

# **Predicting Lapse Rate in Life Insurance Using Machine Learning Algorithms**

**Mahlodi Kgare**

61946966

**Submitted in accordance with the requirements for the degree**

***MSc in Statistics***

***in the***

***Department of Statistics***

***At the***



**Supervisor: Professor Bhekisipho Twala**

## Declaration

Name: Mahlodi Tears Kgare

Student Number: 61946966


Degree: MSc in Statistics

Exact wording of the title of the dissertation as appearing on the electronic copy submitted for examination: Predicting Lapse Rate in Life Insurance Using Machine Learning Algorithms.

I declare that the above dissertation is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I submitted the dissertation to originality checking software and that it falls within the accepted requirements for originality.

I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

Signature:  \_\_\_\_\_

Date: 07 September 2021

## **Acknowledgements**

Firstly, I would like to thank my supervisor Professor Bhekisipho Twala for his willingness to help, guide and advice throughout my research. I appreciate all his valuable inputs. I would also like to thank my family for continued support. Lastly, I would like to thank the Lord for giving me the strength to complete this research.

## Abstract

Policy lapse is a vital component in life insurance as it affects future pricing and impacts the solvency of the life insurer. Accurate prediction of lapse will help the insurers to implement personalised retention strategies based on the model's outcome. The major contribution of the dissertation is the empirical comparison and benchmark of nine machine learning classifier models (i.e. Decision Tree, Gradient Boost, Random Forest, Support Vector Machine trained with linear kernel, Support Vector Machine trained with polynomial kernels, Neural Network trained with Levenberg-Marquardt, Neural Network trained with backpropagation) with traditional algorithms (i.e., Logistic Regression with forward variable selection and Logistic Regression with backward variable selection) for life insurance lapse predictions. The models' accuracy was observed over two different insurer datasets with different distributions (Insurer 1 and Insurer 2) and different feature selection methodology namely, Principal Component Analysis (PCA) and Chi-squared. Accuracy, F-measure, sensitivity, specificity, and Receiver Operating Characteristics Curve (ROC) were used as performance measures. The results show the strong prediction ability of ensemble models (Gradient Boost and Random Forest) over single classifiers, and there is a strong indication that suitable parameter tuning and model boosting improve the model performance. The best overall classifier is Gradient Boosting with an accuracy of 92%, 76% and F-measure of 92%, 84% for Insurer 1 and Insurer 2 datasets, respectively. The study recommends the use of ensemble models instead of single model classifiers as they have been proven to work better when predicting life insurance lapses.

**Keywords:** Decision tree; generalised linear models, logistic regression; lapse; machine learning

## Table of Contents

<b>Declaration</b> .....	<b>ii</b>
<b>Acknowledgements</b> .....	<b>iii</b>
<b>Abstract</b> .....	<b>iv</b>
<b>List of Tables</b> .....	<b>viii</b>
<b>List of Figures</b> .....	<b>x</b>
<b>Abbreviations and Acronyms</b> .....	<b>xi</b>
<b>CHAPTER ONE</b> .....	<b>1</b>
<b>INTRODUCTION</b> .....	<b>1</b>
1.1 Background of Study .....	1
1.2 Problem Statement.....	2
1.3 Significance of Study .....	3
1.4 Aims and Objectives .....	3
1.5 Lapse Determinants .....	3
1.6 Insurance Solvency .....	4
1.7 International Financial Reporting Standards – Insurance Contracts.....	6
1.8 Internet of Things and Cyber Risk in Insurance.....	7
1.9 Big Data Governance .....	8
1.10 Overview of the Dissertation Structure .....	9
<b>CHAPTER TWO</b> .....	<b>10</b>
<b>LITERATURE REVIEW</b> .....	<b>10</b>
2.1 Methods.....	10
2.1.1 Logistic regression.....	11
2.1.2 Support vector machines.....	16
2.1.3 Neural networks.....	22
2.1.4 Decision trees.....	26
2.1.4.1 Decision trees pruning methods.....	27
2.1.4.2 Classification and regression trees.....	29
2.1.4.3 Iterative dichotomiser3 .....	29
2.1.4.4 C4.5.....	29

2.1.4.5 Chi-squared automatic interaction detector .....	30
2.1.5 Ensemble models .....	31
2.1.5.1 Random forest.....	31
2.1.5.2 Gradient boosting .....	32
2.1.6 Applications .....	32
2.1.6.1 Underwriting .....	32
2.1.6.2 Pricing optimisations .....	33
2.1.6.3 Customer lifetime value.....	33
2.1.6.4 Cancellations.....	34
2.1.7 Hybrid and ensemble models .....	34
2.1.8 Single classifiers comparisons.....	36
2.1.9 Model optimisations .....	37
2.1.10 Machine learning challenges .....	38
2.1.10.1 Imbalanced data.....	38
2.1.10.2 Overfitting.....	39
2.1.10.3 Missing data .....	40
2.1.11 Critical review .....	43
<b>CHAPTER THREE METHODOLOGY AND RESULTS .....</b>	<b>45</b>
3.1 Introduction.....	45
3.2 Model's Setup.....	45
3.2.1 Data.....	45
3.2.1.1 Dataset 1 – Insurer 1 .....	45
3.2.1.2 Dataset 2: Insurer 2.....	47
3.2.2 Data pre-processing .....	48
3.2.2.1 Missing data imputation.....	49
3.2.2.2 Replacing outliers.....	52
3.2.2.3 Categorical variables encoding .....	52
3.2.2.4 Feature scaling.....	53
3.2.2.5 Imbalanced data.....	54
3.2.3 Feature selection .....	55
3.2.3.1 Chi-square.....	55
3.2.3.2 Principal component analysis .....	56

3.2.4 Model training and validation .....	56
3.2.4.1 Logistic regression .....	56
3.2.4.2 Support vector machine .....	57
3.2.4.3 Neural network .....	58
3.2.4.4 Decision tree .....	58
3.2.4.5 Gradient boosting .....	58
3.2.4.6 Random forest.....	58
3.2.5 Performances measures.....	59
3.3 Results.....	61
3.3.1 Data analysis .....	61
3.3.1.1 Insurer 1: Data exploration .....	61
3.3.1.2 Insurer 2: Data exploration .....	63
3.3.1.3 Variable importance .....	63
3.3.2 Logistic regression.....	65
3.3.3 Support vector machine.....	65
3.3.4 Neural network .....	66
3.3.5 Trees models.....	67
3.3.6 Model comparisons .....	68
3.3.7 Results discussions .....	72
<b>CHAPTER FOUR.....</b>	<b>75</b>
<b>CONCLUSION .....</b>	<b>75</b>
4.1 Summary .....	75
4.2 Findings and Recommendations .....	75
4.3 Limitations and Future Work.....	77
<b>REFERENCES.....</b>	<b>78</b>

## List of Tables

Table 2.1: Characteristics of DT Methods .....	31
Table 2.2: Cross-validation.....	39
Table 2.3: Literature Summary .....	42
Table 3.1: Insurer 1 – Variable Statistics.....	47
Table 3.2: Insurer 2 – Variable Statistics.....	48
Table 3.3: Insurer 1 – Missing Values Pattern.....	50
Table 3.4: Variable Stats Before and After Imputation .....	51
Table 3.5: Dummy Variables Imputation Example .....	52
Table 3.6: Logistic Regression Setup.....	57
Table 3.7: Support Vector Machine Setup.....	57
Table 3.8: Confusion Matrix .....	59
Table 3.9: Insurer 1 – Lapses Per Year.....	62
Table 3.10: Insurer 1 – Lapses by Non-payments.....	63
Table 3.11: Insurer 2 – Lapses Per Age Group and Gender .....	63
Table 3.12: Insurer 1 – Fit Statistics: Logistic Regression.....	65
Table 3.13: Insurer 2 – Fit Statistics: Logistic Regression.....	65
Table 3.14: Insurer 1 – Fit Statistics: Support Vector Machine.....	66
Table 3.15: Insurer 2 – Fit Statistics: Support Vector Machine.....	66
Table 3.16: Insurer 1 – Fit Statistics: Neural Networks.....	66



Table 3.17: Insurer 2 – Fit Statistics: Neural Networks.....	67
Table 3.18: Insurer 1 – Fit Statistics: Tree Models.....	67
Table 3.19: Insurer 2 – Fit Statistics: Tree Models.....	68
Table 3.20: Average Model Performance (Training and Validation).....	69
Table 3.21: Accuracy Per Prediction Band.....	70
Table 3.22: Policies Per Prediction Band.....	70
Table 3.23: Area Under Curve.....	72

## List of Figures

Figure 2.1: Linearly Separable Data (Skilltohire, 2020) .....	17
Figure 2.2: Optimal Hyperplane (Skilltohire, 2020).....	18
Figure 2.3: Linearly Inseparable Data; Kernel Trick (Wilimitis, 2018) .....	18
Figure 2.4: High Level Overview of a 3 Layered Neural Network (Hongsheng, 2021) ..	23
Figure 3.1: Variable Selection Summary .....	55
Figure 3.2: Insurer 1 – Demographic Distribution .....	62
Figure 3.3: Insurer 1 – Variable Importance .....	64
Figure 3.4: Insurer 1 – Variable Importance .....	64
Figure 3.5: Insurer 1 – ROC Curve: Chi-square .....	71
Figure 3.6: Insurer 2 – ROC Curve: Chi-square .....	71

## Abbreviations and Acronyms

AI	:	Artificial Intelligence
ANN	:	Artificial Neural Network
ANOVA	:	Analysis Of Variance
ASE	:	Average Squared Error
AUC	:	Area Under Curve
CART	:	Classification And Regression Tree
CDF	:	Cumulative Distribution Function
CHAID	:	Chi-Squared Automatic Interaction Detector
Cloglog	:	Complementary Log-Log
DAC	:	Distribution Accuracy
DF	:	Degree Of Freedom
DT	:	Decision Trees
ERM	:	Empirical Risk Minimisation
FN	:	False Negative
FNN	:	Feedforward Neural Network
FP	:	False Positive
GB	:	Gradient Boost
GLM	:	Generalised Linear Models
ID3	:	Iterative Dichotomiser 3
IFRS	:	International Financial Reporting Standards
IoT	:	Internet Of Things
LR	:	Logistic Regression
LRT	:	Likelihood Ratio Test
MAR	:	Missing At Random
MCAR	:	Missing Completely at Random
MCR	:	Minimum Capital Requirement
ML	:	Machine Learning
MNAR	:	Missing Completely at Random
MPL	:	Multilayer Perceptron
NN	:	Neural Network
OECD	:	Organisation For Economic Co-Operation and Development
PAC	:	Prediction Accuracy
PCA	:	Principal Component Analysis
RBF	:	Radial Basis Function
ReLU	:	Rectified Linear Activation Function
RF	:	Random Forest
RNN	:	Recurring Neural Network

RNN	:	Recurring Neural Network
ROC	:	Receiver Operating Characteristic Curve
SAM	:	Solvency Assessments Management
SCR	:	Solvency Capital Requirement
SMOTE	:	Synthetic Minority Over-Sampling Technique
SOM	:	Self-Organising Maps
SRM	:	Structuctural Risk Minimisation
SVM	:	Support Vector Machine
Tanh	:	Tangent Hyperbolic Function
TN	:	True Negative
TP	:	True Positive
VIF	:	Variance Inflation Factor
XGBoost	:	Extreme Gradient Boost

# CHAPTER ONE

## INTRODUCTION

This chapter outlines the background of life insurance, lapses and machine learning (ML), the problem statement, aim and objectives of the study, the significance of the study, the general view of recent implementations within life insurance and the layout of the rest of the study.

### 1.1 Background of Study

Life insurance is a financial insurer that pays out a lump sum assured value to beneficiaries when a policyholder dies. It plays a significant role in families by providing financial assistance after the passing of a loved one. Life insurance is a vital component of the economy as it provides opportunities such as employment to marketing distributors, insurance brokers, and direct agents (Ogutu, 2012).

In the insurance industry, a lapse is defined as the cancellation of a policy cover due to non-payment of a premium (Financial Sector Conduct Authority, 2015). A policy does not necessarily lapse every time a premium payment has been missed. The policyholder is given a grace period to settle the payment prior to the lapse. The life insurance company is liable to pay out the benefit to the client, in the case where a claim is within the grace period.

Most life insurance companies in South Africa have a premium collection system that automatically debits the client's bank account on a specified day to avoid lapses. However, clients can dispute or reverse the payments. An increase in the lapse rate at an early stage of the policy will result in the insurance company not being able to recover the initial expenses incurred to obtain the policy or hidden costs which include advertising, company infrastructure such as the call centre, and administration costs (Vasudev, Bajaj

& Alegre Escolano, 2016). This may result in premium increases for future policies which impact the policyholders negatively.

Data has always been at the heart of the insurance industry but because the life insurance industry is highly regulated, it is difficult for insurers to adapt to new technologies. However, there has been an increase in the use of artificial intelligence, robotics, and ML in areas such as fraud detection, underwriting, and claims processing within the insurance industry. In this study, future lapse rates of life insurance companies are predicted using nine ML classification models, namely; Decision Tree (DT); Gradient Boost (GB); Random Forest (RF); Support Vector Machine models trained with linear kernel (SVM-Linear) and polynomial kernels (SVM-Polynomial), Neural Network (NN) models trained with Levenberg-Marquardt (NN-Levenberg) and backpropagation; and Logistic Regression (LR) models with variable selection through forward (LR-Forward) and backward (LR-Backward) processes. The models will be compared to each other based on their level of prediction accuracy and their generalisation ability. Two different datasets will be used for testing and validating the models.

## **1.2 Problem Statement**

Modelling lapses is important to manage and control future risks or uncertainties that may arise in the insurance business. An increase in the lapse rate directly affects the company's book size, pricing, statutory reserve, market-consistent embedded value, and other risk management decisions. A high rate of lapses will have a significant impact on premiums. It can damage the reputation of a company which will result in lesser new entrants and more policyholders lapsing (Eling & Kochanski, 2012). A huge number of unexpected lapses may result in possible liquidation and insolvency of the company (Barsotti, Milhaud & Salhi, 2016). It is important for life insurers to properly assess and model their exposure to lapse risks and understand cancellations behaviour as accurately as possible (Biagini, Huber, Jaspersen & Mazzon, 2021; Barsotti *et al.*, 2016).

### **1.3 Significance of Study**

The outcome of the dissertation will help the life insurance industry to make efficient retention decisions based on the models' outcomes, and minimise risks associated with losing customers. It will help life insurers to efficiently plan finances, minimise the prediction uncertainties and alert the insurers to early warnings of cancellations. In the case of accurate model predictions, customers will benefit from reduced premium rates and enjoy the benefits that come with retention strategies. The study will also contribute to the existing academic literature.

### **1.4 Aims and Objectives**

This dissertation aims to illustrate the predictive power of different ML classification models when predicting life insurance lapses; to measure the models' sensitivity and generalisation abilities using different life insurance datasets; to illustrate the impact of different feature selection methodology on the models and to highlight features that directly drive lapses using in-depth data analysis.

The hypothesis of the dissertation is to test that ML algorithms give better predictions than traditional methods when predicting lapses (i.e.,LR) and to test and compare the prediction power of ensemble models and single classifiers.

The below subsections (1.5-1.9) discuss some of the features that have been found to trigger lapses and what is happening generally within the insurance industry.

### **1.5 Lapse Determinants**

There are several reasons for policy lapses (Outreville, 1990; Carson & Forster, 2000; Russell, Fier, Carson & Dumm, 2013). These include the policyholder having found a competitive rate with another insurer, the policyholder becoming unemployed and no longer able to afford to pay premiums, and the insured no longer being interested in the product (Outreville, 1990; Carson & Forster, 2000; Russell *et al.*, 2013). Policy lapses may also be influenced by economic risk factors such as tax relief, financial markets,

interest rates, inflation, GDP, and dynamics such as contract features, a firm's reputation, competition, and regulations (Barsotti *et al.*, 2016).

Botha (2017) indicates that there is an increase in life insurance policy take ups amongst low-income earners, and this is the group that seems to be most at risk of lapsing policies. This has been highlighted as being unique to the South African context (Botha, 2017). The study also shows that income, savings, and debt are significant predictors of lapses.

Valdez, Vadiveloo and Dias (2014) claim that individuals who usually cancel their policies do so because they have had the opportunity to look elsewhere, whereas those that stay are usually at a higher risk of death. Individuals with health risks and uninsurable issues do not usually lapse their policies (Valdez *et al.*, 2014; Xong & Kang, 2019). Policy cancellations also depend on the age of the policyholder (Mojekwu, 2011). In Nigeria specifically, young people seem to take up policies and terminate them very early because of the economic challenges in Nigeria (Mojekwu, 2011), whereas in South Africa, young people do not consider taking up life insurance at an early age (Marx, 2018).

Eling and Kochanski (2012) indicate that most insurance studies focus on environmental variables that impact lapses and not necessarily policyholder characteristics as individual data is confidential and not easily accessible.

## **1.6 Insurance Solvency**

In South Africa, Solvency Assessments Management (SAM) has been implemented as a tool to monitor and avoid the insolvency of insurers (Sibindi, 2014). The SAM can be described as a risk-based regulatory framework for South African insurers that measures the financial soundness of a company (Deloitte Touche Tohmatsu Limited, 2016). It is largely based on Solvency II which is Europe's risk-based regulatory framework. Features of SAM are however specific to the South African market (Deloitte Touche Tohmatsu Limited, 2016).

SAM is based on three pillars: Pillar 1 measures the quantitative financial soundness of an insurer and is based on a company's balance sheet and capital requirements; pillar 2



aims to measure the qualitative soundness of the insurer and to establish a system of sound governance and risk management; and pillar 3 looks at the reporting and disclosures (Deloitte Touche Tohmatsu Limited, 2016; Jansen van Vuuren, Reyers & van Schalkwyk, 2017).

As part of the SAM balance sheet, insurers need to calculate two capital values namely Solvency Capital Requirement (SCR) and Minimum Capital Requirement (MCR). The MCR is the minimum capital that an insurer has to protect policyholders and continue to operate whereas the SCR is the minimum value that an insurer has to hold to remain solvent (Deloitte Touche Tohmatsu Limited, 2016). Different risks are considered as part of the calculation of the SCR namely, market risks and underwriting risks. Underwriting risks can be divided into different risks (Michorius, 2011) namely:

1. Lapse risk – this is the risk of loss associated with the rates of policy lapses, surrenders, terminations, and renewals. Lapse risk is a significant contributor to underwriting risks (EIOPA, 2011; Barsotti *et al.*, 2016).
2. Mortality risk – this is the risk of loss that is associated with mortality rates. An increase may result in an increase in insurance liability.
3. Longevity risk – this risk is also associated with mortality rates where a severe decrease may result in increased insurance liability.
4. Disability risk – the risk of loss that is associated with the rate of disabilities, sickness, and morbidity.
5. Life expense risk – the risk of loss that results from the expenses that are associated with servicing insurance and reinsurance contracts.
6. Life catastrophe risk – this is a risk of loss that is associated with catastrophic events that may occur; this risk may result in significant uncertainties in pricing (Michorius, 2011).
7. Retrenchment risk – this is a risk of loss that is associated with adverse change in insurance liabilities, resulting from changes of retrenchment inception rates used in pricing.

A study conducted by KPMG (2019) illustrates that most insurance companies become insolvent mainly because of poor risk and decision management, and by the time the company is declared insolvent there is usually nothing that can be done to save them. Barsotti *et al.* (2016) indicate that regulators and risk managers must understand lapse dynamics so that they can identify the real risks embedded in the life insurance contracts and exposure to massive lapses, surrenders, and cancellations.

### **1.7 International Financial Reporting Standards – Insurance Contracts**

The implementation of International Financial Reporting Standards (IFRS17) which is set to become effective in 2023 has been a headache for most insurers. The implementation affects more than 450 listed insurers that are using IFRS17 standards (Yeoh, 2017). These include life, non-life, and re-insurers. IFRS17 is a profit reporting tool that was implemented to give standard accounting reporting so it would be easy to compare business performances across the globe. The standards are set by the International Accounting Standards Board.

IFRS17 was implemented with the aim of better alignment, consistency, and transparency in the insurance industry. It is not aimed at changing how insurers run their businesses but how they report on them. It replaces the currently used IFRS4, which was introduced in 2004 as an interim standard with the aim of solving some of the comparison issues that were created by IFRS4.

The IFRS4 allows companies to use their own local accounting reporting to measure insurance contract issues (International Accounting Standard Board, 2017). The IFRS17 focuses mostly on three ideas namely, that the future cash flow of an insurer should be calculated based on current assumptions rather than historic liability calculations; measurements must allow for risk adjustments; and insurers must report on all financial earnings and not just the finances that have been received through contractual service margins (PWC, 2017).

According to the PWC (2020) report, IFRS17 has the potential to harness the data, improve financial reporting, and improve decision making. However, the implementation will affect a lot of business areas such as finance, actuarial systems, product designs, remuneration policies, budgeting, and forecasting methodologies.

### **1.8 Internet of Things and Cyber Risk in Insurance**

Internet of things (IoT) refers to the ability to connect devices, objects, and systems to other devices through the internet to leverage data collection. According to an article produced by Behm, Deetjen, Kaniyar, Methner and Münstermann (2019) from McKinsey & Company, IoT will change the world in the coming years and devices will be a huge part of that change. They also mention that 127 new devices are connecting to the internet every second (Behm *et al.*, 2019). Network devices owned by people increased from 12.5 billion in 2010 to 25 billion in 2015, and it is estimated to increase to 50 billion in 2025 (Behm *et al.*, 2019). IoT allows real-time data collection through several devices which could help insurers with real-time analysis that can improve accuracy in predictions, can reduce fraud, and can help facilitate processes like claims quickly. In life insurance specifically, some insurers have included wellness programmes to track blood pressure, daily steps, and other health routines through digital devices such as smartphones and wearable devices. With the huge amount of data being constantly updated and collected through IoT, insurance companies will be able to customise products for the insured and improve customer experience. Underwriting could provide real-time pricing and according to Morgan (2018), IoT can cut the cost of claims by 30% which can lead to decreased premiums. The huge concerns of IoT are regulations and data privacy.

As much as data, software, hardware, and IoT are generally increasing, the hacking of devices is also increasing. Cyber security is information technology security that protects cyber environments (e.g., systems, networks, and data) of an organisation from unauthorised attacks (Seemma, Nandhini & Sowmiya, 2018). Cyber risk has been mentioned by Fintechfutures (2019) as one of the top seven challenges that are faced by insurance companies. Similarly, PWC (2020) has ranked cyber risk as a principal concern of insurers in South Africa and it was on the top five concerns globally. Cyber risk is,

however, more of a concern in the short-term insurance industry than in the long-term insurance industry. Recently (i.e., August 2020), Experian, which is a credit information agency in South Africa, was exposed to a data breach that affected approximately 24 million South Africans and 793,749 business entities. Insurers collect a vast amount of personal data and therefore it is one of the prime targets for cyber criminals activities such as identity theft and financial gain through extortion (IAISConsultation, 2019). Cyber security incidents can damage the reputation of a business and cause disruption of the business which may result in a significant loss.

### **1.9 Big Data Governance**

The insurance sector is driven by a large, increasing amount of granular and detailed data, both structured and unstructured, which traditional processing technologies cannot handle (Badr , Mohamed & Mohamed, 2018). Big data technologies have changed the way insurers collect, analyse, and manage data effectively (Boodhun & Jayabalan, 2018). Traditionally, insurers used to deal with structured data only for analysis and business decisions. Currently, it is important to consider unstructured datasets that can be collected through social media which could have an impact on an insurer's brand, products, and customers' perception of the insured (Badr *et al.*, 2018). The current challenge of insurers is to identify the unstructured datasets that have the potential to give the greatest value (Badr *et al.*, 2018).

Big data is currently dominating areas such as fraud detection, pricing optimisation, customer experience and insight, automation, risk assessments, and marketing (Badr *et al.*, 2018). According to Boodhun and Jayabalan (2018), life insurance companies are still reliant on standard actuarial formulas to predict mortality rates, premiums, and lapses. However, there have been developments in carrying out predictive modelling to improve business performance. As much as big data is an exciting revelation in the insurance industry, the great benefits come with risks. Regulation and governance of such platforms are very important to protect both the policyholder and insurer. The Organisation for Economic Co-operation and Development (OECD) and the Commission's Independent High-Level Expert Group on Artificial Intelligence (HLAG AI) have published ethics

guidelines on areas of AI in insurance that should be monitored for trustworthy AI. Based on the guidelines, AI should be lawful (respecting all laws and regulations), ethical and robust, it must be able to contribute to a fair and just society, and all implementations must be traceable (OECD, 2020). The guidelines also illustrate requirements that big data and AI should meet for them to be certified as trustworthy (OECD, 2020):

1. Human agency and oversight – AI systems must empower people and respect their human rights.
2. Technical robustness and safety – AI systems must be secure, accurate, reliable, resilient, and reproducible to minimise unintentional harm.
3. Privacy and data governance – adequate mechanisms and policies must be implemented to ensure data protection and privacy and prevent misuse of data.
4. Transparency – the system must be always transparent, ensuring that users are aware when they are interacting with AI systems, and they should know all the capabilities and limitations of the system.
5. Diversity, non-discrimination, and fairness – AI systems must be fair and be accessible to all.
6. Societal and environmental well-being – AI systems must be environmentally friendly, and they should take into account their social and societal impact.
7. Accountability – mechanisms should be put in place to ensure accountability of AI systems and their outcomes. A proper audit must be done on the data, algorithms, and processes.

### **1.10 Overview of the Dissertation Structure**

The rest of the dissertation is structured as follows; Chapter Two reviews literature, highlighting the detailed theory of ML models and their applications in the life insurance industry. It also discusses the gaps in previous literature and how this dissertation fills those gaps. Chapter Three discusses the models' setup and gives a detailed view of how the models were assessed. It also discusses and interprets the models' results. Chapter Four concludes the dissertation with a summary, findings, limitations of the study and future work.

## **CHAPTER TWO**

### **LITERATURE REVIEW**

This chapter discusses ML algorithms and critically reviews the methods that were considered for this study i.e., LR, support vector machine and NN. It also discusses their common applications within the life insurance industry, their successes, their general limitations, and limitations observed within life insurance. The chapter further discusses the common challenges that may arise when setting up a ML experiment based on literature.

#### **2.1 Methods**

Artificial Intelligence (AI) and ML have become part of everyday life and their applications are likely to increase in the near future. Machine learning (ML) is based on a hypothesis that machines should learn and improve through experience (Alzubi Nayyar & Kumar, 2018; Lake, Ullman, Tenenbaum & Gershman, 2017). There are three types of learning, namely, supervised learning, unsupervised learning, and reinforcement learning (Burri, Burri, Bojja & Buruga, 2019). Supervised learning is a ML task that uses a training dataset to learn the mapping function from input to target (labelled responses) by looking at several input-output examples (Nasteski, 2017). Unsupervised learning is a ML task that only considers input data without a labelled target to make inferences (Nasteski, 2017). Supervised learning is the most used ML technique in the insurance industry (Burri *et al.*, 2019).

Machine learning (ML) is a well-established concept in the life insurance industry. However, insurers use mostly Generalised Linear Models (GLMs) for lapse predictions. A GLM is defined as an algorithm that models the relationship between a dependable variable whose outcome will be predicted (target variable) and one or more explanatory variables (Goldburd, Khare, Tevet & Guller, 2016). It is a statistical technique introduced by Nelder and Wedderburn (1972). It was first applied in insurance rating by Cheek, McCullagh and Nelder (1990). Generalised Linear Models (GLMs) have been widely

applied since then in non-life insurance rates and have become a standard tool for ratings (Duan, Chang, Wang, Chen & Zhao, 2018). Examples of GLMs include linear regression, Analysis of Variance (ANOVA), Analysis of Covariance, LR, Poisson Regression, and multinomial response. Actuaries in the insurances sector prefer GLMs to model lapses because GLMs can capture many input variables, they can capture interactions between input variables, they can easily be translated to actuarial software such as prophet and EARNIX, and computational time is lesser than most ML algorithms (Ducuroir, Zians & Miller, 2016; Hendrych, 2019).

### **2.1.1 Logistic regression**

Logistic Regression (LR) is a modelling technique that predicts the probability of a binary response based on one or more independent variables using a link function, meaning that there can only be two outcomes, 0 or 1; therefore, in this dissertation, lapse or non-lapse (Chakure, 2019). Predicting lapse rates is a classification problem. Classification models are predominantly used in data science, making them a principal component of ML (Soofi & Awan, 2017). The main aim of classification is to find the decision boundary that separates the data into distinct classes (Chakure, 2019).

In GLMs, a link function links nonlinear relations of responses (0;1) to linear predictors that are unbounded ( $-\infty, \infty$ ). There are three commonly used link functions in LR; namely, Logit, Probit and Complementary log-log functions (Damisa, Bello, Ajadi, Agboola, Tasi'u & Musa, 2017; Prasetyo, Kuswanto, Iriawan & Ulama, 2019; Mauchant, Rice, Riley, Leber, Samarov & Forster, 2011).

Logit function is the mathematical function that takes any form of a linear combination of predictor variables and converts it into two distinct classes (0,1). It uses the cumulative distribution function (CDF) of a standard logistic distribution to enforce probabilities to fall between 0 and 1 (Damisa *et al.*, 2017; Prasetyo *et al.*, 2019; Mauchant *et al.*, 2011).

It is described by equation 2.1 (Boateng & Abaye, 2019).

$$\text{logit}(y) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1\chi_1 + \dots + \beta_k\chi_k, \quad (2.1)$$

where,

- $y$ , is the dichotomous outcome.
- $\chi_1, \dots, \chi_k$ , are the predictor variables such as sum insured, premium frequency, entry age, policy term, gender, and type of cover.
- $\beta_0, \beta_1, \dots, \beta_k$ , are the regression (model) coefficients,  $\beta_0$  is the intercept.
- $p$  is the proportion of the data with lapse outcome,  $1 - p$  is the probability of non-lapse. The ratio (i. e.  $\frac{p}{1-p}$ ) is called the ratio of odds and the logit is therefore the logarithm of odds. It measures the strength of the relationship between the predictor and response variables (Young, Simon & Pardoe, 2014).

The inverse of a logit function is a logistic function. It is often called a sigmoid function and resembles an s-shape (Bernstein, 2016).

Probit link uses the inverse CDF of a standard normal distribution to enforce probabilities to fall between 0 and 1 (Damisa *et al.*, 2017; Prasetyo *et al.*, 2019; Mauchant *et al.*, 2011).



It is described by equation 2.2 (Piegorisch, 1992).

$$\text{Probit}(y)=\Phi^{-1}(y), \quad (2.2)$$

Complementary log-log (cloglog) uses a cumulative function distribution of the standard extreme value-distribution to convert real numbers to 0;1 (Mauchant *et al.*, 2011). Cloglog is represented by equation 2.3 (Piegorisch, 1992).

$$c \log \log(y) = \log\{-\log(1 - p)\}, \quad (2.3)$$

Both logit and probit are symmetric, meaning that the link approaches 0 and 1 at the same pace whereas cloglog is asymmetric. Literature shows that logit and probit usually yield similar results because of their symmetric nature, whereas the cloglog function will give different results but similar substantive conclusions as both logit and probit functions (Gill, 2001; Mauchant *et al.*, 2011). Gill (2001) recommends using the logit function as it can handle outliers and the data with too much variability as compared to both probit and cloglog.

Fitting the LR model requires that we estimate the values of unknown coefficients or parameters,  $\beta_0, \beta_1, \dots, \beta_k$ . Parameters reflect the association between independent and dependent variables (Park, 2013). Data points are usually fixed, and one might need to play around with parameters to maximise probabilities. Least Square Estimation is the commonly used method to estimate parameters in a linear model. Logistic Regression (LR) parameters are commonly estimated using Maximum Likelihood Estimator (Bewick, Cheek & Ball, 2005).

Given unknown parameters  $\beta_0, \beta_1, \dots, \beta_k$ , a likelihood function tells us the probability of observing or reproducing the original data, thus, how well parameters explain the data (Park, 2013). For observed data  $x_1, \dots, x_n$  and  $y_1, \dots, y_n$ , the likelihood function is described by equation 2.4 (Park, 2013).

$$L = \prod_{i=1}^n p(y/x)^{Y_i} (1 - p(y/x))^{1-Y_i}, \quad (2.4)$$

Log of likelihood is represented by equation 2.5.

$$L = \log(L) = \sum_{i=1}^n Y_i \log[p(y/x)] + \left( n - \sum_{i=1}^n Y_i \right) \log[1 - p(y/x)], \quad (2.5)$$

The maximised likelihood estimator is the maximum value of the parameter for which the probability of reproducing the observed data is maximised.

Assessment of the overall significance of the fitted model can be determined by using the likelihood ratio test (LRT). The LRT tests the deviance of the likelihood under the full model, thus the model with all predictor variables and the likelihood of a null model, thus a model with intercepts only. The LRT is represented by equation 2.6 (Newsom, 2021).

$$\begin{aligned} G^2 &= \text{Deviance}_0 - \text{Deviance}_1, \\ &= -2 \ln \left( \frac{L_0}{L_1} \right) = [-2 \log(L_0)] - [-2 \log(L_1)], \end{aligned} \quad (2.6)$$

where  $L_1$  is the likelihood of the full model and  $L_0$  is the likelihood of the null model. The estimated value of  $G^2$  is approximately equal to the Chi-squared value with the degree of freedom (df) equal to the number of predictors in the model (Newsom, 2021). If

significant, that means the combination of predictors contributes significantly to the outcome. The LRT can also test the likelihood under the full model and the likelihood under the reduced model, where reduced means dropping some of the predictor variables.

Alternatively, the Wald test statistic is used to assess the goodness of fit and to assess the contribution of each predictor in the fitted model. It is the ratio of a squared coefficient to the squared standard error of the coefficient. It can be represented by equation 2.7 (Abbas & Mohammed, 2020).

$$W_j = \frac{\beta_j^2}{SE_{\beta_j}^2}, \quad (2.7)$$

A major challenge when building a logistic model is selecting the variables to be included in the model. Some researchers collect as many variables as possible for their logistic model, however, it is easier to miss the link between the explanatory variables and the event occurrence if the model has too many variables.

A model with too many features may result in increased multicollinearity, variables redundancy, overfitting of the model due to optimistic results on the training sets and not on testing and validation sets (Chowdhury & Turin, 2020). It is vital to ensure that all significant variables are used to train the model.

There are three common variable selection methodologies (i.e., forward variable selection, backward variable selection, and stepwise variable selection). In forward variable selection, the model is started with no variables, then iteratively adds the significant ones individually until the set stopping criteria (p-values less than the threshold) or until all the significant variables are added (Austin & Tu, 2004); whereas, with the backward variable selection, selection starts with the full set of variables, then iteratively eliminate the insignificant ones until stopping criteria or until there are no more variables to be eliminated. Stepwise is a combination of both forward and backward selection (Austin & Tu, 2004). It allows selection procedures to move in both forward and

backward directions. The process allows dropped variables to re-enter the model and be re-evaluated (Chowdhury & Turin, 2020).

Multicollinearity refers to a situation in which two or more explanatory variables are highly related to each other. In regressions, multicollinearity increases the standard errors of coefficients, making some independent variables to be less significant (Akinwande, Dikko, & Samson, 2015). A little bit of multicollinearity is usually not an issue (Akinwande *et al.*, 2015).

The use of Variance Inflation Factor (VIF) is commonly used to assess the amount of multicollinearity that exists within the model. The VIF reflects how much the variance of an estimated regression coefficient increases when predictors are correlated. A VIF greater than five reflects problematic multicollinearity and should be dealt with to decrease the multicollinearity that exists within the model. The solution to deal with problematic multicollinearity is to remove highly correlated predictor variables (Akinwande *et al.*, 2015).

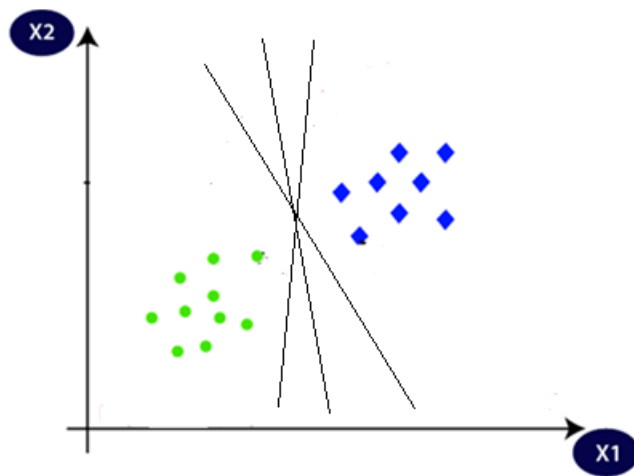
Some of the advantages of LR include their ability to provide good accuracy, they are quick to train, they are good at classifying unknown records, overfitting is usually minimal, they do not assume a linear relationship between independent and dependent variables, and they do not make assumptions about class distributions that are in the feature space (Schreiber-Gregory & Bader, 2018; Chakure, 2019). Some of the drawbacks include their sensitivity to outliers and the data must be large enough for a stable good performance.

### **2.1.2 Support vector machines**

Support vector machine (SVM) was first introduced by Boser, Guyon and Vapnik (1992) at the fifth annual association for computing machinery workshop. The study of SVM has become a very popular area in ML. It is well known for its strong ability to classify the data (Rustam & Ariantari, 2018). The algorithm is widely used in mathematics, science, biology, finance, economics, and biotechnology. It can model complex problems such as text classification, hand-writing recognition, and complex numbers (Girma, 2009). It can

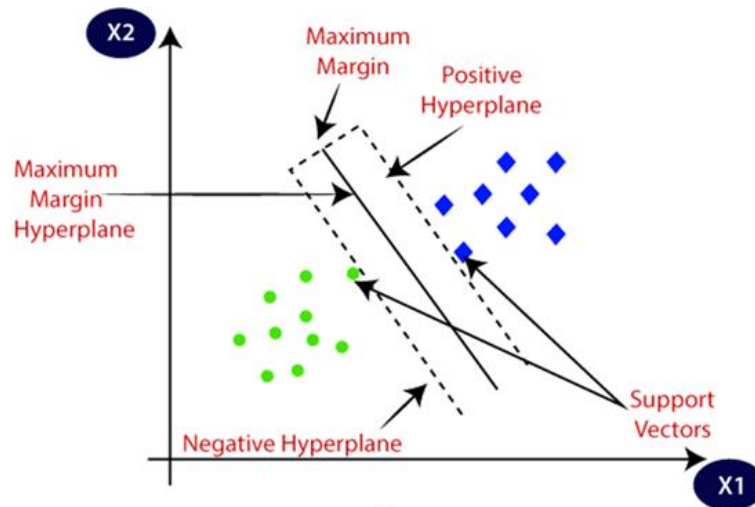
solve both linear and nonlinear issues. Support vector machines (SVMs) work by mapping data to a high dimension space, then finding the best hyperplane that separates the data into two categories (Rustam & Ariantari, 2018). Sometimes it may be found that the data is not linearly separable, and the machine must find the best separating line.

On linearly separable data, SVM searches for the closest points in order to find the separating line; the closest points to the separating line are called support vectors (Berwick, 2003). In Figure 2.1, if the blue square points represent lapses, and the green circle represents policies that have not lapsed, linearly separable data can easily be classified by drawing a straight line.



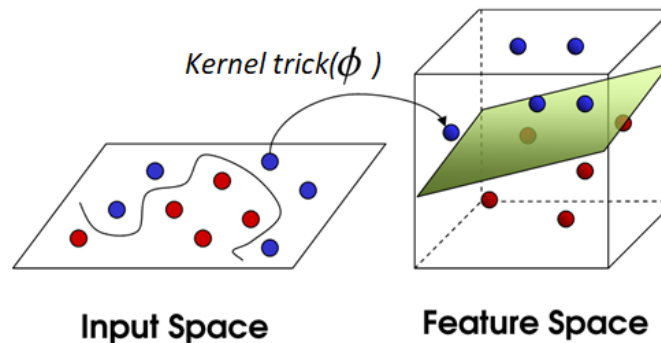
**Figure 2.1: Linearly Separable Data (Skilltohire, 2020)**

The boundary between the linearly separable support vector points is called a hyperplane and the distance between two categorised support vectors is called a margin. As shown in Figure 2.2, multiple straight lines (hyperplanes) can be fitted on the linearly separable data. The maximum margin hyperplane is a hyperplane that maximally separates two classes (blue and greens); this can be regarded as the best hyperplane (Girma, 2009; Awad & Khanna, 2015; Rustam & Ariantari, 2018).



**Figure 2.2: Optimal Hyperplane (Skilltohire, 2020)**

In the real world, data is usually randomly distributed and inseparable, meaning it cannot be separated by just fitting a straight line. In such cases, SVMs introduce kernel tricks to tackle the issue of linearly inseparable data. As illustrated in Figure 2.3, kernel tricks transform the original inseparable data (Figure 2.3 left) into a linearly separable one (Figure 2.3 right) by projecting it into a higher dimensional feature space then applying a linear classifier in that space.



**Figure 2.3: Linearly Inseparable Data; Kernel Trick (Wilimitis, 2018)**

The kernel function maps the transformed data by measuring the similarity of two vectors in any dimension space (Genton & Zhang, 2004). Kernel function takes the dot product of transformed vectors. In a case when the dot product is smaller, it can be concluded that there is no similarity in vectors. The main aim of kernels is to place the data in the

feature space, then apply linear algorithms in that feature space to identify the patterns (Genton & Zhang, 2004). A good kernel should enlarge the separation between the two classes (Williams, Li, Feng & Wu, 2005). Kernel functions can be represented by equation 2.8 (Savas & Dervis, 2019).

$$K(x, y) = \langle \phi(x), \phi(y) \rangle, \quad (2.8)$$

where  $x$  and  $y$  are input vectors,  $\phi$  is a transformation function, and  $\langle, \rangle$  represents the dot product function. The linear kernel is the most used kernel function, especially in text classification problems as most of the text classification cases are linearly separable. It gives the best performance when there are many explanatory variables. It is very basic and faster as compared to other kernel functions. It is represented by equation 2.9 (Brandusoiu & Todorean, 2013; Savas & Dervis, 2019).

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j, \quad (2.9)$$

The radial basis function kernel (RBF) is stationary and can be used when there is no prior knowledge of the data. It is represented by equation 2.10 (Brandusoiu & Todorean, 2013; Savas & Dervis, 2019).

$$K(x, y) = \exp(-\gamma \|x - y\|^2), \quad (2.10)$$

where  $x$  and  $y$  are input vectors,  $\|x - y\|^2$  is the squared distance, gamma measures the distance and the influence of two vectors/points on each other. The best gamma can be determined through cross-validations. Cases when two vectors are close together then

$\|x - y\|$  will be smaller and therefore  $\gamma \|x - y\|^2$  will be larger for  $\gamma > 0$ , meaning, the closer the vectors, the larger the RBF.

The polynomial kernel is a non-stationary kernel. It is well suited for a normalised or standardised training set. It can be represented by the below equation 2.11 (Brandusoiu & Todorean, 2013).

$$K(x, y) = (-\gamma x^T y + r)^d, \quad (2.11)$$

where  $d$  represents the degree of the polynomial. The flexibility of classifiers depends on the degree of polynomials. The lowest degrees of polynomials are similar to linear kernels, whereas higher ones allow flexibility in decision boundaries as compared to linear kernels (Savas & DAVIS, 2019).

The kernel must satisfy Mercer's theorem, that is, it must be positive semi-definite. Although sigmoid kernels are widely used, the sigmoid kernel matrix is however not positive semi-definite for some of the parameters (Lin & Lin, 2003). A sigmoid kernel is represented by equation 2.12 (Lin & Lin, 2003).

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + r), \quad (2.12)$$

where ( $\gamma$  and  $r$ ) are parameters. For  $\gamma > 0$ ,  $\gamma$  can be described as scaling parameters of the input samples and  $r$  represents the shifting parameter that controls the threshold. SVM may perform worse than random if the parameters ( $\gamma$ ,  $r$ ) are not chosen carefully. Sigmoid functions are usually outperformed by RBF (Lin & Lin, 2003).

Hossain and Miah (2016) evaluated six kernel functions: namely, RBF-Gaussian, polynomial, linear, sigmoid, laplacian and ANOVA RBF. The models were evaluated using Area Under Curve (AUC) and F1 scores. Linear models outperformed other kernel functions in terms of the AUC. The study by Nanda, Seminar, Nandika and Maddu (2018)



showed the superiority of the polynomial kernel over linear, RBF and sigmoid, whereas RBF outperformed linear and 3<sup>rd</sup>-degree polynomial kernels in the study by Yekkehkhany, Safari, Homayouni and Hasanlou (2014). These show that the performance of the kernel function is highly dependent on parameter setups, optimisations and tuning.

As with many ML algorithms, the robustness of SVMs depends on how well parameters are adjusted. In a linear kernel, there is only one important parameter to optimise which is C, in the RBF kernel and sigmoid kernel there are 2 parameters: C and gamma, while a polynomial kernel has 3 parameters: C, gamma and polynomial degree (Syarif, Prugel-Bennett & Wills, 2016).

Cost parameter (C) which is the regularisation parameter determines how much misclassification you would allow in the model. The performance of SVM is highly dependent on this parameter. Smaller C results in lower misclassification, that is, it behaves as a soft margin and larger C results in high misclassification, thus, it behaves as a hard margin. A hard margin hyperplane aims for a perfect classification, whereas, soft margin hyperplane will classify “most” of the data accurately while keeping the margin as wide as possible to avoid overfitting (Awad & Khanna, 2015). A small gamma value ( $\gamma$ ) reflects larger margins in the learned model, whereas large gamma reflects a small margin which may lead to overfitting (Suksut, Kaoungku, Kerdprasop & Kerdprasop, 2017).

Grid search is the commonly used method to optimise SVM parameters (Liu & Xu, 2013). The main idea behind the methodology is to find the optimal parameter based on the highest score criterion through an exhaustion search (Liu & Xu, 2013). Grid search creates a model for each combination of parameters. The models are evaluated through cross-validation. For grid search to work, you need to predefine the searching ranges (Liu & Xu, 2013). The methodology can however be time-consuming, especially, when classification accuracy is used as a measure of performance (Liu & Xu, 2013).

Some of the advantages of SVMs include their ability to work well in high dimensional spaces, their risk of overfitting is minimal, they are effective when the marginal separation

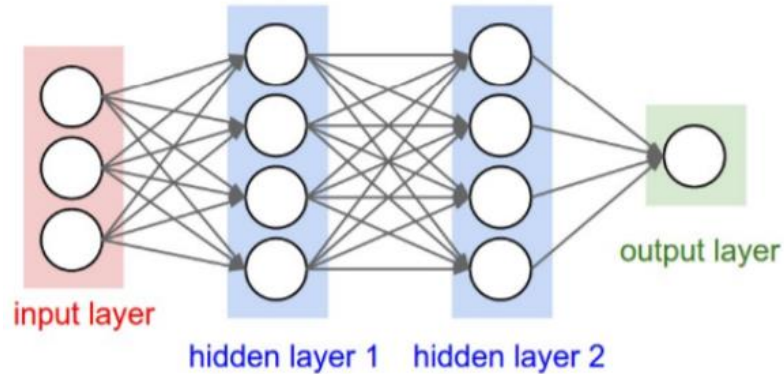
between classes is clear, and with the right kernel functions, they can solve most of the complex problems. Disadvantages include their difficulty in choosing a kernel function and they tend to underperform when features of a data point are more than the data sample (Statinfer, 2019).

### **2.1.3 Neural networks**

Neural Networks (NN) can be described as multi-layered, nonlinear regression models. Based on Nazari and Alidadi (2013), and Panchal and Panchal (2014), NNs are useful for tasks such as pattern recognition, classification, data mining, and medical diagnosis. They have been proven to be successful in sectors such as finance, medicine, engineering applications, geology, and physics (Pukała, 2016).

A NN is a series of neurons or nodes that passes information from one neuron to the other. It comprises neurons, weights, bias, and the activation function. Neurons are grouped in different layers; namely, the input layer, the hidden layer, and the output layer. An input neuron receives information from the outside world in the form of patterns then passes the information to the hidden neurons (Alsaadi & Maad, 2019). The hidden neurons map internal patterns and pass the information to the output. Neurons are connected through edges and every edge has a weight value associated with it (Alsaadi & Maad, 2019). The weight measures the strength and influence of the input on the output neurons. The weights range from negative to positive where zero reflects that there is no influence of the input neuron on the output neuron. The goal is to update these weights to decrease the loss error.

Figure 2.4 shows a high-level overview of a three-layered NN with 3 inputs, 8 hidden units, thus, 4 in the hidden layer 1 and 4 in the hidden layer 2, and an output layer (Hongsheng, 2021).



**Figure 2.4: High Level Overview of a 3 Layered Neural Network (Hongsheng, 2021)**

Neural networks (NNs) can be represented by the formula below:

$$Z = \sum_{i=1}^n a_i w_i + b , \tag{2.13}$$

where,

- $a_i$  represents the impute from 1 to  $n$
- $w$  represents the weight of parameters.
- $b$  represents the bias: This is a special extra input to the neurons; it is always 1. It can be used as a threshold to determine if the activation function should move forward or backwards.

The activation function calculates the weighted sum of inputs and add bias, then decides whether a neuron should be activated or not. It introduces nonlinearity into the neuron outputs by transforming the summation that is usually unbounded  $(-\infty, \infty)$  to a value that is between 0 and 1 or -1 and 1 depending on the selected activation function (Feng & Lu, 2019). Without the activation function, the output of a layer will basically be the linear function of the previous layer (Feng & Lu, 2019).

The commonly used activation functions are sigmoid function, Tangent hyperbolic function (Tanh) and Rectified linear activation function (ReLU). The sigmoid function in NNs is similar to the sigmoid function in LR. It is nonlinear, it transforms unbounded inputs into the range 0 and 1 and it resembles an s-shape (Feng & Lu, 2019). This activation function is mostly used in classification problems where the output is expected to be between the range of 0 and 1. The major drawbacks of the sigmoid function include vanishing gradient on gradient-based methods such as backpropagation. The issue means that the gradient weight on the network becomes exponentially smaller and approaches zero as they go through multiple layers, making it difficult for the model to update the weights (Hu, Zhang & Ge, 2021). The network will then learn slowly or refuse to learn further (Szandała, 2020).

Sigmoid functions are non-zero centred, meaning, they always produce nonnegative values. A big change in the input value may result in a very small impact on the output value as it must conform to a small range (Datta, 2020). The sigmoid function is given by equation 2.14 (Feng & Lu, 2019).

$$f(x_i) = \frac{e^{x_i}}{1 + e^{x_i}} = \frac{1}{1 + e^{-x_i}}, \quad (2.14)$$

Tanh transforms unbounded real number output into the range of -1 and 1. Larger values will be closer to 1 and smaller values will be closer to -1. It is similar to the sigmoid function except that it is zero centred. It however suffers the issue of vanishing gradients as well. It is more preferred than sigmoid because the convergence is faster as the outputs are zero centred.

Tanh function is given by equation 2.15 (Feng & Lu, 2019).

$$\tanh(x_i) = \frac{\sinh x_i}{\cosh x_i} = \frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}} = \frac{2}{1 + e^{-2x_i}} - 1 = 2\text{sigmoid}(2x_i) - 1, \quad (2.15)$$

ReLU is the widely used activation function since its invention (Feng & Lu, 2019; Hu *et al.*, 2021). It often outperforms both sigmoid and tanh. As indicated in equation 2.16 (Feng & Lu, 2019), it squashes the negative values into zero, however, positive values outputs are unbounded. Squashing all negative values to zero creates a special case of vanishing gradients as many neurons become inactive and output the value 0 (Hu *et al.*, 2021). It is computationally efficient and very quick to converge as opposed to tanh and sigmoid functions.

$$f(x_i) = \max(0, x_i) = \begin{cases} x_i, & x_i > 0 \\ 0, & x_i < 0 \end{cases}, \quad (2.16)$$

Commonly used NNs are Feedforward Neural Network (FNN) and Recurring Neural Network (RNN) algorithms. A FNN is a class of NN where information moves in one direction (forward) from the input through hidden layers to the output. There is no connection between all the nodes that are on the same layer (Du & Swamy, 2014).

An RNN is a class of NNs where information can move both forward and backward (recurrent) by introducing loops in the network. The connection provides the network with the visibility of both the initial information and current information, then builds an output based on the entire history (Rautio, 2019).

Various algorithms are used in training NNs (Vahedi, 2012). The most common one is backpropagation. Backpropagation is a NN training and optimisation algorithm that aims to minimise the cost function by adjusting parameters (weight and bias), thus minimising

the distance between predicted and true value (McDonald, 2017). At the initial stage, weights are chosen randomly, and backpropagation will then compute the weights' adjustments iteratively until the error is minimised (Stoyanova, 2017).

Traditional NNs suffered generalisation abilities (Gunn, 1998). SVM algorithm was introduced to solve generalisation issues in traditional NNs and to serve as alternative training method for conventional NNs. SVM is based on the Structural Risk Minimisation (SRM) principle which outperforms the traditional Empirical Risk Minimisation (ERM) principle employed by conventional NNs (Gunn, 1998). The main difference of the two principle is that SRM aims to minimize the upper bound of generalisation error, whereas the ERM minimises the error in the training set (Gunn, 1998).

Some of the advantages NNs includes their parallel computing capabilities, they can handle missing data, they do not assume a normal distribution, and they can tolerate faults – meaning that the corruption of cells will not prevent it from continuing with the process (Mahanta, 2017; Mijwil, 2018). It is, however, known to be a black box methodology because it is difficult to interpret how the results were calculated (Zhang, Beck, Winkler, Huang, Sibanda & Goya, 2018).

#### **2.1.4 Decision trees**

Decision Trees (DT) build classification or regression models in the form of a tree structure and are mostly used as a regression or classification tool. The tree is referred to as a classification tree when it is used for classification problems and as a regression tree when performing a regression task (Rokach & Maimon, 2014). The algorithm breaks down the input dataset into subsets. Each subset is defined by a specific set of rules and measures. The tree can grow until it reaches specific criteria or rules (Pohjalainen, 2016). A good split of the tree is regarded as the pure one – meaning that one class must be predominant. Impurity measures can be defined as measures of how well the classes are separated. Decision trees (DT) use entropy as impurity measures and information gain for the selection of features that would provide the best split of the data. Information gain

is the difference in entropy before and after the split. Given a collection  $S$  of  $c$  outcomes, entropy can be defined by equation 2.17 (Yang *et al.*, 2007).

$$\text{Entropy}(S) = \sum -p(I) \log_2 p(I), \quad (2.17)$$

where  $p(I)$  is the proportion of  $S$  that belongs to class  $I$ . Information gain for set  $S$  given attribute  $A$  can be represented by equation (2.8) (Yang *et al.*, 2007).

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum \frac{|S_v|}{|S|} \text{Entropy}(S_v), \quad (2.18)$$

where,  $S_v$  is the subset of  $S$  and attribute  $A$  has value  $V$ .

Accuracy can be affected if the tree is too complex. The complexity of the model will lead to overfitting (Breiman, Friedman, Olshen & Stone, 1984). Therefore, any additional splitting of the tree that does not make any difference to the impurity is not useful. The complexity of the tree can be measured by the number of nodes, leaves, and attributes it consists of (Rokach & Maimon, 2014).

The common tree stopping rules are (Patil, 2013):

1. When the tree reaches its maximum depth.
2. When all the training sets belong to the same class.
3. When the cases in the terminal are lesser than the minimum cases in the parent node.

#### **2.1.4.1 Decision trees pruning methods**

Employing strict stopping criteria can result in under fitted models, whereas loosening the stopping criteria may result in an overfitted tree (Patil, 2013). Pruning can be defined as

the process whereby the tree is being reduced by removing sub-branches that provide little power to the generalisation accuracy (Badr *et al.*, 2018). These processes help in reducing model over/underfitting. The criterion for pre-pruning is that when the error of the parent is lesser than the child then prune, else do not prune. Usually, the method is done in a bottom-up fashion.

Some commonly used pruning methodologies include the reduced error pruning method which is the method that seeks to replace the internal nodes with the most frequent class with the aim of improving the accuracy (Rokach & Maimon, 2014). It will continuously check and evaluate if replacing the nodes makes any difference to the accuracy. The process will continue until further pruning will decrease the accuracy of the model (Rokach & Maimon, 2014). Critical value pruning is the most used pruning method. The method sets a threshold then prunes all the nodes that do not reach that critical value (Mingers, 1989). If the critical value is set to be large enough, the resulting tree will be smaller (Mingers, 1989). The minimum error-based pruning method looks at the single tree that yields a minimum error rate for independent datasets (Cai, 2006). The method consists of the below steps (Cai, 2006):

1. Calculate the expected error rate for pruned subtree at each non-leaf node. The equation of calculating the expected error rate is as follows (Cai, 2006):

$$E_k = \frac{n - n_c + k - 1}{n + k}, \quad (2.19)$$

where  $k$  is the number of classes for observation,  $n$  assumes that the greatest number of observations  $n_c$  lie in class  $c$

2. Calculate the expected error rate if the node is not pruned taking into consideration the weight of a node.
3. Do not prune the node if the expected error rate is higher.



#### **2.1.4.2 Classification and regression trees**

Classification and regression trees (CART) work by constructing trees consisting of only two internal edges. The splitting is through towing criteria whereas the pruning is through the cost complex pruning method (Patil, 2013; Rokach & Maimon, 2014). The aim is to find splits that can minimise prediction squared errors (Rokach & Maimon, 2014).

#### **2.1.4.3 Iterative dichotomiser3**

Iterative Dichotomiser 3 (ID3) builds a DT in a top-down fashion based on specification properties. The method iterates through all unused attributes and calculates their entropy or information gain (Sakkaf, 2020). It then makes a node based on the attributes with the lowest entropy or highest information gain. Based on the values of the attributes, branches can be established. The methodology is recursive; the process will continue to create other nodes and branches until the tree classifies all the objects in the training set correctly. This technique was designed to deal with large induction tasks with training datasets containing many attributes (Troles, 2016). Based on a study by Quinlan (1986) with only a few iterations, the method is capable of finding a perfect DT with up to 30 000 objects and 50 attributes in the training set. The disadvantage of the model, however, is that the information gain can result in a multi-value bias when selecting attributes. Other drawbacks include the inability to handle missing data very well, the tree size may be difficult to control, and may require a lot of rules to be set (Wang *et al.*, 2017).

#### **2.1.4.4 C4.5**

C4.5 methodology uses gain ratio instead of information gain to overcome the multi-value bias that may result from the ID3 methodology (Wang *et al.*, 2017). The methodology is an extension of ID3 (Wang *et al.*, 2017). It aims at dealing with issues that ID3 cannot address such as:

- Overfitting – the methodology prunes the tree after it has been created by removing all the branches that do not have any impact on the overall accuracy and replacing them with leaf nodes.

- Handling missing values – the methodology allows missing values as imputes; the values will not be used to calculate both entropy and information gain.
- Handling continuous attributes – the methodology creates a threshold that will split the data where anything less than the threshold will be on the left node and everything above the threshold will be on the right node.

#### **2.1.4.5 Chi-squared automatic interaction detector**

Chi-squared automatic interaction detector (CHAID) assesses the predictor variable to find values that are least significantly different from the target attribute (Patil, 2013). CHAID methodology splits the target attribute into two or more categories; these categories are called the parent node. To split the parent nodes, the algorithm uses statistical tests. It performs some statistical tests to generate the P-value, F-Test, LRT, or Pearson Chi-squared test depending on the type of target attribute to measure the significant difference between inputs and the target attribute (Patil, 2013). The methodology will pair and test two values and if the p-values between the paired values are greater than a certain threshold, it merges the values and then searches for other pairs to be merged. The best splitting attribute will be selected in a way that each child node is composed of similar values. CHAID will create all possible cross-tabulations until there are no values that can be merged and no splitting can be performed (Patil, 2013; Rokach & Maimon, 2014). The methodology will stop when the below criteria are met (Patil, 2013; Rokach & Maimon, 2014):

- 1) The tree has reached its maximum depth.
- 2) The minimum number of cases as a parent has been reached.
- 3) When the minimum number of cases as a child has been reached.

Table 2.1 shows the possible splitting criteria and pruning strategy per DT mode (Singh, 2014).

**Table 2.1: Characteristics of DT Methods**

<u>Model</u>	<u>Splitting Criteria</u>	<u>Pruning Strategy</u>	<u>Missing values</u>
ID3	Information Gain	No pruning	Do not handle missing values
CART	Towing Criteria	Cost-Complexity pruning	Do not handle missing values
C4.5	Gain Ratio	Error Based pruning	Handle missing values
CHAID	P-value F-test Pearson Chi-squared	No Pruning	Handle missing values

**2.1.5 Ensemble models**

**2.1.5.1 Random forest**

Previous studies have shown that ensemble learning models are better than single classifiers (Dietterich, 2000; Wan & Yang, 2013; Lessmann, Baesens, Seow & Thomas, 2015). Ensemble models are algorithms that combine multiple ML algorithms (often called weak learners) into one predictive model that decreases the variance (bagging), bias (boosting), and improves the prediction accuracy (Kim, Min & Han, 2006). Bagging and boosting methodologies can be described as ensemble learning algorithms that aim to improve the accuracy and stability of ML algorithms such as DTs. The difference between the algorithms is that bagging uses bootstrap sampling (i.e., a randomly chosen sample with replacement) of the data to train a potential weak learner, whereas boosting uses the whole dataset to train each learner and gives greater weight to previous misclassified instances. Examples of bagging and boosting methodologies are RF and Adaboost.

Random Forest (RF) was introduced by Breiman (2001) and uses bootstrap sampling to build different unpruned independent DTs. The trees are created by randomly splitting the node of each tree, then searching for the best feature amongst the subset of features. Each tree may likely be inaccurate, but a combination of several trees will improve the accuracy of the tree. Each tree will then cast a vote for the most popular class. The accuracy of RF is measured by the strength of each tree (Breiman, 2001).

Some of the strength of RFs is their ability to handle large datasets with high dimensionality, they are robust to overfitting, they are not sensitive to outliers, features do

not necessarily need to be scaled, they can solve both classification and regression problems, they can handle missing data, and it is easy to set model parameters (Kho, 2018). They can, however, take time to run and the algorithm can be biased towards categorical values.

### ***2.1.5.2 Gradient boosting***

Gradient boosting (GB) is a classification model that ensemble weak models, in most cases DTs. The method aims to boost and optimise both the classification and regression models. Unlike in the RF where the tree models are built independently and results are combined at the end, with GB the weak learners are added iteratively in sequential order and weights for the next model are trained based on the results of the previous model with the aim of reducing the errors resulting from the previous model. A new model will gradually decrease the loss function of the whole ensemble model. This is done through a gradient descent procedure. To control the iterative process that GB follows, regulation parameters must be considered and if this is not done well it can cause overfitting (Bentéjac, Csörgő & Martínez-Muñoz, 2019).

The algorithms often outperform other ML models and there is no need to normalise and standardise features. However, they take time to run, boosting is sequential rather than parallel, and they are sensitive to outliers.

## **2.1.6 Applications**

### ***2.1.6.1 Underwriting***

Underwriting risk assessment is an important procedure done when accepting life insurance applicants. The process is mainly to assess the risk level of an applicant based on company guidelines and to determine the premium prices based on their risk (Biddle, Liu & Xu, 2018; Mashrur, Luo, Zaidi & Robles-Kelly, 2020). The traditional method has been to manually examine the applicant's health, behavioural and financial profile to determine the applicant's level of risk (Maier, Carlotto, Sanchez, Balogun & Merritt, 2019).

Maier *et al.* (2019) explored the use of ML algorithms in the underwriting space and reported greater operational efficiency and a significant decrease of 25% in the time taken to issue a policy. Hutagaol and Mauritsius (2020) found that the use of SVMs enhances the underwriting process, and reported that the use of ML speeds up client risk assessments. Biddle *et al.*'s (2018) study showed the use of ML algorithms in automating and optimising underwriting surveys and improving customer experience.

#### **2.1.6.2 Pricing optimisations**

Actuaries in the life insurance sector have always relied on data to calculate and optimise both the risk and personalised premium rates. It is anticipated that consumers compare prices throughout the market before they make decisions; this makes pricing optimisation a very important section in life insurance (Spedicato, Dutang & Petrini, 2018). Consumers usually look at a combination of product offerings, pricing, and adequate service that an insurer can provide (Abreu, 2019). According to Quotacy (2019), the main factors affecting prices are the type of insurance a client is buying, the applicants health status, and age. There are several studies such as those by Boodhun and Jayabalan (2018), Spedicato *et al.* (2018) and Henckaerts, Côté, Antonio and Verbelen (2020) that showed an improvement in pricing optimisation and risk analysis through the use of ML. Generalised Linear Models (GLMs) have commonly used algorithms in this area, however, studies such as the one performed by Henckaerts *et al.* (2020) showed good performance of the GB, regression trees, and RF over GLMs for price predictions.

#### **2.1.6.3 Customer lifetime value**

Customer lifetime value is an assessment of a customer's future profitability. It is usually used to identify high-value customers for marketing initiatives (Sifa, Runge, Bauckhage & Klapper, 2018). Fang, Jiang and Song (2016) compared RF, linear regression, DTs, SVM, and generalised boosted model for the prediction of customer lifetime profitability. The study indicated that inputs such as customers region, gender, age, and insurance status were the most important determinants of customer profitability.

#### **2.1.6.4 Cancellations**

As has been indicated, price, service, and product are what customers look for when they select an insurer of choice (Quotacy, 2019). Xong and Kang (2019) also reported that premium price is linked to the reasons why people leave an insurer. Price is what people consider when they buy life insurance, and it is also the reason that most leave an insurer. Apart from illustrating the use of ML algorithms in predicting lapses, Xong and Kang (2019) showed the importance of charging reasonable prices based on customers' lapse risk levels. Xong and Kang (2019) used data from a Malaysian insurance company with 800 entries to build and compare NNs, SVMs, LR, and K-nearest-neighbour algorithms for lapse prediction. The variables ranged from policy status variables (this is whether a policy is active or not), frequency of premium payments, policy term, age at policy entry, gender, sum assured, among others. The best model in training was SVMs whereas, in testing, the best model was NNs.

#### **2.1.7 Hybrid and ensemble models**

Recently, researchers seem to be building hybrid models intending to improve the accuracy and generalisation of the models (Miškovic, 2014; Hudaib, Harfoushi, Dannoun & Obiedat, 2015). A hybrid model is a combination of ML algorithms, soft computing, and optimisation methods (Ardabili, Mosavi & Várkonyi-Kóczy, 2020). It combines the strength of all the models for better performance (Miškovic, 2014; Ardabili *et al.*, 2020). In most cases, hybrid models seem to perform better than individual models (Miškovic, 2014; Hudaib *et al.*, 2015; Patil, 2018).

The study by Hudaib *et al.* (2015) illustrated the superiority of hybrid models over single built ML algorithms. They compared hybrid models built from K-means clustering and Multilayer Perceptron Artificial Neural Networks (MLP-ANN), Self-Organising Maps (SOM) and MLP-ANN, Hierarchical Clustering and MLP-ANN, with a normal MLP-ANN to predict churn rates in a Jordanian telecommunications company. The data contained 5,000 randomly selected customers with 11 attributes and 7.6% of the customers were churners. Models were built by initially clustering the data using K-means, SOM, and

hierarchical clustering. Two large datasets resulting from three clustering methodologies were combined as one input. The rest of the data (small clusters) was regarded as outliers and MLP-ANN was then developed using the resulting clustering datasets. Hybrid models created from clustering methodologies and MPL-ANN outperformed the performance of a normal MPL-ANN.

Another advancement in ML is the development of ensemble methods. As discussed under Section 2.1.4, ensemble methods combine multiple ML methods, often called weak learners. There are two classes of algorithms usually associated with ensemble learning, namely bagging, and boosting. The methods are both aimed at improving the stability of ML. They have been proven to be better predictors than single classifiers and to have a good out of sample performance (Dietterich, 2000; Kim *et al.*, 2006; Yang *et al.*, 2007; Lessmann *et al.*, 2015; Gavrishchaka, Yang, Miao & Senyukova, 2018).

Vafeiadis, Diamantaras, Sarigiannidis and Chatzisavvas (2015) illustrated the model improvement that results from boosting single classifier models. Five classification ML algorithms, namely LR, SVM, DT, backpropagation NN, and Naïve Bayes were built using telecommunication open-source data. Support vector machine (SVM) and backpropagation network outperformed other models; both models had an initial accuracy of 94% and F-measure of 77%. AdaBoosting was further applied to SVMs, backpropagation NNs, and DTs to further improve the accuracy of the models. Logistic Regression and Naïve Bayes could not be boosted as they lacked free parameters that could be tuned. Boosting improved the accuracy of the models by between 1 and 4% on all the three models and F-measure of 4.5 to 15%.

Another study by Loisel, Piette and Tsai (2019) compared a different boosting method (i.e. Extreme GB (XGBoost)) with single classifiers, namely SVM, LR, and regression tree (CART) for modelling lapse behaviours. Logistic Regression (LR) predicted 76% of the data correctly, CART predicted 77% correctly, SVM predicted 78% correctly, and XGBOOST 79% correctly. XGBoost outperformed LR, CART and SVM. Furthermore, the result showed that XGBoost was robust on the training sample.

Through modelling, Shao, Li and Liu (2007) proved that ensemble models are less susceptible to overfitting. They looked at three AdaBoost models namely Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. The models were compared to SVMs based on their ability to predict customer churn in the credit debt customer database of an anonymous commercial bank in China. The database had about 20,000 entries but only 1,524 entries with 27 predictor variables were selected for this experiment. Churners in this experiment were described as customers with low credit rates which is slightly different from the common definition of churners. Fifty percent of the observation was used for training the models and 50% for testing the models. From the experiment, it was observed that the AdaBoost is less susceptible to overfitting than most learning algorithms.

Researches have indicated that noisy data can lead to poor prediction accuracy and noise in the model can affect the computational time, as it will take time for the model to learn the data (Gupta & Gupta, 2019). Ensemble methods seem to be better at handling noise in the data than most ML algorithms (Gupta & Gupta, 2019).

### **2.1.8 Single classifiers comparisons**

From the literature it has been observed that ensemble methods often outperform single classifier ML methods, however, in a study conducted by Khan, Manoj, Singh and Blumenstock (2015) SVM outperformed RF. This shows that even though ensemble methods are generally good as opposed to single classifier methods, they may not always be the best options for modelling lapses. Even though there is no clear dominant single classifier ML algorithm, the results are usually not far off each other. This validates the point illustrated by Bolancé, Guillen and Padilla-Barreto (2016), that optimal prediction can be different based on the researcher's aim and what they want to achieve. It also depends on the type of data, data transformations, model parameter tuning, and model optimisations that take place when building a model (Vafeiadis *et al.*, 2015). For example, DTs outperformed NNs (Vahidy, 2012) when predicting churns, whereas in the study by Khan *et al.* (2010) and Goonetilleke and Caldera (2013) NNs outperformed DTs.



### 2.1.9 Model optimisations

Models such as RF ensemble and support vector models require critical parameter tuning for better results (Syarif *et al.*, 2016). Optimisation of parameters is very crucial in these models. The most commonly used methodology for optimising parameters of the models is a grid search, however, the method can be too slow on a larger dataset (Syarif *et al.*, 2016; Martínez, 2017). Rodan, Faris, Alsakran and Al-Kadi (2014) showed the power of optimising SVM parameters by using grid search with a customised evaluation metric. They predicted the churn rate in a Jordanian telecommunication company with a dataset of 5,000 observations and 11 variables. The model developed with optimised parameters was compared to a multilayer perceptron NN with backpropagation learning, K-nearest neighbour, Naïve Bayes, and C4.5 DT models. The optimised SVM outperformed other models by achieving an accuracy of 94.3%.

As shown in Section 2.1.2.3, choosing a kernel function is one of the difficult tasks in an SVM model (Syarif *et al.*, 2016). However, when a suitable kernel function is selected, the model can result in better prediction accuracy.

Siemes (2016) showed the strength of an SVM trained with a polynomial kernel when predicting churn rates of the largest indemnity insurance company in the Netherlands. Four predictive models, namely DT, NN, high-performance SVM, and LR were developed and compared based on the predictive power. As part of pre-processing, customers with missing information were excluded from the datasets. The data was scaled down to 867,598 policyholders, of which 11.35% were churners after data cleaning steps. Data were randomly eliminated from the dataset to prevent imbalances. Models were tested and validated using the following distribution of datasets: 50:50, 60:40, 70:30, 89:11 (non-churners and churners respectively). Support vector machine (SVM) with polynomial kernel outperformed other models for all training and validation distribution experiments.

## **2.1.10 Machine learning challenges**

### **2.1.10.1 Imbalanced data**

Imbalanced data can be described as a classification problem where observations in class distributions are not equal. The majority of ML algorithms require an equal representation of classes for them to perform well (Madasamy & Ramaswami, 2017). A slight skewness of the data can still give good predictions, however, a large gap can affect your prediction accuracy (Madasamy & Ramaswami, 2017). In the case of highly imbalanced data, the model often predicts the majority class effectively but overlooks the minority class (Krawczyk, 2016; Madasamy & Ramaswami, 2017). There are several solutions for dealing with imbalanced data at both the data and algorithm level (Kotsiantis, Kanellopoulos & Pintelas, 2006). Data level solutions focus on modifying the training datasets whereas algorithm solutions focus on modifying existing learners to reduce the bias towards the majority classes (Krawczyk, 2016).

It has been highlighted that studying the data complexity of imbalanced data is important and can influence the choice of resampling methodologies (Luengo, Fernández, García & Herrera, 2011; Santos, Soares, Abreu, Araujo & Santos, 2018). Examples of sampling methods include random under-sampling methodology which aims to eliminate the majority class randomly. It can however eliminate the useful data that is useful for the induction process (Kotsiantis *et al.*, 2006). Burez and van den Poel (2009) compared performances of random sampling, advanced under-sampling, GB, and weighted RF on an unbalanced dataset to predict churns. The result showed better performance of the under-sampling technique over other sampling methods.

Another example is random over-sampling which randomly replicates the minority class. The method makes the same copies of the minority class which can cause overfitting (Chawla, Bower, Hall & Kegelmeyer, 2002; Kotsiantis *et al.*, 2006). There are several methods used for over-sampling classification problems. The SMOTE (Synthetic Minority Over-sampling Technique) is the most commonly used one (Xie, Liang, Dong, Tan & Zhang, 2019). For each minority class, the algorithm calculated the K-nearest neighbour; the K-nearest neighbour are then selected to form synthetic examples.

Ensemble learning is the popular method for dealing with imbalanced data (Galar, Fernández, Barrenechea & Sola, 2012; Krawczyk, Woźniak & Schaefer, 2014; Błaszczycński & Stefanowski, 2015). Chen, Liaw and Breiman (1999) introduced two methods that are based on RF methodology to deal with the imbalanced data (i.e., weighted RF and balanced RF). Weighted RF gives more weight to the minority classes whereas balanced RF combines the downsampling of the majority class technique and ensemble learning idea. It alters the distribution of classes such that the classes are equal. Both methodologies improve the accuracy of predictions.

### 2.1.10.2 Overfitting

Overfitting is when a model performs well on training data and not so well on testing data; this is also known as poor generalisation (Ying, 2019). Overfitting can happen because of noise in the dataset, limited training sets, and complexity of classifiers (Ying, 2019), whereas underfitting can occur when the model is too simple and informed by a few features which make it difficult to learn the dataset. Overfitting can be prevented by performing cross-validation on the dataset – the algorithm uses the initial training set to generate multiple small sets then uses the splits to tune the model. As illustrated in Table 2.2, in a standard K-fold cross-validation, the technique splits the data into k subsets; the first subset will be used for validation and the rest (K-1) will be used for training. The process can be repeated K times with each of the sets being used once as a test set (known as holdout fold). This means that each set will have an opportunity to be used as a training set once and K-1 times as a testing set (Santos *et al.*, 2018; Waseem, 2020).

**Table 2.2: Cross-validation**

K=1	Train	Train	Train	Train	Validation
K=2	Train	Train	Train	Validation	Train
K=3	Train	Train	Validation	Train	Train
K=4	Train	Validation	Train	Train	Train
K=5	Validation	Train	Train	Train	Train

Training with more data can also help prevent overfitting – a small sample of data is more prone to overfitting than a large dataset (Santos *et al.*, 2018; Ying, 2019; Waseem, 2020). The accuracy of the model can stop increasing after a certain point. If the model continues to learn after that point has been reached, validation errors will decrease whereas the training errors increase which causes overfitting. The model can be stopped before reaching the point of overfitting (Ying, 2019; Waseem, 2020).

The use of ensemble models can also help in overfitting prevention as they are unlikely to overfit. They combine several classifiers to improve prediction accuracy. The higher the number of ensemble models chosen, the higher the probability to overfit (Brown & Schmidt, 2009). The study conducted by Brown and Schmidt (2009) also showed that overfitting in ensemble techniques can happen, mostly when the data is small as opposed to large datasets. Pruning capabilities in ensemble models reduce the risk of overfitting. Some researchers have embedded cross-validation in their ensemble methods to overcome overfitting issues and improve their models' performances (Brown & Schmidt, 2009).

### **2.1.10.3 Missing data**

Training datasets with huge proportions of missing data can affect the model's accuracy (Badr, 2019) as some ML methods cannot handle missing data very well (i.e., basic Iterative Dichotomiser3) (Moulana & Hussain, 2014). There are two common ways of dealing with missing values, that is to eliminate variables with missing data or impute the values (Vieira, Proença & Salgado, 2016). Imputation is the process of filling missing values with estimated or observed values, whereas deletion refers to deleting entries or variables with missing data.

One commonly used deletion methodology is listwise deletion which deletes all entries where there are missing values (Norazian, 2013). This is suitable for cases where the data is missing completely at random (MCAR) and the data is large enough (Roy, 2019). When the data is small, you run a risk of losing valuable data thus impacting the statistical power and introducing biasness. Missing completely at random (MCAR) indicates that

there is no relationship between missing data and any observed values (Norazian, 2013). Mean or mode imputation methods can also be used when the data are MCAR. The method calculates the mean or mode of all non-missing values in a variable, then assigns that value to the missing values (Gelman, 2010).

Missing at random (MAR) is when the missing data has a relationship with observed variables and not necessarily missing observations (Norazian, 2013). The possible imputation method for MAR is the multiple imputation method (Song & Shepperd, 2007). The multiple imputation method creates multiple predictions per missing value (Norazian, 2013). The data is first replicated many times, then different numbers will randomly be predicted and imputed from all sets. All datasets will then be combined. If the character of the missing data does not meet characteristics of both MCAR and missing not at random (MNAR), then the data is MNAR; thus, the missing data is dependent on missing observations.

Table 2.3 summarises some of the applications of ML and their performances.

**Table 2.3: Literature Summary**

Year	Author	Title	Models	Industry	Model Performance
2019	Xong & Kang	A Comparison of Classification Models for Life Insurance Lapse Risk.	Neural Network, Support Vector Machine, Logistic Regression and K- Nearest-Neighbor	Insurance	The best model in training was Support Vector Machines whereas in testing, the best model was Neural Networks
2018	Sabbeh	Machine-Learning Techniques for Customer Retention: A Comparative Study	Logistic Regression, Decision Tree, Support Vector Machine, K-Nearest Neighbour, Ada Boost, Stochastic Gradient Boosting, Naive Bayes, Random Forest, Multi-Layer Perceptron which is an Artificial Neural Network and Linear Discriminant Analysis	Telecommunication	ADA boost and Random Forest outperformed other models
2017	Aleandri	Modeling Dynamic Policyholder Behavior through Machine Learning Techniques	Logistic Regression with Bagging Classification Trees	Insurance	Bagging Classification Trees outperformed Logistic Regression.
2016	Bolancé et al	Predicting defection in non-life motor and home insurance.	Logistic Regression, Conditional Tree and Support Vector Machine	Insurance	Support Vector Machine slightly outperformed Logistic Regression and Conditional trees.
2015	Hassouna et al.	Customer Churn in Mobile Markets: A Comparison of Techniques.	Logistic Regression and Decision Trees	Mobile Industry	Decision Tree outperformed Logistic Regression.
2014	Rodan et al.	A Support Vector Machine Approach for Churn Prediction in Telecom Industry	Support Vector Machine, Multilayer Perceptron Neural Network with backpropagation learning algorithm, K-Nearest Neighbour, Naive Bayes and C4.5 Decision Trees models	Telecommunication	Support Vector Machine outperformed other models
2013	Goonetilleke and Caldera	Mining Life Insurance Data for Customer Attrition Analysis	Neural Network and Decision Trees	Insurance	Decision Trees
2012	Shaaban et al.	A Proposed Churn Prediction Model.	Neural Networks, Support Vector Machines and Decision Trees	Mobile service provider.	Support Vector Machine outperformed Neural Network and Decision Trees.
2011	Eling and Kiesenbauer	What Policy Features Determine Life Insurance Lapse? An Analysis of the German Market.	Generalised Linear Model	Insurance	Generalised Linear Model
2010	Khan et al.	Applying Data Mining to Customer Churn Prediction in an Internet Service Provider.	Decision Trees, Logistic Regressions and Neural Network	Internet Service provider	Neural Network outperformed Logistic Regression and Decision Trees
2009	Tsai and Lu	Customer churn prediction by hybrid neural networks	Hybrid models combined from (Artificial Neural Network + Artificial Neural Network) and (Self Organizing Maps (SOM) + Artificial Neural Network) and a baseline ANN	Telecommunication	(ANN + ANN) hybrid model outperformed (SOM + ANN) hybrid model.
2008	Coussement and Van den Poel	Churn prediction in subscription services: An application of Support Vector Machines while comparing two parameter-selection techniques.	Support Vector Machine, Random Forest and Logistic Regressions	Subscription Services	Support Vector Machine outperformed Logistic Regressions, however it was surpassed by Random Forests.

### 2.1.11 Critical review

Modelling lapses can be a difficult task as they can be influenced by many parameters ranging from macro and microeconomic factors, products, client behaviour, among others. From the recent research reviewed there is evidence that ML classification algorithms such as NNs, SVMs, and DTs have been widely used and compared in industries such as telecommunication, banking, and some areas of insurance (Shao *et al.*, 2007; Tsai & Lu, 2009; Hudaib *et al.*, 2015; Vafeiadis *et al.*, 2015; Geschiere, 2017; Sabbeh, 2018). The algorithms have proven to have predictive power as compared to traditional statistical methods in these industries. Although there are no clear dominant ML algorithms, ensemble methods seemed to be consistently outperforming many single ML classifiers. (Dietterich, 2000; Kim *et al.*, 2006; Yang *et al.*, 2007; Lessmann *et al.*, 2015; Gavrishchaka *et al.*, 2018; Sabbeh, 2018; Loisel *et al.*, 2019).

Confusion matrix, AUC, and ROC curve have been the most used methods to evaluate the model's predictive accuracy (Bolancé *et al.*, 2016; Siemes, 2016; Xong & Kang, 2019). Machine learning (ML) has proved to be able to handle big data very well, can easily find patterns within the data, can easily learn the data, and can be easily automated; however, the models can take time to run.

It has been observed that life insurers generally prefer GLMs for lapse modelling prediction and the most used GLM is LR (Ducuroir *et al.*, 2016; Hendrych, 2019). Logistic Regression (LR) is consistently used as a benchmark comparison to other ML algorithms (i.e., NNs, SVM, DTs) and it is consistently outperformed by those models (Vafeiadis *et al.*, 2015; Aleandri, 2017; Sabbeh, 2018).

There is no clear indication of what the most powerful ML algorithm is. Vafeiadis *et al.* (2015) and Bolancé *et al.* (2016) point out that optimal prediction may differ depending on datasets and how models are optimised. From the literature in Table 2.3, it can be observed that different models react differently on different datasets. Xong and Kang (2019) compared multiple ML algorithms on lapse predictions. Conclusions about model superiority were based on one insurer dataset. Similarly, Rodan *et al.* (2014), Tsai and

Lu (2009) and Shaaban, Helmy and Khedr (2012) compared multiple models and made conclusions based on one dataset. Aleandri (2017) and Goonetilleke and Caldera (2013) modelled dynamic behaviour and customer attrition in life insurance respectively, however, they compared less than two models over a single dataset. Based on this research gap identified, that is, comparing multiple ML algorithms over multiple datasets, this dissertation critically evaluated nine ML algorithms on two different datasets with different distributions. Parameters were tuned and optimised the same way on both datasets. Models were evaluated based on their ability to classify lapses correctly, and their ability to generalise well using different performance measures. The dissertation also tested if there was a clear dominating high performing model based on two datasets.



# CHAPTER THREE

## METHODOLOGY AND RESULTS

### 3.1 Introduction

This chapter illustrates the process of building models and presents their performances. It is divided into three parts, namely, the model's setup, results, and the discussion of results. The model's setup is divided into data pre-processing, feature selection method, training and validation processes, whereas the results sections interpret and discuss the model's findings on the training and validation stages.

### 3.2 Model's Setup

#### 3.2.1 Data

##### 3.2.1.1 Dataset 1 – Insurer 1

Historical lapse data from an anonymous insurer was used in this study as inputs for the models. The data was obtained from Kaggle open-source repository (Moon,2019). The datasets initially consisted of six categorical and 26 numerical variables with a total of 51,865 unique policyholders who are principal members. Inception dates for these policies were between January 2017 to August 2020.

Four files were downloaded, namely historical payments, client's data, policy data, and lapse information. Historical payments had five columns, namely, Policy ID which is the policy identifier, the amount paid which reflects premium payments, date paid which is the premium payment date, postdate and the premium due date. The client's data contained the policy identifier, gender, birthdate, title, and addresses. Policy data had policy level information such as policy effective date, products, premium, sum assured, policy effective date, agent code, branch signed up code, etc. The final file contained the policy identification and lapse information.

The payment history file consisted of historical payments from 2017 to 2018. The 2019 historical payment data was not provided, meaning that policyholders with effective dates from 2019 did not have payment history information. Tenure was calculated for all the policies without lapse status. The calculation was based on the date difference between the effective date and the last recorded effective date which was January 2020. For policies that had lapsed, tenure was calculated based on the lapsed date and effective date. All four datasets were then merged into one file.

Historical payments datasets were longitudinal whereas the rest of the datasets were static per policyholder. The data was aggregated on a policy level in preparation for data analytics and modelling. Only data from 2017 to 2019 was considered for both training and validation of the models; 2020 data did not have both premium and lapse status information. Extra variables such as tenure, number of received payments, number of missed payments, and count of effective date were calculated from existing variables. The following were the data assumptions:

1. Policies with multiple effective dates: It was assumed that these policies were cancelled before and are now reinstated.
2. Count of effective date: This will indicate the number of cancellations per policy.
3. The last policy effective date recorded was on the 8th of January 2020. For all policies without lapse status, tenure was calculated based on the date difference between their effective date and the last recorded effective date which is the 8th of January 2020.

Table 3.1 shows the variable statistics ordered by the amount of missing data (highlighted in green). Nmiss is the number of missing values, N is the number of non-missing values. For numeric variables, minimum, maximum, mean, and standard deviation of the variables were calculated. The variable statistics summary already highlights some of the data challenges like a high number of missing values on some variables (highlighted in green) and outliers on the minimum birthdate (circled in red). This has been dealt with in Section 3.2.2. The company's average customer was born in 1983, the number of

premium payments per customer ranges from 0 to 48. The average sum assured is 135683. 1900 in the lapse year variable is a default value that reflects active policies.

**Table 3.1: Insurer 1 – Variable Statistics**

Variable	N Miss	N	Missing(%)	Minimum	Maximum	Mean	Std Dev
NAD_ADDRESS1	21515	28542	43%				
NAD_ADDRESS2	20699	29358	41%				
NPH_TITLE	20176	29881	40%				
NPH_BIRTHDATE	20175	29882	40%	1899	2015	1983	
NPH_SEX	20175	29882	40%				
TOTAL_PAID_AMNT	17972	32085	36%	0	661178	28108	30419
CNT_NON_PAYMENTS	17972	32085	36%	0	47	7	6
CNT_PAYMENTS	17972	32085	36%	0	48	8	7
NON_PAYMENT_RATIO	17972	32085	36%	0	1	0	0
TENURE	364	49693	1%	0	43	21	11
LAST_EFFECTIVE_DATE	0	50057	0%				
CNT_NP2_EFFECTDATE	0	50057	0%	1	6	1	0
CNT_PPR_PROD CD	0	50057	0%	1	3	1	1
AVG_NPR_PREMIUM	0	50057	0%	145	261973	2168	4003
SUM_NPR_PREMIUM	0	50057	0%	244	785919	5831	9949
CNT_NPH_LASTNAME	0	50057	0%	1	8	2	1
CNT_CLF_LIFEC D	0	50057	0%	1	6	2	1
AVG_NPR_SUMASSURED	0	50057	0%	0	32815387	135683	194861
CNT_NLO_TYPE	0	50057	0%	1	4	2	0
SUM_NLO_AMOUNT	0	50057	0%	142	325901	1692	2616
LAPSE	0	50057	0%	0	1	0	0
LAPSE_YEAR	0	50057	0%	1900	2019		
AAG_AGCODE	0	50057	0%				
PCL_LOCATCODE	0	50057	0%				
CATEGORY	0	50057	0%				

### 3.2.1.2 Dataset 2: Insurer 2

The second dataset was also extracted from Kaggle (2019). The main purpose of the data was to predict future premiums and lapse rates. The data consisted of 668,027 policyholders which were inceptioned from 2011 November to 2019 August, thus eight years of data. The data consisted of 20 variables. The variables ranged from channels, policy types, payment mode, policy status, benefits, premium information, and policy inception details. On the policy status, there are five levels, namely lapse, surrender, in force, expired, and death. A target variable was formed from the policy status variable, one as a lapse and the rest of the categories were zero. The main aim of looking at the second

dataset is to test how the models will react on a completely different dataset with a different distribution. Table 3.2 indicates the variable statistics of the data. Similar to Table 3.1, the statistics have been ordered by the percentage of missing data. Nmiss is the number of missing values, N is the number of non-missing values. For numeric variables, minimum, maximum, mean, and standard deviation of the variables were observed. Policy issue date ranges from 2011 November to 2019 August. Maximum policy entry age is 70 years. The average premium rate is 83.

**Table 3.2: Insurer 2 – Variable Statistics**

Variable	N Miss	N	Missing(%)	Minimum	Maximum	Mean	Std Dev
Issue Date	123967	61593	67%	2011-11	2019-08		
NON LAPSE GUARANTEED	121274	64286	65%				
SUBSTANDARD RISK	121274	64286	65%	0	200	0	4
NUMBER OF ADVANCE PREMIUM	121274	64286	65%	0	5	0	0
INITIAL BENEFIT	121274	64286	65%	0	0	0	0
Full Benefit?	121274	64286	65%	1	1	1	0
Policy Year (Decimal)	121274	64286	65%	0	7.75	4	2
Policy Year	121274	64286	65%	1	8	4	2
Premium	121274	64286	65%	0	994	83	173
POLICY_NUMBER	0	185560	0%				
CHANNEL1	0	185560	0%	1	8	4	2
CHANNEL2	0	185560	0%	1	3	3	1
CHANNEL3	0	185560	0%	0	82	8	14
ENTRY AGE	0	185560	0%	0	70	32	13
SEX	0	185560	0%				
POLICY TYPE 1	0	185560	0%	1	20	5	4
POLICY TYPE 2	0	185560	0%	1	88	22	19
POLICY TYPE 3	0	185560	0%	1	5	2	1
PAYMENT MODE	0	185560	0%				
POLICY STATUS	0	185560	0%	0	1	1	0
BENEFIT	0	185560	0%	0	980	47	87

### 3.2.2 Data pre-processing

Data pre-processing is one of the important processes in ML that transforms the raw data into a desirable input. In this study, the researcher did some basic descriptive statistics to understand the data layout. A data audit was performed to check for data inconsistencies such as outliers, errors, incorrect date formats, and data duplication to ensure that the

data was clean and ready for modelling. SAS Enterprise Guide and SAS Enterprise Miner were used to clean and create models.

### ***3.2.2.1 Missing data imputation***

In Insurer 1's dataset, all demographic and payment information had missing values. From the missing value pattern shown in Table 3.3, values were missing at random, thus missing data had a relationship with observed variables. The percentage of missing demographics (40%) was concerning. These variables were removed.

Literature shows that a high number of missing values may affect statistical inference. It is not clear what percentage of missing values is acceptable. Vieira et al. (2016) said that rejecting a huge number of missing values (>50%) is not risk-free as it may lead to a loss of predictive power. Schafer (1999) says a missing rate of 5% or less is inconsequential. Bennet (2001) says 10% of missingness might lead to statistical bias. Raymond and Roberts (1987) recommended that a variable with more than 40% of missing values should be deleted. Madley-Dowd, Hughes, Tilling and Heron (2019) showed that imputation methods such as multiple imputations reduce bias even though the percentage of missing variables is large enough. Based on this literature, payment history with 39% missing values were imputed as they have been proven from literature to be a significant predictor of lapses (Eling & Kiesenbauer, 2011).

**Table 3.3: Insurer 1 – Missing Values Pattern**

Group	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Freq	11271	25	17025	210	1	9	1	615	203	8	377	129	8	19744	43	318	70
Percent	22.52	0.1	34.01	0.4	0	0	0	1.2	0.4	0	0.8	0.3	0	39.44	0.1	0.6	0.1
NAD_ADDRESS1	X	X	X	X	X	X	X	.	.	.	.	.	.	.	.	.	.
NAD_ADDRESS2	X	X	X	X	X	.	.	X	X	X	.	.	.	.	.	.	.
NPH_TITLE	X	X	X	X	.	X	X	X	X	X	X	X	X	.	.	.	.
NPH_BIRTHDATE	X	X	X	X	X	X	X	X	X	X	X	X	X	.	.	.	.
NPH_SEX	X	X	X	X	X	X	X	X	X	X	X	X	X	.	.	.	.
TOTAL_PAID_AMNT	X	X	.	.	X	X	.	X	.	.	X	.	.	X	X	.	.
CNT_NON_PAYMENTS	X	X	.	.	X	X	.	X	.	.	X	.	.	X	X	.	.
CNT_PAYMENTS	X	X	.	.	X	X	.	X	.	.	X	.	.	X	X	.	.
NON_PAYMENT_RATIO	X	X	.	.	X	X	.	X	.	.	X	.	.	X	X	.	.
LAST_EFFECTIVE_DATE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CNT_NP2_EFFECTDATE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CNT_PPR_PRODCD	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
AVG_NPR_PREMIUM	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SUM_NPR_PREMIUM	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CNT_NPH_LASTNAME	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CNT_CLF_LIFECD	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
AVG_NPR_SUMASSURED	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CNT_NLO_TYPE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SUM_NLO_AMOUNT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LAPSE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LAPSE_YEAR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
AAG_AGCODE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PCL_LOCATCODE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CATEGORY	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
TENURE	X	.	X	.	X	X	X	X	X	.	X	X	.	X	.	X	.

As illustrated in Section 2.1.10.3, the literature shows that data imputation method decisions should be based on the patterns of the missing values. The multiple imputation method is a commonly used algorithm for data that is missing at random. Studies by Soares, Santos, Abreu, Araujo and Santos (2018) and Chambers (2000) showed that both predictive accuracy and feature distribution accuracy are important in the imputation selection method. Soares *et al.*, (2018) also mentioned that imputation techniques must preserve the distribution of the original sets.

In this thesis, the distribution based random imputation methodology available on SAS E-Miner was used to impute the missing values. The method replaced the missing data based on the random percentile of the probability distribution of non-missing values. The methodology was selected since it does not change the data distribution that much, it did not change the range of the original data and it resulted in good accuracy. That is; it satisfies imputation criteria illustrated by Chambers (2000), that the procedure must

maximise the preservation of the original values (PAC- prediction accuracy) and it must maintain the distribution of original values (DAC-Distribution Accuracy). Table 3.4 shows that distribution seemed similar even after imputations.

Skewness and kurtosis measure the shape of the distribution. Kurtosis measures the tail relative to the normal distribution. From Table 3.4, it can be observed that all payment variables (premiums, sum assured, paid amounts, and NLO amount) have high kurtosis, meaning that they are heavy-tailed, or they have outliers. The mean of 1919 on policy lapse year is the results of a default value (i.e., 1900) on all active policies.

**Table 3.4: Variable Stats Before and After Imputation**

Before Imputation									
Variable	Mean	Deviation	Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
AVG_NPR_PREMIUM	2168	4003	50057	0	145	1480	261973	23	970
AVG_NPR_SUMASSURED	135683	194861	50057	0	0	104987	32815387	99	15997
CNT_CLF_LIFECD	2	1	50057	0	1	1	6	1	-1
CNT_NLO_TYPE	2	0	50057	0	1	2	4	-1	0
CNT_NP2_EFFECTDATE	1	0	50057	0	1	1	6	5	27
CNT_NPH_LASTNAME	2	1	50057	0	1	1	8	2	3
CNT_PPR_PRODCD	1	1	50057	0	1	1	3	1	-1
LAPSE	0	0	50057	0	0	0	1	2	1
LAPSE_YEAR	1919	43	50057	0	1900	1900	2019	2	1
SUM_NLO_AMOUNT	1692	2616	50057	0	142	1339	325901	47	4910
SUM_NPR_PREMIUM	5831	9949	50057	0	244	4707	785919	30	1616
CNT_NON_PAYMENTS	7	6	32085	17972	0	4	47	1	0
CNT_PAYMENTS	8	7	32085	17972	0	6	48	1	0
NON_PAYMENT_RATIO	0	0	32085	17972	0	0	1	-1	2
NPH_BIRTHDATE	1983	23	29882	20175	1899	1991	2015	-1	-1
TENURE	21	11	49693	364	0	19	43	0	-1
TOTAL_PAID_AMNT	28108	30419	32085	17972	0	18224	661178	3	20
After Imputation									
Variable	Mean	Deviation	Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
AVG_NPR_PREMIUM	2168	4003	50057	0	145	1480	261973	23	970
AVG_NPR_SUMASSURED	135683	194861	50057	0	0	104987	32815387	99	15997
CNT_CLF_LIFECD	2	1	50057	0	1	1	6	1	-1
CNT_NLO_TYPE	2	0	50057	0	1	2	4	-1	0
CNT_NP2_EFFECTDATE	1	0	50057	0	1	1	6	5	27
CNT_NPH_LASTNAME	2	1	50057	0	1	1	8	2	3
CNT_PPR_PRODCD	1	1	50057	0	1	1	3	1	-1
LAPSE	0	0	50057	0	0	0	1	2	1
LAPSE_YEAR	1919	43	50057	0	1900	1900	2019	2	1
SUM_NLO_AMOUNT	1692	2616	50057	0	142	1339	325901	47	4910
SUM_NPR_PREMIUM	5831	9949	50057	0	244	4707	785919	30	1616
IMP_CNT_NON_PAYMENTS	7	7	50057	0	0	4	47	1	3
IMP_CNT_PAYMENTS	9	7	50057	0	0	6	48	1	2
IMP_NON_PAYMENT_RATIO	0	0	50057	0	0	0	1	-1	2
IMP_NPH_BIRTHDATE	1984	24	50057	0	1899	1991	2015	-1	0
IMP_TENURE	21	11	50057	0	0	19	43	0	-1
IMP_TOTAL_PAID_AMNT	32549	49246	50057	0	0	18224	661178	7	86

For Insurer 2's dataset, all the entries which were missing both issue date and policy year were deleted from the dataset. Furthermore, observations with policy status categories such as surrender policy, expired, and death were removed. Policy status only composed of in force and lapse data, thus 30% (55,592) of the original dataset.

### 3.2.2.2 Replacing outliers

All outliers within the extreme upper (> 95% percentile) and lower tail (<5% percentile) of the distribution were replaced with 95% and 5% percentile, respectively. If the value was too high, it was still replaced with a high number (95% percentile).

### 3.2.2.3 Categorical variables encoding

Some of the ML libraries do not take categorical variables as input. In this study, all the categorical variables were converted to numerical variables using the one hot encoding methodology. The methodology works best with nominal categories, which are categories that cannot be easily ordered. As illustrated by Table 3.5, It creates dummy variables as extra features based on the distinct categories in a feature. The categorical variables need to be converted to integers first, then create binary features (i.e., 0, 1) from the integers. Integers are replaced based on their alphabetical order. Integers are assigned ranges from 0 to n-1, where n is the number of distinct classes.

**Table 3.5: Dummy Variables Imputation Example**

CATEGORY	<u>INTERGERS</u>	<u>CATEGORY</u> <u>1750CEH</u>	<u>CATEGORY</u> <u>8DALFYO</u>	<u>CATEGORY</u> <u>GWW4FYB</u>	<u>CATEGORY</u> <u>LXSLG6M</u>	<u>CATEGORY</u> <u>M1ZXYVG</u>	<u>CATEGORY</u> <u>R821UZV</u>
CATEGORY_1750CEH	0	1	0	0	0	0	0
CATEGORY_8DALFYO	1	0	1	0	0	0	0
CATEGORY_GWW4FYB	2	0	0	1	0	0	0
CATEGORY_LXSLG6M	3	0	0	0	1	0	0
CATEGORY_M1ZXYVG	4	0	0	0	0	1	0
CATEGORY_R821UZV	5	0	0	0	0	0	1
CATEGORY_8DALFYO	1	1	0	0	0	0	0

The issue with one hot encoding methodology is its ability to cause multicollinearity. Some of the dummy features might have to be removed. This was dealt with in the later stage of the thesis. For categories with more than 10 distinct variables, label encoding was used



to replace the categories, which means categories were only replaced with integers; extra dummy variables were not created as this could have resulted in computational issues because of the memory consumption.

#### **3.2.2.4 Feature scaling**

Feature scaling can be described as a technique that scales variables to the same range. This is an important process in data pre-processing because some high values within the feature may tend to dominate other features when fitting the model. The commonly used techniques for feature scaling are data normalisation and data standardisation. Standardisation transforms variable values in a way that it will have a mean of zero and a variance of one, whereas normalisation transforms the data to take up values between zero and one. All variables were standardised using equation 3.1 (Peshawa, Muhammad & Rezhna, 2014)

$$Z = \frac{(x_i - u)}{s}, \tag{3.1}$$

Normalisation is represented by the below equation (Peshawa et al.,2014).

$$Z = \left( \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \right), \tag{3.2}$$

where Z is new normalised/standardised value,  $x_i$  is the data point ( $x_1, x_2 \dots x_n$ ), u is the sample mean, s is the sample standard deviation,  $x_{\min}$  is the sample minimum and  $x_{\max}$  is the sample maximum.

Both normalisation and standardisation are sensitive to outliers. They also possess challenges on the testing(unseen) data if the values of the testing data fall outside the

range on trained data (Cao *et al.*, 2016). However, literature shows that scaled models perform better than unscaled models (Cao *et al.*, 2016)

### **3.2.2.5 Imbalanced data**

The dataset of Insurer 1 had an issue of imbalanced data. Lapses were rare as many policyholders were staying on the books. There were 15% lapses and 85% non-lapses. Over-sampling minority cases to balance the data was considered and SMOTE methodology was used to oversample the minority class and balance the data. SMOTE synthesised new examples by looking at the minority classes at random then finding their K-nearest neighbourhood.

Elreedy and Atiya (2019) mentioned that SMOTE performs well when the number of K is smaller. One neighbourhood (i.e., K=1) would produce best results, however, it will produce synthesised values that are highly correlated to the original values, resulting in a lesser impact on the classification model (Elreedy & Atiya, 2019). In this study, K was initially set to 11. SMOTE was further modified by adding a distance threshold. All neighbourhoods above the threshold were eliminated. This setting resulted in varied values of nearest neighbourhood (K) for each point. A study by Pradipta, Wardoyo, Musdholifah and Sanjaya (2021) showed superiority of radius based SMOTE over the classic SMOTE. McInroy (2016) achieved good performance on a modified SMOTE over normal SMOTE by adding a distance threshold when calculating number of neighbours.

SMOTE process was coded as below.

1. Output minority cases, thus 15,9% of the total data.
2. Run a PROC MODECLUS procedure on SAS Enterprise guide to create 11 nearest neighbours around each standardised observation where target=1 (lapse). That is k=11. The process outputs the density estimates, nearest neighbourhoods and the distance between the observation and the nearest neighbour.
3. Set distance threshold then eliminate all neighbours that are above the threshold.
4. Synthesise new random cases between the original samples and the nearest neighbour in step 3.

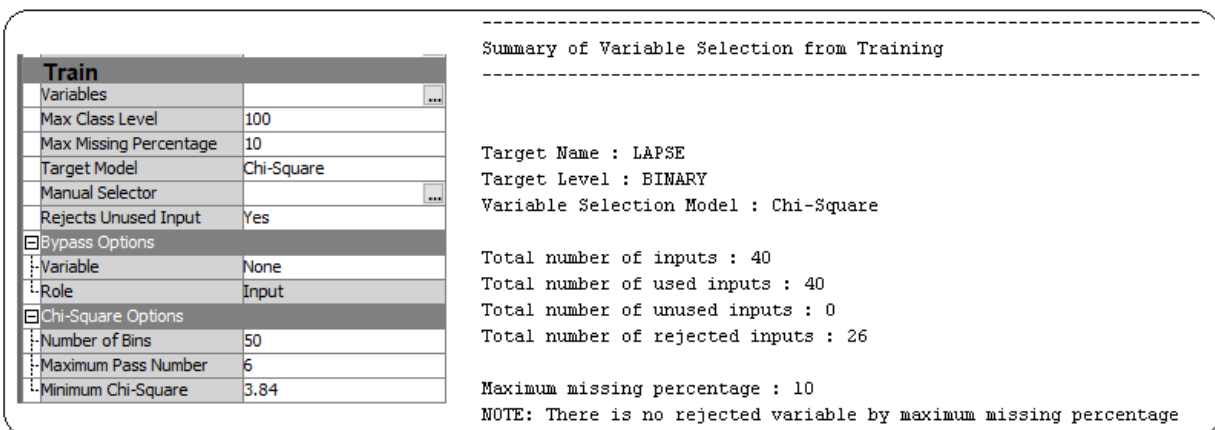
- Union the original dataset (with both minority and majority cases) with the new synthesised output.

### 3.2.3 Feature selection

#### 3.2.3.1 Chi-square

Even though there were only 40 variables after creating dummies for Insurer 1, feature engineering methods were considered to select only features which were relevant in predicting the target. Models were compared when variables were selected with the Chi-square test and PCA. The Chi-squared test analyses the relationship between variables and targets then ranks the variables based on their variable importance. In other words, it tests the level of dependency of a variable on a target.

The Chi-square method on variable selection node on SAS Enterprise Miner was used on a training set. As illustrated on Figure 3.1, all the categorical variables with a Chi-square value of  $P(\text{Chi-square statistic} > 3.84) \leq 0.05$  were kept. Twenty six variables were rejected by the Chi-square selection method. The remaining 14 variables were ranked based on their level of importance. It is better to have a model with lesser features than a model with many features that are not relevant to the target. Having lesser but relevant features may improve accuracy, computational time, and reduce the possibility of model overfitting.



**Figure 3.1: Variable Selection Summary**

### **3.2.3.2 *Principal component analysis***

Principal Component Analysis (PCA) methodology aims to reduce the dimensionality of a dataset by transforming the data to fewer variables while keeping most of the information. The method combines different variables to form new variables which are called principal components. The model often puts most of the information on the first principal component which then accounts for the most variation (Jaadi, 2020). The transformation process is done through calculations of eigenvectors and eigenvalues of covariance or correlation matrix. Eigenvalues of a covariance matrix were used in this dissertation. The number of principal components on this dissertation was set to 31. The same methods were followed for all the models in comparison. It was, however, difficult to interpret new variables formed as it was a combination of many variables to acquire a lot of information from them. For LR, the variables were further selected using the forward and backward variable selection method.

### **3.2.4 Model training and validation**

The models were trained and validated with a data partition of 60% training set and 40% validation set. The aim was to train and validate on both imbalanced and balanced datasets. However, testing on imbalanced data resulted in 99% accuracy, which was misleading as the model predicted the majority class for almost all examples. This illustrated the point that measuring accuracy alone can be misleading in cases of imbalanced data.

#### **3.2.4.1 *Logistic regression***

Logistic Regression (LR) models are trained with a comparison of forward and backward variable selection methods. Both methods are described in Section 2.1.1. As shown in

Table 3.6, variables with a P-value that is greater than 0.05 were regarded as less significant. Logit link function was used as a mapping function. Convergence and parameters estimates were automatically optimised.

**Table 3.6: Logistic Regression Setup**

	<b>Criteria</b>
Model selection	Forward and backward
Technique	GLM
Link Function	Logit
Variable significance	P= <0,05
Goodness of Fit	Akaike Information Criterion
Performance	Classification Matrix

**3.2.4.2 Support vector machine**

Support vector machines (SVMs) models with linear kernel function and polynomial kernel function were trained and compared. The penalty criteria (C) on both models were set to 1. The polynomial degree on the polynomial kernel was set to 2. Grid search methodology could not be used to optimise parameters as it was not yet available on the software. However, penalty parameter C=1 and the polynomial degree of 2 gave good results. Table 3.7 summarises the criteria settings for SVM.

**Table 3.7: Support Vector Machine Setup**

	<b>Polynomial</b>	<b>Linear</b>
DESCR	VALUE	
Task Type	C_CLAS	C_CLAS
Optimization Technique	Interior Point	Interior Point
Scale	YES	YES
Kernel Function	Polynomial	Linear
Kernel Degree	2	N/A
Penalty Method	C	C
Penalty Parameter	1	1
Tolerance (Max Iteration)	25	25
Tolerance	0,000001	0,000001
Execution Mode	Single-Machine	Single-Machine
Number of Threads	4	4

### **3.2.4.3 Neural network**

A multilayer perceptron NN with backpropagation learning and a multi-layered perceptron with Levenberg-Marquardt learning were built. Both models had three hidden layers. The maximum allowed iteration rate of 1000 was set on both learning algorithms as the models were having challenges to converge. Random bias and initial weights were assigned.

### **3.2.4.4 Decision tree**

The trees were built by specifying the split rules to maximise the split decision log-worth. Log-worth can be described as the statistic that is used to prune and grow the tree. It measures the best splitting rule that best classifies the target. The trees were automatically pruned, and entropy was used for evaluating splitting criteria. The maximum splitting rule was set to 10. The maximum depth splitting enabled the tree to be split up into 10 generations of root nodes. The original node (root node) is generation zero. Children of the original node are the first generation, children of the first generation are the second generation and so on. Due to the automatic pruning that was set, the model would have lesser generations. Since the study was dealing with a binary classifier, the maximum branch per node was set to 2. Leaf sizes were limited to a minimum of 10. Leaf size is basically the number of observations in each subset.

### **3.2.4.5 Gradient boosting**

The model was set up to have almost the same splitting rules as was in the normal DT. Maximum depth was set to 10, meaning that each DT would have 10 generations of root nodes. The maximum branch was two as the study was dealing with a binary target. The reuse variable was set to 2 – this meant that a variable could be used twice for splitting if it yielded the best results. The leaf fraction was set to 0.01. This was the minimum fraction of training observation a new branch was allowed to have out of the total training observation in the data. Several fractions were tested and 0.01 gave the best results.

### **3.2.4.6 Random forest**

An RF tree with a maximum of 100 individual trees was built on SAS E-Miner. The environment used PROC HPFOREST for building RFs (SAS, 2016). A RF is an ensemble

of several DTs. Similar to a normal tree and GB, for tuning parameters, a maximum depth splitting rule of 10 was set. Associations between variables and the target must be above a significant level on  $p=0.05$  so that a node can be split.

### 3.2.5 Performances measures

All models were compared for their level of accuracy and their statistical power. Confusion matrix also known as classification matrix was used to critically evaluate the models. A confusion matrix is a metric that is used to evaluate classifier models and provide insights into the predictions. It is illustrated in Table 3.8 on the following page. With a confusion matrix, it is easier to see if the model is constantly mislabelling one of the classes as another.

**Table 3.8: Confusion Matrix**

	Predicted Classes	
Actual Classes	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Accuracy rate which is a measure of correct prediction from overall cases was calculated by the below formula from the classification matrix.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3.3)$$

The higher the accuracy the better the model. However, this is not always the case especially when it comes to imbalanced data; as mentioned before, the results can be

misleading if interpreted in isolation. The following matrices were evaluated in conjunction with the accuracy rate.

Precision or positive predicted value is the number of true positives divided by total positive predictions. Precision focuses on how many positives were correctly predicted from all positives. Low precision indicates a high number of false positives.

$$Precision = \frac{TP}{TP + FP}, \quad (3.4)$$

Sensitivity or recall is the number of true positives divided by the actual number of all positives. Low recall indicates a high number of false negatives.

$$Recall = \frac{TP}{TP + FN}, \quad (3.5)$$

Specificity is the number of true negatives divided by the actual number of all negatives. A high number reflects the model was good at identifying true negatives.

$$Specificity = \frac{TN}{TN + FP}, \quad (3.6)$$

Misclassification error (ME) is the value of all wrongly classified predictions from total observations.

$$ME = \frac{FN + FP}{TN + TP + FN + FP}, \quad (3.7)$$

F-measure or F1-score is the balance between precision and recall and can be described by the below formula.



$$F1\_Score = 2 * \left( \frac{Precision * Recall}{Precision + Recall} \right), \quad (3.8)$$

ROC measures the relationship between true-positive (i.e., sensitivity) and false-positive (1-specificity). True positives can be described as outcomes where models predict the positive actual class correctly; in our case, when the model predicted lapses correctly, whereas true negative was when the model predicted non-lapses correctly. The ROC visualises the probability of an outcome at different thresholds.

A perfect classifier is towards the top left where the sensitivity rate is higher, and the false positive rate is less. The worst classifier is closer to the baseline. Anything in between reflects a better classifier. Coordinates (0, 1) represent a perfect model, meaning that all events are predicted correctly. The baseline shows points where the true positive rate equals the false positive rate, meaning that the rate of predicting lapses correctly and the rate of predicting non-lapses are the same.

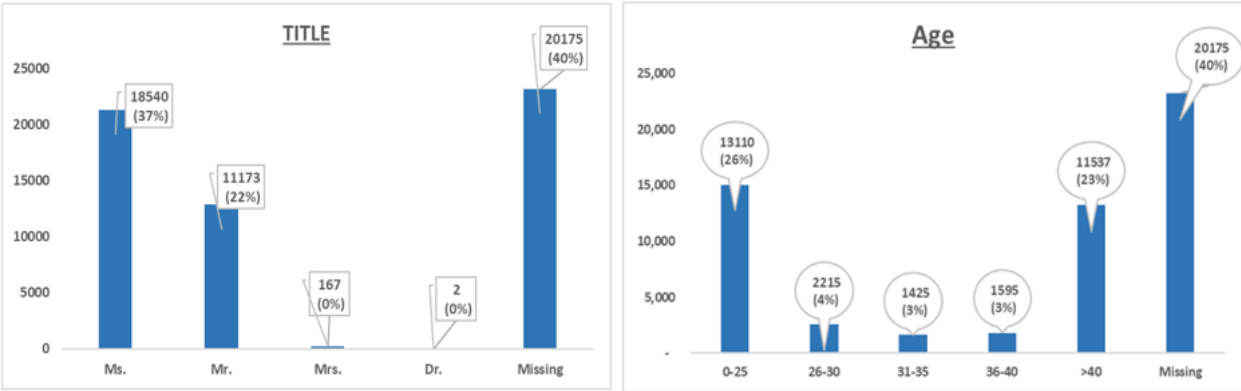
The AUC takes the value between zero and one where anything below 0.5 reflects a not so good model. 0.5 is similar to the baseline in ROC, which shows that the model does not have the discriminatory ability; 0.8 to 0.9 represent an excellent model; and anything between 0.5 and 0.8 shows a fairly good model (Mandrekar, 2010).

### **3.3 Results**

#### **3.3.1 Data analysis**

##### **3.3.1.1 Insurer 1: Data exploration**

Figure 3.2 shows the demographic distribution of Insurer 1. There are more female policyholders (38%) than males (22%) and 40% of the gender variable was unclassified. Most policyholders are in the extreme ends of the age distribution; 26% are less than 25 years and 23% are greater than 40 years. It was not so clear if the age captured was at the policy entry age or the age at the end of the reporting period (i.e., last recorded date).



**Figure 3.2: Insurer 1 – Demographic Distribution**

Table 3.9 shows the distribution of lapses per year. Policyholders that lapsed in the same year their policies commenced were 3.1% (410), 3.3% (608), and 4.4% (848) in the year 2017, 2018, and 2019, respectively. The number of policyholders that lapsed after a year of commencement was 14% for 2017, and 18% for 2018. The total number of lapsed policies were 15.9% whereas active policies were 84.1%. This clearly shows the issue of imbalanced data.

**Table 3.9: Insurer 1 – Lapses Per Year**

Effective_Dates	CNT_POLICIES	Policies(%)	LAPSE			Active	Active Policy(%)	Lapsed policies (%)
			2017	2018	2019			
2017	13274	26.5%	410	1953	926	9985	19.9%	6.6%
2018	17927	35.8%	0	608	3233	14086	28.1%	7.7%
2019	18856	37.7%	0	0	848	18008	36.0%	1.7%
<b>Total</b>	<b>50057</b>	<b>100.0%</b>	<b>410</b>	<b>2561</b>	<b>5007</b>	<b>42079</b>	<b>84.1%</b>	<b>15.9%</b>
<b>Total (%)</b>			<b>0.8%</b>	<b>5.1%</b>	<b>10.0%</b>	<b>84.1%</b>		

Table 3.10 presents the non-payment ratio for all lapsed policies. Only 9% of total lapsed policies seemed to have been consistent with their premium payments.

**Table 3.10: Insurer 1 – Lapses by Non-payments**

		LAPSES				
		2017	2018	2019	Total	Total (%)
Non_ payments (%)	0-20%	9	144	553	706	9%
	21-40%	80	449	750	1279	16%
	41-60%	296	1840	2496	4632	58%
	61-80%	12	109	178	299	4%
	>80%	13	19	18	50	1%
	Missing	0	0	1012	1012	13%
	Total	410	2561	5007	7978	1
	Total(%)	5%	32%	63%	100%	

**3.3.1.2 Insurer 2: Data exploration**

Table 3.11 represents the distribution of lapse by age and gender for Insurer 2. Thirty-four percent of the data is in force and 66% of the data had already lapsed. This is an unusual distribution of lapse and non-lapse. Lapse is usually a minority case. Of the 66% lapsed policies, 37% are males. The highest number of lapses (19%) are in the age group of 0-25 years.

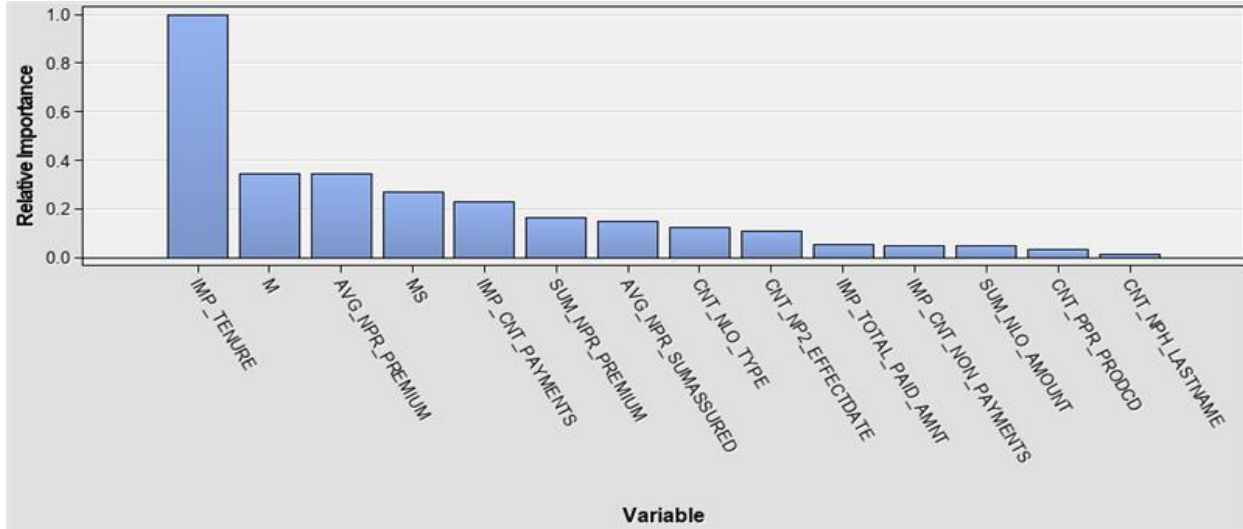
**Table 3.11: Insurer 2 – Lapses Per Age Group and Gender**

	Non-Lapse	Lapse	Non-Lapse(%)	Lapse(%)
0-25	2547	5100	5%	9%
26-30	1066	2334	2%	4%
31-35	1095	2344	2%	4%
36-40	1050	2137	2%	4%
>40	2829	3787	5%	7%
<b>F</b>	<b>8587</b>	<b>15702</b>	<b>15%</b>	<b>28%</b>
0-25	2963	5822	5%	10%
26-30	1576	3596	3%	6%
31-35	1542	3540	3%	6%
36-40	1364	2857	2%	5%
>40	3137	4905	6%	9%
<b>M</b>	<b>10582</b>	<b>20720</b>	<b>19%</b>	<b>37%</b>
<b>Total</b>	<b>19169</b>	<b>36422</b>	<b>34%</b>	<b>66%</b>

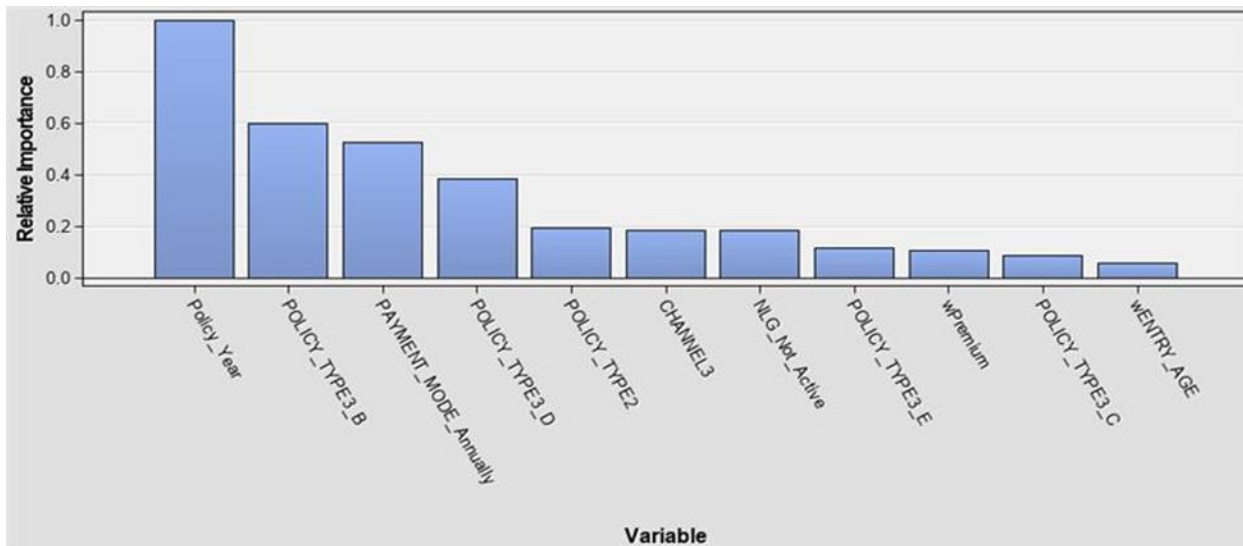
**3.3.1.3 Variable importance**

Figures 3.3 and 3.4 show the ranking of important variables when predicting lapses through the Chi-squared variable selection method for Insurer 1 and Insurer 2

respectively. Tenure was the most significant predictor of lapses for Insurer 1. Similarly, tenure was the highest significant predictor of lapses for Insurer 2. Payment and premium information are some of the variables that drive lapses for both insurers.



**Figure 3.3: Insurer 1 – Variable Importance**



**Figure 3.4: Insurer 1 – Variable Importance**

### 3.3.2 Logistic regression

Table 3.12 and Table 3.13 compares the performance of forward and backward variable selection in LR for Insurer 1 and Insurer 2 datasets respectively. Both forward and backward selection methods resulted in 22% misclassification on training and validation sets on Insurer 1. Similarly, for Insurer 2, both backward and forward model selection had the same misclassification rate of 26% and the same average squared error (ASE) of 18%. ASE shows the average squared difference between the actual value and the predicted value. The lower the value, the better the model.

**Table 3.12: Insurer 1 – Fit Statistics: Logistic Regression**

	Selected Model	Train: Misclassification Rate	Validation: Misclassification Rate	Train: Average Squared Error	Validation: Average Squared Error
Logistic Regression-Forward	Y	0.21612	0.21507	0.14708	0.1468
Logistic Regression-Backward		0.21688	0.21674	0.14705	0.14683

**Table 3.13: Insurer 2 – Fit Statistics: Logistic Regression**

	Selected Model	Train: Misclassification Rate	Validation: Misclassification Rate	Train: Average Squared Error	Validation: Average Squared Error
Logistic Regression- Forward		0.25688	0.25198	0.18075	0.17826
Logistic Regression- Backward	Y	0.25735	0.25155	0.18093	0.17846

### 3.3.3 Support vector machine

Table 3.14 and Table 3.15 compares the performance of polynomial kernel and linear kernel when predicting lapses for Insurer 1 and Insurer 2 respectively. A second-degree SVM-polynomial kernel performed better than an SVM trained on linear kernel on both datasets. For Insurer 1, the misclassification rate was 2% higher on linear kernel (21%) than on the polynomial kernel (19%).

**Table 3.14: Insurer 1 – Fit Statistics: Support Vector Machine**

	Selected Model	Train: Misclassification Rate	Validation: Misclassification Rate	Train: Average Squared Error	Validation: Average Squared Error
SVM- Polynomial Kernel	Y	0.1923	0.17171	0.17523	0.17539
SVM-Linear Kernel		0.21937	0.2174	0.16715	0.16698

**Table 3.15: Insurer 2 – Fit Statistics: Support Vector Machine**

	Selected Model	Train: Misclassification Rate	Validation: Misclassification Rate	Train: Average Squared Error	Validation: Average Squared Error
SVM- Polynomial Kernel	Y	0.25796	0.25345	0.19644	0.1953
SVM-Linear Kernel		0.27289	0.27144	0.19283	0.19147

### 3.3.4 Neural network

Table 3.16 and Table 3.17 compared the performance of backpropagation learning with Levenberg-Marquardt on a multilayer perceptron NN for Insurer 1 and Insurer 2 respectively. For Insurer 1, the NN trained with Levenberg-Marquardt had a misclassification rate of 19% and the NN trained with backpropagation had a misclassification rate of 21%. The ASE was very minimal for both models (i.e., < 14%). Similarly, Levenberg-Marquardt learning outperformed backpropagation learning on the Insurer 2 dataset. Both NN learnings performed well, and the results were close enough.

**Table 3.16: Insurer 1 – Fit Statistics: Neural Networks**

	Selected Model	Train: Misclassification Rate	Validation: Misclassification Rate	Train: Average Squared Error	Validation: Average Squared Error
Neural Network- Levenberg	Y	0.19634	0.19444	0.13248	0.13248
Neural network-Backprob		0.21662	0.21279	0.14451	0.14392

**Table 3.17: Insurer 2 – Fit Statistics: Neural Networks**

	Selected Model	Train: Misclassification Rate	Validation: Misclassification Rate	Train: Average Squared Error	Validation: Average Squared Error
Neural Network- Levenberg	Y	0.25030	0.24568	0.17640	0.17420
Neural network-Backprob		0.25818	0.25201	0.18335	0.18110

### 3.3.5 Trees models

Table 3.18 and Table 3.19 compared the DT model which is a single classifier and ensemble models; namely, GB and RF for Insurer 1 and Insurer 2 respectively. For Insurer 1, the RF had a misclassification rate of 10% in training and 12% in validation. Gradient boost (GB) resulted in a good misclassification rate of 8% and 9% for training and validation sets, respectively. Normal DT resulted in misclassification of 13% for both training and validation sets. The ASE for all the models was very minimal (less than or equal to 10%) on both training and validation sets. For Insurer 2, all the tree models resulted in a misclassification rate of 24%.

**Table 3.18: Insurer 1 – Fit Statistics: Tree Models**

	Selected Model	Train: Misclassification Rate	Validation: Misclassification Rate	Train: Average Squared Error	Validation: Average Squared Error
Gradient Boosting	Y	0.07949	0.08772	0.06026	0.06586
Random Forest		0.10335	0.12412	0.08313	0.09461
Decision Tree		0.12790	0.13135	0.09457	0.09826

**Table 3.19: Insurer 2 – Fit Statistics: Tree Models**

	Selected Model	Train: Misclassification Rate	Validation: Misclassification Rate	Train: Average Squared Error	Validation: Average Squared Error
Gradient Boosting		0.24019	0.24309	0.16554	0.16931
Random Forest		0.23803	0.24309	0.16679	0.16987
Decision Tree	Y	0.24020	0.24119	0.17130	0.17223

**3.3.6 Model comparisons**

According to Siemes (2016), there is no perfect performance measure. In this dissertation, models were evaluated on six performance measures, and the dominating best model across the measures was crowned the best modelling method when predicting lapses. Table 3.20 shows the performance results for the nine models that were built using two different insurance datasets. The colours in the table represent the best model (green) and worst model (yellow) per performance measure.

All the tree-based methodologies (i.e., GB, RF, and normal DTs) showed the best results as they dominated across all performance measures. They outperformed other algorithms when the data was trained and validated using different feature selection methods; namely, PCA and Chi-squared and when the data was trained on different insurer datasets. Hassouna *et al.* (2015) and Sabbeh (2018) also presented similar results where the tree models outperformed other ML algorithms. Support vector machine (SVM) with a linear kernel showed the overall worst performance across all datasets.

The GB method had the best overall performance with 91.6% accuracy on average (training and validation average), 92.3% precision, 90.8% sensitivity, 92.5 specificity, and 91.6% F-measure; followed by RF with 88.6% average accuracy, 86.4% precision, 91.7% sensitivity, 85.5 specificity, and 89.0% F-measure for variables selected by Chi-square test on Insurer 1’s dataset.

Random forest (RF) had the best performance of 86.0% average accuracy, 83.1% precision, 90.4% sensitivity, 81.6% specificity, and 86.6% F-measure for variables selected through PCA on Insurer 1’s dataset.



Best precision was achieved by the GB method on all the datasets, i.e., 92.3% 84,2% and 75.5% on Insurer 1 Chi-squared, Insurer 1 PCA and Insurer 2 Chi-squared respectively. This implies that the model was able to predict actual lapses (true positives) correctly. This is what was more important in this dissertation, as insurers want to know who is likely to lapse so that they can implement adequate retention strategies. People who are unlikely to leave were not the target in this experiment. Support vector machine (SVM) with a linear kernel showed the worst precision compared to other models.

Random forest (RF) showed the overall best sensitivity – this implies that a high number of actual lapses were correctly identified by the model. The best F-measure for Insurer 1 was GB and RF for Insurer 2.

All the models resulted in the worst specificity (36.5% on average) and very high sensitivity on the Insurer 2 dataset (94.9% on average), that is, the models were good at identifying policyholders that will lapse, but they do have limitations when identifying policyholders that will not lapse. This was on a 50% threshold across all the models.

**Table 3.20: Average Model Performance (Training and Validation)**

		Random Forest	Gradient Boosting	Decision Tree	SVM Polynomial	SVM Linear	LR-Forward	LR-Backward	NN Levenberg	NN Back-Prob	
Insurer1	Chi-Squared	Accuracy	88,6%	91,6%	87,0%	80,8%	78,2%	78,4%	78,3%	80,5%	78,5%
		Misclassifica	11,4%	8,4%	13,0%	19,2%	21,8%	21,6%	21,7%	19,5%	21,5%
		Precision	86,4%	92,3%	86,4%	77,1%	75,1%	75,7%	75,4%	79,3%	76,3%
		Specificity	85,5%	92,5%	86,2%	74,0%	72,0%	73,2%	72,6%	78,4%	74,3%
		Sensitivity	91,7%	90,8%	87,9%	87,6%	84,3%	83,7%	84,0%	82,5%	82,7%
		F- measure	89,0%	91,6%	87,1%	82,0%	79,4%	79,5%	79,5%	80,9%	79,4%
	PCA	Accuracy	86,0%	85,9%	82,9%	80,9%	79,3%	79,4%	79,4%	80,9%	79,7%
		Misclassifica	14,0%	14,1%	17,1%	19,1%	20,7%	20,6%	20,6%	19,1%	20,3%
		Precision	83,1%	84,2%	81,3%	77,6%	76,3%	77,0%	76,7%	79,8%	77,5%
		Specificity	81,6%	83,4%	80,4%	75,0%	73,6%	74,9%	74,4%	79,0%	75,6%
		Sensitivity	90,4%	88,4%	85,5%	86,7%	84,9%	84,0%	84,4%	82,8%	83,9%
		F- measure	86,6%	86,2%	83,3%	81,9%	80,4%	80,4%	80,3%	81,3%	80,5%
Insurer 2	Chi-Squared	Accuracy	75,9%	75,8%	75,8%	74,4%	72,8%	74,6%	74,6%	75,2%	74,5%
		Misclassifica	24,1%	24,2%	24,2%	25,6%	27,2%	25,4%	25,4%	24,8%	25,5%
		Precision	74,5%	75,5%	74,9%	73,1%	71,4%	74,1%	74,0%	74,8%	74,1%
		Specificity	37,6%	42,5%	39,7%	32,5%	26,0%	37,5%	36,9%	39,9%	37,6%
		Sensitivity	96,1%	93,4%	94,9%	96,5%	97,4%	94,0%	94,4%	93,8%	93,9%
		F- measure	84,0%	83,5%	83,7%	83,2%	82,4%	82,9%	82,9%	83,2%	82,8%

Table 3.21 shows accuracy per probability band for all nine models for Insurer 1. Probabilities were divided into 10 bands from the highest to the lowest. Performance was monitored at each band. Table 3.22 reflect the distribution of total policies at each probability band, i.e., in Table 3.22, GB predicted that 29.8% of total policies had a 0-10% chance of lapsing. As shown in Table 3.21, the model was 99% accurate at predicting policies in that 0-10% band.

Decision tree (DT) and RF classify most of the proportion of the data on either the lowest probability band (0-10% band; 26.2% average of total policies) or the highest band (90-100% band; 25% of total policies). The models had 99% and 98% prediction accuracy respectively on these bands. Most of the predictions on SVM lie between 30-70% chance of lapsing. Thus, it had moderate lapse probabilities for majority of the policies.

**Table 3.21: Accuracy Per Prediction Band**

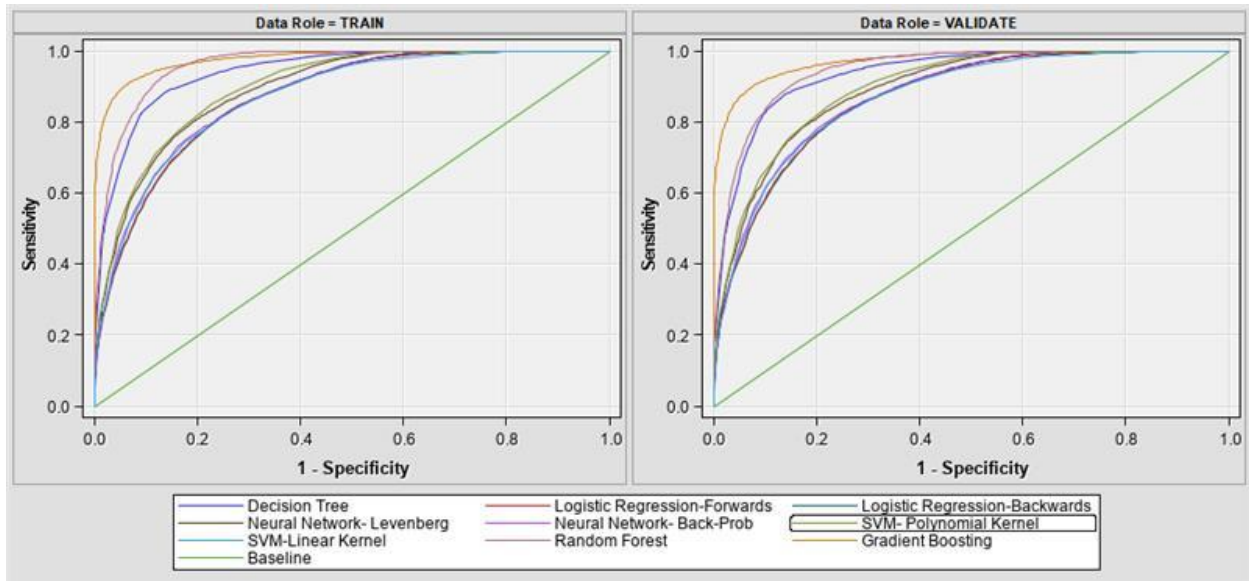
Model	0-10%	10-20%	20-30%	30-40%	40-50%	50-60%	60-70%	70-80%	80-90%	90-100%	Overall Accuracy
Gradient Boosting	98,9%	90,8%	83,0%	72,8%	40,3%	55,5%	70,2%	82,5%	92,7%	99,6%	92,1%
Random Forest	100,0%	99,6%	95,0%	80,0%	40,3%	60,2%	72,0%	84,7%	92,9%	98,1%	89,7%
Decision Tree	98,6%	86,3%	74,5%	66,0%	44,7%	54,4%	64,4%	77,1%	83,6%	96,3%	87,2%
Svm- Polynomial Kernel	100,0%	100,0%	99,9%	99,4%	23,8%	54,4%	81,0%	95,0%	97,2%	92,0%	80,8%
Neural Network- Levenberg	99,4%	87,4%	74,2%	64,0%	43,6%	52,4%	65,4%	75,8%	86,4%	94,6%	80,4%
Logistic Regression-Forwards	97,0%	81,4%	67,5%	64,7%	40,1%	53,0%	62,7%	74,8%	85,2%	95,7%	78,4%
Neural Network- Back-Prob	98,2%	81,0%	69,7%	64,5%	41,5%	48,7%	63,4%	74,5%	87,0%	96,4%	78,3%
Logistic Regression-Backwards	97,3%	82,2%	69,2%	64,2%	39,4%	52,8%	63,1%	75,3%	85,2%	95,6%	78,3%
Svm-Linear Kernel	99,2%	99,7%	98,0%	90,6%	33,4%	50,2%	70,0%	84,2%	93,8%	98,0%	78,1%
Average Band Accuracy	98,7%	89,8%	81,2%	74,0%	38,6%	53,5%	68,0%	80,4%	89,3%	96,3%	

**Table 3.22: Policies Per Prediction Band**

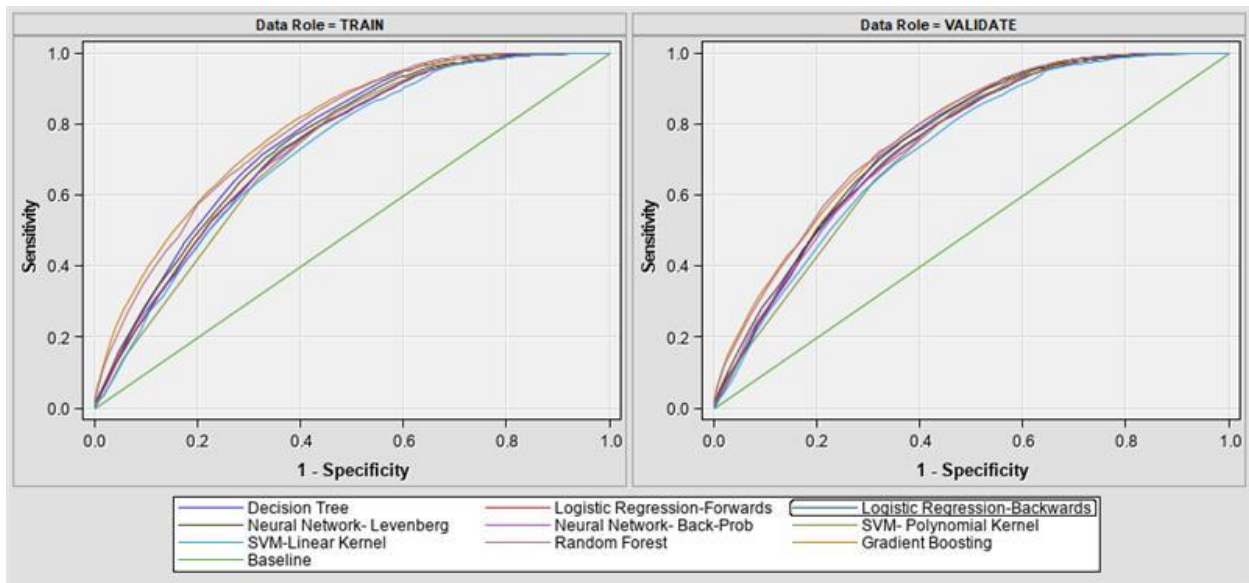
Total Policies Per Prediction Band											
	0-10%	10-20%	20-30%	30-40%	40-50%	50-60%	60-70%	70-80%	80-90%	90-100%	Total
Decision Tree	25,0%	11,6%	4,0%	7,3%	1,1%	4,3%	1,0%	5,5%	14,2%	26,0%	100%
Gradient Boosting	29,8%	8,7%	5,1%	4,0%	3,2%	2,9%	3,1%	4,2%	6,8%	32,2%	100%
Logistic Regression-Backwards	20,4%	5,9%	5,3%	6,2%	6,7%	8,2%	10,7%	13,9%	13,0%	9,7%	100%
Logistic Regression-Forwards	20,6%	6,5%	5,5%	6,0%	6,3%	7,7%	10,2%	13,6%	14,1%	9,5%	100%
Neural Network- Back-Prob	20,2%	5,3%	5,8%	6,8%	8,0%	7,6%	8,9%	12,3%	17,1%	8,1%	100%
Neural Network- Levenberg	21,4%	1,4%	10,4%	8,3%	6,7%	6,3%	7,5%	10,9%	12,7%	14,4%	100%
Random Forest	23,7%	6,0%	6,5%	5,4%	5,0%	5,4%	6,9%	9,3%	14,9%	16,9%	100%
SVM- Polynomial Kernel	0,1%	1,2%	4,8%	11,1%	26,1%	15,4%	27,8%	11,8%	1,5%	0,1%	100%
SVM-Linear Kernel	0,3%	2,9%	7,3%	14,2%	19,3%	11,5%	17,2%	15,8%	8,8%	2,7%	100%

As illustrated in Figure 3.5, Insurer 1 models performed well based on the ROC curve as all the curves are towards the top left corner. This reflects the discriminating ability of the

models. Similarly, as illustrated in Figure 3.6, models performed well on the Insurer 2 dataset. The models had better results on the Insurer 1 dataset than the Insurer 2 dataset.



**Figure 3.5: Insurer 1 – ROC Curve: Chi-square**



**Figure 3.6: Insurer 2 – ROC Curve: Chi-square**

Table 3.23 shows the AUC for all models on all datasets. Insurer 1 had an AUC of +0.8 for all models, which reflects a good-excellent model. Similarly, AUC for Insurer 2 shows a good performance (+0.7)

**Table 3.23: Area Under Curve**

Model description	Insurer1 Chi_Square		Insurer1 PCA		Insurer2-Chi Square	
	Train	Validation	Train	Validation	Train	Validation
Random Forest	0.96	0.95	0.95	0.92	0.78	0.77
Gradient Boosting	0.98	0.97	0.95	0.93	0.79	0.77
Decision Tree	0.94	0.94	0.91	0.89	0.76	0.75
Neural Network-Levenberg	0.89	0.89	0.90	0.90	0.75	0.76
SVM-Polynomial	0.90	0.90	0.90	0.90	0.73	0.74
Neural Network-Back probaga	0.87	0.88	0.88	0.88	0.74	0.74
Logistic Regression-Forwards	0.87	0.87	0.88	0.88	0.74	0.74
LogisticRegression- Backwars	0.87	0.87	0.88	0.88	0.74	0.74
SVM-Linear Kernel_PCA	0.87	0.87	0.88	0.88	0.72	0.73

**3.3.7 Results discussions**

Ćurak, Podrug and Poposki (2015) showed that the most influential factor of policy lapsation is the change in the financial status of the policyholder, income level and the duration of the policy. This study also found that tenure and premium payments are in the top three reasons for lapses in both insurers’ datasets, additionally; premium and sum assured information were also found to be the contributing factor for lapses. Only nine percent of the total lapses on Insurer 1 paid more than 80% of their policies, thus they have been consistent with their payments. Insurer 2 dataset also showed that younger policyholders (i.e., <25) lapses more than the older policyholders. Mojekwu( 2011) also showed that in Nigeria, young people take up policies and terminate them early. The reason young people are terminating their policies early is illustrated by Valdez *et al.* (2014), he showed that this may be because the younger policyholders might be still looking elsewhere. However, in this study, the data showed that the mid-age group are staying (>25 and <40). A quarter of total lapses (24%) are older policyholders. This was not expected as Valdez *et al.* (2014) showed that Individuals with health risks and uninsurable issues do not usually lapse their policies. Health risk and uninsurable is usually highly correlated to age. The results also showed that most policyholders will lapse within a year of getting a life insurance policy.

In this study, the feature selection method, namely, Chi-square was compared with PCA based on their performance accuracy. The Chi-squared method outperformed PCA on tree models, namely, GB, DT and RF with a percentage difference of 5.8%, 4.1% and 2.6% respectively. However, it was slightly outperformed by PCA on NNs, LR and SVM models by an average percentage difference of 0.8%. The Chi-squared method resulted in 14 forward significant variables using the P-value as the selection threshold, whereas PCA resulted in new variables where the interpretation was not as simple as the Chi-squared. A study by Ravichandran (2016) also showed the superiority of Chi-square performance when compared to other dimensionality reduction methods like PCA, Information Gain, Gain Ratio, and Quantile Regression model. His study also resulted in lesser time taken and lesser selected variables than the other methodologies explored.

Logistics models were trained using forward variable selection and backward selection methods. Similar to the results presented by Maxwell and Obinna (2018), there was not much difference between the backward and forward variable selection methods. Thus, both selection methods produced the same model on all datasets.

This study also found that the polynomial kernel performs better than the linear kernel on SVMs. The percentage difference in accuracy was 2.6% and 1.6% on average for Insurer 1 and Insurer 2 respectively. The average area under the curve for a polynomial kernel outperformed linear kernel for Insurer 1 (0.90 and 0.73) and Insurer 2 (0.88 and 0.72) respectively. These results contradict the findings illustrated by Hossain and Miah (2016), where linear outperformed polynomial kernel on both F-measure and AUC when predicting customer churns.

From the literature, different researchers made different conclusions about the performances of kernels. Polynomial showed superiority over linear, RBF and sigmoid kernels in the study by Nanda *et al.* (2018). RBF outperformed linear and 3<sup>rd</sup>-degree polynomial kernels on the study by Yekkehkhany *et al.* (2014) and linear models outperformed RBF-Gaussian, polynomial, linear, sigmoid, laplacian and ANOVA RBF on a study by Hossain and Miah (2016). This clearly shows that the performance of kernels highly depends on parameter optimisations.

As expected, even though the models were built similarly, they performed differently on different datasets. This may have been caused by different reasons, ie., the distribution and variation of the data, the models' architecture, parameter optimisations set up, different resulting features through variable selections methods, the models may have been more suitable on one dataset than the other.

All the Insurer 2 models trained through Chi-squared variable selection model resulted in the worst average specificity (36.5%), whereas, on Insurer 1, the same models had good average specificity (78.9%) These further illustrate that models perform differently on different datasets.

Gradient Boosting (GB) and RF consistently outperformed single classifiers. This finding reiterates that ensemble model generally performs better than the single classifiers (Dietterich, 2000; Kim *et al.*, 2006; Yang *et al.*, 2007; Lessmann *et al.*, 2015; Gavrishchaka *et al.*, 2018; Sabbeh, 2018; Loisel *et al.*, 2019).

# CHAPTER FOUR

## CONCLUSION

### 4.1 Summary

High lapse rates can damage an insurance company's reputation and may lead to insolvency. The accurate prediction will help the insurer to implement customised retentions strategies and minimise the risk that comes with losing clients. This dissertation aimed to illustrate the predictive power of different classifier models, their robustness, flexibility, sensitivity, and generalisation ability when presented with a different dataset. The dissertation also aimed to illustrate the impact of different feature selection methodologies on the models and highlight features that directly drive lapses using in-depth data analysis.

### 4.2 Findings and Recommendations

Nine ML algorithms were built, namely three tree models (i.e., a DT, GB, and RF); two SVM models (i.e., SVM trained with linear kernel and SVM trained with the polynomial kernel); two NNs (i.e., NN Levenberg-Marquardt, backpropagation NN); and two LR models with variable selection through forward and backward selection process. Models were built and compared based on two variable selection methodologies namely PCA and Chi-squared on two different insurer datasets.

All the models performed well (i.e., +70% accuracy, precision, sensitivity, and F-measure) on both the training and validation sets. The models were robust, and they showed the ability to generalise well. The accuracy percentage difference between training and validation was less than 5% for all the models. This study has shown empirical evidence on application of ML models in lapse predictions.

Although different models performed differently on different datasets, the ensemble model (i.e., GB) gave the overall best average performance of 91.6% accuracy, 92.3% precision,

90.8% sensitivity, 85.5% specificity, and 91.6% F-measure followed by RF with 86.0% accuracy, 83.1% precision, 90.4% sensitivity, 81.6% specificity, and 86.6% F-measure on Insurer 1's dataset with variables selected through Chi-square. Similarly, the ensemble models (i.e., GB, RF) showed the best results for variables selected through PCA. The same picture was observed on Insurer 2's dataset. SVM with a linear kernel consistently showed the overall worst performance across all datasets.

The study also found that the tree models place most of the policies on extremely high or extremely low probability bands as opposed to other models. A quarter (25%) of all policyholders had a 90-100% chance of lapsing on average. Another quarter (26%) had a 0-10% chance of lapsing. The models were 99% and 98% accurate at predicting those lower (0-10%) and higher (90-100%) probability bands. We recommend that the insurer must have solid retentions strategies for the 26% of policyholders with the highest chance of lapsing as the tree models were 98% accurate on average at identifying them.

The Chi-squared variable selection method improved accuracy on tree models by 4.2% on average. However, NNs, SVMs, and LRs produced similar models when trained on PCA and Chi-squared.

Both forward and backward logistic models gave the same model, most of the literature showed similar results. Polynomial kernel consistently outperformed linear kernel on all the datasets, however, the percentage difference was very small.

On both datasets, policy tenure was the most significant lapse predictor. Other important features included premium payments mode (i.e., annually, monthly, quarterly, and semi-annually), sum assured, and payment information. This study shows empirical evidence that younger clients are at risk of lapsing their policies. Insurers should have good retentions strategies for younger clients.

The findings showed that appropriate parameter tuning and model boosting improved the prediction of lapses in life insurance industry. These findings support the current idea of the importance of boosting ML algorithms, and it also illustrates the predictive power of



ensemble learning over single classifiers. This dissertation suggests that insurers base their model implementation on a trial and test of different ML algorithms rather than just one model. It also recommends the use of ensemble models over single classifiers when predicting lapses in life insurance. Insurers must be on the lookout of newer prediction and optimization techniques.

### **4.3 Limitations and Future Work**

Due to time constraints, some topics were discovered in the study, but we could not dive into them fully, i.e., In this study, variable selection through forward and backward LR gave similar results. There is quite a lot of feature selection methodology available on R, Python, and other statistical packages that we would like to explore, i.e., Boruta, Least Absolute Shrinkage and Selection Operator and Recursive Feature Elimination. Also, this study compared single classifiers with ensemble models. We would like to incorporate hybrid models in the future.

It is quite challenging to find secondary lapse data as the information is too private. Both datasets that were used had few dependent variables, and they were both not big enough. We would like to incorporate economic features as well the credit information of the policyholder in the data as they have been proven to be a significant predictor of lapses in the financial industry.

## REFERENCES

- Abbas, A. & Mohammed, A. (2020). Inferences about the use of linear regression and logistic regression. Inferences about the use of linear regression and logistic regression. *International Journal of Recent Scientific Research*, 11(8): 39547–39552. <https://doi.org/10.24327/ijrsr.2020.1108.5525>
- Abreu, J. (2019). *Customer lifetime value in insurance*. Nova Information Management School, Universidade Nova de Lisboa. Available at: <https://run.unl.pt/bitstream/10362/62423/1/TAA0027.pdf> [Accessed on: 10 October 2020].
- Akinwande, M.O., Dikko, H.G. & Samson, A. (2015). Variance inflation factor: As a condition for the inclusion of suppressor variable(s) in regression analysis. *Open Journal of Statistics*, 05(07): 754–767. <https://doi.org/10.4236/ojs.2015.57075>
- Aleandri, M. (2017). Modeling dynamic policyholder behavior through machine learning techniques. University of La Sapienza, Rome, Dept. of Statistical Sciences.
- Alsaadi, A. & Mijwil, M. (2019). *Overview of neural networks*. Available at: <https://www.researchgate.net/publication/332655457> [Accessed 12 July 2020].
- Alzubi, J., Nayyar, A. & Kumar, A. (2018). Machine learning from theory to algorithms: An overview. *Journal of Physics: Conference Series*, 1142(1): 1–15.
- Ardabili, S., Mosavi, A. & Várkonyi-Kóczy, A.R. (2020). Advances in machine learning modeling reviewing hybrid and ensemble methods. *Lecture Notes in Networks and Systems*, 101(August): 215–227. [https://doi.org/10.1007/978-3-030-36841-8\\_21](https://doi.org/10.1007/978-3-030-36841-8_21)
- Austin, P.C. & Tu, J.V. (2004). Automated variable selection methods for logistic regression produced unstable models for predicting acute myocardial infarction mortality. *Journal of Clinical Epidemiology*, 57(11): 1138–1146. <https://doi.org/10.1016/j.jclinepi.2004.04.003>

Awad, M. & Khanna, R. (2015). Efficient learning machines: Theories, concepts, and applications for engineers and system designers. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, (July): 1–248. <https://doi.org/10.1007/978-1-4302-5990-9>

Badr, W. (2019). 6 different ways to compensate for missing values in a dataset (data imputation with examples). *Towards Data Science*. Available at: <https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779> [Accessed on: 20 October 2020].

Badr, Y., Mohamed, N. & Mohamed, A. (2018). Recent trends in big data analytics towards more enhanced insurance business models. *International Journal of Computer Science and Information Security*, 15(12): 39–45.

Barsotti, F., Milhaud, X. & Salhi, Y. (2016). Lapse risk in life insurance: Correlation and contagion effects among policyholders' behaviors. *Insurance: Mathematics and Economics*, 71(January): 317–331. <https://doi.org/10.1016/j.insmatheco.2016.09.008>

Behm, S., Deetjen, U., Kaniyar, S., Methner, N. & Münstermann, B. (2019, February 4). Digital ecosystems for insurers : Opportunities through the Internet of Things. [Online]. *McKinsey & Company*. Available at: <https://www.mckinsey.com/industries/financial-services/our-insights/digital-ecosystems-for-insurers-opportunities-through-the-internet-of-things#>

Bentéjac, C., Csörgő, A. & Martínez-Muñoz, G. (2019). *A comparative analysis of XGBoost*. Universidad Autónoma de Madrid. Available at: <http://arxiv.org/abs/1911.01914> [Accessed on: 10 October 2020].

Bennett DA. How Can I Deal With Missing Data In My Study? *Aust N Z J Public Health*. 2001;25(5). <https://doi.org/10.1111/j.1467-842X.2001.tb00294.x>

Boser, B.E., Guyon, I.M. & Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the Fifth Annual 28 Workshop on Computational*

*Learning Theory*, 9. [https://doi.org/10.1007/978-3-540-30573-6\\_7](https://doi.org/10.1007/978-3-540-30573-6_7)

Bernstein, M. (2016) *Sigmoid functions*. Available at: <https://mbernste.github.io/files/notes/SigmoidFunction.pdf> [Accessed on: 10 October 2020].

Berwick, R. (2003) *An idiot's guide to support vector machines (SVMs): A new generation of learning algorithms key ideas*. Massachusetts Institute of Technology. Cambridge, MA. Available at: <http://www.cs.ucf.edu/courses/cap6412/fall2009/papers/Berwick2003.pdf> [Accessed on: 12 November 2020].

Bewick, V., Cheek, L. & Ball, J. (2005). Statistics review 14: Logistic regression. *Critical Care*, 9(1): 112–118). <https://doi.org/10.1186/cc3045>

Biagini, F., Huber, T., Jaspersen, J.G. & Mazzon, A. (2021). Estimating extreme cancellation rates in life insurance. *Journal of Risk and Insurance*, 88(4): 971–1000. <https://doi.org/10.1111/jori.12336>

Biddle, R., Liu, S. & Xu, G. (2018) Automated underwriting in life insurance: Predictions and optimisation. In *Proceedings of Australasia Database Conference*. Cham, Switzerland: Springer. <https://doi.org/10.1007/978-3-319-92013-9>

Błaszczczyński, J. & Stefanowski, J. (2015). Neighbourhood sampling in bagging for imbalanced data. *Neurocomputing*, 150(PB): 529–542. <https://doi.org/10.1016/j.neucom.2014.07.064>

Boateng, E.Y. & Abaye, D.A. (2019). A review of the logistic regression model with emphasis on medical research. *Journal of Data Analysis and Information Processing*, 07(04), 190–207. <https://doi.org/10.4236/jdaip.2019.74012>

Bolancé, C., Guillen, M. & Padilla-Barreto, A.E. (2016). Predicting probability of customer churn in insurance. *Lecture Notes in Business Information Processing*, 254(January): 82–91. [https://doi.org/10.1007/978-3-319-40506-3\\_9](https://doi.org/10.1007/978-3-319-40506-3_9)

Boodhun, N. & Jayabalan, M. (2018). Risk prediction in life insurance industry using supervised learning algorithms. *Complex & Intelligent Systems*, 4(2): 145–154. <https://doi.org/10.1007/s40747-018-0072-1>

Botha, A. (2017). *Financial behaviours of customers as determinants for risk aversion and insurance consumption in South Africa* (Unpublished master's thesis). University of Pretoria, Pretoria. Available at: [https://repository.up.ac.za/bitstream/handle/2263/64886/Botha\\_Financial\\_2017.pdf?sequence=1&isAllowed=y](https://repository.up.ac.za/bitstream/handle/2263/64886/Botha_Financial_2017.pdf?sequence=1&isAllowed=y)

Brandusoiu, I. & Todorean, G. (2013). Churn prediction in the telecommunications sector using support vector machines. *Fascicle of Management and Technological Engineering* <https://doi.org/10.15660/auofmte.2013-1.2772>

Breiman, L. (2001). Random forests. *Machine Learning*, 1–33. <https://doi.org/10.14569/ijacsa.2016.070603>

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and regression trees. Statistics/probability series*. Wadsworth & Brooks/Cole Advanced Books & Software. <https://doi.org/10.1201/9781315139470>

Brown, J.M. & Schmidt, N.A. (2009). Getting the most out of conferences. *Nursing*, 39(4): 52–55. <https://doi.org/10.1097/01.nurse.0000348419.49806.37>

Burez, J. & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3 PART 1): 4626–4636. <https://doi.org/10.1016/j.eswa.2008.05.027>

Burri, R.D., Burri, R., Bojja, R.R. & Buruga, S.R. (2019). Insurance claim analysis using machine learning algorithms. *International Journal of Innovative Technology and Exploring Engineering*, 8(6 Special Issue 4): 577–582. <https://doi.org/10.35940/ijitee.F1118.0486S419>

Cai, J. (2006). Decision tree pruning using expert knowledge. The Graduate Faculty of The University of Akron Available at: [https://etd.ohiolink.edu/apexprod/rws\\_etd/send\\_file/send?accession=akron1158279616&disposition=inline](https://etd.ohiolink.edu/apexprod/rws_etd/send_file/send?accession=akron1158279616&disposition=inline) [Accessed on 12 August 2020].

Cao, X. H., Stojkovic, I. & Obradovic, Z. (2016). A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC Bioinformatics*, 17(1). <https://doi.org/10.1186/s12859-016-1236-x>

Carson, M.J. & Forster, M.D. (2000). Suitability and life insurance policy replacement. *Journal of Insurance Regulation*, 18(4): 427–447.

Chakure, A. (2019). *Logistic regression, getting started with logistic regression theory*. Available at: <https://medium.com/@aaaanchakure/logistic-regression-18c126a94460> [Accessed on: 20 October 2020].

Chambers, M. (2000). Queuing network construction using artificial neural networks (Unpublished doctoral dissertation). Columbus: Ohio State University.

Chawla, N.V., Bower, K., Hall, L. & Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(February): 321–357. <https://doi.org/10.1613/jair.953> [Accessed on: 12 October 2020].

Cheek, P.J., McCullagh, P. & Nelder, J.A. (1990). *Generalized linear models*. 2nd edition. New York: Chapman and Hall. Available at: <http://www.utstat.toronto.edu/~brunner/oldclass/2201s11/readings/glmbook.pdf> [Accessed on: 13 October 2020].

Chen, C., Liaw, A. & Breiman, L. (1999). *Using random forest to learn imbalanced data*. Department of Statistics, UC Berkeley.

Ćurak, M., Podrug, D. & Poposki, K. (2015). Policyholder and insurance policy features as determinants of life insurance lapse - evidence from Croatia. *Economics and Business Review*, 15(3): 58–77. <https://doi.org/10.18559/ebr.2015.3.5>

Chowdhury, M.Z.I. & Turin, T.C. (2020). Variable selection strategies and its importance in clinical prediction modelling. *Family Medicine and Community Health*, 8(1). <https://doi.org/10.1136/fmch-2019-000262>

Datta, L. (2020). *A survey on activation functions and their relation with Xavier and HE normal initialization*. Available at: <http://arxiv.org/abs/2004.06632> [Accessed 10 July 2020].

Deloitte Touche Tohmatsu Limited. (2016). *SAM pillar 1 requirements for solo insurers and insurance groups training manual*. Available at: [https://www2.deloitte.com/content/dam/Deloitte/za/Documents/financial-services/SAM\\_Pillar1\\_Training\\_Manual\\_Revised.pdf](https://www2.deloitte.com/content/dam/Deloitte/za/Documents/financial-services/SAM_Pillar1_Training_Manual_Revised.pdf) [Accessed on: 12 October 2020].

Damisa, S., Bello, Y., Ajadi, N., Agboola, S., Tasi'u, M. & Musa, F.N. (2017). On the comparison of some link functions of binary response analysis under symmetric and asymmetric assumptions. *Biomedical Statistics and Informatics*, 2(5): 145–149.

Dietterich, T.G. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1857: 1–15. [https://doi.org/10.1007/3-540-45014-9\\_1](https://doi.org/10.1007/3-540-45014-9_1)

Du, K.L. & Swamy, M.N.S. (2014). *Neural networks and statistical learning*. London: Springer. <https://doi.org/10.1007/978-1-4471-5571-3>

Duan, Z., Chang, Y., Wang, Q., Chen, T. & Zhao, Q. (2018). A logistic regression based auto insurance rate-making model designed for the insurance rate reform. *International Journal of Financial Studies*, 6: 18. <https://doi.org/10.3390/ijfs6010018>

Ducuroir, F., Zians, J. & Miller, A. (2016). *Lapse rate models in life insurance and a practical method to foresee interest rates dependencies*. Available at: [www.reactfin.com](http://www.reactfin.com) [Accessed on: 12 October 2020].

EIOPA. (2011). *EIOPA report on QIS5 for solvency II*. Available at: [https://register.eiopa.europa.eu/Publications/Reports/QIS5\\_Report\\_Final.pdf](https://register.eiopa.europa.eu/Publications/Reports/QIS5_Report_Final.pdf) [Accessed on: 20 October 2020].

Eling, M., & Kiesenbauer, D. (2011). *WHAT POLICY FEATURES DETERMINE LIFE INSURANCE LAPSE? AN ANALYSIS OF THE GERMAN MARKET* (No. 95). <https://www.ivw.unisg.ch/~media/internet/content/dateien/instituteundcenters/ivw/wps/wp95.pdf>

Eling, M. & Kochanski, M. (2012). Research on lapse in life insurance - What has been done and what needs to be done? *The Journal of Risk Finance*, 14: 392–418. <https://doi.org/10.1108/JRF-12-2012-0088>

Elreedy, D., & Atiya, A. F. (2019). A Novel Distribution Analysis for SMOTE Oversampling Method in Handling Class Imbalance. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11538 LNCS, 236–248. [https://doi.org/10.1007/978-3-030-22744-9\\_18](https://doi.org/10.1007/978-3-030-22744-9_18)

Fang, K., Jiang, Y. & Song, M. (2016). Customer profitability forecasting using Big Data analytics: A case study of the insurance industry. *Computers and Industrial Engineering*, 101. <https://doi.org/10.1016/j.cie.2016.09.011>

Financial Sector Conduct Authority. (2015). *Terminology: Life insurance quarterly conduct of business return (CBR 2015)*. Available at: [https://www.fsca.co.za/Regulatory Frameworks/Documents for Consultation/CBR Guidelines \(LT\) Final 10122015.pdf](https://www.fsca.co.za/Regulatory Frameworks/Documents for Consultation/CBR Guidelines (LT) Final 10122015.pdf) [Accessed on: 10 October 2020].

Feng, J. & Lu, S. (2019). Performance analysis of various activation functions in artificial neural networks. *Journal of Physics: Conference Series*, 1237(2). <https://doi.org/10.1088/1742-6596/1237/2/022030>



Fintechfutures. (2019). *The top trends that impacted the insurance industry in 2019*. Available at: <https://www.fintechfutures.com/2019/10/the-top-trends-that-impacted-the-insurance-industry-in-2019/> [Accessed on: 27 October 2020].

Galar, M., Fernandez, A., Barrenechea, E. & Sola, H.B. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches', *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 42(4): 463–484. <https://doi.org/10.1109/TSMCC.2011.2161285>

Gavrishchaka, V.V., Yang, Z., Miao, R. & Senyukova, O. (2018). Advantages of hybrid deep learning frameworks in applications with limited data. *International Journal of Machine Learning and Computing*, 8(6): 549–558. <https://doi.org/10.18178/ijmlc.2018.8.6.744>

Gelman, A. (2010). Missing-data imputation. *Data Analysis Using Regression and Multilevel/Hierarchical Models*, 529–544. <https://doi.org/10.1017/cbo9780511790942.031>

Genton, M.G. & Zhang, H.H. (2004). Compactly supported radial basis function kernels. *The Institute of Statistics Mimeo Series, North Carolina State University*, 2570: 1–22.

Geschiere, M. (2017). *Predicting the lapse rates of AllSecur* (Unpublished master's thesis). Erasmus University, Rotterdam. Available at: <https://thesis.eur.nl/pub/38375/Geschiere.pdf> [ Accessed on: 20 November 2020].

Girma, H. (2009). A tutorial on support vector regression. *Center of Experimental Mechanics*, 18. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>

Gill, J. (2001). *Generalized linear models: A unified approach*. Thousand Oaks, CA: Sage.

Goonetilleke, T.L.O. & Caldera, H.A. (2013). Mining life insurance data for customer attrition analysis. *Journal of Industrial and Intelligent Information*, 1(1): 52–58. <https://doi.org/10.12720/jiii.1.1.52-58>

Gunn, S. R. (1998). *Support Vector Machines for Classification and Regression*. University of Southampton. Faculty of Engineering, Science and Mathematics.

Gupta, S. & Gupta, A. (2019). Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161: 466–474.

<https://doi.org/10.1016/j.procs.2019.11.146>

Hassouna, M., Tarhini, A., Elyas, T. & Trab, M.S.A. (2015). Customer churn in mobile markets: A comparison of techniques. *International Business Research*, 8(6): 224–237.

<https://doi.org/10.5539/ibr.v8n6p224>

Henckaerts, R., Côté, M.-P., Antonio, K. & Verbelen, R. (2020). Boosting insights in insurance tariff plans with tree-based machine learning methods. *North American Actuarial Journal*, (April): 1–31. <https://doi.org/10.1080/10920277.2020.1745656>

Hendrych, R. (2019). *Modelling of life lapse rates*. Available at:

[https://www.actuaria.cz/uploads/files/news/id1009/aktuarsky\\_seminar\\_20191206.pdf](https://www.actuaria.cz/uploads/files/news/id1009/aktuarsky_seminar_20191206.pdf)

[Accessed on: 12 October 2020].

Hongsheng, L.I. (2021). *Introduction to deep learning convolutional neural networks*. Department of Electronic Engineering The Chinese University of Hong Kong

<http://dl.ee.cuhk.edu.hk/slides/cnn.pdf>

Hossain, M.M. & Miah, M.S. (2016). Evaluation of different SVM kernels for predicting customer churn. *2015 18th International Conference on Computer and Information Technology, ICCIT 2015*: 1–4. <https://doi.org/10.1109/ICCITechn.2015.7488032>

Hu, Z., Zhang, J. & Ge, Y. (2021). Handling vanishing gradient problem using artificial derivative. *IEEE Access*, 9: 22371–22377.

<https://doi.org/10.1109/ACCESS.2021.3054915>

Hudaib, A., Harfoushi, O., Dannoun, R. & Obiedat, R. (2015). Hybrid data mining models for predicting customer churn. *International Journal of Communications, Network and System Sciences*, 8(05): 91–96. <https://doi.org/10.4236/ijcns.2015.85012>

IAIS Consultation. (2019). *Cyber risk in the insurance sector report of the A2ii – IAIS consultation call*. Available at: <https://a2ii.org/en/knowledge-center/emerging-topics/cyber-risk-in-the-insurance-sector-a2iiiais-consultation-call> [Accessed on: 20 October 2020].

International Accounting Standard Board (IASB). (2017). *IFRS® standards effects analysis: IFRS 17 insurance contracts*. Available at: <https://www.ifs.org/-/media/project/insurance-contracts/ifs-standard/ifs-17-effects-analysis.pdf> [Accessed on: 13 October 2020].

Jaadi, Z. (2020). *A step by step explanation of principal component analysis*. Available at: <https://builtin.com/data-science/step-step-explanation-principal-component-analysis> [Accessed on: 9 October 2020].

Jansen van Vuuren, L., Reyers, M. & Van Schalkwyk, H. (2017). Assessing the impact of solvency assessment and management on risk management in South African insurance companies. *Southern African Business Review* (1997), 21(1): 129–149.

Hutagaol, J.B. & Mauritsius, T. (2020). Risk level prediction of life insurance applicant using machine learning. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(2): 2213–2220. <https://doi.org/10.30534/ijatcse/2020/199922020>

Kaggle. (2019). Life insurance data. Kaggle. Available at: <https://www.kaggle.com/blackclover1/life-insurance-policy-data/metadata> [Accessed on: 12 February 2020]

Khan, A.A., Jamwal, S. & Sepehri, M.M. (2010). Applying data mining to customer churn prediction in an internet service provider. *International Journal of Computer Applications*, 9(7): 8–14. <https://doi.org/10.5120/1400-1889>

Khan, M.R., Manoj, J., Singh, A. & Blumenstock, J.E. (2015). Behavioral modeling for churn prediction: Early indicators and accurate predictors of custom defection and loyalty. *Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015*,

107: 677–680. <https://doi.org/10.1109/BigDataCongress.2015> [Accessed on: 12 October 2020].

Kho, J. (2018). Discover the real world advantages and drawbacks of the random forest: *Towards Data Science*. Available at: <https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706> [Accessed on: 15 October 2020].

Kim, M.J., Min, S.H. & Han, I. (2006). An evolutionary approach to the combination of multiple classifiers to predict a stock price index. *Expert Systems with Applications*, 31(2): 241–247. <https://doi.org/10.1016/j.eswa.2005.09.020>

Kotsiantis, S., Kanellopoulos, D. & Pintelas, P. (2006). Handling imbalanced datasets : A review. *Science*, 30(1): 25–36. [https://doi.org/10.1007/978-0-387-09823-4\\_45](https://doi.org/10.1007/978-0-387-09823-4_45)

KPMG. (2019). *The South African insurance industry survey 2011*. Available at: <https://home.kpmg/content/dam/kpmg/za/pdf/south-african-insurance-survey-2019.pdf> [Accessed on: 10 October 2020].

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5(4): 221–232. <https://doi.org/10.1007/s13748-016-0094-0>

Krawczyk, B., Woźniak, M. & Schaefer, G. (2014). Cost-sensitive decision tree ensembles for effective imbalanced classification. *Applied Soft Computing Journal*, 14(PART C), 554–562. <https://doi.org/10.1016/j.asoc.2013.08.014>

Kunert, R. (2017). *SMOTE explained for noobs – Synthetic minority over-sampling technique line by line*. Available at: [https://rikunert.com/SMOTE\\_explained](https://rikunert.com/SMOTE_explained) [Accessed on: 21 September 2020].

Lake, B.M., Ullman, T.B., Tenenbaum, J.D. & Gershman, S.J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40(2012): 1–58. <https://doi.org/10.1017/S0140525X16001837>

Lessmann, S., Baesens, B., Seow, H.-V. & Thomas, L.C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1): 124–136. <https://doi.org/10.1016/j.ejor.2015.05.030>

Lin, H.-T. & Lin, C.-J. (2003). A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. *Neural Computation*. Available at: <https://www.researchgate.net/publication/2478380> [Accessed 10 July 2020].

Liu, Z. & Xu, H. (2013). Kernel parameter selection for support vector machine classification. *Journal of Algorithms & Computational Technology*, 8(2): 163.

Loisel, S., Piette, P. & Tsai, J. (2019). Applying economic measures to lapse risk management with machine learning approaches. Available at: <https://arxiv.org/abs/1906.050871> [Accessed on: 12 February 2021].

Luengo, J., Fernández, A., García, S. & Herrera, F. (2011). Addressing data complexity for imbalanced data sets: Analysis of SMOTE-based oversampling and evolutionary undersampling. *Soft Computing*, 15(10): 1909–1936. <https://doi.org/10.1007/s00500-010-0625-8>

Mauchant, D., Rice, K.D., Riley, M.A., Leber, D., Samarov, D. & Forster, A.L. (2011). *Analysis of three different regression models to estimate the ballistic performance of new and environmentally conditioned body armor*. Available at: <https://doi.org/10.6028/NIST.IR.7761> [Accessed on: 12 January 2021].

Madasamy, K. & Ramaswami, M. (2017). Data imbalance and classifiers: Impact and solutions from a big data perspective. *International Journal of Computational Intelligence Research*, 13(9): 2267–2281. Available at: <http://www.ripublication.com> [Accessed on: 10 October 2020].

- Madley-Dowd, P., Hughes, R., Tilling, K., & Heron, J. (2019). The proportion of missing data should not be used to guide decisions on multiple imputation. *Journal of Clinical Epidemiology*, 110, 63–73. <https://doi.org/10.1016/j.jclinepi.2019.02.016>
- Mahanta, J. (2017). Introduction to neural networks, advantages and applications. *Towards Data Science*. Available at: <https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207> [Accessed on: 20 October 2020].
- Maier, M., Carlotto, H., Sanchez, F., Balogun, S. & Merritt, S. (2019). Transforming underwriting in the life insurance industry. *31st AAAI Conference on Innovative Applications of Artificial Intelligence*: 9373–9380. <https://doi.org/10.1609/aaai.v33i01.33019373>
- Mandrekar, J.N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9): 1315–1316. <https://doi.org/10.1097/JTO.0b013e3181ec173d>
- Martínez, P.M. (2017). *Smart optimization of hyper-parameters in support vector machines*. Universitat de Barcelona. Available at: <https://upcommons.upc.edu/handle/2117/117926> [Accessed on: 10 October 2020].
- Marx, P. (2018, March 20). Benefits of buying life insurance in your 20s. [Online]. *Sanlam*. Available at: [https://www.sanlam.co.za/mediacentre/media-category/expert-opinions/Benefits of Buying Life Insurance in Your 20s](https://www.sanlam.co.za/mediacentre/media-category/expert-opinions/Benefits%20of%20Buying%20Life%20Insurance%20in%20Your%2020s) [Accessed on: 10 August 2020].
- Mashrur, A., Luo, W., Zaidi, N. & Robles-Kelly, A. (2020). Machine learning for financial risk management: A survey. *IEEE Access*, 8: 203203–203223. <https://doi.org/10.1109/ACCESS.2020.3036322>
- Maxwell, I.A. & Obinna, N. (2018). A comparative study of some variable selection techniques in logistic regression. *European Journal of Mathematics and Computer Science*, 5(1). Available at: [www.idpublications.org](http://www.idpublications.org) [Accessed 20 November 2020].

McDonald, C. (2017). Machine learning fundamentals (I): Cost functions and gradient descent. *Towards Data Science*. Available at: <https://towardsdatascience.com/machine-learning-fundamentals-via-linear-regression-41a5d11f5220> [Accessed on: 12 October 2020].

Mcinroy, B. (2016). *Abstract SMOTE And Performance Measures for Machine Learning Applied to Real-Time Bidding*.

Michorius, C.Z. (2011). *Modeling lapse rates: Investigating the variables that drive lapse rates*. Zeist, The Netherlands, Faculty of Management and Governance. Available at: <http://essay.utwente.nl/61317/> [Accessed on: 10 October 2020].

Mijwil, M.M. (2018). Artificial neural networks advantages and disadvantages. *Researchgate*, 2(1): 18. Available at: <https://www.researchgate.net/publication/323665827> [Accessed on: 10 October 2020].

Mingers, J. (1989). An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 243: 227–243.

Miškovic, V. (2014). Machine learning of hybrid classification models for decision support. *Sinteza 2014 - Impact of the Internet on Business Activities in Serbia and Worldwide*, Belgrade, Singidunum University, Serbia: 318–323. <https://doi.org/10.15308/sinteza-2014-318-323>

Mojekwu, J.N. (2011). Study of modes of exit of life-insurance policy holders in Nigeria: Trends and patterns. *International Business Research*, 4(3). <https://doi.org/10.5539/ibr.v4n3p182>

Morgan, B. (2018). Here's how IoT will impact the insurance claims process. *Forbes*. Available at: <https://www.forbes.com/sites/blakemorgan/2018/05/16/heres-how-iot-will-impact-the-insurance-claims-process/?sh=68b29cc5366e> [Accessed on: 28 October 2020].

Moulana, M. & Hussain, M. (2014). An implementation of optimal ID3 based decision tree

algorithm. *International Journal of Applied Engineering Research*, 9(12): 1935–1941.

Moon. (2019). Insurance . Available at: <https://www.kaggle.com/temmyzeus/zindi-zimnat-dataset/metadata> [Accessed on: 12 February 2020]

Nanda, M.A., Seminar, K.B., Nandika, D. & Maddu, A. (2018). A comparison study of kernel functions in the support vector machine and its application for termite detection. *Information (Switzerland)*, 9(1). <https://doi.org/10.3390/info9010005>

Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons*, 4(December): 51–62. <https://doi.org/10.20544/horizons.b.04.1.17>

Nazari, M. & Alidadi, M. (2013). Measuring credit risk of bank customers using artificial neural network. *Journal of Management Research*, 5(2): 17. <https://doi.org/10.5296/jmr.v5i2.2899>

Nelder, J.A. & Wedderburn, R.W.M. (1972). Generalized linear models. *Journal of the Royal Statistical Society*, 135(3): 370–384. Available at: <http://www.jstor.org/stable/2344614> [Accessed on: 10 October 2020].

Newsom, J.T. (2021). *Categorical data analysis for the behavioral and social sciences*. Spring 2021 Course Syllabus. [http://web.pdx.edu/~newsomj/cda/class/syllabus\\_21.pdf](http://web.pdx.edu/~newsomj/cda/class/syllabus_21.pdf) Accessed on: [01 December 2021]

Norazian, M.N. (2013). Roles of imputation methods for filling the missing values: A review. *Advances in Environmental Biology*, 7(12): 3861–3869. Available at: [https://www.researchgate.net/publication/259772975\\_Roles\\_of\\_Imputation\\_Methods\\_for\\_Filling\\_the\\_Missing\\_Values\\_A\\_Review](https://www.researchgate.net/publication/259772975_Roles_of_Imputation_Methods_for_Filling_the_Missing_Values_A_Review) [Accessed on: 12 October 2020].

OECD. (2020). *The impact of big data and artificial intelligence (AI) in the insurance sector*. Available at: <https://www.oecd.org/finance/The-Impact-Big-Data-AI-Insurance-Sector.pdf> [Accessed on: 15 October 2020].

Ogotu, N.W. (2012) *The impact of business environmental factors on marketing of*



*general insurance products in Kenya: A case of insurance companies in Nairobi* (Unpublished master's thesis). Kenyatta University, Kenyatta.

Outreville, J.F. (1990). Whole-life insurance lapse rates and the emergency fund hypothesis. *Insurance: Mathematics and Economics*, 9: 249–255.

Savas, C. & Dosis, F. (2019). The impact of different kernel functions on the performance of scintillation detection based on support vector machines. *Sensors (Switzerland)*, 19(23). <https://doi.org/10.3390/s19235219>

SAS. (2016). SAS ® Enterprise Miner. [https://documentation.sas.com/doc/en/emhpprcref/14.2/emhpprcref\\_hpforest\\_overview.htm#:~:text=The%20HPFOREST%20procedure%20is%20a,a%20target%20value%20from%20inputs](https://documentation.sas.com/doc/en/emhpprcref/14.2/emhpprcref_hpforest_overview.htm#:~:text=The%20HPFOREST%20procedure%20is%20a,a%20target%20value%20from%20inputs). [Accessed on: 01 December 2021]

Seemma, P., Nandhini, S. & Sowmiya, M. (2018). Overview Of Cyber Security. *Ijarccce*, 7(11): 125–128. <https://doi.org/10.17148/ijarccce.2018.71127>

Skilltohire. (2020). *Support Vector Machines*. <https://medium.com/@skilltohire/support-vector-machines-4d28a427ebd> [Accessed on: 01 December 2021].

Suksut, K., Kaoungku, N., Kerdprasop, N. & Kerdprasop, K. (2017). Parameter Optimization With Restarting Genetic Algorithm For The Forest Type Classification. *International Journal Of Machine Learning And Computing*, 7(6): 213–217. <https://doi.org/10.18178/ijmlc.2017.7.6.649>

Strike, K. D. (2001). Software Cost Estimation with Incomplete Data. *IEEE Transactions on Software Engineering*, 27, 890–908.

Panchal, F.S. & Panchal, M. (2014). Review On Methods Of Selecting Number Of Hidden Nodes In Artificial Neural Network. *International Journal Of Computer Science And Mobile Computing*, 3(11): 455–464. Available at: <https://www.ijcsmc.com/> [Accessed on: 15 October 2020].

- Park, H.A. (2013). An Introduction To Logistic Regression: From Basic Concepts To Interpretation With Particular Attention To Nursing Domain. *Journal Of Korean Academy Of Nursing*, 43(2): 154–164. <https://doi.org/10.4040/jkan.2013.43.2.154>
- Patil, K. (2018). A Survey On Machine Learning Techniques For Insurance Fraud Prediction. *Helix*, 8(6): 4358–4363. <https://doi.org/10.29042/2018-4358-4363>
- Patil, P. (2013). Tutorial On Decision Trees. *IJACKD Journal of Research*, 2(1): 32–43.
- Piegorsch, W.W. (1992). Complementary Log Regression For Generalized Linear Models. *The American Statistician*, 46(2).
- Peshawa J. Muhammad Ali, Rezhna H. Faraj (2014). Data Normalization and Standardization: A Technical Report, Machine Learning Technical Reports, 1(1), 1-6.
- Pohjalainen, V. (2016). *Predicting service contract churn with decision tree models*. Aalto University, School of Science. Available at: <https://aaltodoc.aalto.fi/handle/123456789/24444> [Accessed on: 15 October 2020].
- Pradipta, G. A., Wardoyo, R., Musdholifah, A., & Sanjaya, I. N. H. (2021). Radius-SMOTE: A New Oversampling Technique of Minority Samples Based on Radius Distance for Learning from Imbalanced Data. *IEEE Access*, 9, 74763–74777. <https://doi.org/10.1109/ACCESS.2021.3080316>
- Prasetyo, R.B., Kuswanto, H., Iriawan, N. & Ulama, B.S.S. (2019). A Comparison Of Some Link Functions For Binomial Regression Models With Application To School Drop-Out Rates In East Java. *AIP Conference Proceedings*, 2194. <https://doi.org/10.1063/1.5139815>
- Pukała, R. (2016). Use of neural networks in risk assessment and optimization of insurance cover in innovative enterprises. *Engineering Management in Production and Services*, 8(3): 43–56. <https://doi.org/10.1515/emj-2016-0023>
- PWC. (2017). *Contracts – Lessons learned to date*. Available at:

<https://www.pwc.com/id/en/publications/Actuarial/ifrs17-insurance-contracts.pdf>

[Accessed on: 12 October 2020].

PWC. (2020a). 'Banana skins' poll reflects industry risk perception. Available at: <https://www.pwc.co.za/en/press-room/cyber-risk-and-regulation-rank-as-top-risks-for-insurers.html#:~:text=Cyber risk was ranked as,on the combined global survey> [Accessed on: 27 October 2020].

PWC. (2020b). *IFRS 17 for insurers*. Available at: <https://www.pwc.com/gx/en/industries/financial-services/insurance/ifrs.html> [Accessed on: 28 October 2020].

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, 1(1): 81–106.

Quotacy. (2019). *The cost of life insurance: A guide to life insurance quotes*. Available at: [https://www.quotacy.com/ebooks/a\\_guide\\_to\\_life\\_insurance\\_pricing.pdf](https://www.quotacy.com/ebooks/a_guide_to_life_insurance_pricing.pdf) [Accessed on: 19 July 2020].

Raymond, M., & Roberts D.(1987). A Comparison of Methods for Treating Incomplete Data in Selection Research. In *Education and Psychological Measurement*, 47,3-26.

Ravichandran, S. & Ramasamy, C. (2016). Performance comparison of dimensionality reduction methods using MCDR. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, 3. Available at: [www.ijirae.com](http://www.ijirae.com) [Accessed 12 November 2020].

Rautio, A. (2019). *Churn prediction in saas using machine learning* (Unpublished master's thesis). Tampere University, Tampere. Available at: <https://trepo.tuni.fi/handle/123456789/27579> [Accessed on: 15 October 2020].

Rodan, A., Faris, H., Alsakran, J. & Al-Kadi, O. (2014). A support vector machine approach for churn prediction in telecom industry. *Information (Japan)*, 17(8): 3961–3970.

Rokach, L. & Maimon, O. (2014). *Data mining with decision trees theory and applications*. 2nd Edition. NJ: World Scientific Publishing Co. Available at: <https://doc.lagout.org/Others/Data Mining/Data Mining with Decision Trees Theory and Applications %282nd ed.%29 %5BRokach %26 Maimon 2014-10-23%5D.pdf>. [Accessed

on: 10 October 2020].

Roy, B. (2019). All about missing data handling. *Towards Data Science*. Available at: <https://towardsdatascience.com/all-about-missing-data-handling> [Accessed on: 10 October 2020].

Russell, D.T., Fier, S.G., Carson, J.M. & Dumm, R.E. (2013). An empirical analysis of life insurance policy surrender activity. *Journal of Insurance Issues*, 36(1): 35–57.

Rustam, Z. & Audia Ariantari, N.P.A. (2018). Support vector machines for classifying policyholders satisfactorily in automobile insurance. *Journal of Physics: Conference Series*, 1028(1). <https://doi.org/10.1088/1742-6596/1028/1/012005>

Sabbeh, S.F. (2018). Machine-learning techniques for customer retention: A comparative study. *International Journal of Advanced Computer Science and Applications*, 9(2): 273–281. <https://doi.org/10.14569/IJACSA.2018.090238>

Sakkaf, Y. (2020). Decision trees: ID3 algorithm explained. *Towards Data Science*. Available at: <https://towardsdatascience.com/decision-trees-for-classification-id3-algorithm-explained-89df76e72df1#:~:text=Invented by Ross Quinlan%2C ID3,moment to create a node> [Accessed on: 20 October 2020].

Santos, M.S., Soares, J.P., Abreu, P.H., Araujo, H. & Santos, J. (2018). Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches. *IEEE Computational Intelligence Magazine*, 13(4): 59–76. <https://doi.org/10.1109/MCI.2018.2866730>

Schafer JL (1999) Multiple imputation: a primer. *Stat Methods in Med* 8(1):3–15. doi:10.1177/096228029900800102

Schreiber-Gregory, D. & Bader, K. (2018). Logistic and linear regression assumptions: violation recognition and control. *Midwest SAS User Group*, (January): 1–21. Available at: [https://www.lexjansen.com/wuss/2018/130\\_Final\\_Paper\\_PDF.pdf](https://www.lexjansen.com/wuss/2018/130_Final_Paper_PDF.pdf)

Shao, J., Li, X. & Liu, W. (2007). The application of AdaBoost in customer churn

prediction. *Proceedings - ICSSSM'07: 2007 International Conference on Service Systems and Service Management*. <https://doi.org/10.1109/ICSSSM.2007.4280172>

Shaaban, E., Helmy, Y. & Khedr, A. (2012). A proposed churn prediction model. *Mona Nasr / International Journal of Engineering Research and Applications (IJERA)*. Available at: [www.ijera.com](http://www.ijera.com) [Accessed 21 July 2020]

Sheela, K.G. & Deepa, S.N. (2014). Selection of number of hidden neurons in neural networks in renewable energy systems. *Journal of Scientific and Industrial Research*, 73(10): 686–688.

Sibindi, A.B. (2014). Life insurance, financial development and economic growth in South Africa. *Risk Governance and Control: Financial Markets and Institutions*, 4(3): 7–15. <https://doi.org/10.22495/rgcv4i3art1>

Siemes, T. (2016). *Churn prediction models tested and evaluated in the Dutch indemnity industry* (Unpublished master's thesis). Open University of the Netherlands, Heerlen.

Sifa, R., Runge, S., Bauckhage, C. & Klapper, D. (2018). Customer lifetime value prediction in non-contractual freemium settings: Chasing High-value users using deep neural networks and SMOTE. *Proceedings of the 51st Hawaii International Conference on System Sciences*, 9: 923–932. <https://doi.org/10.24251/hicss.2018.115>

Singh, S. (2014). Comparative study ID3, CART and C4. 5 decision tree algorithm : A survey. *International Journal of Advanced Information Science and Technology*, 27(27): 97–103. Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.4929&rep=rep1&type=pdf> [Accessed on: 12 October 2020].

Soares, P.J., Santos, M.S., Abreu, P.H., Araujo, H.J., & Santos, J.A.M. (2018). Exploring the effects of data distribution in missing data imputation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: 251-263. Springer Verlag. doi:10.1007/978-3-030-01768-2\_21.

Goldburd, M., Khare, A., Tevet, D. & Guller, D. (2016). *Generalized linear models for insurance rating*. 2nd edition. CAS MONOGRAPH SERIES. Available at: <https://www.casact.org/pubs/monographs/index.cfm?fa=goldburd-Khare-Tevet-monograph05> [Accessed on: 12 February 2021].

Song, Q. & Shepperd, M. (2007). Missing data imputation techniques. *International Journal of Business Intelligence and Data Mining*, 2(3): 261–291. <https://doi.org/10.1504/IJBIDM.2007.015485>

Soofi, A. & Awan, A. (2017). Classification techniques in machine learning: Applications and issues. *Journal of Basic & Applied Sciences*, 13(September): 459–465. <https://doi.org/10.6000/1927-5129.2017.13.76>

Spedicato, G.A., Dutang, C. & Petrini, L. (2018). Machine learning methods to perform pricing optimization. A Comparison with standard GLMs. *Variance*, 12(1): 69–89.

Statinfer. (2019). *204.6.8 SVM: Advantages disadvantages and applications*. Available at: <https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/> [Accessed on: 28 October 2020].

Stoyanova, D. (2017). *ANN for optimization on large-scale structural acoustics models* (Unpublished master's thesis). Uppsala Universitet, Uppsala. Available at: <https://uu.diva-portal.org/smash/get/diva2:1158095/FULLTEXT01.pdf> [Accessed on: 20 October 2020].

Sur, P., Chen, Y., & Candès, E. J. (2019). The likelihood ratio test in high-dimensional logistic regression is asymptotically a rescaled Chi-square. *Probability Theory and Related Fields*, 175(1–2): 487–558. <https://doi.org/10.1007/s00440-018-00896-9>

Syarif, I., Prugel-Bennett, A. & Wills, G. (2016). SVM parameter optimization using grid search and genetic algorithm to improve classification performance. *Telkomnika (Telecommunication Computing Electronics and Control)*, 14(4): 1502–1509. doi: 10.12928/TELKOMNIKA.v14i4.3956

Szandala, T. (2020). Review and comparison of commonly used activation functions for

deep neural network. In *Bio-inspired neurocomputing, studies in computational intelligence*: 903. Edited by A.K. Bhoi et al. Singapore: Springer Nature. <https://doi.org/10.1007/978-981-15-5495-7>

Troles, J. (2016). A critical analysis of the ID3 algorithm and its successor C4.5. *Paper presented at Seminar AI: Past, Presence, Future Applied Informatics*, University of Bamberg. Available at: [https://cogsys.uni-bamberg.de/teaching/ws1718/sem\\_m2/JonasDarioTroles\\_ID3Analysis.pdf](https://cogsys.uni-bamberg.de/teaching/ws1718/sem_m2/JonasDarioTroles_ID3Analysis.pdf) [Accessed on: 10 October 2020].

Tsai, C.F. & Lu, Y.H. (2009). Customer churn prediction by hybrid neural networks. *Expert Systems with Applications*, 36(10): 12547–12553. <https://doi.org/10.1016/j.eswa.2009.05.032>

Vafeiadis, T., Diamantaras, K., Sarigiannidis, G. & Chatzisavvas, K. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, 55. <https://doi.org/10.1016/j.simpat.2015.03.003>

Vahedi, A. (2012). The predicting stock price using artificial neural network. *Journal of Basic and Applied Scientific Research*, 2(3): 2325–2328.

Vahidy Rodpysh, K. (2012). Model to predict the behavior of customers churn at the industry. *International Journal of Computer Applications*, 49(15): 12–16. <https://doi.org/10.5120/7702-1059>

Valdez, E.A., Vadiveloo, J. & Dias, U. (2014). Life insurance policy termination and survivorship. *Insurance: Mathematics and Economics*, 58: 138–149. <https://doi.org/10.1016/j.insmatheco.2014.06.011>

Vasudev, M., Bajaj, R., & Alegre Escolano, A. (2016). *Title: On the drivers of lapse rates in life insurance.* [http://diposit.ub.edu/dspace/bitstream/2445/115586/1/TFM-CAF\\_RahejaBajaj.pdf](http://diposit.ub.edu/dspace/bitstream/2445/115586/1/TFM-CAF_RahejaBajaj.pdf)

Vieira, S., Proença, H.M. & Salgado, C. (2016). Missing data. *In: Secondary analysis of*



*electronic health records*. Cham: Springer: 1–427. <https://doi.org/10.1007/978-3-319-43742-2>

Wan, S. & Yang, H. (2013). Comparison among methods of ensemble learning. *Proceedings - 2013 International Symposium on Biometrics and Security Technologies, ISBAST 2013*, July 2013: 286–290. <https://doi.org/10.1109/ISBAST.2013.50>

Wang, Y., Li, Y., Song, Y., Rong, X. & Shuaishuai, Z. (2017). Improvement of ID3 algorithm based on simplified information entropy and coordination degree. *Algorithms*, 10: 124. <https://doi.org/10.3390/a10040124>

Waseem, M. (2020, April 24). What is overfitting in machine learning and how to avoid it? [Online blog]. *Edureka*. Available at: <https://www.edureka.co/blog/overfitting-in-machine-learning/> [Accessed on: 13 September 2020].

Wilimitis, D. (2018, December 12). *The kernel trick in support vector classification towards data science*. Available at: <https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f> [Accessed on: 02 December 2021]

Williams, P., Li, S., Feng, J., & Wu, S. (2005). Scaling the kernel function to improve performance of the support vector machine. *Lecture Notes in Computer Science*, 3496(I), 831–836. [https://doi.org/10.1007/11427391\\_133](https://doi.org/10.1007/11427391_133)

Xie, W., Liang, G., Dong, Z., Tan, B. & Zhang, B. (2019). An improved oversampling algorithm based on the samples' selection strategy for classifying imbalanced data. *Mathematical Problems in Engineering*, 1–13. <https://doi.org/10.1155/2019/3526539>

Xong, L.J. & Kang, H.M. (2019). A comparison of classification models for life insurance lapse risk. *International Journal of Recent Technology and Engineering*, 7(5): 245–250.

Yang, N., Li, T. & Song, J. (2007). Construction of decision trees based entropy and rough sets under tolerance relation. *International Journal of Computational Intelligence Systems*, 1515–1519. <https://doi.org/10.2991/iske.2007.258>

Yekkehkhany, B., Safari, A., Homayouni, S., & Hasanlou, M. (2014). A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(2W3), 281–285. <https://doi.org/10.5194/isprsarchives-XL-2-W3-281-2014>

Yeoh, J. (2017). IFRS 17 insurance contracts: A brief history of IFRS 17. *IFRS 17 Workshop*, 14 September. Available from: <https://www.actuaries.org.uk/system/files/field/document/Joanna%20Yeoh.pdf> [Accessed on: 12 October ].

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2): 1–7. <https://doi.org/10.1088/1742-6596/1168/2/022022>

Young, D., Simon, L. & Pardoe, I. (2014). *Regression methods*. PennState, Elberly College of Science. Available from: <https://online.stat.psu.edu/stat501/lesson/welcome-stat-501> [Accessed on: 12 January 2021].

Zhang, Z., Beck, M.W., Winkler, D.A. Huang, B., Sibanda, W. & Goyal, H. (2018). Opening the black box of neural networks: Methods for interpreting neural network models in clinical applications. *Annals of Translational Medicine*, 6(11): 216–216. <https://doi.org/10.21037/atm.2018.05.32>