MSc Dissertation

Logistic Regression Methods Versus Machine Learning Techniques In Status and Severity Prediction of South African Covid-19 Laboratory Test Data

By

Mark Strickett

Student Number

1849246

Supervisors

Dr. Charles Chimedza

Dr. Farai Mlambo

A research report submitted to the Faculty of Science, University of the Witwatersrand, in partial fulfilment of the requirements for the degree of Master of Science

June 12, 2023

Contents

List of Figures

List of Tables

Dedication

I would like to dedicate this dissertation to my beloved parents who have been a continuous source of strength and inspiration. Your commitment and unwavering support over my entire educational career has enabled me to be in this position and to finish this study.

Acknowledgement

I would like to extend my thanks a deep appreciation to the **NHLS**, who provided that data that was used in this research, and to **Dr. Diederick van der Westhuizen**, of the NHLS, who helped obtain the data used in this research as well as providing guidance and advice surrounding the medical aspects of this research.

It is with sincere appreciation and gratitude that I thank the two men, who without this research would not have been possible, **Dr. Charles Chimedza** and **Dr. Farai Mlambo** of the School of Statistics and Actuarial Science at the University of the Witwatersrand. Thank you for your continued support, effort, guidance and time that you put into this research.

Finally I wish to extend my deepest gratitude and thanks to my **Mother, Family and Girlfriend**. Thank you for all the advice and motivation you have bestowed upon me over the course of my research not to mention all the sacrifices that you all made to make this all possible. Thank You.

June 12, 2023
Mark Strickett

List of Acronyms

AI - Artificial Intelligence
ALT - Alanine Aminotransferase
ANN - Artificial Neural Networks
ANFIS - Adaptive Network-based Fizzy Inference System
ANCOVA - Analysis of Covariance
ANOVA - Analysis of Variance
APACHE - Acute Physiology and Chronic Health Evaluation
AST - Aspartate Aminotransferase
AUC - Area Under the Curve
BLR - Baseline-category Logistic Regression
BNP - Brain Natriuretic Prp Tide
Covid-19 - Coronavirus disease 2019
cTnI - Cardiac Troponin I
DL - Deep learning
DM - Diabetes Mellitus
EDA - Exploratory Data Analysis
FA - Factor Analysis
HIV - Human Immunodeficiency Virus
KMO - Kasier, Meyer, Olkin
MAE - Mean Absolute Error
MICE - Multivariate Imputations by Chained Equations
ML - Machine Learning
MLP-ICA - Multi-layered Perceptron-imperialist Competitive Algorithm
MLPNN - Multilayer Perceptron Neural Network

MLR - Multinomial Logistic Regression
MSA - Measure of Sampling Adequacy
MSE - Mean Square Error
NHLS - National Health Laboratory Service
NIR - No-information Rate
NN - Neural Networks
OLR - Ordinal Logistic Regression
PCA - Principal Component Analysis
PMM - Predictive Mean Matching
RF - Random Forest
RMSE - Root Mean Square Error
ROC - Receiver Operating Curve
RT-PCR - Reverse Transcriptase Polymerase Chain Reaction
SCG - Scaled Conjugate Gradient
SELU - Scaled Exponential Units
SNN - Self-normalising Neural Network
SRS - Stratified Random Sampling
TB - Tuberculosis
VIF - Variance Inflation Factor

Abstract

The Covid-19 pandemic severely impacted on the lives of individuals around the world. Even now as the number of vaccinations has increased and there are fewer cases of Covid-19, knowledge of ones' Covid-19 status remains important. It remains important as it impacts on the lives of family, friends, co-workers and the general public. Therefore, having tools such as the logistic regression and machine learning modelling techniques, in conjunction with the Reverse Transcriptase Polymerase Chain Reaction (RT-PCR), antigen and rapid Covid-19 tests only enables people to be more informed about their Covid-19 infection status.

The aim of this study is to predict the Covid-19 status and severity of an individual using machine learning techniques and logistic regression methods on South African laboratory test data and determine the performance of each method.

The data used in this study was supplied by the National Health Laboratory Service (NHLS) and under went cleaning and preparation phases after which the data was split into four different datasets. The datasets underwent confounding variable analysis, Principal Component Analysis (PCA) and Factor Analysis (FA) before two methods of variable selection were used to arrive at the final four datasets. Each dataset was then used to create five models (Random Forest (RF), Self-normalising Neural Network (SNN), Multinomial Logistic Regression (MLR), Ordinal Logistic Regression (OLR), and Baseline-category Logistic Regression (BLR)), these models were then used to predict the response variable given a test set of data. The performance of each model was then reviewed and discussed.

The results show that the machine learning techniques outperformed the logistic regression methods. The best set of results produced for Dataset 1 was an Area Under the Curve (AUC) of 75.43% by the BLR model, an accuracy of 79.93% by the RF model, a Kappa score of 0.3385 by the SNN and a mean balanced accuracy of 60.85% achieved by the SNN. Dataset 2 saw the SNN produce the best AUC, Kappa score and mean balanced accuracy with values of 62.48%, 0.1960 and 54.66% respectively. The best accuracy score was achieved by the RF model (78.1%). Dataset 3 and Dataset 4 saw the same outcomes arise. The RF model produced the best AUC and accuracy, 71.58% and 74.5% for Dataset 3 and 63.04% and 75.51% for Dataset 4. However the SNN produced the best kappa scores and mean balanced accuracy values for both datasets, 0.3719 and 62.31% for Dataset 3 and 0.2576 and 57.56% for Dataset 4.

The results of the study show that the machine learning techniques outperform the logistic regression methods in status and severity prediction of South African Covid-19 laboratory test data and that the best performing machine learning technique was the self-normalising neural network. Overall the models and networks performed the best when using Dataset 3. The results provide evidence that the machine learning techniques can be used as an indicative tool for Covid-19 status and severity prediction rather than a confirmation tool.

Declaration

I declare that this research titled “Logistic Regression Methods Versus Machine Learning Techniques In Status and Severity Prediction of South African Covid-19 Laboratory Test Data," is my own work. This dissertation is being submitted for the degree of Master of Science in Statistics to the University of the Witwatersrand, Johannesburg and has been conducted under the supervision of Dr. Charles Chimedza and Dr. Farai Mlambo. It has not been submitted before for any degree or examination in any other University.

June 12, 2023
Mark Strickett

Ethical Clearance

Ethical clearance was granted by the Human Research Ethics Committee (Medical) of the University of the Witwatersrand and clearance certificate No. M221097 was issued. Furthermore the Academic Affairs and Research department of the NHLS granted this study permission to use the data, reference No. PR2227145.

Chapter 1

Introduction

1.1 Introduction

The Coronavirus disease 2019 (Covid-19) has impacted and affected everyone across the world unequally (NaserGhandi et al., 2020; Pokhrel and Chhetri, 2021) and even though the virus is now under control in countries around the world and has less of a social and economic impact, the virus still poses a threat to many people in South Africa (Naidu, 2020). Due to the increased number of people being vaccinated there are now fewer testing locations available for people to test. Knowledge of one's Covid-19 status is still important as it impacts the lives of family, friends, co-workers and the general public. Therefore, having tools such as the logistic regression and machine learning modelling techniques, in conjunction with the Reverse Transcriptase Polymerase Chain Reaction (RT-PCR), antigen and rapid Covid-19 tests only enables people to be more informed about their Covid-19 infection status. It can subsequently speed up the testing process and boost the accuracy of testing and thus help control the spread of Covid-19.

Machine learning has established itself as an acceptable method for prediction in recent times because of the advances in technology and availability of data. This has led to the more 'classical' statistic methods being somewhat left behind (Dangeti, 2017; Ij, 2018). Hence in this dissertation the goal is to directly compare the newer machine

learning methods against the classical statistics methods and review the performance of the two approaches.

1.2 Background

The Covid-19 virus is an infectious disease and “has been classified as a Global Pandemic by the World Health Organisation" (Harapan et al., 2020, p.667). Infection of the Covid-19 virus occurs through the transition of respiratory droplets (NaserGhandi et al., 2020). The virus can be spread through direct contact with respiratory droplets. This can occur as a results of coughing or sneezing for example. The virus can also be spread at a secondary level by means of indirect contact meaning transfer via contact with a contaminated surface or an object (for example a door-handle).

Covid-19 infections can be identified by using several different methods. There are three different tests that an individual in South Africa can take. The three tests are the Antibody test, the antigen test, and the RT-PCR test. The RT-PCR and antigen tests are defined as diagnostic tests and show the infection status of an individual. The diagnostic test can be further broken down with the RT-PCR test being a molecular test and the antigen test being a rapid test. The antibody test as its name suggests is an antibody test. It is used to detect antibodies for the Covid-19 virus in an individual’s blood. The antibody test is not a diagnostic test. The RT-PCR test has shown to be the most accurate test and hence is also the most common test (NaserGhandi et al., 2020).

The question that needs answering is, are there other techniques that can be used in alongside the RT-PCR test to speed up testing and improve the accuracy of the test results? Testing is an expensive process and this means that not everyone who may need to be tested in the country will have the necessary resources to be tested. This excludes people from the testing process thereby prohibiting accurately assessing the spread of the virus amongst the overall population (Yang et al., 2020).

Another factor to consider is an individual's location. An individual situated in an urban area will have greater access to testing facilities than an individual in a rural area (Yang et al., 2020). It can also increase the time taken to obtain the results once tested (Yang et al., 2020).

Knowing one's infection status is very important and therefore it is crucial to have the ability to be tested and to obtain the test results in good time to manage and control the spread of Covid-19.

1.3 Research Question

Can the Covid-19 infection status and severity (if positive) of an individual be predicted using statistical modelling and machine learning techniques based on routine laboratory test data?

1.4 Aims and Objectives

The aims and objectives of this study are described below.

1.4.1 Aims

The aims of the study are to carry out exploratory data analysis on the Covid-19 data. To determine significant variables in predicting Covid-19 status within the Covid-19 data and to model the status and severity of individuals in the Covid-19 dataset using a variety of classical Statistical models and machine learning methods. The study aims to predict whether an individual is positive (the severity of the case) or negative for Covid-19.

1.4.2 Objectives

The objectives of the study are:

1. To carry out exploratory data analysis:
    - Confounding variable analysis
    - Factor analysis
    - Principal component analysis
2. To fit the following machine learning models to the data:
    - Deep learning neural networks
    - Random forest
3. To fit logistic regression models to the data:
    - Multinomial logit model
    - Baseline-category logit model
    - Ordinal logit model
4. To measure the performance of the different modelling methods in predicting outcomes

Chapter 2

Literature Review

2.1 Introduction

This section features the literature and theoretical review of methods used in this research. The literature review demonstrates how similar techniques to those used in this research are used in different contexts and what the outcomes of those processes were. The theoretical review outlines and explains the methods and process that will be used in the research.

2.2 Literature Review

A recent study was conducted to assess the likelihood of predicting Covid-19 by utilising routine laboratory blood tests and Machine Learning (ML) techniques (Yang et al., 2020). ML is a subset of Artificial Intelligence (AI) that gives software and systems the ability to learn and adapt from historical data without being programmed to do so. The Yang et al. (2020) study created a ML model that used 27 laboratory test results as well as age, sex, and race to predict if an individual was Covid-19 positive or negative. The data used was obtained from a New York hospital and was based on test results from 5 893 patients. The research team used four different machine learning models and they established that the gradient boosting decision tree model was the superior model based on the selected data and the number of variables. The model achieved an

Area Under the Curve (AUC) of 0.854, which was contained in the 95% confidence interval (0.829-0.878). They concluded that using routine laboratory test results offers the chance to identify Covid-19 earlier than the RT-PCR test and therefore could be used in conjunction with the RT-PCR test (Yang et al., 2020).

Being able to obtain the best test results more rapidly in an under resourced country, like South Africa, forms part of the motivation for this research project. The objectives are to determine if; other statistical and machine learning models can be more accurate, if the South African data has similar results, whether more or less variables will change the results and whether having a larger dataset (more individual's test results) changes the outcome.

Since the Covid-19 outbreak has been classified as a pandemic, there have been several epidemiological models created across the world with the aim of projecting the number of cases and deaths associated with the outbreak. The Pinter et al. (2020) study set out to predict the number of Covid-19 infected patients and to predict the mortality rate of the Covid-19 outbreak using a hybrid machine learning approach rather than the common susceptible-infected-resistant based approach. The study referred to as a base for this proposed research used two machine learning methods namely, multi-layered perceptron-imperialist competitive algorithm (MLP-ICA) and adaptive network-based fuzzy inference system (ANFIS) on their obtained time series Covid-19 data. They created two scenarios for time series prediction.

The models were trained and then validated against the actual outcomes after which it compared the performance of each method in predicting the total cases and total mortality rate. The Pinter et al. (2020) study used root mean square error (RMSE) and the coefficient of determination (R^2) to compare the two methods. The MLP-ICA method produced a RMSE of 167.88 and a coefficient of determination of 99.771% for predicting the total cases. This was marginally better than what the ANFIS method could achieve, RMSE of 194.10 and coefficient of determination of 95.63%. The ANFIS method produced the following results for predicting the total mortality rate: RMSE of 15.25 and a coefficient of determination of 74.91%. This was again outperformed

by the MLP-ICA method, which achieved a RMSE of 8.32 and a coefficient of determination of 99.86% (Pinter et al., 2020).

Overall, the Pinter et al. (2020) study produced promising results and is motivation for fitting similar methods such as self-normalising neural networks and random forests to model the Covid-19 outbreak.

Knowing the severity of a Covid-19 patient would allow health care facilities to improve care delivery and allocation of finite resources. The ability to predict a Covid-19 patient's severity therefore becomes very valuable for both the patient and the health care facility treating the patient. The Aktar et al. (2021) study set out to predict the severity of Covid-19 patients using machine learning, correlation, and comparison methods. Two datasets, that were relatively small, were used. The first data set consisted of 89 positive Covid-19 patients and the second consisted of 1945 positive Covid-19 patients. Using the two data sets the study identified the most significant and associative parameters (Aktar et al., 2021) using two different tests. Chi-square tests were used to identify the categorical variables and student t-tests were used to determine which continuous variables to retain. The tests were conducted at a significance level of 5%.

Two different analysis approaches were used to predict the severity of each patient, namely student t-test and machine learning methods. Gradient boosting methods performed the best and returned a 89% accuracy, whilst the other methods returned accuracy values of greater than 80%. Aktar et al. (2021) demonstrated that blood parameters are significant in predicting Covid-19 patient severity. The study concluded that blood of covid-19 patients can be used to identify high and low risk patients and this gives hospitals the opportunity to optimize facilities and resources when treating Covid-19 patients.

Wibowo et al. (2021) did a study on creating a prediction tool that can be used to control the spread of the outbreak in Indonesia. The model chosen to do this was a logistic regression model. The paper looks at the time frame of 2nd of March 2020 to the 12th of November 2020. This overlaps with the time frame used in this research paper.

This makes the Wibowo et al. (2021) study a relevant reference to the results of this study. This study looked at two groups of data namely, patients who were infected with Covid-19 and patients who recovered from Covid-19. The study found logistic regression values for the infected to be 8.114748 for the intercept parameter and 0.750743 for the slope parameter. The values for the recovered patients were 9.360925 for the intercept parameter and 0.788334 for the slope parameter. The prediction accuracy of the model was tested and gave the same results for both the infected and recovered patients. These values were a mean absolute error (MAE) of 2%, a mean square error (MSE) of 0% and a R^2 value of 99%.

The Wibowo et al. (2021) study concluded that there was “an upward trend in both Covid-19 sufferers and people recovering from Covid-19 in Indonesia."

Xu et al. (2020) did a study to “identify the determinants of illness severity of Covid-19 based on ordinal responses." The dataset used in the study comprised of patients from three different provinces of China. The time frame used in this study was from the 1^st of January 2020 to the 8^th of March 2020. The patients were split into the following categories “moderate, severe and critical illness" (Xu et al., 2020). The study then used two ordinal logistic regression models to identify the explanatory variables of illness severity. These models were a univariate model and multivariate model. The risk factors were explored using a cumulative ordinal logit model. The results of the Xu et al. (2020) study showed that “significantly higher proportion of patients showing abnormal alanine aminotransferase (ALT) and aspartate aminotransferase (AST), and significantly higher levels of C-reactive protein, neutrophil count, serum potassium, cardiac troponin I (cTnI) myohaemoglobin, PCT, brain natriuretic pep tide (BNP) and D-dimer were observed in the critically ill group and the severely ill group than in the moderately ill group." The study also found significantly lower “levels of haemoglobin and serum albumin" present in the critically and severely ill groups.

The Xu et al. (2020) study successfully identified determinants of illness severity which could then be used to optimise hospital resources and treat patients in the most effective way possible.

Logistic regression has been used to predict Covid-19 as described above but it is not limited to Covid-19. The Kurt et al. (2008) and Nusinovici et al. (2020) studies created logistic regression models to predict two different diseases namely, coronary artery disease and major chronic diseases respectively.

The Kurt et al. (2008) study performed a retrospective analysis on 1245 patients. A logistic regression model was used to predict if a patient had coronary artery disease. The model used 8 predictor variables, “age, sex, family history of CAD, smoking status, diabetes mellitus, systemic hypertension, hypercholesterolemia, and body mass index" (Kurt et al., 2008). The logistic regression model produced an AUC of 75.3%.

The Nusinovici et al. (2020) conducted a prospective study that compared logistic regression for prediction to ML algorithms. The comparison of the two methods was done by looking at the prediction ability of the two methods for the following cardiovascular diseases, hypertension, chronic kidney disease, and diabetes. The study used 6 762 Asian adults as a dataset. The study reflected that the Logistic regression model outperformed the ML algorithms in predicting chronic kidney disease and diabetes with AUCs of 90.5% and 76.8% respectively. As for predicting cardiovascular diseases and hypertension the logistic regression model performed less than 1% worse then the ML algorithms. The overall performance of the logistic regression model indicates that logistic regression is a powerful prediction tool for disease prediction. The overall goal of this study is to determine which method, logistic regression or machine learning, is best at predicting the status and severity of Covid-19 based on South African data. At the heart of this goal is the battle between regression and machine learning in determining which method is better at predicting outcomes.

Looking at the capability of both, machine learning and logistic regression, in predicting Covid-19 status and severity the above mentioned studies have all had success in different context and with different datasets. However the desired situation is a competition between both methods on the same dataset. The Mohammadi et al. (2021) study does exactly that it compares the performances of an artificial neural network (ANN) and a logistic regression model in characterising Covid-19 infected patients in Iran. The study used 29 variables and 1043 suspected Covid-19 patients who were all

hospitalised. Statistical analysis of the variables was conducted using an analysis of variance (ANOVA) test, chi-square test and Krusal-Wallis test for continuous, categorical and ordinal variables respectively (Mohammadi et al., 2021).

The logistic regression model used Stratified Random Sampling (SRS) for testing and training sampling because of the imbalanced nature of the data (Mohammadi et al., 2021). The ANN used was a Multilayer Perceptron Neural Network (MLPNN). The neural network had the following characteristics: "...input layer, one hidden layer, and the output layer" (Mohammadi et al., 2021, p.307). The neural network was trained using a "...scaled conjugate gradient (SCG) back-propagation algorithm" (Mohammadi et al., 2021, p.307).

The results of the Mohammadi et al. (2021) study were; the logistic regression model had an AUC of 99.2% whilst the ANN had an AUC of 99.9%. Further more the other performance measures showed that the ANN model achieved a 100.0% sensitivity, a specificity of 97.6% and an achieved an accuracy score of 99.4%. On the other hand the logistic regression model achieved a sensitivity of 99.1%, which is 0.9% down on the ANN, however it did achieve the same a specificity of 97.6%. Lastly the accuracy of the logistic regression model was slightly down (0.7% less) and achieved an accuracy of 98.7% (Mohammadi et al., 2021).

Overall the Mohammadi et al. (2021) study shows that both models perform admirably and are successful in predicting the status of Covid-19 patients but ultimately the ANN model outperforms the logistic regression model.

The Al-Hindawi et al. (2021) study debated the pros and cons of traditional statistics methods and machine learning methods in the context of the Covid-19 outbreak. The study notes that machine learning is still a novel method and has very few clinically used programs for prediction and prognostication, whilst traditional statistic methods have been widely implemented in healthcare (Al-Hindawi et al., 2021).The Acute Physiology and Chronic Health Evaluation (APACHE) score is used in hospitals in the Intensive Care units and the Model of End-Stage Liver Disease (MELD) is used

in clinics, both of these scores are based on linear regression models. The study further highlights the opaque and obfuscate nature of neural networks (Al-Hindawi et al., 2021). This is a complete contrast to regression analysis which can be easily interpreted and provides causality (Al-Hindawi et al., 2021). that is not to say machine learning should not be considered because its methods can minimise bias whilst still maximising data utilization an area where linear regression falls short in comparison.

The Al-Hindawi et al. (2021) study concludes that the increasing availability of highly dimensional data provides more opportunity for machine learning models to be implemented in everyday use but this however does not make classical statistical analysis and modelling inadequate. Ultimately each method has its merits and the context in which the method is used will determine which method will be superior.

2.3 Blood Parameters Identified

Based on a number of sources the following parameters found in Table 2.1 seem to be the most significant parameters in predicting Covid-19 infection status an severity from Blood tests (Ahamad et al., 2020; Aktar et al., 2021; Hesse et al., 2020; Wu et al., 2020).

Table 2.1: Blood Parameters Identified

White Blood Cells	Lymphocyte Count	Neutrophil Count
Basophil Count	Monocyte Count	Red Blood Cells
Red Blood Cell Distribution Width	Mean Corpuscular Haemoglobin	Mean Corpuscular Volume
Lactate Dehydrogenase	Ferritin	C-reactive Protein
Glucose	Albumin	Globulin
Total Protein	Haemoglobin	Haematocrit
Indirect bilirubin	Creatine	Alkaline Phosphate
Sodium	Chloride	Magnesium
Calcium	Anion Gap	Troponin
Lipase	Procalcitonin	

2.4 Theoretical Review

The theoretical review defines and mathematically explains the processes and concepts used in this study namely, Predictive Mean Matching Imputation, Multicollinearity, Confounding Variables, Multinomial Logistic Regression, Baseline-Category Logit Model, Ordinal Logit Model, Deep Learning and Neural Networks and Random Forests.

2.4.1 Imputation: Predictive Mean Matching Imputation

Predictive mean matching (PMM) imputation is a type of nearest neighbour imputation. It uses a scalar predictive mean function to determine what is the nearest neighbour. The predictive mean matching method is split into two parts/steps. (1) Estimation; the predictive mean function is estimated. (2) Imputation; the missing values are identified and for each missing value its nearest neighbour is found in the data, based on the predictive mean function. The value of the nearest neighbour is used to impute the missing value (Yang and Kim, 2020).

In this study the Predictive Mean Matching Imputation that will be used is from *mice* package in R. The method for the Multivariate Imputations by Chained Equations (MICE) van Buuren et al. (2015) will be Predictive Mean Matching (pmm in coding software) with parameters of 10 multiple imputations and 10 iterations.

2.4.2 Multicollinearity

Multicollinearity is a statistical occurrence where two or more independent/predictor variables have a linear relationship between them (Daoud, 2017; Rekkas, 2011). To test for multicollinearity in this study the variance inflation factor (VIF) will be used. The formula for the VIF, as presented by Senthilnathan (2019), is displayed below in Equation 2.1.

$$VIF = \frac{1}{(1 - R^2)} \tag{2.1}$$

where R^2 is the correlation coefficient.

Table 2.2: VIF Score and Multicollinearity Interpretation

VIF Score	Interpretation
1	No Multicollinearity
1 - 5	Low Level of Multicollinearity
5 - ∞	High Level of Multicollinearity
∞	Perfect Multicollinearity

To conduct this test the *vif()* function will be used from the *car* package in R (RDocumentation, 2020).

2.4.3 Confounding Variables

Confounding variables or “confounders" (as they are sometimes known) are variables that are correlated with the dependent variable and other predictors. The correlation can be positive or negative. Pourhoseingholi et al. (2012) defines these as variables that affect other variables by distorting their true effects/relationships. Prysby et al. (2013, p.1) says “...they confound the ‘true’ relationship between two variables."

The Mantel-Haenszel test will be used to investigate confounding variables as well as using analysis of covariance ANCOVA (Pourhoseingholi et al., 2012).

2.4.4 Boruta Feature Selection

The Boruta feature selection method was created to detect all significant variables in a classification framework (Degenhardt et al., 2019). The primary objective of this method is to use statistical analysis and several RF runs to evaluate the significance of the true predictor factors with that of the random, so-called shadow variables (Kursa et al., 2010). The number of predictor variables is doubled in each run by including an extra copy of each variable. These shadow variables’ values are produced by randomly permuting the original values across observations, which breaks the relationship with the result. On the expanded data set, an RF is trained, and the variable importance

values are gathered. A statistical test is run for each variable to compare its importance value to all of the shadow variables.

2.4.5 Logistic Regression

Logistic regression is a statistical method used for classification and predictive analysis. It is a regression method, like any other, in which the relationship between the response variable and the predictor variables are investigated. The characteristic that makes logistic regression unique is that the response variable can be continuous or categorical. (DiGangi and Hefner, 2013; Nick and Campbell, 2007; Wang et al., 2019)

2.4.5.1 Multinomial Logistic Regression

Multinomial logistic regression is an extension of simple binary logistic regression, the difference being multinomial logistic regression allows for three or more categories of the dependent variable. It is used to predict categorical dependent variables with multiple explanatory variables, dichotomous or continuous. Maximum likelihood estimation is used to determine the probability of being in a specific category. The assumptions of multinomial logistic regression are independence of the dependent variable and no multicollinearity. An attractive property of multinomial logistic regression is that it does not assume linearity, normality, or homoscedasticity. A disadvantage of multinomial logistic regression is that it struggles to predict dependent variables that are ordered. (Erkan and Yildiz, 2014; Starkweather and Moske, 2011)

Let $\delta_{it} = P(Z_i = t), t = 1, 2, ..., T, i = 1, 2, ..., n$. The Multinomial logits are given by

$$log\left(\frac{\delta_{it}}{1-\delta_{it}}\right). \tag{2.2}$$

The Multinomial logit model has form

$$log\left(\frac{\delta_{it}}{1-\delta_{it}}\right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + ... \tag{2.3}$$

or equivalently,

$$\delta_{it} = \left(\frac{\exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + ...)}{1 + \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + ...)}\right) \tag{2.4}$$

where α and β_t are the parameters of the model and x_i represents the independent variables.

2.4.5.2 Baseline-Category Logit Model

Baseline-category logit modelling is a process where multi-category logit models pair a baseline category with all remaining categories. With the interest being to determine "...the odds of being in one category relative to being in a designated category, called the baseline category, for all pairs of categories" (Chow, 2022, p.8). Binary logistic regression is the basis on which baseline-category logistic regression is developed. The assumptions of baseline-category logistic regression are independence of the dependent variable and no multicollinearity. (Agresti, 2003; Brophy et al., 2011; Erkan and Yildiz, 2014; Huang, 2017)

Let $\delta_t = P(Z = t), t = 1, 2, ..., T$. The Baseline-category logits are

$$log\left(\frac{\delta_t}{\delta_T}\right). \tag{2.5}$$

The Baseline-category logit model has the form

$$log\left(\frac{\delta_t}{\delta_T}\right) = \alpha_t + \beta_t x \tag{2.6}$$

or equivalently,

$$\delta_t = \delta_T * \exp(\alpha_t + \beta_t x) \tag{2.7}$$

where α_t and β_t are the parameters of the model and x is the independent variable.

2.4.5.3 Ordinal Logit Model

Ordinal logit modelling is a statistical method that is used to model an ordered categorical response variable (ordinal variable) and explanatory variables. Parry (2016, p.1) says "...an ordinal variable is a categorical variable for which there is a clear ordering of the category levels." The assumptions of ordinal logistic regression are no multicollinearity, either ordinal, continuous or categorical independent variables, and the dependent variable is ordered and has at least three categories (Erkan and Yildiz,

2014). Ordinal logistic regression is an extension of binary logistic regression (Bender and Grouven, 1997; Parry, 2016).

Let Z be an ordinal response variable with categories $1, 2, ..., T$ then,

Let $\delta_i = P(Z = i)$. The cumulative probabilities are

$$P(Z \leq t) = \delta_1 + ... + \delta_t \tag{2.8}$$

where $t = 1, 2, ..., T$. The cumulative logits are

$$\begin{aligned} logit[P(Z \leq t)] &= log\left(\frac{P(Z \leq t)}{1 - P(Z \leq t)}\right) \\ &= log\left(\frac{P(Z \leq t)}{P(Z > t)}\right) \\ &= log\left(\frac{\delta_1 + ... + \delta_t}{\delta_{t+1} + ... + \delta_T}\right) \end{aligned} \tag{2.9}$$

where $t = 1, 2, ..., T - 1$.

2.4.6 Artificial Neural Networks

An artificial neural network (ANN) is a mathematical representation of the neural network found in the brain. NNs are used for classification and prediction of non-linear problems (Sharma et al., 2017; Wang, 2003). It is a network made up of 3 general layers, the input layer, the hidden layer and the output layer which all consist of a number of nodes/neurons that are all connected to each other (Wu and Feng, 2018). The nodes across the different layers are connected to each other and carry an associated weight. The activation function then specifies how the input weight is transformed into an output (Sharma et al., 2017). The activation functions provide the non-linearity required to solve complex non-linear functions and can be unique to layers or specific nodes (Wu and Feng, 2018). Figure 2.1 and 2.2 show the architecture of a neural network and the structure of a node within a neural network. ANNs are not without limitations. ANNs are difficulty to interpret due to its "black box" nature of the interaction between the different neurons. Furthermore, ANNs are more complex and time consuming and

can lead to overfitting (Walczak, 2019).

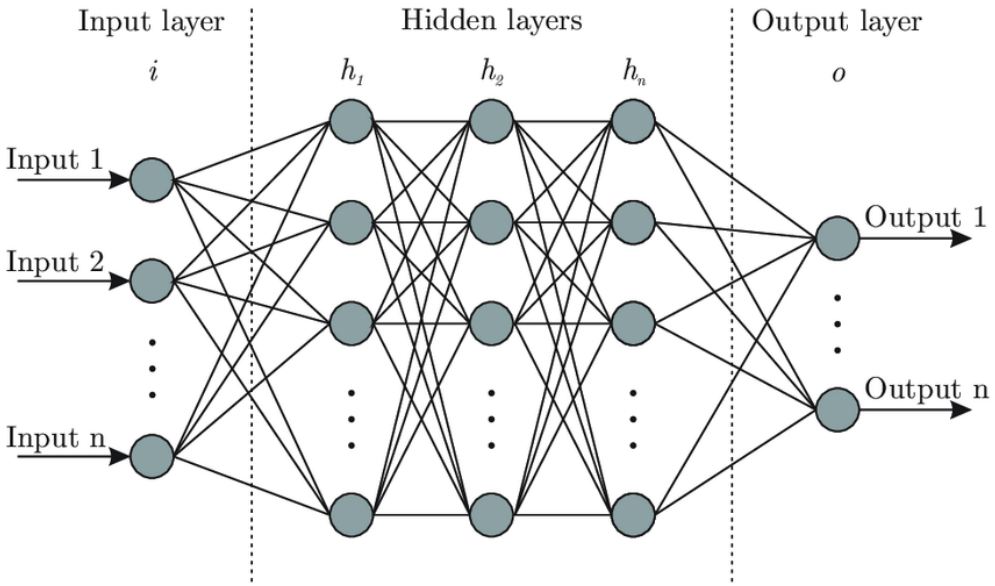

Figure 2.1: Artificial Neural Network Architecture (Bre et al., 2018)

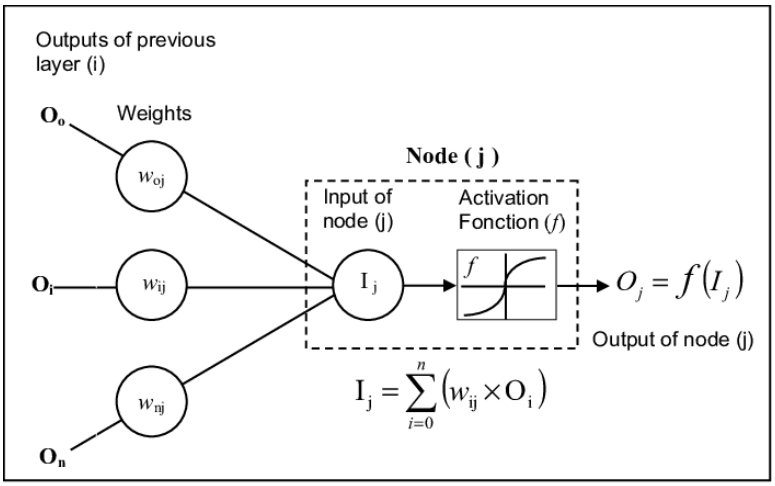

Figure 2.2: Structure of a Node within an Artificial Neural Network (Ghedira and Bernier, 2004)

The output of an ANN can be computed using the following equation:

$$y_k = \eta \left(b_k + \left(\sum_j \xi(b_j + \sum_i v_{ij} x_i) \right) w_{jk} \right) \tag{2.10}$$

where y_k is the k^{th} output of the network, w_{ij} is the weight on the connection between the j^{th} hidden layer neuron and the k^{th} output neuron, b_k is the bias for the k^{th} output neuron, η is the activation function for neurons in the output layer, v_{ij} is the weight on the connection between the i^{th} input and the j_{th} hidden layer neuron, b_j is the bias for the j^{th} hidden layer neuron and ξ is the activation function for neurons in the hidden layer. (Mbuvha, 2020)

2.4.7 Random Forest Modelling

Random Forest methods are used for model prediction by taking a collection of independent trees and averaging them. It can be used in several models because it is easy to train and tune. It has its own validation process by using the Out-Of-Bag samples to calculate the test error. Random forests do not over fit because of the law of large numbers (the average of a set of results tends to the expected value as the number of trials tends to infinity) therefore, making random forests an effective prediction tool for classification and regression problems (Breiman, 2001; Nasejje, 2020). There are however limitations to using random forest modelling, predictions may take a considerable amount of time depending on what is being predicted. Further, being a predictive tool means that random forest modelling cannot describe the relationships between variables (Louppe, 2014).

Definition 1. *A random forest is a classifier consisting of a collection of tree-structured classifiers*

$$f(\boldsymbol{x}, \Psi_k),\ k = 1, ... \tag{2.11}$$

where the Ψ_k are independent identically distributed random vectors and each tree in the random forest makes a single vote for the most popular class at input $\boldsymbol{x}$ (Breiman, 2001).

Having drawn the training set at random from the distribution of the random vector $\mathbf{x}, Y$ and with an ensemble of classifiers $f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})$ the margin function can be define as:

$$mg(\mathbf{X}, Y) = av_k\mathbf{I}(f_k(\mathbf{X}) = Y) - \max_{j \neq Y} av_k\mathbf{I}(h_k(\mathbf{X}) = j) \tag{2.12}$$

where $\mathbf{I}()$ is the indicator function.(Breiman, 2001)

The margin measures the degree to which the mean number of votes at **X**, Y for the correct class eclipses the mean vote for any other class. The greater the margin the greater the degree of confidence in the classification.

Breiman (2001) defined the generalization error as follows:

$$PE^* = P_{\mathbf{x},Y}(mg(\mathbf{x}, Y) < 0) \tag{2.13}$$

where the subscripts **x**, Y indicate that the probability is over the support of **x**, Y.

In random forests, $f_k(\mathbf{x}) = f(\mathbf{x}, \Psi_k)$. Breiman (2001) proved that for the Strong Law of Large Numbers and the tree structure that:

Theorem 1. *As the number of trees increases, for almost surely all sequences* $\Psi_1, \Psi_2, ...,$ PE^* *converges to*

$$P_{\boldsymbol{x},Y}(P_\Psi(f(\boldsymbol{x}, \Psi) = Y) - \max_{j \neq Y} P_\Psi(f(\boldsymbol{x}, \Psi) = j) < 0) \tag{2.14}$$

Random forests converge to a generalization error as the number of trees increase, rather than over-fitting. This is as a result of Theorem 1.

The Random Forest Algorithm presented by Nasejje (2020) is set out as follows:

1. Construct bootstrap datasets of equal size to the training dataset.
2. This process will exclude a portion of the training dataset which is called out-of-bag and is used in the validation process.
3. Grow a random forest tree based on the bootstrapped data.

- At each node choose a number of features that could be used as the split feature.
- Thereafter choose the best split feature and split point based on a predetermined impurity measure.
- Split node into 2 daughter nodes.
- Repeat this process until a minimum node size is obtained.

4. Output the ensemble of trees.

5. Make prediction on new data point (test dataset).

In a classification problem if the class prediction of the b^{th} tree in the forest is $\hat{C}_b(x)$, then

$$\hat{C}_{rf}^{B} = \text{majority vote}[\hat{C}_b(x)]_{b=1}^{B} \tag{2.15}$$

where B is the number of trees in the forest.

2.4.8 Model Performance Measures

These are the performance measures that will be used to evaluate the performance of each of the models created in this study. These performance measure will be compared to one another so as to determine the best performing model.

Figure 2.3 was obtained from Beheshti (2022) and shows the structure of a confusion matrix. The performance measures' formulae will be based on this structure of the confusion matrix.

1. **Accuracy**
   Accuracy is the number of correct predictions over the total number of predictions (Quinn, 2020).

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.16}$$

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

Figure 2.3: Structure of a Confusion Matrix

2. **Sensitivity**

   Sensitivity is defined as the proportion of the “positive class" correctly predicted, where the “positive class" is one of the categories chosen by the research (Beheshti, 2022).

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.17}$$

3. **Specificity**

   Specificity is defined as the proportion of the “negative class" correctly predicted, where the “negative class" is a category chosen by the research (Yedida, 2018).

$$Specificity = \frac{TN}{FP + TN} \tag{2.18}$$

4. **Cohen’s Kappa**

   Cohen’s Kappa statistic shows how well classifiers predictions matched the actual dependent variable responses of the dataset, while controlling for the accuracy of a random classifier (Quinn, 2020).

$$K = \frac{p_0 - p_e}{1 - p_e} \tag{2.19}$$

   Where p_0 is equal to probability of agreement and p_e is equal to probability of random agreement. Table 2.3 indicates how Cohen’s Kappa score is interpreted.

Table 2.3: Cohen's Kappa Score (McHugh, 2012)

Cohen's Kappa Score	Interpretation
≤ 0	No agreement
0.01–0.20	None to slight agreement
0.21–0.40	Fair agreement
0.41– 0.60	Moderate agreement
0.61–0.80	Substantial agreement
0.81–1.00	Almost perfect agreement
1.00	Perfect agreement

5. **No-Information Rate (NIR)**
   The NIR is a measure that shows the accuracy obtainable if the majority class was always predicted (Quinn, 2020).

6. **Receiver Operating Curve (ROC)**
   The ROC is a plot of sensitivity against 1 - specificity (Mellor et al., 2013).

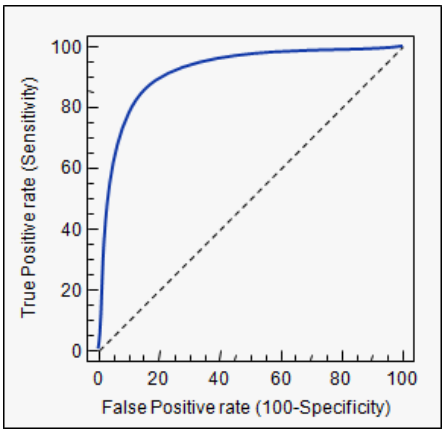

Figure 2.4: Example of a Receiver Operating Curve obtained from Yedida (2018).

7. **Area Under Receiver Curve (AUC)**
   The AUC is the area between the diagonal line and the ROC curve and ranges from 0.5 up to 1 (Mellor et al., 2013).

Figure 2.5 Shows how you create a binary confusion matrix from a non-binary confusion matrix in order to calculate the performance measures mentioned above.

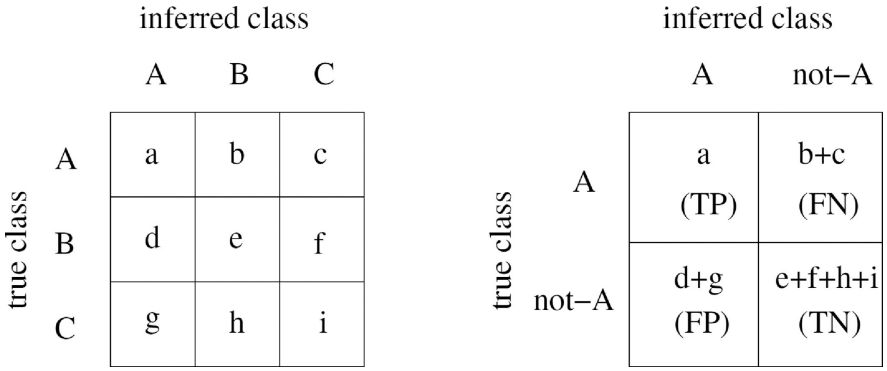

Figure 2.5: Structure of a Non-binary Confusion Matrix

Chapter 3

Methodology

3.1 Introduction

This section features the research outline, data description, exploratory data analysis (EDA), machine learning methods and logistic regression methods. The research outline shows the route this study intends to go to achieve its objectives and produce accurate and informative results. The data being used in this study is set out and explained in the data description subsection and the EDA further explains some of the properties and relationships of the data. Then finally the two method sections explain how the four different methods will be carried out to produce results which can then be interpreted and discussed.

3.2 Research Outline

1. **Clean and prepare the data**
   This entails removing any irrelevant variables, errors, or invalid responses.

2. **Imputation**
   This entails assigning a value to a variable that had a missing value to begin with.

3. **Exploratory data analysis**
   The properties and statistics of the dataset will be explored and processes such as confounding variable analysis, factor analysis and principal component analysis will be carried out.

4. **Variable selection and dataset creation**
   Univariate p-value significance tests as well as Boruta feature selection will be used to determine which variables to include and exclude.

5. **Split the dataset into training and test sets.**
   The ratio of the training set to test set will be 70:30 respectively, which is ideal for the methods used in this study as proved by Nguyen et al. (2021).

6. **Fit machine learning models**
   The random forest and deep learning neural network models will be fitted to the dataset with the aim of predicting the Covid category variable. The Covid category variable has four categories namely, severe, moderate, mild and negative.

7. **Fit logistic regression models**
   The baseline-category logit and ordinal logit models will be fitted to the dataset with the aim of predicting the Covid category variable. The Covid category variable has four categories namely, severe, moderate, mild and negative.

8. **Performance measures will be used to determine the best performing model.**
   The performance measures that will be used are Accuracy, Sensitivity, Specificity, Cohen's Kappa, NIR, ROC and AUC.

3.3 Data Description

The data being used in this project was provided by the University of the Witwatersrand - Faculty of Health Sciences and was originally obtained through the National Health Laboratory Service (NHLS) of South Africa.

The extracted data, which is the data used in the Hesse et al. (2020) study, is for all individuals who had a Covid-19 RT-PCR test via the NHLS from 1 March 2020 to 7 July 2020. The data has been pre-processed by Dr. Dieter van der Westhuizen. The characteristics of the data are as follows:

- The data is anonymous to prevent traceability.
- Chronic disease test data of 6 months prior to the Covid-19 RT-PCR test. The Chronic disease test is for diseases such as Human Immunodeficiency virus (HIV), Tuberculosis (TB) and Diabetes Mellitus (DM)
- Analyte data 7 days prior to and 14 days after Covid-19 RT-PCR test. This filtering was done by Dr. Dieter van der Westhuizen and was done to ensure relevancy of the analyte data.
- All individuals are aged 18 and older.
- All individuals with indeterminate RT-PCR test results were removed.

The size of the data after the preliminary cleaning is **61 330 observations on 50 variables**.

3.4 Data Preparation and Performance Evaluation

Once the final dataset is obtained and no more variable selection or imputation needs to be done then the modelling process can begin. The process involves splitting the dataset into two sets namely a test set and a training set. The training and test sets are divided into a ratio of 70:30 respectively. Simple random sampling was used to derive the training and test sets. After which each of the respective prediction methods can

be fitted.

Once the data had been fitted to the respective models, the models were then be evaluated using 5-fold cross validation on the training data, which is the same process used in the Yang et al. (2020) study. This entailed the data being partitioned to create five approximately equal subsets. Once this had been done a subset from five available subsets was selected. This selected subset was then used as the validation set, whilst the remaining four subsets were used to train the model. This process was then repeated another four times to ensure that each subset had been selected as the validation subset.

The final models will be obtained for the training set after cross-validation has taken place. The final models will then be used to predict the test sets' Covid-19 classification outcomes (severe, moderate, mild, negative). Finally, a confusion matrix will be used to compare the test sets and the respective models predictions.

3.5 Exploratory Data Analysis

Once the dataset has been cleaned and prepared exploratory data analysis will be done. The goal of the EDA is to gain a better understanding of the dataset and its properties. The dataset will be taken and an investigation into the percentage split of the four categories (severe, moderate, mild, negative) will be done. The properties of the data such as age, race and gender will be explored. Then confounding data analysis, factor analysis and principal component analysis will be carried out. This will help determine the relationships between the variables and which variables best explain the data.

3.6 Machine Learning Methods

This section outlines the two machine learning methods that will be used namely, random forest modelling and a self normalising neural network.

3.6.1 Random Forest Modelling

The random forest model will be created using the Kuhn et al. (2020) *train* package in R. The *rf()* (random forest) function from the *train* package will be used.

3.6.2 Self-Normalising Neural Network (SNN)

A variety of SNNs will be implemented using R-studio with different number of layers, hidden layers, and neurons in the different layers. The SNN will be created using the *remotes, keras, tensorflow* packages in R.

SNNs are able to produce high-level abstract representations and automatically converge to a mean of zero and unit variance. SNNs use scaled exponential units (SELUs) as activation functions. These activation functions introduce self-normalisation. (Klambauer et al., 2017)

Klambauer et al. (2017) stated “the convergence property of SNNs allows" multi layered deep networks to be trained, and “employ strong regularisation" which allows learning to be highly robust.

The different SNNs performance will be reviewed, after which the best performing model will be selected and will be used on the test dataset.

3.7 Logistic Methods

This section focuses on the classical statistics methods of this study namely logistic regression methods. The logistic regression methods that will be used are multinomial, baseline-category and ordinal logit models.

3.7.1 Multinomial Logit Model

The Multinomial logit model will have the following form:

$$log\left(\frac{\delta_{it}}{1-\delta_{it}}\right) = \alpha + \beta_1 age + \beta_2 gender + \beta_3 creatininepeGFR + ... \quad (3.1)$$

Where α is a constant determined by the function in R. β_t is the coefficient of the independent predictors and will also be determined by Maximum Likelihood estimation. The x values are from the selected variables after feature selection has taken place. See Table 4.1.

The Multinomial logit model will be implemented using the *multinom* R function from the *nnet* package.

3.7.2 Baseline-Category Logit Model

The Baseline-category logit model will have the following form:

$$log\left(\frac{\delta_t}{\delta_T}\right) = \alpha_t + \beta_t x \quad (3.2)$$

Where α is a constant determined by the function in R. β_t is the coefficient of the independent predictors and will also be determined by the function in R. The x value is the explanatory variable.

The Baseline-category logit model will be implemented using the “mblogit" R function from the “mclogit" package.

3.7.3 Ordinal logit Model

The Ordinal logit model will have the following form:

$$logit[P(Y \leq t|x)] = \alpha_t + \beta x \quad (3.3)$$

Where α_t is the individual intercept value for each cumulative logit determined by the function in R. β is the slope for all cumulative logits and will also be determined by

the function in R. The x value is the explanatory variable.

The ordinal logit model will be implemented using the *polr* and the *stargazer* R functions from the *MASS* and *stargazer* packages.

Chapter 4

Analysis

4.1 Introduction

This section contains the calculations and workings of this research. The original dataset was cleaned and prepared. This made the dataset ready for missing data analysis and thereafter imputation. After imputation was completed the dataset was ready to be used. Exploratory data analysis and variable selection was carried out, thereafter the data was split into training and test sets. Finally the machine learning models and logistic regression models were created.

4.2 Cleaning and Preparation of the Data

The original dataset had 1 870 309 observations and 152 variables. The dataset was filtered to remove all duplicates. This left the dataset with 842 197 observations and 81 variables. The original dataset had a large number of duplicates as each time an individual was moved from one ward to another they were recorded. The large number of initial variables is a result of the primary users of the data coding duplicate variables for the primary study. The dataset was then filtered to results that were obtained within a 3 week period around the RT-PCR test (7 days before and 14 days after) leaving it with 61 330 observations and 81 variables, see Appendix B.1. This dataset was then further investigated and non-essential variables were removed. Non-estential in that they were

describing characteristics of the individual for example the paient ID number given to the individual by the hospital or the amount of test results for an individual. These variables were patientid, hba1c_cat, amnttestresults, nvals, registereddatecov2pcr, registereddatebiochem, and datedif. Finally 619 individuals were removed as the dataset had no gender for these individuals. This left the dataset with 60711 observations and 74 variables after cleaning but before further preparation.

Preparation began by coding in the Covid-19 category variable (negative, mild, moderate and severe) this was done using the RT-PCR test result, the ward that each individual was admitted to and the level of care each individual received. Once the Covid-19 category variable was coded all the variables used to code the Covid-19 category variable were removed. This was done to remove all linearly dependent variables and to ensue no multicollinearity was created. This meant that the dataset had 60 711 observations and 64 variables.

Further preparation needed to be conducted with the dataset undergoing an investigation for any further linearly dependent predictor variables/ multicollinearity. This investigation was conducted using manual inspection as well as R. The manual inspection involved checking each of the variables and determining if two variables explained the same result, when this was the case one of the variables were removed. The investigation found that the following variables needed to be removed, nlr, viralloadunsuppressedvalue, viralloadunsuppressedtextvalue, hba1c_controlled, hivstatusvalue, hivviralloadstatus, hivstatusbyelisa, cd4cat,cd4cat_3, cd4cat_350_700, hba1c_cat_perpatient, hba1cbypatient, tbgenexpert, hivelisaencoded, viralloadtextencoded, cd4status, viralloadtextstatus, immaturecellsf, IL6combinedf, viralloadunsuppressednumvalue. This left the dataset with 60 711 observations and 44 variables.

4.2.1 Multiple Dataset Creation

The preparation phase of this research produced 4 dataset from the original data. The datasets were produced due to different rules/ideas on dealing with the missing values. All the datasets were created by applying the respective rules to the dataset of 60 711

observations and 44 variables.

The 1^{st} dataset was created by removing all variables with more than 90% missing values. This left the dataset with **60 711 observations and 39 variables**. The 2^{nd} dataset removed all variables with more than 50% missing values and therefore had **60 711 observations and 14 variables**. The 3^{rd} dataset was created by removing all variables with more than 90% missing values and all individuals who had more than 50% missing values. This left the dataset with **17 762 observations and 39 variables**. Finally the 4^{th} dataset was created by removing all variables with more than 50% missing values and all individuals who had more than 50% missing values. This left the dataset with **44 614 observations and 14 variables**. This was done in order to analyse the impact of the missing values on the results/study.

4.3 Imputation

Missing data analysis was then conducted on the datasets. The missing value percentage for each variable was conducted and is shown in Table 4.1 and the missing value plot is shown by Figure 4.1. Table 4.1 and Figure 4.1 show that for some variables there are a large number of missing values in the dataset. To overcome this problem PMM imputation will be used to obtain values for each of the variables. The *mice()* package in R was used to conduct the PMM imputation.

The imputation method was able to calculate values for all but three of the missing variables. The imputation method used was unable to calculate values for the following variables Myoglobin, Rapid screen and Procalcitonin. Hence these variables were dropped from the model.

Table 4.1: Missing Value Percentage of each Variable

Variable	Missing Value %
Gender	00.00
Age	33.42
Creatinine plus eGFR	18.21
Urea	28.55
Albumin	64.00
Total Bilirubin	62.48
Alanine Transaminase (ALT)	55.94
Aspartate Transaminase (AST)	69.20
Gamma Glutamyl Transferase (GGT)	66.21
Lactate Dehydrogenase (LDH)	84.37
Creatine kinase-MB (CK-MB)	98.81
High-sensitivity cardiac Troponin I	93.73
High-sensitivity Troponin (HS-Trop)	94.06
NT-proBNP	95.95
Procalcitonin Rapid screen	99.95
Procalcitonin Sensitivity	92.53
C-reactive Protein	49.29
Cortisol	98.43
HbA1c	84.46
Glucose Random	97.49
Ferritin	88.64
Myoglobin	98.92
Erythrocyte Sedimentation Rate	89.96
International Normalized Ratio (INR)	81.62
Fibrinogen	97.59
D-dimer	82.90
Red Cell Count	30.15
Haemoglobin	30.15
Haematocrit	30.18
Mean Cell Volume	30.18
Mean Cell Haemoglobin	30.18
Mean Cell Haemoglobin Concentration	30.18
Platelet Count	30.28
Neutrophils	72.80
Lymphocytes	72.81
Monocytes	72.84
Eosinophils	73.32
Basophils	73.42
Neutrophil-Lymphocyte Ratio (NLR)	72.81
CD4 count	82.97
HIV Status	0
TB Status	0

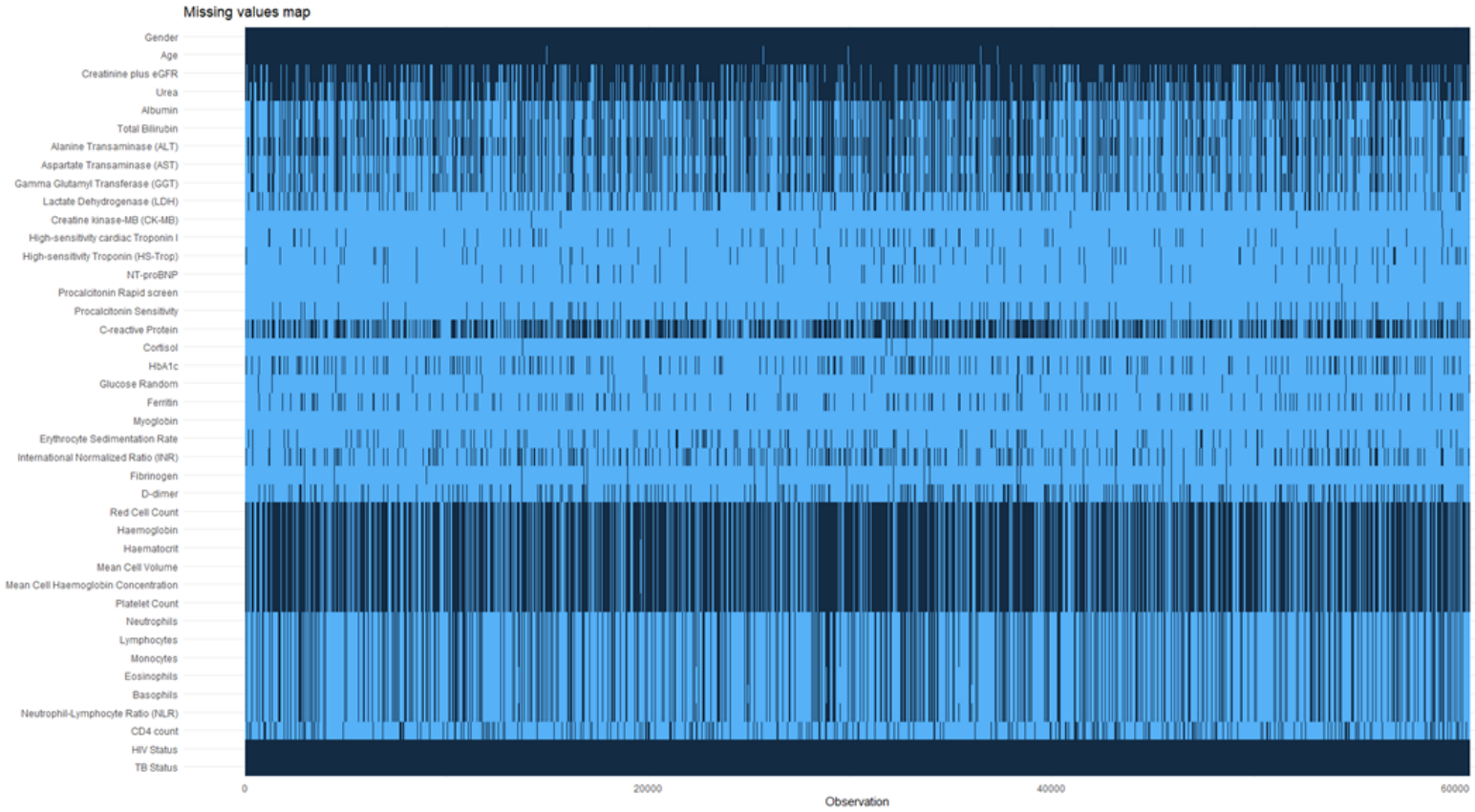

Figure 4.1: Missing Value Plot of each Variable

4.4 Exploratory Data Analysis

4.4.1 Descriptive Statistics

This section describes the datasets, the properties of the four datasets as well as the relationships between variables within the datasets. Bar graphs and box plots will be used to indicate the properties of the datasets and its variables.

Taking a look at the Covid-19 category variable (response variable) datasets one and two had 47 202, negative cases, which accounts for 77.75% of the datasets. Datasets three and four had 11 970 and 33 139 negative cases, which accounted for 67.39% and

74.28% of the datasets respectively. This indicated that the datasets were imbalanced. This was expected due to the type of data that was being dealt with in this research. Taking a look at the positive Covid-19 categories, for Dataset 1 and Dataset 2 there were 9 279 mild cases of covid-19 this accounted for 15.28% of the tested individuals in the datasets. There were 4 115 moderate cases of Covid-19 present in the datasets this was equivalent to 6.78%. Finally there were 115 severe Covid-19 cases in the datasets and this equated to 0.19% of the datasets. Looking at Dataset 3 there were 3 955 mild cases which accounted for 22.27% of the tested individuals in the dataset. There were 1 772 moderate cases which was equivalent to 9.98% of the dataset and there were 65 severe Covid-19 cases in the dataset and this equated to 0.37%. Finally Dataset 4 consisted of 7 782 mild cases which was equal to 17.44% of the dataset, 3 588 moderate cases which accounted for 8.04% of the dataset and lastly there were 105 severe cases and this equalled to 0.24% of the dataset.

The characteristics of the four datasets in terms of the response variable were fairly similar with Dataset 1 and Dataset 2 having the same response variable characteristics. Datasets three and four differed in terms of the response variable characteristics and this was a direct result of the creation process, explained in Section 4.2. Figure 4.2 visually confirms that the response variable of the datasets are imbalanced.

The datasets also contained information about two other diseases namely HIV and TB. The question that arose by having these two other diseases present in the dataset was, is there a relationship between Covid-19 infection status and the status of the two diseases?

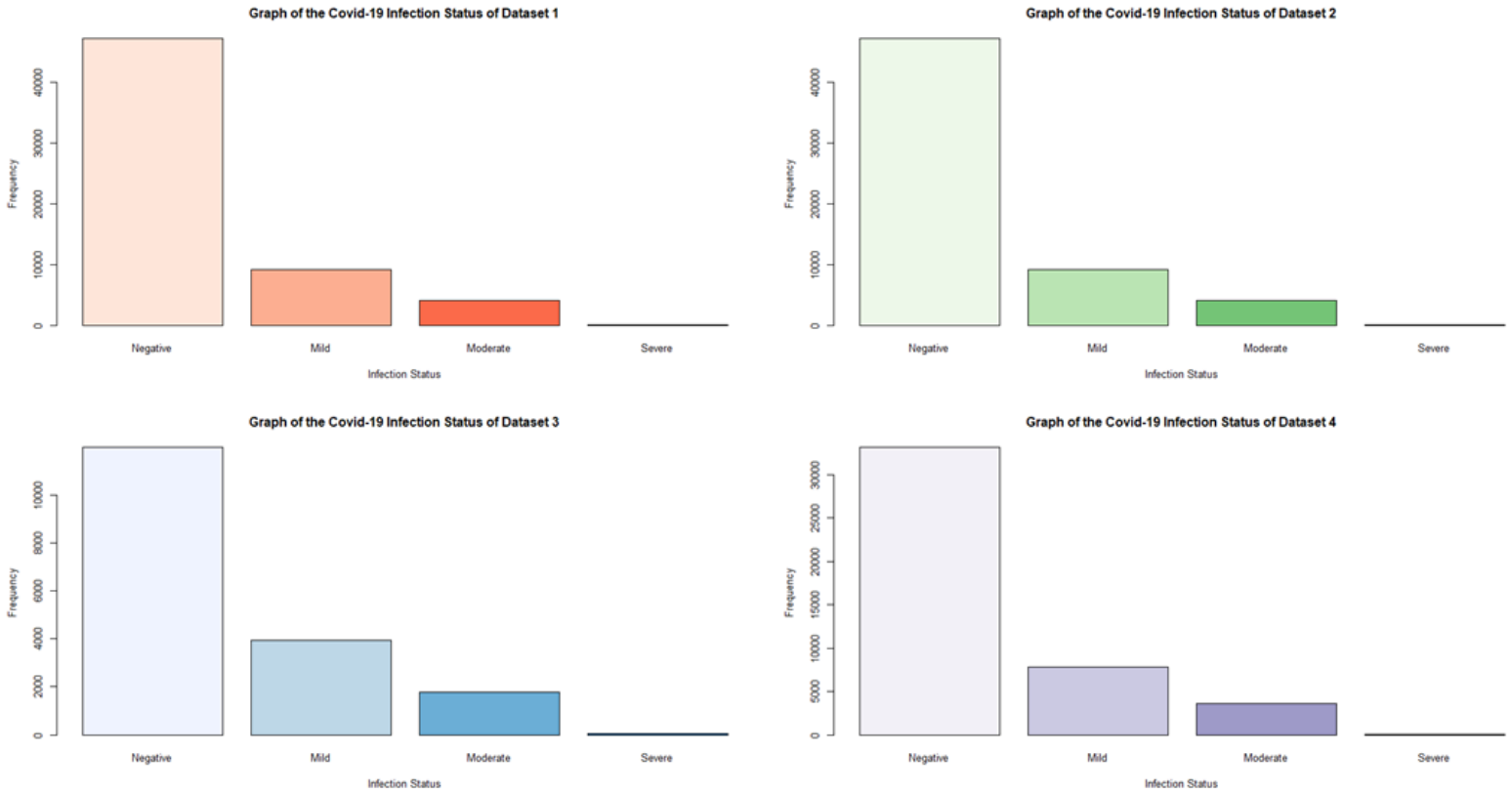

Figure 4.2: Bar Graph of the Covid-19 Infection Status for the four datasets

Figure 4.3 and Figure 4.4 show the distribution of the Covid-19 infection status of an individual with TB and HIV respectively. As expected majority of the individuals tested negative and this was expected because the dataset is imbalanced and had more negative results, however, there were a few interesting observations to be made. The total number of TB and HIV patients for Dataset 1 and Dataset 2 were 3 434 and 14 279 respectively. Whilst for Dataset 3 the total number of TB and HIV patients were 1 141 and 4 773. respectively. Finally for Dataset 4 there were 2 322 TB patients and 9 067 HIV patients. These figures show the datasets contained more individuals who have HIV than individuals who have TB.

Figures 4.3, 4.4 and Tables 4.2, 4.3 indicate that there seems to be no link between being positive for TB or HIV the severity of Covid-19.

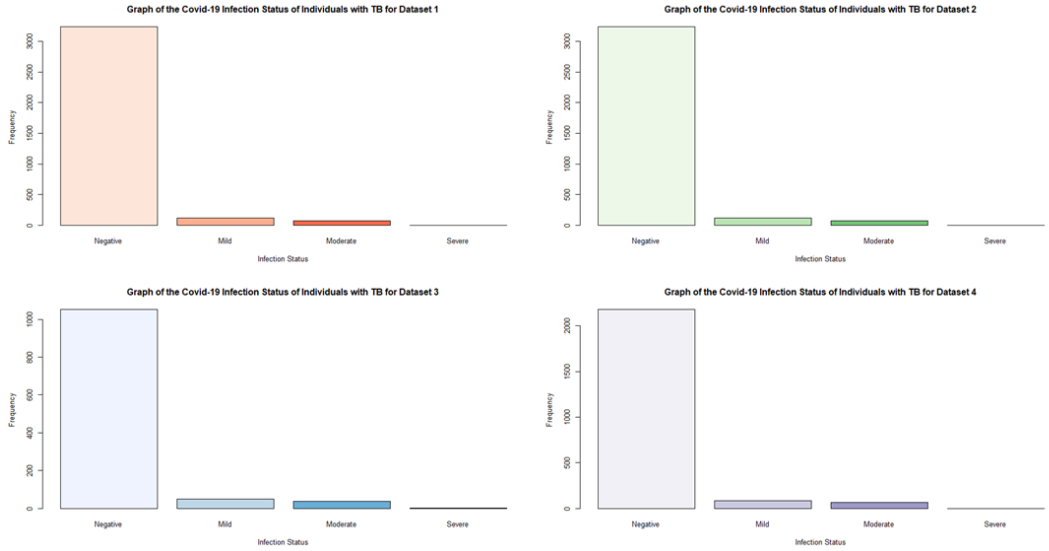

Figure 4.3: Plot of the Covid-19 infection status of individuals who are positive for TB for the four datasets

Table 4.2: Covid-19 infection status of individuals who are positive for TB

		Covid-19 Infection Status			
Dataset	**TB Cases**	Negative	Mild	Moderate	Severe
Dataset 1	3 434	3 239 (94.32%)	117 (3.41%)	77 (2.24%)	1 (0.03%)
Dataset 2	3 434	3 239 (94.32%)	117 (3.41%)	77 (2.24%)	1 (0.03%)
Dataset 3	1 141	1 051 (92.11%)	50 (4.38%)	39 (3.42%)	1 (0.09%)
Dataset 4	2 322	2 177 (93.76%)	81 (3.49%)	63 (2.71%)	1 (0.04%)

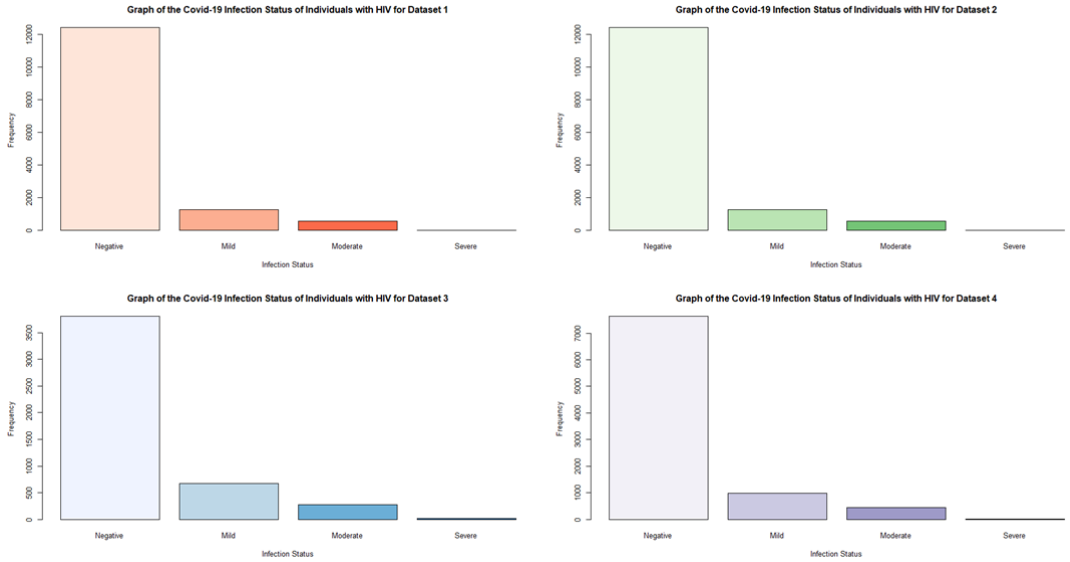

Figure 4.4: Plot of the Covid-19 infection status of individuals who are positive for HIV for the four datasets

Table 4.3: Covid-19 infection status of individuals who are positive for HIV

		Covid-19 Infection Status			
Dataset	**HIV Cases**	Negative	Mild	Moderate	Severe
Dataset 1	14 279	12 432	1 274	553	20
Dataset 2	14 279	12 432	1 274	553	20
Dataset 3	4 773	3 806	672	281	14
Dataset 4	9 067	7 630	973	445	19

Figure 4.5 shows how the age variable was distributed for the various datasets. The youngest individual in all of the datasets was 18 years old, however the oldest individual was not the same for all the datasets. In the first two datasets the oldest individual was 119 years old. The oldest individual in the second two datasets was 115 years old. The box-plots revealed that the age variable was positively skewed and that there

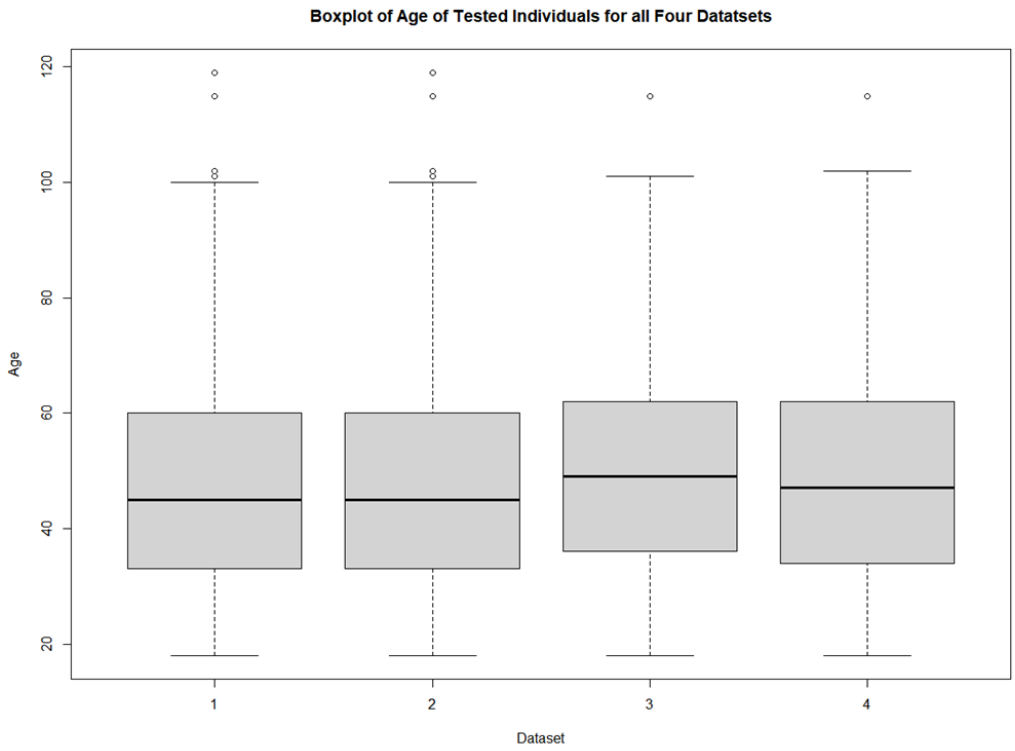

Figure 4.5: Boxplots of the age variable for the four different datasets.

were outliers present in the data, these were individuals older than 100. 50% of the individuals in datasets 1, 2, 3 and 4 were between the ages of 33, 33, 36 and 34 (1^{st} Quartile) and 60, 60, 62 and 62 respectively (3^{rd} Quartile). Figure 4.5 allows for a comparison between the four datasets to be easily made. The comparison showed that there were small differences between the box-plots but overall the characteristics of the age variable were almost identical.

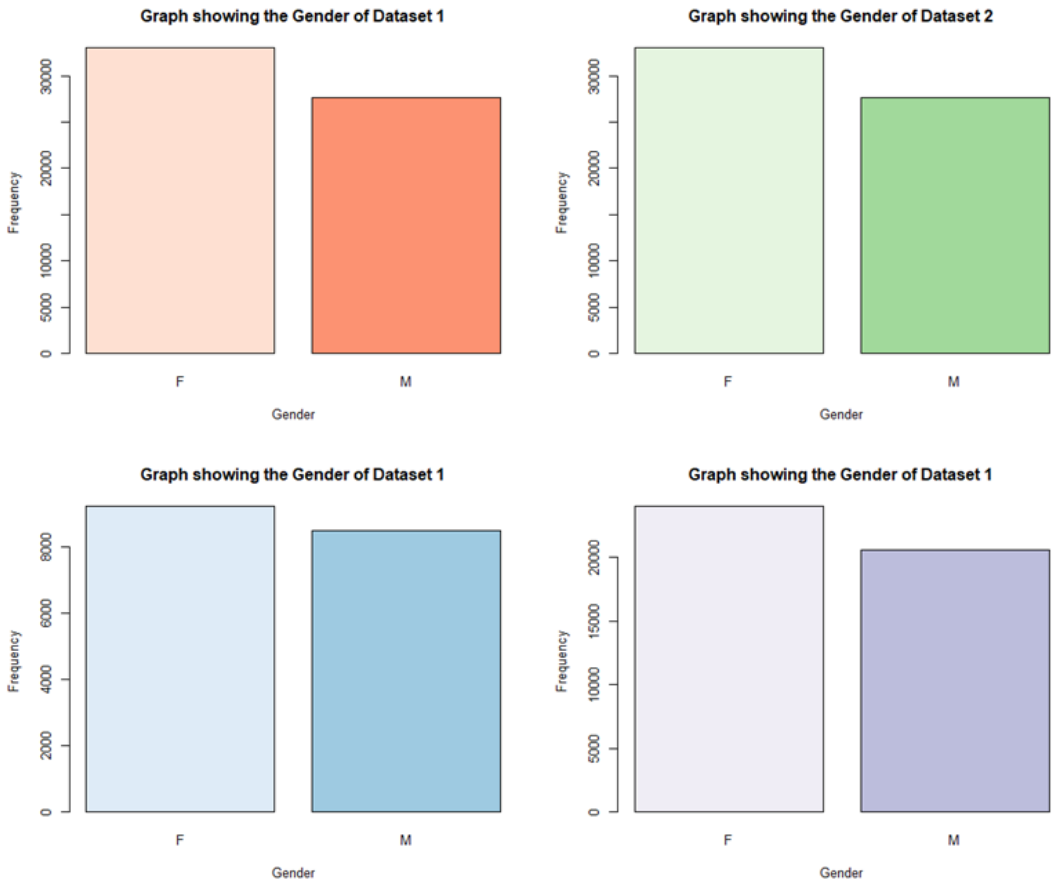

Figure 4.6: Graph of the Gender variable for the four datasets.

Looking into the gender variable of the four datasets. Figure 4.6 shows that there are more females than males resent in each of the datasets. Dataset 1 and Dataset 2 had 33 054 females and 27 657 males, Dataset 3 had 9 246 females and 8 516 males and lastly Dataset 4 had 24 030 females and 20 584 males.

The distribution of the remaining variables were investigated and three of those variables are presented below (Figure 4.7, 4.8, 4.9). The variables presented below are Red Cell Count, Haemoglobin and CRP.

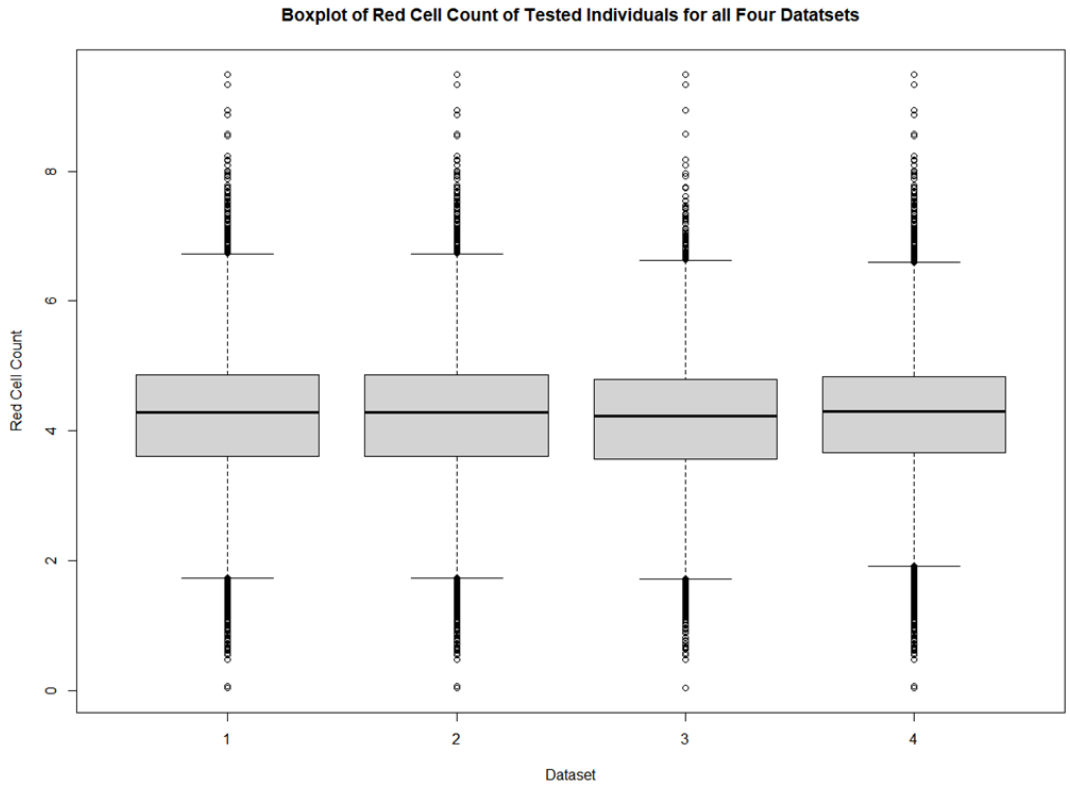

Figure 4.7: Boxplot of the Red Cell Count Variable for the four datasets.

The Red Cell Count variable displayed in Figure 4.7 had very similar characteristics for all four datasets, with the minimum values for each being 0.04 and a maximum of 9.49 for each dataset. There were slight differences in the first quartile, mean, median and third quartile values but these differences were very small. Which meant that the characteristics for the red cell count variable for all datasets were almost identical. The Red Cell Count mean for all the datasets ranged from 4.154 to 4.235. The Box-plots suggest that the variable was not skewed but does show there were a number of outliers present.

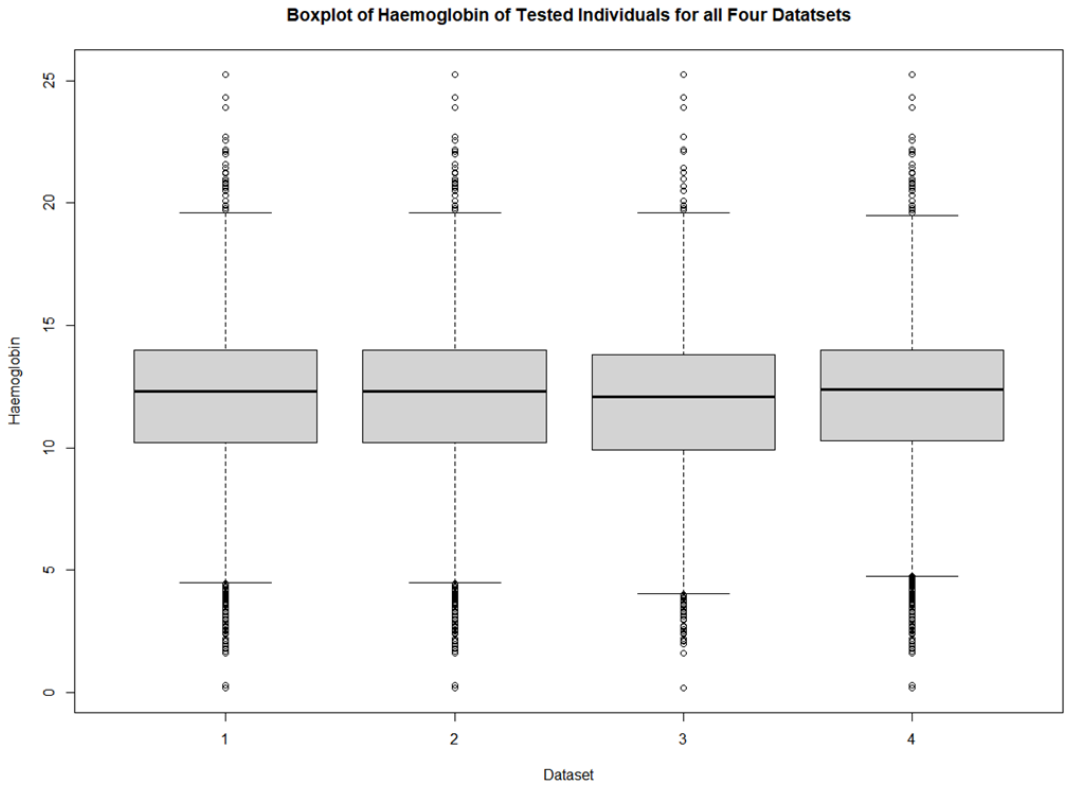

Figure 4.8: Boxplot of the Haemoglobin Variable for the four datasets.

Figure 4.8 shows the distribution of the Haemoglobin variable. The box-plots indicate that the Haemoglobin variable was not skewed and was actually symmetrical for all of the datasets, however the datasets did have a large number of outliers according to the box-plots. The outliers had a value either less than 5 or greater than 20. Among the four datasets the minimum and maximum values were the same and these values were 0.2 and 25.25 respectively. The four plots indicated that there were slight difference between the datasets but no major differences were present.

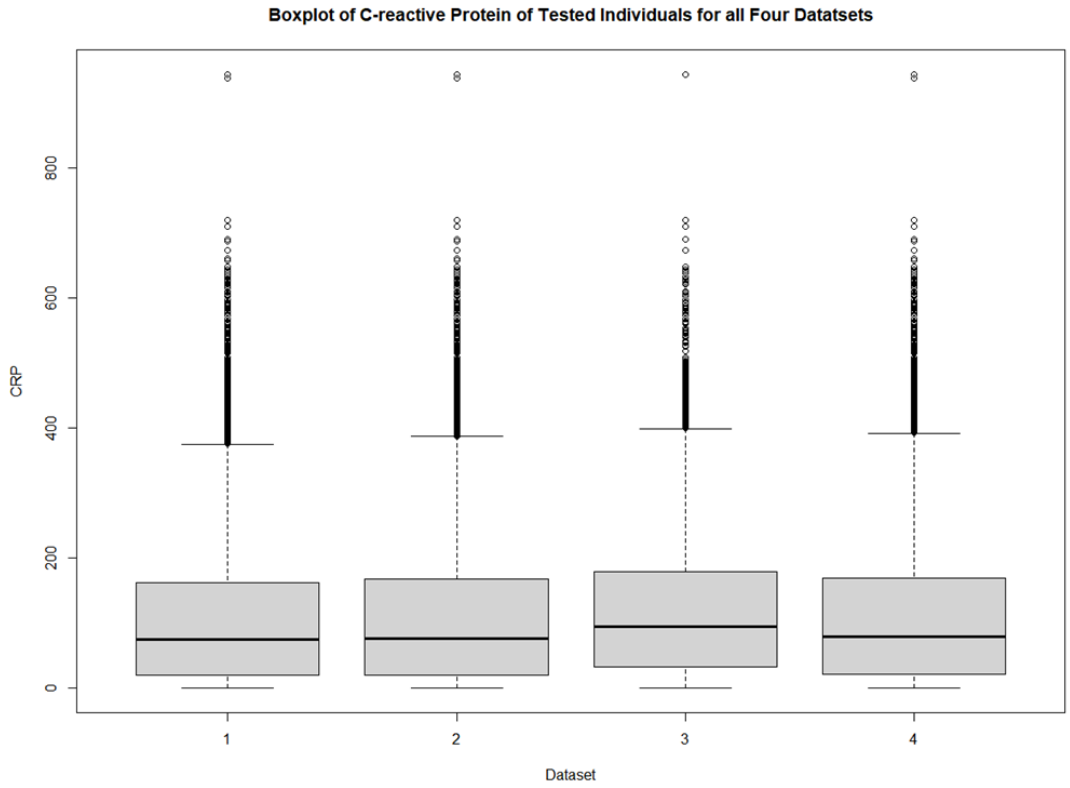

Figure 4.9: Boxplot of the CRP Variable for the four datasets.

The box-plots in Figure 4.9 are skewed to the right and have a large number of outliers. The outliers have a large magnitude. The box-plots show that the outliers were values of 400 or greater for all four datasets. Comparing the different datasets the box-plots indicate that the means were different for each dataset. The means for the dataset were as follows 106.3, 109, 120.1 and 110.9 for dataset 1, 2, 3 and 4 respectively. The respective IQRs were 142 (162 - 20), 147 (167 - 20), 147 (179 - 32) and 148.5 (169.5 - 21) for dataset 1, 2, 3 and 4. In each of the four plots there was a maximum value that was greater than all the other values. This individual was present in all four datasets and had a value of 944.

The fact that the variables above contain a large number of outliers is not a negative, in fact it is desirable as it is these outlier values compared to the “normal" values that will allow the various regression and machine learning methods to predict the different

levels of the Covid-19 category response variable.

4.4.2 Confounding Variable Analysis

In order to complete the confounding variable analysis each of the datasets had to first undergo one-hot encoding this was done in order to convert all categorical variables into binary variables. The one-hot encoding was carried out using the one-hot package in R (Gorman, 2018). One-hot encoding involves taking all the factors in a dataset and turning them into numeric binary variables. This is done by taking a factor with m-levels and creating m new binary variables where the m^{th} variable has a 1 if the factor indicated the m-level or 0 if it did not.

Once the four datasets had undergone one-hot encoding the datasets had the following characteristics, Dataset 1 had 60 711 observations on 38 variables, Dataset 2 had 60 711 observations on 14 variables, Dataset 3 had 17 762 observations on 38 variables and Dataset 4 had 44 614 observations on 14 variables.

To investigate if confounding variables are present in the respective datasets an iterative process was used whereby a variance inflation factor score for each variable in each dataset was obtained and the highest value above five was removed. A threshold value of five was used for this study and this was based on the Becker et al. (2015) study. This process was repeated until no more variables could be removed.

For Dataset 1, 2 and 4 only two variables needed to be removed. These variables were mean cell haemoglobin and haemoglobin. Dataset 3 required three variables to be removed namely, mean cell haemoglobin, haemoglobin and red cell count. Table 4.4 shows the VIF scores for each variable that was removed.

Table 4.4: VIF scores for removed variables for each dataset

	VIF Score			
Variable	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Mean Cell Haemoglobin	130.94	120.70	139.81	150.56
Haemoglobin	49.65	48.32	203.22	48.45
Red Cell Count	0	0	79.41	0

The next phase of the analysis involved removing the linearly dependent variables in each dataset. Three variables were removed for each dataset namely, HIV negative status variable, TB negative status variable and male gender variable. These variables were removed because they were fully explained by a zero of another binary variable. HIV negative status is explained by the HIV positive status variable being equal to one and the same is true for the TB negative status variable and the male gender variable.

4.4.3 Principal Component Analysis

Before the principal component analysis could be performed the response variable needed to be removed from the each of the datasets, this was required in order to determine which components explain the various patterns and trends in each dataset.

For each of the four datasets, once the response variable was removed a PCA function in R, *prcomp* (R Core Team, 2020), was used to carry out the principal component analysis. After which the variance for each dataset was computed and was used to create a scree plot as well as a cumulative proportion of variance plot. The results and plots are shown below.

4.4.3.1 PCA Dataset 1

The PCA function in R-studio returned the importance of 38 components, where 98.9% of the variance was explained by the first three components. The total variance was

fully explained by the first 18 principal components.

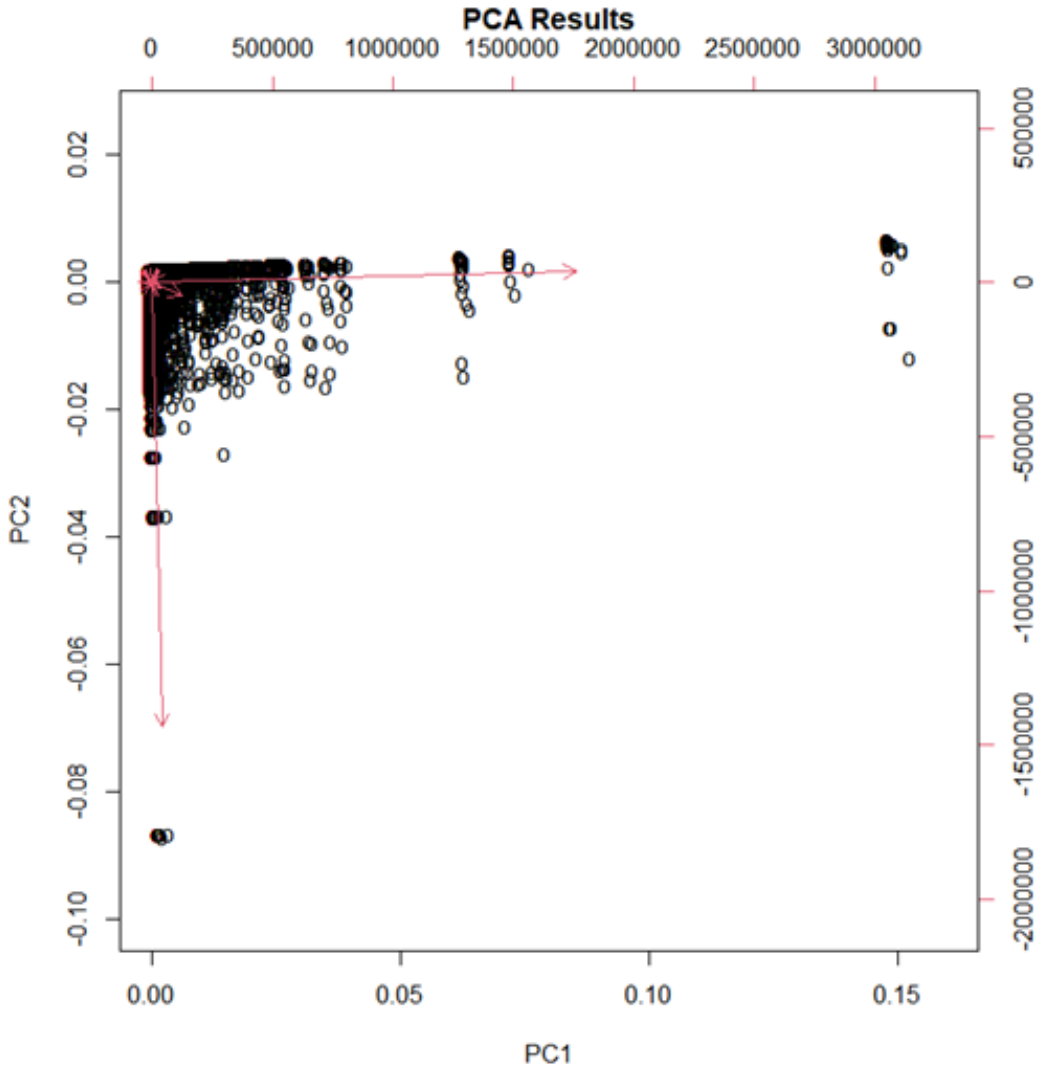

Figure 4.10: Principal Component Analysis Plot for Dataset 1

Figure 4.10 shows the points are clustered around the origin of all the axes, the axes represent the variables of the dataset. The plot shows two main axes and all the points in the plot are within the general bound of these two main axes. The plot shows the other axes/variables are not as important as the main two and this is shown by the size/-

magnitude of the other axes.

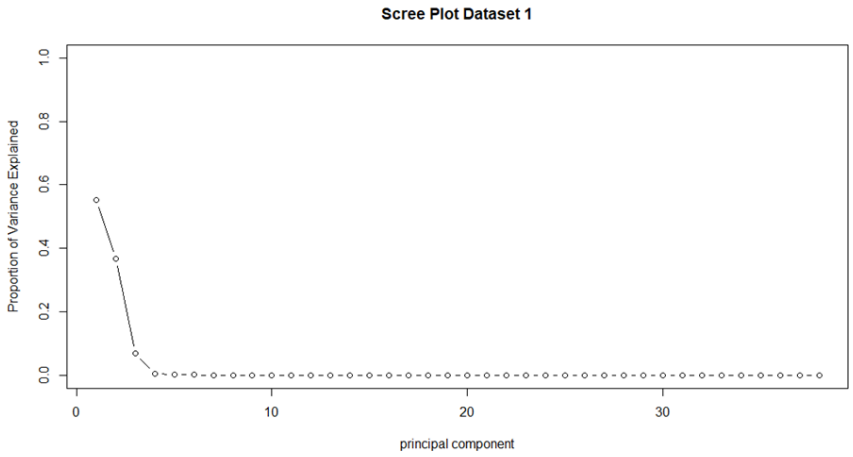

Figure 4.11: Scree Plot for Dataset 1 Showing the Proportion of Variance Explained by the Principal Components

Figure 4.11 is a Scree plot for Dataset 1 and it suggests that only 3 components are required to explain most of the variability in the dataset. This is in-line with the initial findings of the PCA as the first three components explain 98.9% of the variance in the dataset.

4.4.3.2 PCA Dataset 2

The PCA function in R returned the importance of 13 components, where 99.46% of the variance was explained by the first four components. The total variance was fully explained by the first 10 principal components.

Figure 4.12 shows, like Figure 4.10, that majority of the points are clustered around the origin of all the axes. This plot shows that there are three main axes that the points follow/lie-on. Once again the points seem to be somewhat bound by these main axes. The plot shows the other axes/variables are not as important as the main three and this is shown by the size/magnitude of the other axes.

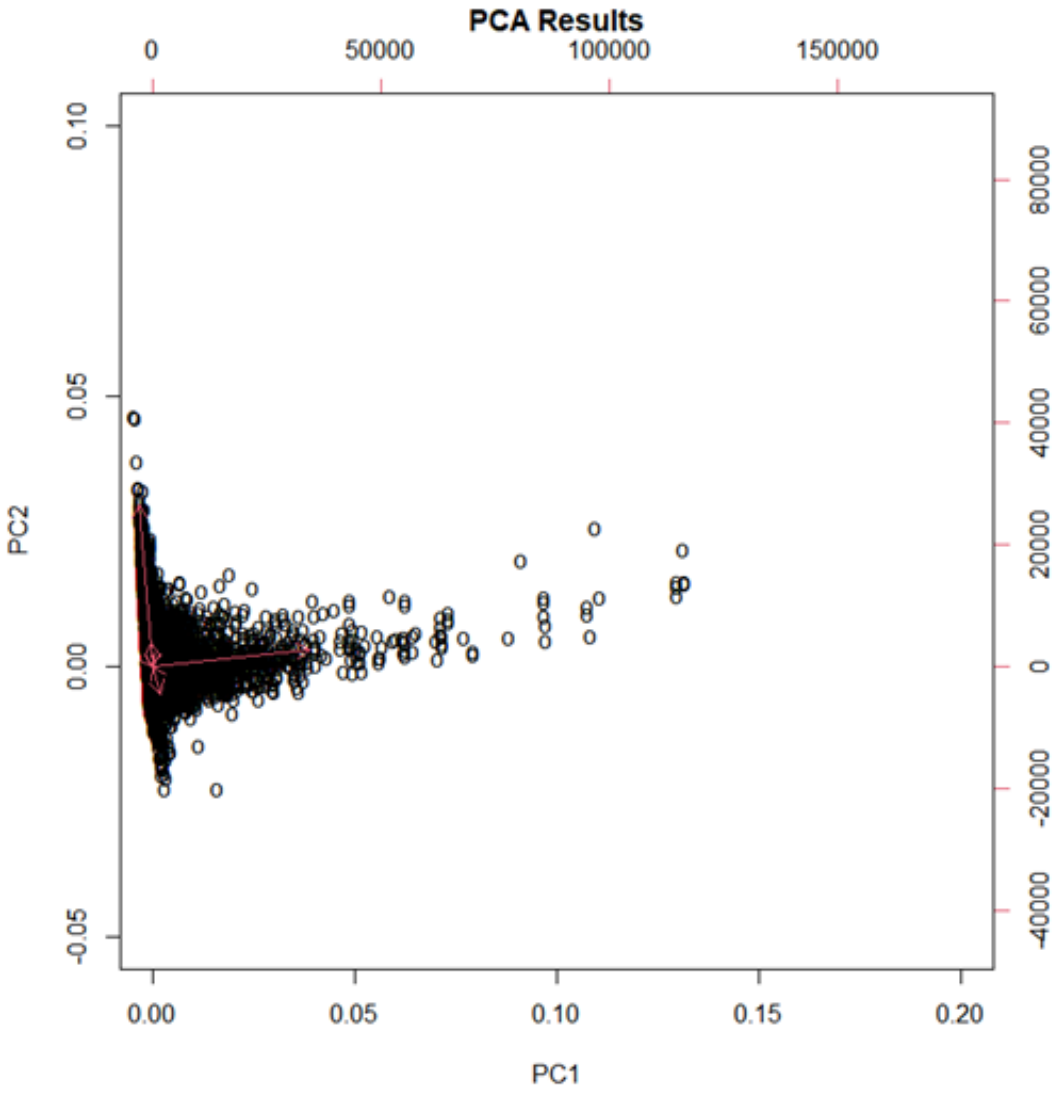

Figure 4.12: Principal Component Analysis Plot for Dataset 2

Figure 4.13 is a Scree plot for Dataset 2 and it suggest four components are required to explain the variance of the dataset. This is because the first four components explain 99.46% of the variance in the dataset.

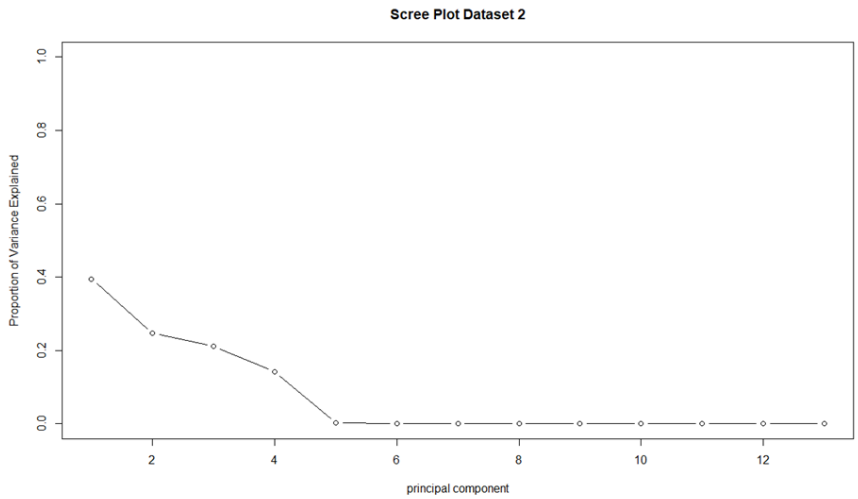

Figure 4.13: Scree Plot for Dataset 2 Showing the Proportion of Variance Explained by the Principal Components

4.4.3.3 PCA Dataset 3

The PCA function in R returned the importance of 38 components, where 98.70% of the variance was explained by the first three components. The total variance was fully explained by the first 18 principal components.

Figure 4.14 is almost identical to Figure 4.10. It once again shows that majority of the points are clustered around the origin of all the axes and that there are two main axes that the points follow/lie-on. Once again the points seem to be somewhat bound by these main axes. The plot shows the other axes/variables are not as important as the main three and this is shown by the size/magnitude of the other axes.

Figure 4.15 is a Scree plot for Dataset 3 and it suggests, like with Figure 4.11, that only 3 components are required to explain the variance of the dataset. This is expected as the first three components explain 98.7% of the variance in the dataset.

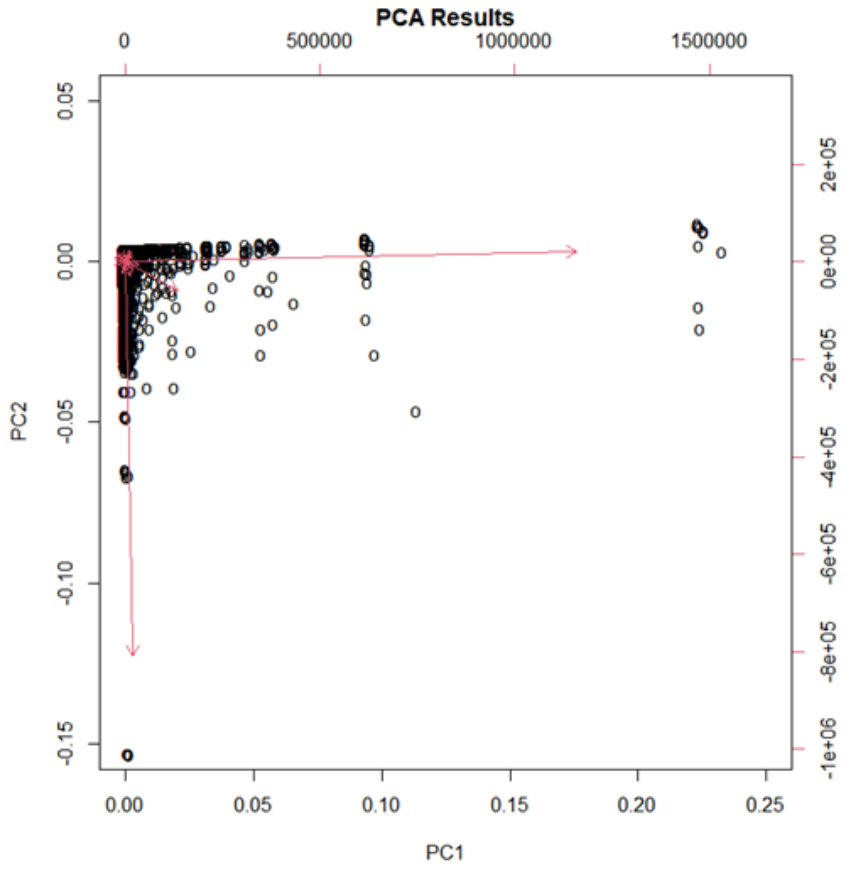

Figure 4.14: Principal Component Analysis Plot for Dataset 3

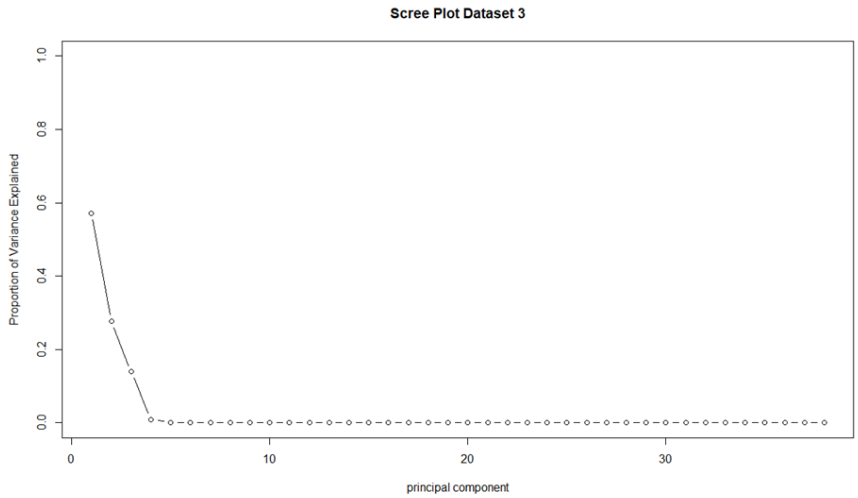

Figure 4.15: Scree Plot for Dataset 3 Showing the Proportion of Variance Explained by the Principal Components

4.4.3.4 PCA Dataset 4

The PCA function in R returned the importance of 13 components, where 99.45% of the variance was explained by the first four components. The total variance was fully explained by the first 10 principal components.

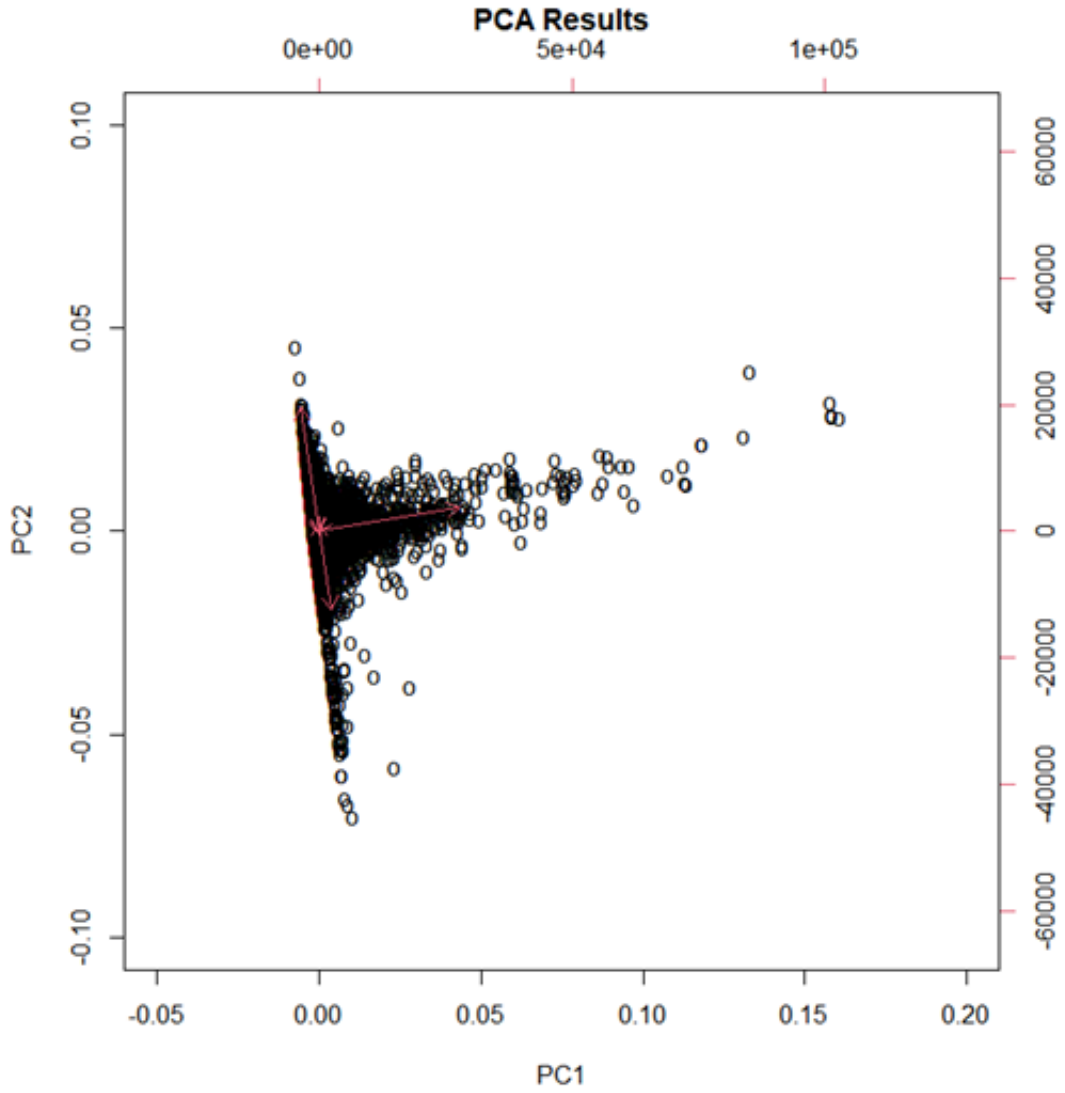

Figure 4.16: Principal Component Analysis Plot for Dataset 4

Figure 4.16 is almost identical to Figure 4.12 and shows that majority of the points are clustered around the origin or along the three main axes. The plot shows the other axes/variables are not as important as the main three and this is shown by the size/-magnitude of the other axes.

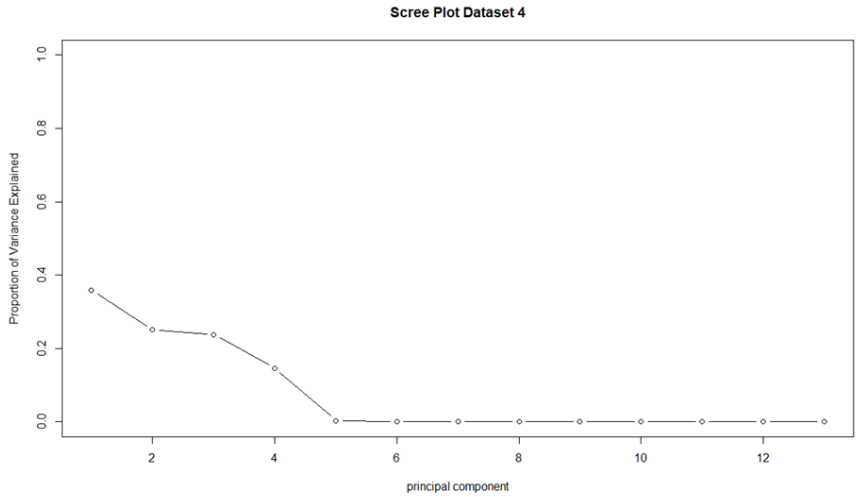

Figure 4.17: Scree Plot for Dataset 4 Showing the Proportion of Variance Explained by the Principal Components

Figure 4.17 is a Scree plot for Dataset 4. It has a strange shape with no real change between the proportion of variance explained by component two and three. Taking that into account the Scree plot still suggests four components are required to explain the variance of the dataset. This makes sense as 98.9% of the variance in the dataset is explained by the first four components.

4.4.4 Factor Analysis

Factor analysis was carried out using the “psych" package in R (Revelle, 2022). The process began by obtaining the Kaiser, Meyer, Olkin (KMO) Measure of Sampling Adequacy and conducting Bartlett’s sphericity Test for each dataset. This was done using the *KMO* and *cortest.bartlett* functions in the psych package. The KMO test and Bartlett’s sphericity test are used to determine if factor analysis can be conducted on a given dataset and produce desirables outcomes. There are varying opinions on what proportion, for the KMO test, is considered the acceptable level for factor analysis to produce desirable outcomes (Dziuban and Shirkey, 1974; Hill, 2011). Values used range from 0.5 to 0.8, in this study the guidelines set out by Shrestha (2021) were used and values were interpreted in accordance to these guidelines.

To determine the number of factors for each dataset the *VSS* (Very Simple Structure) function in the “psych" package was used. The function determines how many factors should be used for four different criteria, namely VSS complexity one, VSS complexity two, MAP and BIC. The VSS method was chosen over the Kaiser-Guttman normalization rule (Kaiser, 1992) because it is a more modern approach (Auerswald and Moshagen, 2019).

Once the number of factors was determined factor analysis was carried out using the “fa" function in R (Revelle, 2022), with the “varimin" rotation and the maximum likelihood factor method. The “varimin" rotation was selected as it produced the best results when tested.

4.4.4.1 FA Dataset 1

A correlation calculation was conducted to see the various correlations between the explanatory variables. The following correlations were the most significant; 0.7555 between Haematocrit and Red Cell Count, 0.7106 between Alanine Transaminase and Aspartate Transaminase.

The KMO test was then conducted and produced a measure of sampling adequacy (MSA) score of 0.62, which according to Shrestha (2021) indicates that the sampling

was mediocre. Bartlett's test produced a p-value < 0.05 and therefore in conjunction with the KMO test meant that factor analysis could be conducted. The MSA scores for each variable are shown in the Appendix D.

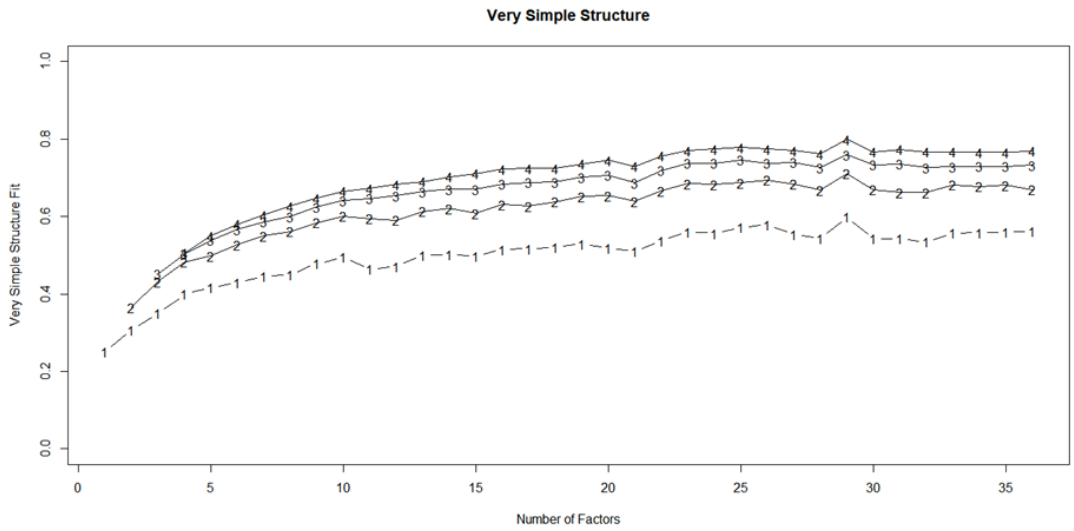

Figure 4.18: Factor Analysis Very Simple Structure Plot Showing the Number of Factors to be used for Dataset 1

Figure 4.18 indicates that the number of factors to use, based on the VSS function, is 29. Hence 29 factors were used when running the factor analysis function in R-studio.

Figure 4.19 indicates the groupings of the various explanatory variables for the first two loadings. Dataset 1's factor analysis was carried out using 29 factors as explained above, however Figure 4.19 shows only a few distinct groupings, namely in the top left, top right and bottom right corners of the figure. The top left grouping is the most distinct grouping and is made up of the red cell count, platelet count, albumin, fibrinogen and haematocrit variables which are all related to blood and hence are explained by a single factor. However majority of the points in the plot are closely clustered together which makes it difficult to recognise all the factors clearly. To further understand the factor analysis the sum of square loadings and variance of the factors are shown in

Table 4.5.

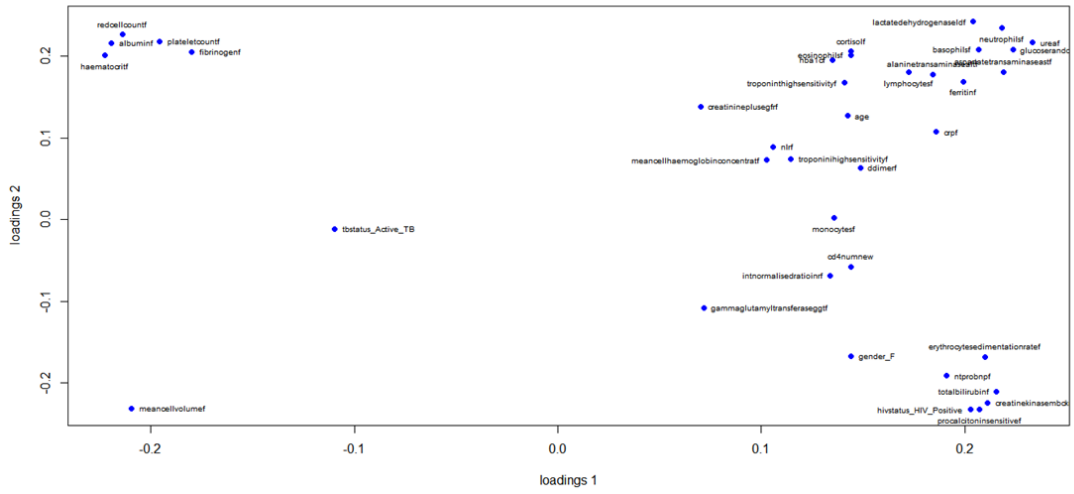

Figure 4.19: Factor Analysis Plot for Dataset 1

Table 4.5 is ordered by the sum of square loading values and shows that for Dataset 1 that the 29 factors explain 66.1% of the total variance present in the dataset. This is an acceptable level and indicates that the datasets contain complex variables and respective relationships that can be reduced to a number of factors.

Table 4.5: Sum of Square loadings and Variance of the factors for Dataset 1

Factor	SS Loadings	Proportion of Variance	Cumulative Variance
27	1.192	0.031	0.031
1	1.185	0.031	0.063
2	1.141	0.030	0.093
8	1.121	0.029	0.122
7	1.100	0.029	0.151
4	1.077	0.028	0.179
6	1.024	0.027	0.206
5	1.017	0.027	0.233
25	1.004	0.026	0.259
10	0.989	0.026	0.286
3	0.960	0.025	0.311
9	0.891	0.023	0.334
29	0.886	0.023	0.358
21	0.833	0.022	0.379
18	0.805	0.021	0.401
14	0.787	0.021	0.421
22	0.765	0.020	0.442
17	0.765	0.020	0.462
28	0.751	0.020	0.481
19	0.742	0.020	0.501
12	0.728	0.019	0.520
24	0.722	0.019	0.539
15	0.721	0.019	0.558
16	0.703	0.018	0.577
26	0.689	0.018	0.595
23	0.685	0.018	0.613
11	0.673	0.018	0.630
20	0.614	0.016	0.647
13	0.529	0.014	0.661

4.4.4.2 FA Dataset 2

The Correlation calculation for dataset only produced one noteworthy correlation; 0.7863 between Haematocrit and Red Cell Count.

The KMO test for Dataset 2 produced a MSA score of 0.44 which indicates the factor analysis will not produce desirable outcomes (Shrestha, 2021). A detailed look into the individual variable MSA scores (see Appendix D) shows that the MSA is affected by the mean cell volume variable as it has an individual MSA score of 0.16. Bartlett's test produced a p-value < 0.05 which indicates there are common factors. Hence factor analysis was conducted to see what factors could be discovered and what portions of the dataset could be explained.

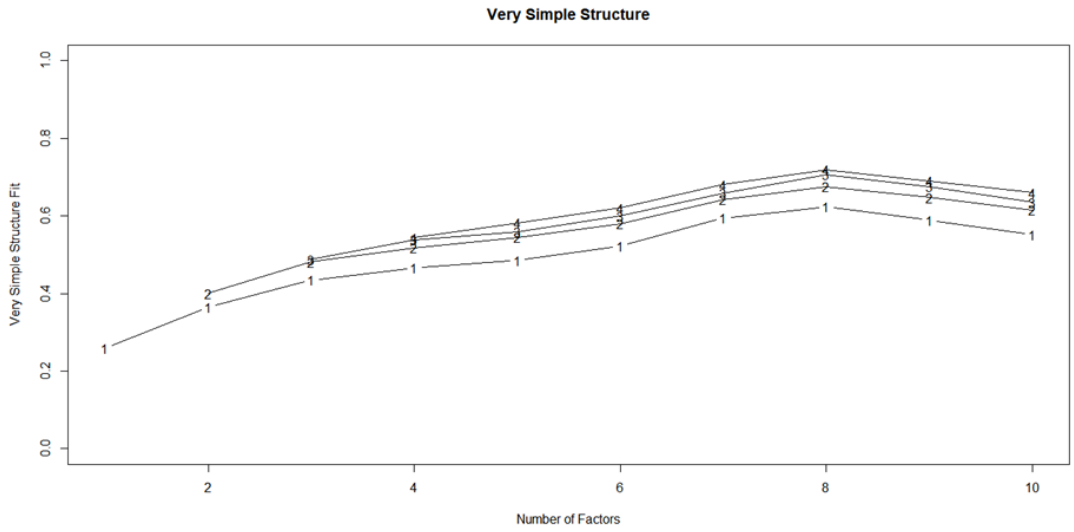

Figure 4.20: Factor Analysis Very Simple Structure Plot Showing the Number of Factors to be used for Dataset 2

The results of the VSS function in R revealed that eight factors were to be used when factor analysis was conducted for Dataset 2. Figure 4.20 confirms the result of eight factors.

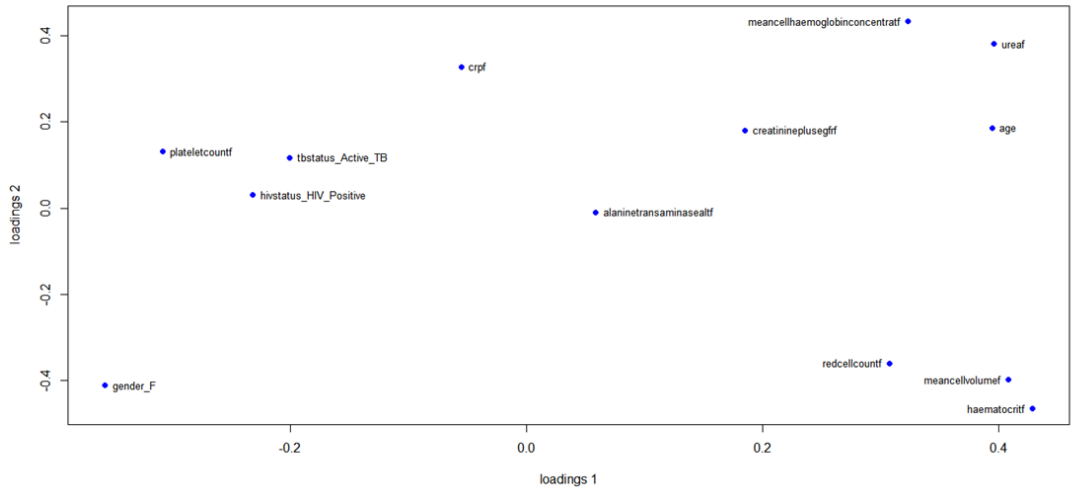

Figure 4.21: Factor Analysis Plot for Dataset 2

Figure 4.21 shows the plot of the first two loadings for the explanatory variables for Dataset 2. Dataset 2 had eight factors for factor analysis as suggested by 4.20. Figure 4.21 indicates groupings between some the variables. Some of the variables are isolated, but other variables appear to be grouped. There is a grouping on the bottom left corner of the plot that suggests the variables are explained by a single factor, the red cell count, haematocrit and mean cell count are linked to blood. Figure 4.21 shows it is easier to identify potential factors because there are fewer variables. To further understand the factor analysis the sum of square loadings and variance of the factors are shown in Table 4.6.

Table 4.6 shows that Dataset 2 was only able to produce average results. This was expected as Dataset 2 produced a MSA score of 0.44 which was below the minimum threshold of 0.5. Its eight factors explained 56.5% of the total variance present in the dataset. This result indicates that the dataset has complex variables and relationships between variables that are not able to be explained by a single factor.

Table 4.6: Sum of Square loadings and Variance of the factors for Dataset 2

Factor	SS Loadings	Proportion of Variance	Cumulative Variance
4	1.220	0.094	0.094
2	1.217	0.094	0.187
6	1.005	0.077	0.265
1	0.918	0.071	0.335
3	0.823	0.063	0.399
7	0.771	0.059	0.458
5	0.739	0.057	0.515
8	0.655	0.050	0.565

4.4.4.3 FA Dataset 3

Dataset 3 produced one significant correlation calculation result. The significant correlation was 0.8071 between Alanine Transaminase and Aspartate Transaminase.

The KMO test produced a MSA score of 0.62 and Bartlett's test produced a p-value of < 0.05, meaning that factor analysis can be conducted on Dataset 3.

Figure 4.22 indicates that there were 28 factors to be used for Dataset 3. This is seen by the value for all four lines reaching the maximum value at factor 28.

Figure 4.23 shows the plot of the explanatory variables for the first two loadings. Dataset 3's factor analysis was carried out using 28 factor. Figure 4.23 shows three large groupings in the top left, top right and bottom left corners of the plot. A large number of the variables are grouped in the for mentioned corners but there are still many variables isolated away from these corners. To further understand the factor analysis the sum of square loadings and variance of the factors are shown in Table 4.7.

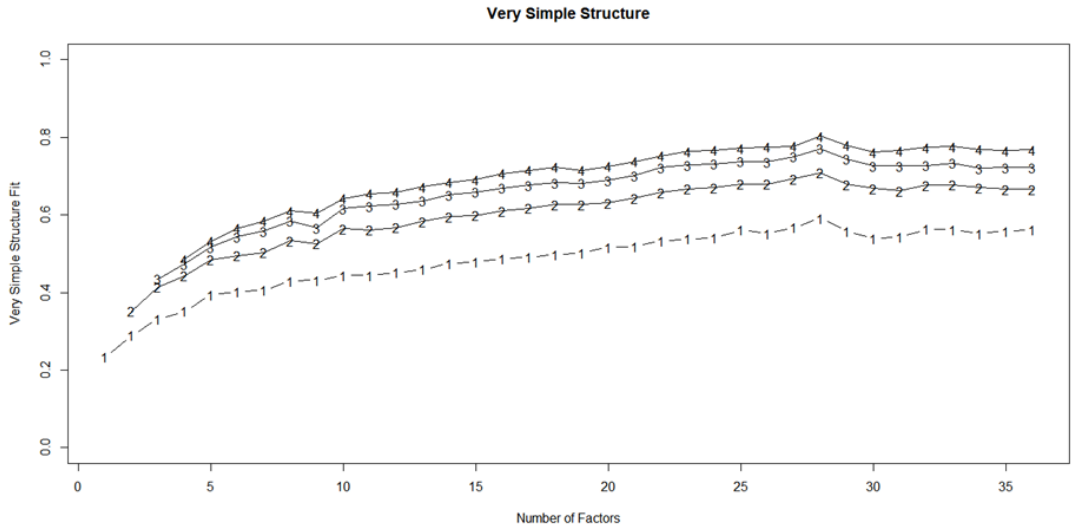

Figure 4.22: Factor Analysis Very Simple Structure Plot Showing the Number of Factors to be used for Dataset 3

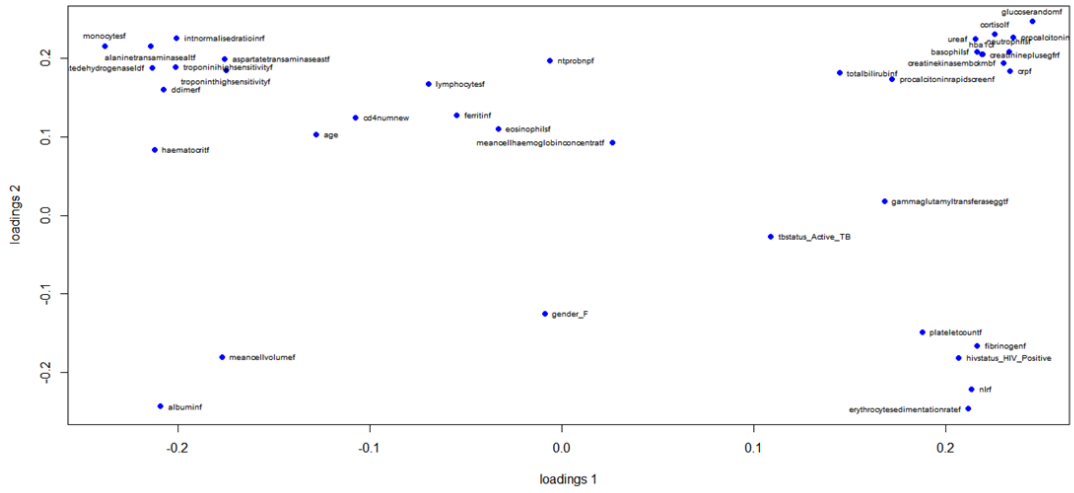

Figure 4.23: Factor Analysis Plot for Dataset 3

Table 4.7 shows that for Dataset 3 the 28 factors are able to explained 69.4% of the total variance present in the dataset, this is the most of all the datasets. The performance of the factor analysis is satisfactory and indicates that the dataset can be reduced and still

explain most of the variance contained in the dataset.

Table 4.7: Sum of Square loadings and Variance of the factors for Dataset 3

Factor	SS Loadings	Proportion of Variance	Cumulative Variance
2	1.316	0.035	0.035
1	1.277	0.034	0.068
3	1.259	0.033	0.101
15	1.063	0.028	0.129
7	1.062	0.028	0.157
5	1.048	0.028	0.185
9	1.032	0.027	0.212
20	1.012	0.027	0.239
13	0.999	0.026	0.265
11	0.982	0.026	0.291
27	0.961	0.025	0.316
8	0.952	0.025	0.341
14	0.922	0.024	0.365
10	0.905	0.024	0.389
25	0.903	0.024	0.413
19	0.898	0.024	0.437
6	0.896	0.024	0.460
28	0.876	0.023	0.483
24	0.873	0.023	0.506
4	0.867	0.023	0.529
26	0.864	0.023	0.552
16	0.842	0.022	0.574
22	0.837	0.022	0.596
12	0.836	0.022	0.618
21	0.783	0.021	0.639
23	0.717	0.019	0.657
17	0.707	0.019	0.676
18	0.673	0.018	0.694

4.4.4.4 FA Dataset 4

The Correlation calculation for Dataset 4, like Dataset 2, only produced one noteworthy correlation 0.7486 between Haematocrit and Red Cell Count.

The KMO test for Dataset 4 produced a MSA score of 0.49 which like Dataset 2 indicates the factor analysis wont produce desirable outcomes, however like Dataset 2 Bartlett's test produced a p-value of < 0.05 which indicated common factors were contained in the dataset. A detailed look into the individual variable MSA scores showed that of the 13 variables only three had MSA scores below 0.5 and they were Red Cell Count (0.46), Haematocrit (0.47), and Mean Cell Volume (0.19). The individual MSA scores show that the Mean Cell Volume variable produced the lowest score and was responsible for pulling the overall MSA score below the threshold. Hence, factor analysis was conducted to see what factors could be discovered and what portions of the dataset could be explained.

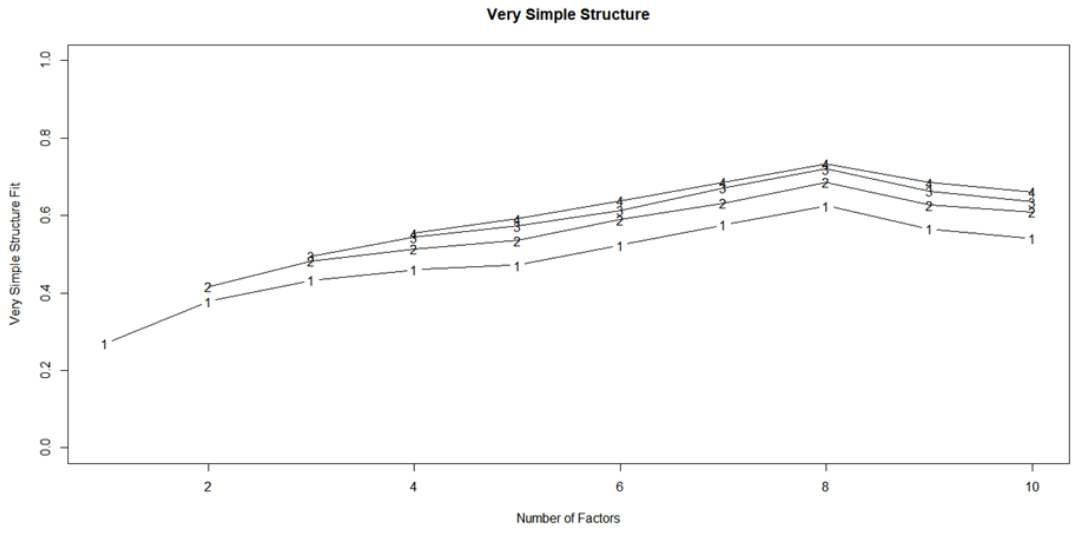

Figure 4.24: Factor Analysis Very Simple Structure Plot Showing the Number of Factors to be used for Dataset 4

Figure 4.24 indicates that the number of factors to use based using the VSS function in R was 8. Hence 8 factors were used when running the factor analysis function in R.

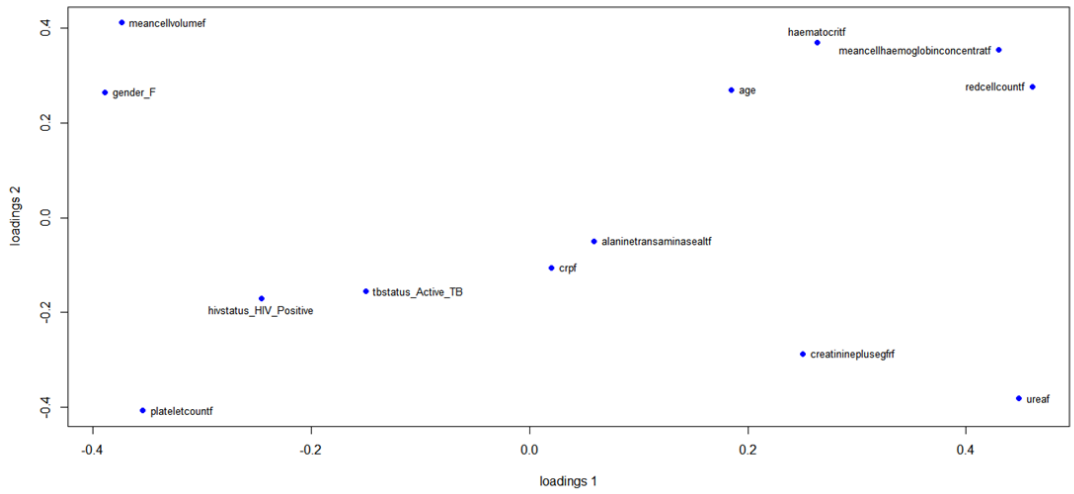

Figure 4.25: Factor Analysis Plot for Dataset 4

Figure 4.25 shows the plot of the first two loadings for the explanatory variables for Dataset 4. Dataset 4 had eight factors as suggested by 4.24. Figure 4.25 indicates that the variables are spread out and there are no real groupings between the variables. The closest groupings that can be seen are in the centre and the top right corner of the plot between CRP and ALT and red cell count and mean cell haemoglobin concentrate respectively. Figure 4.25 once again shows it is easier to identify potential factors when there are fewer variables. To further understand the factor analysis the sum of square loadings and variance of the factors are shown in Table 4.8.

Table 4.8: Sum of Square loadings and Variance of the factors for Dataset 4

Factor	SS Loadings	Proportion of Variance	Cumulative Variance
3	1.268	0.098	0.098
8	1.113	0.086	0.183
5	1.065	0.082	0.265
1	0.994	0.076	0.342
6	0.979	0.075	0.417
4	0.978	0.075	0.492
2	0.860	0.066	0.558
7	0.763	0.059	0.617

Table 4.8 shows that Dataset 4 performed reasonably well. It outperformed Dataset 2 which was the other dataset not to have obtained a MSA score above 0.5. The dataset had eight factors that explained 61.7% of the total variance present in the dataset. This is reasonable but as seen in Figure 4.25 shows the dataset can only be slightly reduced.

4.5 Variable Selection

The variable selection process was carried out using the processes outlined in the research outline, Section 3.2. The univariate significance test was used first, with the goal of the univariate test being to identify potential variables that may need to be excluded from the dataset because they have no significance in predicting the response variable. Secondly, a Boruta feature selection algorithm was used (Kursa et al., 2010). This is a multivariate test that looks at the significance of a variable in the presence of all the variables. Ultimately the final decision on what variables to keep in the datasets and which variables to remove was based on the final results of the Boruta feature selection algorithm because it looked at the importance of each variable in the presence of all the other variables.

4.5.1 Univariate Test

The method used to carry out the tests is the same method used in the Yang et al. (2020) study, whereby a univariate test was conducted on each variable to determine if it was significant in relation to the Covid Category response variable. An ANOVA test was conducted using the AOV (Analysis of Variance Model) R-studio package to determine which variables were significant. The ANOVA test results of the insignificant variables for each dataset are shown in Table 4.9. The significance level used in the test was 5%.

Table 4.9: ANOVA test results of the insignificant variables

Variable	Sum sq	Mean sq	F value	Pr(>F)	Significant	Dataset
Creatinine plus eGFR	58 460	19 486	1.174	0.318	No	1
Urea	399	133.15	1.564	0.196	No	1
Aspartate Transaminase (AST)	126 500	42 156	0.881	0.45	No	1
Ferritin	$6.759e^{+07}$	22 530 525	2.19	0.087	No	1
Lymphocytes	123	41.07	0.781	0.504	No	1
Basophils	0	0.1609	0.309	0.819	No	1
Creatinine plus eGFR	55 169	18 390	1.12	0.339	No	2
Urea	491	163.81	1.882	0.13	No	2
Lactate Dehydrogenase	$1.192e^{+06}$	397 383	1.014	0.385	No	3
Ferritin	$2.112e^{+08}$	70 407 074	2.302	0.075	No	3
Lymphocytes	345	114.93	1.784	0.148	No	3
Basophils	0	0.1526	0.415	0.742	No	3
Alanine Transaminase (ALT)	138 600	46 217	1.665	0.172	No	4

Table 4.9 shows that there are different insignificant variables for the different datasets. This is to be expected as the different datasets have different make-ups/structures and hence different variables. The table shows that Dataset 1 had the most insignificant variables and that Dataset 4 had the least amount of insignificant variables.

4.5.2 Boruta Feature Selection

The Boruta feature selection algorithm was run on all four of the datasets and bearing in mind the results of the univariate tests. The Boruta algorithm gives each variable in the dataset a result, either Important, unimportant or shadow. The results of the Boruta algorithm are shown in the figures below.

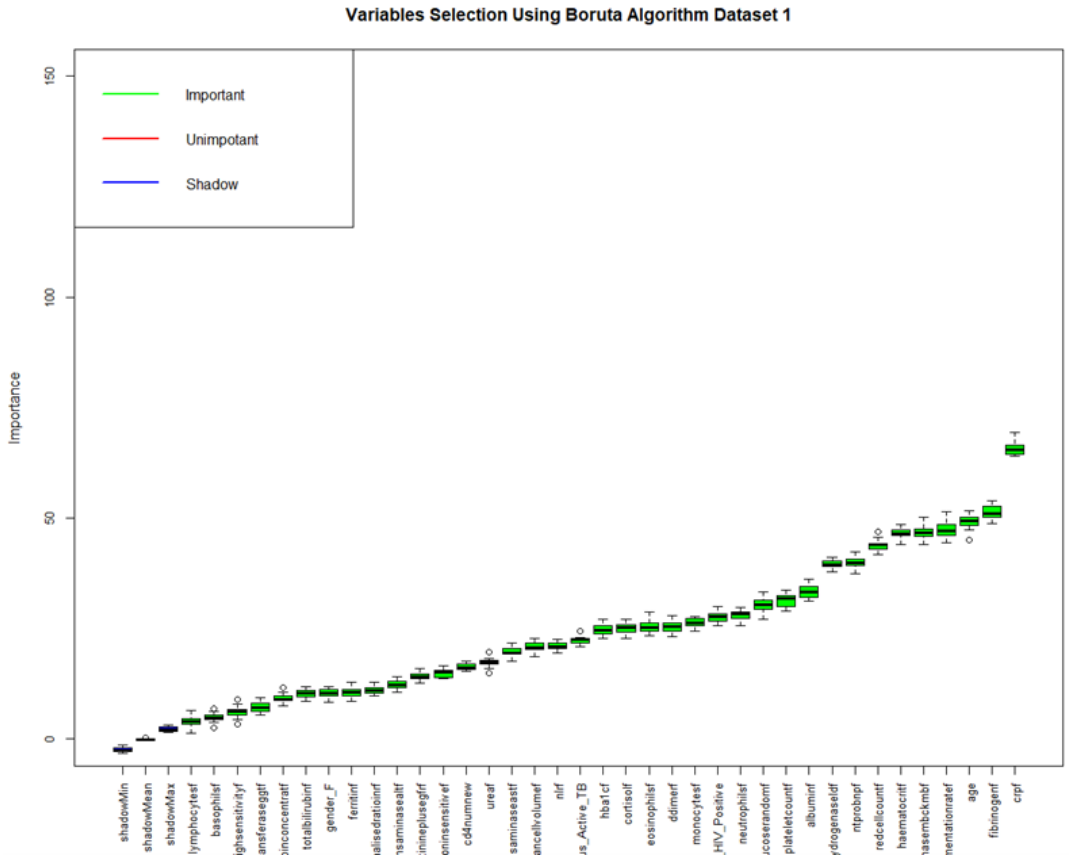

Figure 4.26: Variable Selection for Dataset 1 using the Boruta Feature Selection Algorithm.

Figure 4.26 shows that out of the 38 predictor variables the Boruta algorithm has determined that only one variable is unimportant. The unimportant variable is the High-sensitivity Troponin I variable. It is worthy to note that the univariate test found the High-sensitivity Troponin I variable to be significant and none of the variables found

to be insignificant in the univariate test were found to be unimportant by the Boruta algorithm. Figure 4.26 shows that the most important variable is the Creatinine plus eGFR variable.

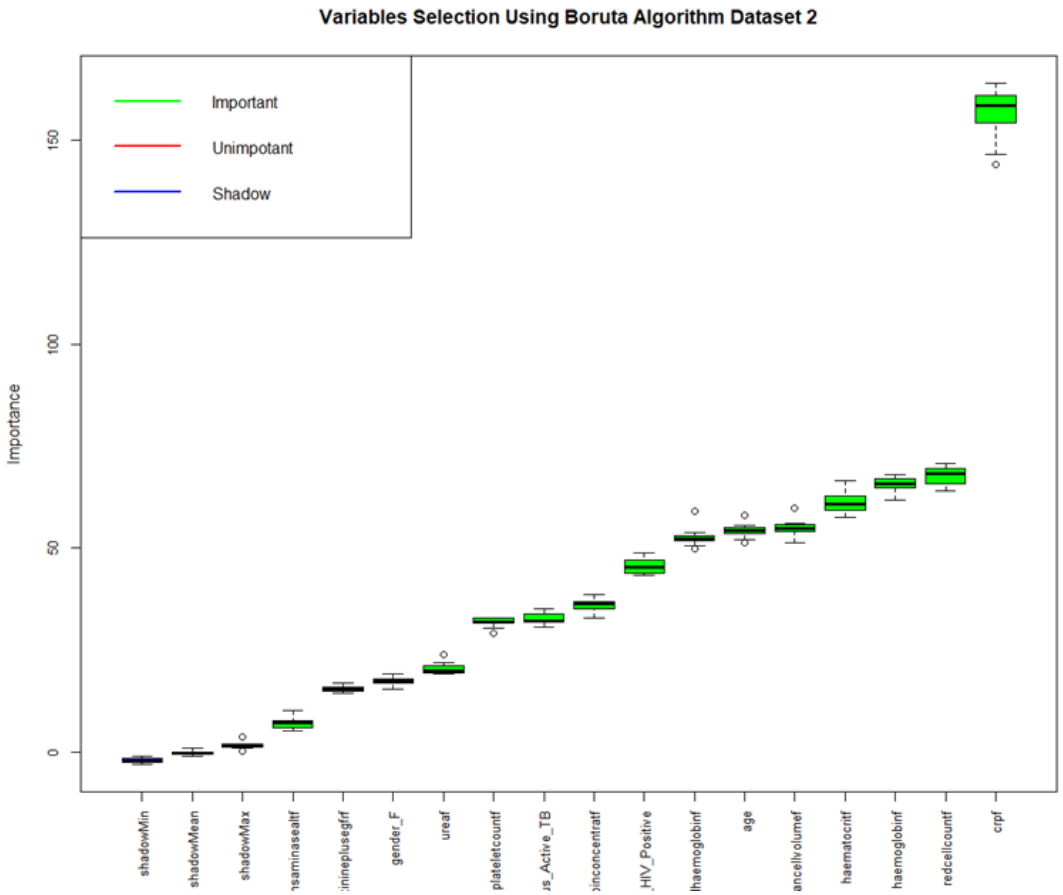

Figure 4.27: Variable Selection for Dataset 2 using the Boruta Feature Selection Algorithm.

Figure 4.27 shows that out of the 13 predictor variables the Boruta algorithm has determined that all the variables are important. This is once again a different result to the univariate test results. Figure 4.27 shows that the most important variable for dataset 2 is once again the Creatinine plus eGFR variable but in this case it has a greater importance value (approximately double the importance value in dataset 1).

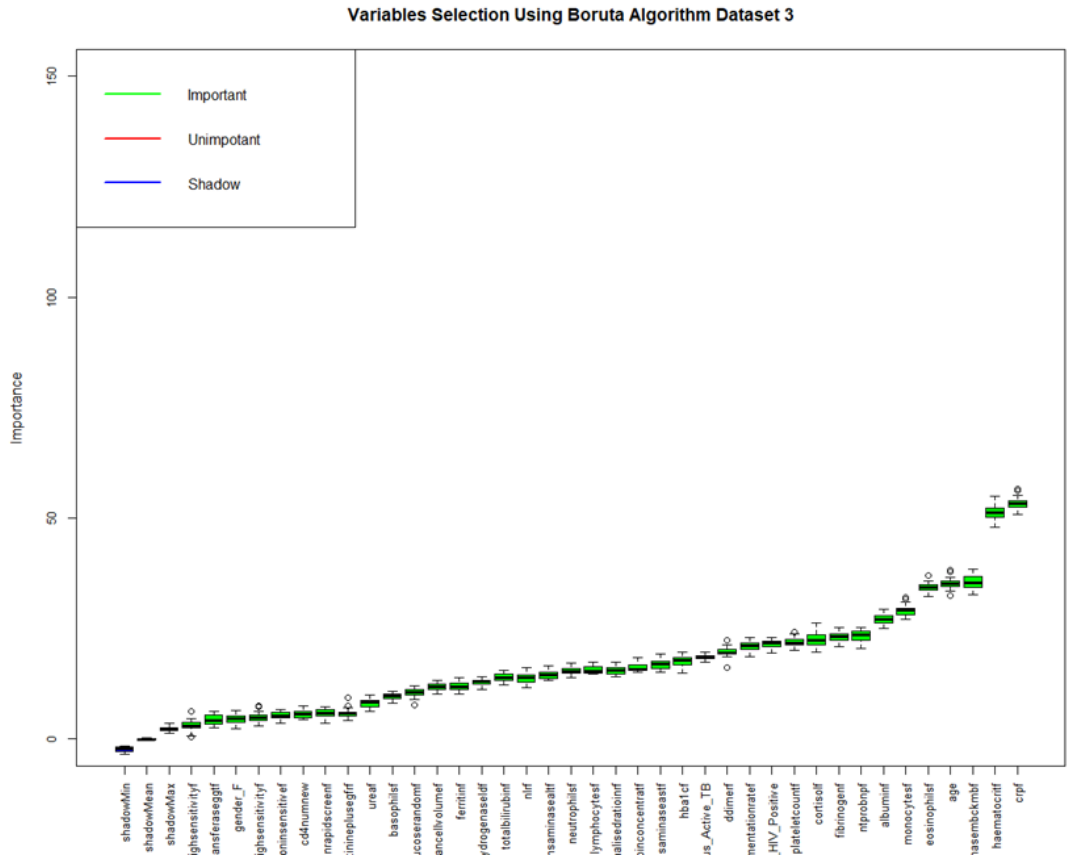

Figure 4.28: Variable Selection for Dataset 3 using the Boruta Feature Selection Algorithm.

Figure 4.28 shows that out of the 38 predictor variables the Boruta algorithm has determined that all the variables are important. Figure 4.28 shows that the most important variable for dataset 3, like dataset 1 and 2, is the Creatinine plus eGFR variable but this time has an importance value of less than 60. The figure also indicates the scale of the importance values is the least of the three datasets so far.

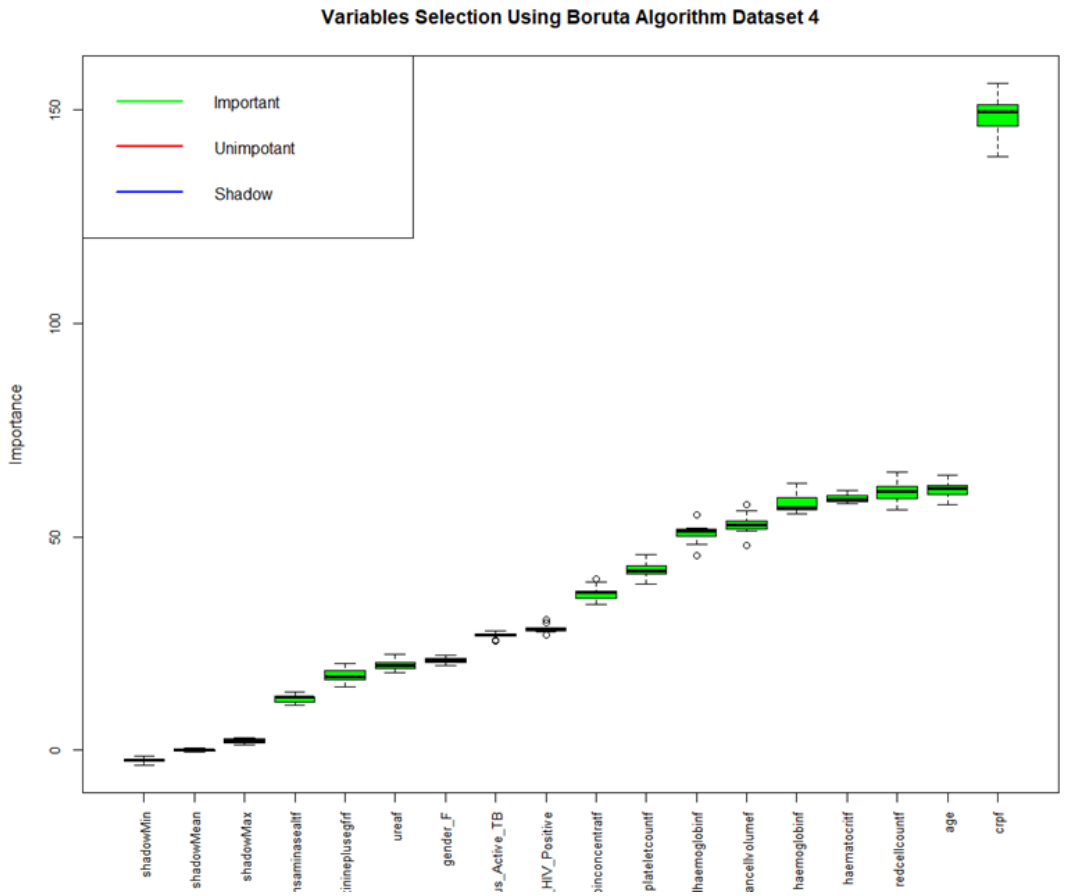

Figure 4.29: Variable Selection for Dataset 4 using the Boruta Feature Selection Algorithm.

Figure 4.29 is very similar to Figure 4.27 and shows that out of the 13 predictor variables the Boruta algorithm has determined that all the variables are important. Figure 4.29 shows that the most important variable for dataset 4 is the same as all the other datasets, the Creatinine plus eGFR variable.

An observation that can be made about all the Boruta figures is that there is a large discrepancy/gap between the most important variable and the second most important variable. This indicates that the Creatinine plus eGFR variable plays a very significant role in predicting the response variable.

4.6 Final Datasets

The four datasets were finalised after variable selection and were then split into training and test sets. The ratio of training set to test set was 70:30. Nguyen et al. (2021) proved that for the methods and models used in this study that the 70:30 ratio produced the best results. This ratio was chosen due to the characteristics of the training and test sets for the four datasets are represented in Table 4.10

Table 4.10: Final Datasets Characteristics

Dataset	Variables	Total Observations	Training Set Obs.	Test Set Obs.
1	38	60 711	42 497	18 214
2	14	60 711	42 497	18 214
3	39	17 762	12 433	5 329
4	14	44 614	31 229	13 385

4.7 Machine Learning Models

4.7.1 Random Forest Modelling

All the random forest models were created using the “randomForest" function in the “randomForest" package in R-studio. The models were first trained using the training subset of each dataset. A confusion matrix and out-of-bag (OOB) estimate error was obtained for each model to give an indication of the model’s performance. The models were then used in conjunction with the test subset of each dataset to see the models true performance. A confusion matrix and other performance measures, mentioned in Section 2.4.7, were used to determine the performance of each model.

Table 4.11 shows that the random forest model for Dataset 1 produces the lowest out-of-bag estimate error (19.77%) of the four datasets and that the random forest model for Dataset 3 produced the worst out-of-bag error (24.18%). All the models produce high out-of-bag estimated errors this indicates that the models struggle to correctly

predict the response variable, even though the models are trained on the same data they are validated on. This is an indication that the random forest models may struggle to correctly predict the response variables for the test datasets.

Table 4.11: Random Forest Models for the Four Datasets

Dataset	Type of RF	No. of Trees	No. of Split variables tried	OOB Estimates Error
1	Classification	500	6	19.77%
2	Classification	500	3	21.38%
3	Classification	500	6	25.93%
4	Classification	500	3	24.18%

4.7.1.1 Random Forest Model 1

Table 4.12 shows that random forest model 1 predicts predominately negative results and does not predict any severe results even though there are severe results present in the dataset. The model produces an accuracy of 79.93% which is only 2.4% better than the no-information rate shown in Table 4.13. Table 4.13 does show however that the model is significant compared to the no-information rate. The Model achieves a Kappa score of 0.2043 which as per Table 2.3 indicate slight agreement between the model predictions and actual response variables of the dataset.

Table 4.12: Test Data Confusion Matrix for Random Forest Model 1

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	13 992	2 224	1 138	35
Mild	129	559	98	5
Moderate	1	5	8	0
Severe	0	0	0	0

Table 4.13: Results from Test Set of Random Forest Model 1

Accuracy	95% CI	NIR	P-value	Kappa
0.7993	(0.7934, 0.8051)	0.7753	$2.061e^{-15}$	0.2043

Table 4.14 shows a detailed look into the class specific results of random forest model 1. The table shows the sensitivity, the specificity and the balanced accuracy of each class. The negative class has a very high sensitivity which is desirable but this is as a result of the model predicting a large number of negative responses. Looking at the sensitivity of the other class the scores are poor. The specificity again shows that the negative class prediction performance was good, however the prediction performance of the other classes was poor. The balanced accuracy shows the accuracy for each classes prediction performance when taking into account the structure/skewed nature of the data. The balanced accuracy shows that the mild class was predicted the best out of all the classes (59.20%). The negative class only achieved a balanced accuracy of 57.79%.

Table 4.14: Class Results for Random Forest Model 1

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9908	0.1991	0.0064	0
Specificity	0.1650	0.9849	0.9996	1
Balanced Accuracy	0.5779	0.5920	0.5030	0.5000

The overall multi-class AUC for random forest model 1 is 72.42%. This is broken down further into the specific class versus class ROC and AUC values shown in Figure 4.30. The figure shows that the performance of the first three ROCs, involving the negative class, perform well achieving AUCs of 82.4%, 76% and 85%. The last three plots show a decrease in performance with AUC values of 67%, 52.2% and 51.4%.

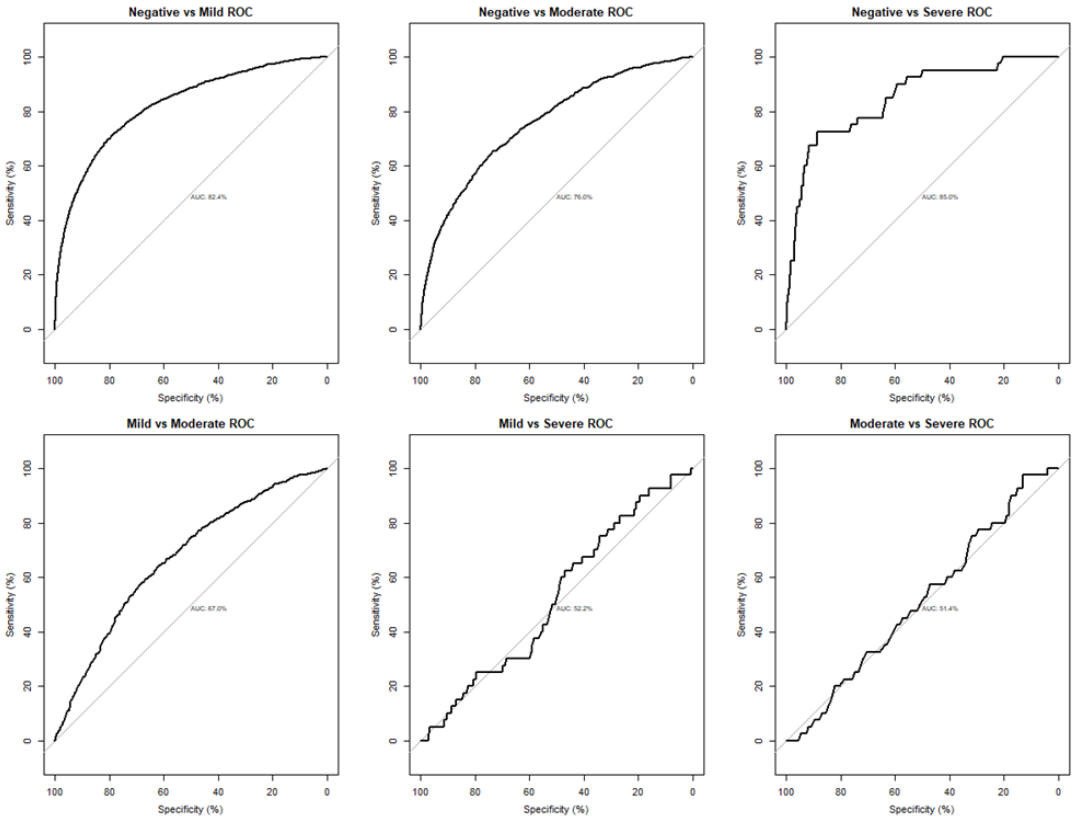

Figure 4.30: Receiver Operating Curves for Dataset 1

4.7.1.2 Random Forest Model 2

Evaluating the performance of random forest model 2 Table 4.15 shows that once again the model predicts predominately negative responses and does not predict any severe responses. The model produces an accuracy of 78.1% which is less than random forest model 1. Table 4.16 indicates that the p-value for the test against the NIR is equal to 0.03411 which at a 5% significance level shows that random forest model 2 is significant and differs from the NIR. However the model is only able to achieve a Kappa score of 0.1332 which as per Table 2.3 indicate little to no agreement between the model predictions and actual response variables of the dataset.

Table 4.15: Test Data Confusion Matrix for Random Forest Model 2

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	13 842	2 425	1 094	32
Mild	279	383	150	8
Moderate	1	0	0	0
Severe	0	0	0	0

Table 4.16: Results from Test Set of Random Forest Model 2

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.781	(0.7749, 0.787)	0.7753	0.03411	0.1332

Table 4.17 again shows a detailed look into the class specific results of the random forest model. The negative class achieves desirable sensitivity and specificity values but the remaining classes, especially the moderate and severe classes, produce poor results. The balanced accuracy shows that the negative class was predicted the best out of all the classes (55.62%) which was only slightly better than the mild class with a balanced accuracy of 55.40%.

Table 4.17: Class Results for Random Forest Model 2

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9802	0.1364	0	0
Specificity	0.1322	0.9716	0.9999	1
Balanced Accuracy	0.5562	0.5540	0.5000	0.5000

The overall multi-class AUC for random forest model 2 is 60.79%. This is broken down further into the specific class versus class ROC and AUC values shown in Figure 4.31. The ROC plots show the same trend as with random forest model 1, the first three plots show that performance of the first three cases is better than the lost three cases. The worst case is the last plot which shows moderate versus severe ROC.

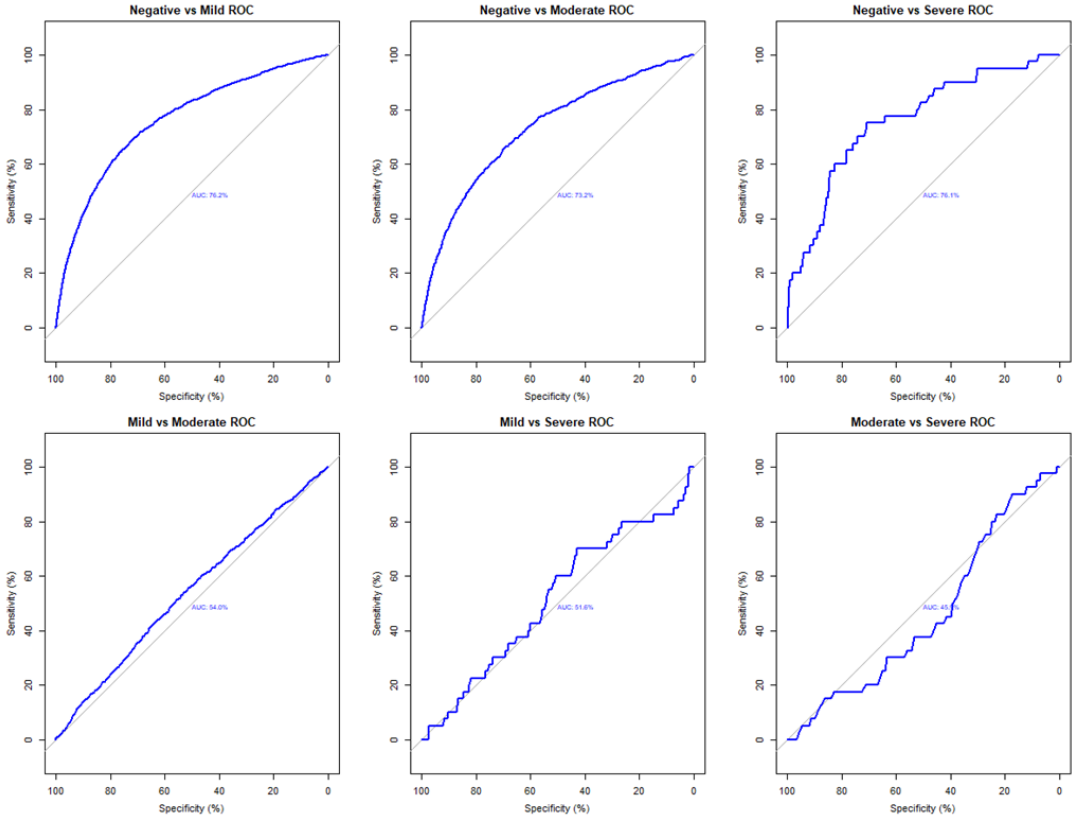

Figure 4.31: Receiver Operating Curves for Dataset 2

4.7.1.3 Random Forest Model 3

Table 4.18 shows that random forest model 3 performs well in predicting both negative and mild and moderate responses. This is seen by the majority predicted class in Table 4.18 for each row is equal to its actual response variable. The model, like the previous two models, is unable to predict any serve responses. The accuracy of the model is 74.50%, which is less than the other two previous model but this is because of the

nature of Dataset 3. The NIR for Dataset 3 is only 0.678 compared to 0.7753 for the previous two model. The p-value shown in Table 4.19 shows that random forest model 3 is significant and produces the best Kappa score (0.3306) of the three models thus far. The Kappa score indicates that there is fair agreement between the model predictions and actual response variables of the dataset.

Table 4.18: Test Data Confusion Matrix for Random Forest Model 3

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	3 510	776	356	16
Mild	95	408	84	1
Moderate	8	21	52	2
Severe	0	0	0	0

Table 4.19: Results from Test Set of Random Forest Model 3

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.745	(0.7331, 0.7566)	0.678	$2.2e^{-16}$	0.3306

Table 4.20 shows the class specific results of random forest model 3. The negative class achieved desirable sensitivity and specificity values, Whilst the mild and moderate classes produced their respective best sensitivity and specificity values. Table 4.20 shows the best balanced accuracy results thus far with the negative class achieving 65.12%, the mild class achieving 64.75%, the moderate class achieving 54.96% and the severe class once again achieving 50% since no severe perditions were made.

Table 4.20: Class Results for Random Forest Model 3

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9715	0.3386	0.1057	0
Specificity	0.3310	0.9564	0.9936	1
Balanced Accuracy	0.6512	0.6475	0.5496	0.5000

The overall multi-class AUC for random forest model 3 is 71.58%. This is broken down further into the specific class versus class ROC and AUC values shown in Figure 4.32. Once again the first three plots show a similar trend to the first 2 RF models but unlike the previous ROC plots for the previous two models the Mild versus Severe case produces a promising AUC value of 71.6%.

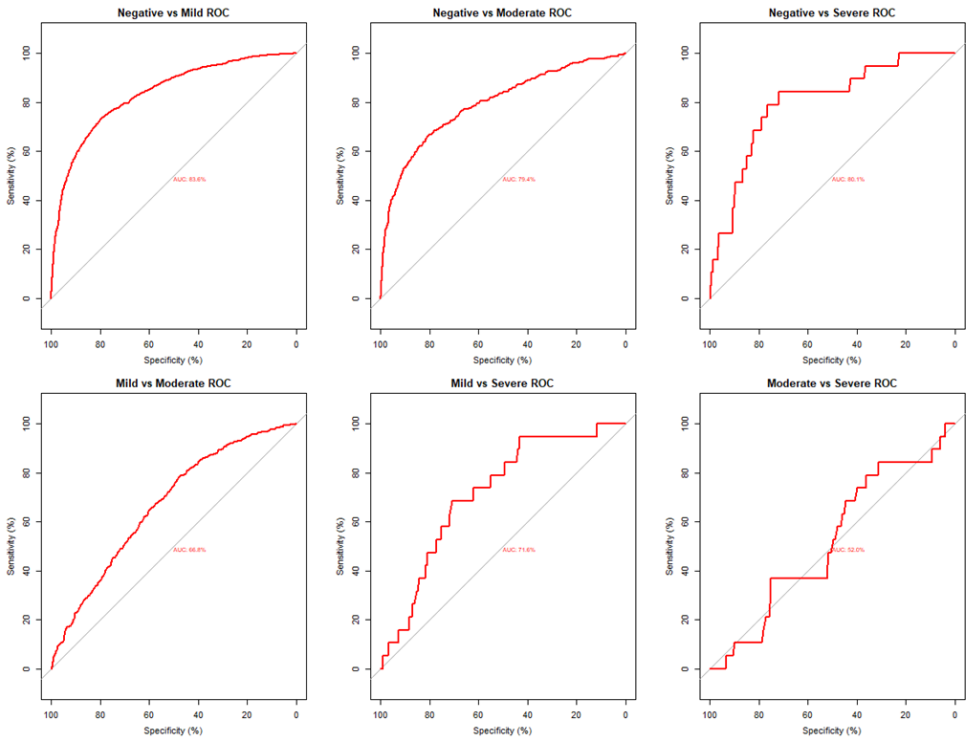

Figure 4.32: Receiver Operating Curves for Dataset 3

4.7.1.4 Random Forest Model 4

Table 4.21 reveals that random forest model 4 predicts mostly negative and mild responses. Although it does predict two moderate responses both are incorrect and just as with the previous models no severe results are predicted by the model. The model produces an accuracy of 75.51% which better than the NIR of 0.7388 and the p-value of Table 4.22 shows that the model is significant. The Model achieves a Kappa score of 0.2027, which as per Table 2.3 indicates slight agreement between the model predictions and actual response variables of the dataset.

Table 4.21: Test Data Confusion Matrix for Random Forest Model 4

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	9 562	1 790	911	32
Mild	325	545	212	6
Moderate	2	0	0	0
Severe	0	0	0	0

Table 4.22: Results from Test Set of Random Forest Model 4

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7551	(0.7477, 0.7624)	0.7388	$8.247e^{-06}$	0.2027

Table 4.23 takes a look at the class specific results of random forest model 4. The negative class has a high sensitivity which is desirable but this helped by the nature/structure of the dataset. Looking at the sensitivity and specificity of the other class are poor. The balanced accuracy shows that the negative class was predicted the best out of all the classes (59.26%) however this was only slightly better than the balance accuracy of the mild class (59.21%).

Table 4.23: Class Results for Random Forest Model 4

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9669	0.2334	0	0
Specificity	0.2182	0.9809	0.9998	1
Balanced Accuracy	0.5926	0.5921	0.4999	0.5000

The overall multi-class AUC for random forest model 4 is 63.04%. This is broken down further into the specific class versus class ROC and AUC values shown in Figure 4.33. This figure shows the familiar pattern with the cases including the negative class outperforming the other cases and the worst case being that of the moderate versus severe ROC.

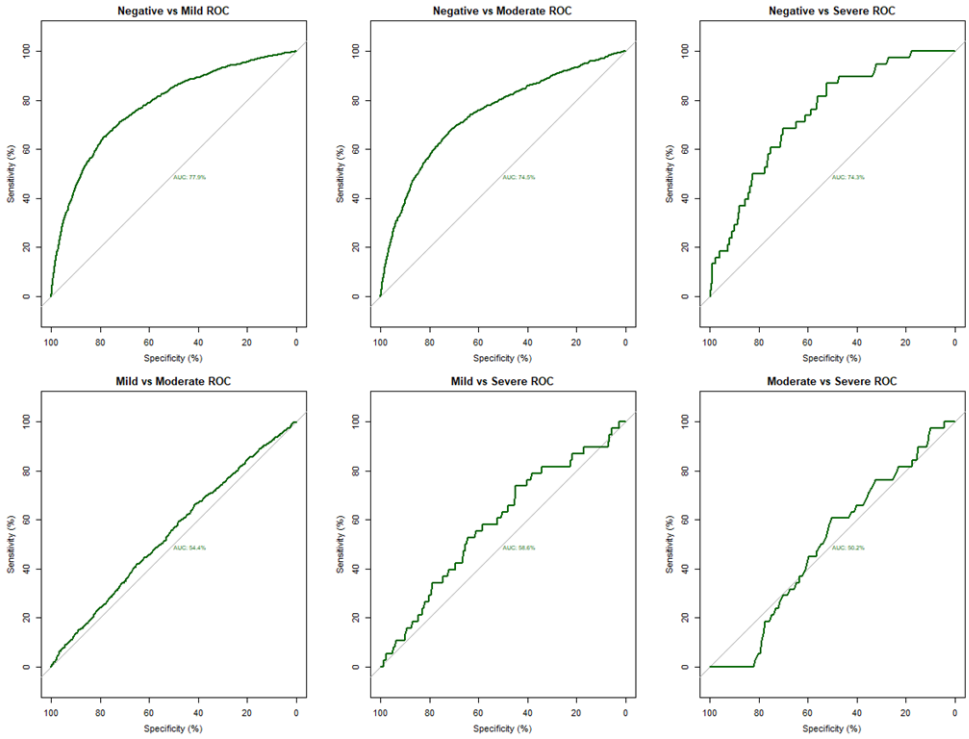

Figure 4.33: Receiver Operating Curves for Dataset 4

4.7.2 Neural Networks

All the Self-normalising Neural Networks were created using the “keras_model_sequential" function in the *“Keras"* and *“tensorflow"* package in R-studio. The models were first trained using the training subset of each dataset. A graph for each model shows the training accuracy and loss for each model over 100 epochs, these graphs gave an indication of each model’s performance. The models were then used in conjunction with the test subset of each dataset to see the models true performance. A confusion matrix and other performance measures, mentioned in Section 2.4.7, were used to determine the performance of each model.

The self-normalising neural network was created with two hidden layers. The input shape of the neural network was equal to the number of explanatory variables in each dataset however the input layer for all neural networks had 128 units/neurons and used a Scaled Exponential Linear Units (SELU) activation function. Both hidden layers also used the SELU activation function but the output layer used the softmax activation function which is built for multiple-class problems. The SELU activation function is used in the hidden layers because it ensures that the neural network self-normalises. The loss function applied for all neural networks was the categorical cross-entropy in conjunction with the adaptive moment estimation (ADAM) learning rate optimizer.

4.7.2.1 Self-normalising Neural Network 1

Figure 4.34 shows the loss error and mean accuracy performance of the neural network one on the training dataset. The figure shows that over the 100 epochs the loss error decreases from 0.61 to 0.37 and that the mean accuracy increase from 0.77 to 0.85. The training results indicate a promising model.

Tables 4.24, 4.25 and 4.26 show the performance of neural network one using the test dataset. The confusion matrix shows that the neural network is able to predict all four classes. The majority of predictions are negative responses which is expected due to the nature of the data being used. The accuracy of the network is 77.43%. This is lower than the NIR and hence the p-value shown in Table 4.25 is greater than 5%. Although the Accuracy is less than the NIR the network still produces a Kappa score

of 0.3385 which indicates fair agreement between the model predictions and actual response variables of the dataset.

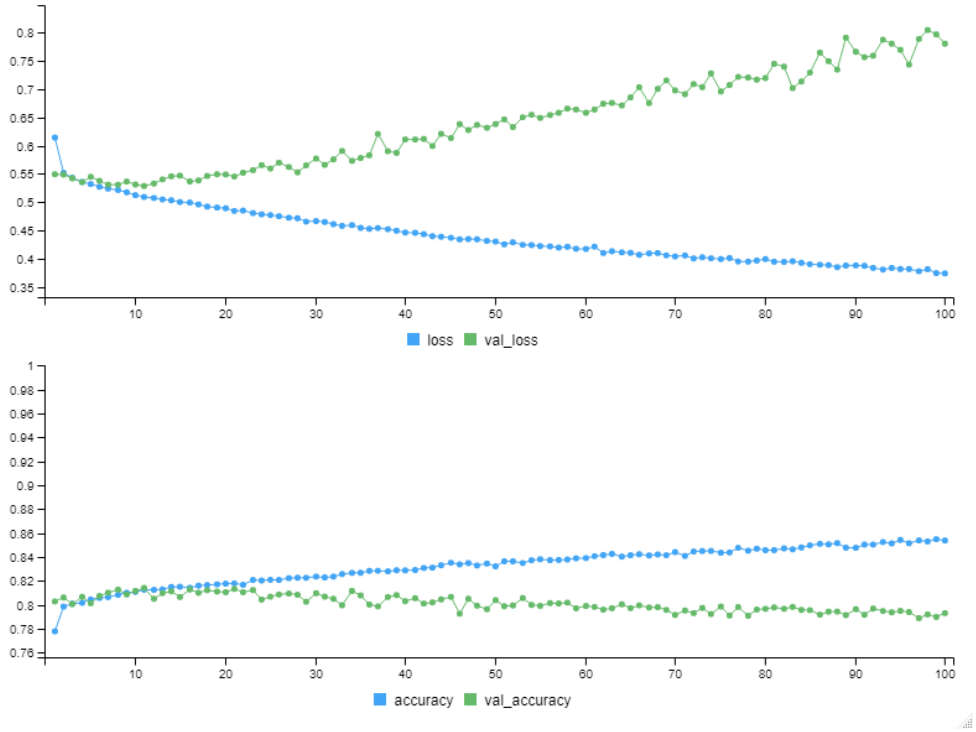

Figure 4.34: Accuracy and Loss Graphs for the Training of Self-normalising Neural Network One over a 100 Epochs

Table 4.24: Test Data Confusion Matrix for Self-normalising Neural Network 1

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	12 812	1 454	722	20
Mild	671	966	166	14
Moderate	561	334	325	5
Severe	78	54	31	1

Table 4.25: Results from Test Set of Self-normalising Neural Network 1

Accuracy	95% CI	NIR	P-value	Kappa
0.7743	(0.7682, 0.7804)	0.7753	0.6293	0.3385

The class specific performance results shown in Table 4.26 and show that the specificity and sensitivity for the negative class is the best and the worst for the severe class. The balanced accuracy is above 60% for three of the four classes, with the negative class achieving the best balanced accuracy of 68.53%.

Table 4.26: Class Results for Self-normalising Neural Network 1

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9072	0.3440	0.2613	0.0250
Specificity	0.4633	0.9778	0.9470	0.9910
Balanced Accuracy	0.6853	0.6444	0.6041	0.5000

The overall multi-class AUC for Self-normalising Neural Network one is 67.56%. Figure 4.35 shows the class versus class breakdown of the ROCs and AUC values. A similar pattern to that of the random forest models is evident with the slight change being that the worst class case is the mild versus severe ROC.

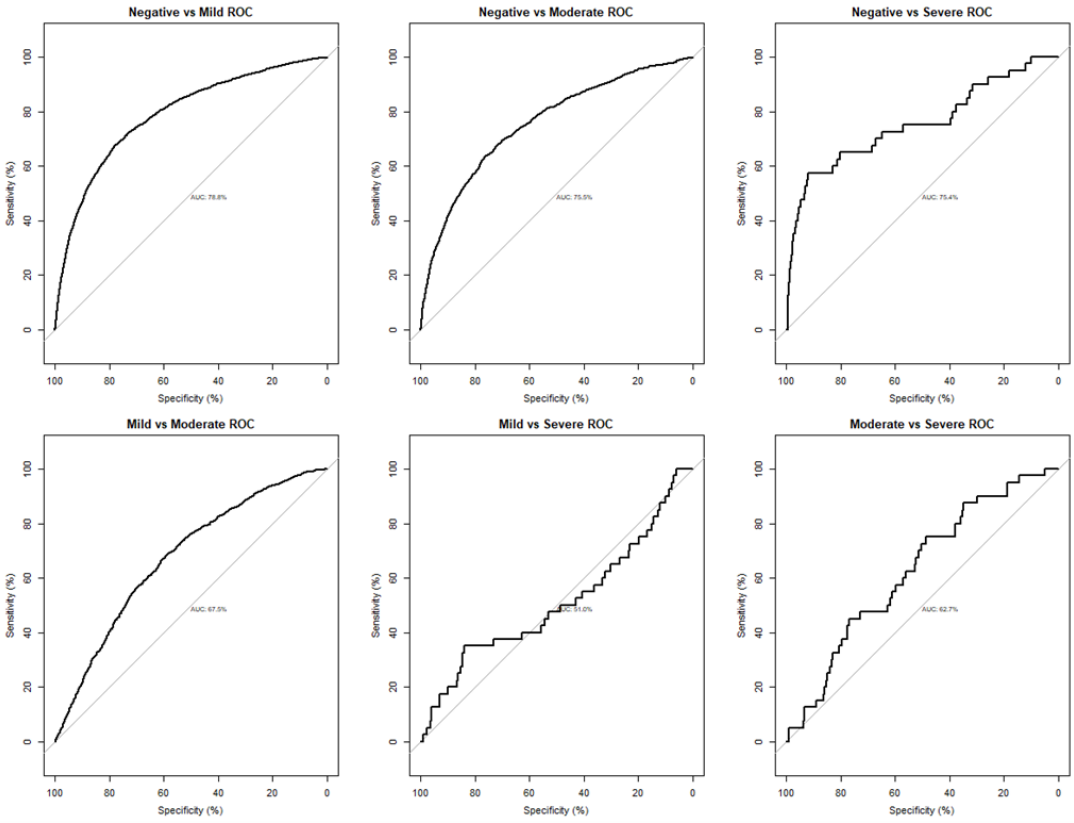

Figure 4.35: Receiver Operating Curves for Self-normalising Neural Network 1

4.7.2.2 Self-normalising Neural Network 2

Figure 4.36 shows the loss error and mean accuracy performance of neural network two on the training dataset. The figure shows that over the 100 epochs the loss error decreases from 0.66 to 0.55 and that the mean accuracy increase from 0.76 to 0.79. This shows that compared to neural network one, neural network two struggled to decrease the loss and struggled to increase the mean accuracy over the 100 epochs. The training performance suggests that neural network two won't perform as well and neural network one.

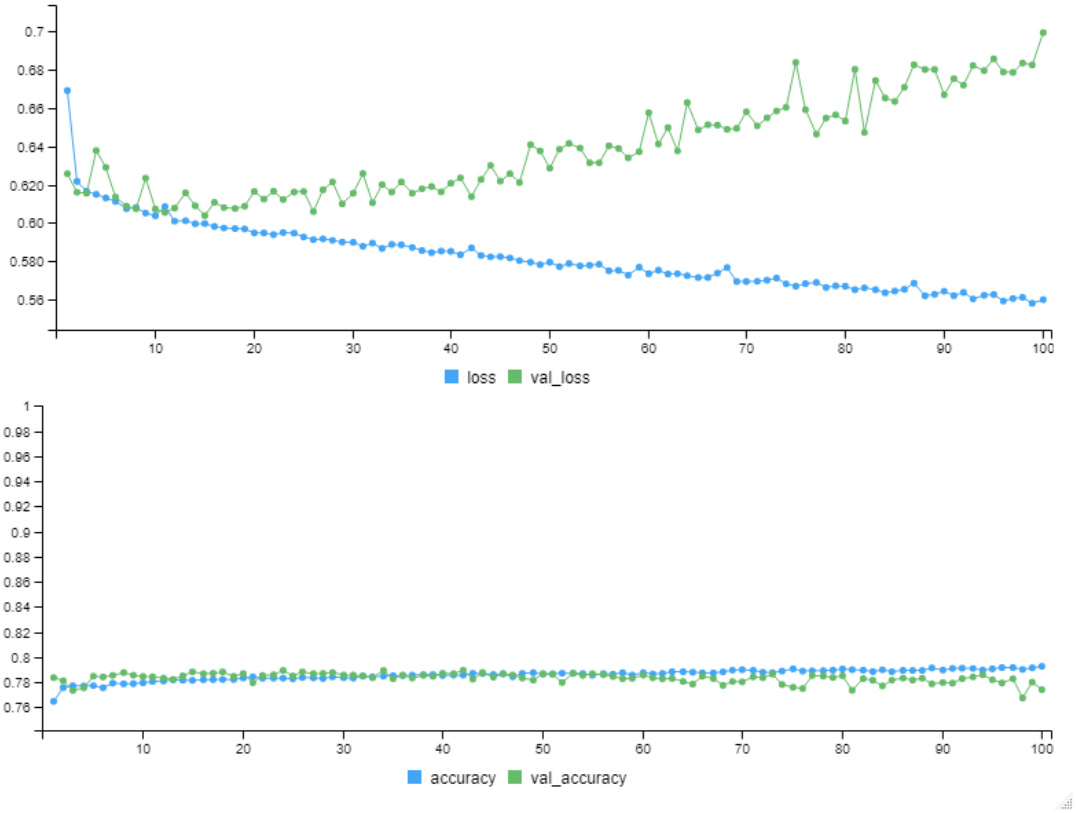

Figure 4.36: Accuracy and Loss Graphs for the Training of Self-normalising Neural Network Two over a 100 Epochs

Table 4.27, 4.28 and 4.29 shows the performance of neural network two using the test dataset. The confusion matrix shows that the neural network is able to predict all four classes, however it is unable to correctly predict any severe responses. The majority of predictions, just like neural network one, are negative responses. The accuracy of the network is 75.19%. This is much lower than the NIR and hence the p-value shown in Table 4.28 is greater than 5% it is in fact equal to 1. The neural network produces a kappa score of 0.196 which is significantly less than the score achieved by neural network one. The Kappa score of indicates none to slight agreement between the model predictions and actual response variables of the dataset.

Table 4.27: Test Data Confusion Matrix for Self-normalising Neural Network 2

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	13 131	1 906	888	33
Mild	508	489	179	5
Moderate	271	213	76	2
Severe	212	200	101	0

Table 4.28: Results from Test Set of Self-normalising Neural Network 2

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7519	(0.7456, 0.7582)	0.7753	1	0.196

The class specific performance results are shown in Table 4.29 and show that the specificity and sensitivity for the negative class is the best and the worst for the severe class. The balanced accuracy for the negative class is the best with a value of 61.95% and the severe class achieves a poor balanced accuracy of 48.59% which is expected as the neural network fails to correctly predict any severe responses.

Table 4.29: Class Results for Self-normalising Neural Network 2

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9298	0.1742	0.0611	0
Specificity	0.3091	0.9551	0.9714	0.9718
Balanced Accuracy	0.6195	0.5646	0.5162	0.4859

The overall multi-class AUC for Self-normalising Neural Network model 2 is 62.48%. Figure 4.37 shows a similar pattern to all the ROC figures up to this point. It is worth

mentioning that the last three plots show the worst performances of the mild versus moderate, mild versus severe and moderate versus severe ROCs.

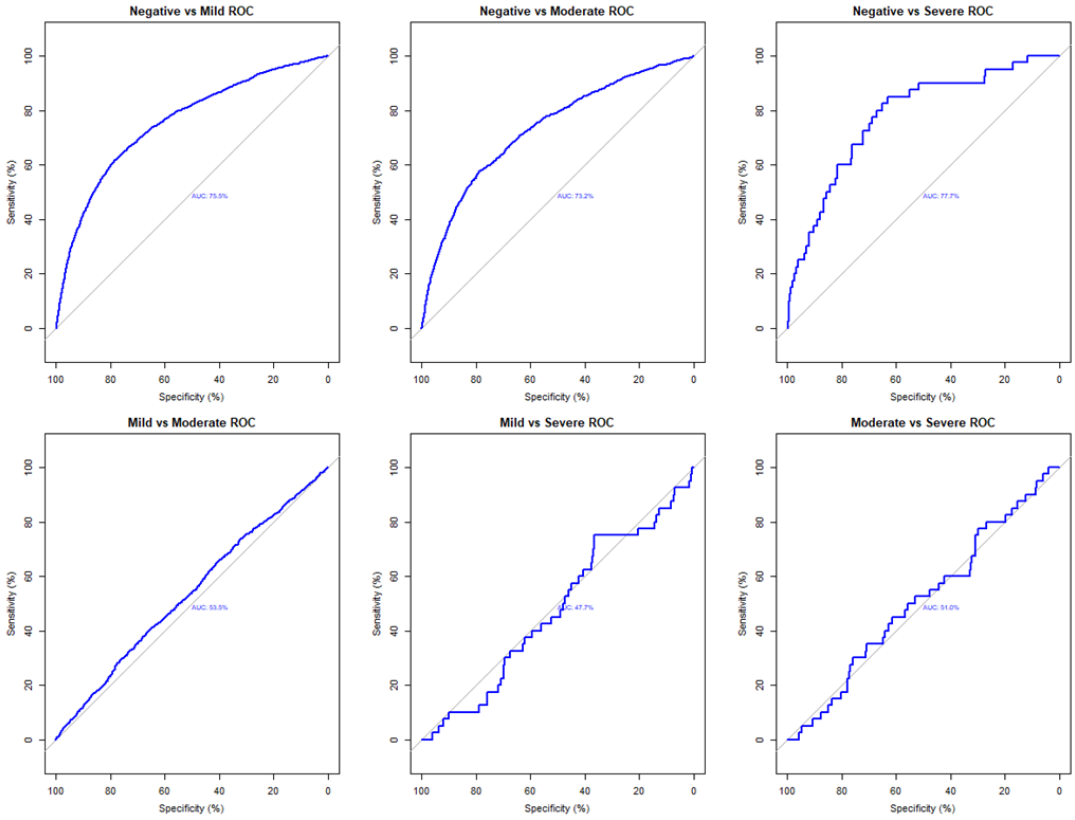

Figure 4.37: Receiver Operating Curves for Self-normalising Neural Network 2

4.7.2.3 Self-normalising Neural Network 3

Figure 4.38 shows the loss error and mean accuracy performance of neural network three on the training dataset. The figure shows that over the 100 epochs the loss error decreases from 0.81 to 0.34, the best decrease of the networks so far, and that the mean accuracy increase from 0.68 to 0.86. The increases and decreases in mean accuracy and loss respectively suggests that neural network three has the potential to be the best performing neural network.

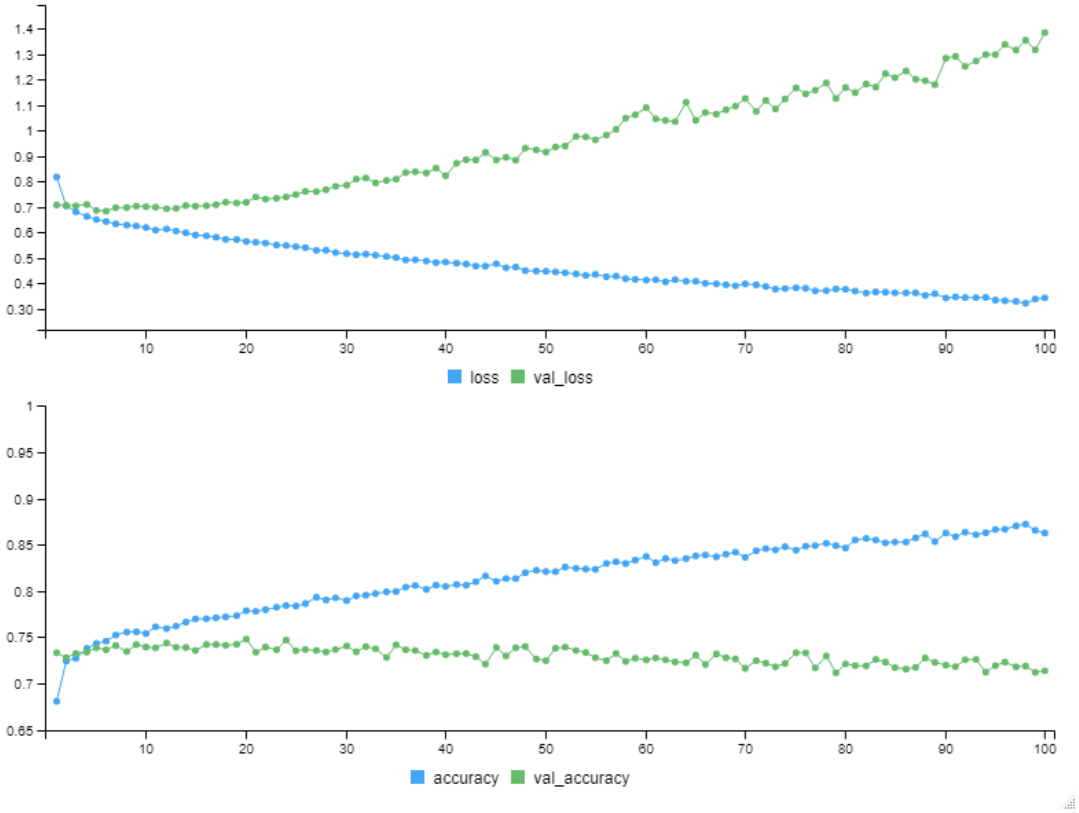

Figure 4.38: Accuracy and Loss Graphs for the Training of Self-normalising Neural Network Three over a 100 Epochs

The performance of neural network three is shown in the following tables, Table 4.30, 4.31 and 4.32. The confusion matrix shows that the neural network is able to predict all four classes but, like neural network two, is unable to correctly predict any severe responses. The accuracy of the network is 71.78% which is less than the previous two neural networks but Table 4.31 shows that the NIR for Dataset 3 is only 0.678. The network is significant in comparison to the NIR and the is confirmed by the p-value of $1.787e^{-10}$. The network is able to achieve a Kappa score of 0.3719 which indicates fair agreement between the model predictions and actual response variables of the dataset.

Table 4.30: Test Data Confusion Matrix for Self-normalising Neural Network 3

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	3 159	560	244	10
Mild	293	510	85	4
Moderate	143	117	156	5
Severe	18	18	7	0

Table 4.31: Results from Test Set of Self-normalising Neural Network 3

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7178	(0.7055, 0.7298)	0.678	$1.787e^{-10}$	0.3719

The class specific performance results shown in Table 4.32 and are the best results seen for the first three classes. The specificity and sensitivity for the mild and moderate classes are the best of the networks thus far. The balanced accuracy for the first three classes are also the best seen thus far. 70%, 66.53% and 63.11% for the negative, mild and moderate classes respectively.

Table 4.32: Class Results for Self-normalising Neural Network 3

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.8743	0.4232	0.3171	0
Specificity	0.5256	0.9074	0.9452	0.9919
Balanced Accuracy	0.7000	0.6653	0.6311	0.4960

The overall multi-class AUC for Self-normalising Neural Network model 3 is 69.42%. The breakdown of the ROC performances shown in Figure 4.39 follow the expected

pattern with the negative versus the mild ROC showing the best performance and the moderate versus severe ROC showing the worst performance.

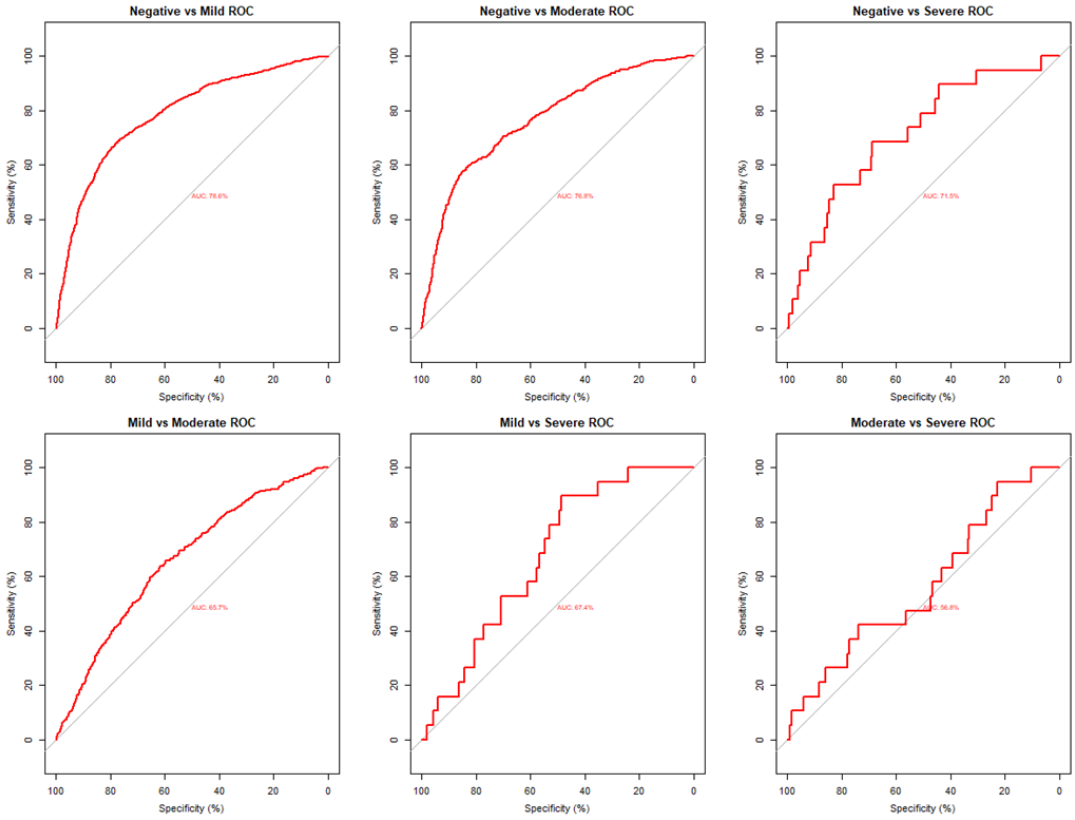

Figure 4.39: Receiver Operating Curves for Self-normalising Neural Network 3

4.7.2.4 Self-normalising Neural Network 4

Figure 4.40 shows the loss error and mean accuracy performance of neural network one on the training dataset. The figure shows that over the 100 epochs the loss error decreases from 0.73 to 0.59 and that the mean accuracy increase from 0.72 to 0.77. This neural network has a similar training performance to neural network two and suggest that the model may struggle to predict the correct responses when using the test dataset.

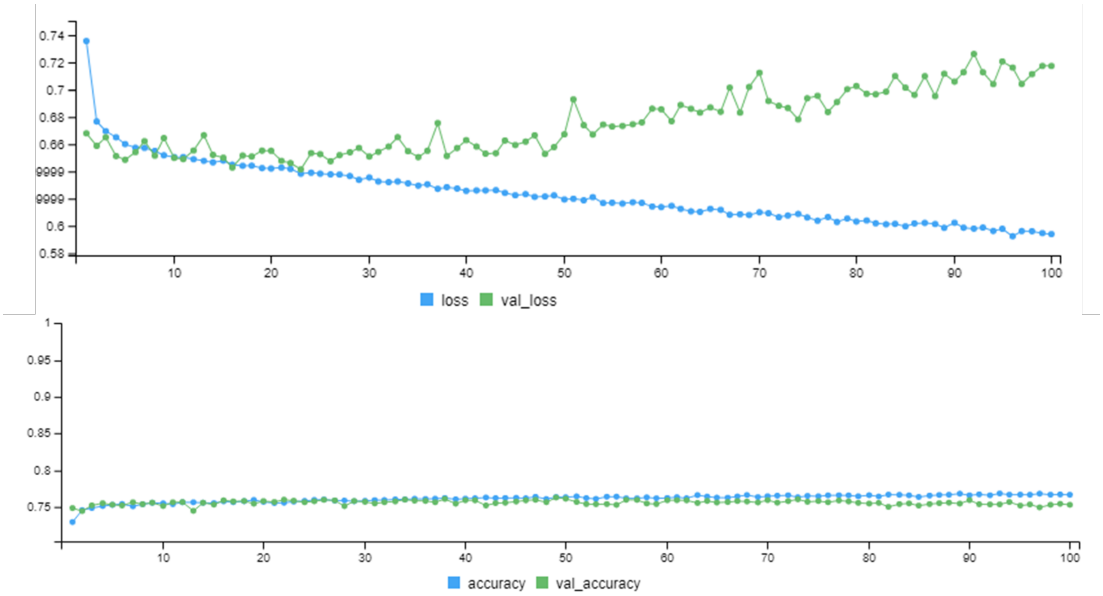

Figure 4.40: Accuracy and Loss Graphs for the Training of Self-normalising Neural Network Four over a 100 Epochs

Table 4.33, 4.34 and 4.35 show the performance of neural network four using the test dataset. The confusion matrix shows that the neural network is able to predict all four classes and can correctly predict the severe class, even if it only correctly predicts two responses. The accuracy of the network is 72.47%. This is lower than the NIR and hence the p-value shown in Table 4.34 is greater than 5%. The network produces a Kappa score of 0.2576 which indicates slight agreement between the model predictions and actual response variables of the dataset.

Table 4.33: Test Data Confusion Matrix for Self-normalising Neural Network 4

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	8 859	1 353	690	25
Mild	734	779	326	11
Moderate	148	90	60	0
Severe	148	113	47	2

Table 4.34: Results from Test Set of Self-normalising Neural Network 4

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7247	(0.7170, 0.7322)	0.7738	0.9999	0.2576

The class specific performance results shown in Table 4.35 and show that the specificity and sensitivity for all the classes. The negative class performance the best and achieves a balanced accuracy of 65.22%. The mild class produces a balanced accuracy of 61.83%. Finally the moderate and severe classes produce similar results, a balanced accuracy of 51.70% and 51.48% respectively.

Table 4.35: Class Results for Self-normalising Neural Network 4

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.8958	0.3336	0.0534	0.0526
Specificity	0.4085	0.9031	0.9806	0.9769
Balanced Accuracy	0.6522	0.6183	0.5170	0.5148

The overall multi-class AUC for Self-normalising Neural Network model 4 is 62.72%. Figure 4.39 shows that the last three cases perform significantly worse than the first three cases.

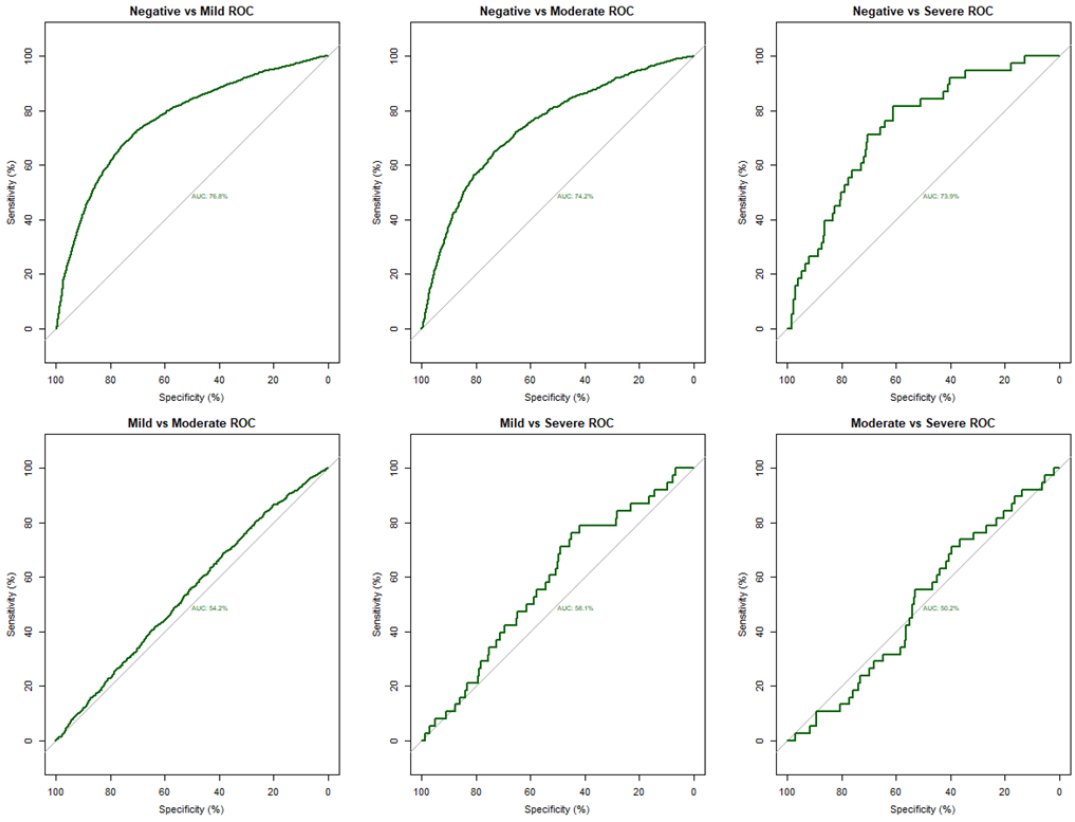

Figure 4.41: Receiver Operating Curves for Self-normalising Neural Network 4

4.8 Logistic Regression Models

Logistic regression was conducted on all four of the datasets using three different methods namely, multinomial, ordinal and baseline-category logistic regression. The process started by training the models using the training set of each dataset. The trained model was then used to predict the response variable for a new dataset (the test dataset). The performance of the model was then reviewed and conclusions about each model were made.

4.8.1 Multinomial Logistic Regression

The Multinomial logistic regression models were created in r-studio using the *multinom* function in the *nnet* package.

4.8.1.1 Multinomial Logistic Regression Model 1

Multinomial logistic regression model 1's intercept and significant parameter estimates are shown in Table 4.36. Table 4.36 indicates that the intercept, Haematocrit, HIV Status and TB Status estimates have the greatest impact on the model and its performance. Model 1's performance is shown in Table 4.37 and 4.38. The class specific performance of model 1 is shown in Table 4.39. Model 1 is created using Dataset 1. The confusion matrix shows that the multinomial logistic regression model is able to predict all four classes. However it fails to correctly predict any severe class responses. The accuracy of model 1 is 77.29%. This is lower than the NIR of 77.53% and hence the p-value shown in Table 4.38 is greater than 5%. Model 1 produces a Kappa score of 0.2106 which indicates slight agreement between the model predictions and actual response variables of the dataset.

Table 4.36: Significant Coefficients for Multinomial Logistic Regression Model 1

Variable	Mild Estimate	Moderate Estimate	Severe Estimate
Intercept	-11.88	-11.59	-0.94
Gender Female	0.31	0.17	-1.18
Age	0.02	0.01	0.01
Albumin	-0.02	0.03	-0.01
Glucose Random	0.01	0.01	-0.02
International Normalized Ratio (INR)	-0.18	0.23	0.46
Fibrinogen	0.23	0.09	-0.05
Red Cell Count	0.72	-0.21	0.15
Haematocrit	-2.11	5.65	0.49
Mean Cell Haemoglobin Concentration	0.15	0.16	-0.08
Neutrophils	-0.07	-0.03	0.02
Monocytes	0.02	-0.33	-0.04
Eosinophils	-0.38	-0.05	-0.72
HIV Status	-1.20	-0.36	1.10
TB Status	-2.65	0.12	-2.20

Table 4.37: Test Data Confusion Matrix for Multinomial Logistic Regression Model 1

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	13 288	2001	1069	32
Mild	758	774	156	8
Moderate	45	20	16	0
Severe	31	13	3	0

Table 4.38: Results from Test Set of Multinomial Logistic Regression Model 1

Accuracy	95% CI	NIR	P-value	Kappa
0.7729	(0.7668, 0.779)	0.7753	0.7854	0.2106

The class specific performance results show that the negative class achieved the best sensitivity and specificity values of the four classes but the mild class achieves the best balanced accuracy value. 60.79% compared to the 59.14% achieved by the negative class. The severe class achieves a balance accuracy score less 50% which is expected as the model fail to correctly predict any severe responses.

Table 4.39: Class Results for Multinomial Logistic Regression Model 1

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9409	0.2756	0.0129	0
Specificity	0.2419	0.9402	0.9962	0.9974
Balanced Accuracy	0.5914	0.6079	0.5045	0.4987

The overall multi-class AUC for Multinomial Logistic Regression Model 1 is 68.50%. Figure 4.42 keeps to the established ROC plot established so far. It however does show that the mild versus severe case produces the worst AUC of 51.8% compared to the moderate versus severe case that produces an impressive AUC of 63.9%.

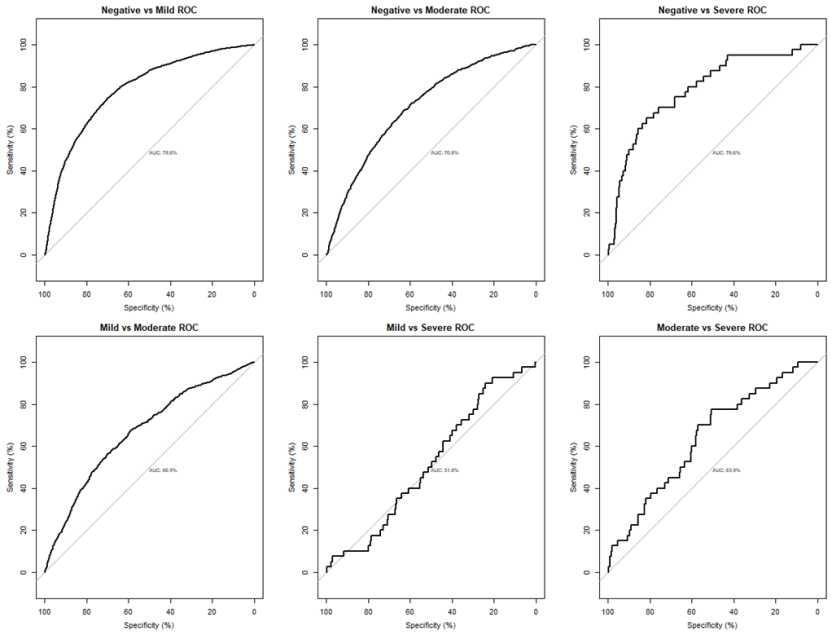

Figure 4.42: Receiver Operating Curves for Multinomial Logistic Regression Model 1

4.8.1.2 Multinomial Logistic Regression Model 2

Dataset 2 was used to create model 2, Multinomial logistic regression model 2's intercept and significant parameter estimates are shown in Table 4.40. Table 4.40 indicates that the intercept, Haematocrit, Red Cell Count and TB Status estimates have the greatest impact on the model and its performance. Model 2's performance is shown in Table 4.41 and 4.42. Further the class specific performance of model 2 is shown in Table 4.43. The confusion matrix shows that model 2 is unable to predict all four classes, it is only able to predict the negative and mild classes. The accuracy of model 2 is 77.30%. This is lower than the NIR of 77.53% and hence produces a p-value of 0.7749 shown in Table 4.38. Model 2 produces a very poor Kappa score of 0.0257 which indicates no agreement between the model predictions and actual response variables of the dataset.

Table 4.40: Significant Coefficients for Multinomial Logistic Regression Model 2

Variable	Mild Estimate	Moderate Estimate	Severe Estimate
Intercept	-8.12	-7.97	-10.01
Gender Female	0.34	0.34	0.38
Age	0.02	0.01	0.01
Red Cell Count	0.80	1.03	0.74
Haematocrit	-5.01	-8.43	-7.14
Mean Cell Volume	0.02	0.04	0.03
Mean Cell Haemoglobin Concentration	0.06	0.01	0.01
HIV Status	-0.64	-0.74	-0.22
TB Status	-1.48	-1.08	-1.69

Table 4.41: Test Data Confusion Matrix for Multinomial Logistic Regression Model 2

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	13 995	2 723	1 215	38
Mild	127	85	29	2
Moderate	0	0	0	0
Severe	0	0	0	0

Table 4.42: Results from Test Set of Multinomial Logistic Regression Model 2

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.773	(0.7669, 0.7791)	0.7753	0.7749	0.0257

The class specific performance results further show the poor performance of model 2. The negative class sensitivity and specificity values are good however this is due to the nature of the data and the fact that the model predicts mostly negative results. The balanced accuracy of all four classes are less than desirable with the best performing class being the mild class with a balanced accuracy of 51%.

Table 4.43: Class Results for Multinomial Logistic Regression Model 2

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9910	0.0303	0	0
Specificity	0.0284	0.9897	1	1
Balanced Accuracy	0.5097	0.5100	0.5000	0.5000

The overall multi-class AUC for Multinomial Logistic Regression Model Two is 60.11%. Figure 4.43 shows that the mild versus moderate, mild versus severe and the moderate versus severe struggle and produces AUC values of 53.4%, 51.5% and 50.4% respectively. The performance of the first three case is significantly better than the last three, however it is noticeable that the performance of the ROCs is significantly less for Dataset 2 for all models.

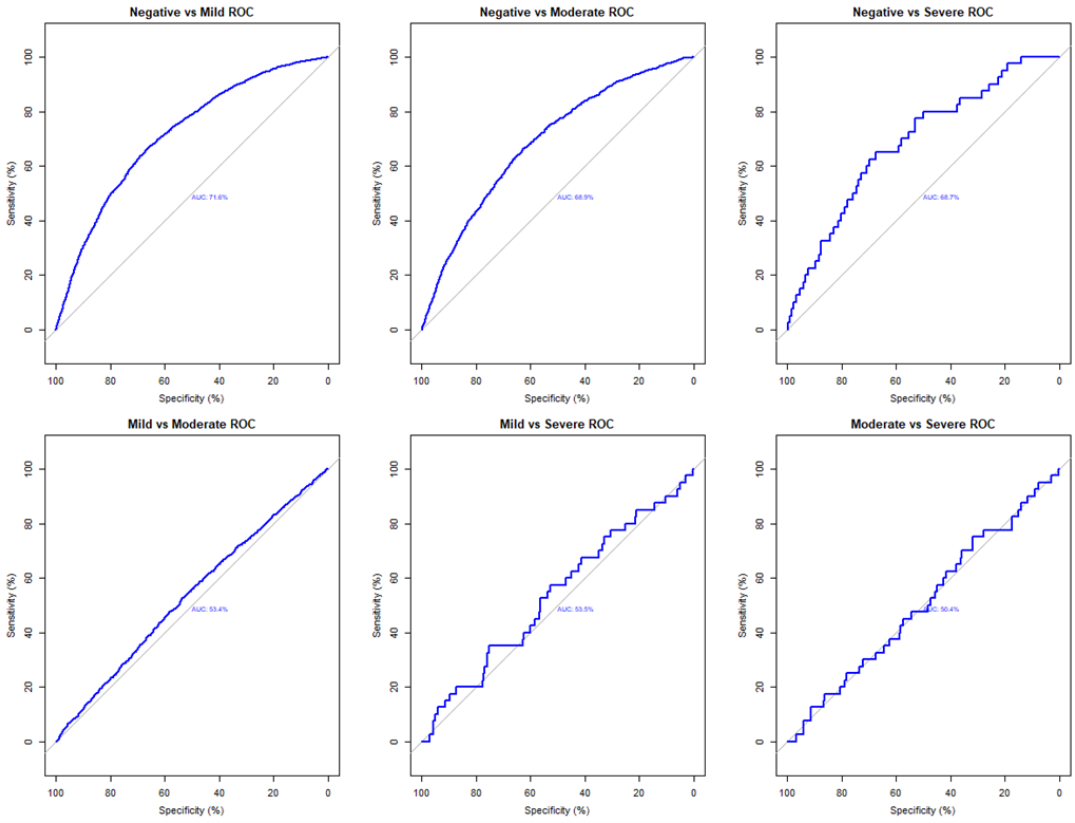

Figure 4.43: Receiver Operating Curves for Multinomial Logistic Regression Model 2

4.8.1.3 Multinomial Logistic Regression Model 3

Model 3 was created using Dataset 3. Multinomial logistic regression model 3's intercept and significant parameter estimates are shown in Table 4.44. Table 4.44 indicates that the intercept, Haematocrit (only the mild estimate), Eosinophils and TB Status estimates have the greatest impact on the model and its performance. The intercept, Eosinophils and TB Status coefficients are all negative. Model 3's performance is shown in Table 4.45 and 4.46. The class specific performance of model 3 is shown in Table 4.47. The confusion matrix shows that model 3 is able to predict all four classes, however it is unable to correctly predict any severe responses. The accuracy of model 2 is 71.23% which is lower than the first two models but as has been seen with Dataset 3 for the previous types of models, the NIR for Dataset 3 is much lower. The NIR is 67.8% and hence means model 3 is significantly different and this is backed up by the

p-value of $3.292e^{-08}$ found in Table 4.46. Model 3 produces the best Kappa score for the multinomial logistic regression, a score of 0.3339 which indicates fair agreement between the model predictions and actual response variables of the dataset.

Table 4.44: Significant Coefficients for Multinomial Logistic Regression Model 3

Variable	**Mild Estimate**	**Moderate Estimate**	**Severe Estimate**
Intercept	-7.63	-3.85	-1.37
Gender Female	0.45	-0.19	-0.05
Age	0.02	0.01	0.02
HbA1c	0.12	-0.9	-0.4
Glucose Random	-0.03	0.01	0.01
International Normalized Ratio (INR)	-0.14	-0.37	-0.45
Haematocrit	14.59	-0.13	0.45
Mean Cell Volume	-0.04	0.01	-0.02
Monocytes	-0.34	-0.49	0.02
Eosinophils	-3.85	-4.17	-0.60
HIV Status	-0.26	-1.06	0.70
TB Status	-1.33	-0.99	-0.97

Table 4.45: Test Data Confusion Matrix for Multinomial Logistic Regression Model 3

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	3 164	612	302	14
Mild	372	557	114	3
Moderate	64	34	75	2
Severe	9	2	1	0

Table 4.46: Results from Test Set of Multinomial Logistic Regression Model 3

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7123	(0.7, 0.7245)	0.678	$3.292e^{-08}$	0.3339

The class specific performance results show that the negative a mild classes perform well. The balanced accuracy of the negative class is 66.75% and the balanced accuracy of the mild class, which is the best of the four classes, is 67.18%. The severe class produce a balanced accuracy of 49.89% and is an expected result since not severe responses were correctly predicted.

Table 4.47: Class Results for Multinomial Logistic Regression Model 3

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.8757	0.4622	0.1524	0
Specificity	0.4592	0.8814	0.9785	0.9977
Balanced Accuracy	0.6675	0.6718	0.5655	0.4989

The overall multi-class AUC for Multinomial Logistic Regression Model Three is 69.54%. Figure 4.44 shows interesting outcomes for the ROCs performance. The negative versus mild case produces the best AUC (78.9%) and the worst case is the moderate versus severe (57.8%). The interesting outcomes are that the performances of the mild versus moderate (70.4%) and mild versus severe (75.3%) are better than the negative versus severe case (66%).

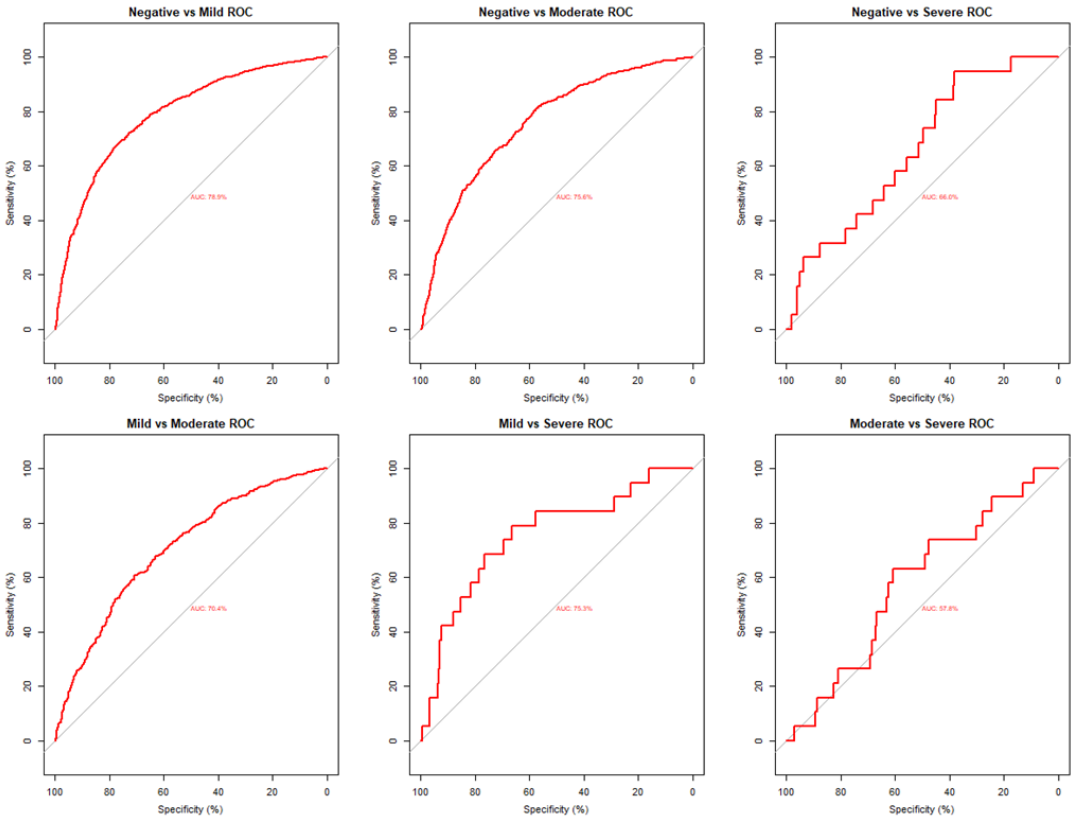

Figure 4.44: Receiver Operating Curves for Multinomial Logistic Regression Model 3

4.8.1.4 Multinomial Logistic Regression Model 4

The final multinomial logistic regression model, model 4 was created using Dataset 4. Multinomial logistic regression model 4's intercept and significant parameter estimates are shown in Table 4.48. Table 4.48 indicates that the intercept, Haematocrit and TB Status estimates have the greatest impact on the model and its performance. Once again the values of the coefficients for the intercept, Haematocrit and TB Status are all negative. Model 4's performance is shown in Table 4.49, 4.50 and 4.43. The confusion matrix shows that model 4, like model 2, is unable to predict the moderate and severe class responses, it is only able to predict the negative and mild classes. The accuracy of model 4 is 73.62%. This is lower than the NIR of 73.88% and hence produces a p-value of 0.7578 shown in Table 4.50. Model 4 produces an unsurprisingly poor Kappa score of 0.0515 which indicates no agreement between the model predictions

and actual response variables of the dataset.

Table 4.48: Significant Coefficients for Multinomial Logistic Regression Model 4

Variable	Mild Estimate	Moderate Estimate	Severe Estimate
Intercept	-8.06	-8.57	-12.36
Gender Female	0.44	0.47	0.01
Red Cell Count	0.71	0.91	0.78
Haematocrit	-2.27	-6.07	-7.50
Mean Cell Haemoglobin Concentration	0.09	0.05	0.09
HIV Status	-0.37	-0.49	-0.15
TB Status	-1.58	-0.87	-1.45

Table 4.49: Test Data Confusion Matrix for Multinomial Logistic Regression Model 4

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	9 700	2 181	1 068	34
Mild	189	154	55	4
Moderate	0	0	0	0
Severe	0	0	0	0

Table 4.50: Results from Test Set of Multinomial Logistic Regression Model 4

Accuracy	95% CI	NIR	P-value	Kappa
0.7362	(0.7286, 0.7436)	0.7388	0.7578	0.0515

The class specific performance results further show the poor performance of model 4. The balanced accuracy of all four classes are less than desirable with the best performing class being the mild class with a balanced accuracy of 52.18% and the negative class having a balanced accuracy of 52.09%.

Table 4.51: Class Results for Multinomial Logistic Regression Model 4

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9809	0.0660	0	0
Specificity	0.0609	0.9776	1	1
Balanced Accuracy	0.5209	0.5218	0.5000	0.5000

The overall multi-class AUC for Multinomial Logistic Regression Model Four is 60.97%. Figure 4.45 shows that the ROC performance of the various cases reverts back the established pattern seen previously. The best performing case is the negative versus mild case (72.5%) and the worst performing case is the moderate versus severe case (54%).

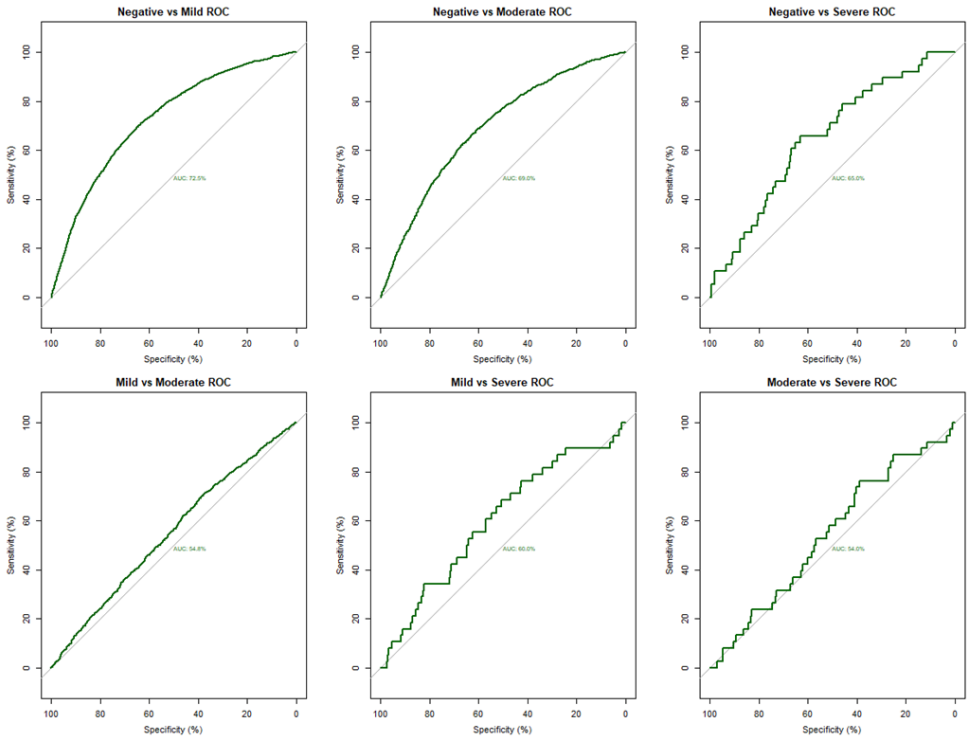

Figure 4.45: Receiver Operating Curves for Multinomial Logistic Regression Model 4

4.8.2 Ordinal Logistic Regression

The Multinomial logistic regression model ignored the implied ordering in the cases (Severe>Moderate>Mild>Negative) the Ordinal logistic regression model considers order. The Ordinal logistic regression models were created in r-studio using the “polr" function in the “MASS" package.

4.8.2.1 Ordinal Logistic Regression Model 1

The “polr" function failed and was unable to fit the Ordinal logistic regression model using Dataset 1. R returned an “initial value in ’vmmin’ is not finite" error. Multiple different starting values were used including the regression coefficients from the other ordinal models and the error was still produced. This shows that the problem is more than just an initial value problem. Further analysing the problem revealed it could be complete separation or quasi-complete separation (Bruin, 2011). Complete separation is when a predictor variable is completely separated by the outcome variable. Quasi-complete separation is when a predictor variable or a combination of predictor variables are almost completely separated by the outcome variable (Bruin, 2011). Attempts were made to resolve the problem but still the ordinal model was unable to fitted to Dataset 1. Therefore the conclusion is that an ordinal model is unable to be used for Dataset 1.

4.8.2.2 Ordinal Logistic Regression Model 2

Ordinal logistic regression model 2 was created using the training dataset of Dataset 2. Table 4.52 shows the significant coefficients for the model. The table indicates that the intercept, Haematocrit and TB Status coefficients have the greatest impact on the model. The model’s performance was then tested using the test dataset of Dataset 2. Model 2’s performance is shown in Table 4.53, 4.54 and 4.55. Table 4.53 shows that model 2 mainly predicts negative responses. The model is unable to predict any severe responses and even though it can predict moderate responses it fail to correctly predict any moderate responses. The model achieves an accuracy of 77.44% which is less than the NIR value of 77.53%. This means that model 2 is not a significant model compared to a naive model that just predicts negative responses. Based on the other performance

measure of model 2 it is unsurprising that the model produces the worst Kappa score of all models. A score of 0.0048 indicates no agreement between the model predictions and actual response variables of the dataset.

Table 4.52: Significant Coefficients for Ordinal Logistic Regression Model 2

Variable	**Coefficients**
Gender Female	0.33
Age	0.01
Urea	-0.01
Red Cell Count	0.86
Haematocrit	-6.22
Mean Cell Volume	0.03
Mean Cell Haemoglobin Concentration	0.04
HIV Status	-0.67
TB Status	-1.32
Intercept	**Values**
Negative|Mild	7.15
Mild|Moderate	8.57
Moderate|Severe	12.34

Table 4.53: Test Data Confusion Matrix for Ordinal Logistic Regression Model 2

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	14 098	2 785	1 237	39
Mild	8	7	7	0
Moderate	16	16	0	1
Severe	0	0	0	0

The class specific performance results further show the model performs poorly. The balanced accuracy of all four classes are less than desirable with the best performing

Table 4.54: Results from Test Set of Ordinal Logistic Regression Model 2

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7744	(0.7683, 0.7805)	0.7753	0.6225	0.0048

class being the negative class with a balanced accuracy of 50.29%. The performance of ordinal logistic regression model 2 is very poor.

Table 4.55: Class Results for Ordinal Logistic Regression Model 2

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9983	0.0025	0	0
Specificity	0.0076	0.9990	0.9981	1
Balanced Accuracy	0.5029	0.5008	0.4990	0.5000

The overall multi-class AUC for Ordinal Logistic Regression Model Two is 59.87%. The class versus class ROCs are shown in Figure 4.46. The figure shows the trend established with Dataset 2. The first three cases perform significantly better than the last three cases with the later producing AUC just greater than 50%. The best performing cases is the negative versus mild case (71.5%).

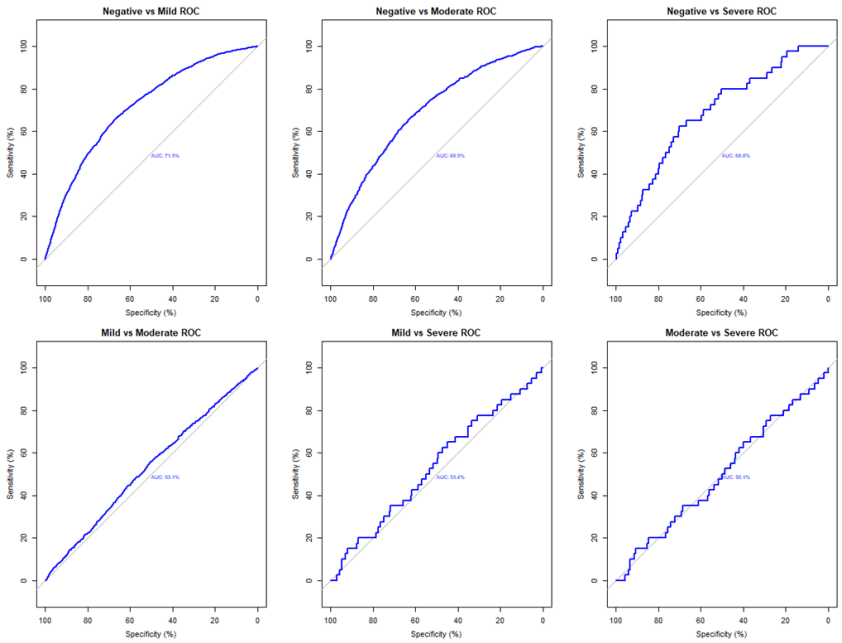

Figure 4.46: Receiver Operating Curves for Ordinal Logistic Regression Model 2

4.8.2.3 Ordinal Logistic Regression Model 3

Ordinal logistic regression model 3 was created using Dataset 2. Model 3's intercepts values and significant variable coefficients are shown in Table 4.56 and indicates that the Haematocrit, Eosinophils, TB Status and intercept values have the greatest impact on the model and its performance. Model 3's performance is shown in Tables 4.57, 4.58 and 4.59. The confusion matrix shows that model 3 can successfully predict the negative, mild and moderate classes, however model 3 is unable to predict any severe responses. Model 3 achieves an accuracy of 70.13% which is grater than the NIR of 67.8%. The p-value confirms that model 3 is significant in comparison to the NIR. The model produces a Kappa score of 0.1971 which indicates none to slight agreement between the model predictions and actual response variables of the dataset.

Table 4.56: Significant Coefficients for Ordinal Logistic Regression Model 3

Variable	**Coefficients**
Gender Female	0.37
Age	0.01
Urea	0.01
Total Bilirubin	-0.01
Creatine kinase-MB (CK-MB)	0.01
HbA1c	0.01
International Normalized Ratio (INR)	-0.25
Fibrinogen	0.15
D-dimer	-0.01
Haematocrit	4.75
Mean Cell Volume	-0.01
Mean Cell Haemoglobin Concentration	0.13
Neutrophils	-0.02
Lymphocytes	-0.01
Monocytes	-0.47
Eosinophils	-2.61
Basophils	0.20
Neutrophil-Lymphocyte Ratio (NLR)	-0.02
HIV Status	-0.62
TB Status	-1.34
Intercept	**Values**
Negative|Mild	6.49
Mild|Moderate	8.17
Moderate|Severe	11.75

Table 4.57: Test Data Confusion Matrix for Ordinal Logistic Regression Model 3

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	3 470	923	363	14
Mild	120	233	95	4
Moderate	23	49	34	1
Severe	0	0	0	0

The class specific performance results further shows the performance of the model.The

Table 4.58: Results from Test Set of Ordinal Logistic Regression Model 3

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7013	(0.6888, 0.7135)	0.678	0.0001348	0.1971

sensitivity of the negative class is desirable but the model is unable to achieve that level of sensitivity for the remaining classes. The balanced accuracy of classes are 60.14% for the negative class, 57.01% for the mild class, 52.70% for the moderate class and 50% for the severe class since no predictions were made for the severe class.

Table 4.59: Class Results for Ordinal Logistic Regression Model 3

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9604	0.1934	0.0691	0
Specificity	0.2424	0.9469	0.9849	1
Balanced Accuracy	0.6014	0.5701	0.5270	0.5000

The overall multi-class AUC for Ordinal Logistic Regression Model Three is 63.01%. This is the worst AUC value for Dataset 3 across all models and Figure 4.47 shows the breakdown of the case versus case ROCs and AUC values. The figure shows that the best case performance belongs to the negative versus mild case (77.9%) whilst the worst performing case is the mild versus moderate case as it achieves an AUC below 50% (49.3).

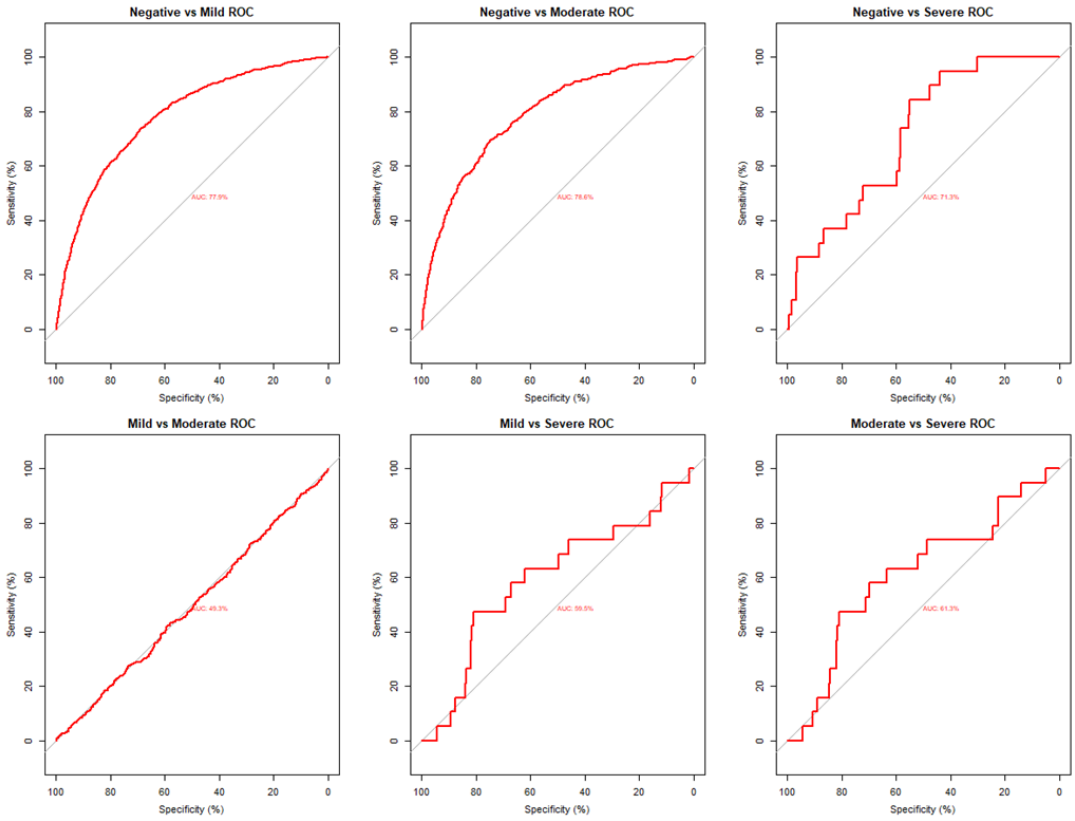

Figure 4.47: Receiver Operating Curves for Ordinal Logistic Regression Model 3

4.8.2.4 Ordinal Logistic Regression Model 4

Ordinal logistic regression model 4 was created using Dataset 4. The intercept values and significant variable coefficients are shown in Table 4.60. The intercept, Haematocrit and TB Status values have the greatest impact on the model and its performance. Model 4's performance is shown in Tables 4.61, 4.62 and 4.63. Table 4.61 shows that model 4, like model 2, mainly predicts negative responses. The model is able to predict all four classes, however it only predicts one severe response and the prediction is incorrect. The model achieves an accuracy of 73.75% which is less than the NIR value of 73.88%. This means that model 4 is not a significant model compared to a naive model that just predicts negative responses. This is the same outcome as with model 2.The model produces a poor Kappa score of 0.0143 which indicates no agreement between the model predictions and actual response variables of the dataset.

Table 4.60: Significant Coefficients for Ordinal Logistic Regression Model 4

Variable	**Coefficients**
Gender Female	0.42
Age	0.02
Urea	-0.01
Red Cell Count	0.77
Haematocrit	-3.76
Mean Cell Volume	0.01
Mean Cell Haemoglobin Concentration	0.08
HIV Status	-0.40
TB Status	-1.27
Intercept	**Values**
Negative|Mild	7.36
Mild|Moderate	8.82
Moderate|Severe	12.56

Table 4.61: Test Data Confusion Matrix for Ordinal Logistic Regression Model 4

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	9 839	2 288	1 108	37
Mild	27	28	10	1
Moderate	22	19	5	0
Severe	1	0	0	0

Table 4.62: Results from Test Set of Ordinal Logistic Regression Model 4

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7375	(0.73, 0.745)	0.7388	0.6352	0.0143

Table 4.63 shows the class specific performance results of model 4. The sensitivity

and specificity of the negative class is desirable but the model is unable to achieve that level of sensitivity or specificity for the remaining classes. The balanced accuracy of classes are poor for all classes. 50.65% for the negative class, 50.43% for the mild class, 50.06% for the moderate class and 50% for the severe class.

Table 4.63: Class Results for Ordinal Logistic Regression Model 4

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9949	0.0120	0.0045	0
Specificity	0.0180	0.9966	0.9967	1
Balanced Accuracy	0.5065	0.5043	0.5006	0.5000

The overall multi-class AUC for Ordinal Logistic Regression Model Four is 59.40%. Figure 4.48 shows once again with the ordinal logistic regression method that the worst performing case is the mild versus moderate case, it only achieves an AUC value of 55.4%. The negative versus mild class produces an AUC value of 72.5% which is the best of the six cases but is less than some of the previous cases across different datasets and models.

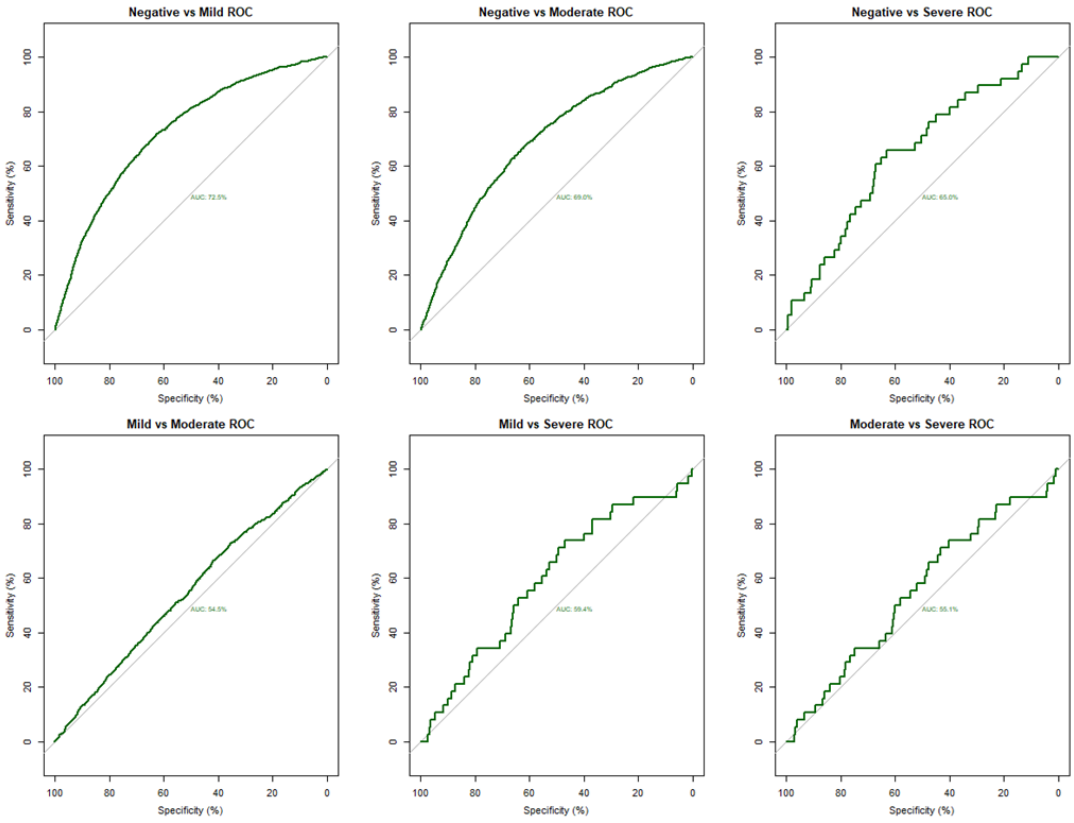

Figure 4.48: Receiver Operating Curves for Ordinal Logistic Regression Model 4

4.8.3 Baseline-category Logistic Regression

The Baseline-category logistic regression models were created in r-studio using the “vglm" function in the “VGAM" package.

4.8.3.1 Baseline-category Logistic Regression Model 1

Baseline-category logistic regression model 1 was created using Dataset 1. Model 1’s intercept values and significant variable coefficients are shown in Table 4.64 and the table indicates that the Haematocrit, Eosinophils, HIV Status, TB Status and intercept values have the greatest impact on the model. Model 1’s performance is shown in Table 4.65, 4.66 and 4.67. The confusion matrix for model 1 shows that model is able to predict all four classes, however it is unable to correctly predict the severe

class. The model achieves an accuracy of 79.24% and hence has an associated p-value of $1.476e^{-08}$ when testing against the NIR of 77.53%. This means that model is a significant model compared to a naive model that just predicts negative responses. The model produces a Kappa score of 0.2257 which indicates slight agreement between the model predictions and actual response variables of the dataset.

Table 4.64: Significant Coefficients for Baseline-category Logistic Regression Model 1

Variable	Mild Estimate	Moderate Estimate	Severe Estimate
Intercept	-7.62	-7.10	-6.76
Gender Female	0.51	0.29	0.36
Age	0.02	0.01	0.01
Albumin	-0.01	0.02	-0.03
Fibrinogen	0.25	0.08	0.32
Red Cell Count	0.92	0.20	0.53
Haematocrit	-5.49	1.48	-4.45
Mean Cell Haemoglobin Concentration	0.02	0.01	0.01
Neutrophils	-0.07	-0.05	0.01
Lymphocytes	-0.02	0.02	-0.01
Monocytes	0.05	-0.71	-0.18
Eosinophils	-0.99	-2.28	-0.57
Basophils	0.09	0.09	0.01
HIV Status	-0.66	-0.89	-0.47
TB Status	-1.55	-1.25	-0.85

Table 4.65: Test Data Confusion Matrix for Baseline-category Logistic Regression Model 1

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	13 726	2 112	1 076	35
Mild	374	675	137	5
Moderate	22	21	31	0
Severe	31	13	3	0

Table 4.66: Results from Test Set of Baseline-category Logistic Regression Model 1

Accuracy	95% CI	NIR	P-value	Kappa
0.7924	(0.7864, 0.7982)	0.7753	$1.476e^{-08}$	0.2257

The class specific performance results of model 1 are available in Table 4.67. The sensitivity and specificity of the negative class are desirable but the model is unable to achieve this level of sensitivity or specificity for the remaining classes. The balanced accuracy values show that the mild class prediction performance is the best with a value of 60.35%. The negative class performance is just behind with 59.22%. There is a substantial drop off to the last two classes with the moderate class achieving a balanced accuracy of 51.12% and the severe class having a balanced accuracy of 50%.

Table 4.67: Class Results for Baseline-category Logistic Regression Model 1

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9720	0.2404	0.0249	0
Specificity	0.2124	0.9665	0.9975	1
Balanced Accuracy	0.5922	0.6035	0.5112	0.5000

The overall multi-class AUC for Baseline-category Logistic Regression Model 1 is 75.43%. This is the best AUC achieved across all models and all datasets. Figure 4.49 breaks down the multi-class ROC down into its six cases. The figure shows the performance of five out of the six cases is good with the lowest of the five being an AUC value of 63.1% by the moderate versus severe case. However the figure does highlight the fact that the mild versus severe case performs significantly worse than the other cases. The mild versus severe case only manages an AUC of 52.1%.

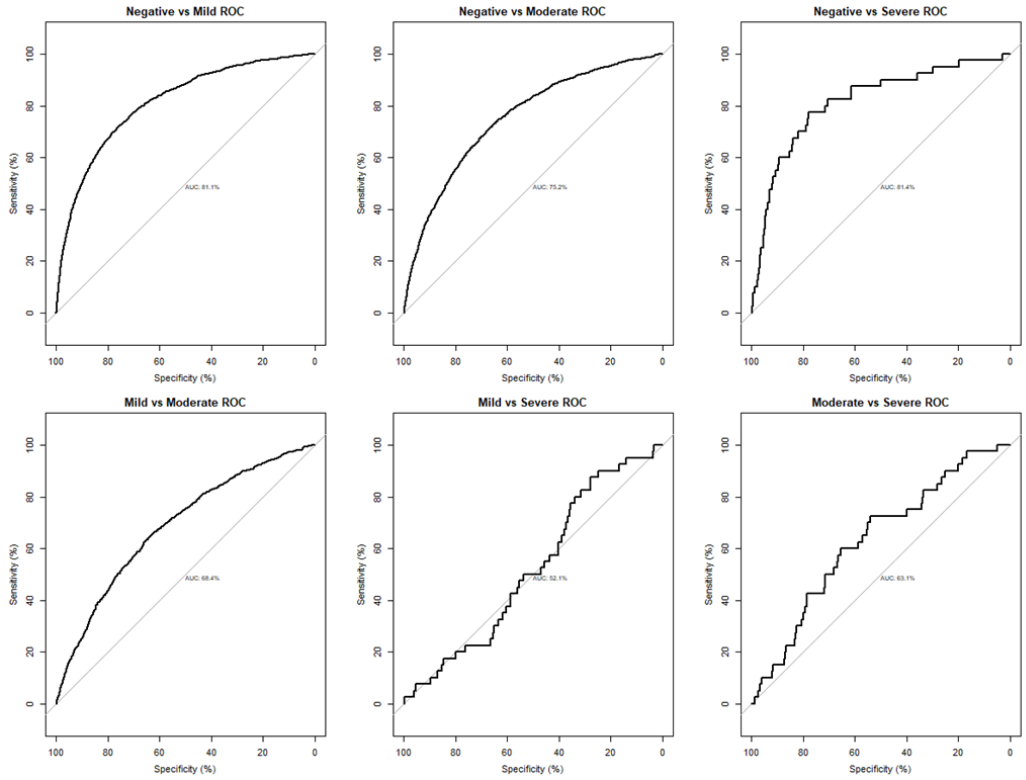

Figure 4.49: Receiver Operating Curves for Baseline-category Logistic Regression Model 1

4.8.3.2 Baseline-category Logistic Regression Model 2

Baseline-category logistic regression model 2 was created using Dataset 2. Model 2's intercept values and significant variable coefficients are shown in Table 4.68 and indicates that the Haematocrit, Red Cell Count, HIV Status, TB Status and intercept values have the greatest impact on the models performance. Model 2's performance is shown in Table 4.69, 4.70 and 4.71 and is similar to the previous models created using Dataset 2. The confusion matrix for model 2 shows that model is unable to predict all four classes, it can only predict negative and mild cases. The model achieves an accuracy of 77.31% which is worse than the naive model, this is shown by the NIR of 77.53%. The p-value found in Table 4.70 is hence greater than 5%. The model produces a Kappa score of 0.0253 which indicates no agreement between the model predictions and actual response variables of the dataset.

Table 4.68: Significant Coefficients for Baseline-category Logistic Regression Model 2

Variable	Mild Estimate	Moderate Estimate	Severe Estimate
Intercept	-8.05	-8.01	-8.51
Gender Female	0.34	0.34	0.29
Age	0.02	0.01	0.01
Red Cell Count	0.79	1.04	0.43
Haematocrit	-4.93	-8.54	-2.75
Mean Cell Volume	0.02	0.04	0.01
Mean Cell Haemoglobin Concentration	0.06	0.01	0.02
HIV Status	-0.65	-0.74	-0.32
TB Status	-1.48	-1.08	-0.32

Table 4.69: Test Data Confusion Matrix for Baseline-category Logistic Regression Model 2

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	13 997	2 7234	1 216	38
Mild	125	84	28	2
Moderate	0	0	0	0
Severe	0	0	0	0

Table 4.70: Results from Test Set of Baseline-category Logistic Regression Model 2

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7731	(0.7669, 0.7792)	0.7753	0.7696	0.0253

The class specific performance results further show the model performs poorly. The balanced accuracy of all four classes are less than 51% which is undesirable. The performance of baseline-category logistic regression model 2 is poor.

Table 4.71: Class Results for Baseline-category Logistic Regression Model 2

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9912	0.0299	0	0
Specificity	0.0279	0.9899	1	1
Balanced Accuracy	0.5095	0.5099	0.5000	0.5000

The overall multi-class AUC for Baseline-category Logistic Regression Model Two is 60.06%. Figure 4.50 shows that the baseline-category logistic regression model 2 produces similar results to the other models on Dataset 2. The performance of the first three cases are significantly better than the last three cases. the last three cases again produce UC values just greater than 50%. The last case is the worst performing case as it only achieves a AUC values of 50.4%.

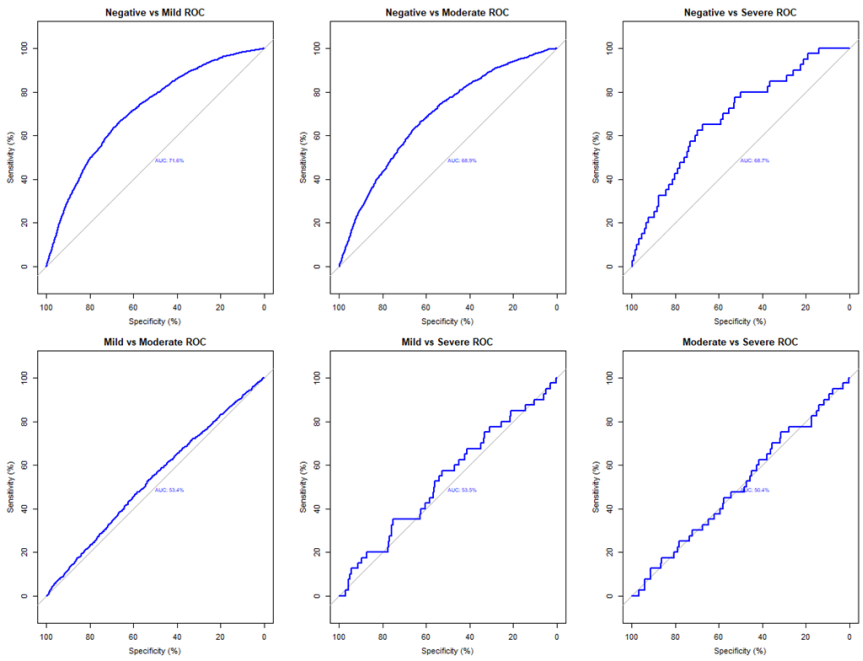

Figure 4.50: Receiver Operating Curves for Baseline-category Logistic Regression Model 2

4.8.3.3 Baseline-category Logistic Regression Model 3

Baseline-category logistic regression model 3 was created using Dataset 3 and its intercept values and significant variable coefficients are shown in Table 4.72 and indicates that the Haematocrit, Eosinophils, TB Status and intercept values have the greatest impact on the performance of the model. Model 3's performance is shown in Table 4.73, 4.74 and 4.75. The confusion matrix for model 3 shows that model is able to predict three of the four classes, however it is unable to predict the severe class. The model achieves an accuracy of 72.34% and achieves a p-value of $3.632e^{-13}$ when testing against the NIR of 67.8%. This means that model is a significant model compared to a naive model. The model produces a Kappa score of 0.3147 which indicates fair agreement between the model predictions and actual response variables of the dataset.

Table 4.72: Significant Coefficients for Baseline-category Logistic Regression Model 3

Variable	Mild Estimate	Moderate Estimate	Severe Estimate
Intercept	-8.48	-6.60	-8.70
Gender Female	0.58	0.28	0.34
Age	0.02	0.01	0.01
Urea	0.01	0.01	0.01
HbA1c	0.12	-0.09	0.05
International Normalized Ratio (INR)	-0.25	-0.27	-0.26
Haematocrit	9.40	1.24	3.90
Mean Cell Haemoglobin Concentration	0.14	0.13	0.09
Neutrophils	-0.04	-0.03	-0.01
Lymphocytes	-0.02	0.02	0.01
Monocytes	-0.17	-0.49	-0.15
Eosinophils	-2.65	-1.97	-0.82
Basophils	0.53	-0.19	-0.15
HIV Status	-0.46	-0.85	-0.44
TB Status	-1.61	-1.06	-0.73

Table 4.73: Test Data Confusion Matrix for Baseline-category Logistic Regression Model 3

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	3 330	707	357	14
Mild	241	463	73	3
Moderate	42	35	62	2
Severe	0	0	0	0

Table 4.74: Results from Test Set of Baseline-category Logistic Regression Model 3

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7234	(0.7112, 0.7354)	0.678	$3.632e^{-13}$	0.3147

The class specific performance results of model 3 are shown in Table 4.67. The sensitivity and specificity of the negative class are desirable but the sensitivity and specificity for the remaining classes decrease and increase respectively to less than desirable values. The balanced accuracy values show that the mild class prediction performance is the best with a value of 65.37%. The negative class performance is just behind with 64.67%. There is a substantial drop off to the last two classes with the moderate class achieving a balanced accuracy of 55.48% and the severe class having a balanced accuracy of 50% since no predictions were made.

Table 4.75: Class Results for Baseline-category Logistic Regression Model 3

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9217	0.3842	0.1260	0
Specificity	0.3718	0.9231	0.9837	1
Balanced Accuracy	0.6467	0.6537	0.5548	0.5000

The overall multi-class AUC for Baseline-category Logistic Regression Model Three is 69.97%. Figure 4.51 breaks down the case by case ROC and AUC performances of the model. The negative versus mild case performs the best with an AUC of 80% whilst the moderate versus severe case performs the worst with an AUC of 57.4%. It is worth noting the performance of the mild versus moderate case (AUC of 71.5%) and the mild versus severe case (AUC of 77.5%) these are significant performances.

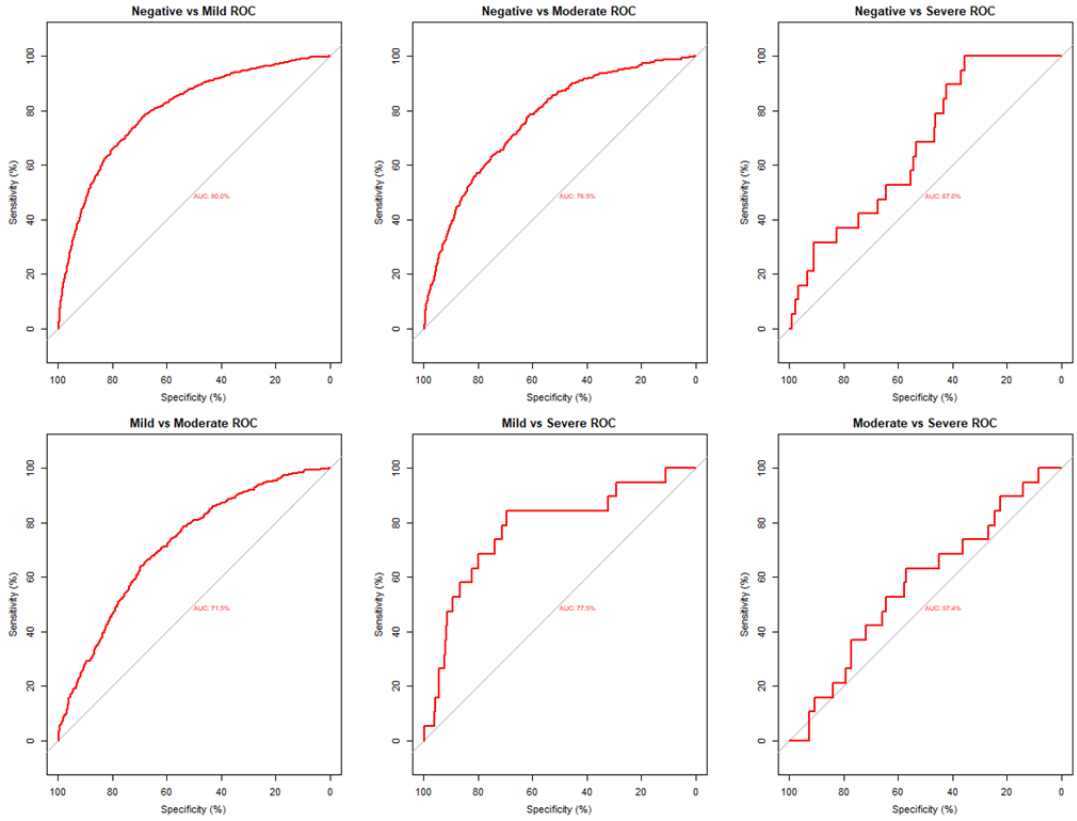

Figure 4.51: Receiver Operating Curves for Baseline-category Logistic Regression Model 3

4.8.3.4 Baseline-category Logistic Regression Model 4

Dataset 4 was used to create Baseline-category logistic regression model 4. Model 4's intercept values and significant variable coefficients are shown in Table 4.72 and indicates that the Haematocrit, Red Cell Count, TB Status and intercept values have the

greatest impact on the model and its performance. Model 4's performance is shown in Table 4.77, 4.78 and 4.79. The confusion matrix for model 4 shows that model is only able to predict the negative and mild classes, it fails to predict any moderate and severe cases. The model achieves an accuracy of 73.62% which is less than the NIR of 73.88%. This means that model is not a significant model compared to a naive model that just predicts negative responses. This is confirmed by the p-value being greater than 5%. The model produces a Kappa score of 0.0515 which indicates no agreement between the model predictions and actual response variables of the dataset.

Table 4.76: Significant Coefficients for Baseline-category Logistic Regression Model 4

Variable	Mild Estimate	Moderate Estimate	Severe Estimate
Intercept	-8.06	-8.57	-12.37
Gender Female	0.44	0.47	0.01
Red Cell Count	0.71	0.91	0.78
Haematocrit	-2.27	-6.07	-7.49
Mean Cell Haemoglobin Concentration	0.03	0.09	0.05
HIV Status	-0.37	-0.49	-0.16
TB Status	-1.58	-0.87	-1.45

Table 4.77: Test Data Confusion Matrix for Baseline-category Logistic Regression Model 4

	Actual			
Predicted	Negative	Mild	Moderate	Severe
Negative	9 700	2 181	1 068	34
Mild	189	154	55	4
Moderate	0	0	0	0
Severe	0	0	0	0

Table 4.78: Results from Test Set of Baseline-category Logistic Regression Model 4

Accuracy	**95% CI**	**NIR**	**P-value**	**Kappa**
0.7362	(0.7286, 0.7436)	0.7388	0.7578	0.0515

The class specific performance results further show the model 4's poor performance. The balanced accuracy of the negative class is 52.09% and 52.18% for the mild class. The last two classes both have balanced accuracy values of 50% since no predictions were made by the model for those classes.

Table 4.79: Class Results for Baseline-category Logistic Regression Model 4

	Class			
Statistics	Negative	Mild	Moderate	Severe
Sensitivity	0.9809	0.0660	0	0
Specificity	0.0609	0.9776	1	1
Balanced Accuracy	0.5209	0.5218	0.5000	0.5000

The overall multi-class AUC for Baseline-category Logistic Regression Model Four is 60.98%. This only slightly better than baseline-category logistic regression model 2 and Figure 4.52 shows the case by case ROC and AUC performance. The best performance is the negative versus mild case with a AUC of 72.5%. The worst performance is a close mater, with the moderate versus severe case the worst with an AUC of 54% but the second worst is the mild versus moderate case with an AUC of 54.8%.

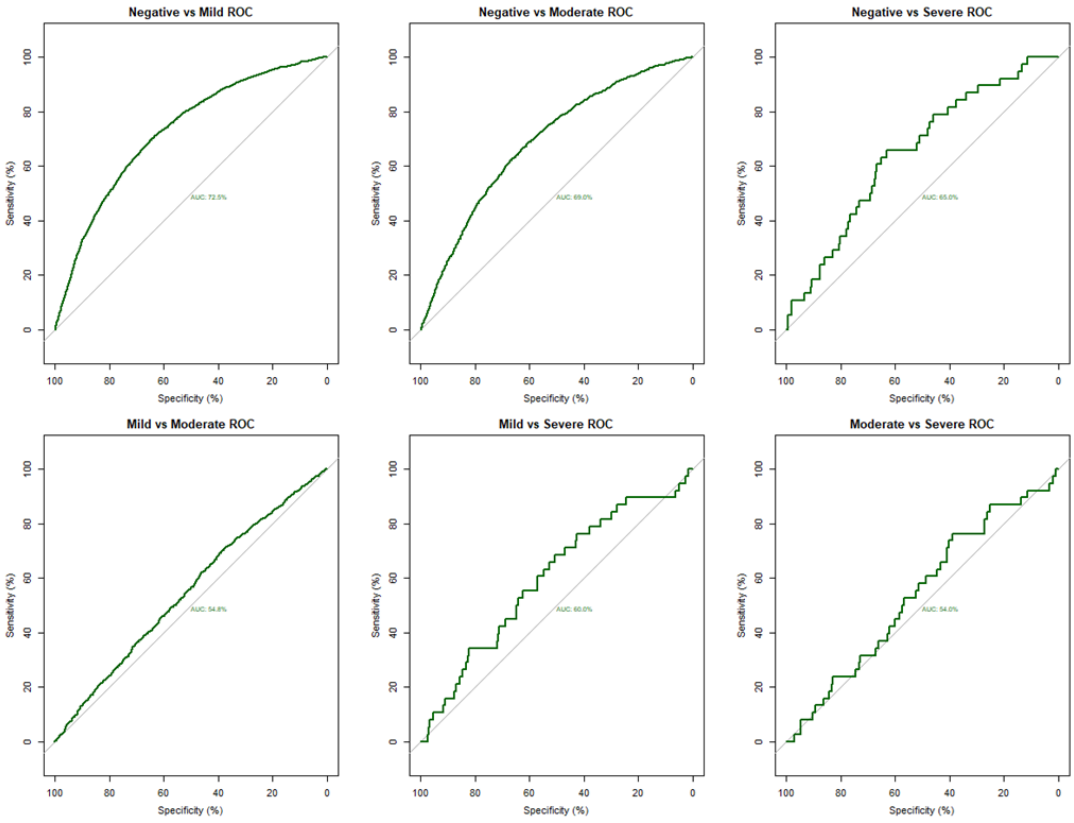

Figure 4.52: Receiver Operating Curves for Baseline-category Logistic Regression Model 4

Chapter 5

Summary and Discussion

The results and interpretations of Chapter 4 are summarized and discussed in this section in order to review and understand the performance of each model. A conclusion on the models and this study performance can then be made.

Analysis of the models and network performances was done by looking at the respective dataset performances of each model and network and then and overall performance review across all datasets. Tables 5.1, 5.2, 5.3 and 5.4 show the summarised performance of each model/network for each dataset.

Table 5.1 shows that for Dataset 1 the baseline-category logistic regression model produced the best AUC, whilst the random forest produced the best accuracy. However the self-normalising neural network produced the best Kappa score and the best mean balanced accuracy but it does also produce the second worst AUC value. The best model for Dataset 1 is the self-normalising neural network with the baseline-category logistic regression model being the second best model.

Table 5.1: Performance summary of all methods used for Dataset 1

	Methods				
Measures	RF	SNN	MLR	OLR	BLR
AUC	0.7242	0.6756	0.6850	-	0.7543
Accuracy	0.7993	0.7743	0.7729	-	0.7924
Kappa Score	0.2043	0.3385	0.2106	-	0.2257
Mean Balanced Accuracy	0.5432	0.6085	0.5506	-	0.5517

Table 5.2 reveals that the self-normalising neural network is the best model as it produces the best AUC value, Kappa score and mean balanced accuracy value. The best accuracy value was produced by the random forest model. The table also shows the logistic regression models struggled with Dataset 2.

Table 5.2: Performance summary of all methods used for Dataset 2

	Methods				
Measures	RF	SNN	MLR	OLR	BLR
AUC	0.6079	0.6248	0.6011	0.5987	0.6006
Accuracy	0.7810	0.7743	0.7519	0.7744	0.7731
Kappa Score	0.1332	0.1960	0.0257	0.0048	0.0253
Mean Balanced Accuracy	0.5276	0.5466	0.5049	0.5007	0.5049

Looking and Tables 5.3 and 5.4 show that the performance od the models for the two datasets are the same in that for both datasets the random forest model produces the best AUC and Accuracy values, whilst the self-normalising neural network produces the best Kappa score and mean balanced accuracy value. For both of the datasets either model could be considered the best, however to determine a best model the performance of the two models in the respective “not best" measures needs to be evaluated.

Therefore, evaluating the performance of the self-normalising neural network in terms of its AUC and accuracy and the random forest in terms of its Kappa score and mean balanced accuracy, reviles that the self-normalising neural network performs slightly better for both datasets.

Table 5.3: Performance summary of all methods used for Dataset 3

	Methods				
Measures	RF	SNN	MLR	OLR	BLR
AUC	0.7158	0.6942	0.6954	0.6301	0.6997
Accuracy	0.7450	0.7178	0.7123	0.7013	0.7234
Kappa Score	0.3306	0.3719	0.3339	0.1971	0.3147
Mean Balanced Accuracy	0.5871	0.6231	0.6009	0.5496	0.5888

Table 5.4: Performance summary of all methods used for Dataset 4

	Methods				
Measures	RF	SNN	MLR	OLR	BLR
AUC	0.6304	0.6272	0.6097	0.5940	0.6098
Accuracy	0.7551	0.7247	0.7362	0.7375	0.7362
Kappa Score	0.2027	0.2576	0.0515	0.0143	0.0515
Mean Balanced Accuracy	0.5462	0.5756	0.5107	0.5029	0.5107

Comparing the performances of the models and networks across the datasets is done by looking at the respective AUC values, Kappa scores and mean balanced accuracy values. The Accuracy value is neglected because it is affected by the structure and make up of each dataset. Across all four datasets the machine learning methods outperform the logistic regression methods in every measure bar the AUC value for Dataset

1 where the baseline-category model performed the best. This also happens to be the highest achieved AUC value (75.43%) for all the models across all the datasets. The highest achieved Kappa score (0.3719) belongs to the self-normalising neural network of Dataset 3. 62.31% was the highest achieved mean balanced accuracy value and it was achieved by the self-normalising neural network of Dataset 3. Overall the models and networks produced the best values and scores for Dataset 3 and the worst values and scores for Dataset 2.

Chapter 6

Conclusion, Limitations and Recommendations

6.1 Conclusion

The results show that the machine learning techniques outperform the logistic regression methods in status and severity prediction of South African Covid-19 laboratory test data and that the best performing machine learning technique was the self-normalising neural network.Overall the models and networks performed the best when using Dataset 3. The results provide evidence that the machine learning techniques can be used as an indicative tool for Covid-19 status and severity prediction rather than a confirmation tool.

6.2 Limitations

The dataset used in this study is from an early period of the Covid-19 pandemic, 1 March 2020 to 7 July 2020, this was at a time when information about the virus was still limited. Now there is more information and data available surrounding Covid-19 and specifically the best laboratory test to conduct and which early indicators to look for. Hence important laboratory test that were conducted later on would not be in-

cluded in this dataset.

The original dataset obtained from the NHLS had a large number of missing values. This meant in order to carry out analysis on the data imputation had to be used. This can introduces statistical bias into the data and it is because of this fact that four different dataset were created to minimises the effects of the statistical bias introduced through imputation.

6.3 Recommendations

6.3.1 Updated Dataset

Carry out a new study using the same/similar methods but using a newer/updated dataset with better early indicator tests and more complete data. Then contrast the results with this study and see if the models are able to produce better or similar results.

6.3.2 Other Viruses and Infectious Diseases

The logistic regression and machine learning methods could also be used to predict the status and severity of other viruses and infectious diseases by following the same/similar processes and procedures as conducted in this study. This would require obtaining laboratory test results of the patients who were tested for other virus or diseases. The results could then be compared to this study and conclusions about which virus/disease do the methods perform best on.

Bibliography

Agresti, A. (2003). *Categorical data analysis*. John Wiley and Sons. 23

Ahamad, M. M., Aktar, S., Rashed-Al-Mahfuz, M., Uddin, S., Liò, P., Xu, H., Summers, M. A., Quinn, J. M., and Moni, M. A. (2020). A machine learning model to identify early stage symptoms of sars-cov-2 infected patients. *Expert Systems With Applications*, 160:113661. 19

Aktar, S., Ahamad, M. M., Rashed-Al-Mahfuz, M., Azad, A., Uddin, S., Kamal, A., Alyami, S. A., Lin, P.-I., Islam, S. M. S., Quinn, J. M., et al. (2021). Machine learning approach to predicting covid-19 disease severity based on clinical blood test data: statistical analysis and model development. *JMIR Medical Informatics*, 9(4):e25884. 15, 19

Al-Hindawi, A., Abdulaal, A., Rawson, T. M., Alqahtani, S. A., Mughal, N., and Moore, L. S. (2021). Covid-19 prognostic models: A pro-con debate for machine learning vs. traditional statistics. *Frontiers in digital health*, 3. 18, 19

Auerswald, M. and Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. *Psychological methods*, 24(4):468. 62

Becker, J.-M., Ringle, C. M., Sarstedt, M., and Völckner, F. (2015). How collinearity affects mixture regression results. *Marketing Letters*, 26(4):643–659. 53

Beheshti, N. (2022). Guide to confusion matrices and classification performance metrics. 28, 29

Bender, R. and Grouven, U. (1997). Ordinal logistic regression in medical research. *Journal of the Royal College of Physicians of London*, 31(5):546. 24

Bre, F., Gimenez, J. M., and Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158:1429–1441. 5, 25

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32. 26, 27

Brophy, C., Connolly, J., Fagerli, I., Duodu, S., and Svenning, M. M. (2011). A baseline category logit model for assessing competing strains of rhizobium bacteria. *Journal of Agricultural, Biological, and Environmental Statistics*, 16(3):409–421. 23

Bruin, J. (2011). newtest: command to compute new test @ONLINE. 113

Chow, M. (2022). Baseline-category logit model: Stat 504. Technical report, Pennsylvania State University. 23

Dangeti, P. (2017). *Statistics for machine learning*. Packt Publishing Ltd. 9

Daoud, J. I. (2017). Multicollinearity and regression analysis. In *Journal of Physics: Conference Series*, volume 949, page 12009. IOP Publishing. 20

Degenhardt, F., Seifert, S., and Szymczak, S. (2019). Evaluation of variable selection methods for random forests and omics data sets. *Briefings in bioinformatics*, 20(2):492–503. 21

DiGangi, E. A. and Hefner, J. T. (2013). Ancestry estimation. *Research methods in human skeletal biology*, pages 117–149. 22

Dziuban, C. D. and Shirkey, E. C. (1974). When is a correlation matrix appropriate for factor analysis? some decision rules. *Psychological bulletin*, 81(6):358. 62

Erkan, A. and Yildiz, Z. (2014). Parallel lines assumption in ordinal logistic regression and analysis approaches. *International Interdisciplinary Journal of Scientific Research*, 1(3):8–23. 22, 23

Ghedira, H. and Bernier, M. (2004). The effect of some internal neural network parameters on sar texture classification performance. In *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, volume 6, pages 3845–3848. IEEE. 5, 25

Gorman, B. (2018). *mltools: Machine Learning Tools*. R package version 0.3.5. 53

Harapan, H., Itoh, N., Yufika, A., Winardi, W., Keam, S., Te, H., Megawati, D., Hayati, Z., Wagner, A. L., and Mudatsir, M. (2020). Coronavirus disease 2019 (covid-19): A literature review. *Journal of Infection and Public Health*, 13(5):667–673. 10

Hesse, R., van der Westhuizen, D., and George, J. (2020). Covid-19 related laboratory analyte changes and the relationship between sars-cov-2 and hiv, tb and hba1c in south africa. *Clinical, Biological and Molecular Aspects of COVID-19*. 19, 34

Hill, B. D. (2011). *The sequential Kaiser-Meyer-Olkin procedure as an alternative for determining the number of factors in common-factor analysis: A Monte Carlo simulation*. Oklahoma State University. 62

Huang, Y. (2017). Multicategory logit models. Technical report, University of Chicago. STAT226 Winter 2017 Chapter 6. 23

Ij, H. (2018). Statistics versus machine learning. *Nat Methods*, 15(4):233. 9

Kaiser, H. F. (1992). On cliff's formula, the kaiser-guttman rule, and the number of factors. *Perceptual and Motor Skills*, 74(2):595–598. 62

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in neural information processing systems*, 30. 36

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., Team, R. C., et al. (2020). Package 'caret'. *The R Journal*, 223. 36

Kursa, M. B., Jankowski, A., and Rudnicki, W. R. (2010). Boruta–a system for feature selection. *Fundamenta Informaticae*, 101(4):271–285. 21, 73

Kurt, I., Ture, M., and Kurum, A. T. (2008). Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease. *Expert systems with applications*, 34(1):366–374. 17

Louppe, G. (2014). Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*. 26

Mbuvha, R. (2020). Multilayer perceptrons. Technical report, University of the Witwatersrand. STAT3033 Notes. 26

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282. 9, 30

Mellor, A., Haywood, A., Stone, C., and Jones, S. (2013). The performance of random forests in an operational setting for large area sclerophyll forest classification. *Remote Sensing*, 5(6):2838–2856. 30, 31

Mohammadi, F., Pourzamani, H., Karimi, H., Mohammadi, M., Mohammadi, M., Ardalan, N., Khoshravesh, R., Pooresmaeil, H., Shahabi, S., Sabahi, M., et al. (2021). Artificial neural network and logistic regression modelling to characterize covid-19 infected patients in local areas of iran. *Biomedical journal*, 44(3):304–316. 17, 18

Naidu, T. (2020). The covid-19 pandemic in south africa. *Psychological Trauma: Theory, Research, Practice, and Policy*, 12(5):559. 9

Nasejje, J. (2020). Ensemble methods and random forests. Technical report, University of the Witwatersrand. STAT3033 Notes. 26, 27

NaserGhandi, A., Allameh, S. F., and Saffarpour, R. (2020). All about covid-19 in brief. *New Microbes and New Infections*, 35. 9, 10

Nguyen, Q. H., Ly, H.-B., Ho, L. S., Al-Ansari, N., Le, H. V., Tran, V. Q., Prakash, I., and Pham, B. T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, 2021. 33, 79

Nick, T. G. and Campbell, K. M. (2007). Logistic regression. *Topics in biostatistics*, pages 273–301. 22

Nusinovici, S., Tham, Y. C., Yan, M. Y. C., Ting, D. S. W., Li, J., Sabanayagam, C., Wong, T. Y., and Cheng, C.-Y. (2020). Logistic regression was as good as machine learning for predicting major chronic diseases. *Journal of clinical epidemiology*, 122:56–69. 17

Parry, S. (2016). Ordinal logistic regression models and statistical software: What you need to know. *Cornell University Statistical Consulting Unit*. 23, 24

Pinter, G., Felde, I., Mosavi, A., Ghamisi, P., and Gloaguen, R. (2020). Covid-19 pandemic prediction for hungary; a hybrid machine learning approach. *Mathematics*, 8(6):890. 14, 15

Pokhrel, S. and Chhetri, R. (2021). A literature review on impact of covid-19 pandemic on teaching and learning. *Higher Education for the Future*, 8(1):133–141. 9

Pourhoseingholi, M. A., Baghestani, A. R., and Vahedi, M. (2012). How to control confounding effects by statistical analysis. *Gastroenterology and Hepatology from Bed to Bench*, 5(2):79. 21

Prysby, C., Scavo, C., and Association, A. P. S. (2013). *SETUPS: Voting Behavior: The 2012 Election*. Inter-university Consortium for Political and Social Research. 21

Quinn, R. (2020). Common evaluation measures for classification models. 28, 29, 30

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 54

RDocumentation (2020). Vif: Variance inflation factor. Technical report, DataCamp. 21

Rekkas, M. (2011). Chapter 8: Multicollinearity. Technical report, Simon Fraser University. Department of Economics. 20

Revelle, W. (2022). *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois. R package version 2.2.9. 62

Senthilnathan, S. (2019). Usefulness of correlation analysis. *Available at SSRN 3416918*. 20

Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, 6(12):310–316. 24

Shrestha, N. (2021). Factor analysis as a tool for survey analysis. *American Journal of Applied Mathematics and Statistics*, 9(1):4–11. 62, 66

Starkweather, J. and Moske, A. K. (2011). Multinomial logistic regression, 2011. Technical report, University Information Technology. 22

van Buuren, S., Groothuis-Oudshoorn, K., Robitzsch, A., Vink, G., Doove, L., and Jolani, S. (2015). Package 'mice'. *Computer software*. 20

Walczak, S. (2019). Artificial neural networks. In *Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction*, pages 40–53. IGI global. 25

Wang, Q., Yu, S., Qi, X., Hu, Y., Zheng, W., Shi, J., and Yao, H. (2019). Overview of logistic regression model analysis and application. *Zhonghua yu fang yi xue za zhi [Chinese journal of preventive medicine]*, 53(9):955–960. 22

Wang, S.-C. (2003). Artificial neural network. In *Interdisciplinary computing in java programming*, pages 81–100. Springer. 24

Wibowo, F. et al. (2021). Prediction modelling of covid-19 outbreak in indonesia using a logistic regression model. In *Journal of Physics: Conference Series*, volume 1803, page 12015. IOP Publishing. 15, 16

Wu, J., Zhang, P., Zhang, L., Meng, W., Li, J., Tong, C., Li, Y., Cai, J., Yang, Z., Zhu, J., et al. (2020). Rapid and accurate identification of covid-19 infection through machine learning based on clinical available blood test results. *MedRxiv*. 19

Wu, Y.-c. and Feng, J.-w. (2018). Development and application of artificial neural network. *Wireless Personal Communications*, 102(2):1645–1656. 24

Xu, K., Zhou, M., Yang, D., Ling, Y., Liu, K., Bai, T., Cheng, Z., and Li, J. (2020). Application of ordinal logistic regression analysis to identify the determinants of illness severity of covid-19 in china. *Epidemiology and Infection*, 148. 16

Yang, H. S., Hou, Y., Vasovic, L. V., Steel, P. A., Chadburn, A., Racine-Brzostek, S. E., Velu, P., Cushing, M. M., Loda, M., Kaushal, R., et al. (2020). Routine laboratory blood tests predict sars-cov-2 infection using machine learning. *Clinical chemistry*, 66(11):1396–1404. 10, 11, 13, 14, 35, 74

Yang, S. and Kim, J. K. (2020). Asymptotic theory and inference of predictive mean matching imputation using a superpopulation model framework. *Scandinavian Journal of Statistics*, 47(3):839–861. 20

Yedida, R. (2018). Evaluating classification models. 5, 29, 30

Appendix A

R Code

The R code for the study is found in the link below:

https://drive.google.com/drive/folders/1mTOCQsfyNoCVAgfBn-CYbH9OzDfaXnDe?usp=share_link

Appendix B

Datasets

```
'data.frame':   61330 obs. of  81 variables:
 $ gender                          : chr  "F" "F" "F" "F" ...
 $ age                             : int  54 79 46 57 41 29 67 34 49 86 ...
 $ patientid                       : int  264324 264328 264331 264336 264343 264371 264372 264375 2
 $ nlr                             : num  NA NA NA NA NA NA NA NA NA NA ...
 $ tbgenexpert                     : chr  "." "." "." "." "." ...
 $ hivelisaencoded                 : chr  "." "." "." "." "." ...
 $ amnttestresults                 : int  3 12 2 28 7 10 5 1 22 6 ...
 $ nvals                           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ ward_present                    : int  0 0 1 0 1 1 1 0 0 0 ...
 $ icu_present                     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ high_present                    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ covid_present                   : int  NA NA 0 NA 0 0 0 NA NA NA ...
 $ intensive_present               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ critical_present                : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hba1c_cat                       : chr  "" "" "" "" "" ...
 $ registereddatecov2pcr           : chr  "18 May 2020" "27 May 2020" "13 Jun 2020" "03 Jun 2020" .
 $ registereddatebiochem           : chr  "16 May 2020" "27 May 2020" "10 Jun 2020" "04 Jun 2020" .
 $ datedif                         : int  -2 0 -3 1 0 -2 0 -1 8 1 ...
 $ viralloadtextencoded            : chr  "." "." "." "." "." ...
 $ interleukin6_new                : num  NA NA NA NA NA NA NA NA NA NA ...
 $ IL6combined                     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ cov2pcrnew                      : chr  "Negative" "Negative" "Negative" "Positive" ...
 $ care_present                    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ severecovidvalue                : chr  "Non-severe SARS-COV2" "Non-severe SARS-COV2" "Non-severe
 $ creatinineplusegfrf             : num  3 33 54 55 78 ...
 $ ureaf                           : num  42.4 5.5 4.1 3.7 6.45 ...
 $ albuminf                        : num  NA NA NA 33 NA 30 NA NA NA 50 ...
 $ totalbilirubinf                 : num  NA NA NA 12.5 10 8 NA NA NA 14 ...
 $ alaninetransaminasealtf         : num  NA NA NA 18.5 NA 52 NA NA 37.5 27 ...
 $ aspartatetransaminaseastf       : num  NA NA NA 32 NA 35 NA NA NA 37 ...
 $ gammaglutamyltransferaseggtf    : num  NA NA NA 24.5 NA 41 NA NA NA 47 ...
 $ lactatedehydrogenaseldf         : num  NA NA NA 430 NA 441 NA NA NA NA ...
 $ creatinekinasembckmbf           : num  NA NA NA NA NA NA NA NA NA NA ...
 $ troponinihighsensitivityf       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ troponinthighsensitivityf       : num  NA NA NA NA NA NA NA NA NA 36 ...
 $ ntprobnpf                       : num  NA NA NA NA NA ...
 $ procalcitoninrapidscreenf       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ procalcitoninsensitivef         : num  NA NA NA 11.9 NA ...
 $ crpf                            : num  NA NA NA NA 79.5 301 NA NA 51 14 ...
 $ cortisolf                       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ hba1cf                          : num  NA NA NA 13.8 11.6 NA NA NA NA NA ...
 $ glucoserandomf                  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ glucosefastingf                 : logi  NA NA NA NA NA NA ...
 $ ferritinf                       : num  NA NA NA 289 NA NA NA NA NA NA ...
 $ myoglobinf                      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ erythrocytesedimentationratef   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ intnormalisedratioinrf          : num  NA 1.03 NA 1.19 NA ...
 $ fibrinogenf                     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ ddimerf                         : num  NA 2.52 NA 0.34 2.77 ...
 $ redcellcountf                   : num  2.5 4.54 4.95 4.8 4.51 ...
 $ haemoglobinf                    : num  6.9 13.3 8.7 11.9 13.2 ...
 $ haematocritf                    : num  0.211 0.412 0.396 0.379 0.39 ...
 $ meancellvolumef                 : num  84.4 90.7 80 79 86.5 ...
 $ meancellhaemoglobinf            : num  27.6 29.3 17.6 24.8 29.3 ...
 $ meancellhaemoglobinconcentratf: num  32.7 32.3 22 31.4 33.8 ...
 $ plateletcountf                  : num  254 217 493 178 212 ...
 $ neutrophilsf                    : num  NA NA NA 11.5 NA ...
 $ lymphocytesf                    : num  NA NA NA 1.32 NA ...
 $ monocytesf                      : num  NA NA NA 0.25 NA ...
 $ eosinophilsf                    : num  NA NA NA 0 NA ...
 $ basophilsf                      : num  NA NA NA 0.01 NA ...
 $ immaturecellsf                  : num  NA NA NA 0.01 NA ...
 $ nlrf                            : num  NA NA NA 8.72 NA ...
 $ IL6combinedf                    : num  NA NA NA NA NA NA NA NA NA NA ...
 $ hba1cbypatient                  : num  NA NA NA 13.8 11.6 NA NA NA NA NA ...
 $ cd4numnew                       : num  NA NA NA NA NA NA NA NA 59.5 NA ...
 $ cd4cat                          : chr  "" "" "" "" "" ...
 $ cd4cat_3                        : chr  "" "" "" "" "" ...
 $ hivstatusbyelisa                : int  0 0 0 0 0 0 0 0 0 0 ...
 $ viralloadtextstatus             : int  0 0 0 0 0 1 0 0 0 0 ...
 $ hivviralloadstatus              : int  0 0 0 0 0 0 0 0 1 0 ...
 $ cd4status                       : int  0 0 0 0 0 0 0 0 2 0 ...
 $ hivstatusvalue                  : int  0 0 0 0 0 1 0 0 3 0 ...
 $ hivstatus                       : chr  "HIV Negative or Unknown" "HIV Negative or Unknown" "HIV
 $ tbstatus                        : chr  "GeneXpert Negative or Unknown" "GeneXpert Negative or Ur
e or Unknown" ...
 $ hba1c_cat_perpatient            : chr  "" "" "" "HBA1C >/= 10" ...
 $ hba1c_controlled                : chr  "" "" "" "HbA1c >=6.5" ...
 $ viralloadunsuppressedtextvalue: int  0 0 0 0 0 0 0 0 0 0 ...
 $ viralloadunsuppressednumvalue : num  NA NA NA NA NA ...
 $ viralloadunsuppressedvalue    : chr  "" "" "" "" ...
 $ cd4cat_350_700                : chr  "" "" "" "" ...
```

Figure B.1: Dataset after filtering for a 3 week test data period

```
Classes 'data.table' and 'data.frame':  60711 obs. of  38 variables:
 $ gender_F                        : int  1 1 1 1 0 1 0 0 0 1 ...
 $ age                             : num  54 79 46 57 41 29 67 34 49 86 ...
 $ creatinineplusegfrf             : num  3 33 54 55 78 ...
 $ ureaf                           : num  42.4 5.5 4.1 3.7 6.45 ...
 $ albuminf                        : num  25 39 35 33 32 ...
 $ totalbilirubinf                 : num  12 6 27 12.5 10 8 8 23 4 14 ...
 $ alaninetransaminasealtf         : num  11 32 21 18.5 22.5 52 12 19 37.5 27 ...
 $ aspartatetransaminaseastf       : num  88 13 42 32 103 35 24 44 25 37 ...
 $ gammaglutamyltransferaseggtf    : num  31 37 82 24.5 16 41 31 21 43 47 ...
 $ lactatedehydrogenaseldf         : num  288 498 830 430 300 ...
 $ creatinekinasembckmbf           : num  148 58 58 13 36 2 23 8 28 0 ...
 $ troponinthighsensitivityf       : num  25 355 13 17 9.25 11 26.5 24.5 55 36 ...
 $ ntprobnpf                       : num  2300 860 289 5048 113 ...
 $ procalcitoninsensitivef         : num  0.416 0.285 1.383 11.94 0.04 ...
 $ crpf                            : num  24 122 10 69 79.5 301 69 96 51 14 ...
 $ cortisolf                       : num  1750 428 953 794 578 529 195 1750 437 47 ...
 $ hba1cf                          : num  7.2 6 6.6 13.8 11.6 ...
 $ glucoserandomf                  : num  6.7 6.7 4.3 19.8 7.8 ...
 $ ferritinf                       : num  17 33 110 289 564 ...
 $ erythrocytesedimentationratef   : num  62 6 62.5 35 25 84 122 20 126 20 ...
 $ intnormalisedratioinrf          : num  1.09 1.03 1.09 1.19 1.29 ...
 $ fibrinogenf                     : num  3 1.5 3.5 5.7 6.9 ...
 $ ddimerf                         : num  17.6 2.52 9.7 0.34 2.77 ...
 $ redcellcountf                   : num  2.5 4.54 4.95 4.8 4.51 ...
 $ haematocritf                    : num  0.211 0.412 0.396 0.379 0.39 ...
 $ meancellvolumef                 : num  84.4 90.7 80 79 86.5 ...
 $ meancellhaemoglobinconcentratf: num  32.7 32.3 22 31.4 33.8 ...
 $ plateletcountf                  : num  254 217 493 178 212 ...
 $ neutrophilsf                    : num  10.77 7.72 10.03 11.51 6.66 ...
 $ lymphocytesf                    : num  1.49 1.76 2.56 1.32 2.4 ...
 $ monocytesf                      : num  0.79 0.71 0.42 0.25 0.183 ...
 $ eosinophilsf                    : num  0.09 0.72 0.015 0 0.08 ...
 $ basophilsf                      : num  0.01 0.0686 0.01 0.01 0.06 ...
 $ nlrf                            : num  11.94 6.33 4.62 8.72 3.78 ...
 $ cd4numnew                       : num  12 330 623 3 674 ...
 $ hivstatus_HIV_Positive          : int  0 0 0 0 0 1 0 0 1 0 ...
 $ tbstatus_Active_TB              : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Covid_Cat                       : Factor w/ 4 levels "Negative","Mild",..: 1 1 1 2 3 3 3
```

Figure B.2: Dataset 1 after all cleaning and preparation has been complete

```
Classes 'data.table' and 'data.frame':  60711 obs. of  14 variables:
 $ gender_F                        : int  1 1 1 1 0 1 0 0 0 1 ...
 $ age                             : num  54 79 46 57 41 29 67 34 49 86 ...
 $ creatinineplusegfrf             : num  3 33 54 55 78 ...
 $ ureaf                           : num  42.4 5.5 4.1 3.7 6.45 ...
 $ alaninetransaminasealtf         : num  287 20 61 18.5 64 52 31 15 37.5 27 ...
 $ crpf                            : num  54.2 1 191 68 79.5 ...
 $ redcellcountf                   : num  2.5 4.54 4.95 4.8 4.51 ...
 $ haematocritf                    : num  0.211 0.412 0.396 0.379 0.39 ...
 $ meancellvolumef                 : num  84.4 90.7 80 79 86.5 ...
 $ meancellhaemoglobinconcentratf: num  32.7 32.3 22 31.4 33.8 ...
 $ plateletcountf                  : num  254 217 493 178 212 ...
 $ hivstatus_HIV_Positive          : int  0 0 0 0 0 1 0 0 1 0 ...
 $ tbstatus_Active_TB              : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Covid_Cat                       : Factor w/ 4 levels "Negative","Mild",..: 1 1 1 2 3
```

Figure B.3: Dataset 2 after all cleaning and preparation has been complete

```
Classes 'data.table' and 'data.frame':  17762 obs. of  39 variables:
 $ gender_F                          : int  1 1 0 1 1 1 1 1 0 0 ...
 $ age                               : num  57 29 49 86 30 70 33 30 43 47 ...
 $ creatinineplusegfrf               : num  55 41.5 57.4 34 41 ...
 $ ureaf                             : num  3.7 1.9 3.2 8.9 2.9 ...
 $ albuminf                          : num  33 30 34 50 22 31 26 35 32 42 ...
 $ totalbilirubinf                   : num  12.5 8 8 14 2 18 18 3 15 6 ...
 $ alaninetransaminasealtf           : num  18.5 52 37.5 27 9 43 15 33 21 17 ...
 $ aspartatetransaminaseastf         : num  32 35 50 37 20 38 33 28 43 45 ...
 $ gammaglutamyltransferaseggtf      : num  24.5 41 30 47 87 180 56 53 113 29 ...
 $ lactatedehydrogenaseldf           : num  430 441 198 254 586 ...
 $ creatinekinasembckmbf             : num  50 50 1 41 35 59 2 0 28 1 ...
 $ troponinihighsensitivityf         : num  5 13 33.5 20 14 6 16 391 297 56 ...
 $ troponinthighsensitivityf         : num  182.5 415 37.5 36 161 ...
 $ ntprobnpf                         : num  1330 211 185 5592 5 ...
 $ procalcitoninrapidscreenf         : num  0 0 41.7 54.7 41.7 ...
 $ procalcitoninsensitivef           : num  11.94 0.19 0.07 0.2 1.3 ...
 $ crpf                              : num  396 301 51 14 207 ...
 $ cortisolf                         : num  1069 2901 365 131 1750 ...
 $ hba1cf                            : num  13.8 7 13 5.7 6.6 ...
 $ glucoserandomf                    : num  34 28 38.4 5.8 9 ...
 $ ferritinf                         : num  289 74 38 19 603 ...
 $ erythrocytesedimentationratef     : num  48 74 34 4 110 108 120 3 9 33 ...
 $ intnormalisedratioinrf            : num  1.19 1.37 1.16 1.56 1.15 ...
 $ fibrinogenf                       : num  8.3 8.8 7.3 4.85 4.7 ...
 $ ddimerf                           : num  0.34 0.22 0.37 2.38 2.4 ...
 $ haematocritf                      : num  0.379 0.416 0.312 0.441 0.335 ...
 $ meancellvolumef                   : num  79 92.2 91.6 93.4 97.3 ...
 $ meancellhaemoglobinconcentratf    : num  31.4 34.1 34.9 32.4 33.6 ...
 $ plateletcountf                    : num  178 193 418 206 239 ...
 $ neutrophilsf                      : num  11.51 10.82 9.02 8.05 14.61 ...
 $ lymphocytesf                      : num  1.32 1.23 1.69 1.01 1.81 ...
 $ monocytesf                        : num  0.25 0.25 0.71 0.63 1.2 ...
 $ eosinophilsf                      : num  0 0 0.08 0.01 0 ...
 $ basophilsf                        : num  0.01 0 0.0333 0.03 0.005 ...
 $ nlrf                              : num  8.72 8.8 5.52 7.97 7.77 ...
 $ cd4numnew                         : num  631 303 59.5 638 629 860 586 84 411 54 ...
 $ hivstatus_HIV_Positive            : int  0 1 1 0 1 0 1 0 0 0 ...
 $ tbstatus_Active_TB                : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Covid_Cat                         : Factor w/ 4 levels "Negative","Mild",..: 2 3 1 1 1 2
```

Figure B.4: Dataset 3 after all cleaning and preparation has been complete

```
Classes 'data.table' and 'data.frame':  44614 obs. of  14 variables:
 $ gender_F                          : int  1 1 1 1 0 1 0 0 1 1 ...
 $ age                               : num  54 79 46 57 41 29 67 49 86 71 ...
 $ creatinineplusegfrf               : num  3 33 54 55 78 ...
 $ ureaf                             : num  42.4 5.5 4.1 3.7 6.45 ...
 $ alaninetransaminasealtf           : num  29.3 30 71 18.5 32.5 ...
 $ crpf                              : num  316 9.5 315 99 79.5 301 315 51 14 30 ...
 $ redcellcountf                     : num  2.5 4.54 4.95 4.8 4.51 ...
 $ haematocritf                      : num  0.211 0.412 0.396 0.379 0.39 ...
 $ meancellvolumef                   : num  84.4 90.7 80 79 86.5 ...
 $ meancellhaemoglobinconcentratf    : num  32.7 32.3 22 31.4 33.8 ...
 $ plateletcountf                    : num  254 217 493 178 212 ...
 $ hivstatus_HIV_Positive            : int  0 0 0 0 0 1 0 1 0 0 ...
 $ tbstatus_Active_TB                : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Covid_Cat                         : Factor w/ 4 levels "Negative","Mild",..: 1 1 1 2 3 3
```

Figure B.5: Dataset 4 after all cleaning and preparation has been complete

Appendix C

Figures

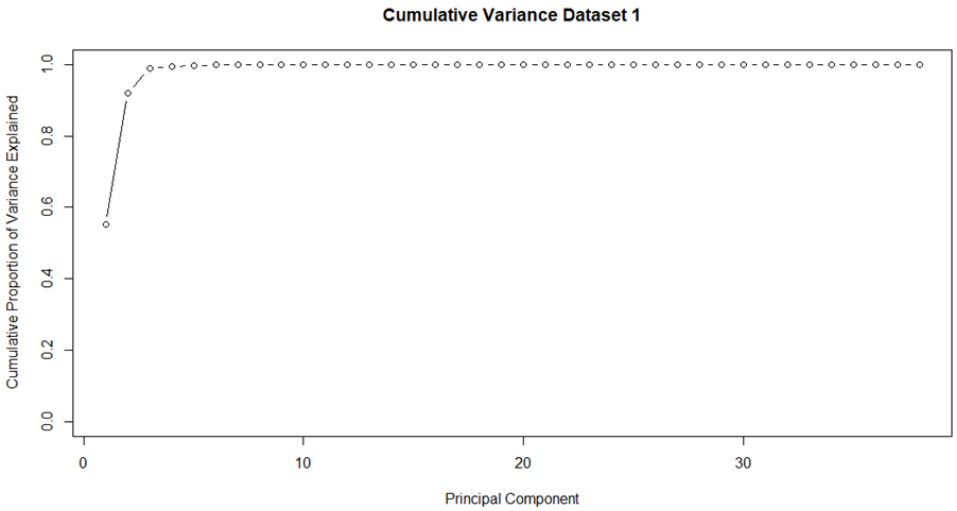

Figure C.1: Plot Showing the Cumulative Proportion of Variance Explained by the Principal Components of Dataset 1

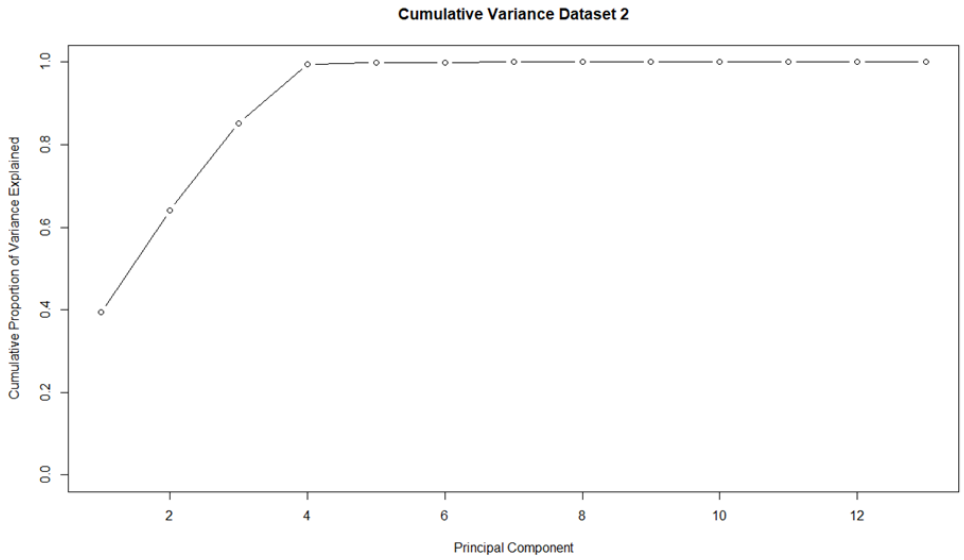

Figure C.2: Plot Showing the Cumulative Proportion of Variance Explained by the Principal Components of Dataset 2

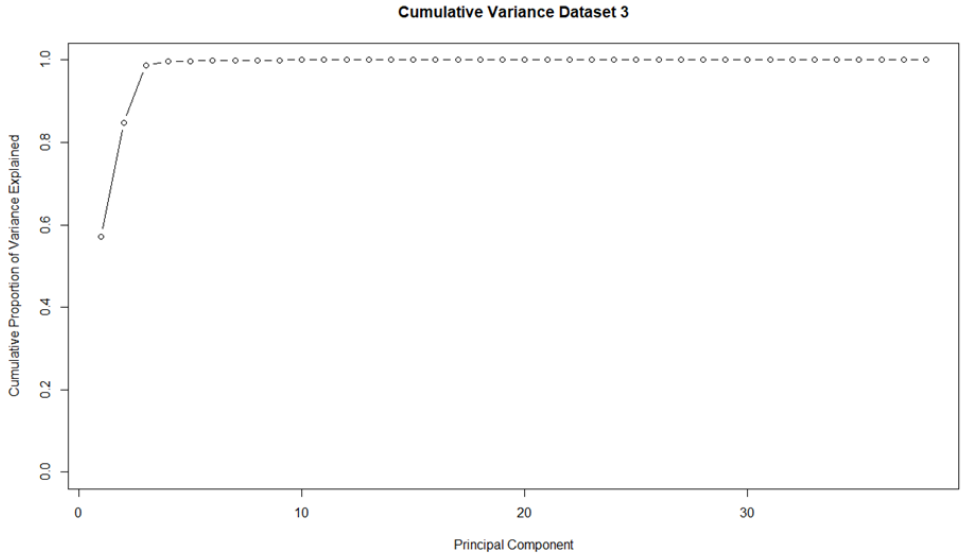

Figure C.3: Plot Showing the Cumulative Proportion of Variance Explained by the Principal Components of Dataset 3

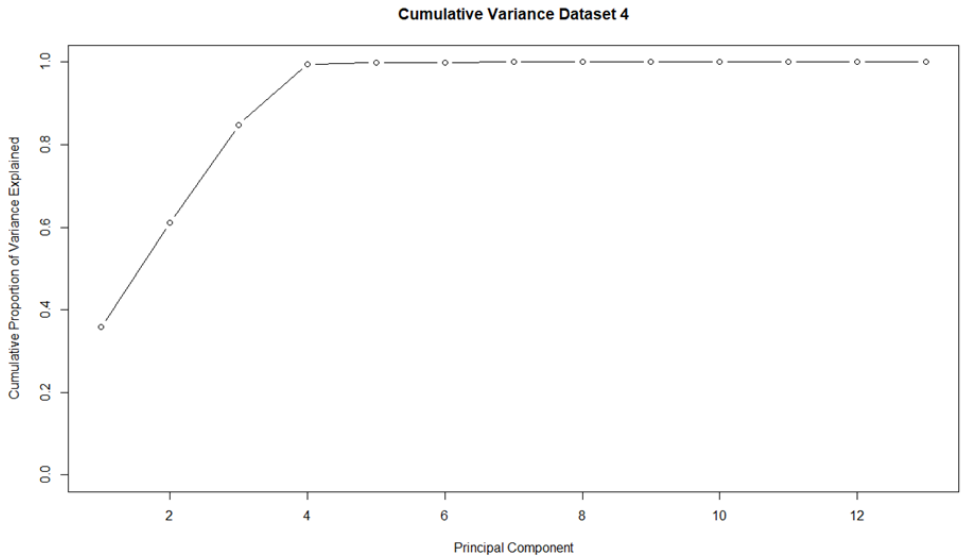

Figure C.4: Plot Showing the Cumulative Proportion of Variance Explained by the Principal Components of Dataset 4

Appendix D

Tables

Table D.1: MSA Variable Scores for Dataset 1

Variable	MSA Score
Gender Female	0.63
Age	0.56
Creatinine plus eGFR	0.69
Urea	0.69
Albumin	0.78
Total Bilirubin	0.70
Alanine Transaminase (ALT)	0.63
Aspartate Transaminase (AST)	0.64
Gamma Glutamyl Transferase (GGT)	0.66
Lactate Dehydrogenase (LDH)	0.67
Creatine kinase-MB (CK-MB)	0.44
High-sensitivity cardiac Troponin I	0.55
High-sensitivity Troponin (HS-Trop)	0.59
NT-proBNP	0.72
Procalcitonin Sensitivity	0.67
C-reactive Protein	0.72
Cortisol	0.72
HbA1c	0.46
Glucose Random	0.55
Ferritin	0.72
Erythrocyte Sedimentation Rate	0.80
International Normalized Ratio (INR)	0.81
Fibrinogen	0.59
D-dimer	0.78
Red Cell Count	0.54
Haematocrit	0.53
Mean Cell Volume	0.19
Mean Cell Haemoglobin Concentration	0.53
Platelet Count	0.49
Neutrophils	0.63
Lymphocytes	0.55
Monocytes	0.62
Eosinophils	0.47
Basophils	0.55
Neutrophil-Lymphocyte Ratio (NLR)	0.71
CD4 count	0.49
HIV Status	0.56
TB Status	0.70

Table D.2: MSA Variable Scores for Dataset 2

Variable	MSA Score
Gender Female	0.58
Age	0.52
Creatinine plus eGFR	0.54
Urea	0.55
Alanine Transaminase (ALT)	0.53
C-reactive Protein	0.69
Red Cell Count	0.43
Haematocrit	0.43
Mean Cell Volume	0.16
Mean Cell Haemoglobin Concentration	0.52
Platelet Count	0.51
HIV Status	0.66
TB Status	0.64

Table D.3: MSA Variable Scores for Dataset 3

Variable	MSA Score
Gender Female	0.57
Age	0.53
Creatinine plus eGFR	0.58
Urea	0.66
Albumin	0.78
Total Bilirubin	0.75
Alanine Transaminase (ALT)	0.63
Aspartate Transaminase (AST)	0.64
Gamma Glutamyl Transferase (GGT)	0.60
Lactate Dehydrogenase (LDH)	0.70
Creatine kinase-MB (CK-MB)	0.46
High-sensitivity cardiac Troponin I	0.58
High-sensitivity Troponin (HS-Trop)	0.55
NT-proBNP	0.74
Procalcitonin Rapid Screen	0.46
Procalcitonin Sensitivity	0.64
C-reactive Protein	0.73
Cortisol	0.56
HbA1c	0.45
Glucose Random	0.54
Ferritin	0.78
Erythrocyte Sedimentation Rate	0.70
International Normalized Ratio (INR)	0.60
Fibrinogen	0.54
D-dimer	0.76
Haematocrit	0.67
Mean Cell Volume	0.42
Mean Cell Haemoglobin Concentration	0.58
Platelet Count	0.52
Neutrophils	0.60
Lymphocytes	0.50
Monocytes	0.49
Eosinophils	0.42
Basophils	0.54
Neutrophil-Lymphocyte Ratio (NLR)	0.65
CD4 count	0.38
HIV Status	0.71
TB Status	0.77

Table D.4: MSA Variable Scores for Dataset 4

Variable	MSA Score
Gender Female	0.61
Age	0.51
Creatinine plus eGFR	0.55
Urea	0.55
Alanine Transaminase (ALT)	0.52
C-reactive Protein	0.67
Red Cell Count	0.46
Haematocrit	0.47
Mean Cell Volume	0.19
Mean Cell Haemoglobin Concentration	0.59
Platelet Count	0.57
HIV Status	0.69
TB Status	0.67