

Predictive machine learning for underwriting life and health insurance

Yafei (Patricia) Wang

Presented at the Actuarial Society of South Africa's 2021 Virtual Convention
19–22 October 2021

ABSTRACT

Underwriting is a risk assessment process that classifies insurance applications into different categories. Traditional underwriting is costly, time-consuming and perceived as a barrier for the underserved population. Despite many attempts to automate life and health insurance underwriting, the dominant approach is a mix between rule-based engines and traditional underwriting. Applications are first assessed by rule-based engines – approximately a third of all applications are processed successfully by these engines. The remaining applications are reviewed by underwriters or even referred to the reinsurers. The rule-based engines have no predictive ability of applications that do not fit into the existing rules encoded into the engines. Therefore, the natural-step improvement to the current approach is predictive machine learning models.

This research aims to construct predictive machine learning models to predict underwriting decisions for life and health insurance applications, using reinsurer data that are predominantly applications with complex medical conditions and large sum insured. The models are designed to provide an end-to-end solution, so machine learning techniques such as natural language processing and clustering analysis are used to process real-world data; in particular, free-text descriptions of impairments and occupations, which the traditional statistical models cannot process. Text mining tools such as word clouds are used as part of preliminary analyses and give some insights into medical underwriting. Various feature selection methods such as mutual information and recursive

feature elimination are used to improve prediction accuracies. Lastly, machine learning algorithms such as XGB, Random Forest and bagging are used to predict the underwriting decisions.

The accuracies of various machine learning models are then compared: the extreme gradient boost algorithm with a combination of feature selection methods performs the best, with 94% accuracy on the training set and 71% accuracy on the testing set. This result is a significant improvement from the rule-based engines that can process only a third of the applications. The feature ranking function of the extreme gradient boost algorithm gives additional underwriting insights such as the features that are the most important for the model to determine underwriting decisions. The remarkable increase in prediction accuracies, as well as additional functionalities such as feature ranking, indicate that predictive machine learning models could be used to improve the current underwriting process significantly.

KEYWORDS

Machine learning, underwriting, data analytics, natural language processing, word cloud, feature selection, decision tree, random forest, extreme gradient boosting, bagging, Adaboost

CONTACT DETAILS

Patricia Wang, patriciawangyafei@gmail.com

LinkedIn: <https://www.linkedin.com/in/patricia-wang-1446894a/>

1. INTRODUCTION

1.1 Background

Underwriting is a risk assessment process that categorises insurance applicants into different risk classes. According to the Chartered Insurance Institute, underwriters need to ‘assess the risk decide whether or not to accept the risk, or how much to accept, determine the terms, conditions and scope of cover to be offered; and calculate a suitable premium’ (Chartered Insurance Institute, 2020).

In the case of life and health insurances, underwriters typically make decisions based on various risk factors such as the applicant’s age, health status, lifestyle, occupation, medical history, the type of products and the sum insured. For a healthy applicant, the underwriting decision is typically to accept the application with standard terms and no further information is required; for an individual in poor health, the decision is either to decline or to accept the application with extra premium expressed as a percentage of the standard premiums. Underwriters also decide the magnitude of this percentage, for example, an additional 125% of the standard premiums due to the additional risk that the applicant carries. This percentage is called loading. In summary, underwriters need to decide if applications should be accepted or declined, and the loading to be applied to the standard rates if applicable.

The dominant automated underwriting approach is rule-engines storing rules translated from the existing underwriting guidelines; alternative approaches such as predictive models are uncommon (Batty & Kroll, 2009: p.11). However, only a finite number

of rules can be stored in a system, so rule-engines cannot process applications that do not fit into any existing rules. In this situation, underwriters will be called upon to review the application, and they may even refer the case to the reinsurer(s) for an opinion if the medical conditions of the applicant are too complex, or the sum insured is too large.

The current underwriting process can take 3–4 weeks on average (Cornelius, 2015), because, underwriters may need additional medical test results such as CT scans for many applications. The existing process is very inefficient and has several undesirable consequences:

- The underwriting process ‘is often cited as a barrier for underserved populations to insure themselves’ (Batty & Kroll, 2009: p.4), partially because medical tests cost money before the applications are even accepted.
- The review process is lengthy, which impacts negatively on the customer experience and could lead to lost business opportunities, for example, ‘only 40% of U.S. households own individual life policies’ (Prudential, 2015).
- Underwriting decisions could be prone to human errors, particularly those complicated cases that are accessed by human underwriters.

1.2 Objectives

The life insurance companies that have successfully implemented automated underwriting are not using it for medical underwriting; instead, the automated system is used for ‘simplified issue and non-medical underwriting’, or ‘small- to medium-sized policies’ (Batty et al., 2010: p.3). Life insurers that have tried to replicate medical underwriting using automated systems have had little success and have found that underwriters still need to review many applications (Batty et al., 2010: p.4). Companies that implemented automated medical underwriting systems found it challenging to issue more than one-third of policies directly from the system (Batty et al., 2010: p.11). In other words, this means that the rule-engines have no predictive ability if an application does not fit into any of the rules in the system. In a loose sense of accuracy, rule-engines can make correct underwriting decisions for only approximately a third of all applications.

This research aims to assess whether predictive machine learning models can be used in the context of life and health insurance policies, in particular, medical underwriting of applications with complicated medical conditions and large sum insured.

Desirable outcomes of this research are to build predictive machine learning models that can:

- achieve high accuracy of predictions of underwriting decisions from the machine learning algorithms, given rule-engines can make correct decisions for only a third of the applications (Batty et al., 2010: p.11);
- label applications such that, if the insurer is mainly concerned with high-risk applications, the labels can be used to flag these policies;
- provide underwriting insights that can be used to improve the current process; for example, models can reveal features that have high predictive powers of underwriting decisions.

Data containing free-text descriptions of medical conditions and occupations are pre-processed using machine learning techniques such as natural language processing and k-means clustering analysis.

Several predictive machine learning algorithms are then constructed using processed data. These include Random Forests, Decision Tree, Gradient Boosting, Extreme Gradient boosting, bagging algorithm, AdaBoost, Support Vector Machine, Stochastic Gradient Descent, K Neighbours, and ordinal logistic regression. Both classifier and regression models are constructed and compared in this research. A broad range of models is constructed to ensure that the best-performing model achieves a high accuracy score.

A secondary objective is to draw insights from the modelling results to make actionable business recommendations. For example, the modelling results can rank features by their significance at predicting underwriting decisions, so insurers should capture these variables as accurately as possible. On the contrary, some features have negligible impacts so they could be omitted to reduce the amount of application processing time.

1.3 Overview

Section 2 discusses the existing research pertaining to various aspects of this topic, although there is no extensive literature on the exact subject of this research. Section 3 outlines the methodologies and briefly discusses the rationale for employing various machine learning techniques. Section 4 introduces the datasets used for this research, details the machine learning techniques used for preprocessing free-text data, and ends with preliminary data analyses. Section 5 explains various methods used for feature selection and dimension reduction. Section 6 discusses the models, compares their performances, and interprets the results. Section 7 discusses the business applications of the model results and the adaptations for underwriting for specific insurance products. Section 8 concludes this research, discusses the limitations of this research, and makes suggestions for further research.

2. LITERATURE

2.1 Data source and treatment of missing data

There is a lack of public life insurance data with sufficient detail and dimensions, and the majority of existing research uses online databases that are pre-anonymised and preprocessed in the sense that variables in these databases are already numerical or categorical. However, real data are rarely strictly numerical and categorical. For example, descriptions of medical conditions and occupations in the applications are texts, and a large part of underwriters' work is to read through descriptions of medical impairments and decide if an insurable medical condition is applicable. Similarly, applicants do not come with labels of occupation classes, and underwriters need to decide which occupational classes applicants belong to, based on their job descriptions. Being able to handle free-text data is thus an essential feature of an implementable predictive model that provides an end-to-end business solution. Existing research using datasets that are already in numeric formats skip this functionality of processing non-numerical variables entirely. In fact, insurers found

desirable underwriting systems should ‘have a broad array of functionalities including processing medical lab tests’ and ‘more developed reflective applications (which provide details into impairments)’ (Batty et al., 2010: p.11).

Furthermore, without the knowledge of how original data are transformed into numerical and categorical variables, it is difficult to make assumptions about the nature of missing data. Thus, the treatment of missing data amongst the existing research could be misguided. Missing values can be one of the three scenarios:

- Missing Completely At Random (MCAR): the missing and the observed values have no systemic difference;
- Missing At Random (MAR): the systemic difference between missing and observed values can be explained and thus predicted by the observed values;
- Missing Not At Random (MNAR): the systemic difference between missing and observed value cannot be predicted by the observed values (Sterne et al., 2009).

In their paper, Boodhun and Jayabalan (2018) assumed that the missing values were MAR because the visualised missing values appeared to be random, so multiple imputation techniques were used to replace the missing data. However, MAR may not be valid: Boodhun and Jayabalan identified 13 variables with missing values, all of which are medical conditions, family medical history or insurance history, which are numbered but not described, so users do not know what each variable means. For example, ‘Medical_Condition_10’ is identified as having the most missing values, but it is unknown what exact condition this variable is.

The meanings of variables with missing values are important because it is entirely plausible that these missing values are MNAR and cannot be imputed. For example, ‘Medical_Condition_10’ has 58824 missing values out of 59381 data points, so 99% of the values are missing. It is therefore likely that ‘Medical_Condition_10’ is a rare medical condition that most applicants never heard of and did not have, so they left the field blank. In this case, data are missing precisely because this medical condition is not applicable. Similarly, insurance history may have been left blank because the applicant has not bought any prior insurance. Therefore, MNAR is more plausible than MAR.

In their paper, Jakobsen et al. (2017) discussed the practical considerations of using imputation techniques and summarised common scenarios where imputation is not appropriate, and these scenarios include:

- the percentage of missing data is either negligible (less than or equal to 5%);
- the percentage of missing data is significant (greater than or equal to 40%);
- data are only missing on the outcome variables;
- neither MCAR nor MNAR is plausible.

When MAR is not plausible, using multiple imputation could lead to bias in the substantive analysis (Sterne et al., 2009).

Furthermore, Boodhun and Jayabalan used the Multivariate Imputation via Chained Equations (MICE) package in R to carry out imputation (Boodhun & Jayabalan, 2018), and

this package is developed with the assumption that the missing values follow multivariate normal distributions (van Buuren & Groothuis-Oudshoorn 2011). Given that all the missing values in Boodhun and Jayabalan's dataset are binary dummy variables, the multivariate normality assumption may be invalid, in which case using MICE R package to replace missing values may be inappropriate.

2.2 Feature selection and dimension reduction

Dimension reduction projects the original data into a lower-dimensional space, and Principal Component Analysis (PCA) is one of the most commonly used dimension reduction methods, which generates new features from the original features with minimal loss of explained variance (Sarveniazi, 2014). It is an excellent tool for increasing computational efficiency and visualising the decision boundary if the dimensions can be reduced to two or three components. However, for high-dimension data with hundreds of features, the interpretation of new features created can be obscure and difficult to explain, particularly in a business context. It is even more challenging to draw any actionable business recommendations.

Unlike PCA, feature selection does not create new features but selects a subset of the original features. Several feature selection methods are discussed below.

Filter methods can be based on correlation or mutual information. Correlation-based filter methods may use the correlations between input variables only and eliminate the input variables that are highly correlated because they are redundant. Mutual information methods typically select the features based on entropies between input and output variables. Filter methods are computationally efficient, less prone to overfitting, and independent from the machine learning algorithm used for modelling. However, they may include redundant features that are highly correlated to the output variable, or eliminate features that are not highly correlated to the outcome variable in isolation, but provide useful information when used jointly with other features (Chandrashekar & Sahin, 2014). The modeller should apply their domain knowledge to check features to be eliminated by filter methods before removing them from the data.

Wrapper methods interact with the machine learning algorithm used for modelling. Wrapper methods are more accurate than filter methods but more computationally expensive and prone to overfitting (Chandrashekar & Sahin, 2014). It also requires the modeller to predefine the number of features to be selected. Given that insurance data are typically large with high dimensions and many noisy features, without knowing the optimal number of features at the outset, using wrapper methods is impractical for this research.

Recursive Feature Elimination (RFE) is essentially a backward selection method that works particularly well with tree-based algorithms such as Random Forest (Kuhn & Johnson, 2019). RFE trains the model, computes the feature importance using a score function, and eliminates the least important features recursively (Kuhn & Johnson, 2019). An additional advantage is that RFE can be implemented with cross-validation in sklearn, which reduces overfitting. Lastly, RFE with cross-validation does not require the modeller to predefine the

number of features to be selected; instead, the method will determine the optimal number of features.

Each feature selection or dimension reduction method has its advantages and drawbacks. This research considers various methods, assesses each method concerning its practicality, efficiency, accuracy and applicability in the business context. This research uses a combination of feature selection methods, and the rationale is discussed in more detail in section 5.2.

2.3 Machine learning models

This section gives a brief overview of the more advanced or less familiar machine learning algorithms used for modelling. Some well-known algorithms, such as logistic regression and decision tree, are not discussed even though they are used for this research.

Instead of growing one classification tree in the case of Decision Tree (DT), Random Forest (RF) grows many trees, each of which is constructed using a subset of the data samples and a subset of features. The data subsets and feature subsets are chosen at random, so RF is less prone to overfitting than DT. RF performs well on imbalanced data, or large high-dimensional datasets with missing values, and can rank features by their importance (Breiman et al., 1984).

AdaBoost trains weak learners on weighted versions of the dataset and combines the outputs through a weighted majority vote to produce the final output. Higher weights are given to more accurate learners (Hastie et al., 2009). Traditionally AdaBoost handles multiclass classification by converting the problem into multiple binary classification problem; Zhu et al. (2006) proposed an algorithm ‘SAMME’ that extends AdaBoost into a single multiclass problem and ‘adaptively implements the multiclass Bayes rule by fitting a forward stagewise additive model’. In their paper, Zhu et al. (2006) also demonstrated that SAMME outperforms traditional AdaBoost slightly on six benchmark datasets. For this research, the SAMME algorithm is implemented to maximise the performance of AdaBoost.

Gradient boosting is similar to AdaBoost but aims to minimise the residual errors by optimising the loss function at each iteration (Friedman, 2001).

XGBoost is a scalable tree boosting algorithm based on gradient boosting. It has several improvements compared to the gradient boosting: for example, XGB adds a penalty term to the loss function, so XGB is designed to select models with less complexity to avoid overfitting (Chen & Guestrin, 2016). Its other advantages include its ability to handle sparse data, and the parallel and distributed computation that increases learning speed (Chen & Guestrin, 2016).

Bagging algorithms are similar to boosting algorithms in the sense that they also combine weak classifiers to increase the performance of the final output. Bagging algorithms train weak learners in parallel independently on randomly bootstrapped samples of the original training set, whereas boosting algorithms adjust weights to the weak learners sequentially (Quinlan, 1996). In his paper, Dietterich (2000) found that boosting performs better than bagging on datasets with little noise, but bagging performs better on datasets

with substantial noise. However, Dietterich's paper mainly compared Adaboost and bagging, and it was published before the invention of XGB. The comparison of these two ensemble algorithms is discussed further in section 6.

Support Vector Machine (SVM) is a popular classification algorithm that transforms non-linear input variables into a feature space, and a decision boundary is constructed (Cortes & Vapnik 1995). This algorithm requires the modeller to specify the kernel function to be used. A comparison of different kernel functions for SVM-based classification shows that the Radial Basis Function (RBF) kernel performs the better than linear and polynomial kernels (Yekkehkhany et al., 2014).

Stochastic gradient descent (SGD) is a popular optimisation algorithm that uses approximate gradients estimated from samples of the training data and updates the parameters iteratively (Tsuruoka et al., 2014). When the loss function is specified as 'log', the algorithm effectively trains logistic regression through SGD. L1 regularisation is better L2 in practice since many of the weights of unimportant features can be zero under L1 and more compact model is selected (Tsuruoka et al., 2014). Indeed, it has been demonstrated empirically that SGD for L1-regularised log-linear models performs well on several high-dimension datasets (Tsuruoka et al., 2014).

3. METHODOLOGY

The methodologies used for this research are illustrated in Figure 1.

The first step is to process free-text data, in particular, descriptions of impairments and occupations. Free-text data are typically entered into the data-capture system manually by administrators. Such unstructured data contains words of the same meaning but in different variations. For example, alcoholism may be described as 'excessive alcohol consumption', 'alcoholic', and so on. In addition to different variations, free texts are prone to typing errors or other issues associated with natural languages. Typically, underwriters can recognise these errors and different word variations or synonyms. Natural language processing is used to imitate this process, so words appear in lower case and their word origins. Lastly, some descriptions of impairments or occupations do not exist in the medical condition list or occupation list, and underwriters need to apply judgements as to which medical condition(s) and occupations are applicable. Clustering analysis is applied to imitate this thought process.

In short, the first step aims to imitate the process that underwriters use to correct errors in text data and apply judgements to these data. The machine learning techniques are discussed in detail with examples in section 4.2.

The second step is to transform the text data into numerical data, so further statistical analysis and modelling can be carried out. This step requires knowledge of how data are captured and what each column in the data means.

The third step is to look at the missing data and decide on the appropriate treatment of the missing data, as well as carry out preliminary data analysis. This step is typically the starting point of previous research.

The next step is to apply various machine learning techniques for dimension reduction

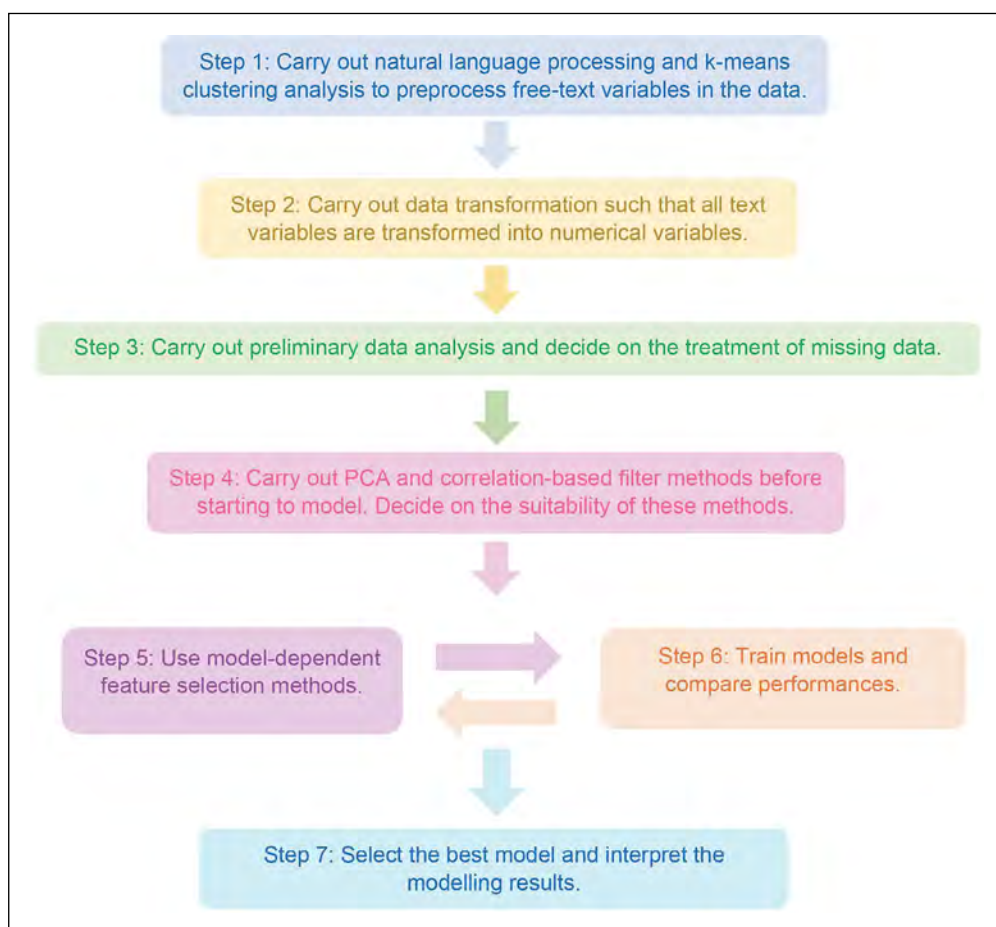


FIGURE 1 Methodology flowchart

and feature selection. PCA is used to demonstrate that it is not appropriate in the context of insurance data. Correlation-based filters are applied to select a subset of features before modelling because these feature selection methods are independent of the underlying models. The threshold for feature elimination is set to be very conservative, and the features to be eliminated are sensibility-checked before being removed. This ensures that no important feature is deleted before model training.

In addition to correlation-based filter methods, other feature selection methods such as mutual information method and recursive feature elimination (RFE) are used for this research in step 5. The model training is performed on various subsets of features selected by a combination of these feature selection methods in step 6. Steps 5 and 6 are performed iteratively because some feature selection methods such as RFE are dependent on the machine learning models. Furthermore, some feature selection methods such as mutual information method require the modeller to specify the number of features to be

selected, but this number is unknown at the outset. Iterative selection compares the model performances using various numbers of features and eliminates the need to predefine the number of features to be selected. A combination of various feature selection methods is relatively computationally efficient and avoids the pitfalls of each method used in isolation.

Ten machine learning algorithms are considered for this research. Underwriters firstly decide whether the application can be accepted; if an application is too risky, it will be declined, and no further assessment is required, so this is a binary classification problem. Next, underwriters need to decide whether loadings need to be applied to the accepted applications; if an application is accepted on standard rates, no further assessment is required, so this is also a binary classification problem. Lastly, underwriters need to decide how much loading to apply for those applications that have higher risks than standard or preferred applications, but the risks are still acceptable. The loadings are multiples of 25% and rarely exceed 400% because a loading higher than 400% would mean the application is too expensive for the applicant or too risky to accept for the insurer. On the rare occasions that loadings exceed 400%, the application would be heavily scrutinised by underwriters on a case-by-case basis. In other words, loadings can only be 25%, 50%, 75% up to 400%. As a result, the underwriting decision can be modelled as a multiclass classification problem. Class labels and descriptions are summarised in Table 1.

TABLE 1 Class labels and descriptions

Class Label	Description
0	loading is 0, the application is accepted on standard terms
1	loading is $1 * 25\% = 25\%$, the application is accepted with loading 25%
2	loading is $2 * 25\% = 50\%$, the application is accepted with loading 50%
3	loading is $3 * 25\% = 75\%$, the application is accepted with loading 75%
4	loading is $4 * 25\% = 100\%$, the application is accepted with loading 100%
5	loading is $5 * 25\% = 125\%$, the application is accepted with loading 125%
6	loading is $6 * 25\% = 150\%$, the application is accepted with loading 150%
7	loading is $7 * 25\% = 175\%$, the application is accepted with loading 175%
8	loading is $8 * 25\% = 200\%$, the application is accepted with loading 200%
9	loading is $9 * 25\% = 225\%$, the application is accepted with loading 225%
10	loading is $10 * 25\% = 250\%$, the application is accepted with loading 250%
11	loading is $11 * 25\% = 275\%$, the application is accepted with loading 275%
12	loading is $12 * 25\% = 300\%$, the application is accepted with loading 300%
13	loading is $13 * 25\% = 325\%$, the application is accepted with loading 325%
14	loading is $14 * 25\% = 350\%$, the application is accepted with loading 350%
15	loading is $15 * 25\% = 375\%$, the application is accepted with loading 375%
16	loading is $16 * 25\% = 400\%$, the application is accepted with loading 400%
20	the application is accepted with loading greater than 400%
100	declined applications

There are several reasons that underwriting decisions, including loadings, could be considered as a multiclass classification problem. Firstly, the natural underwriting decision process resembles a decision tree, so it fits the nature of classification models better, in particular, tree-based models. Even with variable transformation, the output variable, underwriting decision, and the input variables are unlikely only to have linear relationships for which regression models are more appropriate. Secondly, the loadings have discrete values as multiples of 25%, so they are not continuous variables and thus need to be modified to be modelled by classic regression models. As explained above, the loadings are one of 16 possible values, and rarely exceed 400% so they behave more like ordinal variables which can be modelled by classification models. Lastly, underwriting assessment stops when an application is declined, and no loadings would be assigned to declined applications. If loadings were to be modelled using regression models, artificial loadings need to be assigned for the declined applications for modelling. In this case, declines application is labelled as '100'.

Ordinal logistical regression models are included in the research because they are well studied and understood, so they are useful for drawing insights. For this research, all major machine learning algorithms are trained, including logistic regression models and classifiers such as decision tree and random forests.

The prediction accuracies are computed to compare model performances. Micro precision scores, as opposed to macro precision scores, are computed because the dataset is imbalanced, as shown in Figure 2.

The micro-average precision score is calculated from the individual classes' true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs); whereas the macro-average F1-score is calculated as the arithmetic mean of individual classes' F1-

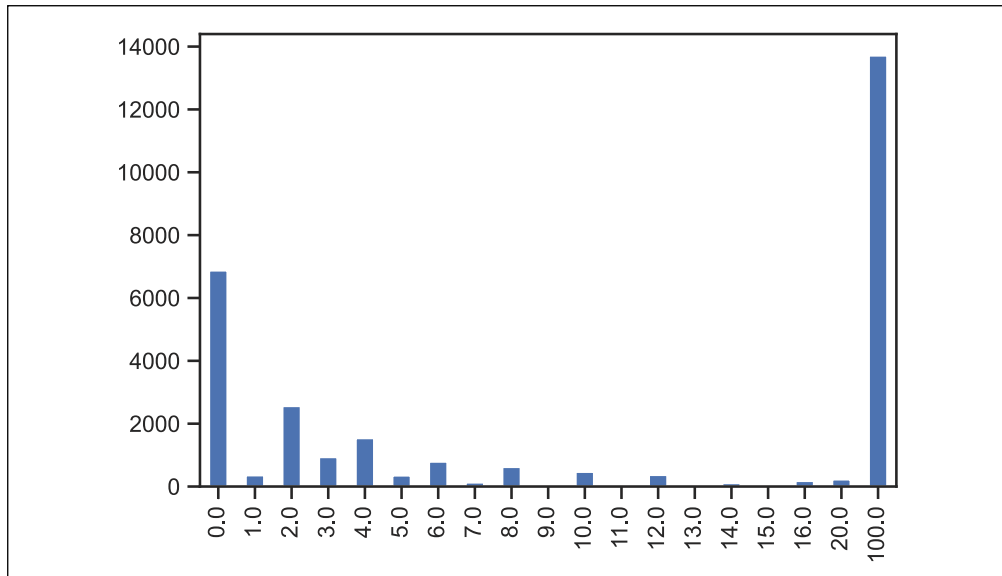


FIGURE 2 Frequencies of class label

score. Using macro scores therefore would give more weight to smaller classes. In some circumstances, the macro score may be considered more appropriate, for example, the user is concerned with the underwriting decisions for risky classes only, and might want to assign a high weight to class '20'. However, this research aims to achieve a high prediction for all applications; therefore, high weights for smaller classes are inappropriate because the accuracy scores would be biased towards small classes with sparse data.

4. DATA

This section discusses various aspects of data, with an emphasis on section 4.2, which explains in detail how data are preprocessed. In practice, predictive machine learning algorithms are uncommon for modelling underwriting decisions, partially because the real-world data contain free-texts, so several machine learning techniques are used to resolve this issue.

4.1 Data source

The data used for this research are a reinsurer's dataset. Reinsurers act as the insurers for insurance companies, so typically policies with complex medical conditions and large sum insured are referred to reinsurers. This means that the dataset is skewed towards the applications with complex medical underwriting risks and large sums insured. Although the dataset does not represent the population, this is advantageous for the aim of this research – assessing the applicability of predictive machine learning models in replicating medical underwriting and for applications with a large sum insured.

The primary dataset contains 29317 applications from January 2017 to June 2020. There are 37 columns in the data, several of which are free-text variables. The detailed descriptions of these columns in this dataset are summarised in Appendix 1. This dataset contains information about the applicants such as age, gender, weight and height, their health status such as impairments, the product information such as sum insured. Each row is an application, and each column is a feature of applications.

Two supplementary data files are used to assist in processing the primary data: the first is a list of medical conditions and their associated symptoms for underwriting. The second supplementary data file is an occupation list that categorises different jobs into four occupational classes. Occupational class 1 is the lowest risk group and class 4 is the highest.

The two supplementary lists are in-house lists based on medical research and experience analysis. Insurers or reinsurers typically have similar in-house lists that reflect their underwriting guidelines.

4.2 Pre-processing

Two columns in the primary dataset are free-text data that require preprocessing: 'AssessmentImpairment' and 'Occupation'. Appendix 10.2.1 is an extract of the primary dataset, showing these two columns for several applications.

Column 'AssessmentImpairment' contains descriptions of impairments or symptoms

of each applicant; if this column is empty, it means that the applicant does not have any known pre-existing medical condition. Typically, underwriters would match the impairments in the application to the impairments in the medical condition list, which is mapped to a medical condition; this medical condition is then used with other information such as age and gender to look up the correct loading in the rule-engines. For example, the impairment description in the second row is 'back pain'; this exact impairment can be found in the medical condition list, and it is mapped to condition 'skin, muscles, bones, joints, or connective tissue'.

Sometimes, spellings or word variations in the application may be different from those used in the condition or occupation list. For example, an impairment in the primary data (Appendix 10.2.1), 'Hodgkin disease', does not match any of its variations in the condition list in Appendix 10.2.2. This problem is resolved by using natural language processing which converts all words into lower case, removes all punctuation, converts all words into their word origins through stemming and lemmatisation so the exact match can be found.

If an exact match is not found, underwriters will check for similar impairments listed in the medical condition list to find a suitable condition. For example, an impairment in the primary data, 'wrist complaints', is not one of the listed impairments. However, searching for 'wrist' in the condition list would yield several results such as 'wrist break', 'wrist injury', and 'wrist pain', all of which are mapped to condition 'skin, muscles, bones, joints, or connective tissue'. Given the vague description 'wrist complaints', it can be reasonably deduced that condition 'skin, muscles, bones, joints, or connective tissue' is applicable. This thought process of an underwriter is imitated using a Python function 'get_close_matches' which returns a list of close matches to the word of interest, sorted in descending order of similarity scores. The closest match is used for mapping to a suitable condition. In the example here, the closest match that the Python function returns is 'wrist pain', which is what a human underwriter would have used too.

If neither an exact nor similar impairment is found, underwriters would typically apply judgements to find an applicable condition. This process is imitated by clustering analysis: all impairments in the condition list are grouped by their corresponding conditions, so 35 clusters are created to represent 35 conditions, each of which contains many impairments all belonging to the cluster. Cluster analysis is performed on impairments without an exact or a close match, so they are mapped to the most appropriate condition cluster given the impairments. For example, an impairment in the primary data is 'underweight' for which neither an exact nor a close match can be found. However, condition 'other' includes multiple impairments such as 'weight loss' and 'slimming', so an underwriter would map 'underweight' to 'other' condition. Indeed, the clustering analysis maps 'underweight' to 'other' as an underwriter would.

The occupations in the primary data are processed similarly by natural language processing and clustering analysis, so they are mapped into the correct occupational classes. If no occupation information is available in the application, the occupational class is labelled as 0, so the coefficient of occupational class in a regression analysis has no impact on applications with no occupation data.

After mapping impairments and occupations into the right conditions and occupational classes, 35 dummy variables are created for each of the 35 conditions, and one categorical variable is created for occupational class with values 0 to 4.

The resulting primary data used for modelling contain the output variable ‘loading’ and 118 input variables or features.

4.3 Missing data

There are five variables with missing values, and their correlations with one another are visualised in Figure 3. BMI is calculated as the weight divided by the square of the height measured in metres, so it is highly correlated with height and weight. There are some missing values in ‘EntryAge’ and ‘FaceAmount’, but these two are not correlated with each other or with BMI, height or weight.

The percentages of missing values are examined and illustrated in Appendix 3. The percentages of ‘EntryAge’ and ‘FaceAmount’ missing are less than 1%, so they are negligible and deleting these missing values would not affect the modelling results. However, the missing ‘BMI’, ‘Height’ and ‘Weight’ are approximately 40%.

Furthermore, the correlations of missing variables with other variables are computed and illustrated in a correlation matrix (Appendix 4), which reveals that the missing ‘BMI’, ‘Height’ and ‘Weight’ can be entirely attributed to a few companies, possibly as a result of these companies’ data capturing issues or their application form design. This pattern suggests that the values are MNAR and cannot be imputed.

Due to large percentages of missing values, and that MNAR is plausible, imputation to replace missing values is not appropriate, so only complete data are used for modelling.

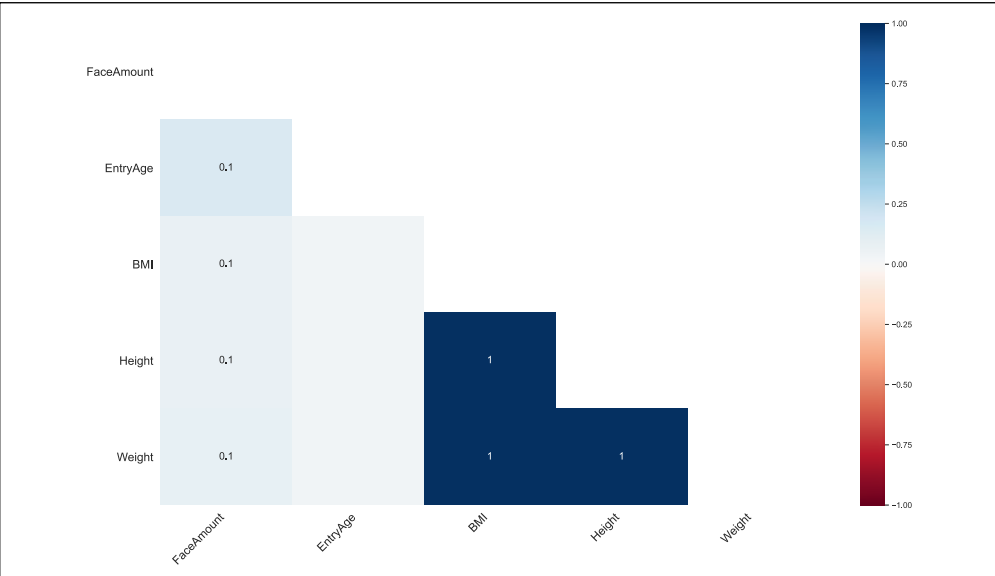


FIGURE 3 Variables with missing values and their correlations

4.4 Preliminary data analysis

There are only five continuous numerical input variables, and pairwise scatterplots of these five variables are shown in a matrix (Appendix 5). As seen in the scatterplots, if entry age is high, that is, the applicant is over 60, the application is more likely to be declined than younger applicants in their 30s or 40s. There are some outliers in BMI, height, weight, 'FaceAmount' and 'EntryAge'. These are data errors, for example, BMI smaller than ten or greater than 100 are erroneous, so they are removed.

The other input variables are either dummy variables or categorical variables. Due to large dimensionality, plotting pairwise scatterplots in one matrix would result in visual clutter, so only six variables are chosen as examples. The scatterplot matrix is attached in Appendix 6. The plot of 'DateInputYear' suggests that there were more declined cases in 2017 than other years, suggesting a change in the underwriting guideline. The blue line, which represents accepted applications on standard terms, peaks for applicants with no heart-related conditions, suggesting that this medical condition is an important feature for modelling underwriting decisions.

After removing missing values and outliers, there are 18182 applications in the data.

5. FEATURE SELECTION

5.1 Principal component analysis

PCA aims to transform a high-dimensional dataset into a lower-dimensional dataset with minimal loss of explained variance. Figure 4 shows the explained variance against the number of new features derived from the original features using PCA.

It is clear that at least 100 new features are required to obtain near 100% explained variance, and at least 85 new features are required to obtain approximately 95% explained variance.

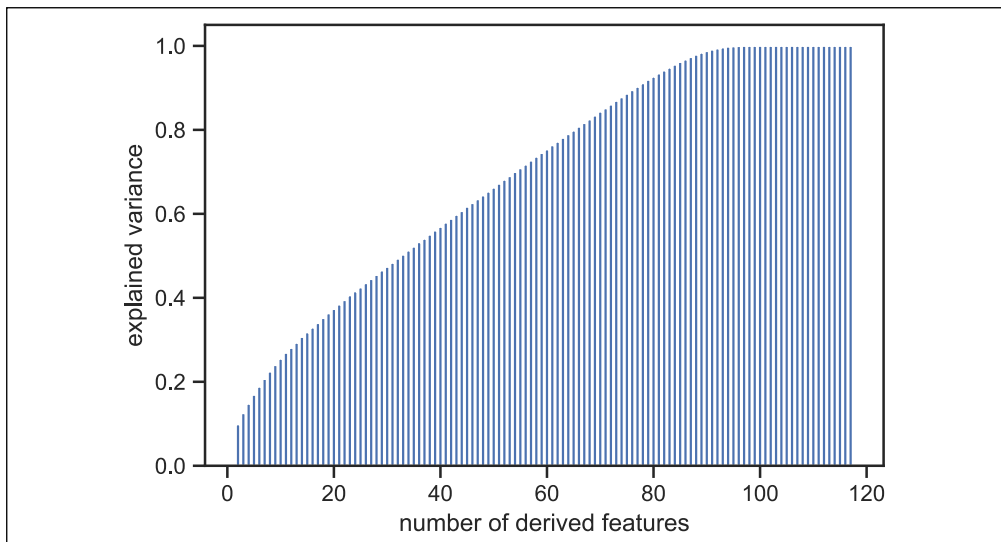


FIGURE 4 Explained variance vs number of derived features for PCA

variance, so 5% of the variance is lost before model training. PCA is not useful at reducing the number of features for this dataset.

Another use of PCA is visualising the decision boundary that separates different classes when the original dataset is projected into a two-dimension feature space. For demonstration purposes, four class labels are used to avoid visual clutter. These four classes are shown as scatterplots in a two-component PCA in Figure 5.

There is no clear decision boundary between these four classes, and the boundaries would be even less clear with all 19 classes shown.

Since PCA is not useful at reducing the dimensions or visualising decision lines, and the interpretation of new features generated is obscure, PCA is not used.

5.2 Feature selection methods

This research uses a combination of correlation-based filter methods, mutual information, and recursive feature elimination. These methods are chosen with the following considerations:

- **Practicality:** some methods such as wrapper methods are too computationally expensive to be practical for insurance data with large dimensions; the three methods used are relatively computationally efficient.

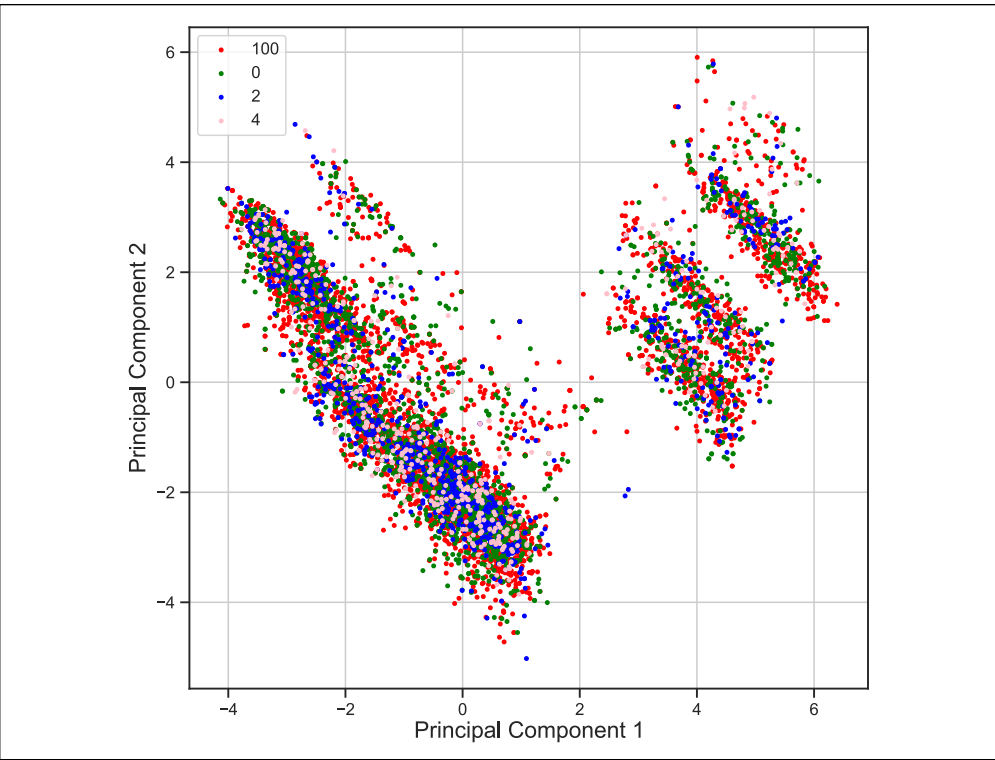


FIGURE 5 Two-component PCA

- **Parameter-tuning:** the recursive method can determine the optimal number of features to be selected; the mutual information method is adapted to do so using the loop described below.
- **Accuracy:** correlation-based filter methods may eliminate features which are important, and the candidate features are checked before being removed. The other two methods are fairly accurate.
- **Business applicability:** PCA is considered but not used; the rationale is discussed above.

First, correlation-based filters are applied to remove features with constant values.

Next, the correlations between all the input variables are computed. The variables with correlations higher than 0.97 are deemed redundant and candidates for removal. The threshold is set at 0.97 to be conservative to avoid deleting important variables. Four variables have correlations higher than 0.97 with preceding variables: 'FollowUp', 'LumpSum', 'ADB' and 'Mortality'. 'FollowUp' is a dummy variable indicating whether the application is a follow-up or a new application and equals to 1 minus the preceding indicator 'New', so 'FollowUp' can be removed. For example, if the application is a follow-up, the follow-up indicator for this application would be 1 and the 'new' indicator for this application would be 1 minus 1 which equals to zero and vice versa. 'ADB' stands for 'Accidental Death Benefit' so it is the same as 'Accident' and thus can be removed. 'Mortality' and 'LumpSum' are highly correlated because products exposed to mortality risk are always paid as a lump-sum benefit. Of the 18182 applications, 14154 are exposed to mortality risk, all of which have lump-sum benefits, and there are 14323 applications with lump-sum benefits. Their relationships are illustrated in Figure 6. 'Mortality' can be treated as a baseline category of product risk when other product risk indicators such as 'Disability' are in the feature subset; thus, 'Mortality' is removed and 'LumpSum' is kept. After a reasonability check, three out of four candidate features are removed, which results in a subset of 107 features.

RFE with cross-validation has the capability of determining the optimal number of features, so it is used next. Five-fold cross-validation is used to reduce overfitting. RFE is model-dependent, and XGB is used as the underlying algorithm. As discussed in the literature

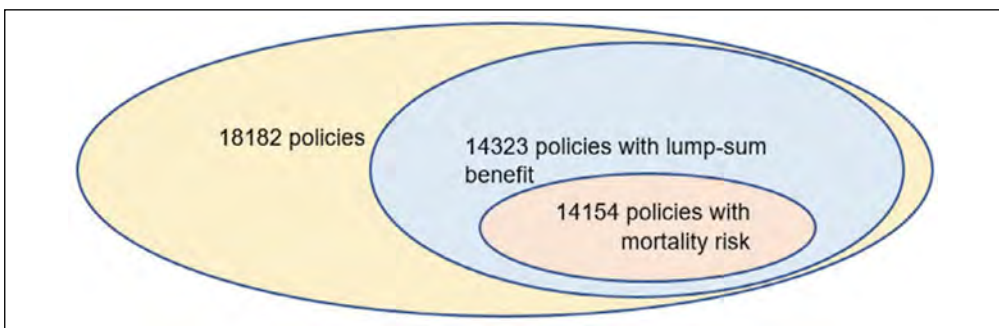


FIGURE 6 Relationship between 'LumpSum' and 'Mortality'

review, tree-based models benefit more than other algorithms when used with RFE, and XGB has an in-built penalty to avoid overfitting compared to other tree-based algorithms, so it is used as the underlying algorithm. Furthermore, XGB performs the best on the data with the original feature set compared to other models. F1 micro is used as the scoring function due to data being imbalanced. The cross-validation score against the number of features selected is plotted in Figure 7.

We can see that the score increases steeply before 40; thereafter, the score increases marginally with each additional feature before peaking at 67, after which the score starts to drop a little.

The mutual information method is then used to select features further from the 67 features. This method requires the modeller to specify the number of features to be selected, but this is unknown, so a loop is used to compute the accuracies of all algorithms on the testing set for k features, where $k = \{5, 6, \dots, 67\}$. The performance of the testing dataset using the feature selected by each loop is compared, and multiple loops can produce the same maximum accuracy score. The loop that maximises the performances across all models is chosen, and the number of features in this loop is 43, which contains ‘BMI’, ‘FaceAmount’, ‘EntryAge’, ‘LumpSum’, ‘OccClass’, ‘heart’, ‘neurological’, ‘psychiatric’, ‘diabetes’, amongst other features.

6. MODELS AND MODEL PERFORMANCES

6.1 Models

There are ten machine learning algorithms included: Random Forests, Decision Tree, Gradient Boosting, Extreme Gradient boosting, bagging algorithm, AdaBoost, Support Vector Machine, Stochastic Gradient Descent, K Neighbours, and ordinal logistic regression. Some are classification models; others are regression models.

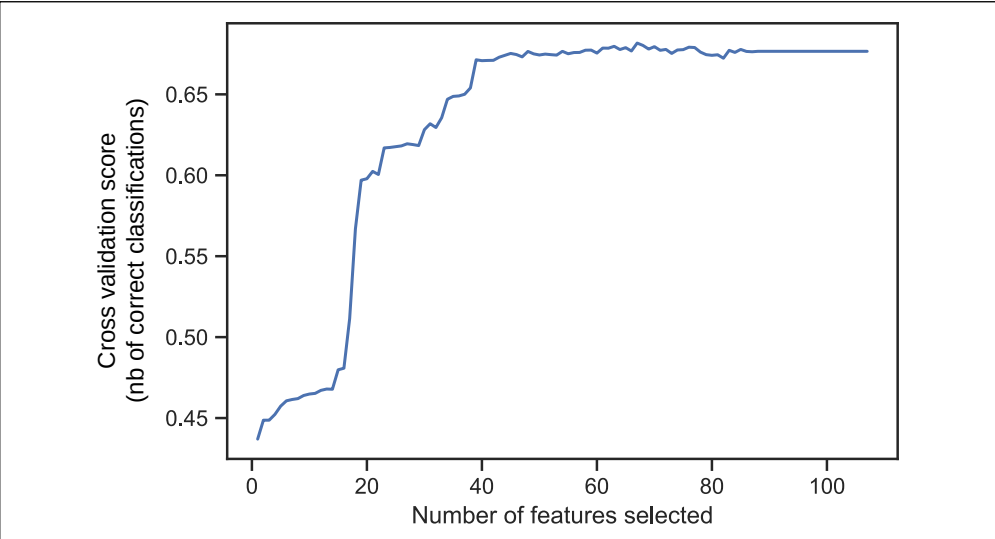


FIGURE 7 Score vs number of features selected

The parameters are tuned such that the accuracy score is maximised for each classifier. For some parameters, the range can be estimated due to the nature of the problem. For example, for boosting or bagging algorithms, the ‘max_feature’ is set to be no greater than the total number of features selected. Parameter ‘n_estimators’ is the number of trees built for tree-based algorithms. It is set to be 80 initially. This parameter, together with other parameters such as learning rate, are finetuned using a five-fold randomised search after the optimal subset of feature is selected. A randomised search of optimal parameters is done on parameters that may reduce overfitting. For example, eta is the parameter that controls the model complexity of XGB and may reduce overfitting, so it is one of the parameters that is tuned for XGB.

For Adaboost, ‘SAMME’ algorithm is implemented, so the underwriting decision is modelled as a single multiclass classification as opposed to multiple binary classifications. The theories are outlined in section 2.3.

For the SGD algorithm, the loss function is set to ‘log’ so the underlying learner is logistic regression and the penalty term is set to be L1 regularisation. Lasso regression is chosen because it removes noisy features; whereas L2 regularisation reduces the weights on these features.

For the SVM algorithm, Radial Based Function (RBF) is used as the kernel. Insurance underwriting decisions are unlikely to have linear or log-linear relationships only with the input variables, due to the complex nature of medical underwriting, so the linear kernel is unlikely to be suitable. As discussed in the literature section, previous studies have found that RBF generally performs better than polynomial, so RBF is chosen.

Default settings are used for other algorithms.

6.2 Performances

Table 2 shows the model performances with no feature selection or parameter tuning; Table 3 shows the model performances after feature selection and parameter tuning.

TABLE 2 Model performances before feature selection and parameter tuning

	Model_Name	Precision	Recall	Train_Accuracy	Test_Accuracy	F1_Score
7	XGBClassifier	0.71	0.71	0.94	0.71	0.71
1	RandomForestClassifier	0.70	0.70	0.94	0.70	0.70
4	BaggingClassifier	0.70	0.70	0.94	0.70	0.70
5	GradientBoostingClassifier	0.66	0.66	0.94	0.66	0.66
0	DecisionTreeClassifier	0.60	0.60	0.93	0.60	0.60
9	KNeighborsClassifier	0.47	0.47	0.66	0.47	0.47
2	LogisticRegression	0.44	0.44	0.44	0.44	0.44
3	AdaBoostClassifier	0.44	0.44	0.44	0.44	0.44
6	SVC	0.44	0.44	0.44	0.44	0.44
8	SGDClassifier	0.44	0.44	0.44	0.44	0.44

TABLE 3 Model performances after feature selection and parameter tuning

	Model_Name	Precision	Recall	Train_Accuracy	Test_Accuracy	F1_Score
7	XGBClassifier	0.71	0.71	0.94	0.71	0.71
1	RandomForestClassifier	0.70	0.70	0.94	0.70	0.70
4	BaggingClassifier	0.70	0.70	0.94	0.70	0.70
5	GradientBoostingClassifier	0.68	0.68	0.94	0.68	0.68
0	DecisionTreeClassifier	0.63	0.63	0.91	0.63	0.63
9	KNeighborsClassifier	0.49	0.49	0.66	0.49	0.49
2	LogisticRegression	0.44	0.44	0.44	0.44	0.44
3	AdaBoostClassifier	0.44	0.44	0.44	0.44	0.44
6	SVC	0.44	0.44	0.44	0.44	0.44
8	SGDClassifier	0.23	0.23	0.24	0.23	0.23

Noticeable differences are observed between accuracies of training and test sets across all models, indicating overfitting. Although feature selection methods and parameter tuning are used to reduce the extent of overfitting, this is often inevitable with real-world data. That said, feature reduction is still useful: the performances of gradient boost and decision tree models improved from 0.66 to 0.68, and 0.60 to 0.63, respectively.

The best performing algorithm is XGB, followed by RF and Bagging. All three algorithms combine outputs of weak learners into the final output, and the underlying weak learners are decision trees. This is not surprising, because the natural process that human underwriters use to reach a decision resembles a decision tree.

XGB is the winner of multiple Kaggle challenges (Chen & Guestrin, 2016). Some medical conditions are rare, so there are sparse data for these conditions. The accuracy score stops improving or even decreases a little after the optimal subset of features is selected, so there are features that are noises in the dataset. XGB handles sparse data well and is designed to avoid overfitting by penalising complexity, so it is unsurprisingly the best performer here. Another major advantage of XGB is the in-built penalty term added to the loss function, so the algorithm prefers models with less complexity and thus is less prone to overfitting.

The bagging algorithm indeed outperforms other boosting algorithms, namely AdaBoost and Gradient Boost. The insurance dataset contains noise, so this result echoes Dietterich’s research that bagging algorithms perform better than boosting algorithms on datasets with many noises (Dietterich, 2000).

Not surprisingly, logistic linear regression, SGD with log loss function and SVM with RBF kernel did not perform well. The underwriting decision resembles a decision tree, so the relationships between the outcome variable and input variables are unlikely to be solely logistic linear. Furthermore, regression models typically do not perform well on datasets with many features, which is the case here.

However, regression models are well studied and understood. The coefficients of input variables are summarised in Appendix 7. An extract is shown in Table 4.

TABLE 4 Coefficient extract

feature name	standard	decline
FaceAmount	5.92E-06	6.03831E-06
EntryAge	2.23E-08	5.85535E-08
BMI	1.33E-08	3.3228E-08
Height	9.86E-08	2.17097E-07
Weight	4.17E-08	1.00749E-07
New	2.99E-10	8.43409E-10
OccClass	9.9E-10	2.22577E-09
heart	-2.8E-11	4.56341E-11
neurological	2.2E-12	3.07357E-11
psychiatric	1.63E-11	2.67144E-11
diabetes	-1E-11	1.18047E-11
investigation	-1.9E-12	2.72756E-12

As seen above, ‘heart’ has a positive coefficient for label ‘decline’, so an application with cardiovascular conditions is more likely to be declined. On the other hand, ‘heart’ has a negative coefficient for label ‘standard’, so an application has no cardiovascular condition is more likely to be accepted on standard terms. The similar applies for ‘diabetes’.

The best performer, XGBoost, achieved 94% accuracy on the training dataset and 71% accuracy on the testing set, which is a rather significant improvement from a random guess for a 19-class classification problem. From a practical point of view, it is also a good improvement to the current rule-engines that found it ‘challenging to issue more than one-third of policies directly from the system’ (Batty et al., 2010). It is worth noting that these scores are achieved despite several limitations which are discussed in section 8.3. With further developments such as more data with more detailed and accurate information, greater computational power to implement more efficient feature selection methods and hyperparameters for each predictive algorithm, it is possible to improve further the accuracies achieved on the test dataset. The results show that the XGBoost is undoubtedly a promising predictive machine learning algorithm in the context of automating underwriting decisions.

7. BUSINESS APPLICATION

The most obvious application to improve the current underwriting process is to implement predictive machine learning algorithms to predict underwriting decisions. XGB shows significant improvement compared to the rule-engines, and there is scope to improve the accuracy score of XGB even more with increased computation power and more high-quality data.

Another obvious application is to implement natural language processing, clustering analysis for data mining and preprocessing into the underwriting process. This should reduce the amount of work that underwriters are currently doing by checking the impairments

and occupations against underwriting manuals or having to enter these into the rule-engines system to reach a decision. Another byproduct advantage of implementing these preprocessing techniques is that data errors such as typos and informal descriptions are handled in the process.

The feature importance ranking under XGB of all features is illustrated in Appendix 8. It is clear that ‘LumpSum’ is the most important feature, and that the accuracy stops increasing after ‘accident’ is selected. This demonstrates that some information captured in the application is unnecessary for underwriting. Insurers can cross-check with other functions such as pricing and risk management to see if these features are necessary; if not, these features can be removed to speed up the application process, which should, in turn, improve customer experience. Some features are important and relatively simple to capture, but oddly some insurers do not capture these features. For example, BMI is ranked as one of the most important features by all feature selection methods, yet several insurers do not capture this feature in their application process. This should be investigated.

In addition to the above, the models can be adapted for different insurance products. For example, microinsurance products targeted at the lower-income class with a cap on the sum insured. Another example is group products with limited underwriting required and sum insured typically capped. These products typically have a smaller sum insured and simple product design. The underwriting decision is typically declining or accepting the application. For these products, a subset of data of small sum-insured products can be extracted for modelling, and the outcome variables can be transformed into a dummy variable indicating accept or decline only. The problem becomes a binary classification, and the accuracy achieved by XGB is 83% without feature selection or parameter tuning. The feature ranking is then tailored to the product of interest only, as opposed to all products. For example, ‘no condition’, as opposed to ‘LumpSum’, is the most important feature for this kind of product. The application form should be then streamlined to capture these features only, thereby significantly speeding up the underwriting process.

An interesting observation is that the optimal feature subset contains medical conditions such as ‘heart’, ‘neurological’, ‘psychiatric’, ‘diabetes’, but not ‘cancer’, so from an underwriting point of view, more research should be devoted to understanding the health and risk impacts of cardiovascular disease, as opposed to cancer.

For companies that would like to implement predictive machine learning models in parallel to employing underwriters, predictive algorithms can be used to flag policies for review. For example, insurers concerned about policies with large underwriting loadings can extract policies with labels above the desired threshold for analysis.

8. CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

The best performing predictive algorithm is XGB, using a combination of correlation-based filter methods, mutual information method, and recursive feature elimination for feature selection. The accuracy score is 94% on the training set and 71% on the test set, and

the precision score is 71%. This is a significant improvement from the existing rule-based underwriting engine that can handle only a third of all applications.

For insurers that focus on the riskiest group of applications with loading larger than 400%, the predictive model produces a label for this group of policies, so this can be used to flag policies for review, should insurers wish to implement the predictive model in parallel to underwriters.

Finally, the output of models can be used to improve the underwriting process in the short term. For example, BMI is revealed to be a significant predictor, so insurers aiming to improve underwriting accuracies are encouraged to capture this variable as accurately as possible. The model can be adapted to suit the requirements of more niche insurance products such as microinsurance that typically targets low-income groups with simple accept or decline decisions. In these cases, the accuracy of models can be even higher, and the feature rankings are tailored to the product.

In summary, the desired outcomes discussed in section 1.2 are achieved.

8.2 Limitations and future developments

The data used for this research are a reinsurer's dataset containing only referred policies that cannot be processed by the rule-based engines. This is particularly suitable for the aim of this research – assessing the applicability of predictive machine learning models in replicating medical underwriting and for applications with a large sum insured. However, it means that the prediction accuracies would have been higher using a dataset more representative of the general insured population which contains more standard simple applications.

For the purpose of this research, the models need to predict the underwriting decision as well as the actual loading for loaded cases. In practice, it is more likely that the outcome is a binary (decline/accept) or three-factor (decline/load/standard) variable. This means that when the algorithm is implemented in practice, the prediction accuracy should be higher still.

All the predictive machine learning algorithms such as XGB are supervised algorithms. The class labels are the underwriting decisions that underwriters made in the past. This means effectively that the accuracy scores compare the decisions that the machine learning algorithms reached, and the decisions that underwriters made. This means that the class labels are subject to human errors, so the difference between class labels and predicted decisions should be further analysed by experts.

After removing missing data and outliers, the dataset contains only 18182 data points but 118 features. Some machine learning models require a large number of data points for each feature, and 18182 data points are insufficient. This could contribute to the poor performance of some models such as SVM. More data may improve performances of these models.

Although removing missing data is the most appropriate treatment for this dataset, the missing data points are entirely attributed to a couple of companies' applications. This could result in some biases in the result if these companies have significantly different

underwriting practices to the rest of the companies. However, given the competitive pressure in the market, this is unlikely.

Currently ‘occupation’ is filled with job titles but the nature of work for some jobs with the same job title could be very different. For example, a director at a construction company would spend significant time on site; whereas a director at a consultant company would work predominantly in an office. The former would fall into class 3, whereas the latter would be class 1. If the application form is redesigned to include a brief job description, the model performances could improve. Another practical limitation is that medical conditions are entered manually, so there are errors in the data; some mandatory checks in the administration system should improve prediction accuracies.

The combination of feature selection methods and underlying algorithms is a design choice that could be further studied. For example, RFE could be tested on other classifiers. Furthermore, there is no clear guidance on the choice of feature selection methods – alternative methods such as heuristic search methods for feature selection exist. The final choice is based on the existing research on various aspects using non-insurance data. All these design choices could be studied to further improve the accuracies or predictive power of machine learning algorithms applied to the underwriting decisions in the context of life and health insurance.

REFERENCES

- Batty, M & Kroll, A (2009). Automated life underwriting – A survey of life insurance utilization of automated underwriting systems. Sponsored by the Society of Actuaries, prepared in Deloitte, December 2009: p.4 & p.11. <https://www.soa.org/globalassets/assets/files/research/projects/research-life-auto-underwriting.pdf>
- Batty, M, Moore, D & McCarty, M (2010). Automated life underwriting: Phase 2 Study of automated life insurance underwriting. Sponsored by the Society of Actuaries, prepared in Deloitte Consulting LLP, August 2010: p.3, p.4 & p.11. <https://www.soa.org/globalassets/assets/Files/Research/Projects/research-auto-life-underwriting-2.pdf>
- Boodhun, N & Jayabalan, M (2018). Risk prediction in life insurance industry using supervised learning algorithms, *Complex Intell. Syst.* **4**, 145–154 (2018). <https://doi.org/10.1007/s40747-018-0072-1>
- Breiman, L, Friedman, J, Olshen, R & Stone, C (1984). *Classification and Regression Trees*, Wadsworth, Belmont
- Chandrashekar, G & Sahin, F (2014). A survey on feature selection methods. *Computers and Electrical Engineering* **40**(2014) 16–28
- Chartered Insurance Institute (2020). Insurance underwriting process (IF3). <https://www.cii.co.uk/learning/qualifications/unit-insurance-underwriting-process-if3/>
- Chen, T & Guestrin, C (2016). XGBoost: A scalable tree boosting system, KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2016, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Cornelius, N (2015). Why does it sometimes take so long to be approved for life insurance? *Quotacy*, 25 September 2015. <https://www.quotacy.com/why-does-it-take-so-long-to-be-approved-for-life-insurance>
- Cortes, C & Vapnik, V (1995). Support-vector networks. *Machine Learning* **20**, 273–297, Kluwer Academic Publishers. <https://doi.org/10.1007/BF00994018>
- Dietterich, TG (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomisation. *Machine Learning*, **40**, 139–157, Kluwer Academic Publishers
- Friedman, J (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, **29**(5), 2001
- Hastie, T, Tibshirani, R & Friedman, J (2009). *Elements of Statistical Learning*, Ed. 2, Springer
- Jakobsen, JC, Gluud, C, Wetterslev, J & Winkel, P (2017). When and how should multiple imputation be used for handling missing data in randomised clinical trials – a practical guide with flowcharts. *BMC Medical Research Methodology*, Vol 17, Article number: 162. <https://doi.org/10.1186/s12874-017-0442-1>
- Kuhn, M & Johnson, K (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*, Chapter 11.3, Recursive Feature Selection, Chapman & Hall/CRC Data Science Series. <https://bookdown.org/max/FES/recursive-feature-elimination.html>
- Prudential (2015). Prudential Life Insurance Assessment – Can you make buying life insurance easier? <https://www.kaggle.com/c/prudential-life-insurance-assessment/overview/>

- Quinlan, JR (1996). Bagging, boosting, and C4.5, Proceedings of the thirteenth national conference on Artificial intelligence – Volume 1, August 1996, pp 725–730.
<https://dl.acm.org/doi/10.5555/1892875.1892983>
- Sarveniazi, A (2014). An actual survey of dimensionality reduction. *American Journal of Computational Mathematics*, 2014, 4, 55–72
- Sterne, JAC, White, IR, Carlin, JB, Spratt, M, Royston, P, Kenward, MG, Wood, AM & Carpenter, JR (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *BMJ* 2009; 338:b2393 (June 2009), <https://doi.org/10.1136/bmj.b2393>
- Tsuruoka, Y, Tsujii, J & Ananiadou, S (2009). Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty, ACL '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Vol 1, 477–485. <https://dl.acm.org/doi/10.5555/1687878.1687946>
- van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3), 1–67. <https://doi.org/10.18637/jss.v045.i03>
- Yekkehkhany, B, Safari, A, Homayouni, S & Hasanlou, M (2014). A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XL-2/W3, 281–285
- Zhu, J, Rosset, S, Zou, H & Hastie, T (2006). Multi-class AdaBoost, *Statistics and its interface*, Vol. 2, (2009), 349–360, 0.4310/SII.2009.v2.n3.a8

10 APPENDICES

10.1 Appendix 1 – Data description

Column	Description
Company	Insurer's company name, anonymised to protect data
DateInputYear	Calendar year in which the application is submitted
NoofAssessment	Number of assessments carried out on the application
ProductRisk	Broad product risk, e.g. mortality risk or disability risk
AssessmentDetailID	ID assigned to the application
Decision&Details	Underwriting decision: decline, accepted or accepted with loading
DateInputMonthYear	Calendar year and month in which the application is submitted
FaceAmount	Sum insured
SABand	Sum insured band; 3 categories are available
Uw	The name of underwriter who assessed the application
DateOfBirth	Date of birth of the applicant
ExtraMortality	Extra loading applied due to extra mortality risk
FlatExtra	Flat loading applied due to various risk
LoadingPercentPremium	A percentage loading applied as a catch-all
FollowUpIndicator	Indicator whether the application is a follow-up application
EntryAge	The age of the applicant
EntryAgeBand	The age band of the applicant, bands are in 10 years
AssessmentImpairment	Descriptions of medical impairments
ParticipatingCMO	The name of medical doctors who assisted medical underwriting
NonmedRisk	Non-medical risk such as dangerous hobbies
ProductTotalline	Broad product line, e.g. life insurance, TPD
ResponsetimeMinutes	The time spent on assessing the applications
Gender	Gender of the applicant
Occupation	Occupation or job title of the applicant
BMI	BMI of the applicant
OccupationClassCd	Letters, cannot be used directly without knowing the code meaning
AdditionalConditionsNonmed	Additional non-medical conditions
BenefitType	Benefit type can be lump sum or annuity
Decision	Broad underwriting decision, either accept or decline
Indexation	Whether the benefit is indexed
Index%	The percentage of indexation
MaritalStatusCd	Marital status of the applicant
PlanTypeLongname	Detailed product plan type
Height	Height of the applicant
Weight	Weight of the applicant
New	Whether the application is new
FollowUp	Whether the application is a follow-up

10.2 Appendix 2 – Extracts of datasets

10.2.1 EXTRACT OF THE PRIMARY DATASET

FollowUpIndicator	EntryAg	EntryAg	AssessmentImpairment	Participating	NonmedRisk	Product	Respon	Gender	Occupation	BMI
NewAssessment	27	<31	wrist complaints	0		IP	410	f	pursuits instructor	
NewAssessment	27	<31	back pain	0		IP	410	f	pursuits instructor	
FollowUp	71	>70		0		Life	19	m		
NewAssessment	71	>70		0		Life	73	m		
NewAssessment	53	51-60		0		IP	136	f	Sales assistant	25.2664
FollowUp	42	41-50		0		IP	43	m		
NewAssessment	47	41-50		0		IP	216	m	Director	
FollowUp	61	61-70	coronary heart disease	0		Life	8	f		
FollowUp	61	61-70	sleep apnoea	0		Life	8	f		
FollowUp	61	61-70	gastnc bypass	0		Life	8	f		
NewAssessment	61	61-70	coronary heart disease	0		Life	215	f		
NewAssessment	61	61-70	sleep apnoea	0		Life	215	f		
NewAssessment	61	61-70	gastric bypass	0		Life	215	f		
NewAssessment	45	41-50		0		IP	131	m		
NewAssessment	39	31-40	Hodgkin disease	1		Life	144	m	Prison officer	30.5568
NewAssessment	38	31-40		0		Life	237	f		
NewAssessment	57	51-60	synovial sarcoma	0		Life	165	m	Director	28.9827
NewAssessment	20	<31		0		Life	548	m	Labourer	23.7196
NewAssessment	34	31-40	myelodysplasia	1		Life	296	f	Manager	
NewAssessment	68	61-70		1		Life	353	f		26.68
NewAssessment	51	51-60		0		Life	109	m	consultant cardiologist	
FollowUp	37	31-40	aneurysm	1		Life	1095	m		23.2668
FollowUp	43	41-60		1		IP	1139	m	Sales manager	26.2346
FollowUp	43	41-50		1		IP	1139	m	Sales manager	26.2346
NewAssessment	45	41-50		0		Life	33	m	RAF Captain	
NewAssessment	50	41-50		0		Life	1150	m	Sales	
NewAssessment	50	41-50		0		Life	1150	f		
NewAssessment	33	31-40	linitus	0		IP	174	m		
NewAssessment	33	31-40	shoulder problems	0		IP	174	m		
NewAssessment	33	31-40	headaches	0		IP	174	m		
FollowUp	70	61-70	depression	0		Life	165	f	Teacher	24.087

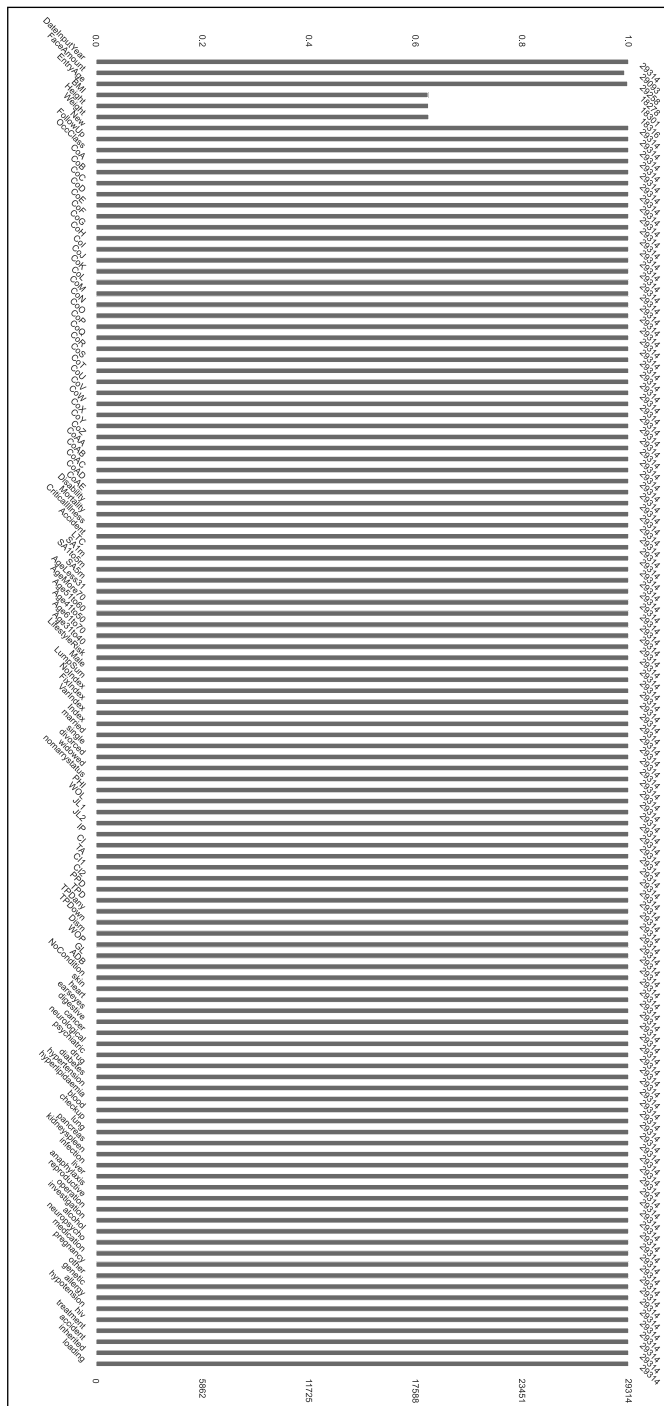
10.2.2 SUPPLEMENTARY LIST: EXTRACT OF THE CONDITION LIST

SYNONYM	CATEGORY
Hodgkin's Disease	CANCERS, GROWTHS OR TUMOURS
Hodgkin's lymphoma	CANCERS, GROWTHS OR TUMOURS
Hurthle cell tumour of the thyroid	CANCERS, GROWTHS OR TUMOURS
Hutchinson's freckle	CANCERS, GROWTHS OR TUMOURS
Hydatid cyst	CANCERS, GROWTHS OR TUMOURS
Hydatid mole	CANCERS, GROWTHS OR TUMOURS
Hydatidiform mole	CANCERS, GROWTHS OR TUMOURS
Hydrocele - canal of Nuck	CANCERS, GROWTHS OR TUMOURS
Hyperplasia	CANCERS, GROWTHS OR TUMOURS
Idiopathic bone cavity	CANCERS, GROWTHS OR TUMOURS
Ileal tumour	CANCERS, GROWTHS OR TUMOURS
Immunoblastic lymphoma	CANCERS, GROWTHS OR TUMOURS
Infiltrating lipoma	CANCERS, GROWTHS OR TUMOURS
Inflammation of nail bed	CANCERS, GROWTHS OR TUMOURS
Inflammation of the soft tissues around the teeth	CANCERS, GROWTHS OR TUMOURS
Initis (non-traumatic)	CANCERS, GROWTHS OR TUMOURS
Intermediate grade lymphoma	CANCERS, GROWTHS OR TUMOURS
Intestinal parasite	CANCERS, GROWTHS OR TUMOURS
Intestinal polyp	CANCERS, GROWTHS OR TUMOURS
Intracerebral abscess	CANCERS, GROWTHS OR TUMOURS

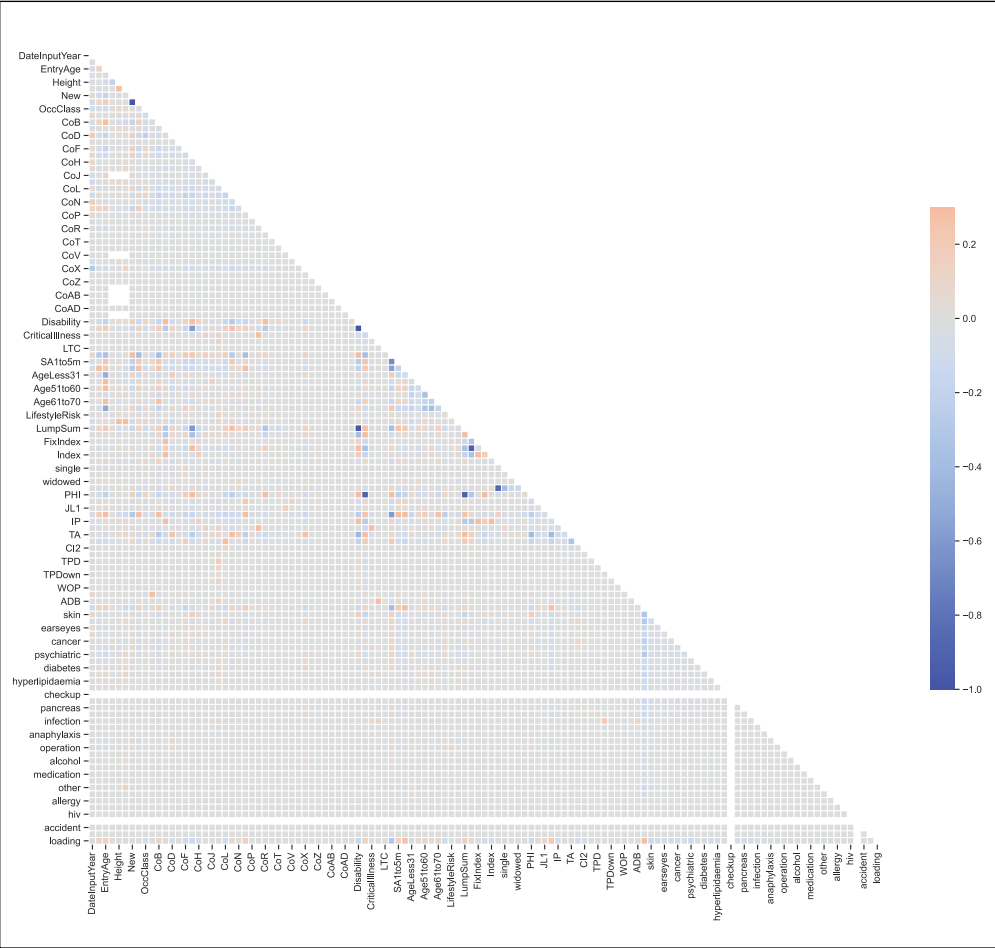
10.2.3 SUPPLEMENTARY LIST - OCCUPATION LIST

Occupation	OccCulapsastion
Accountant/Accountancy Assistant	1
Actor/Actress (No Stunt Work)	3
Actor/Actress (Stunt Work)	3
Actuary/Actuarial Assistant	1
Acupuncturist	2
Admin. Staff/Managers (Office Work)	1
Advertising Agent/Clerk/Executive	1
Aerial Erector*	4
Agricultural Worker	4
Aircraft Engineer	3

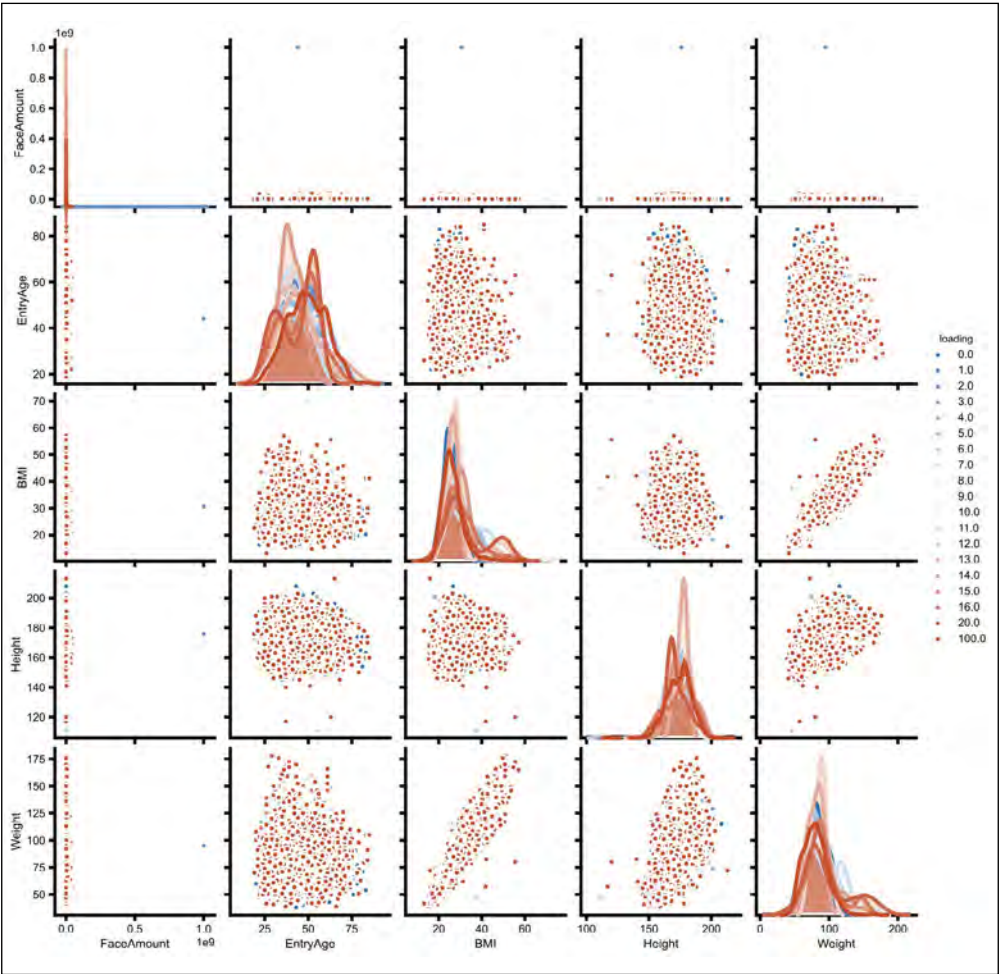
10.3 Appendix 3 – Missing value percentages



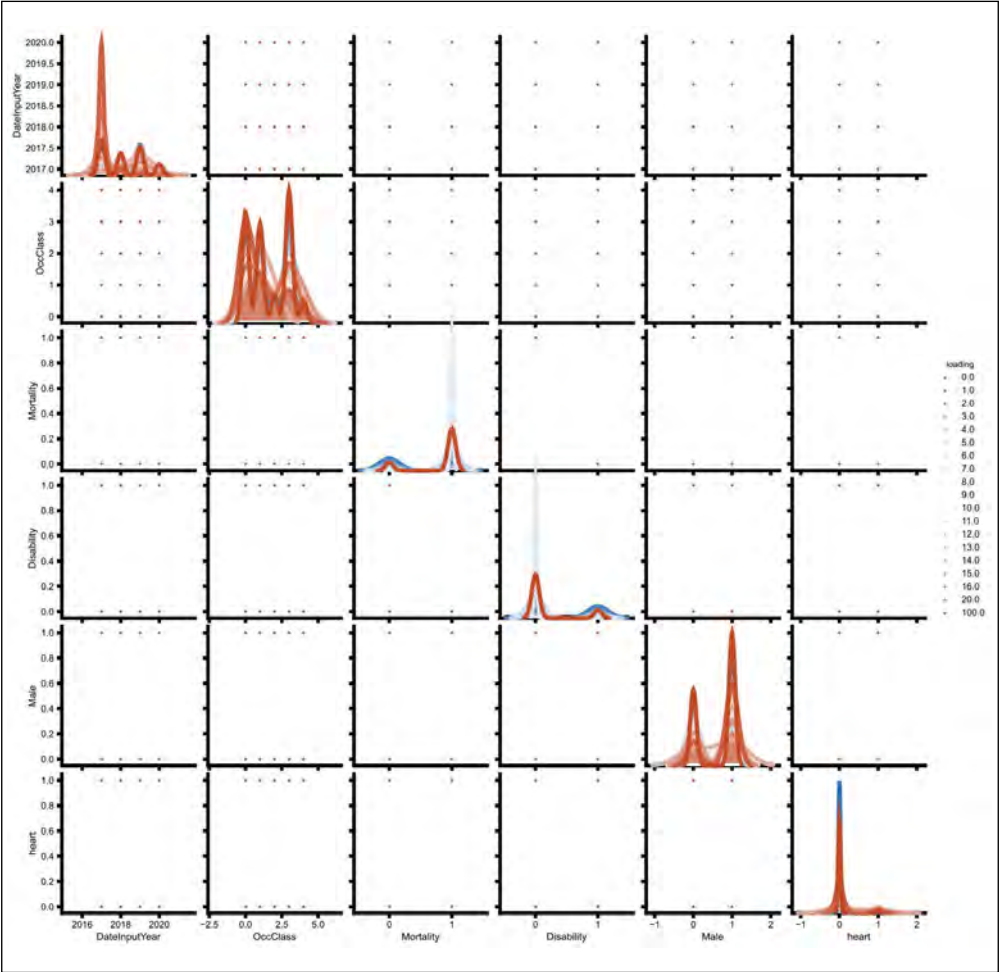
10.4 Appendix 4 – Correlation matrix of missing values and other variables



10.5 Appendix 5 – Scatterplot matrix of continuous input variables



10.6 Appendix 6 – Scatterplot matrix of categorical input variables



10.7 Appendix 7 Coefficients of logistic regression

feature name	standard	loading 1	loading 2	loading 3	loading 4	loading 5	loading 6	loading 7	loading 8	loading 9	loading 10	loading 11	loading 12	loading 13	loading 14	loading 15	loading 16	loading 20	decline
0 DietQual	1.1270E-07	-3.6119E-07	-2.9673E-07	-2.8298E-07	-9.1078E-08	-2.3108E-07	-1.6828E-07	-2.0008E-07	-2.1059E-07	-3.5010E-07	-3.4701E-07	-8.3495E-07	-2.1784E-07	-1.9088E-07	-1.6568E-07	-1.9298E-07	-1.5359E-06	-1.3913E-06	2.2145E-06
1 FatAmount	5.9153E-06	4.6829E-06	5.8204E-06	3.4796E-06	5.4107E-06	2.0096E-06	1.8628E-06	2.0008E-06	3.8541E-06	5.5876E-06	6.4968E-06	-8.3495E-07	3.1544E-06	-1.2054E-05	-8.6907E-06	-1.2149E-05	-2.7955E-06	-1.3913E-06	6.0381E-06
2 FatType	2.2327E-08	-7.8465E-08	-8.5027E-08	-6.3706E-08	-1.9544E-08	-4.2001E-08	-3.8393E-08	-4.4455E-08	-4.5577E-08	-3.7921E-08	2.9775E-08	-3.8181E-08	-4.8345E-08	-4.1898E-08	-3.6381E-08	-4.2529E-08	-3.2405E-08	-2.9823E-08	5.8553E-08
3 BMI	1.3318E-08	-4.7720E-08	-3.7489E-08	-3.7420E-08	-1.1191E-08	-2.8057E-08	-2.6034E-08	-2.5591E-08	-2.6404E-08	-2.4041E-08	-4.5573E-08	-4.2211E-08	-2.7615E-08	-2.6143E-08	-2.1787E-08	-2.6706E-08	-1.9316E-08	-1.5361E-08	3.3228E-08
4 Height	9.8368E-08	-3.1082E-08	-2.6286E-08	-2.4687E-08	-7.8651E-09	-2.0059E-08	-1.6269E-08	-1.7356E-08	-1.8128E-08	-1.5015E-08	-3.0977E-08	-1.5607E-08	-1.6967E-08	-1.6297E-08	-1.4271E-08	-1.7739E-08	-1.3177E-08	-1.2098E-08	1.7070E-07
5 Weight	4.7149E-08	-1.4389E-08	-1.2010E-08	-8.7327E-09	-6.4957E-09	-8.0179E-09	-7.2556E-09	-7.8572E-09	-8.0179E-09	-7.2556E-09	1.3870E-09	7.2521E-09	-8.2837E-09	-7.9936E-09	-6.5668E-09	-8.0749E-09	-5.8213E-09	-4.7691E-09	1.0049E-07
6 New	2.9853E-05	-1.3004E-06	3.2744E-11	9.2720E-06	-1.4008E-11	-8.4167E-11	-5.6384E-11	6.7139E-11	1.7665E-11	6.5054E-11	1.2544E-06	1.6493E-11	-7.6884E-11	6.8401E-11	6.5960E-11	6.901E-11	6.3128E-11	6.5106E-11	8.8449E-06
7 OccClass	9.8992E-20	-3.2710E-10	-1.4965E-10	-2.3222E-10	9.3220E-11	-1.7783E-10	-1.5852E-10	1.9157E-10	-1.8443E-10	-1.6945E-10	3.3157E-10	1.7866E-10	-2.0399E-10	-1.8526E-10	-1.4613E-10	-1.8326E-10	-1.1458E-09	-1.6023E-10	2.2527E-09
8 CoA	9.7881E-11	-7.6785E-11	-4.9402E-11	-8.8490E-11	-2.8593E-11	-3.7268E-11	-3.8914E-11	-1.7662E-11	-5.6407E-11	-5.0461E-11	6.8434E-11	-9.2312E-11	-2.82E-11	-1.1592E-11	-1.2800E-11	-1.1574E-11	-1.1407E-11	-1.2935E-11	1.4326E-10
9 CoB	-4.0199E-11	1.0128E-12	-7.8743E-11	-2.7266E-12	1.6043E-12	6.1635E-14	3.8914E-12	-1.4227E-12	-5.6707E-13	-5.0461E-13	6.8434E-11	-9.2312E-11	-2.82E-11	-1.1592E-11	-1.2800E-11	-1.1574E-11	-1.1407E-11	-1.2935E-11	1.4326E-10
10 CoC	8.5979E-11	-4.0919E-11	1.3184E-11	5.1721E-12	2.2311E-11	-1.6947E-11	7.8665E-12	-1.7240E-11	-1.8168E-11	-1.9083E-11	1.2109E-11	-1.5787E-11	-2.3115E-11	1.9216E-11	-1.5292E-11	-1.9046E-11	-1.9684E-11	-2.2462E-11	1.9381E-10
11 CoG	2.1294E-10	3.1779E-11	4.2745E-11	1.2893E-11	1.8907E-12	-3.0814E-11	-3.4243E-11	-2.7897E-11	-3.5487E-11	-2.4001E-11	-3.75E-11	-2.6475E-11	-2.356E-11	-2.355E-11	-2.243E-11	-2.3377E-11	-2.6165E-11	-2.6999E-11	1.1901E-10
12 CoH	1.2457E-11	-1.6176E-12	3.2271E-12	-1.8066E-12	1.5805E-12	-1.2836E-12	-1.6533E-12	-7.8429E-13	-1.4081E-12	-4.6147E-13	1.6536E-12	-4.7630E-13	-1.0939E-12	-4.3408E-13	-4.7849E-13	-4.3310E-13	-5.6944E-13	-5.9499E-13	9.2638E-13
13 CoI	6.3699E-11	-2.2192E-12	3.9733E-12	-1.1207E-11	-1.6319E-11	-1.2112E-11	-1.4915E-11	-1.1918E-11	-1.6130E-11	-8.0319E-12	1.9316E-11	-1.0426E-11	-9.9673E-12	-1.0801E-11	-9.9587E-12	-1.1324E-11	-1.1533E-11	-1.1533E-11	9.8213E-13
14 CoK	8.1961E-11	-5.2529E-12	-1.8658E-11	-2.7153E-11	-2.7129E-11	-9.8744E-12	-1.8352E-11	8.6991E-12	-1.5847E-11	4.9772E-12	1.3068E-11	-1.4470E-12	-1.2204E-11	-4.6078E-12	-4.1770E-12	-6.0134E-12	-7.2264E-12	-7.4986E-12	1.1279E-10
15 CoL	8.8397E-12	-3.6054E-11	2.0085E-11	-2.2135E-11	1.9009E-11	9.3301E-12	8.7229E-12	2.1005E-12	7.3717E-12	-6.4408E-12	1.1984E-11	-2.9930E-12	-4.2798E-12	-8.1376E-12	5.5730E-12	9.0024E-12	-1.1114E-11	-1.2405E-11	8.6872E-11
16 CoM	7.9645E-11	-5.8142E-11	-6.1895E-11	-2.6562E-11	-1.5089E-11	-1.2697E-11	-3.1939E-11	1.1987E-11	-1.1810E-11	-3.4748E-11	-3.4867E-11	-1.0758E-11	-5.2712E-12	-1.1208E-11	-1.0233E-11	-1.1279E-11	-1.7384E-12	-1.4270E-11	2.8261E-10
17 CoN	8.6172E-13	-4.5342E-14	6.0125E-13	-7.9986E-14	9.6288E-14	-2.9296E-14	-5.7535E-14	-1.0076E-14	-3.6685E-14	-5.8093E-15	5.7360E-14	-2.5638E-11	-3.3936E-11	-5.2985E-15	-5.5202E-15	-7.0951E-15	-7.4533E-15	-7.4533E-15	-9.8213E-13
18 CoO	2.8232E-11	-2.1804E-11	-2.6927E-11	2.2531E-11	-1.842E-11	-1.3098E-11	-7.5098E-12	-8.9025E-12	1.4369E-12	-2.7049E-12	3.9403E-12	-2.0370E-12	7.4899E-13	-6.8153E-12	6.6544E-13	7.2445E-12	2.1701E-11	4.0259E-11	1.0900E-10
19 Disability	4.2608E-10	-2.2135E-11	8.8921E-11	-2.1431E-11	-2.1386E-11	-5.5497E-11	-5.7949E-11	-6.0897E-11	-7.8098E-11	-5.3155E-11	8.1962E-11	-5.8384E-11	-7.1826E-11	-5.3209E-11	-5.8205E-11	-5.2494E-11	-5.7907E-11	-5.7907E-11	3.4271E-10
20 Age*1000	5.6104E-10	-2.0372E-11	4.9646E-11	-1.5493E-11	9.8448E-12	-1.6779E-11	-3.5847E-11	-4.7006E-11	-2.7452E-11	-3.6763E-11	-3.8687E-11	-3.6238E-11	-3.6706E-11	-4.5483E-11	-2.8617E-11	-4.6725E-11	-1.8752E-11	-1.0700E-11	9.6361E-10
21 Age*1000	2.1008E-10	-5.1172E-11	-4.9348E-11	-3.4882E-11	-5.8473E-12	-2.8639E-11	-4.3887E-11	-2.9380E-11	-3.9409E-11	-2.7702E-11	-5.0496E-11	-2.5638E-11	-3.3936E-11	-2.8800E-11	-2.2813E-11	-3.0084E-11	-2.9844E-11	-3.0862E-11	3.5290E-10
22 Age*1000	2.4504E-10	-5.9215E-11	-1.1727E-11	4.4544E-11	-9.8322E-12	-4.2795E-11	-3.6121E-11	-3.0733E-11	-4.0704E-11	-2.9761E-11	-5.3692E-11	-3.2566E-11	-1.1959E-11	-3.2638E-11	-2.4414E-11	-3.2121E-11	-2.7768E-11	-2.7397E-11	3.2797E-10
23 LifeStyle	6.1855E-12	-1.0245E-12	-1.6195E-12	7.8454E-13	-1.2952E-12	-4.8461E-13	-2.3282E-13	-6.8111E-13	-9.868E-13	-5.3386E-13	1.1123E-12	-5.4375E-13	-8.3798E-13	-5.1595E-13	-5.4037E-13	-5.1529E-13	-5.9584E-13	-6.0269E-13	6.9995E-12
24 LumpSum	1.3553E-08	-1.5691E-10	-2.3396E-10	-1.2070E-10	-2.5697E-11	-8.9778E-11	-3.5847E-11	-4.7006E-11	-2.7452E-11	-3.6763E-11	-3.8687E-11	-3.6238E-11	-3.6706E-11	-4.5483E-11	-2.8617E-11	-4.6725E-11	-1.8752E-11	-1.0700E-11	9.6361E-10
25 Nindex	3.584E-10	-1.5911E-10	-1.5778E-10	-1.2039E-10	-3.878E-11	-8.9778E-11	-3.5847E-11	-4.7006E-11	-2.7452E-11	-3.6763E-11	-3.8687E-11	-3.6238E-11	-3.6706E-11	-4.5483E-11	-2.8617E-11	-4.6725E-11	-1.8752E-11	-1.0700E-11	9.6361E-10
26 Findex	1.6278E-11	-2.0690E-12	1.8701E-12	-3.2679E-12	2.3379E-12	-3.1169E-12	8.2578E-13	-1.8278E-12	-1.7342E-12	-1.3129E-12	-4.4647E-12	-1.3397E-12	-2.6182E-12	-2.6279E-12	-1.3136E-12	-1.2697E-12	-1.5034E-12	-1.5408E-12	1.2493E-10
27 Nindex	1.8317E-10	-3.7169E-11	8.5398E-11	-1.6203E-11	-8.6729E-12	-2.2002E-11	-2.0294E-11	-2.0242E-11	-2.4342E-11	-1.668E-11	3.3397E-11	-1.7897E-11	-2.3847E-11	-1.6734E-11	-1.7289E-11	-1.6594E-11	-1.8890E-11	-1.9264E-11	7.2693E-10
28 Index	1.0265E-10	-2.0372E-11	4.9646E-11	-1.5493E-11	9.8448E-12	-1.6779E-11	-3.5847E-11	-4.7006E-11	-2.7452E-11	-3.6763E-11	-3.8687E-11	-3.6238E-11	-3.6706E-12	-4.5483E-12	-2.8617E-12	-4.6725E-12	-1.8752E-12	-1.0700E-11	9.6361E-10
29 single	6.4662E-12	-4.2666E-12	3.9594E-12	-2.3271E-12	-1.6970E-12	-1.6424E-12	-9.0728E-13	-3.0218E-12	-3.7131E-12	-2.3279E-12	-3.8497E-12	-2.3619E-12	-3.8330E-12	-2.7469E-12	-2.2495E-12	-2.2465E-12	-3.5627E-12	-2.6643E-12	2.2673E-11
30 nonmarriage	5.3161E-10	-1.6737E-10	-3.3153E-10	-1.3347E-10	-3.8726E-10	-1.0979E-10	-8.9571E-10	-9.440E-11	-9.5103E-11	-8.4597E-11	-1.6148E-10	-8.4324E-11	-9.8876E-11	-9.1846E-11	-7.6544E-11	-9.3049E-11	-7.7390E-11	-6.2056E-11	1.1559E-09
31 PH	3.9827E-06	-1.5506E-11	7.182E-11	-1.4537E-11	-1.7791E-11	-4.844E-11	-5.3278E-11	-5.7534E-11	-7.2094E-11	-5.2094E-11	-7.6197E-11	-5.2809E-11	-6.7395E-11	-5.9940E-11	-5.2807E-11	-5.0897E-11	-5.8772E-11	-5.6387E-11	3.2561E-10
32 IL	-5.5629E-04	-4.0589E-12	-3.8744E-12	-1.0399E-12	-1.0399E-12	7.2018E-13	5.8093E-13	6.8868E-13	1.2401E-12	-9.2038E-13	-1.1249E-11	-1.3894E-12	-1.7492E-12	-1.3120E-12	-1.3812E-12	-1.3102E-12	-1.5697E-12	-1.6241E-12	2.3272E-10
33 TA	1.3808E-10	-9.4778E-11	-8.8476E-11	-7.0205E-11	-2.7315E-11	-3.0848E-11	-2.7949E-11	-2.5186E-11	-1.79E-11	-2.2597E-11	-5.1524E-11	-1.9975E-11	-2.2578E-11	-2.9540E-11	-2.0008E-11	-2.9931E-11	-3.1406E-11	-1.8424E-11	4.2549E-11
34 C1	3.2045E-11	-5.1379E-11	1.5922E-11	-7.9102E-11	8.1332E-12	-2.8319E-11	9.1268E-12	-1.3039E-11	-6.2425E-12	-1.0546E-11	1.1963E-11	-1.0539E-11	-9.4067E-12	-2.4615E-11	-4.7071E-12	-2.8768E-11	-1.4316E-11	-1.7158E-11	2.0071E-10
35 C2	3.5653E-12	-3.1775E-12	5.8314E-13	-4.9337E-13	2.8614E-12	-5.8895E-13	1.1113E-12	3.9523E-13	-1.3317E-12	-1.1225E-13	-2.3887E-12	-6.8076E-13	7.1011E-13	-7.6888E-13	-4.3964E-13	-7.6834E-13	-9.8479E-13	-1.0259E-12	6.0299E-12
36 GL	6.5910E-13	-2.8460E-13	-2.0087E-13	-4.6271E-13	-9.9766E-13	-7.3549E-14	-2.9683E-13	-1.6249E-14	7.6995E-13	-1.3218E-14	4.0485E-14	-1.3337E-14	-4.0485E-14	-1.2997E-14	-1.3298E-14	-1.7988E-14	-1.4025E-14	-1.4232E-14	1.5348E-12
37 McCordition	4.6172E-11	-8.5449E-11	-3.4014E-12	-1.4338E-10	-6.2234E-11	-4.0157E-11	-6.8434E-11	-4.0157E-11	-6.8434E-11	-3.187E-11	4.4327E-13	-3.3770E-11	-5.6262E-11	-3.1980E-12	-3.7702E-12	-3.1496E-12	-3.7016E-12	-3.666E-11	8.8243E-10
38 heart	2.8106E-11	-1.6679E-11	8.4979E-12	9.9187E-12	2.4774E-12	-6.6592E-12	9.21134E-12	-8.9832E-12	1.1208E-11	-5.9472E-12	4.4247E-13	-4.7894E-12	6.6491E-12	-6.6491E-12	-1.6491E-12	-7.8289E-12	-2.7083E-12	-6.0691E-12	4.5641E-11
39 neurological	2.1977E-12	-2.3635E-12	1.2408E-12	-4.5883E-12	1.0608E-11	-4.0753E-12	-5.3729E-12	-4.5323E-12	-8.5388E-13	-7.3783E-12	5.7365E-12	-3.8719E-12	-5.0713E-12	-4.0488E-12	-3.8735E-12	-4.009E-12	-4.669E-12	-4.3389E-12	3.0737E-11
40 psychiatric	1.6287E-11	-1.4702E-11	-4.5505E-11	-2.9668E-12	1.2801E-11	-9.8244E-1													

10.8 Appendix 8 – Feature ranking

