Coronavirus disease 2019 (Covid-19) pandemic, many businesses closed down during the national hard lock-down phase leading to job losses. As a result, banks stand to suffer as borrowers will now have a hard time paying back their loans. For example, Capitec bank that focuses on lending to people with lower incomes stands to lose more (Coetzee, 2003). So far, credit risk remain one of the biggest risks that banks and other credit institutions face daily (To, 2013).

Ghatasheh (2014) evaluated the performance of the random forest algorithm. He observed that the algorithm performs well in predicting credit risk with better classification accuracy. Random trees are also easier to understand, making it easy to understand the underlying relations.

Ferreira et al. (2017) conducted a study on the impact of class-imbalance on credit risk prediction. The author used supervised classification models and sampling techniques to deal with class imbalance. Ensembles, cost-sensitive and sampling methods were combined and evaluated along with logistic regression, decision trees, and Bayesian learning schemes. The author observed that sampling techniques performed better than ensembles and cost sensitive approaches.

Alejo et al. (2013) used the multilayer perceptron neural network using three misclassification cost functions to study credit risk prediction. The results showed that the cost functions improve the prediction of instances of the minority class.

In a study of credit risk prediction in social lending by Namvar et al. (2018), the author evaluated the comparison of various classifiers with re-sampling techniques to model credit risk while handling imbalanced data. To avoid bias, the credit predictions from each combination are evaluated with a geometric mean measure, that measures the balance between classification performance for both the majority and minority class. The author observed that combining Random Forest and random under-sampling is very effective when predicting credit risk.

With the increased use of technology, the amount of data also increases. Classification of data becomes a challenge due to class imbalance. Class imbalance occurs where one class has more than other classes, resulting in misleading accuracy and bias when modeling. However, in the last few years there has been improvement regarding approaches to dealing with classification of data. Methods such as Edited nearest Neighbour, Synthetic Minority Oversampling and Adaptive Synthetic approach are more focused

on classification of minority class than the majority class. The minority observations are those that rarely occur but very important.

In a study by Longadge and Dongre (2013), it was reported that data preprocessing, boosting and feature selection are useful regarding data balancing. During data preprocessing new information can be added and irrelevant information can be deleted which helps to balance the data. Whereas boosting is a powerful ensemble learning algorithm. Examples of boosting algorithms are Random Under-Sampling combined with Boosting (RUSBoost) and Synthetic Minority Oversampling and Adaptive Synthetic approach combined wit Boosting (SMOTEBoost). Feature selection method where the number of input variables are reduced can also be used for classification of imbalance data. Some of the examples of feature selection methods are decision trees and recursive feature elimination.

1.2 Aims and objectives of the project

• To identify relevant features that affect credit.

• Predict and categorize the state of credit by designing an artificial neural network that can be used to predict creditworthiness based on the data for a particular loan.

1.3 Contribution

The contribution in this project is to design a feed-forward artificial neural network, select features that are relevant to credit risk prediction, use re-sampling techniques to balance data and experiment with machine learning classifiers to predict credit risk.

1.4 Project outline

The project is structured as follows: In Chapter 2 we discuss credit scoring, features that affect credit, and credit risk models. We further discuss machine learning classifiers and artificial neural networks in Chapter 3. Chapter 4 highlights the methodology of the experiments carried out, and in Chapter 5 we present the results of the experiments. Finally we give a conclusion of the project in Chapter 6 and briefly discuss future work.

2. Credit scoring and credit risk models

In this chapter, we discuss credit scoring, features that affect credit and credit risk models.

2.1 Credit scoring

A credit score is used by lenders to evaluate a credit application. A score is determined only by the information in a credit report and it is used by lenders to predict credit risk. According to Arya et al. (2013) in 1956, Fair, Isaac, and Company proposed a credit score called FICO and it became the most commonly used credit score eversince. FICO score is a three-digit number that ranges from 300 to 850, with 300 being the lowest score and 850 the highest score. It is used by lenders to determine the likelihood of an applicant repaying a loan. The higher the score the lesser the risk. Moreover, this score has an effect on the sum of money an applicant can borrow, the number of months for repayment and the interest rate. For example, an applicant with a higher score my be eligible to borrow a higher amount of money.

Credit scoring is also objective and consistent, that is, it does not favour any particular person therefore eliminating discrimination against applicants. Using credit scoring makes the process of crediting, and approving or rejecting loans quicker and efficient. Additionally, this method can be maintained and improved over time to produce better results that will enhance credit decisions and management of credit. In addition this method also indicates if a loan repayment may be delayed, allowing for adjustments to be made on the loan in order to reduce the risk. Moreover, when creditworthy applicants are approved for loans, and there are no defaults on their side, the bank makes profit. However this is not always the case, if the method completely fails to classify loan applicants due to incorrect analysis or interpretation of the data, the bank loses a lot of money.

2.2 Features that affect credit

The following are features that impact credit risk:

Application-type: Application type refers to whether the application is joint or individual. According to Roslan and Karim (2009) group lending reduces credit risk in such a way that members of the group may monitor and encourage one another to pay their portion of loan instalments on time. If one member defaults, all members will be affected and may be denied loans in future. With individual lending, if a personal loan is used to pay-off other existing loans, the individual may find themselves trapped in a debt cycle and end up defaulting on loans. In a study by Lehner (2009) on group and individual lending in micro-finance, it was observed that micro-finance institutions grant individual loans only when the amount is smaller so that the borrower can be able to pay it back in full and on time.

Loan amount: This is the amount of money that is given to the borrower by the lender (Nazari and Alidadi, 2013). When the loan amount is smaller, it may be insufficient, creating cash flow problems as it does not serve the purpose it was suppose to. Therefore when the amount is sufficient, default rate is lower, that is, borrowers are able to generate cash and repay the principal amount with interest on time.

Term: This is the period that is agreed upon by the lender and borrower for loan repayments (Namvar et al., 2018). The longer it takes to repay a loan, the lower the periodic payments making for good credit. On the other hand, longer repayment periods may be detrimental to the borrower in such a way

that they cannot access future loans until the existing loans are paid back. This may encourage the borrower to pay loans on time with interest or to opt for shorter repayment period which may turn out to be bad. Shorter repayment period might cause the borrower not to generate enough cash to make loan repayments leading to loan default.

Interest rate: A certain percentage of the loan amount that determines the bank's profit (Nazari and Alidadi, 2013). Interest rate of a loan determines the probability of loan default. When the interest rate charged increases, loan default also increases because the amount the applicant has to pay back increases and they may not afford it.

Installment: This is the amount agreed on to repay the loan on regular periods which can be monthly, weekly or quarterly. The installment period is determined based on an individual's sources of income, existing debts and monthly bills. If the loan applicant is able to pay their expenses and loan repayments while still being able to sustain their normal daily life, they will be offered a loan. Otherwise, the applicant stands a high chance of defaulting.

Grade: A grade is a credit rating assigned to loan application (Gupta and Goyal, 2018). Lending club loan grades are A, B, C, D, E,F and G where A indicates lower default risk and G indicates higher default risk. The grade takes into account the borrower's credit and several indicators of credit risk from the credit report and loan application. These indicators may include employment length, annual income, repayment history, existing debts, etcetera.

Employment length: This is the total number of years that the borrower has been employed (Nazari and Alidadi, 2013). Lenders use employment length to determine if a loan applicant has a stable job and if they will be able to make loan repayments if it happens they lose their jobs.

Annual income: This the amount of money the loan applicant makes per annum. Monthly or annual income is used by lenders to calculate the debt-to-income ratio (DTI). DTI is a measure of an applicant's ability to repay a loan. DTI is also a reflection of the portion of the applicant's income that goes towards loan repayment. It is calculated as follows (Namvar et al., 2018):

$$\text{DTI} = \frac{\text{Total monthly bill repayments}}{\text{Gross monthly income}}. \tag{2.2.1}$$

A higher salary is an indication that there will be enough money for loan repayments after subtracting expenses.

2.3 Credit risk models

Credit risk models are used to model and predict loan default (Nilsson and Shan, 2018). In the following two sections 2.3.1 and 2.3.2 we discuss the most common credit risk models, namely, discriminant analysis and logistic regression. Regression analysis uses statistical tools to estimate the relationships between input and output variables (Draper and Smith, 1998).

2.3.1 Discriminant analysis.

Discriminant analysis functions are similar to multiple regression analysis where a plane is fitted to data. However the dependent variable in regression analysis is continuous whereas the dependent variable in discriminant analysis is discrete. Linear Discrimination Analysis (LDA), was one of the first credit scoring models. However, with this method the credit data is not normally distributed. Moreover, the covariance matrices of the good and bad credit classes are unequal. More advanced statistical model was proposed to solve some deficiencies of the LDA model (West, 2000). Discriminant functions classify units into groups (Artís et al., 2011).

2.3.2 Logistic regression.

Logistic regression model,also known as logistic model, is common for modelling the probability of a certain binary events such as default/non-default, win/lose or pass/fail. According to (Cox and Snell, 1989) logistic models are important a they are easy to interpret and can be adjusted to fit the specific problem at hand. Some of the advantages of using this model is that it identifies the correlation between variables, and the relationship between the independent variables and the dependent variable (Fensterstock, 2005). The mean value of the outcome variable (conditional mean) is very important in logistic regression. According to Wi (2000) logistic models apply the same rules as linear regression, therefore the conditional mean can be expressed as a linear function in the following way:

$$E(Y|x) = \beta_0 + \beta 1x, \tag{2.3.1}$$

where Y is the outcome variable, and $x \in R$ denotes the independent variables. For binary data, the conditional mean must be:

$$0 \leq E(Y|x) \leq 1. \tag{2.3.2}$$

To achieve the transformation of $E(Y|x)$ into a linear function, we let :

$$\pi(x) = E(Y|x), \tag{2.3.3}$$

where $\pi(x)$ is expressed as,

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}. \tag{2.3.4}$$

Applying the $\log$ function on both sides, the logistic regression equation becomes:

$$g(x) = \ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \beta' x, \tag{2.3.5}$$

where: $\beta' = (\beta_0, \beta_1, \beta_2, ..., \beta_n)$ is the vector of the risk factors.

Therefore the logistic model is linear in its parameters and has many of the properties of a linear regression.

3. Machine learning concepts

In this chapter we look at machine learning classifiers and artificial neural networks.

3.1 Machine learning classifiers

Machine learning classifiers are algorithms that carry out classification by learning the training data and use this knowledge to classify new instances. This process of learning the training data is called supervised learning. The different machine learning classifiers are discussed below.

3.1.1 K-Nearest Neighbour (KNN).

Given a new data point and two classes, class 1 and 2, the classifier finds the distance between the new data point and existing training data points. Depending on the value of K, where K is the number of training data points nearest to the given data point, the minimum distance is considered amongst the data points. The classifier continues to check the number of nearest neighbors that belong to class 1 and class 2. If there is a maximum number of neighbors that belong to class 1, then the new data point belongs to class 1. However KNN does not have a learning phase, it memorizes the training data and classifies new data based on similarities. There are different distance metrics but below we discuss the Euclidea distance.

Given two data points $P_1(x_1, y_1)$ and $P(x_2, y_2)$,the Euclidean distance between $P_1(x_1, y_1)$ and $P(x_2, y_2)$ can be calculated in the following way:

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \tag{3.1.1}$$

However in this project we will be taking the Euclidean distance between two points which are represented by feature vectors:

$$X_1 = (x_{11}, x_{12}, x_{13}, ..., x_{1d}), \tag{3.1.2}$$

$$X_2 = (x_{21}, x_{22}, x_{23}, ..., x_{2d}), \tag{3.1.3}$$

where d is the dimension of the vector,then

$$d(X_1, X_2) = \sqrt{\sum_{i=1}^{d} (x_{1i} - x_{2i})^2}. \tag{3.1.4}$$

3.1.2 Random Forests (RF).

Random forest is a supervised learning algorithm that is commonly used for carrying out regression and classification tasks using decision trees (Breiman, 2001). The Random classifier creates a forest with multiple decision trees to classify a new observation based on the attributes. Each tree votes for a specific class-makes a classification. The forest chooses the classification with the most votes. The decision trees in Random forests are illustrated as upside-down trees with the root at the top of the tree. The root represents the decision node, the branches represent decision alternatives or outcomes. The decision nodes are being split at each level to form new branches.

For cases where there is more than one feature influencing the classification process, it can be found that some features are more relevant than others. Therefore it is important to place the features in their order of importance, with the most relevant at the root node.

The two popular methods for constructing a decision in decision trees are information entropy and Gini. These methods measure the impurity of a node. A node is said to be pure if it has one class and not if it has multiple classes.

Information entropy:

$$\text{Entropy} = -\sum_{i=1}^{n} \log p(c_i), \tag{3.1.5}$$

where $p(c_i)$ is the probability of class c_i in a node. The maximum value for entropy depends on the number of classes. For two classes the maximum entropy is 1.

Gini index:

$$\text{Gini} = 1 - \sum_{i=1}^{n} p(c_i)^2, \tag{3.1.6}$$

The higher the number of trees, the higher the accuracy. One advantage of the random forest classifier is that it does not lead to model over-fitting.

3.1.3 Naive Bayes (NB).

Naive Bayes is a machine learning classifier based on Bayes theorem (Lewis, 1998). Bayes theorem describes the probability of an event occuring, based on existing knowledge of conditions that might be connected to the event. Naive Bayes assumes that features in the dataset are independent of each other and do not affect each other's performance (Rish et al., 2001). It can be expressed mathematically as :

$$P(A) = \frac{P(B|A)P(A)}{P(B)}, \quad P(B) \neq 0, \tag{3.1.7}$$

where:

A and B are two events,

$P(A|B)$: is called conditional probability. This is the probability of event A occurring given that event B has occurred.

$P(B|A)$: this is the probability of event A occurring given that event B has occurred. It is also called conditional probability

$P(A)$ and $P(B)$: are probabilities of A and B occurring respectively.

There are different types of Naive Bayes classifiers, namely Gaussian, Multinomial, and Bernoulli Naive Bayes.

Multinomial Naive Bayes: is used to classify text in documents, for example classifying normal messages from spam by checking the probability of words that appear in a normal message (Murphy et al., 2006).

Gaussian Naive Bayes: Gaussian Naive Bayes is named after the Gaussian distributions that represent data in the training dataset. Gaussian Naive Bayes calculates the mean value and standard deviation of each feature to plot the Gaussian distributions of each feature in the training dataset.

Bernoulli Naive Bayes: is used to classify discrete binary data. In this project, we have used Gaussian Naive Bayes specifically because the data we are classifying is numeric but not binary.

$$P(C = c_k|X = x) = P(C = c_k) \times \frac{P(X = x|C = c_k)}{P(x)} \tag{3.1.8}$$

Since X and C are random variables, equation (3.1.8) becomes:

$$P(c_k|x) = P(c_k) \times \frac{P(x|c_k)}{P(x)}. \tag{3.1.9}$$

Since in this project we are classifying the creditworthiness of a loan applicant, we let,

$c_k \in C$ be the k classes in which the possible events fall and, $X = \{x_1, x_2, x_3, ..., x_n\}$ is the vector of features. Substituting X into (3.1.9) we get:

$$P(c_k|x_1, x_2, x_3, ..., x_n) = P(c_k) \times \frac{P(x_1|c_k)P(x_2|c_k)P(x_3|c_k)...P(x_n|c_k)}{P(x_1)P(x_2)P(x_3)...P(x_n)}, \tag{3.1.10}$$

$$P(x|c_k) = \prod_{j=1}^{n} P(x_j|c_k), \tag{3.1.11}$$

substituting equation (3.1.11) into equation (3.1.9) we get :

$$P(c_k|x) = P(c_k) \times \frac{\prod_{j=1}^{n} P(x_j|c_k)}{P(x)}. \tag{3.1.12}$$

The higher the value of $P(c_k) \times \prod_{j=1}^{n} P(x_j|c_k)$ the more accurate the classification (Lewis, 1998).

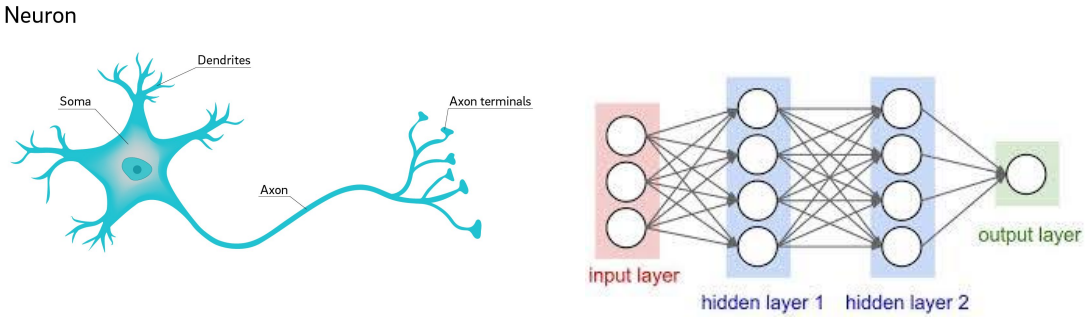
3.2 Artificial Neural Networks (ANN)

In this section, we discuss the biology of artificial neural networks, different types of artificial neural networks, and the learning process of an artificial neural network.

3.2.1 The biology of artificial neural network.

Artificial neural networks are computational models that function similar human nervous system. The brain contains 100 billion neurons and each neuron is connected to about 10^4 other neurons (Leung, 2008). Neurons are cells that process and transmit information in the nervous system. Neurons are made up of four regions namely, the cell body, dendrites, axons, and terminal buttons. Dendrites serve as a communication medium for neurons. Information travels from the terminal buttons of one neuron to the dendrites of another across a synapse. A synapse is a point that joins the terminal button of an axon and the dendrites. Information at the synapse travels in one direction. The axon is a long tube that carries information from the cell body to the terminal button.

The neurons in artificial neural networks are represented by nodes. The input layer functions like the dendrites, it receives information such as training dataset. Training dataset is data that is used to set up a model. The input layer does not carry out any model computations. Information travels from the input layer to the hidden layers across weights that form connections between nodes. In a case where the connection is feed-forward, the information would travel from one layer to another in one direction and each node in each layer is fully connected to all the nodes in the next layers (Parker, 2006). Figure 3.1 illustrates a biological neuron and an artificial neural network.



(a) Source: (Medical Xpress) (b) Source:(Digital trends)

Figure 3.1: Illustration of; (a) a biological neuron and (b) an artificial neural network.

3.2.2 Different types of neural networks.

Feed-forward neural network: Feed-forward neural networks are made up of three-layer,the input layer, the hidden layer, and the output layer. When the network has more than one hidden layer, it is called a deep neural network. The different types of feed-forward neural networks are as follows:

1. **Single layer:** Single-layer feed-forward neural networks contain one input layer and one output layer of the processing units. However, these are not powerful enough to approximate complex functions (Hinton et al., 2006).
2. **Multilayer feed-forward neural network:** Multi-layer feed-forward neural networks comprise of one input layer, one or more hidden layers, and one output layer of the processing units with no feedback connections. Hornik et al. (1989) describes multi-layer feed-forward neural networks as universal approximators because they can approximate any function that can be quantified, with a desired level of accuracy.
3. **Self-organizing neural network:** A self-organizing neural network, also known as Kohonen map is based on unsupervised learning (Yorek et al., 2016). This map is composed of two layers, the input layer, and the Kohonen layer. In the Kohonen layer a map is formed based on clustering of the dataset.

Recurrent neural network: These have directed cycles in their network, that is, if you start at the node and follow the arrows in the network you get back to the node you started at (Schuster and Paliwal, 1997). However, these networks are complicated and very difficult to train. These are very good at modeling sequential data

Convolutional Neural Network (CNN): Convolutional neural networks are deep neural networks mostly used for image processing. With CNN's an image is converted from Red-Green-Blue (RGB) scale to the gray-scale (Albawi et al., 2017). Images in gray-scale can be classified into different classes.

Given that feed-forward neural networks are simple and easy to interpret, we would be using them in this project,specifically multi-layer neural network as they have the ability to approximate any function in order to model credit risk prediction.

3.2.3 The learning process of an artificial neural network.

Generally, an artificial neural network learns in the following way:

1. **Initialization:** Feed input data into the network through the input layer
2. **Forward propagation:** From the input layer data moves to the hidden layers. For each hidden

layer, each input neuron is multiplied by the corresponding weights and added to a bias. An activation function is then applied to the summation to produce an output neuron.In general activations of an artificial neural network can be expressed as follows (Gupta and Goyal, 2018):

$$z_i = W_{j,i}a_j + b_j, \tag{3.2.1}$$

$$a_i = g(y), \tag{3.2.2}$$

where, $W_{j,i}$ represents the weights between neuron i and j and a_j is the activation function of neuron j.

3. **Back propagation:** this alters the weights to the optimal values for the neurons to minimize the error (cost function). For example, in Figure 3.2, we consider a neural network with 4 layers and one neuron in each layer:

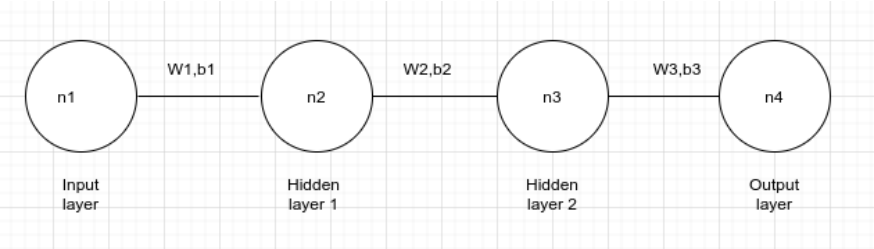


Figure 3.2: A neural network with four layers.

Cost function: An error function is defined which calculates the difference between the actual output and the estimated output of the model. The cost function is expressed as :

$$C(W_1, b_1, W_2, b_2, W_3, b_3, ..., W_n, b_n), \tag{3.2.3}$$

where $(W_1, .., W_n)$ are the weights and $(b_1, ..., b_n)$ are the biases. Let the activation function of the output neuron to be a^L, the activation of the second hidden layer to be a^{L-1} and the output layer to be y. The cost of one training example is:

$$C_0 = (a^L - y)^2, \tag{3.2.4}$$

where,

$$a^L = \sigma(W^L a^{L-1} + b^L), \tag{3.2.5}$$

for simplicity we let,

$$z^L = W^L a^{L-1} + b^L \tag{3.2.6}$$

then

$$a^L = \sigma(z^L). \tag{3.2.7}$$

The sensitivity of the cost function with changes to the weight W^L is calculated as follows:

$$\frac{\partial C_0}{\partial W^L} = \frac{\partial z^L}{\partial W^L}\frac{\partial a^L}{\partial z^L}\frac{\partial C_o}{\partial a^L}. \tag{3.2.8}$$

4. Methodology

4.1 Data source

The data used in this project was obtained from kaggle.com (Lending loan club)[1]. The data consists of 2260668 records and 145 attributes with 2260637 instances being non-defaults and 31 being defaults. Figure 4.1 illustrate the two classes in the dataset.

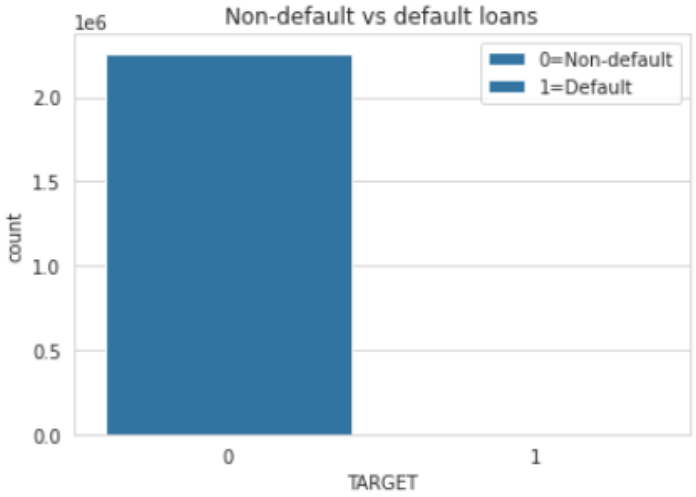


Figure 4.1: Non-default and default classes in the dataset

A total of 11 attributes have been selected which are relevant to the problem discussed in the paper based on the review of features discussed earlier in subsection 2.2. The dataset consists of the independent and dependent variables.

The independent variables are: loan amount, funded amount, funded amount by investors, term, interest rate, instalment, grade, employment length, annual income, issue date, and application type.

The dependent variable is: TARGET(0 and 1) where 1 indicates a borrower will default and 0 indicates a borrower will not default.

4.2 Data preprocessing

4.2.1 Handling missing data.

The dataset contained missing values. However, there are different methods for handling missing data. These are deletion, direct estimation, and imputation techniques (Buhi et al., 2008). Deletion involves discarding cases that contain missing data. Imputation replace missing values by values defined by a rule. The most common imputation techniques are mean and median. These use the mean or median of the available instances to replace the missing values. However, these are useful for instances where the number of missing values is low. Otherwise, they may result in loss of variation within the dataset.

In this project, we used deletion and median imputation to replace missing values.

[1]Lending loan club https://www.kaggle.com/pragyanbo/a-hitchhiker-s-guide-to-lending-club-loan-data.

4.2.2 Splitting data.

In this project, we split our dataset into training and testing with a ratio of 70% to 30%. Due to the severe data imbalance, the 70%:30% data split was the best option for this project.

4.2.3 Balancing of data.

Some of the challenges that are related to imbalanced data are:

•**misleading accuracy:** this is when the model makes predictions it fails to correctly classify any instances of the minority class.

•**biased predictions:** that is,the results of the model favours a certain class, most of the time its the majority class.

In this project, we found that the data imbalance ratio of the two class within our dataset is 0.01%:99.99% respectively. We also observed that when modeling the data, we obtain an accuracy of 99.99% which is misleading. Hence we used sampling to balance the data. Sampling is a commonly used technique for handling data imbalance. It can be categorized into oversampling and under-sampling.

Under-sampling techniques

1. **Random under-sampling:** This method removes some of the instances of the majority so that they balance with the minority class (Mishra, 2017). However, this could lead to discarding cases that may contain important information resulting in a bias in the model.

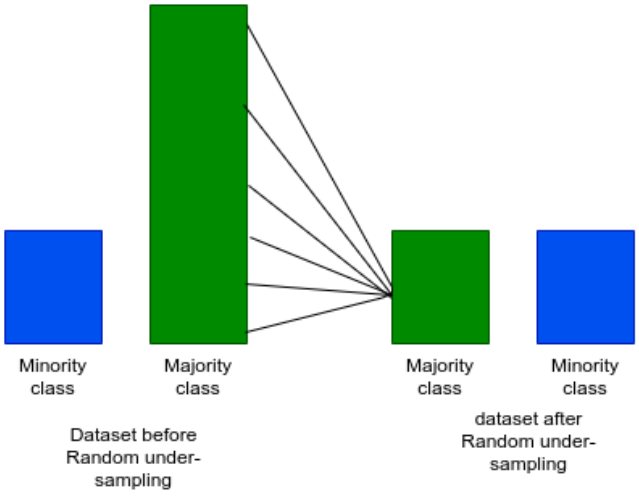


Figure 4.2: An illustration of Random under-sampling.

2. **Nearmiss:** This method calculates the distance between each minority and majority instances, thereafter the instances with the smallest distance in the majority class are selected whereas the instances with a huge distance are deleted.

3. **Tomek links:** Tomek et al. (1976) introduced the Tomek links method. If x and y are two instances belonging to the minority and majority class respectively with distance $d(x, y)$ between them, Batista et al. (2004) shows these instances form a pairwise sample called a Tomek link whenever there is no other instance z in the dataset such that,

$$d(x,z) < d(x,y) \quad \text{or} \quad d(y,z) < d(x,y). \tag{4.2.1}$$

This method eliminates the instances from the majority class within the Tomek link.

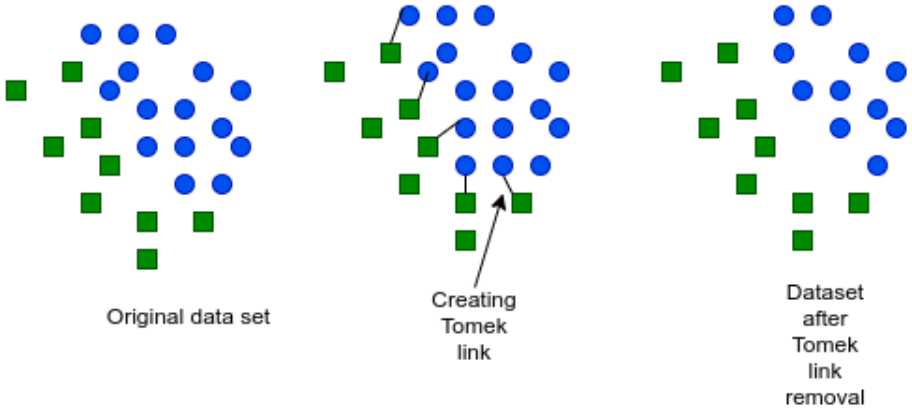


Figure 4.3: An illustration of Tomek links.

4. **Edited Nearest Neighbor (ENN):** The Edited Nearest Neighbour (ENN) method was introduced by Wilson (1972). Given instances from the minority and majority class, if any instance from the minority class has three nearest neighbors from the majority class, the three majority class instances are deleted. ENN increases the classification accuracy of the minority class.

5. **Condensed Nearest Neighbor(CNN):** Hart (1968) introduced the Condensed Nearest Neighbour Rule(CNN).This method works in the following way:

   (a) Given a training set X and a subset of the training set M,the method looks for a data point x whose nearest neighbour from M has a different class.

   (b) The point x is then removed from X and placed in M.

   (c) this process is repeated until no more data points are added to M.

   (d) The set M is then used for classification rather than X.

Over-sampling techniques

There are different oversampling techniques but we will be focusing on only three types, namely random oversampling, Synthetic Minority Oversampling Technique (SMOTE), and Adaptive Synthetic approach (ADASYN).

1. **Random oversampling:** Random oversampling makes multiple copies of the minority classes of the training data to adjust the class distribution of the dataset (Liu et al., 2007). This is one of the oldest methods and it is proven to be powerful. Other than duplicating every sample in the minority class, some of the samples may be randomly chosen with replacement. Figure 4.4 illustrates random oversampling.

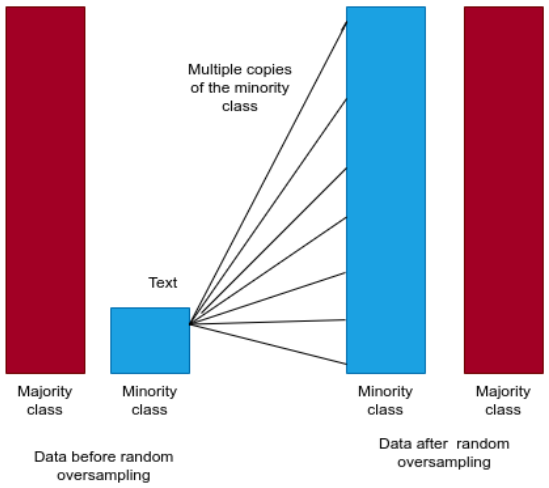


Figure 4.4: An illustration of random oversampling.

2. **SMOTE:** Suppose we have some training data s and f features in the feature space of data. Data is oversampled by taking the feature vector and the k neighbors of the feature vector in such a way that the difference between one of the k neighbors and the current data point is taken, multiplied by random number x which lies between 0 and 1. This creates the synthetic data points which are now added to the dataset (Chawla, 2009). SMOTE is illustrated in Figure 4.5.

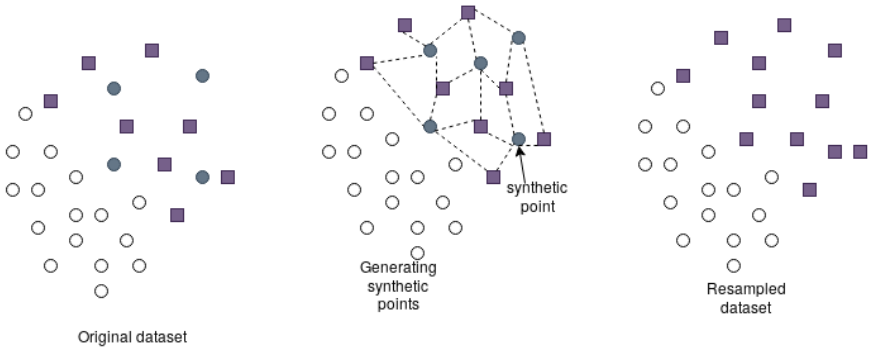


Figure 4.5: An illustration of SMOTE.

3. **ADYSYN:** This technique is an improvement of SMOTE where more focus is placed on the minority class during classification. This method generates more synthetic data points for the instances that are harder to learn in the minority class (He et al., 2008). However, this could lead to oversampling.

In this project we used under-sampling, over-sampling, and combination of over-and-under-sampling (hybrid) techniques. The under-sampling techniques used are Nearmiss and Edited Nearest neighbour combined with Condensed Nearest Neighbour (ENN-CNN). The hybrid technique used is SMOTE and Tomeklinks (SMOTEK). The oversampling technique used is ADASYN.

4.2.4 Activation functions.

Activation functions are used in neural networks to compute the weighted sum of input and biases, of

which is used to decide if a neuron can be activated or not. For multiple inputs the weighted sum is given by:

$$\sum_{i=1}^{n} x_i W_i + b_i = (x_1 W_1 + x_2 W_2 + x_3 W_3 + ... + x_n W_n) + (b_1 + b_2 + ... + b_n), \tag{4.2.2}$$

where x_i is the inputs, W_i is the weights, and b_i is the biases.

The most common activation functions are:

•**Linear** This function is illustrated by a straight line where the output is directly proportional to the input.However this function does perform well with back propagation as all of its derivatives are constants.

•**Binary step** Binary step is a classifier based on threshold base. A threshold value is assigned to decide if an output neuron should be activated or deactivated. It can be expressed as follows (Leung, 2008):

$$f(x) = 1 \quad \text{if } x > 0 \quad \text{, else} \quad 0 \quad \text{if} \quad x < 0. \tag{4.2.3}$$

•**ReLU** Rectified linear unit(ReLU) is a function with range 0 to infinity. It converts all the negative values to zeros, and allows the network to converge very fast.

•**Sigmoid**

A sigmoid is a function that is bounded and differentiable for all real values and has a positive derivative everywhere (Han and Moraga, 1995). In machine learning, sigmoid functions predict probabilities by compressing the output to be between 0 and 1. The sigmoid function can be expressed mathematically as:

$$S(x) = \frac{e^x}{1 + e^x}. \tag{4.2.4}$$

•**Softmax** Softmax also compresses the output to be a range of values between 0 and 1 where the probability of the values adds up to 1. Unlike the sigmoid function, softmax is used for multi-class classification. The softmax function can be expressed mathematically as:

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}, i = 1, ..., K, \quad x = (x_1, ..., x_K) \in \mathbb{R}^K. \tag{4.2.5}$$

•**Tanh** The tanh works almost similar to the sigmoid function. It predicts probabilities by compressing negative input into negative number only and it ranges between -1 and 1.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{4.2.6}$$

All of the above activation functions are illustrated in Figure 4.6,

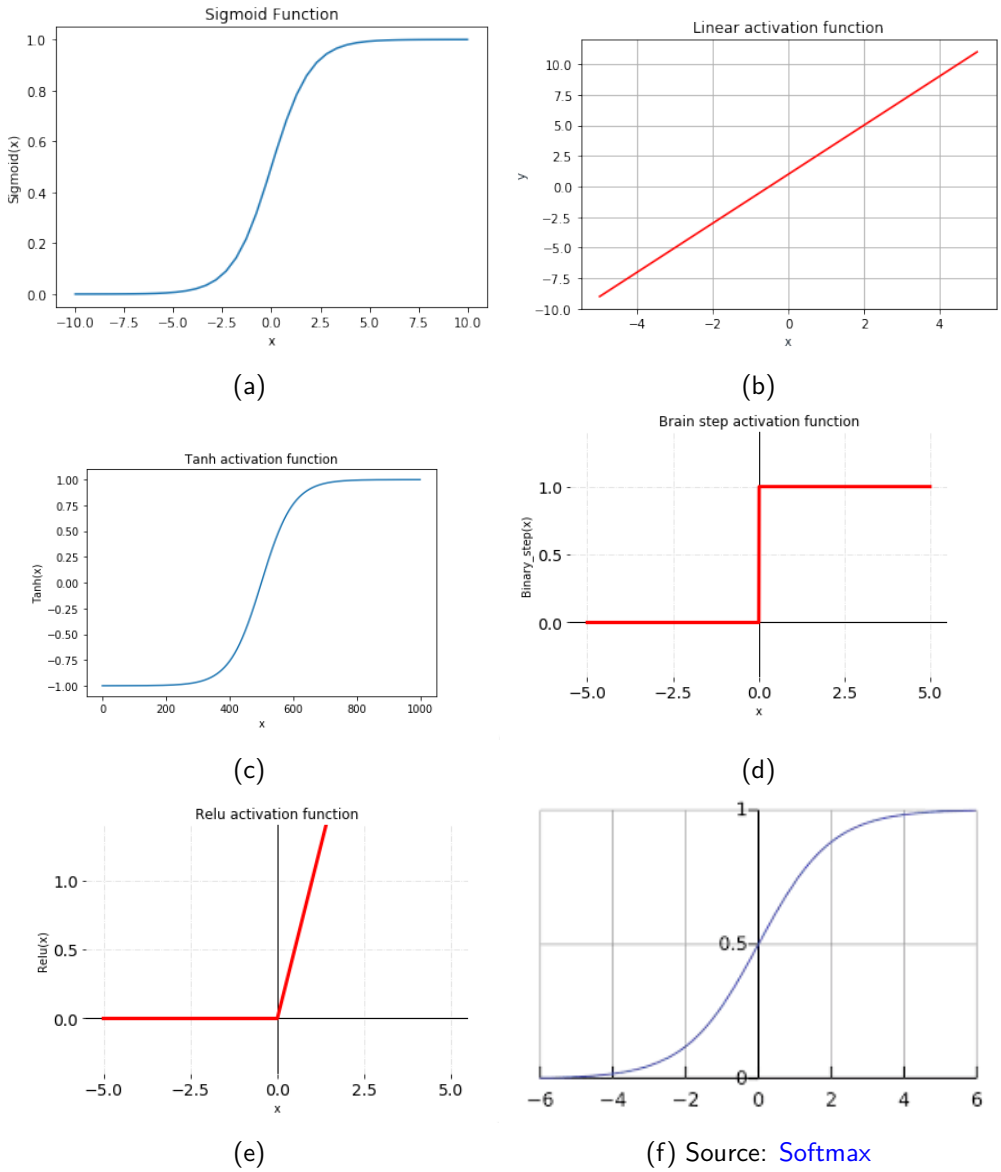


Figure 4.6: An illustration of activation functions; (a) Sigmoid, (b) Linear, (c) Tanh, (d) Binary step, (e) ReLU and (f) Softmax.

In this project we used ReLU activation function in the input and hidden layers, for the output layer we used sigmoid because we wanted to classify our dataset into two classes that are bounded between 0 and 1.

4.2.5 Model Implementation.

In this project we used a multi-layer feed-forward neural network with seven hidden layers and one neuron at the output layer and 10 neurons in the input layer for the classification. Due to limited resources we tuned all our hyper-parameters manually.

4.3 Evaluation metrics

Evaluation measure the performance of statistical or machine learning models. Below are the different types of evaluation metrics.

•**Accuracy:** This is a measure of the number of correct predictions out of the total of predictions made.

•**Precision:** Precision is the proportion of the positive instances that were correctly detected. In this project precision will indicate how good or poor a model performs in classifying default loans. Therefore, we require this value to be high as this will indicate that more default loans were correctly classified. Precision is calculated as follows (Davis and Goadrich, 2006):

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}, \tag{4.3.1}$$

where,

True Positive (TP): All instances that are positive in ground truth and are correctly classified as positive. For example,in this project all instances that are default turn out to be default.

True Negative (TN): These are all the instances that are classified as negative and turn out to be negative.

False Positive (FN): This is a representation of all the instances that are classified to be negative but turn out to be positive.

False Negative (TN): These are all the instances that are classified as negative and turn out to be negative.

•**Recall:** Recall is the proportion of the positive instances that were detected. In this project recall will indicate how good or poorly a model detects default loans. Since our data is imbalanced, we require this value to be high as this will indicate that more default loans were detected and show that there is no bias towards the default class. Recall is calculated as follows (Davis and Goadrich, 2006):

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \tag{4.3.2}$$

•**F1-score** The F1-score shows the balance between the recall and precision score by calculating the weighted average of both. It can be calculated as follows (Chicco and Jurman, 2020):

$$\text{F1-score} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right). \tag{4.3.3}$$

•**Confusion matrix** The confusion matrix shows the distribution of values in a model. It is illustrated in Figure 4.7.

	Class 1 *Predicted*	*Class 2* *Predicted*
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Figure 4.7: A confusion matrix.

Due to the severe data imbalance in this project we did not use accuracy to evaluate the performance of our model, as it will be biased towards the minority class. As a result the accuracy score will be high, which is misleading. We used precision score, recall score, F1-score and confusion matrix.

5. Results and Discussions

In this chapter we discuss the results obtained from the four classifiers, Artificial neural Networks (ANN), K-Nearest neighbour (KNN), Naive Bayes (NB) and Random Forest (RF) combined with the four resampling techniques.

The code for the Artificial neural networks and resampling methods is found here: [1].

The code for the Machine learning classifiers and resampling methods is found here: [2].

5.1 Nearmiss

5.1.1 Nearmiss+ANN.

The table below,Table 5.1,we observe that the accuracy and recall score are 50% which is misleading because precision score and F1-score are very low.This is an indication that the default class was incorrectly classified.

Precision score	0.00
Recall score	0.50
F1 score	0.00

Table 5.1: Evaluation metrics for Nearmiss+ANN

5.1.2 Nearmiss+RF.

From Table 5.1 we observe that Nearmiss and Random Forest failed to classify the default class.The model detected 58% instances of the default to be true however the precision is 0%,meaning the classification made by the model is completely not true.The F1-score further shows that the none of the predictions made by the model were true.This is in the confusion matrix were 435519 instances of the default class were classified as non-default.

Classes	Precision	Recall	F1-score
Non-default	1.00	0.35	0.52
Default	0.00	0.58	0.00

Table 5.2: Evaluation metrics for Nearmiss+Random Forest

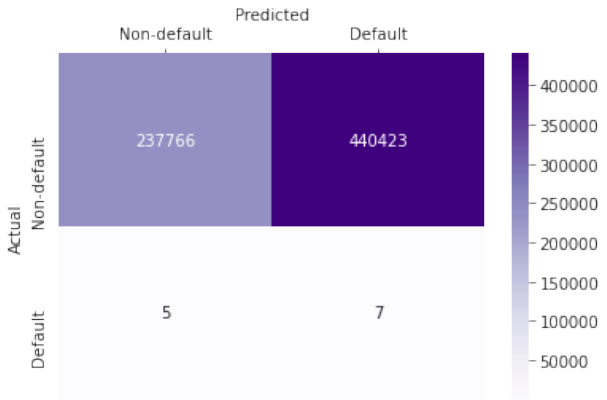


Figure 5.1: Confusion matrix of Nearmiss+random Forest

[1] Resampling and ANN's https://colab.research.google.com/drive/1rCRuCeK3C0l8AboprGCfpGgAwZKIGCXm?usp=sharing.

[2] Resampling and Machine learning classifiers https://colab.research.google.com/drive/13vIS2q62nqNrDMxu_2TRRFeOahICA9aL?usp=sharing.

5.1.3 Nearmiss+NB.

Moreover NearMiss and Naive Bayes incorrectly classified the default class,where the recall score is 83%. However the F1-score and the precision are both 0%. We also observed that the recall score increased from 67% to 83%,indicating a high misclassification percentage when using Nearmiss+ NB. In Figure 5.2 it is seen that 454791 default instances were classified as non-defaults.

Classes	Precision	Recall	F1-score
Non-default	1.00	0.33	0.50
Default	0.00	0.83	0.00

Table 5.3: Evaluation metrics for Nearmiss+NB

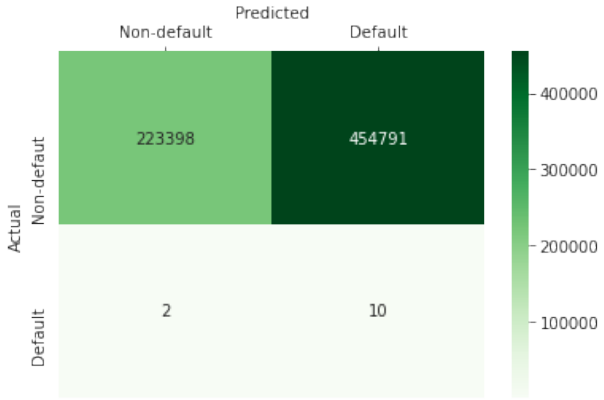


Figure 5.2: Confusion matrix of Nearmiss+KNN

5.1.4 Nearmiss+KNN.

With the combination of NearMiss and KNN the recall score decreases to 75%.However this is not good enough as the precision and F1-score remain at 0%. Again the model misclassified the default class.

Classes	Precision	Recall	F1-score
Non-default	1.00	0.30	0.46
Default	0.00	0.75	0.00

Table 5.4: Evaluation metrics for Nearmiss+KNN

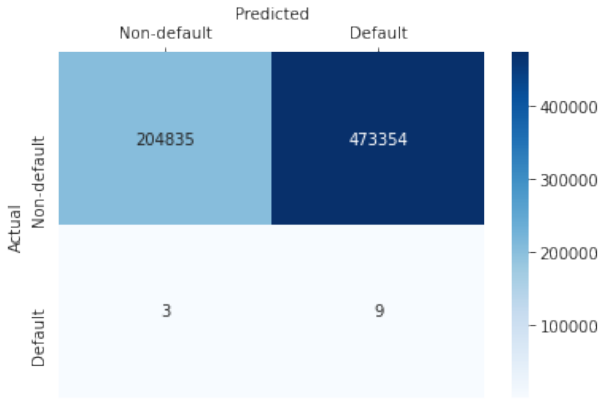


Figure 5.3: Confusion matrix of Nearmiss+NB

5.2 SMOTEK

5.2.1 SMOTEK+ANN.

Table 5.5 shows that the model failed to classify the default class. Both the precision score and F1-score are zero, meaning all the 50% default instances that were detected turned out to be non-default.

Precision score	0.00
Recall score	0.50
F1 score	0.00

Table 5.5: Evaluation metrics for SMOTEK+ANN

5.2.2 SMOTEK+RF.

In Table 5.6 and Figure 5.4 we observed that the model completely failed to detect instances of the default class. None of the instances in the default class were classified to be default.

Classes	Precision	Recall	F1-score
Non-default	1.00	1.00	1.00
Default	0.00	0.00	0.00

Table 5.6: Evaluation metrics for SMOTEK+RF

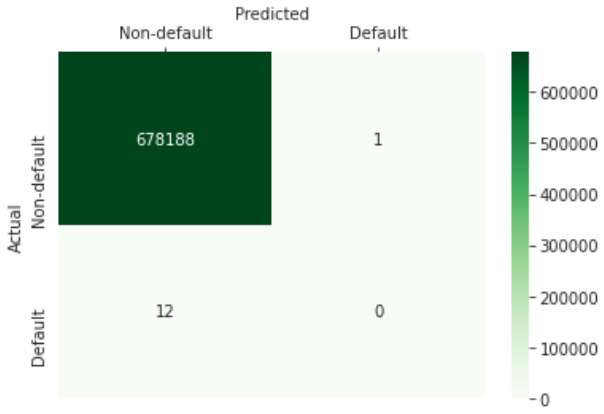


Figure 5.4: Confusion matrix of SMOTEK+RF

5.2.3 SMOTEK+NB.

In Figure 5.5 we observe that only 9 instances of the default class were classified to be truly default whereas 398995 instances of the non-default class were classified as default. Table 5.7 also shows that 75% of the instances of the default class were misclassified. This model performed poorly.

Classes	Precision	Recall	F1-score
Non-default	1.00	0.41	0.58
Default	0.00	0.75	0.00

Table 5.7: Evaluation metrics for SMOTEK+NB

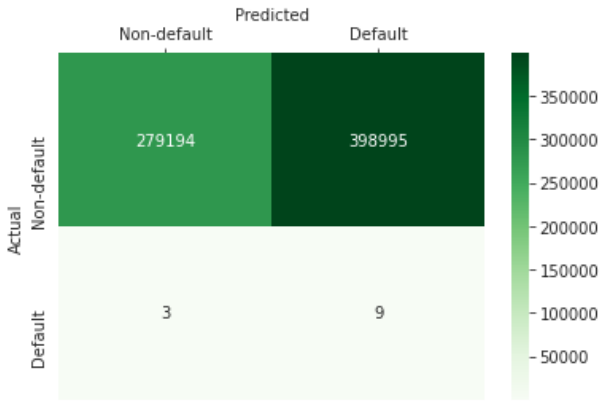


Figure 5.5: Confusion matrix of SMOTEK+NB

5.2.4 SMOTEK+KNN.

Figure 5.6, SMOTEK+KNN failed to classify the default class. None of the instances of the default class were correctly classified as default. This is evident in Table 5.8, were the precision, recall and F1-score are zeros.

Classes	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	0.00	0.00	0.00

Table 5.8: Evaluation metrics for SMOTEK+KNN

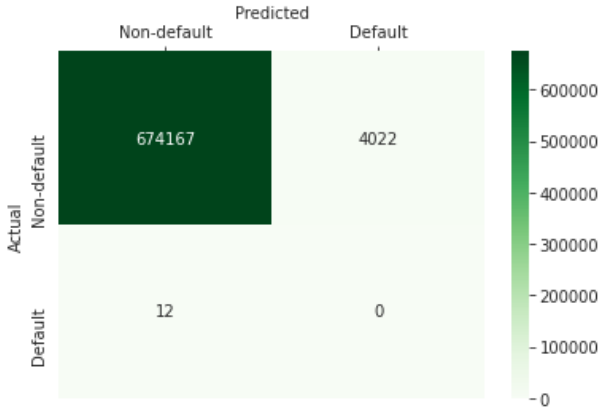


Figure 5.6: Confusion matrix of SMOTEK+KNN

5.3 ENN-CNN

- **ENN-CNN+ANN**

Precision score	0.00
Recall score	0.00
F1 score	0.00

Table 5.9: Evaluation metrics for ENN-CNN+ANN

•ENN-CNN+RF

Classes	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	0.00	0.00	0.00

Table 5.10: Evaluation metrics for ENN-CNN+RF

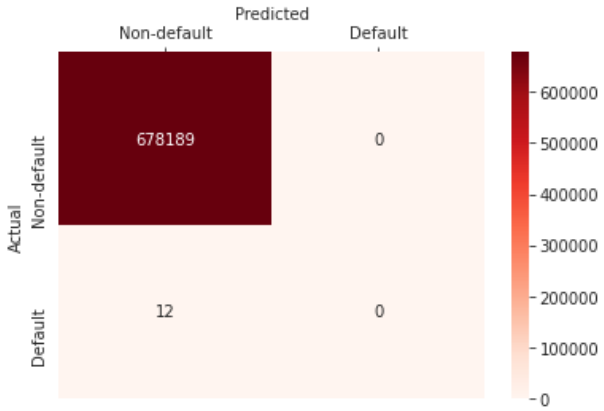


Figure 5.7: Confusion matrix of ENN-CNN+RF

•ENN-CNN+NB

Classes	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	0.00	0.00	0.00

Table 5.11: Evaluation metrics for ENN-CNN+NB

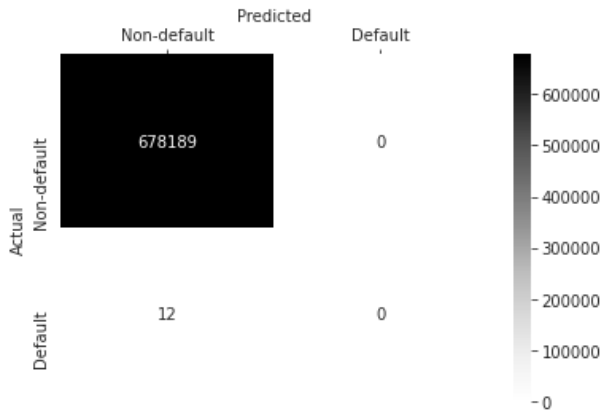


Figure 5.8: Confusion matrix of ENN-CNN+NB

•ENN-CNN+KNN

Classes	Precision	Recall	F1-score
0	1.00	1.00	1.00
1	0.00	0.00	0.00

Table 5.12: Evaluation metrics for ENN-CNN+KNN

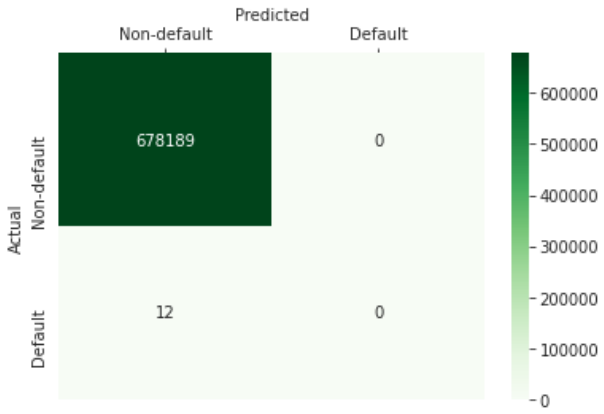


Figure 5.9: Confusion matrix ENN-CNN+KNN

All of the above models completely failed to classify the default class. It is seen in Table 5.9, 5.10, 5.11 and, 5.12 that the precision,recall and F1-score are zeros. We also observed that none of the default instances were detected to be truly default in figures, Figure 5.7, 5.8, and 5.9.

5.4 ADASYN

5.4.1 ADASYN+ANN.

Below in Table 5.13, it is seen that the model fails to classify default loans. The precision score and the F1-score are both zero. Given that the recall score is also low, the model performed poorly.

Precision score	0.00
Recall score	0.00
F1 score	0.00

Table 5.13: Evaluation metrics for SMOTEK+ANN

5.4.2 ADASYN+KNN.

In Table 5.14,precision, recall and, F1-score of the default class are all zeros. The model did not predict any instances of the default class.Figure 5.10 also show that the model failed to classify the default class. It is seen that 1756 non-default instances were classified as default, 12 default instances were classified as non-default, and none of the default instances that were classified turned out to be default.

Classes	Precision	Recall	F1-score
Non-default	1.00	0.99	1.00
default	0.00	0.00	0.00

Table 5.14: Evaluation metrics ADASYN+KNN

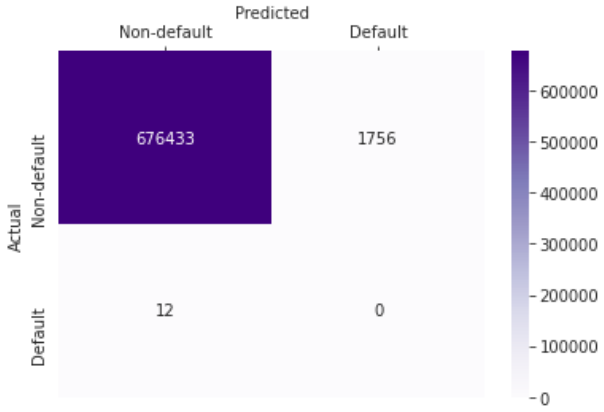


Figure 5.10: Confusion matrix ADASYN+KNN

5.4.3 ADASYN+RF.

In Table 5.15 and Figure 5.11 it is seen that the model did not classify the default class. The recall score, precision and F1-score are all zero. In the confusion matrix none of the fault instances were detected to be default.

Classes	Precision	Recall	F1-score
Non-default	1.00	1.00	1.00
Default	0.00	0.00	0.00

Table 5.15: Evaluation metrics ADASYN+RF

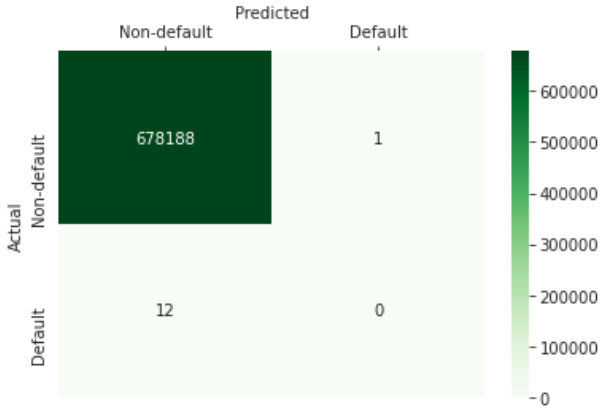


Figure 5.11: Confusion matrix ADASYN+RF

5.4.4 ADASYN+NB.

In Table 5.16 we observed that the model incorrectly classified the default class as the recall score is 75% but the precision and F1-score are both zero. It is also observed in the confusion matrix,Figure 5.12, show that 398995 non-default loans were classified as default. Three default loans were classified as non-default and only 9 default loans were correctly classified.

Classes	Precision	Recall	F1-score
Non-default	1.00	0.41	0.58
Default	0.00	0.75	0.00

Table 5.16: Evaluation metrics ADASYN+NB

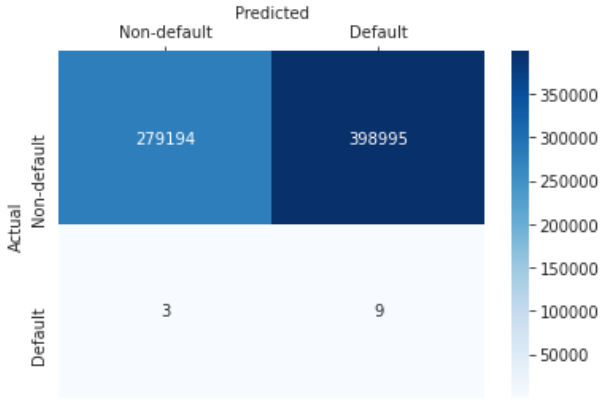


Figure 5.12: Confusion matrix ADASYN+NB

5.4.5 Summary of results.

Classifier	Resampling technique	Precision	Recall	F1-score
ANN	SMOTE and Tomek links	0.00	0.50	0.00
	ENN and CNN	0.00	0.00	0.00
	ADASYN	0.00	0.00	0.00
	Nearmiss	0.00	0.50	0.00
KNN	SMOTE and Tomek links	0.00	0.00	0.00
	ENN and CNN	0.00	0.00	0.00
	ADASYN	0.00	0.00	0.00
	Nearmiss	0.00	0.75	0.00
Random Forest	SMOTE and Tomek links	0.00	0.00	0.00
	ENN and CNN	0.00	0.00	0.00
	ADASYN	0.00	0.00	0.00
	Nearmiss	0.00	0.58	0.00
Naive Bayes	SMOTE and Tomek links	0.00	0.75	0.00
	ENN and CNN	0.00	0.00	0.00
	ADASYN	0.00	0.75	0.00
	Nearmiss	0.00	0.83	0.00

All of the above classifiers combined with re-sampling techniques failed to classify any loans of the default class. However these models are subject to improvement in future.

6. Conclusion

The aim of the research was to predict credit risk using using artificial neural network algorithm. However the dataset was imbalanced data and this led to the model producing misleading accuracy. To solve the issue of imbalanced data we introduced re-sampling techniques,such as NearMiss, Synthetic Minority Oversampling Technique (SMOTE)and Tomek links,Edited Nearest Neighbor (ENN) and Condensed Nearest Neighbor (CNN), and Adaptive Synthetic approach (ADASYN). All of these re-sampling techniques were combined with other classifiers, namely, K-Nearest Neighbor (KNN), Random Forest (RF) and Naive Bayes.

In this project due to lack of resources we could not perform automated hyper parameter tuning on our artificial neural network. Due to the severe data imbalance,the above models failed to correctly classify instances of the default class. None of them was better than the other.

Based on the data challenges and lack of high performance computers, future work will consider improving model performance by sourcing good quality data from financial institutions and conducting hyper-parameter tuning using high performance computers.

Acknowledgements

I want to acknowledge AIMS and it's funders for supporting this work, as well as my supervisor, Prof Phillip Mashele from North-West University, my tutors Dr Rock Stephane KOFFI and Alice Nanyanzi for being there for me and guiding me in every step.

I am also grateful to my family for their massive support and encouraging words. I appreciate my friends and AIMS family for their support as well.

References

Albawi, S., Mohammed, T. A., and Al-Zawi, S. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. IEEE, 2017.

Alejo, R., García, V., Marqués, A., Sánchez, J., and Antonio-Velázquez, J. Making accurate credit risk predictions with cost-sensitive MLP neural networks. In *Management intelligent systems*, pages 1–8. Springer, 2013.

Artís, M., Guillen, M., and Martínez, J. A model for credit scoring: an application of discriminant analysis. *Questiio*, 01 2011.

Arya, S., Eckel, C., and Wichman, C. Anatomy of the credit score. *Journal of Economic Behavior & Organization*, 95:175–185, 2013.

Batista, G. E., Prati, R. C., and Monard, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

Breiman, L. Random forests. *Machine learning*, 45(1):5–32, 2001.

Buhi, E. R., Goodson, P., and Neilands, T. B. Out of sight, not out of mind: Strategies for handling missing data. *American journal of health behavior*, 32(1):83–92, 2008.

Chawla, N. V. Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 875–886. Springer, 2009.

Chicco, D. and Jurman, G. The advantages of the Matthews correlation coefficient (mcc) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):6, 2020.

Coetzee, G. Innovative approaches to delivering microfinance services: The case of capitec bank. *August), MicroSave*, 2003.

Cox, D. R. and Snell, E. J. *Analysis of binary data*, volume 32. CRC press, 1989.

Davis, J. and Goadrich, M. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine Learning*, pages 233–240, 2006.

Digital trends. What is an artificial neural network? here's everything you need to know. digital trends, https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/, Accessed 12 October 2020.

Draper, N. R. and Smith, H. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998.

Fensterstock, A. On the advantages of statistical-based models for credit risk analysis. *Moneywatch. com Available from: http://findarticles. com/p/articles/mi qa3857/is*, 200507, 2005.

Ferreira, L. E. B., Barddal, J. P., Gomes, H. M., and Enembreck, F. Improving credit risk prediction in online peer-to-peer (p2p) lending using imbalanced learning techniques. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 175–181. IEEE, 2017.

Ghatasheh, N. Business analytics using random forest trees for credit risk prediction: A comparison study. *International Journal of Advanced Science and Technology*, 72(2014):19–30, 2014.

Gupta, D. K. and Goyal, S. Credit risk prediction using artificial neural network algorithm. *International Journal of Modern Education and Computer Science*, 11(5):9, 2018.

Han, J. and Moraga, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International Workshop on Artificial Neural Networks*, pages 195–201. Springer, 1995.

Hart, P. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14 (3):515–516, 1968.

He, H., Bai, Y., Garcia, E. A., and Li, S. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008.

Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Hornik, K., Stinchcombe, M., White, H., et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Lehner, M. Group lending versus individual lending in microfinance. Technical report, SFB/TR 15 Discussion Paper, 2009.

Leung, K. M. Introduction to artificial neural networks. *Dalam cis. poly. edu. Diakses*, 14, 2008.

Lewis, D. D. Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer, 1998.

Liu, A., Ghosh, J., and Martin, C. E. Generative oversampling for mining imbalanced datasets. In *DMIN*, pages 66–72, 2007.

Longadge, R. and Dongre, S. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.

Medical Xpress. Why are neuron axons long and spindly? study shows they're optimizing signaling efficiency. Medical Xpress, https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html, Accessed 12 October 2020.

Mishra, S. Handling imbalanced data: SMOTE vs. random undersampling. *Int. Res. J. Eng. Technol*, 4(8):317–320, 2017.

Murphy, K. P. et al. Naive bayes classifiers. *University of British Columbia*, 18:60, 2006.

Namvar, A., Siami, M., Rabhi, F., and Naderpour, M. Credit risk prediction in an imbalanced social lending environment. *arXiv preprint arXiv:1805.00801*, 2018.

Nazari, M. and Alidadi, M. Measuring credit risk of bank customers using artificial neural network. *Journal of Management Research*, 5(2):17, 2013.

Nilsson, M. and Shan, Q. Credit risk analysis with machine learning techniques in peer-to-peer lending market, 2018.

Ogboi, C. and Unuafe, O. K. Impact of credit risk management and capital adequacy on the financial performance of commercial banks in Nigeria. *Journal of emerging issues in economics, finance and banking*, 2(3):703–717, 2013.

Parker, L. E. Notes on multilayer, feedforward neural networks. *CS494/594: Projects in Machine Learning*, 2006.

Rena, R. and Msoni, M. Global financial crises and its impact on the south african economy: A further update. *Journal of Economics*, 5(1):17–25, 2014.

Rish, I. et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

Roslan, A. H. and Karim, M. Z. Determinants of microcredit repayment in Malaysia: The case of Agrobank. *Humanity & Social Sciences Journal*, 4(1):45–52, 2009.

Schuster, M. and Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Spuchl'áková, E., Valašková, K., and Adamko, P. The credit risk and its measurement, hedging and monitoring. *Procedia Economics and Finance*, 24:675–681, 2015.

To, M. Credit risk management and bad debt controlling: case: Anz Vietnam. 2013.

Tomek, I. et al. Two modifications of CNN. 1976.

Wi, D. Applied logistic regression. 2000.

Wilson, D. L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972.

Yorek, N., Ugulu, I., and Aydin, H. Using self-organizing neural network map combined with ward's clustering algorithm for visualization of students' cognitive structural models about aliveness concept. *Computational Intelligence and Neuroscience*, 2016, 2016.