UNIVERSITY OF CAPE TOWN

Applying Imputation and Statistical Learning to Predict Gamma-glutamyl Transferase in Underwriting Data

Student:
Yevashan Perumal
PRMYEV001

Supervisor:
Mr Stefan S Britz

Minor dissertation in partial fulfilment of the degree
M.Sc. specialising in Data Science

DEPARTMENT OF STATISTICAL SCIENCES

July 18, 2023

Abstract

Insurance underwriting can be time-consuming and costly for both insurers and customers. However, the insight gained is of critical importance in addressing the information asymmetry between insurers and customers in terms of establishing a customer's risk profile. Consequently, any test that assists in providing a risk assessment is critical in allowing insurance companies to manage risk and price their products appropriately. Gamma-glutamyl Transferase (GGT) is an enzyme which has been used by insurers in underwriting medical tests as an indicator of potential adverse outcomes. However, due to complexities such as differing underwriting strategies, data collection and data storage issues, not every customer on an insurer's books will have a GGT value or even a complete data profile. This research investigates if statistical techniques such as imputation and supervised learning can be used in conjunction with available medical, demographic, underwriting and policy data to accurately predict GGT values. A combination of multivariate imputation by chained equations (MICE) and extreme-gradient boosted trees (XGBoost) offers a 31% improvement in accuracy compared to a naïve prediction. However, there does appear to be a limit to the performance achieved from all implemented techniques with the analysed dataset, with various model combinations yielding root mean squared error (RMSE) values within a narrow range. In addition, when comparing the predictions from a separate, unlabelled dataset to actual data, it appears as though predictions from the models cannot be reliably deemed to be from the same distribution. This indicates that further research is required before insurers can reliably switch out blood-work based GGT results for those from a supervised learning model.

Keywords: insurance, underwriting, gamma-glutamyl transferase, imputation, supervised learning

Acknowledgements

Firstly, I would like to express my deepest gratitude to Stefan Britz for being a fantastic supervisor. Your guidance, patience and unflappable confidence that we would get this across the line helped me keep going even when my belief wavered. The cricket banter at our check-ins was an unexpected, but very welcome, bonus.

Secondly, I am extremely grateful to INSETA and Old Mutual for offering me the opportunity to further my education. In particular, I would like to thank my manager, Carey-Anne Foulds, for her support from both a professional and personal front. You have set the bar for managers everywhere who have people pursuing studies in their team.

Last, but far from least, I would like to thank everyone in my support structure. To my girlfriend, my family and my friends: thank you for your patience, encouragement and support. You helped keep me (mostly) sane during some of the toughest times I have experienced. I look forward to making up for the moments we missed over the last two years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

Managing risk and ensuring the ongoing financial health of their business is critical for insurance companies. In particular, mitigating risks when issuing policies is vital as it requires managing expected inflows and outflows of related cash.

The use of medical underwriting is commonplace in the modern life and health insurance industry to assist in managing risk. By evaluating a customer's health using medical tests, insurers aim to provide for customers showing markers linked to negative outcomes by either outright rejection of their application or charging higher premiums on their contract to compensate for the elevated risk of the policy being claimed and needing to be paid out.

A variety of medical data can be used in the underwriting process such as blood sugar, smoking indicators, and body mass index (BMI). Gamma-glutamyl Transferase (GGT) tests can be used to ascertain liver damage due to alcohol abuse and raised GGT levels are linked to higher all-cause mortality (Koenig and Seneff, 2015). A major life insurer may not require all customers to go for a full panel of medical tests when applying for a new policy. This may be due to a variety of reasons: firstly, medical tests can be costly, and they must be borne by the insurer or customer. To minimise this, insurers have underwriting strategies that only require specific medical tests for certain customers. Secondly, awaiting the results of tests that are not instantly available may negatively affect customer experience, which companies are eager to avoid. Thirdly, more comprehensive medical testing may require visits to specialised practitioners, such as pathologists, adding further delays and costs. Lastly, strategic decisions may not require medical underwriting under certain criteria. For example, select customers may be exempt if another recently purchased product had required medical tests to be completed.

The details above provide the context in which this research is set. We are now able to define the research objective in the next section.

1.2 Research Objective

The consequence of not sending every customer for all available medical tests is an incomplete test result profile for certain customers in the database. This is particularly the case for tests that are more expensive and require greater effort to conduct, such as the GGT test. Additionally, the complex nature of customer data storage and collection policies at large corporations adds to the difficulty in obtaining complete and trustworthy customer data.

The research aims to investigate how accurately statistical techniques can predict GGT medical test results values using the existing demographic, policy, and biomedical data.

The main approaches this research uses to achieve this objective are imputation and supervised learning. This allows us to investigate sub-goals for this research such as:

1. Which dataset yields the best results: imputed or complete-case? If imputed, does the inclusion of features with a high proportion of missing values affect performance?
2. Which imputation method yields the best result?
3. Which supervised learning algorithm yields the best results?
4. Which combination of data handling technique and supervised learning algorithm yielded the best results?

Comparing the predicted GGT values from the models built on the different datasets to actual data allows an assessment of the most accurate design. This research focuses on prediction accuracy and not inference. The techniques we use are agnostic to the underlying physiological data, intending to avoid domain-specific medical techniques and knowledge. Root mean squared error (RMSE) is used to evaluate model accuracy, which Section 5.1 discusses in detail.

With the overall research objective and sub-goals clarified, the next section examines the potential benefits of the research.

1.3 Benefits of this research

There are numerous benefits to be gained by an insurer reliably predicting GGT results. Firstly, if a model can perform with high confidence in predicting GGT results, only the cases of the highest concern and uncertainty need to be sent for testing. This could reduce costs for the business and customers while improving the customer experience when purchasing a policy.

Secondly, actuarial analysts at a life insurer compare the results of medical tests in their underwriting data to mortality and morbidity experience to gauge their effectiveness. However, in the current scenario insurers have an incomplete picture with missing values created due to the reasons mentioned above. A more complete analysis and improved insights can be generated with a completed dataset. This information feeds into the pricing and underwriting strategy, as described in Section 2.2

Lastly, having a robust underwriting strategy is a competitive advantage for an insurer and can be a differentiating factor in a competitive market. The application of modern machine learning techniques can be used to enhance an underwriting strategy in a landscape where other financial services firms have moved into the life insurance space over the last few years. For example, Venter (2015) describes how in 2015 a large South African bank introduced the first predictive underwriting solution in Africa within their credit life business. This solution utilises existing customer data and a mere 3 questions as inputs to the predictive model, which then provides an underwriting decision.

A different approach to the use of technology in a medical scenario is that of wearable devices. The plethora of sensors providing data in modern wearable technology can be used to provide a more holistic view of an individual's well-being. Machine learning methods can be used to turn this easily available data into predictions on a wearer's health, alleviating the need (to a degree) for customers to undergo medical tests (McKeon, 2021). This research explores modern techniques not traditionally used in the medical insurance field and looks towards potentially modernising parts of the underwriting process in an industry that is generally slow to adopt newer technologies (Wang, 2021).

With the potential benefits discussed above, we now provide an overview of the research structure and summarise how it attempts to answer the research objective.

1.4 Outline

Chapter 2: Literature Review

The literature review examines life insurance at a high level with a focus on how underwriting fits in. Specifically, it highlights how medical tests fit into the process of assessing customer risk and adjusting pricing. We then provide detail on GGT and its usage in the medical and insurance field. The focus shifts to discussing imputation, highlighting its application with medical data. Finally, we examine how machine learning is used in the medical and insurance industries, including notable use cases relevant to this research.

Chapter 3: Data

This chapter describes the data we use for this research. This includes where the data was sourced, the grain of detail and which period it covers. We describe steps taken to explore the data, highlighting particular challenges uncovered. This is not only to become familiar with the dataset but is also an essential step in identifying any preprocessing required before the methodology is applied.

Chapter 4: Methodology

Chapter 4 begins by describing the overall architecture of the experimental design. The following sections detail the imputation methods used, the creation of training and test datasets, and finally the supervised learning methods applied to the various datasets. We briefly discuss how they work intuitively, their benefits and their application.

Chapter 5: Application and Results

This chapter explores how well the various methods achieved the research objective. As a regression problem, absolute performance is difficult to ascertain, therefore relative performance to other techniques is the main focus. However, we provide a comparison to an analogous statistical measure to set a baseline. Furthermore, the distribution of predicted GGT results on unlabelled data is compared to the distribution of actual values to determine if predictions appear to have similar distributions.

Chapter 6: Conclusion and Recommendations

The final chapter reflects on how well the research objective was achieved, and whether it warrants practical usage in a business context. Additionally, recommendations are made on future avenues that can be explored from a research perspective.

Chapter 2

Literature Review

This literature review covers two high-level themes. The first theme provides a brief review of life insurance and underwriting, focusing on the broad mechanics of life insurance and how underwriting is used as a tool in the process. This includes how medical tests are used in underwriting, and specifically how GGT tests fit in. This section is intended to contextualise the research and highlight its potential value. It is not intended to provide a comprehensive description of the various components of life insurance.

The second theme relates to the usage of statistical imputation and machine learning techniques in medical and life insurance settings. In addition to providing a foundation for this research, it gives potential guidance for which techniques may prove most effective in achieving the research objective.

2.1 Life Insurance

As defined by Fontinelle (2021), life insurance at its core is a legal contract between an insurance company and the holder of a policy. The contract is structured such that the insurance company will pay out a sum of money to a designated beneficiary when the insured party in the contract dies; this is in exchange for regular premiums paid by the policy owner.

Insurance enables people to be protected from ruinous financial risk without having to save funds themselves to cover substantial expenses (Cornell et al., 2016). Customers typically purchase life insurance to provide financial security for dependants they leave behind (Mishra and Mishra, 2011). Modern life insurers have several products that increase the range of adverse events to be insured against. Examples of these include products catering for severe illness (e.g. a cancer diagnosis) or disability (Munro and Snyman, 1995).

With the definition of life insurance detailed above, we now discuss how their premiums are set.

2.1.1 Setting of Premiums

To guarantee a claim is paid out, insurance companies use the concept of risk pooling. This entails numerous individuals each making relatively small contributions to a pool, which enables the pool to pay out a large amount to the few participants that do claim a benefit (Davies and Carrin, 2001).

Modern life insurance requires participants in the risk pool to pay a premium that relates to the level of risk they are bringing in, i.e. high-risk customers are asked to pay larger premiums (Pokorski, 1997). In addition to sufficiently funding any claims, the premium must also cover overhead costs and provide a profit for the insurance company.

Given each person should contribute accordingly to their level of risk, the extent to which low-risk customers (who claim less frequently) are subsidising risky customers is minimised. This illustrates the need for an accurate customer risk profile at the sale stage. If this does not occur it enables high-risk customers to purchase too much insurance for too low a cost, misaligned with the risk they are bringing to the pool; this is known as adverse selection (Pokorski, 1997). If adverse selection occurs frequently, insurers will increase premiums to deal with the higher-than-expected claims, making them less competitive for low-risk customers (Pauly and Nicholson, 1999).

However, ignoring any fraudulent non-disclosure by a customer (e.g., an applicant contemplating suicide), there is still an information asymmetry. Customers know more about their health than an insurer when taking up a policy. The process of assessing the risk of a customer in life insurance to address the information asymmetry is known as underwriting and is discussed further in the next section.

2.2 The Purpose of Underwriting

In general, underwriting is the process of accepting a financial responsibility for a fee (Banton, 2022). Maier et al. (2019) describes underwriting in the context of life insurance as calculating an estimate of mortality risk, otherwise known as the expected lifetime of an individual. Actuaries use this to model the cost of covering the mortality risk over a customer's lifetime, which is then transformed into regular premium payments. Performing this risk assessment accurately ensures life insurance policies are priced competitively and profitably (Maier et al., 2019).

Not all insurance products require underwriting; those that ignore a customer's risk profile from a pricing perspective are described as having limited underwriting (Marais, 2019). This research focuses on fully underwritten products, where customer-specific risk factors impact the policy premium. Within fully underwritten products there exists two sub-categories: general and individual underwriting.

General underwriting uses risk factors to allocate a customer to an appropriate risk pool. Criteria often used in general underwriting include "age, gender, medical and family histories, avocation and lifestyle" (Caplan, 2004). These criteria are used to categorise customers into standard risk classifications, where a premium rate can be calculated for that group. An example of such classification is a 49-year-old, non-smoking, female who pays a R200 monthly premium for their life insurance (Marais, 2019). This premium will be the same for all customers that fall under the same description.

Individual underwriting is discussed below, in particular highlighting the place of medical tests.

2.3 Usage of Medical Tests in Underwriting

According to Dodge (2007), "[m]edical underwriting involves the science of evaluating medical information to determine the risk for groups of individuals with various medical conditions." In lay terms, insurance products such as life, health and disability use medical information to help assess the risk of a future claim on a policy.

To determine the health of an applicant a medical questionnaire is completed and a medical examination may be required in the case of large sums insured. The type and number of tests required (i.e. the underwriting strategy) vary from insurer to insurer.

According to Marais (2019), common risk factors examined are "body weight, blood pressure, cholesterol levels, evidence of any medical condition (like heart problems, diabetes, being HIV positive) and family history of specific diseases". If these factors deem a customer to be at higher risk than their standard risk group, a loading can be applied to their premium to compensate. These loadings range from 100% to 250% of the standard premium rate for that risk group. In other words, a loading of 100% would mean the standard premium would then be doubled. Dodge (2007) describes alternative options to loadings such as increasing the elimination period, reducing the coverage amount granted or benefit period, or adding ad-hoc criteria that exclude certain medical conditions from coverage.

Bronsema et al. (2015) further illustrate how medical tests add value to the underwriting process. Their research seeks to investigate the prognostic value of data retrieved from patient health declaration forms and medical test results relating to extra mortality. Features such as blood pressure, lipids, cotinine and glucose levels were some of the variables examined using logistic regression models. The study showed that for life insurance applicants, the body mass index (BMI) followed by the overall assessment of the health declaration were the dominant variables to discriminate between those with and without at least 25% extra mortality.

Proponents of medical underwriting argue it ensures individual premiums are kept as low as possible (Lawson et al., 1999). Conversely, Wang (2021) states that "[t]raditional underwriting is costly, time-consuming and perceived as a barrier for the underserved population". This indicates efforts to reduce the onerous nature of medical underwriting could be beneficial to both insurers and the customers they serve.

The discussion above shows how medical tests have an established position in underwriting. We now provide detail on the key subject of this research, the GGT enzyme.

2.3.1 Usage of Gamma-glutamyl Transferase

This section provides a brief description of GGT and its relationship to predicting overall health. This in turn underlines its value to the risk assessment described in the medical underwriting process above.

According to Kunutsor (2016), GGT is a liver enzyme which has a primary role in the biological processes that help protect cells against oxidants produced during normal metabolism. They go on to state that beyond this, circulating serum GGT has been linked to a wide range of chronic conditions and diseases. These include "non-alcoholic

fatty liver disease, vascular and non-vascular diseases and mortality outcomes".

Pinkham and Krause (2009) describe how "[g]amma-glutamyltransferase (GGT) has been shown to be a marker for metabolic syndrome and diabetes, cardiovascular diseases and mortality, chronic kidney disease, as well as cancer incidence, and liver disease". Using underwriting application data from a life insurer, the study goes on to confirm that when GGT is elevated, mortality risk is significantly increased.

Koenig and Seneff (2015) concurs with these results, stating elevated GGT as a serum marker is linked to an "increased risk to a multitude of diseases and conditions, including cardiovascular disease, diabetes, metabolic syndrome (MetS), and all-cause mortality". Furthermore, when the authors examined literature from several population groups throughout the world, the predictive power of GGT tests remain valid, even taking into account varying ethnic and gender categories.

The research above points to the valuable information on an individual's health risks that can be gleaned from GGT, and the value to be gained by having a robust predictive model of it. This lends itself to the objective of this research.

The writing above establishes the context of life insurance and underwriting, highlighting how having GGT is of immense value to insurance companies. As mentioned in Section 1.1 and 1.2, there are several reasons for GGT values to be missing from an underwriting dataset. One potential method of dealing with missing data entries is imputation, which we discuss in the following section.

2.4 Imputation

Missing data points are a common occurrence when dealing with data, and manifest for a variety of reasons and to differing degrees. Having complete data are critical to producing reliable analyses or accurate predictive models (Zhang et al., 2020). Potential causes of missing values include limited budget, high complexity in the experimental setting, human error in the data capturing and participant withdrawal (Miok et al., 2020). Imputation deals with filling in these missing values.

An important concept in imputation relates to the pattern according to which the data are missing, as this guides which techniques may be appropriate. The three types of ways data can be missing are: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR).

Miok et al. (2020) states that for a value to be considered MCAR it is "independent with respect to possible values of the variable and with respect to other observable variables in the data". In other words, no relationship exists between missing data and any other values observed or unobserved in the data, and there is no discernible pattern to the missing data. van Buuren (2018) states that although MCAR is a convenient scenario, it can often be unrealistic when using real-world data.

MAR is more general than MCAR, and many modern imputation methods start from this assumption (van Buuren, 2018). With MAR, a relationship exists between the missing data in a particular feature and observed values in other features. van Buuren (2018)

provides an example of a weighing scale that may produce more missing values when placed on a soft surface compared to a hard surface. These are not MCAR; however, since we know the surface type we can assume MCAR within those groups. Overall the data in this scenario are MAR. Lastly, if a value is not MCAR or MAR, then it is MNAR. More specifically, MNAR describes where the pattern of missing value is not random and cannot be predicted from other values in the data; the reason behind missing data may be directly related to the value of the information requested (Bennett, 2001).

A simple and quick method of dealing with missing data is list-wise deletion or complete-case analysis. This method deletes the entire record if any single value is missing. However, this method risks losing valuable data and potentially creating bias. This is usually recommended only if a small portion of cases in the data contain missing values (Schafer and Graham, 2002).

2.4.1 Single Imputation

Alternatively to list-wise deletion/complete-case analysis, imputation can be used to fill in missing data points. Examples of basic and commonly used approaches include mean/median/mode imputation, using the last observation carried forward and imputation based on logical rules (van Buuren, 2018).

An issue with the methods above is that they do not take into account relationships between variables. Regression imputation addresses this by building a regression model on available data to predict the missing values. This can be extended to stochastic regression which adds a degree of variability to imputations, but this may result in a high degree of noise being introduced to the data. However, this does not address the bigger issue created by single imputation - that of the additional uncertainty associated with missingness being ignored.

2.4.2 Multiple Imputation

To address the issues mentioned above, Rubin (1978) developed a method for averaging the outcomes across multiple imputed datasets to account for this. The multiple imputation method developed follows three steps: imputation, analysis and pooling. The imputation step begins by choosing the number of completed datasets to be created. In each copy of the dataset the missing values are completed using an imputation procedure. This results in several completed datasets that differ in the values imputed for missing data. The analyses (e.g. linear regression) can be performed on multiple completed datasets separately, with the results (such as the coefficients) combined in the final step. One of the most popular multiple imputation methods is multivariate imputation by chained equations (MICE) (Raghunathan et al., 2001). MICE is one of the key methods used in this research and is discussed in further detail in Section 4.2.2.

2.4.3 Imputation Usage in Medical Data

In the medical field, decisions based on incomplete data could have material consequences. When dealing with electronic health records from laboratory data, missing data could result in poor performance or bias from predictive models (Li et al., 2021).

The research by Li et al. (2021) goes on to investigate how two advanced imputation methods, namely Monotone Multiple Imputation and Multivariate Fully Conditional Specification Imputation, perform on two datasets related to stroke and heart failure respectively. Their investigation into patterns of missing values reveals missing data was not at random and highly correlated with a patient's co-morbidity information, highlighting the added difficulty when dealing with missing medical data. However, the authors were able to conclude the "multi-level imputation algorithm showed smaller imputation error than the cross-sectional method." This research highlights the pitfalls and potential benefits imputation represents when dealing with medical data.

Miok et al. (2020) propose a novel approach to improving imputation performance for biomedical data by leveraging developments in deep learning. The paper highlights potential challenges facing current multiple imputation methods such as correct model specification, assumptions regarding the distribution of data and inability to deal with mixed data types. The authors propose using a Monte Carlo Dropout (MCD) method within a variational auto-encoder. The technique can also handle large amounts of data, non-linear relationships and complex variable interactions such as those present in medical data; these are issues which commonly plague classical multiple imputation methods. Their overarching aim is to generate multiple outcomes for a missing value using de-noising auto-encoders and combining these results using the MCD method. When tested on several publicly available datasets, the variational auto-encoder with MCD showed improvements in imputation error compared to simple auto-encoder techniques.

Imputation is a powerful technique, with applications displayed in a wide range of fields. Medical data in particular seems to have valuable use cases as missing data are a common issue in many medical surveys. Another powerful technique used in the medical field is machine learning. The methodology in Section 4.4 goes into detail regarding the subset of machine learning techniques used in this research. The next section seeks to provide the foundation for their usage.

2.5 Machine Learning

Machine learning is the process used to describe training models with algorithms that learn from the data itself. It has existed for several decades but gained traction in recent years with the vast explosion of data generated via the internet and huge gains in computational power (Brownlee, 2020). The combination of these factors allows algorithms to find patterns in huge datasets that no human could.

One of the main branches of machine learning is supervised learning. This involves having algorithms learn patterns from data using several feature variables in an attempt to predict a known target variable Hastie et al. (2009). Examples of this are models that attempt to predict the price of housing using information such as area, plot size and volume of nearby sales. It is also used in classification problems where the aim is predicting the probability of an event occurring, such as how likely a customer is to purchase a product. Using the data available, a supervised learning model could be trained to predict the value of a GGT test.

2.5.1 Use Cases of Machine Learning in the Medical Field

The use of machine learning in a medical context is by no means novel. Magoulas and Prentza (2001) describes a few potential applications:

- The prediction of disease progression
- Use of decision trees to develop a system to interpret electrocardiograms
- Computer-based medical image interpretation systems to assist in the diagnosis of conditions
- Effective monitoring and alarming of the continuous data generated in an intensive care unit

Machine learning's value is highlighted in a study performed by Holfinger et al. (2022). The paper investigates how a machine learning-based tool using readily available data compares to traditional patient survey tools in predicting occurrences of Obstructive Sleep Apnoea (OSA). The study utilises supervised learning techniques such as random forests, support vector machines and artificial neural networks. These in turn are contrasted with a simple logistic regression model and the traditionally used STOP-BANG patient questionnaire tool. The study used features such as age, BMI, ethnicity and sex as inputs. When measured on the area-under-the-curve (AUC) metric, the Artificial Neural Network (ANN) model was shown to outperform logistic regression and had similar performance to patient-reported STOP-BANG. This may allow for simplified and widespread identification of OSA based on routinely collected information.

Zhang et al. (2020) utilised supervised learning as part of their paper to predict the missing values in a medical dataset. The research tests a unique approach to dealing with missing values by initially using an unsupervised learning technique to fill in the existing data, followed by supervised learning in the form of extreme gradient boosting (XGBoost). The authors hypothesise that an appropriate pre-filling strategy can boost model performance. This high-level structure is similar to the one we describe in Section 4.1 in terms of using a statistical technique to fill in missing values and potentially boost the performance of trained models. The proposed design in Zhang et al. (2020) showed a 20% improvement on average in normalised root mean square deviation (nRMSD) when compared to other state-of-the-art techniques such as 3D-MICE.

2.5.2 A Use Case of Machine Learning in Life Insurance

The emergence of machine learning as a paradigm has resulted in many studies within the life insurance industry on whether its application in various facets could be beneficial compared to traditional methods.

Resisting attempts to automate the underwriting process, life and health insurance continue to use rule-based and traditional underwriting practices. Noting this, Wang (2021) proposes an end-to-end underwriting solution using machine learning techniques to overcome the issues of cumbersome, expensive and potentially ill-suited traditional underwriting practices. The paper highlights the usage of Natural Language Processing (NLP) and clustering techniques in the claim application phase to help categorise claim types and reduce errors. However, for our purposes, the section dedicated to supervised learning being used to create a final underwriting decision (deny, accept,

apply a specific loading) is most encouraging.

A wide variety of popular open-source supervised learning algorithms were compared, with results indicating XGBoost (with a combination of feature selection methods applied) performs best, achieving a 71% accuracy on the test dataset. Compared with the roughly 33% of applications that are successfully processed by rules-based systems, this is a major improvement. Additionally, the feature importance information extracted from the algorithm provides insights into the drivers behind underwriting decisions. These results indicate great potential for the underwriting process to be improved upon significantly (Wang, 2021). Additionally, the study highlights XGBoost's efficacy in an insurance underwriting context when compared to other widely used algorithms.

2.6 Conclusion

This chapter provides information on life insurance and the underwriting process, and how medical tests fit in. It then goes on to briefly discuss imputation and machine learning, illustrating their use cases within the medical and insurance industries. This serves to provide further context and understanding as to why this research is taking place and provides guidance for potential techniques that may be successful. However, before using any techniques we must consider the data we use in this research, which we discuss in the next chapter.

Chapter 3

Data

This chapter explores the data we use in this research. It describes where the data are sourced, followed by a description of select key explorations needed to get a better understanding of the data. This informs the pre-processing steps we take before applying the methodology in Chapter 4. We also provide the actions we take during pre-processing and the reasoning behind them. Finally, it describes the final output of the pre-processing step.

3.1 Data Sourcing

The data are sourced from a large South African life insurer. The data extracted are a portion of a historical underwriting portfolio for a closed book of business that operated between 2014 and 2019. The data extract focuses on keeping as many medical fields as complete as possible, i.e. with few missing values. This aligns with the research objective of using quick, easily available data to predict the more time-consuming expensive GGT results. Given the sensitive nature of the data, utmost care was taken to completely anonymise the data and remove any personally identifiable information.

The dataset contains four overarching groups of features:

1. **Biomedical Data**

   These features consist of results from various medical tests performed during underwriting. They include data such as cholesterol, blood pressure, blood sugar and cotinine values; the latter is used to determine smoker status. Additionally, information such as height and weight are recorded to calculate BMI, both from customer declarations and from a medical practitioner.

2. **Underwriting Information**

   This includes information used in the underwriting process that may inform the customer risk pool and subsequent policy pricing. This includes information such as education levels, income bands and occupation class.

3. **Customer Profile**

   These features consist of customer demographic information such as age, gender and location. This type of data are often used in customer analytics to provide insight into a company's base.

4. **Policy Level**

   The policy level data consists of the customer's product holdings at the insurer. This includes premium values, the number of policies held, how many active benefits are on a policy and the total value of assets managed by the insurer on behalf of the customer.

This brief description of the metadata surrounding the extract provides good context to explore the data in the following section.

3.2 Exploratory Data Analysis and Pre-processing

An exploratory data analysis (EDA) is a key component of the data science process. It allows a better understanding of the dataset's nature and informs data wrangling and modelling decisions. Additionally, the supervised learning methods we use in Section 4.4 require their input data to be in specific formats. The quality of output produced by these models often relies on the quality of the data being input. As such, to ensure high input data quality, a thorough EDA is often paired with several pre-processing actions to address identified issues that may negatively affect supervised learning methods. For example, these can include data type conversions and the removal of duplicate entries.

Below we describe important aspects of the EDA for this research. Additionally, it details how a particular issue is treated in the pre-processing step. This process explores each variable individually, viewing the variable's distribution and applying domain knowledge to ascertain the suitability of the variable for this research. The initial, unprocessed dataset sourced contains **349 533** observations of **83** variables/features.

3.2.1 Feature Selection

Feature selection is an extremely important aspect of ensuring data quality for this research. As mentioned above, each variable is explored individually to investigate its distribution. This exploration assists in identifying potentially unwanted features such as those that are zero-variance or simply empty.

It is at this juncture that the application of domain knowledge is crucial in deciding which variables to remove and those that should remain. Unusable features such as 'policy quote numbers' or 'duration of the underwriting process' are removed. Furthermore, we drop features that may cause target leakage in a supervised learning context, such as cover loading percentage, as they are informed by the GGT value.

In addition to univariate exploration, we examine the relationships between variables. This assists in highlighting variables that are too strongly correlated and may negatively affect the model. An example of this in the dataset occurred with height, weight and BMI. As BMI is a calculation involving height and weight it is easy to see why this occurs and requires a choice on which features to retain.

In terms of the pre-processing steps taken, feature selection decisions result in columns being omitted for the reasoning mentioned above. Additionally, we exclude nominal variables with many categories (>10) as they create computational issues during imputation in Section 4.2. The 30 remaining fields are listed in Table 3.1.

TABLE 3.1: Field names, descriptions and data Type

Field Name	**Description**	**Data Type**
AcceleratorIndicator	Indicates if a policy contains an accelerator type benefit	Binary Indicator
AGE	Customer age	Numeric (integer)
ALL_BEN_COUNT	Number of active benefits on all policies	Numeric (integer)
ALL_CONTRACT_COUNT	Number of policies per customer	Numeric (integer)
ANNUAL_INVESTMENT	Total annual premium contributions by customer	Numeric (float)
ASS_ETHNIC_GROUP	Customer ethnicity	Categorical (nominal)
AUM	Total value of assets under management	Numeric (float)
AutoDeferredIndicator	Indicates if underwriting was automatically deferred	Binary Indicator
BloodSugarValue	Blood Sugar Value	Numeric (float)
CholesterolValue	Cholesterol Value	Numeric (float)
CoverAmount	Cover amount of insurance product	Numeric (integer)
DiastolicBloodPressureValue	Diastolic blood pressure value	Numeric (integer)
DisabilityIncomeClass	Classifies degree of disability	Categorical (ordinal)
DoctorBMIValue	Customer body-mass index, verified by a doctor	Numeric (float)
EducationID_Override	Customer education level	Categorical (ordinal)
EntryANB	Risk rating measure	Numeric (integer)
ExclusionIndicator	Indicates if the customer qualifies for exclusions	Binary Indicator
FARMER_IND	Indicates if the customer is a farmer	Binary Indicator
GammaGTValue	Gamma-glutamyl Transferase value (Target Variable)	Numeric (integer)
GENDER	Customer gender	Binary Indicator
HeightValue	Customer height	Numeric (integer)
IRPCode	Socio-economic classification	Numeric (integer)
LOAIndicator	Life office association customer to indicate fraud/unsuitable customer	Categorical (nominal)
MonthlyIncome_Override	Customer monthly income, validated where possible	Numeric (integer)
OccupationClassCode	Occupation level classification	Categorical (ordinal)
PROF_MARKET_IND	Indicates if customer works in a classical profession	Binary Indicator
PROVINCE	Customer's Residential Province	Categorical (nominal)
PUBLIC_SECTOR_IND	Indicates if customer works for a government institution	Binary Indicator
RestingPulseValue	Customer Resting Pulse Value	Numeric (integer)
SmokerStatus	Indicates if a customer is a smoker or not	Binary Indicator

3.2.2 Outliers

Outliers in this research are defined as values in a feature that are larger than 1.5 times the interquartile range (IQR) above the third quartile or smaller than 1.5 times the IQR below the first quartile. Agarwal and Gupta (2021) state that outliers can distort distributions of data and statistical measures used, which can result in misrepresentations of the underlying data model. The authors go on to state that removal of outliers from data used for modelling purposes can provide improved prediction performance. During the EDA it was found that several numeric fields in the initial dataset suffered from a large volume of outliers.

We remove entries with outliers from the data as appropriate, identifying and removing rows on a column-by-column basis. This approach can be considered aggressive, as other potentially useful information was removed in the process. However, the large size of the initial dataset allows for the retention of sufficient data after we apply this pre-processing step.

3.2.3 Missing values

Out of the 30 total features, 15 have missing data. Figure 3.1 indicates the percentage of missing entries in the features that have any missing values. In addition to existing missing data present in the initial extract, several columns saw an increase in missing values during their conversion from string to numeric. In the case of categorical columns, missing values were often stored as a string such as "NULL", "N/A", or a blank space character and needed to be manually converted to a missing value.

Supervised learning algorithms struggle to deal with missing values, necessitating pre-processing to address them before any models can be built. As one of the research objective questions in Section 1.2 is concerned with how the different ways of handling missing data affect the prediction accuracy of GGT values, we create several datasets to cater for the different intended scenarios.

The treatment of missing values from this research includes dropping any rows with missing values (i.e. case deletion), manual filling of missing values to insert appropriate values via domain knowledge, and various imputation methods detailed in Section 4.2. The manual filling of rows is limited in size and scope where domain knowledge is certain to avoid creating bias in the data.

Figure 3.1 shows all columns with missing values; those with a high degree of missing values are removed in the creation of dataset C in Section 3.3 below. The following columns were dropped for dataset C (compared to A and B):

- AGE
- ALL_CONTRACT_COUNT
- ALL_BEN_COUNT
- ANNUAL_INVESTMENT
- AUM

It is interesting to note that four out of the five columns above are related to customer holdings. While Age is in a similar range of missing value to other variables that were retained, its removal was in part further motivated by computational constraints. If there is a distinct difference in model performance, this may be the particular source.

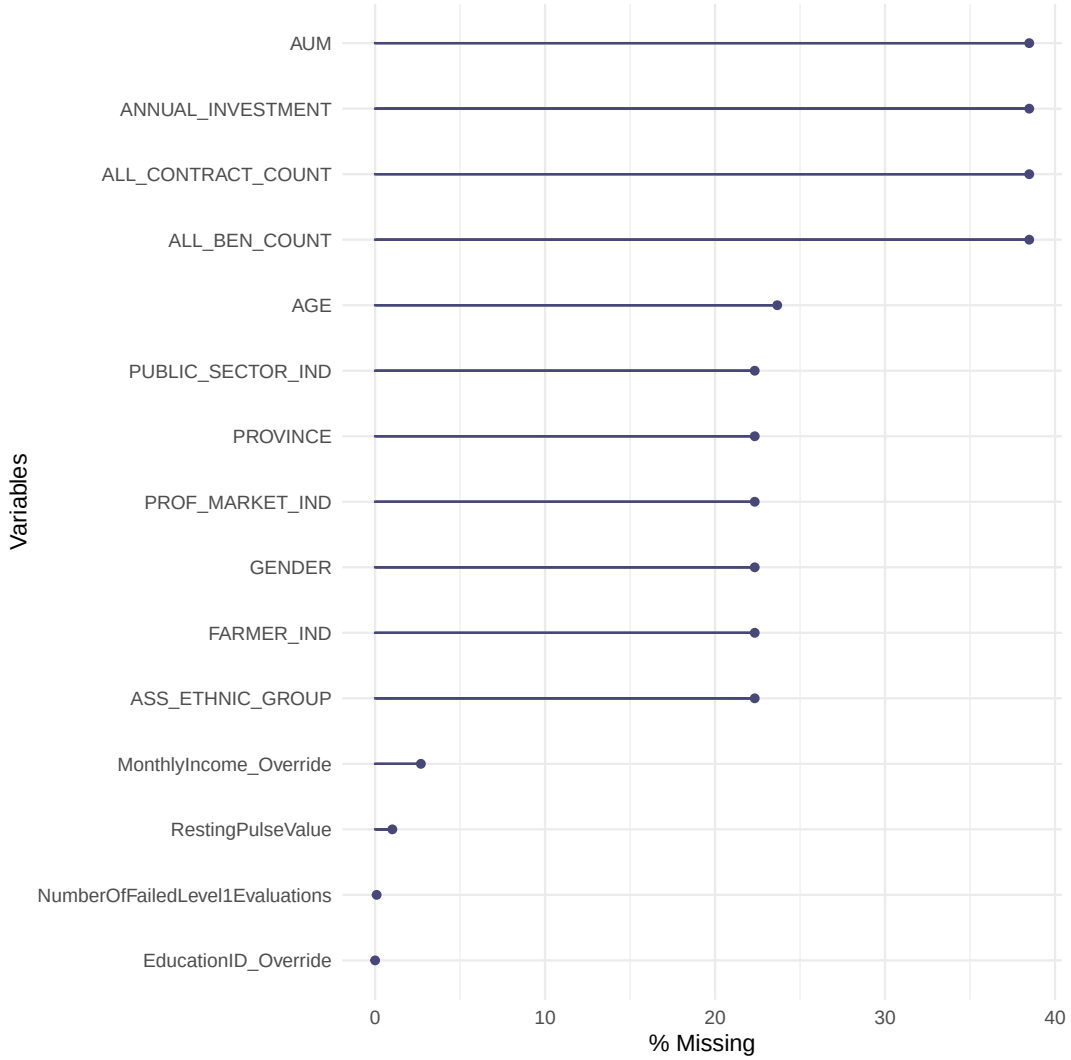


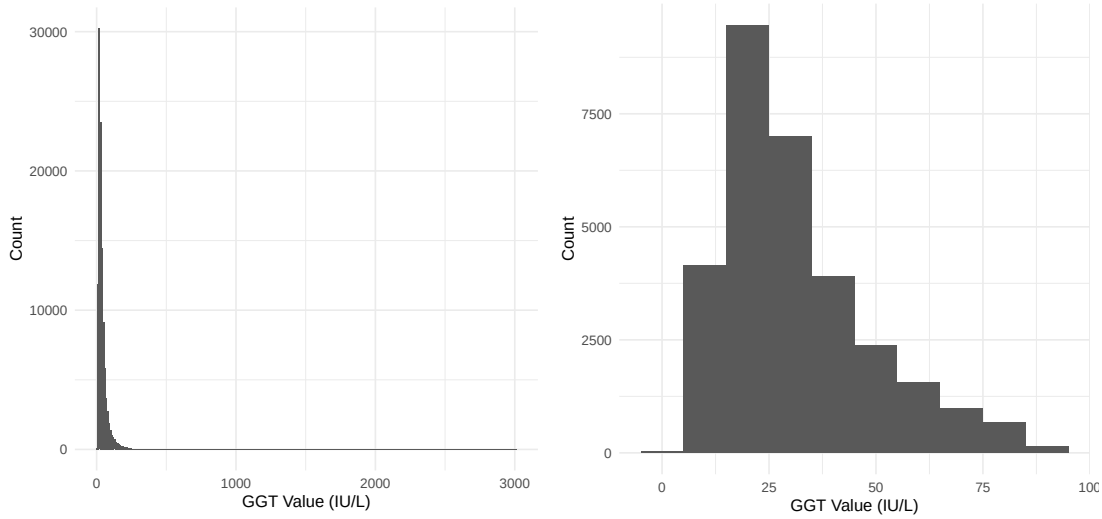
FIGURE 3.1: The percentage of missing values per feature (limited to features with missing values).

3.2.4 Target Variable Distribution

As we are aiming to predict GGT values, examining the distribution of GGT data provides valuable context. Figure 3.2 displays histograms of the GGT value before and after the pre-processing step. As can be seen from the graphic on the left, unprocessed data had extremely large outlier values which skewed the distribution dramatically. While differing methodologies have varying reference ranges for GGT, the most commonly used methods indicate a range of <50 IU/L as a normal range for males (Vroon and Israili, 1990; Mason et al., 2010). Vroon and Israili (1990) describes a normal range as <30 IU/L for females, whereas Mason et al. (2010) indicates a range of <40 IU/L is acceptable.

The pre-processed GGT graphic on the right is a single-peaked distribution, with a distinct skew to the right. The bulk of the values sit within the normal range described above and affirms the data wrangling applied.

This distribution will be of key importance when analysing predictions made by the models built in Section 4.4, as the predictions made should create a similar-looking distribution if they are to be representative.



(A) Histogram of Unprocessed GGT Distribution (B) Histogram of Pre-processed GGT Distribution

FIGURE 3.2: A comparison of GGT histograms before and after pre-processing. This illustrates the skewing effect of outliers and emphasises the need to address them.

Key components of the EDA have been explored above. Due to the nature of the research objective, there are multiple datasets created as a result of the pre-processing. The detail of these datasets is discussed in the next section.

3.3 Pre-processing Results

The main output of this chapter is 3 datasets detailed in Table 3.2 below:

TABLE 3.2: Output of pre-processing the raw Data

Dataset	Description	Size (Rows x Columns)
A	A complete-case dataset that dropped rows with any missing values	30 284 x 30
B	A dataset containing all columns, including those that had a large proportion of missing values	51 378 x 30
C	A dataset excluding columns with a large proportion of missing values, but still containing missing values.	51 378 x 25

We apply the methodology described in Chapter 4 to these datasets. Of particular note is the latter two datasets (B and C), as they will be used in the imputation steps described in Section 4.2. Dataset A has 59% of the rows that B and C have, which is a considerable reduction in the amount of information provided. Meanwhile, dataset C has 86% of the features of A and B; however, the amount of missing data in those features is considerable as shown in Figure 3.1.

The change in the dimension of the raw data to its final processed form indicates the difficulty of working with this particular dataset. The largest changes in dimensionality are due to the removal of duplicate rows and outliers. The former removed 227 489 rows, whilst the latter dropped 59 893 rows. This highlights the effort needed to ensure the quality of data input to the models.

The last output from this section is an unlabelled dataset of 10 690 rows where no GGT value data was available in the initial extract. This dataset is pre-processed similarly to A, B and C, and we use it to analyse the distribution of predicted values from the models we train as per Section 4.1 on a completely unseen dataset. We compare this to the distribution displayed in Figure 3.2 which, if the model generalises well, should be similar. Figure 3.3 provides a visual representation of how the different datasets are created.

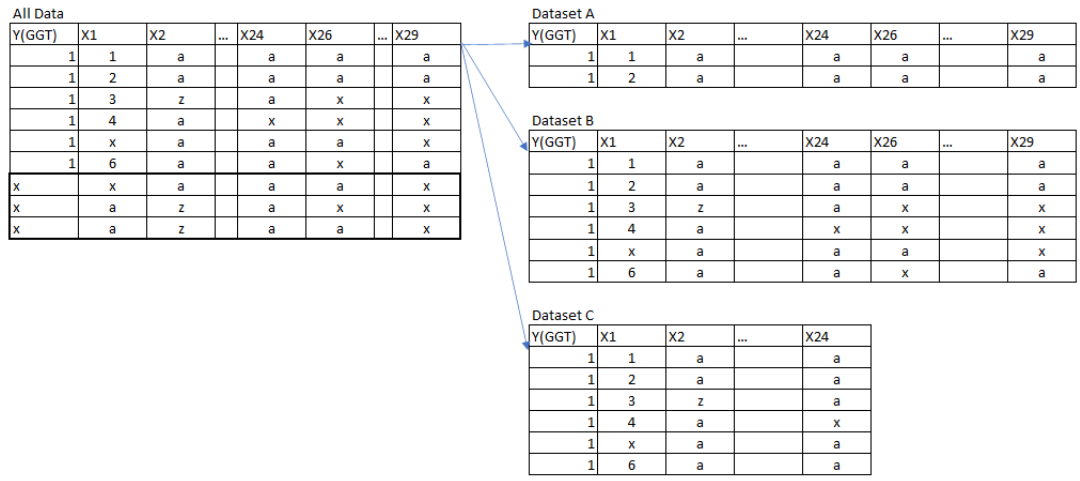


FIGURE 3.3: This Figure illustrates how the data was divided into the 3 datasets used to train models. The unlabelled portion surrounded by the bold border is used to compare the distribution of GGT created by the models to actual data. Rows with an **x** value indicate missing data.

3.4 Conclusion

In this chapter the actions we take to explore and prepare the data for use are detailed. This is an important step in becoming familiar with the data and transforming it into a format that can be used in supervised learning. We provide details on the target variable distribution and missing values in the data as key points of the research. The output of pre-processing will be three datasets to which we apply the methodology.

The first dataset is characterised by any rows or columns with missing values being treated manually where possible or dropped completely. The second dataset retains all the missing features and rows to be used in the imputation step. The third dataset retains most variables with missing features but removes columns with a high percentage of missing values. The variety of datasets necessitates a varied experimental design; we describe this in Section 4.1.

Chapter 4

Methodology

The methodology describes the techniques we use in the research. This chapter provides a high-level background on how the techniques work, explains the reasoning behind their selection and details their application. To begin, we provide the architecture of the experimental design in Section 4.1. It then focuses on imputation, briefly describing the two techniques we use. Lastly, this chapter focuses on the supervised learning models that are trained on the datasets that have been pre-processed and imputed. As this is application-based research, we will not provide rigorous mathematical detail but rather a high-level conceptual description to enable an intuitive understanding.

4.1 Experimental Design

This section describes the order in which we apply the methods in Section 4.2 and Section 4.4. The various combination of datasets, imputation methods and supervised learning techniques produce multiple outputs. The different scenarios we create are in pursuit of an answer to the sub-questions described in Section 1.2, regarding which combination of missing value treatment and supervised learning algorithm produces the most accurate GGT predictions.

The first source of differentiation in scenarios lies in the dataset being used and its treatment of missing values, whether it is manually treating, dropping or applying an imputation method. The use of two imputation methods provides a further point of difference. In the case of multiple imputation, there is the added scenario regarding the path to the final predictions.

We follow two methods, namely aggregation of the completed datasets or averaging the completed datasets. The averaging method combines the multiple completed datasets into one, then builds a model and makes predictions. While this method is not traditional multiple imputation, the motivation for its inclusion is provided below when discussing the various scenarios and considering the context of the research objective. The aggregation method trains a model on each of the completed datasets, then makes predictions using each trained model and finally averages the predictions into one final predicted value. This method allows for variation in the predictions of an incomplete record, making this method closer to the true nature of multiple imputation.

The last source of difference in scenarios comes from two supervised learning algorithms that are applied to all datasets.

Firstly, we apply an imputation method to Dataset B and C (with reference to those created in Section 3.3) to impute the missing values. Once the imputation process is complete, we split the dataset into train and test datasets (see Section 4.3 for further detail). This results in the following scenarios:

TABLE 4.1: Combinations of datasets and imputation methods

Scenario	Dataset	Imputation Method Applied	Output Description
1	A	None	1 complete dataset
2	B	missForest	1 complete dataset
3	C	missForest	1 complete dataset
4	B	MICE	5 datasets combined into 1 final dataset
5	B	MICE	Retained all 5 datasets as output
6	C	MICE	5 datasets combined into 1 final dataset
7	C	MICE	Retained all 5 datasets as output

On each dataset produced per Table 4.1 we train two supervised learning algorithms, namely random forest and XGBoost. This creates 14 unique combinations, the results of which we discuss in Section 5.2.

In the case of scenarios 5 and 7, we train a model on each of the imputed datasets and the final predictions from each model trained on each completed dataset are averaged into a final result. This aligns with the *impute-analyse-combine* framework described below in Section 4.2.2. The literature on utilising tree-based methods in conjunction with MICE is sparse, hence a combined dataset is also created for scenarios 4 and 6 to view if any material differences are achieved. The combined dataset was created by taking the average value of entries among the 5 imputed datasets for numeric columns, whereas categorical columns use the relevant mode value per entry from the imputed datasets.

This section provided a high-level overview of all the techniques used in the research. The next section goes into detail regarding the imputation techniques implemented.

4.2 Imputation

As mentioned in Section 2.4, imputation deals with filling in the missing values in datasets. The techniques used can vary from extremely simplistic rule-based actions to complex statistical techniques. This research will use two methods to provide results from different imputation techniques. The first method is missForest, which is a single imputation method that results in one, completed dataset. The second is multivariate imputation by chained equations (MICE), which results in multiple completed, imputed datasets. A brief description of how each technique works is provided in addition to a brief motivation for its inclusion.

4.2.1 MissForest

The missForest algorithm was developed by Daniel Stekhoven and Peter Bühlmann (Stekhoven and Bühlmann, 2012), with the intent to introduce an imputation algorithm that made as few assumptions as possible about the data and can deal with mixed data types.

MissForest makes use of the random forest algorithm – which is detailed in Section 4.4.1 – and performs the following steps to impute data:

1. Firstly, make an initial guess of the missing value e.g. mean imputation.
2. The procedure then cycles through each of the variables, setting the focus variable as the response variable and training a random forest using the other variables that have available information.
3. The trained model is then used to predict the missing values of the response variable.
4. The imputation procedure is repeated until a stopping criterion (e.g. improvement in mean squared error) is reached.

One iteration of this procedure is illustrated in Figure 4.1.

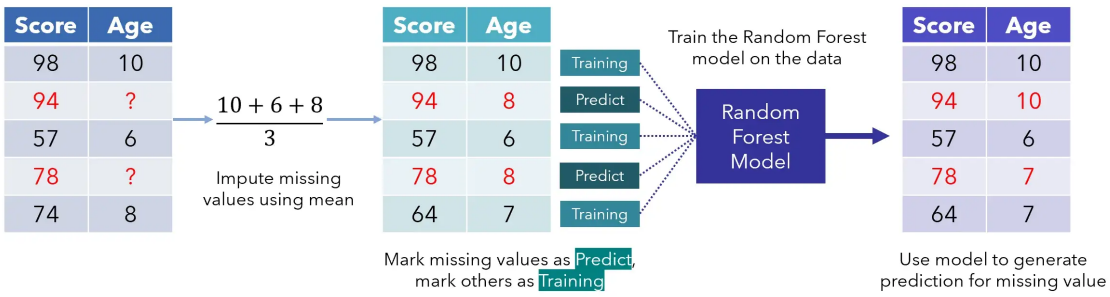


FIGURE 4.1: This Figure illustrates the first iteration of a missForest procedure (Ye, 2020). Age contains missing values, the rows of which are highlighted in red. Initially, the mean is used to fill in the missing values. Age is set as the target variable, with Score as a feature. A model is built on this intermediate dataset, which is used to predict values of Age in the original missing entries.

The random forest algorithm used is non-parametric and performs well on complex datasets due to its ability to capture intricate interactions and non-linear relationships. Testing performed by the authors illustrate instances of MissForest outperformaning well-known imputation algorithms such as KNNimpute and MICE (Stekhoven and Bühlmann, 2012). This makes it a desirable choice for missing value imputation.

The potential drawback of missForest is its current limitation to single imputation, resulting in only one dataset as output. This assumes that any imputed values are correct and leaves no information regarding the uncertainty of the imputation procedure. However, given that the overall aim of this research is using imputation as an augmentation technique in building a better-performing supervised learning model, we may

proceed with this caveat in mind.

The missForest algorithm will be implemented using the missForest package in R (Stekhoven, 2022). Key input hyperparameters and their default values are shown in Table 4.2.

TABLE 4.2: Key hyperparameters for missForest

Hyperparameter	**Default Value**	**Description**
maxiter	10	Defines the maximum cycles described above that can take place.
mtry	$\sqrt{p}$	Defines the number of features sampled at each split. p is the number of variables that contain missing values.
ntree	100	Controls the number of trees grown in each forest.

A drawback of the missForest algorithm is the long runtime required on large datasets with many features that have missing values. This is mainly due to the large number of random forests required to be trained over several iterations. Investigations with this procedure lead to the choice of using default values for these key hyperparameters, as they are most influential on model runtime and computational feasibility. Even at these hyperparameter settings, one full cycle takes more than 24 hours to complete.

4.2.2 Multivariate Imputation by Chained Equations

Multiple imputation was first proposed by Rubin (1987) as an alternative method of dealing with missing data. Where this differs from single imputation, as performed in missForest, is the procedure results in several completed datasets that consist of actual and imputed data instead of one. Multivariate imputation by chained equations (MICE) is one implementation of multivariate, multiple imputation proposed by van Buuren and Groothuis-Oudshoorn (2011).

There are three overall steps described in the typical MICE procedure: imputation, analysis, and pooling. As the name suggests, the first step imputes the missing values and is where the chained equations process takes place. van Buuren (2018) and Azur et al. (2011) describe it in the following steps:

1. Specify an imputation model for each variable.
2. For each variable, fill in any missing values with random draws from the observed data of that variable.
3. The missing values for one variable (for illustrative purposes we will call it $X1$) are set back to missing.
4. The rows of $X1$ without missing data are used to train a model using other variables in the dataset (potentially all other variables). For example, $X1$ is a binary indicator variable and logistic regression is specified as the imputation model, $X1$ becomes the target variable and the other variables become the features.
5. The missing values of variable $X1$ are replaced with predicted values from the model trained in step 4.

6. Steps 3–5 are then repeated for each variable that contains missing data. Once this process has cycled through all the variables any missing values will have been replaced with predictions. Steps 3–5 are repeated for several cycles, with the imputed values being updated at each cycle. This continues until a stopping criterion is reached.

This process will result in one complete dataset with no missing values. The researcher may choose how many of these datasets are created in the imputation step. Imputed values will differ between the datasets, capturing the uncertainty of the missing values.

The procedure will be implemented using the MICE package in R (van Buuren and Groothuis-Oudshoorn, 2011). The default imputation models specified are as follows:

- Numeric – predictive mean matching
- Categorical variable with 2 categories (Binary data) – logistic regression
- Unordered categorical data with >2 categories – polytomous regression
- Ordered categorical data with >2 categories – proportional odds model

Once the imputation stage is complete the analysis may proceed on each dataset as normal. Finally, the results of the multiple analyses are pooled into one final result (van Buuren, 2018). Azur et al. (2011) describes an example of training regression models on each of the imputed datasets, followed by combining the coefficients and parameter estimates of each model into a final set of estimates.

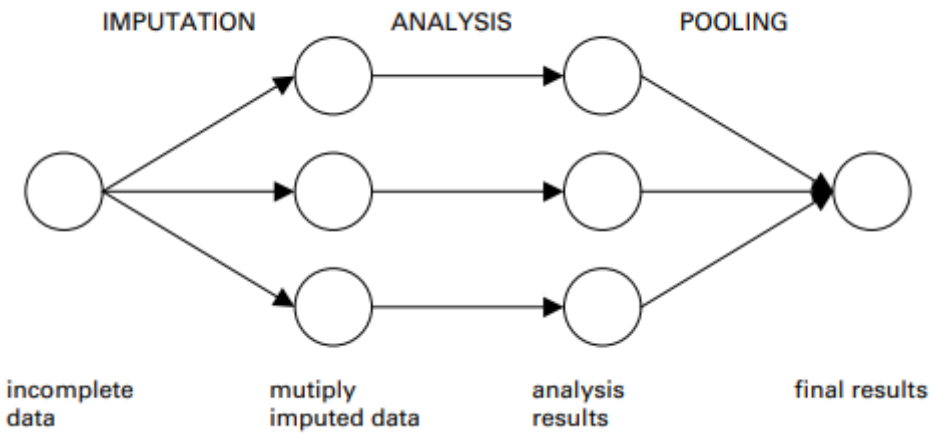


FIGURE 4.2: This Figure displays an example of a typical multiple imputation procedure (van Buuren, 2018). Each circle represents a dataset. The use of multiple datasets containing different imputed values helps to capture the uncertainty around the imputation process and provides more robust results when the analysis is pooled together.

The three high-level procedures of multiple imputation described above are illustrated in Figure 4.2. MICE is one of the most popular multiple imputation methods available and can be used in a range of settings (Miok et al., 2020; Azur et al., 2011). It offers flexibility in terms of the data types it can handle and is able to create multiple sets of

imputed data. This allows for the uncertainty involved in the imputation process to be captured, something not possible in a single imputation procedure.

As mentioned above, we use the MICE package in R. This package offers flexibility and control to the user in terms of specifying the inputs into the imputation procedure, such as selecting the variables to be used and setting the preferred imputation model to be used on each one. The key hyperparameters and their default values for the package are displayed in table 4.3.

TABLE 4.3: Key hyperparameters for MICE

Hyperparameter	**Default Value**	**Description**
m	5	The number of datasets to be imputed
maxit	10	The maximum number of iterations to run before a stopping criterion is reached
method	As per above	Controls the imputation model to be used on each variable

As with missForest, the runtime and computational feasibility need to be taken into account. Due to time and resource constraints, the default hyperparameters were used in this research. While faster than missForest, one complete imputation procedure with MICE requires several hours even with the default hyperparameter settings.

Multiple imputation requires careful consideration when being set up to ensure that reliable imputations are achieved. The imputation model selected, selection of data to be included, appropriate input hyperparameters and checking for convergence are among the questions that need to be asked when setting up the procedure. MICE in particular assumes missing data are MAR by default. It is able to handle MNAR, but additional modelling is required in this form (van Buuren and Groothuis-Oudshoorn, 2011).

Given that true randomness of missing values in data is a difficult condition to meet, the strict requirements of MCAR indicate that this assumption is not suitable for this research. Given the MAR definition that missingness does not depend on unobserved data but does depend on observed values, testing for MNAR would require unavailable information about the missing data (Li, 2013). However, there is currently no way of directly testing whether the missingness in a dataset is MAR or MNAR (Potthoff et al., 2006). Furthermore, the large number of features in the data, with several containing missing values, renders manual investigation into the pattern of missingness extremely challenging. However, examining the description of fields indicates possible relationships between variables that would allow the missing values to be predicted based on the observed values in other features. Acock (2005) indicates that factors such as age, education level, ethnicity and gender are common mechanisms that can be used to determine the pattern of missingness; all of which are currently included in the dataset. An example of this is younger adults may have a larger proportion of missing values in the income feature if they choose not to disclose such information. However, the actual missing income value does not affect the fact that it happens to be missing, which would be the case for NMAR. Considering all features are related to a customer's profile and should contain some systematic relationships between them, MAR may be a reasonable assumption to use for this research.

The discussion above highlights the additional aspects to be cognisant of when using MICE, and imputation methods in general. Once the various imputation procedures have been applied the training and test datasets need to be created. This process is discussed in the section that follows.

4.3 Train-Test Splits

If an algorithm is trained on an entire dataset, there is a risk of learning the structure of that specific instance of data too well. If this algorithm is tested on a new extract of data from the same source as the training data, it may perform poorly; this is known as overfitting (James et al., 2013). To ensure a model generalises well on data it has not been trained on, a common technique used is splitting the dataset into two. One dataset will be used to train the model, while the other is kept aside to evaluate the model once training is complete, i.e. to test the model on unseen data.

Once we apply the imputation procedure, the train-test split is executed. The application of a train-test split for this research requires specific attention to ensure that model performance is comparable across differing datasets. To achieve this we employ the following:

1. Create a unique key at row level on the raw data before pre-processing. This is carried throughout all work done.
2. Using dataset A, perform a random 74:26 train-test split and retrieve the relevant row indices for the training and test datasets.
3. With the test set index and their related unique keys, remove any test set related rows from all datasets.
4. Use the test set indices to create a test dataset from one of datasets A, B or C.

This process ensures that none of the training sets will see any of the data related to the test set, making for a fair and comparable dataset to evaluate the performance of different data treatments and supervised learning combinations. The test set is extracted from dataset B. Some values will have been imputed due to slight differences in pre-processing compared to values in dataset A. However, this particular choice of dataset was made deliberately to more accurately reflect a real-world scenario where missing data are prevalent in input features and may need to be imputed.

The test set would make up a different proportion of each dataset. A middle ground was chosen in terms of the final train-test ratio to ensure a reasonably sized test set for all datasets while leaving sufficient data for model training. The final dataset sizes are as displayed in Table 4.4.

The missForest and MICE techniques above are key components in dealing with the missing values in the datasets. Once the imputation processes have run, the research moves to training supervised learning models. These techniques are discussed in the next section.

TABLE 4.4: Train-test split dataset sizes

Related Dataset	**Overall Size (rows)**	**Training Data (rows)**	**Test Data (rows)**	**Test Data (Proportion of Overall)**
A	30284	22412	7872 (Same dataset)	26%
B	51378	43056		15%
C	51378	43056		15%

4.4 Supervised Learning

The goal of a supervised learning exercise is to use a set of input variables, which may have some influence on an output variable, to predict the output (Hastie et al., 2009). Below we briefly examine a few key concepts related to the supervised learning process that we use. This is followed by a description of the supervised learning techniques implemented in Chapter 5. It briefly motivates why the technique was selected and provides a high-level intuitive understanding of its mechanics.

It is important to note we are explicitly using heuristics that are agnostic to the underlying physiological data. It may be the case that techniques more specific and attuned to a medical context could work but would require specific domain knowledge to set up and execute. A combination of imputation and supervised learning is data agnostic, allowing us to sidestep the need for a specific combination of statistical and medical knowledge.

K-Fold Cross-Validation

The first supervised learning concept we discuss is K-Fold cross-validations. Cross-validation is a valuable technique in assessing how an algorithm may generalise on unseen data, without necessarily having to set aside a separate dataset. If a test set is infeasible due to a lack of data, this allows the maximum amount of data to be used in training the model. K-Fold cross-validation works by segmenting the training dataset into k folds. The model then trains on $k - 1$ folds, while the held-out fold is used as unseen data to validate model performance. The process revolves until each segment has been used as unseen data. The algorithm's performance on each fold can then be viewed, and an average score can be taken to get an overall idea of model performance. This is useful when comparing algorithms as it gives an indication of performance on unseen data while keeping the test set completely untouched (James et al., 2013). Figure 4.3 displays the concept of K-fold cross-validation.

While k can be chosen to be as large as needed, due to computational limitations this research limits use to 5-fold cross-validation.

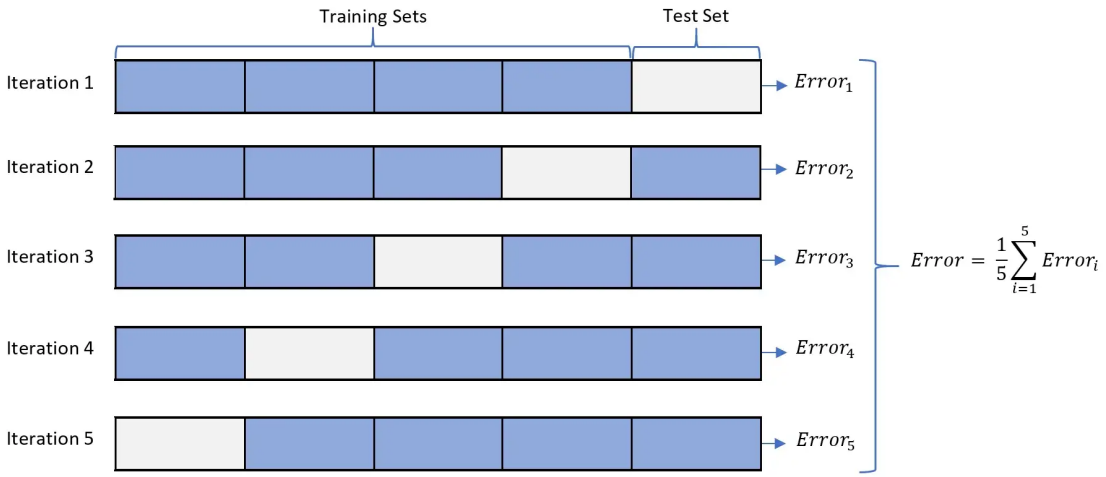


FIGURE 4.3: This Figure is an example of 5-fold cross-validation (Patro, 2021). The blue portions represents the training datasets, while the grey represents the validation datasets. The cross-validation error is obtained by averaging the prediction error from the 5 grey portions.

Hyperparameter Tuning

The performance of supervised learning algorithms is sensitive to their input hyperparameters. For example, in a random forest, this can be the number of trees trained or the learning rate for a neural network. A grid search can be used to determine possible values for these hyperparameters to extract as much performance from a model as possible. A grid search requires an input of a range of values for a hyperparameter that an algorithm will use and trains a model on each of the values supplied. The hyperparameter value for the best-performing model can then be extracted. If values for more than one hyperparameter are supplied a grid search will train several models over all combinations of hyperparameter values.

We have briefly discussed supervised learning and its key concepts of k-fold cross-validation and hyperparameter tuning. The following sections discuss the supervised learning algorithms we use in this research.

4.4.1 Random Forest

Developed by L. Breiman (2001), the random forest algorithm has proven to be highly successful in both classification and regression tasks (Biau and Scornet, 2016). Biau and Scornet (2016) describes the high-level idea behind random forests involving 3 main steps:

1. Sample multiple fractions of the data
2. Train a randomised tree predictor on each fraction
3. Aggregate the predictions of all the trees

The steps described above are visualised in Figure 4.4. Each tree is built on a subset of the data and a subset of variables. In the case of regression, the prediction values

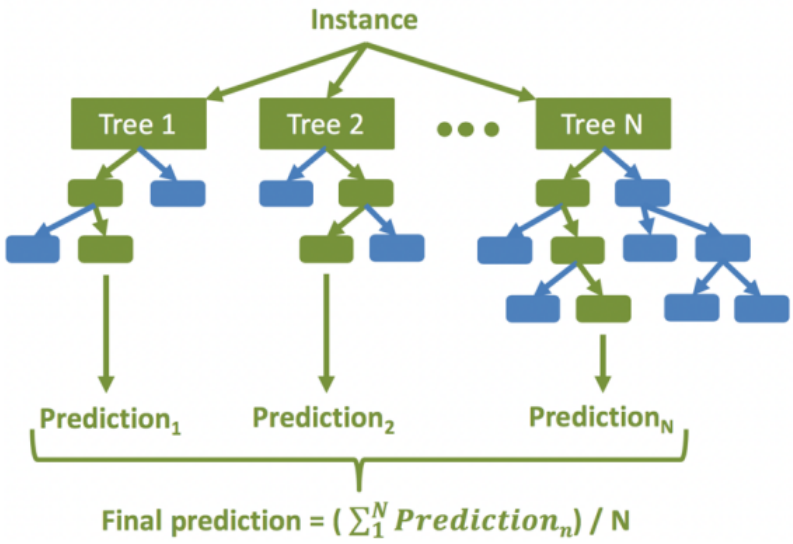


FIGURE 4.4: This Figure provides an example of random forest architecture (McCandless and Haupt, 2019). Random subsets of data and features make up each tree. The prediction from each tree is averaged into a single final prediction. In regression, this can be the average value of the prediction.

from all the trees are averaged into a final prediction. In classification, the class with the majority of votes from each tree is the final prediction.

Random forests are an ensemble type of algorithm and build upon the established techniques of decision trees and bagging (Biau and Scornet, 2016). Randomisation is introduced via two sources: one via random sampling of data, which takes place with replacement. The second is at the node split on each tree: the best split is made from a randomly selected subset of the input features (Bentéjac et al., 2019).

Biau and Scornet (2016) describes several benefits of random forests that contribute to their popularity. It is recognised for its accuracy and ease of use, requiring few hyperparameters to tune. Furthermore, it can deal with smaller datasets and high-dimensional feature spaces. Simultaneously, it is highly parallelisable and can therefore deal with the large datasets common in modern settings while remaining computationally efficient. Furthermore, it has shown success in a wide variety of use cases from air quality predictions to 3D object recognition. Lastly, it is able to return a measure of feature importance. The full technical details of the algorithm are beyond the scope of this research but can be found in Breiman (2001). The aforementioned benefits make it a good candidate to be used in the supervised learning section of this research which trains a model to predict GGT test results.

The random forest algorithm will be implemented using the CARET package in R (Kuhn, 2008). This package also performs any cross-validation and hyperparameter tuning required. We describe key hyperparameters of random forests Table 4.5.

TABLE 4.5: Key hyperparameters for random forest

Hyperparameter	**Description**
ntrees	Number of trees grown in training the model.
mtry	Number of variables selected as candidates for each split. This helps to prevent overfitting.
min.node.size	Number of entries needed in a leaf node. Acts as a form of regularisation.

4.4.2 XGBoost

The XGBoost algorithm was proposed in 2016 by Tianqi Chen and Carlos Guestrin as "a scalable end-to-end tree boosting system" (Chen and Guestrin, 2016). Similar to random forests, XGBoost is an ensemble type of algorithm. However, it is built from the family of boosting algorithms; these combine several weak learners (typically decision trees) into one strong learner (Bentéjac et al., 2019). It does this by iteratively building decision trees to fit the residuals created by the preceding tree. The overarching idea is that with each tree created, the prediction becomes a little closer to the actual values and the overall model improves. Gradient boosted trees have a tendency to over-fit, which has resulted in many of them employing regularisation and shrinkage techniques during the model training process to counteract this.

XGBoost is an implementation of the gradient-boosted trees algorithm, with a strong focus on computational efficiency in tandem with accuracy. According to Chen and Guestrin (2016), it achieves this through specific implementations such as:

- Regularised Learning objective function
- Shrinkage and Column Sub-sampling
- Approximate Greedy Algorithm
- Weighted Quantile Sketch and Parallel Computing
- Sparsity aware split finding

The items listed above all deal with reducing over-fitting and computational load. The last item – sparsity aware split finding – enables XGBoost to deal with missing values in the input features, giving it greater flexibility. An interesting component of the XGBoost implementation is its focus on computational efficiency unrelated to statistics. Two examples of this are cache-aware access to utilise the fastest memory available where possible, and blocks for out-of-core computation to reduce read time off a hard disk. The full technical detail of the algorithm is beyond the scope of this research but can be found in Chen and Guestrin (2016).

The XGBoost algorithm has shown impressive performance in a wide variety of use cases and has come to dominate many Kaggle competitions (Chen and Guestrin, 2016).

Its impressive accuracy on tabular data and computing efficiency make it a good candidate to be used in this research.

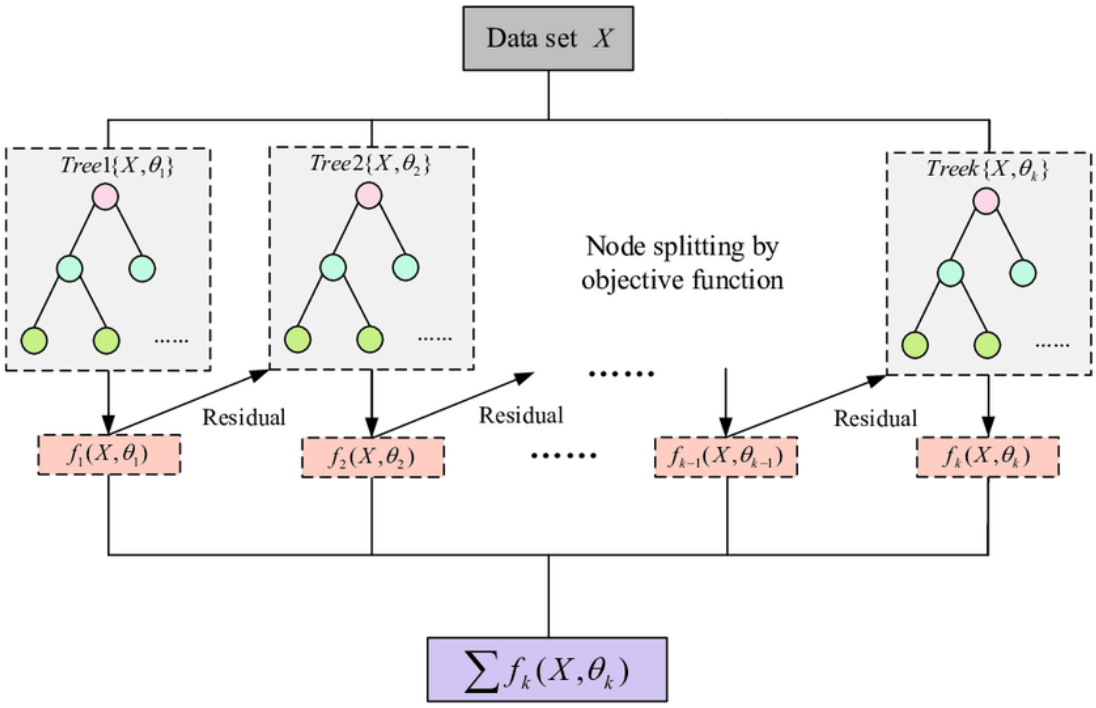


FIGURE 4.5: This Figure displays the boosting procedure used by XGBoost Guo et al. (2020). Successive learners are trained on the residuals (i.e. errors) of the previous learner. All predictions from individual learners are aggregated into a final prediction.

Figure 4.5 illustrates a portion of the XGBoost algorithm. It shows how each sequential tree, which are all weak learners, is trained to predict the errors (residuals) of the previous tree. The final prediction is an aggregate of the initial prediction and the predicted errors provided by the trees.

The XGBoost algorithm will be implemented using the CARET package in R. As with random forests above, this package is able to perform cross-validation and hyperparameter tuning. The key hyperparameters for XGBoost are described in Table 4.6.

TABLE 4.6: Key hyperparameters for XGBoost

Hyperparameter	**Description**
nrounds	Number of rounds for boosting.
max_depth	Maximum depth of a tree. Shallower trees are less complex and less likely to overfit.
eta	Step size shrinkage. This reduces the impact of a single learner's prediction and acts as a form of regularisation.
gamma	Specifies the minimum loss reduction required to further partition a leaf node. A larger gamma will result in a more conservative algorithm
colsample_bytree	Defines proportion of columns that are subsampled for each tree grown
min_child_weight	Minimum number of instances required in each node. A larger value is more conservative
subsample	The proportion of training data that can be randomly sampled prior to growing a tree. This helps to prevent overfitting

4.5 Conclusion

This chapter provides detail on the techniques we apply in this research. Initially, we provide a strategy on how the research goes about achieving its final results presented in Chapter 5. Secondly, a high-level description of the techniques is provided to enable an understanding of how they work without delving into mathematical detail. This chapter also provides some of the advantages of each technique to bolster the decision to use them. The methodology composition is crucial to achieving the goals set out in the research objective.

Chapter 5

Results

This chapter provides results of applying the techniques described in Chapter 4. Firstly, we provide a discussion on how model performance is measured. We follow this with results from all techniques that were applied to the datasets as per Section 4.1. Furthermore, we provide an analysis comparing unlabelled GGT predictions to actual data. Lastly, we review the results and determine if meaningful progress towards the research objective was made.

5.1 Discussion on Performance Metric

Regression models focus on predicting continuous variables, with applications such as predictions for housing prices or temperature. We evaluate regression model performance using metrics to examine the error (i.e. distance) between predicted and actual values. These can be difficult to interpret without context. This is unlike classification models which can be evaluated on absolute performance metrics that are easier to interpret, such as the accuracy of class predictions.

Popular metrics used to measure regression performance are mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE) (Botchkarev, 2018). For this research RMSE was chosen as the key metric used to measure performance. While several metrics can be used to measure regression model performance, there is no clear consensus on which is best. Willmott and Matsuura (2005) puts forward MAE as a superior indicator of average error, whereas Chai and Draxler (2014) advocates for RMSE – particularly when the error distribution is expected to be Gaussian.

Ease of calculation and the units being the same as the predicted variable (unlike MSE which is units squared) are some advantages of RMSE and contributed to it being used as the metric of choice for this research.

The formula used to calculate RMSE is displayed below:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

where $\hat{y}_i$ are the predicted values, y_i are the actual values and N is the number of observations.

As we are unable to evaluate regression model performance in absolute terms, we focus mainly on relative comparisons. However, a simple baseline value is needed as a target to be improved upon to determine if the techniques can add value. This research uses the standard deviation of the data as this baseline. If we were to choose the mean as a value for all predictions, the standard deviation acts as an analogous measure of variation to RMSE. Therefore, for a model to add value it must offer a lower RMSE on its predictions than the standard deviation. Furthermore, the comparison of RMSE on the test set predictions from the different models trained determines the best-performing scenario.

This section discusses the manner of model performance evaluation for this research. In the following section, the results are provided and interpreted.

5.2 Results

When the imputation process converges, diagnostic checks like those shown in Figure 5.1 can be performed to view if the imputed values are reasonable. Each trace line represents one of the five imputed datasets. The aim is to see the trace lines intermingling within a reasonable range. Large values or strict patterns would indicate an issue with the imputation, potentially with the imputation model used. From the diagnostic checks, no major issues were revealed.

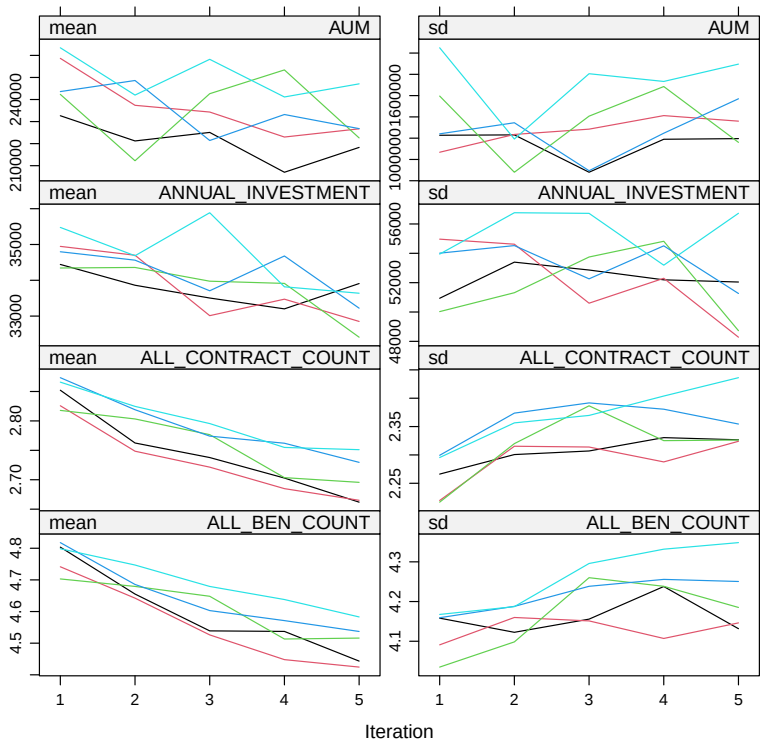


FIGURE 5.1: This Figure is an example of the diagnostic checking offered by the MICE package in R. We examine four features with the most missing values in the data. The trace lines display the mean and standard deviation values of the features across iterations. The intermingling of lines indicates a desirable outcome for this imputation procedure.

Once the imputed data are deemed reasonable, we can analyse results from the supervised learning models that are trained on them. Table 5.1 details the 14 different scenarios and the related result that emerge from the application of the methodology detailed in Section 4.1.

TABLE 5.1: Model performance results

Scenario	Dataset	Imputation Method Applied	Model	Test RMSE
1	A	None	RF	12.415
2	B	missForest	RF	12.463
3	C	missForest	RF	12.385
4	B	MICE - combined	RF	12.503
5	B	MICE - 5 datasets	RF	12.491
6	C	MICE - combined	RF	12.666
7	C	MICE - 5 datasets	RF	12.629
8	A	None	XGB	12.034
9	B	missForest	XGB	11.965
10	C	missForest	XGB	12.210
11	B	MICE - combined	XGB	11.998
12	B	MICE - 5 datasets	XGB	11.725
13	C	MICE - combined	XGB	12.295
14	C	MICE - 5 datasets	XGB	11.888

We evaluate the performance of a model based on RMSE, as described in Section 5.1, using test set data predictions. The test set, as per Section 4.3 was not used by any of the supervised learning methods during training and acts as a measure of how the model generalises and performs on unseen data.

To begin with, we make a comparison to the standard deviation of the data. The standard deviation acts as an analogous measure of variation if all predictions were set to the mean value. Therefore, it acts as a baseline measure to evaluate if model performance is better than a simple prediction. In other words, do the techniques applied add value and enable accuracy better than a guess? The standard deviation of the test set is 17.065. All scenarios in the results appear to perform better than this. The worst performing model occurs in Scenario 6 with an RMSE of 12.666, while the best performing can be found in scenario 12 with an RMSE of 11.725. This is an improvement of 26% and 31% respectively. The lowest RMSE per dataset, model and imputation method respectively are presented in Section 5.4 as they pertain to the research questions posed in Section 1.2.

While it appears that the techniques can add value to GGT prediction, there does appear to be a limit to the accuracy it can provide. This is despite varying the input dataset with differing numbers of rows and columns, imputation techniques and supervised learning algorithms. Additionally, we perform a hyperparameter tuning exercise to find the combination of hyperparameters that results in the lowest RMSE. We use a grid search over the key hyperparameters described in Section 4.4 for every model trained. The initial grid search performed was over a wide range of values for each hyperparameter. Over several iterations, the range of values was narrowed down to those providing the best results. The grid values searched that create the final models are shown in Tables 5.2 and 5.3. Certain hyperparameters such as *nrounds* (for XGBoost) and *num.trees* (for

random forest) display a single value. During iterative training rounds, these values were narrowed down such that the marginal gain in performance was not outweighed by increased runtime.

The resulting scenarios in Table 5.1 show a difference in RMSE of merely 0.941 between the best and worst scenarios. This may indicate that improved performance in GGT may not lie within the application of these techniques, but perhaps a more varied set of input features. Alternatively, an increase in the sheer volume of input data may be where this performance could be extracted, given the large reduction in dataset size from the initial raw extraction as described in Section 3.3.

Another consideration is though the test set has some imputations, do the models perform similarly on a dataset with a higher degree of imputation? Fortunately, the cross-validation results provide a proxy for a heavily imputed test set. Analysis of the cross-validated RMSE reveals that a similar pattern of performance emerges around the various scenarios, indicating that a material performance differentiator may not lie in the degree of imputation in a dataset.

TABLE 5.2: Random forest final grid search values

Hyperparameter	**Values Searched**
num.trees	100
mtry	[34,35,36]
min.node.size	[2,3,4]

TABLE 5.3: XGBoost final grid search values

Hyperparameter	**Values Searched**
nrounds	500
eta	0.1
colsample_bytree	0.8
min_child_weight	1
subsample	0.8
max_depth	[6,12]
gamma	[0,0.1]

5.2.1 Best Performing Model

As Table 5.1 indicates, the best-performing model from an RMSE perspective can be found in scenario 12. This consists of using dataset B, which contains all features – including those with a large degree of missing values. The MICE package was used to apply a multiple imputation procedure to impute any missing values in the features. An XGBoost model was trained on each of the 5 imputed datasets. These models were then used to make predictions on the test data set, with a final prediction being the average of all 5 model predictions.

As mentioned above, a grid search is used to determine the combination of hyperparameters that offered the best performance. From the values searched over described in Table 5.3, we display the best performing hyperparameters for XGBoost from an RMSE perspective in Table 5.4.

TABLE 5.4: Hyperparameters of best performing model

Hyperparameter	**Value**
nrounds	500
eta	0.1
colsample_bytree	0.8
min_child_weight	1
subsample	0.8
max_depth	12
gamma	0.01

While the design above produces the most accurate predictions of GGT in the research, we return to the previously mentioned comment on the perceived ceiling in the predictive performance of the models. Given the small spread of RMSE among the differing scenarios, computational efficiency may dictate a preferred method in practical terms if a slight trade-off in accuracy is acceptable.

If this is taken into account, dataset A is the simplest to prepare, avoiding the long runtime required to perform MICE imputation. Furthermore, the XGBoost algorithm is one built for speed. This describes scenario 8, which produces a test RMSE of 12.034. This is an increase of 2.64% RMSE for simpler implementation and quicker run time. Depending on the scenario this may be preferable.

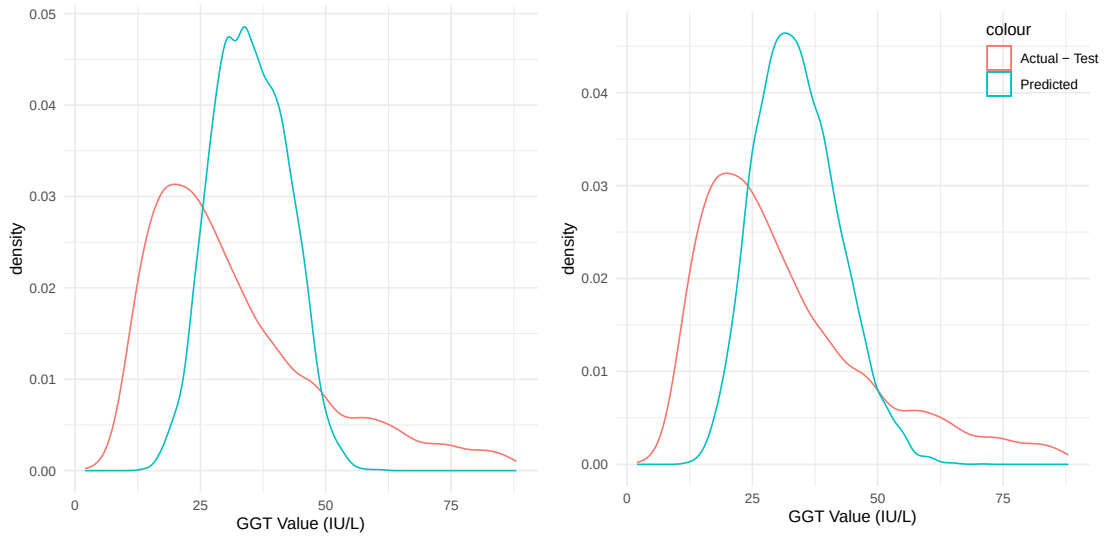
The results above discuss the errors found between predicted and actual values, allowing us to determine the best-performing scenario. In the next section, we examine how the distributions produced from predictions on an unlabelled dataset compare to actual GGT data.

5.3 Predicted vs. Known Distribution

This section utilises the dataset from Section 3.3 of 10 690 rows that had missing GGT values. The best-performing random forest and XGBoost model from an RMSE perspective are used to predict GGT values on this dataset. The distribution of these values is then compared to the distribution of actual GGT values seen in the data.

Figure 5.2 displays distributions of predicted and actual values. It appears for both random forest and XGBoost, the distribution of predicted values does not match those of the actual GGT data. The predicted values appear to have a narrower spread, with a larger proportion of the values appearing between 25 and 50 IU/L. It also appears less skewed than the actual data, with fewer values greater than 50 IU/L.

Table 5.5 provides a 5-number summary of GGT values in the actual data and those predicted by a random forest and XGBoost model. This further emphasises the narrower range when observing the minimum and maximum values. The 3rd XGBoost appears to perform better than the random forest, with values closer to the actual data in each category. The differing mean and median values of the predictions compared to actual data further emphasise the difference in distributions.



(A) Random Forest (B) XGBoost

FIGURE 5.2: A density plot of predicted vs. actual GGT values. The distributions do not appear to be matching, with predicted values occupying in a much narrower range. Visually it does not appear as if the predicted values are obtained from a similar distribution to the actual data.

Given the comparison of two distributions, a Goodness of Fit test such as a two-sample Kolmogorov-Smirnov test could be applied. However, the resulting test statistic will likely have little value due to the large sample sizes (Lazariv and Lehmann, 2018). Therefore, we examine the distributions visually and empirically to draw insights.

This finding presents an issue if the models were to be used in an actuarial analysis to fill in missing GGT data, as predictions made would not fall within the expected values and potentially bias the overall analysis. A note here may be that this particular population may not be representative of the whole, and their related GGT values are indeed within this range. Further testing on wider samples may be needed to confirm if the models are indeed unable to produce predicted values from a similar distribution to the actual data.

TABLE 5.5: Five number summary comparison

	Actual Data	**Unlabelled Prediction: Random Forest**	**Unlabelled Prediction: XGBoost**
Minimum	2	13.56	11.9
1st Quartile	19	29.43	27.94
Median	27	34.6	33.38
Mean	31.72	34.88	34
3rd Quartile	40	40.36	39.45
Maximum	88	61.18	70.96

Now that the results have been discussed they can be evaluated against the original research objectives, which we examine in the next section.

5.4 Evaluating the Research Objective

Table 5.1 allows us to evaluate how well the research addresses the questions posed in Section 1.2. The research goals are restated below and discussed individually in the context of available results.

Overall Question: *How accurately can GGT values be predicted from existing data?*

Initial comparisons of RMSE to the standard deviation of the data indicate a combination of missing value treatment and supervised learning algorithms can add value in terms of accuracy over a simple prediction choice. However, despite varying experimental designs and hyperparameter tuning, there appears to be a limit within the current feature set as to how accurate a model it can produce.

Question 1: *Which dataset yields the best results: imputed or complete-case?*

Table 5.6 displays the lowest RMSE achieved for each dataset. Dataset B produced the lowest RMSE among all scenarios. As per Table 3.2, this dataset contained all 34 input features and retained all 51 378 rows. The input features include those with a high degree of missing entries. Despite this, it appears that having more data available overall was valuable, which is the benefit of retaining all features and rows. This does align with the rule of thumb that more data are usually preferable.

TABLE 5.6: Lowest RMSE achieved per dataset

Dataset	Lowest RMSE Achieved
A	12.034
B	11.725
C	11.888

Question 2: *Which imputation method yields the best result?*

Table 5.7 displays the lowest RMSE achieved for each imputation method used, with MICE (using 5 imputed datasets) achieving the lowest overall value. Additionally, while neither imputation method seems to have gained a significant advantage over the other, Table 5.1 indicates on average MICE produced lower RMSE values than missForest when comparing all scenarios. Furthermore, in execution time MICE had a distinct advantage over missForest. MissForest required more than 24 hours to run, whereas MICE required a fraction of this.

TABLE 5.7: Lowest RMSE achieved per imputation method

Imputation Method	Lowest RMSE Achieved
None	12.034
missForest	11.965
MICE - combined	11.998
MICE - 5 datasets	11.725

Question 3: *Which supervised learning algorithm yields the best results?*

Table 5.8 displays the lowest RMSE achieved for each supervised learning algorithm, with XGBoost returning the lowest RMSE between the two. Furthermore, as per Table 5.1, in every like-for-like scenario pitting random forest against XGBoost, XGBoost has emerged as the superior supervised learning method from an RMSE perspective. Given its dominance in data science competitions, this is not surprising. However, the algorithm itself fails to provide a substantial decrease in RMSE relative to random forests.

TABLE 5.8: Lowest RMSE achieved per supervised learning algorithm

Model	**Lowest RMSE Achieved**
Random Forest	12.385
XGBoost	11.725

Question 4: *Which combination of data handling technique and supervised learning algorithm yielded the best results?*

As mentioned in Section 5.2.1, the combination of MICE (applied to a dataset to impute missing values) and XGBoost (to train a supervised learning model to predict GGT values) yielded the best results. What is important to note is that this scenario retains all 5 complete imputed datasets, and trains an XGBoost model on each one. All the models are then used to make predictions, with the average of all 5 predicted values being output as the final value.

5.5 Conclusion

This chapter displays and interrogates the results of applying the methodology described in Chapter 4. It discusses how success is defined for the models created and determines the structure of the model with the best-performing results as per this defined metric. Lastly, it discusses the results in the context of the research objective and its various sub-goals to determine if they have been met.

Chapter 6

Conclusion and Recommendations

This chapter summarises the research done, evaluates the results against the research objective and provides recommendations for future research that may lead to improved outcomes.

6.1 Summary of Research

In Chapter 1 we provide the context and motivation for the research and its potential benefits to life insurance companies. An overall outline of the research is included to give the reader a high-level overview of what to expect.

Chapter 2 provides a review of relevant literature. To provide sufficient background to the analysis and to orient the reader, this begins with an exploration of life insurance and its mechanisms. A selection of use cases from the medical and insurance field highlights the usage of both imputation and machine learning techniques in the respective fields and provides a basis for the research.

Once the foundation is established, we introduce the data we use in Chapter 3. Data exploration is a critical step in gaining insight and unearthing issues. We describe how the data has been sourced, and the various types of fields included within the dataset. It then describes how the data was explored and highlights the common issues that affect supervised learning such as outliers and zero-variance features. Lastly, it uses information from the exploration to perform any pre-processing that needs to take place before the methodology can be applied as per Chapter 4.

Chapter 4 provides a high-level explanation of how the imputation and supervised learning techniques work and describes the advantages that contribute to their selection for the methodology. It also provides an overall plan of how they are applied to the data to achieve the final results in Chapter 5.

Chapter 5 provides a brief discussion of the metric chosen to measure performance. It then provides the overall results before discussing them. Crucially, it provides a comparison between predicted and actual GGT distributions to see if predictions on unseen, unlabelled data achieve a similar distribution and validates the predictions further.

The descriptions above provide an overview of the objective of each chapter. The following section provides a summary of Section 5.4, where we evaluate the results of the research against the initial research objectives outlined in Section 1.2.

6.2 Evaluation of Research Objective

Section 1.2 states the research objective is to determine how accurately statistical imputation techniques and supervised learning algorithms could be used to predict GGT results using customer, underwriting, policy, and available biomedical data.

The discussion of results in Section 5.2 indicates that the application of techniques described in Chapter 4 does indeed offer gains in accuracy when compared to a simple baseline estimate. The best-performing combination of techniques is MICE for imputation and XGBoost for supervised learning; specifically retaining all 5 datasets and averaging the predictions from 5 models trained on each dataset. Comparisons of the test dataset RMSE achieved by this design to standard deviation resulted in a 31% improvement in accuracy. However, it must be noted that the difference in performance between the various techniques was minimal. This indicates that although imputation and supervised learning do offer advantages over not using them at all, the actual techniques themselves may have an upper limit to the accuracy they can offer.

Furthermore, when comparing distributions of data without actual GGT values to the distribution of GGT values found in the data, the predicted values do not appear to match.

Given these findings, it appears that while the techniques could be useful in terms of providing a more complete analysis of the data where no alternatives exist, further gains in accuracy may be required before it can be deployed for usage in actuarial analysis or indeed as an alternative to pathologist blood tests in front-line sales.

With the progress against the research objective described above, it is prudent to explore avenues of research where improvements may lie. We discuss these potential options in the next section.

6.3 Recommendations

This research has indicated there are possible benefits to supervised learning methods and imputation techniques being used to predict GGT using existing data. The challenge from this point forward is investigating how far accuracy can be improved upon, and indeed if there is a tipping point where life and medical insurance companies may consider the model reliable enough to implement into their business and reduce the need for physical GGT assessment.A few options are open to future research looking to expand into this area:

Firstly, the sourcing of data could be looked into. Though marginal, results indicate that both more entries and features lead to superior results. This could take the form of using datasets over a longer period, having fewer initial filters, using different product sets or perhaps even sourcing data from a variety of insurers. An alternative set of data may be obtained purely from clinical studies involving GGT, removing the demographic and customer-related fields to access more medically detailed features. Alternatives could be investigated if more detailed customer information is needed, such as scraping social media to engineer features to augment and diversify input features.

Secondly, the data wrangling and pre-processing could be expanded upon. Alternative methods in the treatment of issues mentioned in Section 3.2 could be investigated. This would create a different dataset that may result in greater accuracy for models trained on it. For example, the treatment of outliers could be changed in an effort to retain more data entries. This could involve a thorough investigation of the outlier fields in the database and a manual correction of outlier values where appropriate. However, this would be an extremely time-consuming and manual process.

The third potential avenue for improved accuracy would be an investigation into different imputation and supervised learning techniques. A particular technique may lend itself to performing at a higher level with underwriting data. Imputation techniques such as KNN-impute or the expectation-maximisation with bootstrapping algorithm implemented by Amelia II in R (Honaker et al., 2011) are examples of imputation alternatives. In terms of supervised learning, deep learning techniques such as neural networks or well-known algorithms such as support vector machines may yield improved results. The opportunity for research in this spectrum is diverse, as experimental designs can be created in many ways. As illustrated in the Chapter 2, Miok et al. (2020) provides an example by using Monte Carlo Drop-out Autoencoders to augment the multiple imputation techniques used.

Appendix A

Github Repository

The code used during this research to perform the exploratory data analysis, data pre-processing and modelling can be found at this link:

https://github.com/YPerumal/Msc-coding-R.

References

Acock, Alan C (2005). "Working with missing values". In: *Journal of Marriage and family* 67.4, pp. 1012–1028.

Agarwal, Amulya and Nitin Gupta (2021). "Comparison of outlier detection techniques for structured data". In: *arXiv preprint arXiv:2106.08779*.

Azur, Melissa J., Elizabeth A. Stuart, Constantine Frangakis, and Philip J. Leaf (2011). "Multiple imputation by chained equations: what is it and how does it work?" In: *International Journal of Methods in Psychiatric Research* 20.1, pp. 40–49. DOI: https://doi.org/10.1002/mpr.329. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mpr.329. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mpr.329.

Banton, Caroline (2022). *Underwriting*. URL: https://www.investopedia.com/terms/u/underwriting.asp.

Bennett, Derrick A (2001). "How can I deal with missing data in my study?" In: *Australian and New Zealand journal of public health* 25.5, pp. 464–469.

Bentéjac, Candice, Anna Csörgő, and Gonzalo Martínez-Muñoz (2019). "A Comparative Analysis of XGBoost". In: *arXiv preprint arXiv:1911.01914*.

Biau, Gérard and Erwan Scornet (2016). "A random forest guided tour". In: *Test* 25, pp. 197–227.

Botchkarev, Alexei (2018). "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology". In: *arXiv preprint arXiv:1809.03006*.

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Bronsema, Jan, Sandra Brouwer, Michiel R de Boer, and Johan W Groothoff (2015). "The added value of medical testing in underwriting life insurance". In: *PloS one* 10.12. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0145891.

Brownlee, Jason (2020). *Machine Learning is Popular Right Now*. URL: https://machinelearningmastery.com/machine-learning-is-popular/ (visited on 11/24/2022).

Caplan, Arthur L (2004). *Genetics and life insurance: Medical underwriting and social policy*. MIT Press.

Chai, Tianfeng and Roland R Draxler (2014). "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature". In: *Geoscientific model development* 7.3, pp. 1247–1250.

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Cornell, Portia Y, David C Grabowski, Marc Cohen, Xiaomei Shi, and David G Stevenson (2016). "Medical underwriting in long-term care insurance: Market conditions limit options for higher-risk consumers". In: *Health Affairs* 35.8, pp. 1494–1503.

Davies, Philip and Guy Carrin (2001). "Risk-pooling: necessary but not sufficient?" In: *Bulletin of the World Health Organization* 79.7, pp. 587–587.

Dodge, John H (2007). "Predictive Medical Information and Underwriting". In: *The Journal of Law, Medicine & Ethics* 35, pp. 36–39.

Fontinelle, Amy (2021). *Life Insurance: What It Is, How It Works, and How To Buy a Policy*. URL: https://www.investopedia.com/terms/l/lifeinsurance.asp.

Guo, Rui, Zhiqian Zhao, Tao Wang, Guangheng Liu, Jingyi Zhao, and Dianrong Gao (Sept. 2020). "Degradation state recognition of piston pump based on ICEEMDAN and XGBoost". In: *Applied Sciences* 10, p. 6593. DOI: 10.3390/app10186593.

Hastie, Trevor, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, pp. 9–41.

Holfinger, Steven J, M Melanie Lyons, Brendan T Keenan, Diego R Mazzotti, Jesse Mindel, Greg Maislin, Peter A Cistulli, Kate Sutherland, Nigel McArdle, Bhajan Singh, et al. (2022). "Diagnostic Performance of Machine Learning-Derived OSA Prediction Tools in Large Clinical and Community-Based Samples". In: *Chest* 161.3, pp. 807–817.

Honaker, James, Gary King, and Matthew Blackwell (2011). "Amelia II: A Program for Missing Data". In: *Journal of Statistical Software* 45.7, pp. 1–47. DOI: 10.18637/jss.v045.i07.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2013). *An introduction to statistical learning*. Vol. 112. Springer, pp. 15–57.

Koenig, Gerald and Stephanie Seneff (2015). "Gamma-Glutamyltransferase: A Predictive Biomarker of Cellular Antioxidant Inadequacy and Disease Risk". In: *Disease Markers* 2015.

Kuhn, Max (2008). "Building Predictive Models in R Using the caret Package". In: *Journal of Statistical Software* 28.5, 1–26. DOI: 10.18637/jss.v028.i05. URL: https://www.jstatsoft.org/index.php/jss/article/view/v028i05.

Kunutsor, Setor K (2016). "Gamma-glutamyltransferase—friend or foe within?" In: *Liver International* 36.12, pp. 1723–1734.

Lawson, Richard C., John H. Trout, Ken Krehbiel, and Tom Wilder (1999). *Risk Classification in Individually Purchased Voluntary Medical Expense Insurance*. URL: https://www.actuary.org/sites/default/files/pdf/health/risk.pdf (visited on 12/13/2022).

Lazariv, Taras and Christoph Lehmann (2018). "Goodness-of-fit tests for large datasets". In: *arXiv preprint arXiv:1810.09753*.

Li, Cheng (2013). "Little's test of missing completely at random". In: *The Stata Journal* 13.4, pp. 795–809.

Li, Jiang, Xiaowei S Yan, Durgesh Chaudhary, Venkatesh Avula, Satish Mudiganti, Hannah Husby, Shima Shahjouei, Ardavan Afshar, Walter F Stewart, Mohammed Yeasin, et al. (2021). "Imputation of missing values for electronic health record laboratory data". In: *NPJ digital medicine* 4.1, pp. 1–14.

Magoulas, George D and Andriana Prentza (2001). "Machine learning in medical applications". In: *Machine Learning and Its Applications: Advanced Lectures*. Ed. by Georgios Paliouras, Vangelis Karkaletsis, and Constantine D. Spyropoulos. Springer Berlin Heidelberg, pp. 300–307. ISBN: 978-3-540-44673-6. DOI: 10.1007/3-540-44673-7_19. URL: https://doi.org/10.1007/3-540-44673-7_19.

Maier, Marc, Hayley Carlotto, Freddie Sanchez, Sherriff Balogun, and Sears Merritt (2019). "Transforming underwriting in the life insurance industry". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 9373–9380.

Marais, Andries Francois (2019). "A critical evaluation of the justification of discrimination in risk underwriting in the life insurance industry in South Africa". PhD thesis. Stellenbosch: Stellenbosch University.

Mason, Jennifer E, Rodman D Starke, and John E Van Kirk (2010). "Gamma-Glutamyl transferase: a novel cardiovascular risk BioMarker". In: *Preventive cardiology* 13.1, pp. 36–41.

McCandless, Tyler C and Sue Ellen Haupt (2019). "The super-turbine wind power conversion paradox: using machine learning to reduce errors caused by Jensen's inequality". In: *Wind Energy Science* 4.2, pp. 343–353.

McKeon, Jill (2021). *Machine Learning Model Helps Predict Clinical Lab Test Results*. URL: https://healthitanalytics.com/news/machine-learning-model-helps-predict-clinical-lab-test-results.

Miok, Kristian, Dong Nguyen-Doan, Marko Robnik-Šikonja, and Daniela Zaharie (2020). "Multiple Imputation for Biomedical Data using Monte Carlo Dropout Autoencoders". In: *arXiv preprint arXiv:2005.06173*.

Mishra, MN and SB Mishra (2011). *Insurance Principles and Practice*. S. Chand Publishing,New Delhi.

Munro, A Reg and Anton M Snyman (1995). "The life insurance industry in South Africa". In: *Geneva Papers on Risk and Insurance. Issues and Practice*, pp. 127–140.

Patro, Rebecca (2021). *Cross-Validation: K Fold vs Monte Carlo*. URL: https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b.

Pauly, Mark V and Sean Nicholson (1999). "Adverse consequences of adverse selection". In: *Journal of Health Politics, Policy and Law* 24.5, pp. 921–930.

Pinkham, C Allen and Kenneth J Krause (2009). "Liver function tests and mortality in a cohort of life insurance applicants". In: *Journal of insurance medicine (New York, NY)* 41.3, pp. 170–177.

Pokorski, Robert J. (1997). "Insurance underwriting in the genetic era". In: *Cancer* 80.S3, pp. 587–599. DOI: https://doi.org/10.1002/(SICI)1097-0142(19970801)80:3+<587::AID-CNCR8>3.0.CO;2-6. eprint: https://acsjournals.onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291097-0142%2819970801%2980%3A3%2B%3C587%3A%3AAID-CNCR8%3E3.0.CO%3B2-6. URL: https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0142%2819970801%2980%3A3%2B%3C587%3A%3AAID-CNCR8%3E3.0.CO%3B2-6.

Potthoff, Richard F, Gail E Tudor, Karen S Pieper, and Vic Hasselblad (2006). "Can one assess whether missing data are missing at random in medical studies?" In: *Statistical methods in medical research* 15.3, pp. 213–234.

Raghunathan, Trivellore E, James M Lepkowski, John Van Hoewyk, Peter Solenberger, et al. (2001). "A multivariate technique for multiply imputing missing values using a sequence of regression models". In: *Survey methodology* 27.1, pp. 85–96. URL: https://www.researchgate.net/profile/James-Lepkowski/publication/244959137_A_Multivariate_Technique_for_Multiply_Imputing_Missing_Values_Using_a_Sequence_of_Regression_Models/links/543509d30cf294006f737dca/A-Multivariate-Technique-for-Multiply-Imputing-Missing-Values-Using-a-Sequence-of-Regression-Models.pdf.

Rubin, Donald B (1978). "Multiple imputations in sample surveys-a phenomenological Bayesian approach to nonresponse". In: *Proceedings of the survey research methods section of the American Statistical Association*. Vol. 1. American Statistical Association Alexandria, VA, USA, pp. 20–34.

— (1987). *Multiple imputation for nonresponse in surveys*. Vol. 81. John Wiley & Sons.

Schafer, Joseph L and John W Graham (2002). "Missing data: our view of the state of the art." In: *Psychological methods* 7.2, p. 147.

Stekhoven, Daniel J. (2022). *missForest: Nonparametric Missing Value Imputation using Random Forest*. R package version 1.5.

Stekhoven, Daniel J and Peter Bühlmann (2012). "MissForest—non-parametric missing value imputation for mixed-type data". In: *Bioinformatics* 28.1, pp. 112–118.

van Buuren, Stef (2018). *Flexible Imputation of Missing Data. Second Edition.* Boca Raton, FL.: CRC Press.

van Buuren, Stef and Karin Groothuis-Oudshoorn (2011). "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45.3, 1–67. DOI: 10.18637/jss.v045.i03. URL: https://www.jstatsoft.org/index.php/jss/article/view/v045i03.

Venter, Jannie (2015). *Absa Life Pioneers Predictive Medical Underwriting solution using Big Data*. URL: https://www.fanews.co.za/article/life-insurance/9/general/1202/absa-life-pioneers-predictive-medical-underwriting-solution-using-big-data/17800.

Vroon, David H. and Zafar Israili (1990). *Alkaline Phosphatase and Gamma Glutamyltransferase*. URL: https://www.ncbi.nlm.nih.gov/books/NBK203.

Wang, Yafei (2021). *Predictive machine learning for underwriting life and health insurance*. URL: https://www.actuarialsociety.org.za/convention/wp-content/uploads/2021/10/2021-ASSA-Wang-FIN-reduced.pdf.

Willmott, Cort J and Kenji Matsuura (2005). "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". In: *Climate research* 30.1, pp. 79–82.

Ye, Andre (2020). *MissForest: The Best Missing Data Imputation Algorithm?* URL: https://miro.medium.com/max/720/1*m_z8E4HrFtCnHBoDANauTQ.webp.

Zhang, Xinmeng, Chao Yan, Cheng Gao, Bradley A Malin, and You Chen (2020). "Predicting missing values in medical data via XGBoost regression". In: *Journal of Healthcare Informatics Research* 4.4, pp. 383–394.