APPLYING BUSINESS ANALYTICS IN PRACTICE USING DATASETS FROM THORACIC SURGERY CASES

OLAFUYI, SAMUEL OLASUNKANMI
(201453491)

This dissertation was submitted in part fulfilment of requirements for the degree of MSc Information management.

DEPT. OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF STRATHCLYDE

SEPTEMBER, 2015.

DECLARATION

I declare that this dissertation embodies the results of my own work and that it has been composed by myself. Following normal academic conventions, I have made due acknowledgement to the work of others.

I declare that I have sought, and received, ethics approval via the departmental ethics committee as appropriate to my research.

I give permission to the University of Strathclyde, department of computer and information sciences, to provide copies of the dissertation, at cost, to those who may in the future request a copy of the dissertation for private study or research.

I give permission to the University of Strathclyde, department of computer and information sciences, to place a copy of the dissertation in a publicly available archive.

(please tick) Yes [ ] No [ ]
I declare that the word count for this dissertation (excluding title page, declaration, abstract, acknowledgements, table of contents, list of illustrations, references and appendices is 20523.

I confirm that I wish this to be assessed as a Type 1 2 3 4 5
Dissertation (please circle)
Signature:
Date:

ABSTRACT

Data mining in the healthcare sector is a growing field that merges information technology and clinical practise. However, it has suffered some setbacks in terms of acceptability by medical professionals despite its high potential for extracting knowledge from information in medical records.

This research sought to apply machine learning tools in making predictions on patient health status based on anonymised health dataset from a thoracic surgery dataset that included 470 instances and 17 attributes and it described the post-operative life expectancy of the patients within a year after undergoing a surgery procedure. This research revealed the advantages of the data mining process by comparing the performance of several algorithms and proposing the use of the random forest algorithm on an imbalanced health dataset based on its ease of dealing with noise and overfitting problems; its transparency; ease of description and performance measure.

The Weka and R software were used to analyse the datasets. These software programs were first integrated in order to combine the strength of data visualization in R and classification algorithms in Weka during the analysis. Then different pre-processing activities were employed to prepare the data for mining and knowledge discovery, followed by algorithm selection and further post-processing activities. An iterative visualization process was employed throughout the process. Finally, the random forest method was used in predicting the post-operative live expectancy of lung cancer patients with the patients and the results explained. The random forest outperformed other algorithms by correctly diagnosing 84.1% of all thoracic surgery patients that died and 81.3% of all the patients that survived with an overall accuracy of 82.7%.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1. Background and overview

Business analysis is a technology that has evolved gradually over the years and it involves the application of statistical techniques, data and text mining technology, mathematical modelling and machine learning to solve business problems. This term metamorphosised from what was first referred to as "artificial intelligence" in the 1950s which was used to describe a highly technical and specialized way of studying how to create computer software that have the capacity to behave intelligibly to "business intelligence" in the 1980s when it was described as reporting tools used to search trends in historical data. Only less than a decade ago, the term "business analysis" evolved to describe the current trend of what encompasses the information technology and business industries (Hsinchun et al., 2012). Business analysis involves the analytical aspects of business intelligence that depends on algorithms that can statistically determine relationships between data in order to forecast the outcome of events. In other words, business analysis uses historical data is to describe trends or related patterns and to predict future outcome (Ying, 2014). This process results in a synergy between information technology and business processes.

Analysing organisational data has emerged into a significant area of study for both specialists and researchers owing to the increasing awareness of the value of data as an asset for making informed decisions. Today, organisations have the ability to create decision support systems (DSS) that supports both the growing amount of data that is warehoused and the growing customer database (Bose and Sugumaran, 1999) . The term "Big data" which is now a cliché in the world of data analytics and mining is a very significant area of research in business analytics, not only gaining attention in the IT industry but also, the academia. This may be attributed to the continuous advancement of technology, resulting in cheaper electronic storage and the increasing ability of computers to process large amount of data which enables organizations to leverage data in their business processes. According to McKinsey global institute, "Big data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyse" (McKinsey, 2012). Also, Gartner defines big data as "a high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making” (Gobble, 2013). Over the years, the storage capacity of computers has been on an increase from gigabytes to terabytes and now petabytes which explains why in terms of data storage; the present world of information technology is referred to as ‘Petabyte age’. The Wired magazine, in 2008 noted that “The biggest challenge of the Petabyte age won't be storing all that data, it will be figuring out how to make sense of it.” The current interest in data mining activities can be attributed to promotion initiatives of leading research and consulting companies such as Gartner, International Business Machines Corporation (IBM), Systems Applications & Products in Data Processing (SAP) etc. as well as the growth of the social media industry such as Facebook, Twitter and others.

Big data has become a subject of attention and many businesses are looking for technology-based competitive advantage which can be derived from the quality of data they have and the ability to gain meaningful information from them(Goes, 2014). Equally, researchers are looking for ways to extract more knowledge from this technology and there are growing number of vendors that continue to build software and tools for managing organizational data with the aim of reducing its complexity and latency while empowering the business user. It has become evident that storing, managing, and analysing big data will break the traditional trends for measuring organizational success and soon form a standard for differentiating between high-performing and low-performing organizations(Ying, 2014).

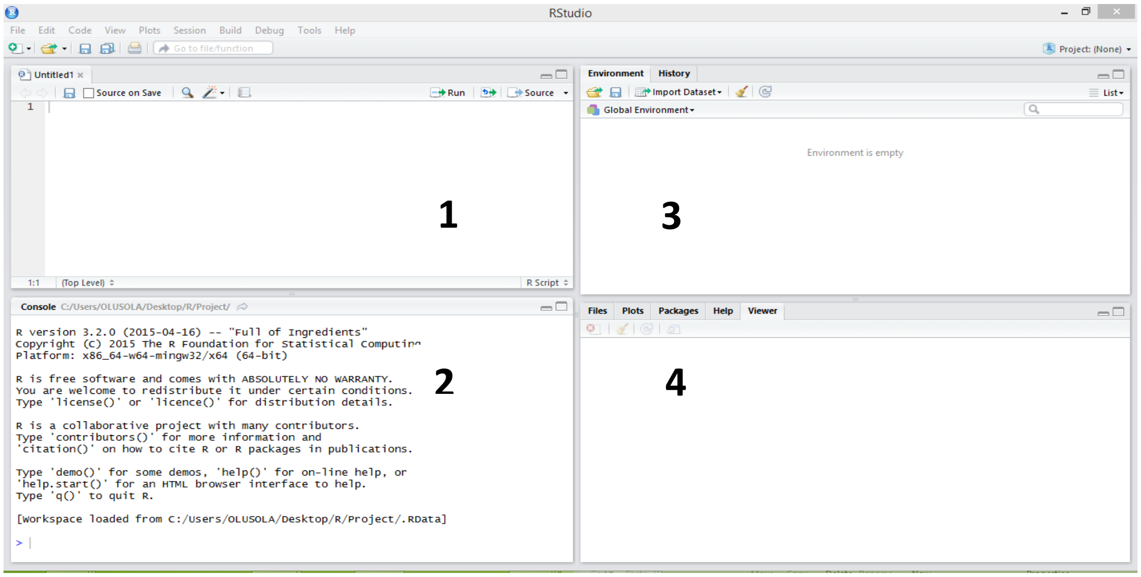
The benefits of efficient data mining cannot be overemphasized: with a strong acumen of statistics and modelling, data mining can be done to improve a variety of functions such as supply chain management through reducing inventory and optimizing supply chain flows; customer selection, loyalty and service through proper identification of customers with the greatest potential; establishing a pricing system that would yield maximum profit; improving human capital through selection of best employees for the right roles; enhancing product and service quality; improving financial performance; research and development of company products and so on (Davenport, 2006). These functions can be applied in retail stores; higher education; e-commerce and market intelligence; e-government and politics; science and technology; health and medicine; security and public safety; sports etc., however only a few organisations are already taking advantage of them. Purdue University has started using a course signals program that identifies the behaviour of their student with different colours and compare them with the behaviour of past successful students. The colour indicator helps them to know students that are not performing to the best of their ability and guides them towards success. Another case study is that of one of the largest retailing company in America known as Target. Target indicated that they were able to use data mining technology to track and analyse customer behaviour. They achieved this by using their vast amount of customer data to identify buying patterns, improve customer satisfaction, and predict future patterns for selecting promotional strategies and increasing profit (Corrigan et al., 2014). On an occasion, maternity-specific mails and coupons meant for a teen girl who was still in high school was received to the utmost surprise of her father who did not know that his daughter was already pregnant. This further demonstrates the growing interest around the advantages derived from efficient handling of organisational data. Other emerging areas of research in business analytics includes text analytics, web analytics, network analytics and mobile analytics (Hsinchun et al., 2012).

1.2. Research context

This research seeks to analyse the existing trend in business analysis in health and medicine otherwise known as health analytics. This would be achieved through the analysis of an imbalanced data set of relevant patient clinical and demographic information with respect to lung cancer patients who have undergone thoracic surgery. The data set is a thoracic surgery data sourced from the UC Irvine machine learning repository website, the centre for machine learning and intelligence systems. It was gathered between 2007 and 2011 by Marek Lubicz, Konrad Pawelczyk, Adam Rzechonek, and Jerzy Kolodziej from Wroclaw thoracic surgery centre for patients that had major lung resections for primary lung cancer. The research database is a part of the national lung cancer registry which is administered by the institute of tuberculosis and pulmonary diseases in Warsaw, Poland. Thoracic surgery postoperative pulmonary complications (PPCs) are usually associated with high incidence of life-threatening events and costs (Sabate et al., 2014), hence the need for a decision support system that could help with future diagnosis and treatment of patients.

The dataset is in a structured form and contains 17 attributes and 470 instances. Data was analysed with the Revolution software R (version 3.2.0) and the Waikato Environment for Knowledge Analysis (Weka) software (version 3.7.12). Their use was demonstrated throughout this work. Both Weka and R are open source free software packages issued under the GNU general public license. The R software has good adaptability and versatility to different files formats (Tippmann, 2015), and offers exceptional interactive analysis for development of statistical and data analysis applications. This software offers users the ability to develop their own analysis by exploring and inventing new approaches, therefore, users are not constrained to “off the shelf tools” available in some other packages. R already has over 500 readily available packages available through the comprehensive R archive network (CRAN) used for various forms of analysis and the number continues to grow (Carslaw and Ropkins, 2012). Work is best done on the R package using a point and click integrated development environment (IDE) system known as the RStudio. As seen in Figure 1.1., it has four main resizable workspaces which are:

1. The scripting interface located on the top left corner.
2. The console located on the bottom left corner.
3. The environment located at the top right corner.
4. The files, plots, packages, help and viewer section located at the bottom right.



```
R version 3.2.0 (2015-04-16) -- "Full of Ingredients"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from C:/Users/OLUSOLA/Desktop/R/Project/.RData]

> 
```

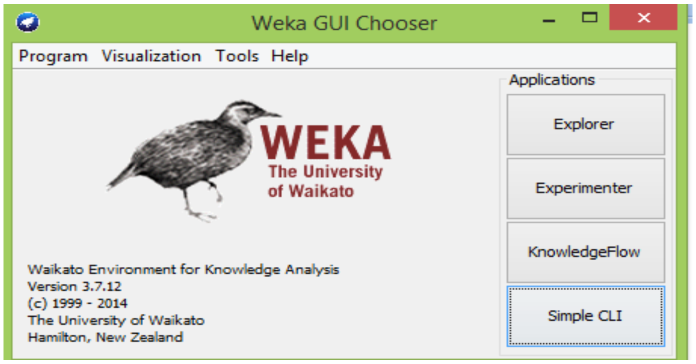
Figure 1.1 The Revolution R workspace.

The scripting interface is used for working on various scripts while the R commands are saved on the console. The environment contains the objects, values, functions in the present working directory and the history tab runs log of R commands.

Weka is a machine learning software developed by the University of Waikato New Zealand which employs machine learning algorithms for data mining tasks. It has four major application interfaces which includes;

1. Explorer.
2. Experimenter.
3. Knowledge flow.
4. Simple command line interface.

By default, the explorer interface has six panels which contain different algorithms used for knowledge discovery and data mining tasks. The experimenter interface is used to compare the performance of various machine learning methods. The knowledge flow interface is a Java Beans application that contains a graphical interface which performs virtually the same operation as the experimenter interface. It uses a work flow structure which can be enhanced by tweaking the parameters of algorithms while the simple command line interface (CLI) is used to write commands for data mining. Figure 1.2 shows the Weka workspace and application interfaces.



Waikato Environment for Knowledge Analysis
Version 3.7.12
(c) 1999 - 2014
The University of Waikato
Hamilton, New Zealand

Figure 1.2 The Waikato Environment for Knowledge Analysis (Weka) GUI chooser.

1.3. Research questions

Using a combination of techniques for supervised learning on an imbalanced data problem, this research sought to demonstrate the various stages to predictive analysis on the one year life expectancy of patients after thoracic surgery by answering the following questions;

1. How the sanity of the data set can be improved for better analysis of patient health status.
2. How to determine the most relevant attribute in the data set in order to improve predictions of the probability of patient survival.

3. How the technique of using boosted algorithms can influence the performance of various algorithms required to classify patient data.
4. What the options are available for measuring classifier performance.
5. The role of exploration through visualization in the data mining process of patient clinical or demographic information.
6. What the strength and weakness from the outcome of the various algorithms applied to the dataset of the organization is during data mining.
7. How prediction results can be applied in reality to explain patient health status.

1.4. Research Objectives

The aim of this dissertation is to show how data mining and statistical techniques can be combined to answer the research questions using clinical and demographic data set for patients with lung cancer.

In order achieve this, this research was focussed on the following objectives;

1. Integration of two software to improve the process of data mining.
2. Recommendation a classification model.
3. Identification of the patterns of dealing with imbalanced dataset.
4. Capturing the essential features of a medical data.
5. Identification of related patterns in a data set by performing exploratory analysis through visualization.
6. Prediction of possible outcome of given the observation of a medical record.

CHAPTER 2

2. LITERATURE REVIEW

This chapter presents an overview of past research works in area of data mining and efforts made to improve its practice. Its presents an outline of the research efforts surrounding the world of business analysis followed by a critique of similar works which forms the basis for this research. Also, the tools used for business analysis is explored with the aim of presenting real world scenarios of its application. The advantage of using multiple software for data analysis is explored, showing existing research work that corroborates this fact. Health data sets often contain imbalanced class, a problem that needs to be surmounted when analysis is done. The final section would take account of research efforts taken to tackle this problem.

2.1. Research efforts in the world of business analysis

Despite the myriad of commercially available tools and software for data analysis and mining today, there seem to be a growing gap in knowledge on how to interpret the information contained in organisational database and act effectively on it (Gandomi and Haider, 2015). Most enterprises today face the challenge of attempting to address the rising number of data sources required for analysis and reporting while those with unstructured data pool face worse data management problems(Grossman and Siegel, 2014). Today, some organisations have huge quantity of data that have been cleaned, summarized, integrated, and stored in large data warehouses while others have their data set in different structure but the question of how these organisations are able to fully harness potential gains from these data remains unanswered (Chopra, 2014).

There are a number of academic researches explaining the principles of business analysis but the focus has mostly been on comparing various algorithms used for describing and processing data with little understanding of the relationship that lies within the data; how the algorithms employed functions and how to generate business value from it (Kumar et al., 2013; Witten and Frank, 2005; Hand, 2006). Though this is highly commendable, they contain various theoretical terminologies, theories and formula that are devoid of demonstrating how these methods can be applied to real life situations. Quinlan stated that determining whether an algorithm is better than the other would be difficult because algorithms behave differently when applied for different tasks (Quinlan, 1994). Maindonald also mentioned that by comparing algorithms, analysts who have gained expertise in a particular method will be biased in getting the best out of other methods (Maindonald, 2006). The applicability of data mining techniques should not be limited to the use of various algorithms but extended to employing a structural step by step from the initial pre-processing to the post-processing stages. This involves stating the data mining goals, sanitizing data and selecting them for queries that are relevant to the requirement of the business (Bose and Sugumaran, 1999). A search at the United Kingdom's government website gov.uk and various grants websites across the United Kingdom indicates the growing interest in research areas that have to do with the management of data to support business decisions. In the 19th Institute of Electrical and Electronics Engineers *(*IEEE) international requirements engineering conference, it was explained that actions to standardize business analysis practices through accreditation bodies with the emergence of various regulators such as the Institute of Business Analysis(IIBA) and organizational standards has offered little help as business analysts have shortcomings in harnessing their skill set (Wever and Maiden, 2011).

In a published paper of the global conference on business & finance proceedings, it was reported that "The time is right for a publication that combines technical specifics with detailed prescriptions for use by managers" (Amadio et al., 2014). With the exception of a few large companies like Google, Facebook and LinkedIn which are information intensive, most corporations still struggle to understand how to apply the challenges and potential of business analysis to make informed decisions on their data (Goes, 2014). This implies that data analysis should be aimed at improving decision making and giving a guide that can be understood and interpreted not just by the technical professionals but also by business people.

2.2. Data analysis for clinical diagnosis

The health care industry has a large amount of historical data driven by record keeping, patient care, compliance and regulatory requirements. There is also a growing need to store these data in digital format in order to reduce cost and improve the quality of health care delivery. In 2011, the big data for the U.S reached 150 exabyte and it has been reported that at the current growth rate, the volume would soon reach the zettabyte scale (Raghupathi and Raghupathi, 2014). The government is becoming increasingly aware of the relevance of effective management of their data as it relates to all areas of the economy. In 2014, the US department of health and human services announced that it would invest over 840 million dollars for a four year period to help

150,000 medical practitioners improve patient outcomes, reduce hospitalizations and reduce unneeded test, an initiative this is highly dependent on improving the quality and efficiency of data that they hold and how knowledge is extracted from them. This further emphasises the importance of data and data mining efforts in the health care industry.

According to WhaTech Channel in their recent medical market report by James Martin in 2015, data analytics in the healthcare can be categorised into five major areas namely big data analytics; health insurance portability and accountability (HIPAA); predictive analytics; genome sequencing; genome wide association studies in electronic medical records and genomics (eMERGE) network. Analytic techniques include:

i. Predictive analytics used for following trends from simulation and modelling technique.
ii. Prescriptive analytics to optimize clinical and financial outcomes.
iii. Descriptive analytics which uses various standard reporting techniques to describe the current situation in a data.

The focus of this research is on predictive analytics in relation to decision supports systems for improving the quality of health care practice. Proper management of human life is involved in healthcare and getting it right the first time is crucial. To support decision making, quality medical data is important and a great deal of work is already being done in the public health domain especially in the area of predictive data mining (Hickey, 2013). Most healthcare organisation are using two sets of data: retrospective data, which are event-based information derived from stored medical records and real-time data which comprises information gathered at the point of administering care to a patient such as blood pressure, pulse rate, body temperature , etc. Administrating support will therefore involve a combination of both retrospective data and real time data to analyse how treatment will work in a particular situation. Though the systems that support decision making functions are widely available today, making effective use of them can be challenging because of its multi-faceted nature of the clinical practice. Some other challenges involved includes the ethical, legal and social constraints on the use of patient data and how they are affected by the data protection (Hickey, 2013).

In 2012, VigiLanz Corp, an American based clinical decision support software company forecasted an increase in demand for real-time clinical decision support surveillance systems to manage patient population across America. The president and chief executive officer, David

Goldsteen, however lamented how the benefits that could be gained from the increased usage of electronic health records (EHR) systems today is highly unexploited. He further stated that "the primary problem with the use of vital clinical data collected today by most heath care providers is that clinicians and other hospital personnel depend largely on retrospective analysis, which is akin to attempting to predict where you are driving by looking into the rear view mirror,". Van Valkenhoef et al highlighted the shortage of operational information systems for data-mining operations and decision support efforts for clinical trial results(Van Valkenhoef et al., 2013). A white paper published by the SAS institute one of the industry leaders in data analysis, revealed another major deterrent to effective health care analysis which is the continuous hesitation of physicians to embrace electronic methods as a support for making medical decisions. It was further explained that some elementary questions needs to be answered in line with the quality of patient's interactions. This may be the treatment procedure that would yield the best result for the patient with a particular genetic profile, patient risks factors with respect to taking a particular medication, how well combination of therapies can assist a patient to undergo a procedure and the protocols that would help gain the best rehabilitation results for a specific population. To answer these questions, clinicians and researchers would need business intelligence traditionally called 'healthcare analytics' in the healthcare practice.

2.3. Classification of health data.

Today, there are several researches in the health domain that applies various classification techniques for making forecasts from patient records. Algorithms for multivariate and bivariate classifications and regressions that use conditional probability distributions; distance based classifiers; tree classifier, etc. are available however, the question arises about how researchers have been able to effectively exploit these tools for efficient data mining.

As proven by Hickey, many researches are limited to the use of a classification algorithm known as Naïve Baiyes due to the belief that it performs well with health data set (Hickey, 2013). This assumption however, does not take cognizance of the diverse data sets in the health domain which may facilitate the need to adopt the application of boosted classifiers as a better way to generate better classification results. Researchers are therefore, constrained to use single models that produce acceptable results or benchmark a subset of models using cross validation results on test sets (Übeyli and Güler, 2005).

Though there is no established theory to serve as a guide for selecting a classification model based on the complex nature of diagnostic tasks (Übeyli and Güler, 2005), researchers must be willing to go through the rigor of performing various tests and experiments in order to select the best tool for classification. A similar work on this data set was done by (Sindhu.V et al., 2014) as published in the international journal of computer technology and applications in India but the tools used in the analysis seemed to limit the applicability of the work. The researcher restricted the work to the use of single classification algorithm without checking the effects of boosting them for optimum performance. Also, despite the fact that it has been established that the use of only accuracy results can be misleading (Goadrich et al., 2006) the researcher restricted his choice of classifier to the accuracy performance of the testing methods which resulted in a weak recommendation. The research also lacked practical application because the results may not be easily interpreted by a non-technical person. Though research work by (Danjuma, 2015) followed the same pattern of Sindhu's work, there was some improvement by an attempt to use the Receiver Operative Characteristic (ROC) area as an additional form of selection criteria for the recommended algorithms but most of the performance metrics were not properly expressed. For example, the ROC area, precision, recall, etc. were wrongly expressed as percentages instead of fractions. The research by (Harun and Alam, 2015) compared more algorithms and attempted to boost the performance of the algorithms which is laudable, however, there was no explanation on how the final results was gotten and more could have been done to explore other ways of boosting the algorithm such as stacking, bagging and boosting to mention a few which are known to do better that other regular boosting methods (Bauer and Kohavi, 1999). The work by (Zięba et al., 2014) gave more insight into the use of ensemble classifiers and how they work but it was theoretically inclined and lacked practical application. The research by (Adam et al., 2014) was detailed in terms of giving a proper explanation of the concepts applied and how the algorithms were boosted however, it was restricted to classifier selection according to percent accuracy and did not employ more advanced methods like the area under the receiver operative characteristic curve or the area under the precision recall curve..

Other classification methods and several proposed performance measures for empirical validation employed on health data are available (Keilwagen et al., 2014). For instance, (Maroco et al., 2011) employed data mining in the prediction of dementia, a disease associated with ageing by comparing the results of seven non parametric classifiers (i.e. classifiers based on the statistical probability distribution of each class). The classifier performance was compared using ROC area, classification accuracy, specificity and sensitivity and it was concluded that the random forests and linear discriminant analysis performed best amongst all the classifiers tested for prediction. Also, (Oztekin et al., 2009) employed logistic regression , decision trees and neural networks to do a survival analysis for patients that underwent graft heart-lung transplant. He then proposed a model that outperforms the conventional prediction models. Overall, despite the fact that organisations places high value on data visualization and trend analysis, these researches were restricted to comparing and selection of algorithms without practical application of how outputs are generated from classification.

2.4. Applications and tools for business analysis

As recognised by (Pavlo et al., 2009), the widely used structured query language (SQL) cannot effectively manage the innumerable rows of stored data most organisations hold today. This has triggered the emergence of various statistical and data mining packages that implements different programming languages such as R, SPSS, Python, Java, SAS, Matlab, C/ C++, Unix, Hadoop, etc. Most academic research work available today uses just one software for analysis, hence, the content of their analysis is somewhat limited and does not depict real life situations. A recent poll by kdnuggets in 2012 indicated that their readers which were mainly professionals in the industry used an average of 2.5 languages with SQL, R, and Python being the most common ones.

Most data analysis software package in existence today come with their strength and weaknesses especially because they are implemented by a specified programming language which presents a limitation in their versatility and functionality .To improve this limitation, some advances have been made to integrate software and make them easy to interact with each other. For example, tools for combination perturbation screen analysis (TOPS) a Java and R – based software toll with a simple graphical user interface(GUI) was developed to make R scripts available to users that have little background of scripting languages and for effective combination of statistical analysis, data exploration and visualization (Muellner et al., 2014). A software known as Ricardo was developed by IBM as a part of an eXtreme Analytic Platform (XAM) project which integrates R and Hadoop for statistical handling system and data management system respectively (Das et al., 2010). Also, a Java Software Framework known as Elki (Environment for Developing KDD) was built to combine the diverse areas of Weka and YALE otherwise known as Rapid Miner into one solution (Achtert et al., 2009). This research would combine the functionalities of R and Weka to improve flexibility and gain optimum results from the strongest aspects of both software.

2.5. The imbalanced class problem.

Class imbalance problem is said to occur when there are many more instances of some class than some others in a data set. This usually results in the classifiers being overwhelmed by the majority class and therefore performs below their optimum ability. Over the years, researchers have worked on imbalanced class dataset as part of their data mining efforts but identifying this as an obstacle to efficient classification became a major concern just over a decade ago (Chawla et al., 2004). Imbalanced class causes a suboptimal classification performance and it is highly prevalent in many domains such as fraud detection; risk management; text classification; medical diagnosis and monitoring etc. For example, a medical record may have cases where only one in every 10000 person is likely to get a certain rare infection all over the world. This is a real scenario faced in the data mining research and efforts are being put in place to address this growing concern through several workshops and research.

It was mentioned by (Chawla et al., 2004) that this problem can be solved either at the data level through oversampling or under sampling or at the algorithmic level by assigning costs to various classes as a check for the imbalanced data. Other methods have been recommended such as neighbourhood cleaning, (Laurikkala, 2001); applying a cost-sensitive boosting algorithm (Sun et al., 2006); using a weighted naïve Bayes predictor at the leaves of decision trees (Hang Yang et al., 2013); separating larger classes into several smaller subclasses according to their proximities (Pramokchon and Piamsa-nga, 2014); synthetic minority oversampling technique (Blagus and Lusa, 2013) etc.

Oversampling, a common method of solving class imbalance problems has received approvals because of its effectiveness in learning imbalanced datasets and criticisms due to increase in training time and over fitting tendencies. Also, under sampling has been criticized because it has the tendency of discarding potentially useful training instances which may worsen the result of the classifier (Zhou and Liu, 2006). Cost sensitive learning is a growing technique used to select classifier algorithms by establishing a trade-off between the costs of the classifier errors. Like the oversampling technique, this technique offers a solution to strengthen the learning of a minority class of an imbalanced dataset at the algorithm level (Hu et al., 2014). Cost sensitive learning makes the trained classification algorithm more cost sensitive by rescaling and reweighting the error rates i.e. the false positive and false negative rates thus, minimizing the cost of misclassifying the training sets. Reweighting applies a threshold moving technique such that output thresholds or volumes are moved towards inexpensive classes while expensive classes would have a lesser threshold making higher costs harder to misclassify. This approach helps to wrap a base classifier, making it cost sensitive by a procedure known as MetaCost proven to always produce a large cost reduction (Domingos, 1999).This research would however adopt a method that suites the peculiarity of the data set.

CHAPTER 3

3. RESEARCH METHODOLOGY

This section gives a description of the methods used to choose an algorithm which was used for the classification of the thoracic datasets. In this research, the R and Weka software programs used for analysis was first integrated. Different pre-processing techniques were employed to sanitize the dataset with some exploration exercise to uncover the relationship that lies within the various attributes of patient data. Then, single and multiple classifier algorithms were applied to the dataset with the aim of choosing the suitable algorithm that was then used to build the classification model.

The typical workflow of a data analyst as described by (Das et al., 2010) involves an initial exploration of the data through visualization, sampling and summarization using statistical tools after which the model is built and explored again through visualization and formal validation procedures to determine its suitability. The model building goes through multiple iterations which may be prompted by feedback from the users or other quality requirements until the final model that would yield optimum classification results for improving business activities and supporting decision making in the healthcare sector is determined. The steps followed are summarized below;

1. Data pre-processing and exploration.
2. Application of data mining algorithms.
3. Post processing of data set.

The diagram in Figure 3.2 illustrates the sequence of these steps taken during this research

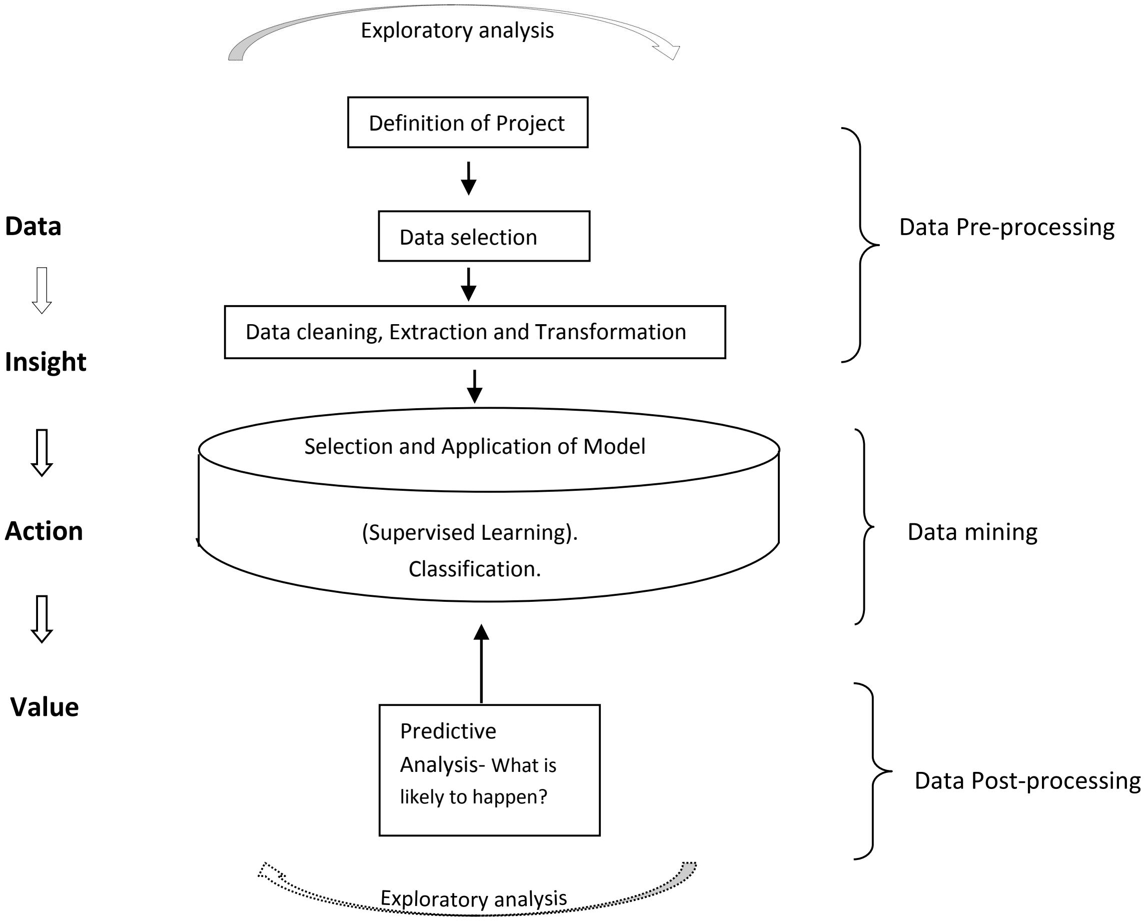


Figure 3.1 Research Methodology.

3.1. Integrating R and Weka

Both Weka and R software were integrated in a unified system in order to make their tools available on a single platform. This gives the advantage of exploiting one of their biggest strength which is the data visualization with gglot2 package (Hornik et al., 2009) and the classification algorithms (Achtert et al., 2009)provided by R and Weka respectively. Both software programs offer two ways of integration: by integrating R into Weka using the Weka RPlugin package or vice versa with RWeka Package. The functionality of R is interfaced into Weka, thus creating a fortified Weka Interface using the steps below on a windows operating system: The researcher;

1. Downloaded and installed both R and Weka on the work station.
2. Downloaded and installed the Rplugin with the script below from the CLI of Weka.

```
Java weka.core.WekaPackageManager -install-package
<http://sourceforge.net/projects/ weka/files/weka-
packages/RPlugin1.1.8.zip/download>
```

3. Installed the “rjava” package from the R Console in the R Studio.
4. Set the environment variable R_HOME to point to the installation directory of the R software.
5. Set R_LIBS_USER to point to the personal directory of the user.
6. Inserted the R executable path in the PATH environment.

After integrating Weka with R, a seventh button known as the RConsole was created in the Weka explorer interface having three panels as seen in Figure 3.2. The RConsole panel is used for writing the R commands while the graph history panel gives a log of executed commands used for plotting graphs. The Rplot gives a view of the most recent graph. This R functionality is also reflected in the knowledge flow interface as an R script executor plugin.

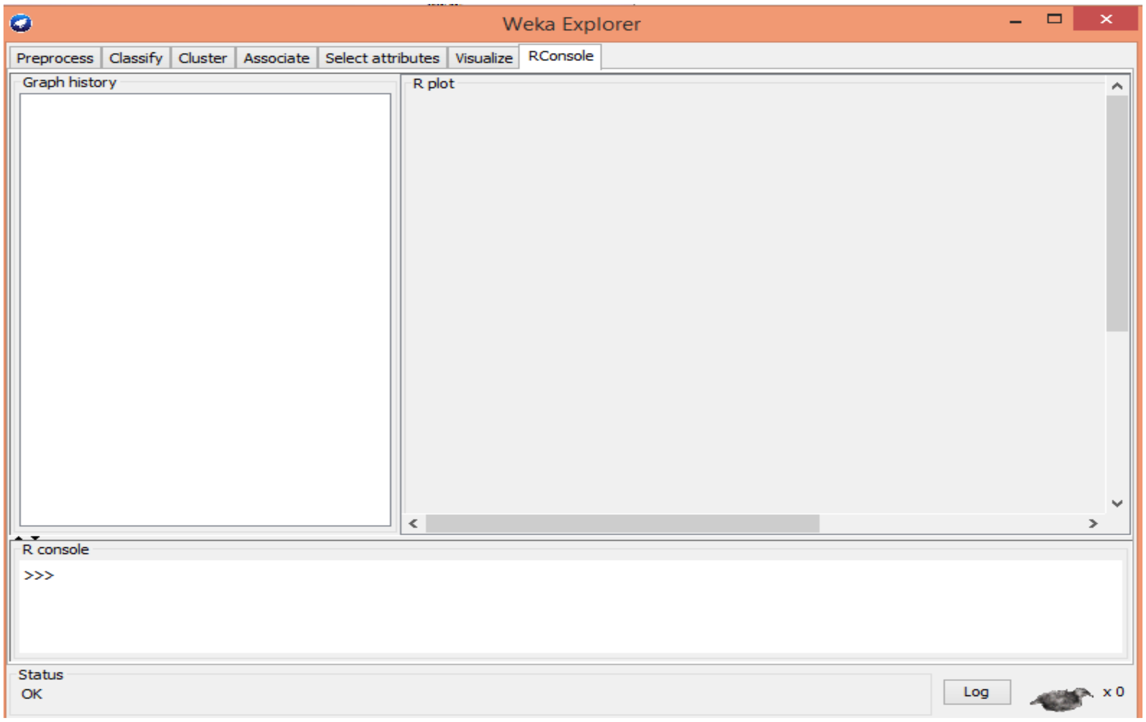


Figure 3.2: A view of the Weka explorer with the RConsole Button.

3.2. Data types and representation.

Generally, data may be stored in a structured, semi structured and unstructured form. Structured data are data that have been classified for easy processing and accessibility and they have the advantage of being easily queried, stored and analysed while unstructured data cannot be readily classified. Semi- structured data is the cross between the two; however, extracting value from unstructured data is the most challenging task (Veeranjaneyulu et al., 2014). Also, datasets are usually represented as either quantitative or qualitative variables within a database to which analysis is applied. Quantitative variables as the name depicts is used to represent countable variables expressed as numbers while qualitative variables are used to represent different categories otherwise known as categorical variables. Between these two broad groupings, there are some distinctions listed below;

1. Qualitative/ categorical variables further divided as;
    - Nominal – Exists without any form of ordering thus allowing for simple classification into a number of categories.
    - Ordinal- Data in this category exists in an orderly way which can be ranked.
    - Binary – This are data categorised as with two possible outcomes.
2. Quantitative variables further divided as;
    - Discrete Variable – Variables exists as whole numbers.
    - Continuous – Variables exists in a range.
        - Ratio
        - Interval

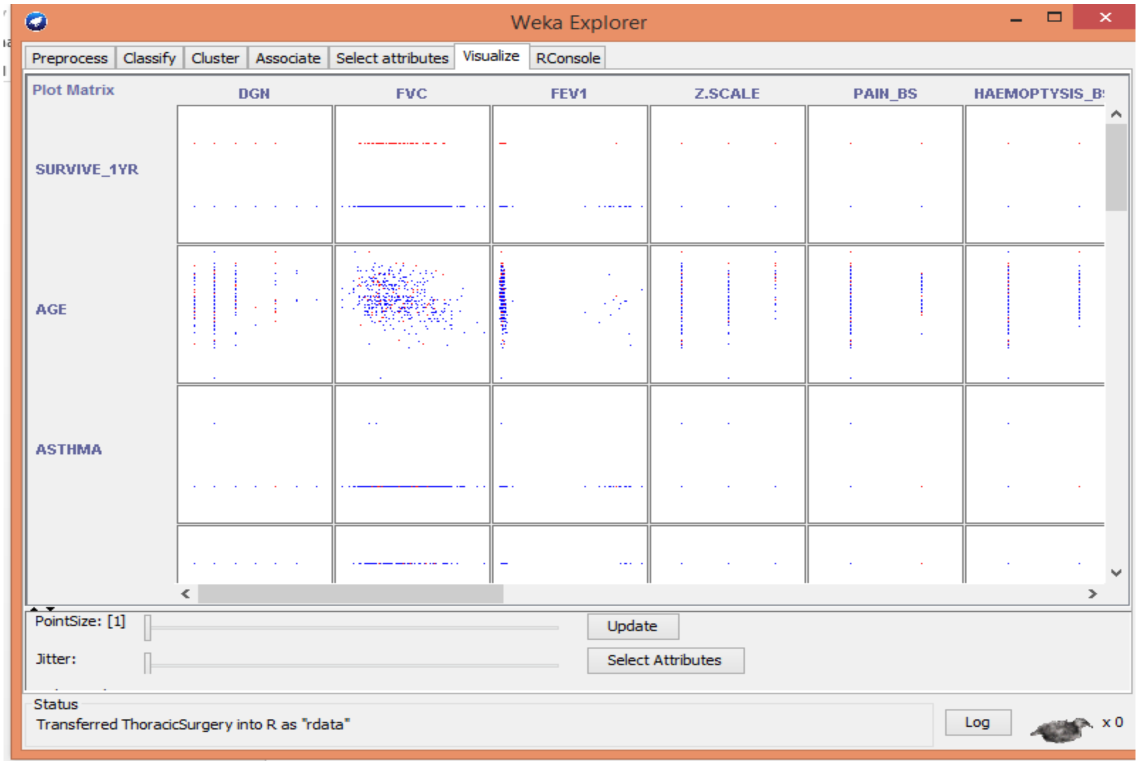
3.3. Data pre-processing and exploration

It is not uncommon to find data sets exhibiting some inconsistencies; therefore, before implementing the data mining algorithms, it is necessary to sanitize the data-set in order to deal with incomplete and noisy entries. This process is known as data pre-processing and it helps to ensure that the knowledge discovery process is easier and more proficient (Dogan and Tanrikulu, 2013). Research has proven that 80% of the data mining effort is usually spent tidying up the data for analysis (Wickham, 2014; Dasu and Johnson, 2003). Occasionally, the process of preparing the data is repeated over the entire course of the analysis and several packages have been developed for various statistical software programs to improve the overall cleaning process.

The steps for this process include cleaning, extraction, integrations and transformations and reductions(Han et al., 2011). Data cleaning ensures that outliers are identified and removed and missing values replaced in order to resolve inconsistencies. Data integration involves the integration of multiple databases in situations where information may need to be merged from different sources. Data reduction involves techniques to reducing the volume of data which would also involve selecting the most relevant features that are proper representative of the data. Discretization is usually done to transform numerical data to categorical data so that the knowledge discovery process can be enhanced.

Efforts to decipher the patterns in a data set by looking at the list of numbers and variables can often be stressful and highly time consuming even for professionals working on a familiar domain. This can easily be alleviated through a productive data exploration exercise. Data exploration involves the use various tools such as of graphs and charts to check for correlations and uncover relationships and hidden patterns within items of a dataset. It is a major technique applied through the whole pre-processing activity. Its advantages are enormous; it gives the researcher the opportunity to understand the properties that the data hold, find patterns in a data set, suggest modelling techniques and strategies, debug analysis and communicate results to other people.

On Weka, the visualize panel in the explorer application is an effective tool for exploring data. Also, the knowledge flow application allows the researcher to create and save charts such as scatter plots, histograms, error plots, ROC curve and the likes however, many of the exploration exercise would be done from the RConsole to overcome the visualization limitations that is encountered in Weka. The boxes in the Weka visualize panel in Figure 3.3 gives a scatter plot of multiple variables typically two variable on the x and y axis and a third variable is coloured to offer more exposition within the data set .



Status
Transferred ThoracicSurgery into R as "rdata"

Figure 3.3 The visualize panel on the Weka Explorer

The panel represented in the Figure 3.4 gives an overview of the patient data and shows plots of the various bar charts and histograms according to the selected class variable. For example, the last bar chart indicates that 70 people died within a year of having a thoracic surgery, represented by the red bar and 400 survived after one year represented by the blue bar.

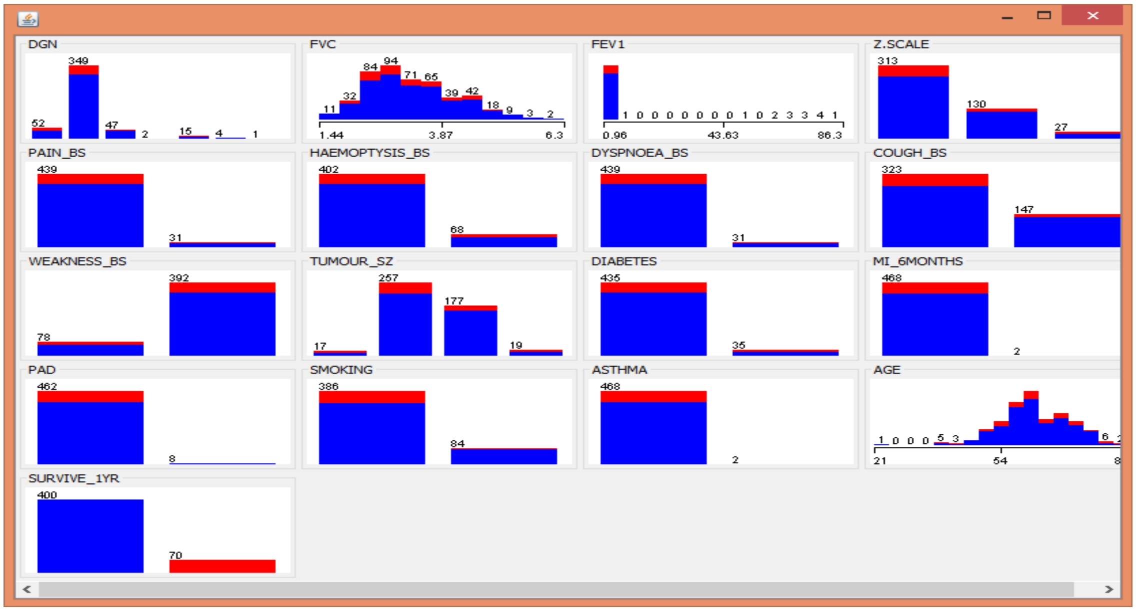


Figure 3.4 The visualize all panel on Weka explorer.

For analysis to be done on the dataset, it is important that the nature of dataset is well understood. The dataset used for this analysis contains information on the post-operative life expectancy of patients with lung cancer within a year and it was divided into 2 classes. Class 1 represents patients that die within one year after thoracic surgery and class 2 represents those that survived.

3.3.1. Renaming attributes and dimensions:

The attributes of the classes were first renamed for ease of analysis and interpretation so that anybody going through the work can easily identify and relate with the analysis performed on the dataset. The following code was run from the RConsole in Weka;

```
Thoracic <- rdata
library(data.table)
setnames(Thoracic,old=c("DGN","PRE4","PRE5","PRE6","PRE7","PRE8","PRE9","PRE10",
"PRE11" , "PRE14", "PRE17","PRE19","PRE25","PRE30","PRE32","AGE","Risk1Yr"),new = c
('DGN','FVC',   'FEV1',   'Z.SCALE',   'PAIN_BS',   'HAEMOPTYSIS_BS','DYSPNOEA_BS',
'COUGH_BS', 'WEAKNESS_BS', 'TUMOUR_SZ', 'DIABETES', 'MI_6MONTHS', 'PAD','SMOKING',
'ASTHMA','AGE', 'RISK_1YR'))
```

It was also necessary to rename the class instances for ease of data interpretation during the analysis.

The ‘TRUE’ and ‘FALSE’ class instances were changed to ‘DIED’ and ‘SURVIVED’ respectively. This is done and saved with the code below;

```
Thoracic$RISK_1YR<- ifelse(Thoracic$RISK_1YR== TRUE, 'DIED', 'SURVIVED');
write.csv(Thoracic,'C:/Users/UserName/Desktop/Thoracic.csv')
```

Table 3.1A -Dataset before renaming.

No.	Attribute	Description	Type
1	DGN	Diagnosis - specific combination of ICD-10 codes for primary and secondary as well multiple tumours if any	Nominal
2	PRE 4	Forced vital capacity FVC	Numeric
3	PRE 5	Volume that has been exhaled at the end of the first second of forced expiration - FEV1	Numeric
4	PRE 6	Performance status - Zubrod scale(PRZ2,PRZ1,PRZ0)	Nominal
5	PRE 7	Pain before surgery(T,F)	Binary
6	PRE 8	Haemoptysis before surgery (T,F)	Binary
7	PRE 9	Dyspnoea before surgery (T,F)	Binary
8	PRE 10	Cough before surgery(T,F)	Binary
9	PRE 11	Weakness before surgery(T,F)	Binary
10	PRE 14	T in clinical TNM - size of the original tumour, from OC11 (smallest) to OC14 (largest)	Nominal
11	PRE 17	Type 2 DM - diabetes mellitus(T,F)	Binary
12	PRE 19	MI (Myocardial infarction)up to 6 months (T,F)	Binary
13	PRE 25	Perioheral arterial Diseases(T,F)	Binary
14	PRE 30	Smoking(T,F)	Binary
15	PRE 32	Asthma (T,F)	Binary
16	AGE	Age at Surgery	Numeric
17	Risk 1Y	1 year survival period - (T)rue value if died (T,F)	Binary

Table 3.1B - Dataset after renaming.

No.	Attribute	Description	Type
1	DGN	Diagnosis - specific combination of ICD-10 codes for primary and secondary as well multiple tumours if any	Nominal
2	FVC	Forced vital capacity FVC	Numeric
3	FEV1	Volume that has been exhaled at the end of the first second of forced expiration - FEV1	Numeric
4	Z.SCALE	Performance status - Zubrod scale(PRZ2,PRZ1,PRZ0)	Nominal
5	PAIN_BS	Pain before surgery(T,F)	Binary
6	HAEMOPTYSIS_BS	Haemoptysis before surgery (T,F)	Binary
7	DYSPNOEA_BS	Dyspnoea before surgery (T,F)	Binary
8	COUGH_BS	Cough before surgery(T,F)	Binary
9	WEAKNESS_BS	Weakness before surgery(T,F)	Binary
10	TUMOUR_SZ	T in clinical TNM - size of the original tumour, from OC11 (smallest) to OC14 (largest)	Nominal
11	DIABETES	Type 2 DM - diabetes mellitus(T,F)	Binary
12	MI_6MONTHS	MI (Myocardial infarction)up to 6 months (T,F)	Binary
13	PAD	Perioheral arterial Diseases(T,F)	Binary
14	SMOKING	Smoking(T,F)	Binary
15	ASTHMA	Asthma (T,F)	Binary
16	AGE	Age at Surgery	Numeric
17	RISK_1YR	1 year survival period indicated as (DIED and SURVIVED)	Binary

The Table 3.1A-B shows the initial and final table after the attributes were renamed. Table 3.1B reflects the names of the attributes that were changed to make interpretation of the parameters easier. This would ease readability so that the contents of the work can be interpreted by non-medical professionals. Also, since there was already many 'True' and 'False' binary attributes, the class attribute was distinguished as 'DIED' and 'SURVIVED' to avoid confusion in the outputted results.

3.3.2. Data cleaning

This involves the removal of missing, incomplete and inconsistent data referred known as dirty data. Techniques in dealing with dirty data include handling of missing values, identification and removal of outliers and correcting inconsistent entries. Owing to the fact that the data was already structured and there was no missing data, much effort was not required in this phase however, a search was done to identify and remove the outliers in the data set.

3.3.2.1. Identification of outliers

According to (Hartwig and Dearing, 1979), "An outlier is a value which lies outside the normal range of the data, i.e., lies well above or well below most, or even all, of the other values in a distribution". In other words, these are unusual occurrences in a dataset which may be caused human errors such as wrong data input leading to inconsistent entries, measurement or execution error due to system default settings, or may even be a genuine like the case of the salary of the head of an organisation having a higher than the average salary of other workers. Also, an outlier can be of utmost priority in data analysis such in the case of fraud detection to depict a fraudulent activity or to find unusual responses to medical treatment (Han et al., 2011). An example can be sighted from an outlier identification program that helped to catch a Marine Corps lance corporal in the US military who made illegal purchases with her pentagon credit card (Gupta and Palmer, 2007).

According to (Horvath and Symonds, 1991) , an outlier can be mathematically written as ;

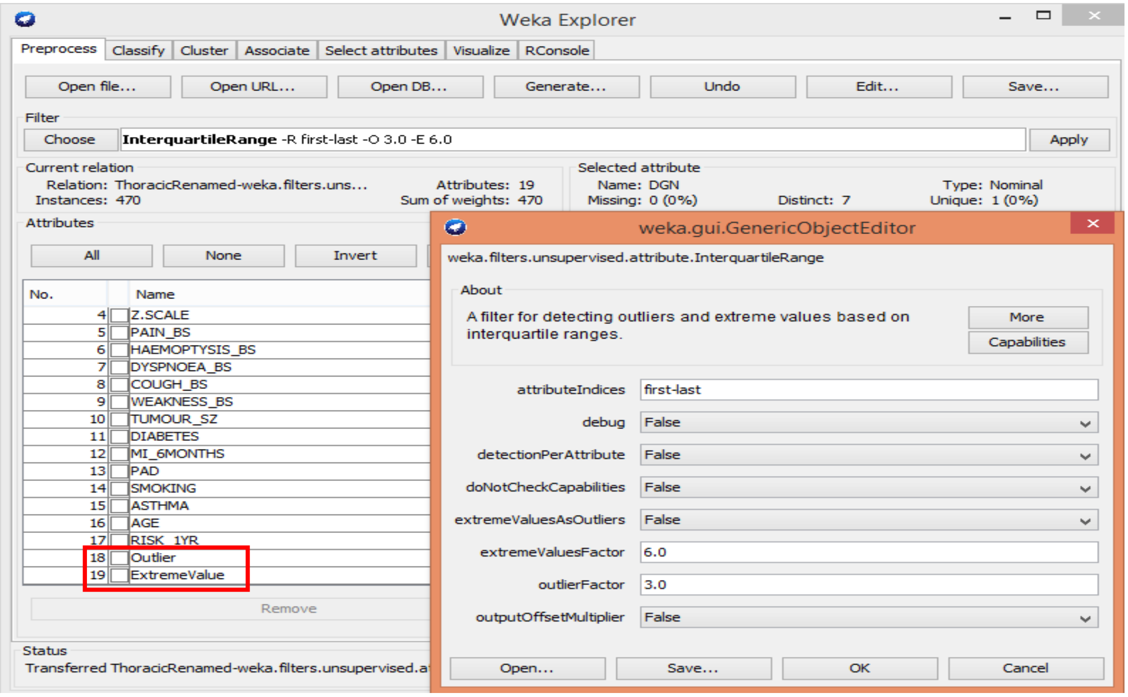
$$Q1 - 1.5(IQR) > \text{Outlier} > Q3 + 1.5(IQR)$$

Where Q1: First quartile which represents 25% of the instances;

Q3: Third quartile which represents 75% of the instances;

IQR: Interquartile range equals **Q3-Q1;**

The process of identifying outliers from the Preprocess tab and RConsole panel is demonstrated below in Figure 3.5A-B. The Weka filter known as the interquartile range filter was used to find the outlier. Weka generated two new attributes namely outlier and extreme values represented by the 18th and 19th attribute respectively in Figure 3.5A while saving and investigating the results obtained revealed the outliers and extreme values as seen in Figure 3.5B



Filter: **InterquartileRange** -R first-last -O 3.0 -E 6.0

Current relation
Relation: ThoracicRenamed-weka.filters.uns...
Instances: 470
Attributes: 19
Sum of weights: 470

Selected attribute
Name: DGN
Missing: 0 (0%)
Distinct: 7
Type: Nominal
Unique: 1 (0%)

Attributes

No.	Name
4	Z.SCALE
5	PAIN_BS
6	HAEMOPTYSIS_BS
7	DYSPNOEA_BS
8	COUGH_BS
9	WEAKNESS_BS
10	TUMOUR_SZ
11	DIABETES
12	MI_6MONTHS
13	PAD
14	SMOKING
15	ASTHMA
16	AGE
17	RISK_1YR
18	Outlier
19	ExtremeValue

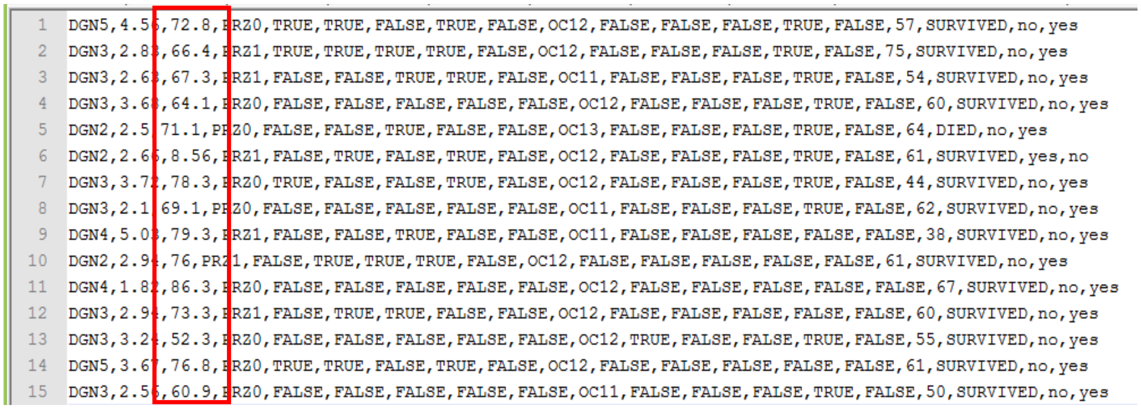
Status
Transferred ThoracicRenamed-weka.filters.unsupervised.a

weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.InterquartileRange

About
A filter for detecting outliers and extreme values based on interquartile ranges.

attributeIndices	first-last
debug	False
detectionPerAttribute	False
doNotCheckCapabilities	False
extremeValuesAsOutliers	False
extremeValuesFactor	6.0
outlierFactor	3.0
outputOffsetMultiplier	False

Figure 3.5A Process of detecting the outliers using WEKA.

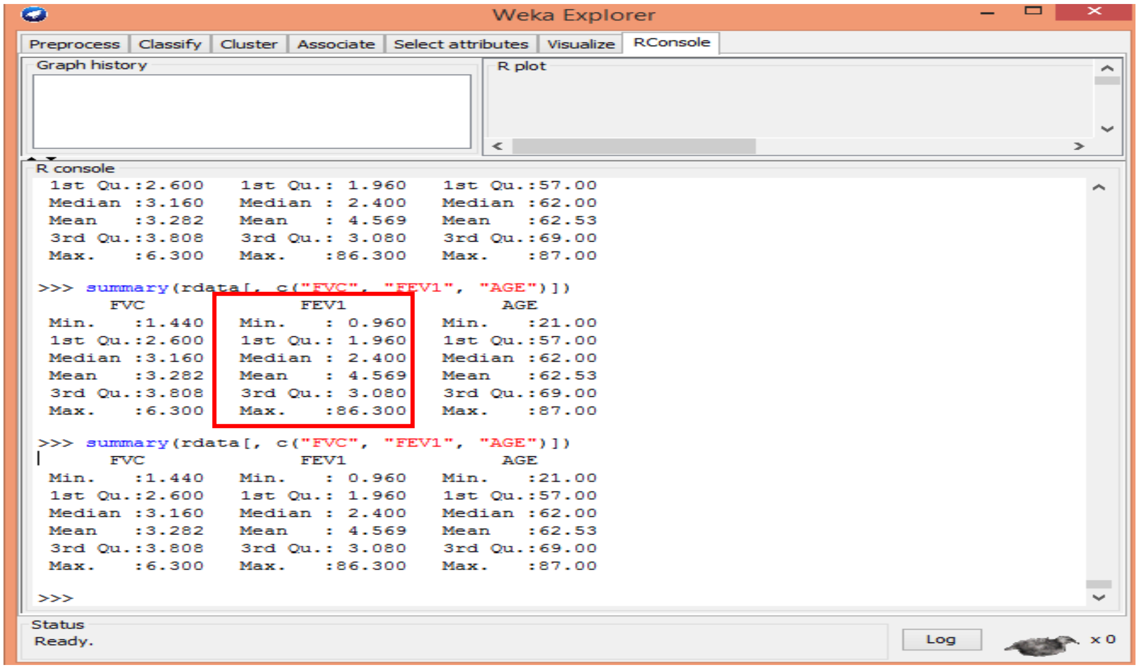


```
DGN5,4.56,72.8,PRZ0,TRUE,TRUE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,57,SURVIVED,no,yes
DGN3,2.88,66.4,PRZ1,TRUE,TRUE,TRUE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,75,SURVIVED,no,yes
DGN3,2.63,67.3,PRZ1,FALSE,FALSE,TRUE,TRUE,FALSE,OC11,FALSE,FALSE,FALSE,TRUE,FALSE,54,SURVIVED,no,yes
DGN3,3.68,64.1,PRZ0,FALSE,FALSE,FALSE,FALSE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,60,SURVIVED,no,yes
DGN2,2.5,71.1,PRZ0,FALSE,FALSE,TRUE,FALSE,FALSE,OC13,FALSE,FALSE,FALSE,TRUE,FALSE,64,DIED,no,yes
DGN2,2.66,8.56,PRZ1,FALSE,TRUE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,61,SURVIVED,yes,no
DGN3,3.72,78.3,PRZ0,TRUE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,44,SURVIVED,no,yes
DGN3,2.1,69.1,PRZ0,FALSE,FALSE,FALSE,FALSE,FALSE,OC11,FALSE,FALSE,FALSE,TRUE,FALSE,62,SURVIVED,no,yes
DGN4,5.03,79.3,PRZ1,FALSE,FALSE,TRUE,FALSE,FALSE,OC11,FALSE,FALSE,FALSE,FALSE,FALSE,38,SURVIVED,no,yes
DGN2,2.94,76,PRZ1,FALSE,TRUE,TRUE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,FALSE,FALSE,61,SURVIVED,no,yes
DGN4,1.82,86.3,PRZ0,FALSE,FALSE,FALSE,FALSE,FALSE,OC12,FALSE,FALSE,FALSE,FALSE,FALSE,67,SURVIVED,no,yes
DGN3,2.94,73.3,PRZ1,FALSE,TRUE,TRUE,FALSE,FALSE,OC12,FALSE,FALSE,FALSE,FALSE,FALSE,60,SURVIVED,no,yes
DGN3,3.24,52.3,PRZ0,FALSE,FALSE,FALSE,FALSE,FALSE,OC12,TRUE,FALSE,FALSE,TRUE,FALSE,55,SURVIVED,no,yes
DGN5,3.67,76.8,PRZ0,TRUE,TRUE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,FALSE,FALSE,61,SURVIVED,no,yes
DGN3,2.56,60.9,PRZ0,FALSE,FALSE,FALSE,FALSE,FALSE,OC11,FALSE,FALSE,FALSE,TRUE,FALSE,50,SURVIVED,no,yes
```

Figure 3.5B Editing the extreme values and outlier output.

Because an identified outlier may be genuine (Dattero et al., 1991), precaution was taken in order to determine attributes that would qualify to be labelled as an outlier in a data set. The values in Figure 3.5B shows that the outliers generated was from a numeric attribute therefore, looking for the summary of the all the numeric attributes in the data to determine what the possible outlier may be, the R code below was written to give the output in figure 3.5C;

```
summary(Thoracic)
```



```
 1st Qu.:2.600   1st Qu.: 1.960   1st Qu.:57.00
 Median :3.160   Median : 2.400   Median :62.00
 Mean   :3.282   Mean   : 4.569   Mean   :62.53
 3rd Qu.:3.808   3rd Qu.: 3.080   3rd Qu.:69.00
 Max.   :6.300   Max.   :86.300   Max.   :87.00

>>> summary(rdata[, c("FVC", "FEV1", "AGE")])
      FVC             FEV1              AGE
 Min.   :1.440   Min.   : 0.960   Min.   :21.00
 1st Qu.:2.600   1st Qu.: 1.960   1st Qu.:57.00
 Median :3.160   Median : 2.400   Median :62.00
 Mean   :3.282   Mean   : 4.569   Mean   :62.53
 3rd Qu.:3.808   3rd Qu.: 3.080   3rd Qu.:69.00
 Max.   :6.300   Max.   :86.300   Max.   :87.00

>>> summary(rdata[, c("FVC", "FEV1", "AGE")])
      FVC             FEV1              AGE
 Min.   :1.440   Min.   : 0.960   Min.   :21.00
 1st Qu.:2.600   1st Qu.: 1.960   1st Qu.:57.00
 Median :3.160   Median : 2.400   Median :62.00
 Mean   :3.282   Mean   : 4.569   Mean   :62.53
 3rd Qu.:3.808   3rd Qu.: 3.080   3rd Qu.:69.00
 Max.   :6.300   Max.   :86.300   Max.   :87.00

>>>
```

Figure 3.5C Detecting the outlier from the RConsole.

The mean, median and mode from the Rconsole give indications that the median is a better measure for central tendency because the outlier will overestimate the mean and produce a biased result. The volume that has been exhaled at the end of the first second of forced expiration (FEV1) has a median of 2.4, a third quartile of 3.0 but the maximum value of 86.3 brings suspicion about the entries in this range. Also from Figure 3.5B, it is obvious that the highlighted FEV1 values from 1 to 15 are outliers because of the median and third quartile values, however, an interesting value was observed on the sixth row with FEV1 value of 8.56. More research needed to be done in order to qualify this variable as an outlier.

As mentioned by (Han et al., 2011) , data visualization could help with detecting outliers especially in a low dimensional dataset since the human eyes are quick and effective in noticing inconsistent patterns. Plotting a box plot or a line graph may be used to view the outlier in the data set. Figure 3.6A gives the dispersion of the forced expiratory volume (FEV1) with respect to the age of the patients. A log attribute was included to get a clearer view of the spread. The outliers are noticed to be separated from the cluster of instances. A spread of all the instances according to the age distribution is indicated on the y-axis while the red line depicts the mean or average point of the FEV1 with respect to the ages. It is observed that the red line peaked at about three points. This is because FEV1 was under-represented for those ages. This is further depicted in Figure 3.6B with the addition of a green line to represent median. In a situation where we have a cluster of instances, the outlier is seen to move the red line which represents the mean upwards;

```
library(ggplot2)
ggplot(Thoracic, aes(AGE, FEV1)) + geom_point(color= 'orange')+geom_line(stat =
'summary', fun.y = mean, col= 'red')+ scale_x_continuous(breaks=seq(0,100,5)) +
scale_y_log10()
```

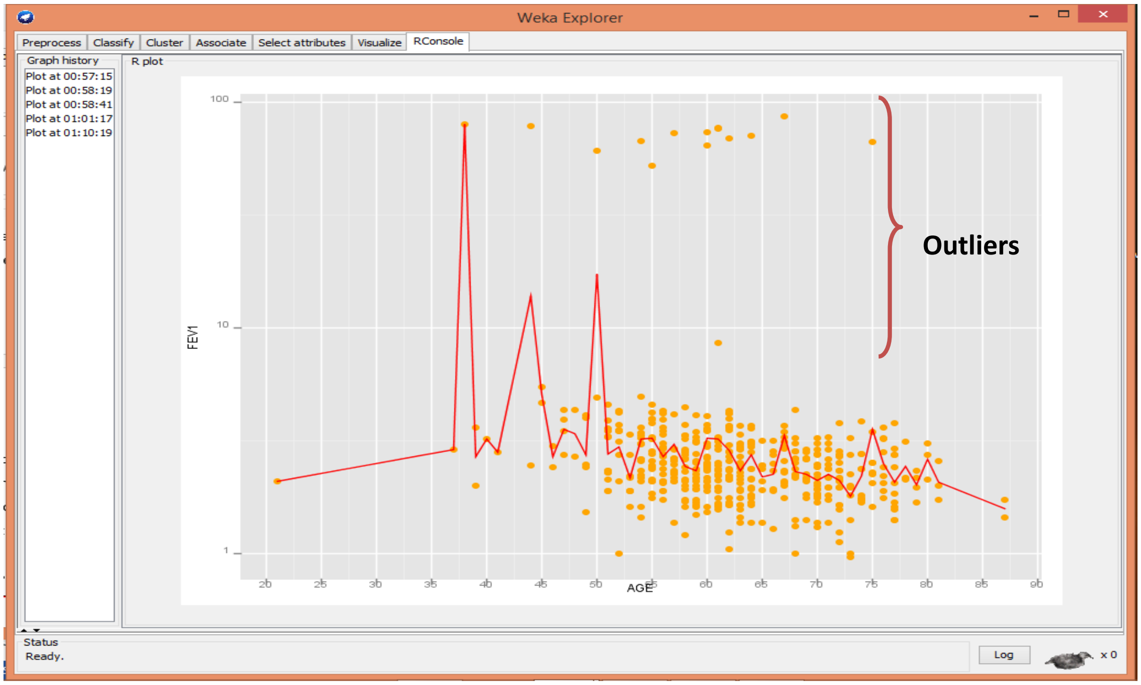


Figure 3.6A A line graph of the mean distribution of the outliers.

```
library(gridExtra);
b1<-ggplot(Thoracic, aes(AGE, FEV1)) + geom_point(color= 'grey')+geom_line(stat =
'summary', fun.y = mean, col='red')+geom_line(stat = 'summary', fun.y = median,
col= 'green') + theme_bw();
b2 <-ggplot(Thoracic, aes(AGE, FVC)) + geom_point(col='grey')+geom_line(stat =
'summary', fun.y = mean, col='red')+geom_line(stat = 'summary', fun.y = median,
col= 'green') + theme_bw();
grid.arrange(b1,b2)
```

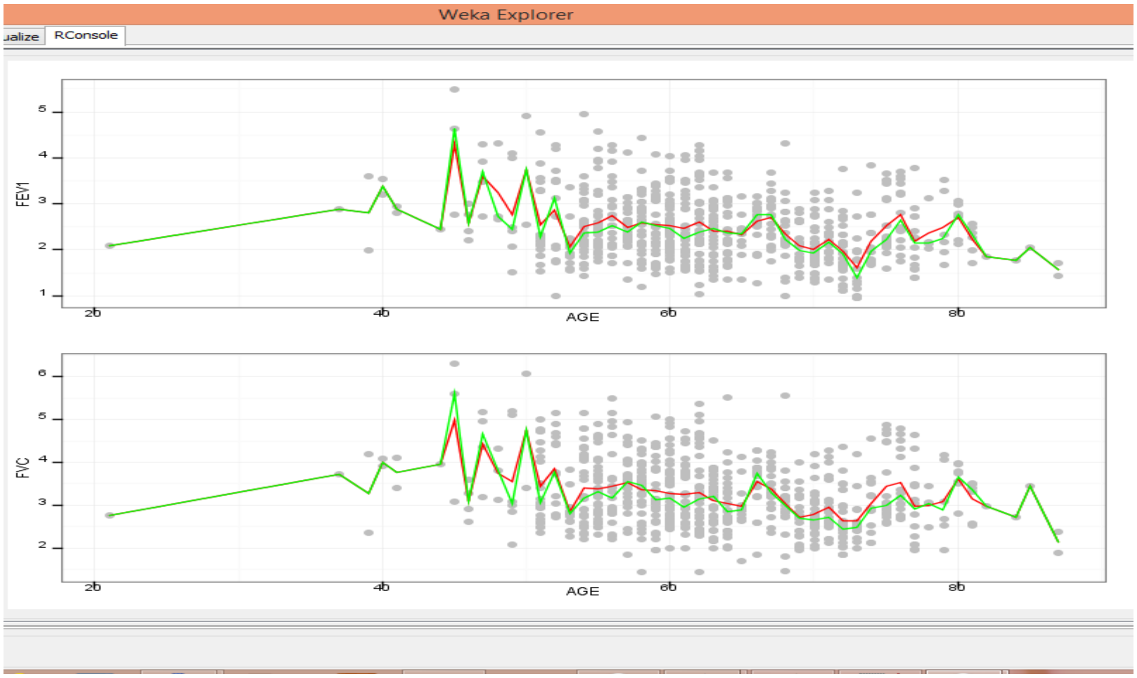


Figure 3.6B A line graph of the mean and median distribution of the outliers.

A box plot was plotted to have a better view of this dispersion using the code below:

```
ggplot(Thoracic, aes(RISK_1YR, FEV1))+ geom_boxplot(fill=c('grey','lightblue'),
outlier.colour = "red", varwidth= T)+ scale_y_log10()+stat_summary(fun.y= mean,
geom = 'point', shape=4)
```

The size of the box in the Figure 3.7 is proportional to the weight of the people that survived and died, indicating that the number of people that survived the surgery was greater than the number of people that died. Also, the median is indicated by the middle line dividing the box plot while the mean is indicated by the crossed sign. From this sign, we can see that the outliers indicated by the red dots have pushed the centre above the median.

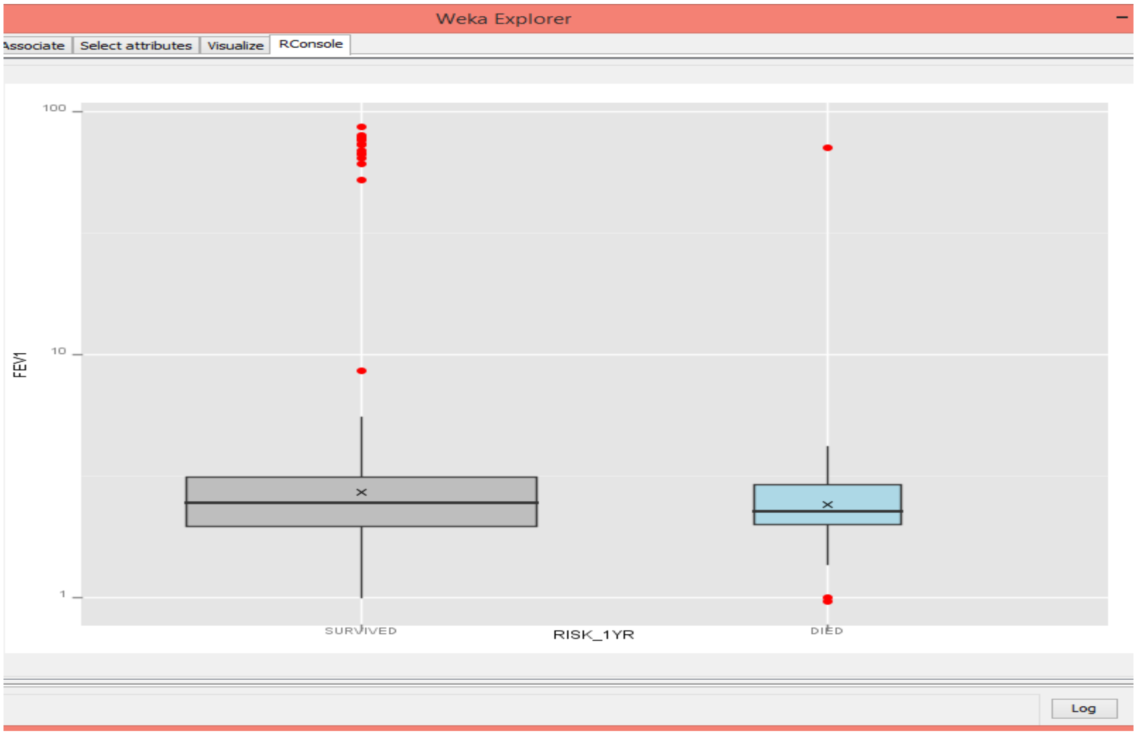
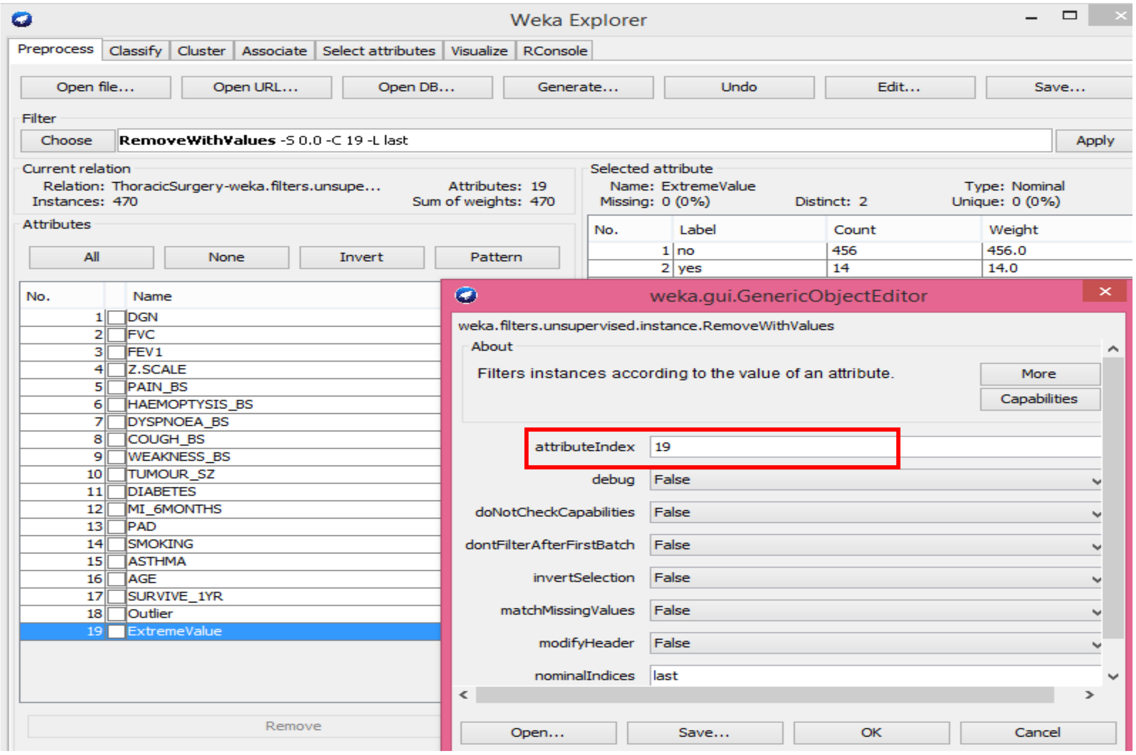


Figure 3.7 Box plot of FEV1 versus RISK_1YR

3.3.2.2. Removal of outliers

To justify the outlier values, analysis was done on the dataset by sorting the FEV1 column in ascending order and it was discovered that for each of the remaining 455 instances the FVC results were greater than their corresponding FEV1 results. Conversely, for the 15 outliers, the least FEV1 value were over 200 percent higher than the corresponding FVC value. The fact that WebMD, a health journal specified the range of the FEV1/FVC ratio for a normal adult as 60-90 percent with a decreasing ratio for aged people and people with an obstructive lung disease further confirms these value as outliers. Finally, the extreme values and outliers generated by Weka was deleted leaving the remaining 455 instances for the classification. This process is illustrated in Figure 3.8.



Filter: RemoveWithValues -S 0.0 -C 19 -L last

Current relation
Relation: ThoracicSurgery-weka.filters.unsupe... Attributes: 19
Instances: 470 Sum of weights: 470

Selected attribute
Name: ExtremeValue Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	no	456	456.0
2	yes	14	14.0

Attributes

No.	Name
1	DGN
2	FVC
3	FEV1
4	Z.SCALE
5	PAIN_BS
6	HAEMOPTYSIS_BS
7	DYSPNOEA_BS
8	COUGH_BS
9	WEAKNESS_BS
10	TUMOUR_SZ
11	DIABETES
12	MI_6MONTHS
13	PAD
14	SMOKING
15	ASTHMA
16	AGE
17	SURVIVE_1YR
18	Outlier
19	ExtremeValue

weka.filters.unsupervised.instance.RemoveWithValues

About
Filters instances according to the value of an attribute.

attributeIndex	19
debug	False
doNotCheckCapabilities	False
dontFilterAfterFirstBatch	False
invertSelection	False
matchMissingValues	False
modifyHeader	False
nominalIndices	last

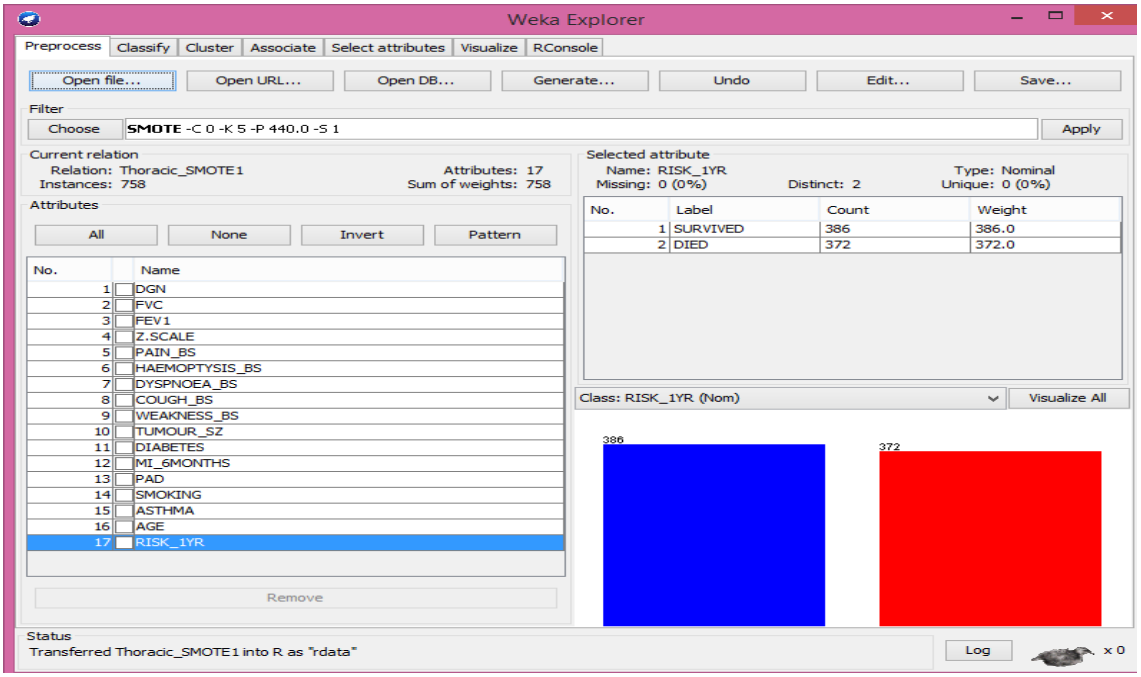
Figure 3.8 Deletion of the outliers and extreme values in Weka.

3.3.3. Balancing the classes

An imbalanced dataset is one of the biggest problems in classification especially in cases where the minority class carries more value than the majority class (Sahin et al., 2013). Difference in classification levels could cause overfitting problems and underperformance of the classification algorithms as it may become biased towards the majority class especially in a high dimensional data (Blagus and Lusa, 2013). To ameliorate this situation, oversampling on the minority data or under sampling on the majority data is a reasonable option. Also, cost sensitive learning could be done when the algorithms are applied especially in cases where the consequence of misclassifying on a minority class attribute would cost more than misclassifying a majority class attribute.

Weka provides the options for these three approaches but considering the peculiarity of this data set, oversampling technique is most preferred because it is a small dataset and contains no missing values. Cost sensitive learning would have also proved as a very good approach but more research efforts from medical professionals may be needed to assign the correct cost metric (i.e. penalties for misclassification) to each class attribute as this is often unknown apriori for a given data set (Pengyi Yang et al., 2009b).

To avoid random oversampling which would be less effective, the synthetic minority oversampling technique otherwise known as the SMOTE technique was applied on the Weka experiment interface. Research has proven than SMOTE performs better than other simple oversampling techniques (Blagus and Lusa, 2013) and it is widely used in bioinformatics for gene prediction, species distribution prediction, etc. SMOTE balances a data set by using the information gained from the data to create synthetic minority class samples. It further identifies samples with the smallest Euclidean distance known as the nearest neighbours and randomly chooses one of them as instances. On Weka, the minority sample was increased by 440 percent and the default 5 nearest neighbours was applied. This process is illustrated in Figure 3.9A-B.



Filter
SMOTE -C 0 -K 5 -P 440.0 -S 1

Current relation
Relation: Thoracic_SMOTE1
Instances: 758
Attributes: 17
Sum of weights: 758

Selected attribute
Name: RISK_1YR
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	SURVIVED	386	386.0
2	DIED	372	372.0

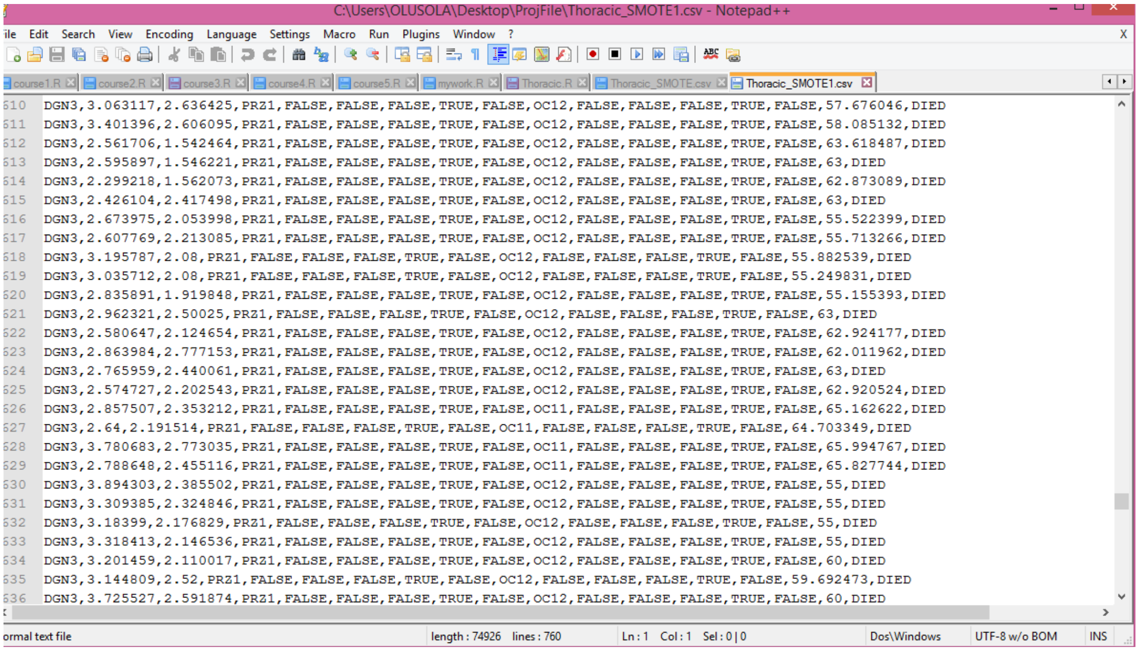
Class: RISK_1YR (Nom)

Attributes

No.	Name
1	DGN
2	FVC
3	FEV1
4	Z.SCALE
5	PAIN_BS
6	HAEMOPTYSIS_BS
7	DYSPNOEA_BS
8	COUGH_BS
9	WEAKNESS_BS
10	TUMOUR_SZ
11	DIABETES
12	MI_6MONTHS
13	PAD
14	SMOKING
15	ASTHMA
16	AGE
17	RISK_1YR

Status
Transferred Thoracic_SMOTE1 into R as "rdata"

Figure 3.9A Using SMOTE for oversampling.


C:\Users\OLUSOLA\Desktop\ProjFile\Thoracic_SMOTE1.csv - Notepad++
ile Edit Search View Encoding Language Settings Macro Run Plugins Window ?
course1.R
course2.R
course3.R
course4.R
course5.R
mywork.R
Thoracic.R
Thoracic_SMOTE.csv
Thoracic_SMOTE1.csv
ormal text file
length : 74926 lines : 760
Ln : 1 Col : 1 Sel : 0 | 0
Dos\Windows
UTF-8 w/o BOM
INS

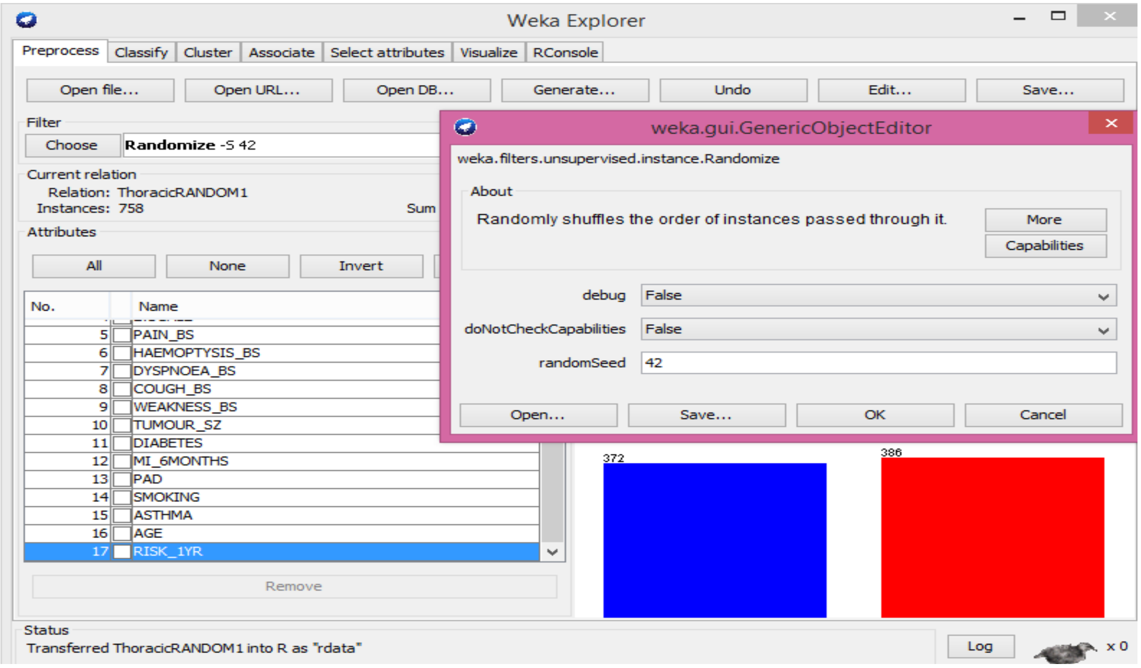
```
DGN3,3.063117,2.636425,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,57.676046,DIED
DGN3,3.401396,2.606095,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,58.085132,DIED
DGN3,2.561706,1.542464,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,63.618487,DIED
DGN3,2.595897,1.546221,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,63,DIED
DGN3,2.299218,1.562073,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,62.873089,DIED
DGN3,2.426104,2.417498,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,63,DIED
DGN3,2.673975,2.053998,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55.522399,DIED
DGN3,2.607769,2.213085,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55.713266,DIED
DGN3,3.195787,2.08,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55.882539,DIED
DGN3,3.035712,2.08,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55.249831,DIED
DGN3,2.835891,1.919848,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55.155393,DIED
DGN3,2.962321,2.50025,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,63,DIED
DGN3,2.580647,2.124654,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,62.924177,DIED
DGN3,2.863984,2.777153,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,62.011962,DIED
DGN3,2.765959,2.440061,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,63,DIED
DGN3,2.574727,2.202543,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,62.920524,DIED
DGN3,2.857507,2.353212,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC11,FALSE,FALSE,FALSE,TRUE,FALSE,65.162622,DIED
DGN3,2.64,2.191514,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC11,FALSE,FALSE,FALSE,TRUE,FALSE,64.703349,DIED
DGN3,3.780683,2.773035,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC11,FALSE,FALSE,FALSE,TRUE,FALSE,65.994767,DIED
DGN3,2.788648,2.455116,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC11,FALSE,FALSE,FALSE,TRUE,FALSE,65.827744,DIED
DGN3,3.894303,2.385502,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55,DIED
DGN3,3.309385,2.324846,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55,DIED
DGN3,3.18399,2.176829,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55,DIED
DGN3,3.318413,2.146536,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,55,DIED
DGN3,3.201459,2.110017,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,60,DIED
DGN3,3.144809,2.52,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,59.692473,DIED
DGN3,3.725527,2.591874,PRZ1,FALSE,FALSE,FALSE,TRUE,FALSE,OC12,FALSE,FALSE,FALSE,TRUE,FALSE,60,DIED
```

Figure 3.9B An edited output of the SMOTE.

After using the SMOTE filter there was an increase of 303 new attributes for patients that died but all the numeric attributes were rounded up to six decimal places as indicated in Figure 3.9B. This is undesirable as it may compromise the behaviour of the applied algorithms. The FVC and FEV1 attributes was therefore rounded up to two decimal places while the age attribute was rounded up to a single decimal place from the RConsole and saved with the code below:

```
Thoracic$AGE <- floor(Thoracic$AGE);
Thoracic$FVC <- round(Thoracic$FVC, 2);
Thoracic$FEV1 <- round(Thoracic$FEV1, 2)
write.csv(Thoracic,'C:/Users/UserName/Desktop/Thoracic.csv')
```

From the edited values in Figure 3.9B, it was also observed that the new instances generated from the SMOTE exercise were clustered together so randomization was done on the set of instances to avoid underperformance and overfitting of the classification algorithms. The randomization process is illustrated in Figure 3.10.



Filter

Randomize -S 42

Current relation

Relation: ThoracicRANDOM1

Instances: 758

Attributes

No.	Name
5	PAIN_BS
6	HAEMOPTYSIS_BS
7	DYSPNOEA_BS
8	COUGH_BS
9	WEAKNESS_BS
10	TUMOUR_SZ
11	DIABETES
12	MI_6MONTHS
13	PAD
14	SMOKING
15	ASTHMA
16	AGE
17	RISK_1YR

weka.gui.GenericObjectEditor

weka.filters.unsupervised.instance.Randomize

About

Randomly shuffles the order of instances passed through it.

debug False

doNotCheckCapabilities False

randomSeed 42

Status

Transferred ThoracicRANDOM1 into R as "rdata"

Figure 3.10. Randomizing the dataset.

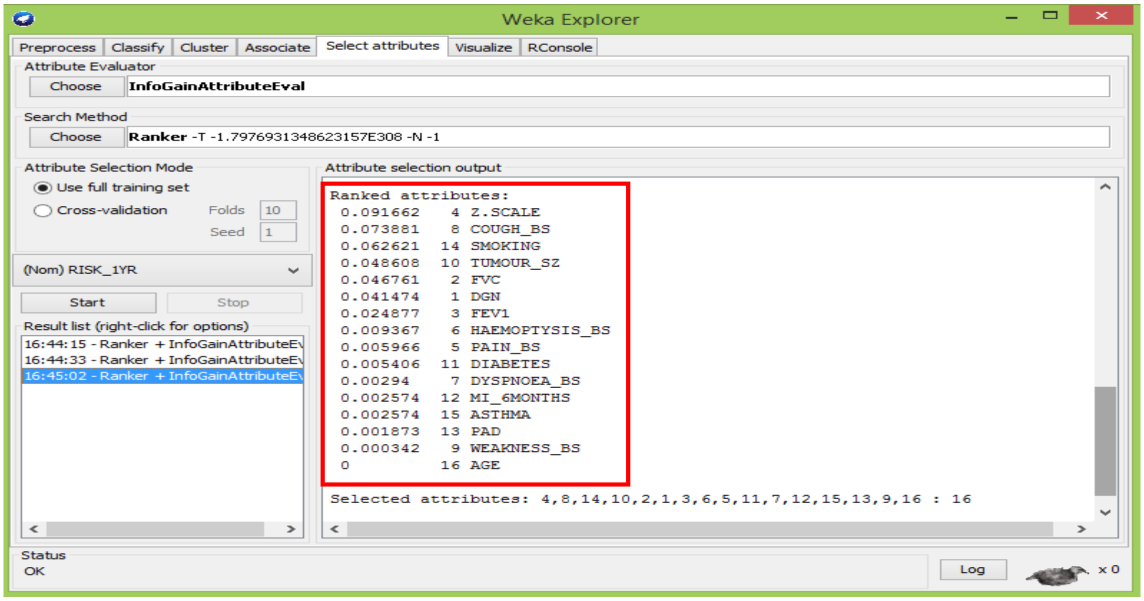
3.3.4. Data reduction and projection

Data reduction and projection otherwise known as feature selection involves steps taking to find relevant features in the data set (Fayyad et al., 1996). Feature selection is a pre- processing step of data mining that is introduced to sieve out variables that are redundant and do not properly represent the dataset for prediction purposes (Rangarajan, 2010). This has become very important in today's machine learning process especially with the advent of high-dimensional data encountered in medical domains such as medicine and biology which makes general classification method difficult (Karegowda et al., 2010). It helps the researcher to improve the performance of the data mining process in terms of prediction accuracy and reduce computation time. It also helps to improve the knowledge discovery process in machine learning. For unsupervised learning, feature selection is used to discover the subset that best describes the data according to some given criteria while for supervised learning, this is done to maximize the classification accuracy. This dataset is of low-dimensionality with just 17 attributes and as a result, feature selection may not have much relevance in terms of reducing the attributes for better classifier performance however, it would help in gaining more understanding of the most relevant attributes in the dataset that may need more exploration through data visualization.

There are two major ways to achieve feature selection for data mining. These are the Individual evaluation or feature ranking approach in which individual features are ranked based on their weights and degree of relevance, and the subset evaluation in which search strategies are employed in order to produce feature subsets (Bolon-Canedo et al., 2013). In addition to these approaches, researchers have suggested other methods such as the wrapper method, the filter method, the embedded method, the hybrid method which combines the filter and wrapper. At this stage, the filter method would be explored while the wrapper method would be used at a later stage when the final classifier has been determined.

3.3.4.1. The Filter Method

Feature selection is done based on the characteristics of each feature in the dataset using a ranker (Rangarajan, 2010). Compared to the wrapper method, the filter method has a lower computational cost and is faster because it ranks classes independent of classifiers which may be a setback on the results generated. Some filter methods do not select the features after ranking, hence, they need to be combined with a search method that adopts selection methods such as forward selection, backward elimination, bi- directional search, best-first search, genetic search, etc. to rank attributes (Karegowda et al., 2010). The Infogain attribute evaluator was applied to rank the attributes. The output is highlighted below in descending order with the best attribute having the highest rank;



Attribute Evaluator
InfoGainAttributeEval

Search Method
Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode
● Use full training set
○ Cross-validation Folds 10 Seed 1

(Nom) RISK_1YR

Result list (right-click for options)

```
16:44:15 - Ranker + InfoGainAttributeEv
16:44:33 - Ranker + InfoGainAttributeEv
16:45:02 - Ranker + InfoGainAttributeEv
```

Attribute selection output

```
Ranked attributes:
 0.091662    4 Z.SCALE
 0.073881    8 COUGH_BS
 0.062621   14 SMOKING
 0.048608   10 TUMOUR_SZ
 0.046761    2 FVC
 0.041474    1 DGN
 0.024877    3 FEV1
 0.009367    6 HAEMOPTYSIS_BS
 0.005966    5 PAIN_BS
 0.005406   11 DIABETES
 0.00294     7 DYSPNOEA_BS
 0.002574   12 MI_6MONTHS
 0.002574   15 ASTHMA
 0.001873   13 PAD
 0.000342    9 WEAKNESS_BS
 0          16 AGE

Selected attributes: 4,8,14,10,2,1,3,6,5,11,7,12,15,13,9,16 : 16
```

Status
OK

Figure 3.11. InfoGain feature selection

It is discovered that the ranker has ranked Z.SCALE as the most relevant attribute while COUGH_BS and SMOKING followed in the second and third place respectively. According to the ranker, this means that when taking a decision of the cause of survival and death of patients that have undergone thoracic surgery, these three attributes are of greater influence.

3.4. Graphics and exploratory data analysis.

This is an important step for getting familiar with the data set. In August 2015, WhaTech a global information technology resource released a medical market report indicating that most of the healthcare institutions place high value on data visualization, historic trend analysis, standardized reporting, forecasting, simulations and scenario development. Data exploration usually involves a graphical transformation of variables into various distributional displays like summary plots i.e. bar charts, pie charts and histograms; time series plots i.e. univariate and multivariate plots; geographical plots i.e. maps and projection maps, three-dimensional plots and simulation plots (Kukuyeva, 2009). Focusing on the best ranked features, various kinds of graphs are used to explore more information within the data set with the aim of having more understanding of the relationship within the data. This can also help in determining whether a variable needs to be transformed to meet some statistical requirements. Taking visualizations of the top five ranked attributes, we have the following representations;

3.4.1. Histograms

This is a special form of bar chart that is used for plotting continuous variables i.e. undiscretized variables. It implies that the spaces between the bars of a normal bar char is absent in a histogram. For example, Figure 3.12 shows a histogram of the frequency of the smokers according to their ages, revealing the ages with the highest number of smokers. This is generated from the R script below;

```
a <- ggplot(Thoracic, aes(AGE))+geom_histogram(aes(fill= SMOKING), binwidth=1,col=
'grey')+ scale_x_continuous(breaks =seq(0,100,2));
b <- ggplot(Thoracic, aes(AGE))+geom_histogram(aes(fill= COUGH_BS), binwidth=1,col=
'grey')+ scale_x_ continuous (breaks =seq(0,100,2));
library(gridExtra);
grid.arrange(a,b, ncol= 1)
```

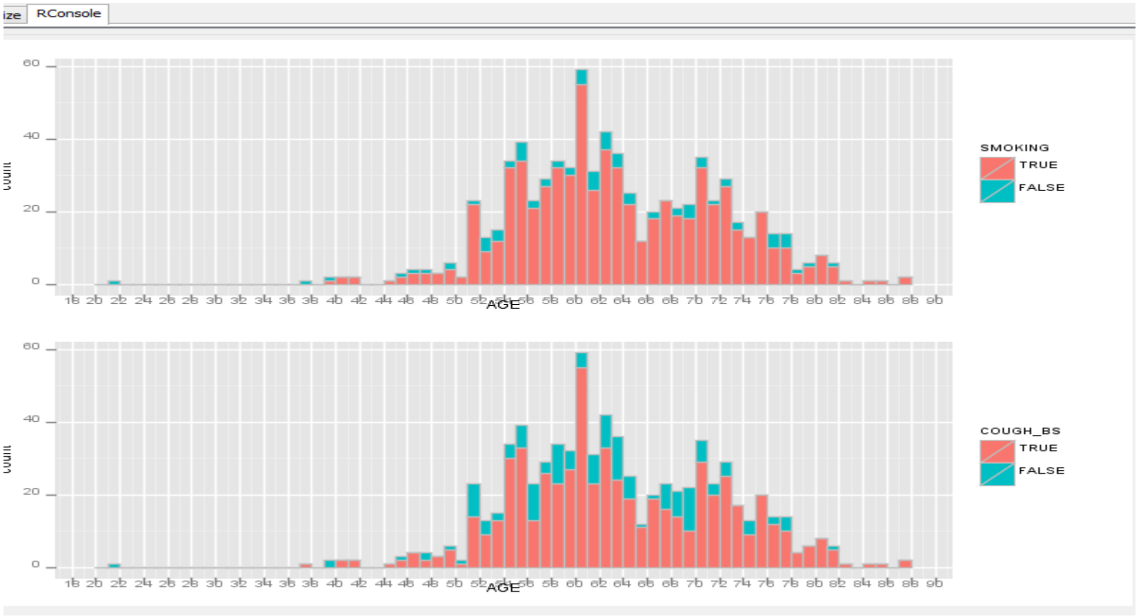


Figure 3.12 Histogram showing the distribution of the SMOKING AND COUGH_BS attribute according to AGE.

The red bars in the graph in Figure 3.12 indicates that an almost similar distribution exists for patients that smoke and have cough. It can therefore be concluded that that patients that smoke are most likely to have cough.

3.4.2. Pie Charts.

Pie chart is used to display frequencies as percentages of instances in each category. Though, it is said to be a bad way of displaying information (Womack, 2014; Muenchen, 2011), their use can be improved especially when it is properly labelled on a discretized variable. Plotting a pie chart for the first and fourth attribute was achieved by the R code below from the Weka RConsole panel.

```
Thoracicnew <- Thoracic$Z.SCALE;
Thoracicnew<-data.frame(Thoracicnew);
library(data.table);
setnames(Thoracicnew, old = c("Thoracicnew"),new = c('Z.SCALE'));
Thoracicnew$Z.SCALE <- reorder(Thoracicnew$Z.SCALE, X = Thoracicnew$Z.SCALE, FUN =
function(x) -length(x));
```

```
at <- nrow(Thoracicnew) - as.numeric(cumsum(sort(table(Thoracicnew)))-
0.5*sort(table (Thoracicnew)));
label=paste0(round(sort(table(Thoracicnew))/sum(table(Thoracicnew)),2) * 100,"%");

q <-ggplot(Thoracicnew, aes(x = factor(1), fill = factor(Z.SCALE))) +
geom_bar(width= 1)+
coord_polar(theta = "y")+ annotate(geom ="text", y = at, x = 1, label= label);

Thoracicnew2 <- Thoracic$TUMOUR_SZ;
Thoracicnew2<-data.frame(Thoracicnew2);
library(data.table);
setnames(Thoracicnew2, old = c("Thoracicnew2"),new = c('TUMOUR_SZ'));
Thoracicnew2$TUMOUR_SZ <- reorder(Thoracicnew2$TUMOUR_SZ, X =
Thoracicnew2$TUMOUR_SZ, FUN = function(x) -length(x));

at <- nrow(Thoracicnew2) - as.numeric(cumsum(sort(table(Thoracicnew2)))-0.5*sort(
table (Thoracicnew2)));
label=paste0(round(sort(table(Thoracicnew2))/sum(table(Thoracicnew2)),2) *
  100,"%");

p <-ggplot(Thoracicnew2, aes(x = factor(1), fill = factor(TUMOUR_SZ))) + geom_bar
(width = 1)+ coord_polar(theta = "y")+ annotate(geom = "text", y = at, x = 1, label
= label);

library(gridExtra);
grid.arrange(q,p, ncol= 2)
```

The chart in figure 3.13 gives a visual display of the percentage distribution of the categories of these variables.

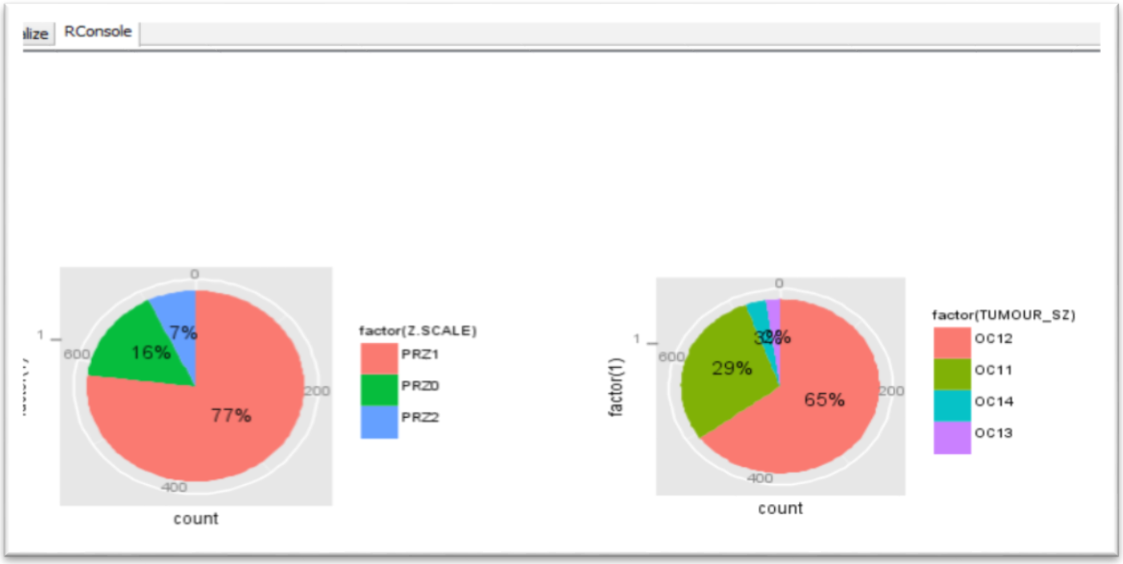


Figure 3.13 A pie chart of the Z.SCALE AND TUMOUR_SZ variables.

The pie charts clearly indicates the highest and lowest percentages of each category of zubrod performance scale and the tumour size of the patients.

3.4.3. Scatter Plots

Scatter plots are very good tools that are used to demonstrate the relationship between continuous variables of a data set, with each variable represented by a point in the graph. When combined with a nominal variable, it functions like a histogram or bar chart. Scatter plots can also be combined with statistical models to show predictions and correlations amongst the variables in a data (Winston Chang, 2012) . For example, plotting the graph of age, the forced vital capacity (FVC) and the volume that has been exhaled at the end of the first second of forced expiration (FEV1) was done with the code below;

```
d1 <-ggplot(Thoracic, aes(AGE, FVC))+geom_point(aes(color=RISK_1YR ));
d2 <-ggplot(Thoracic, aes(AGE, FEV1))+geom_point(aes(color=RISK_1YR ));
grid.arrange(d1, d2 , ncol= 1)
```

Visualizing the distribution of the cluster for variables with the red and green colour indicating the patients that died and survived respectively in Figure 3.14, it can be seen that the dispersion of patients that died seemed to be concentrated around a region while that for the patients that survived were scattered. Taking a deeper look will reveal more patterns such as the values of the FEV1 and FVC that have a higher probability of resulting to death with respect to the age of the patient.

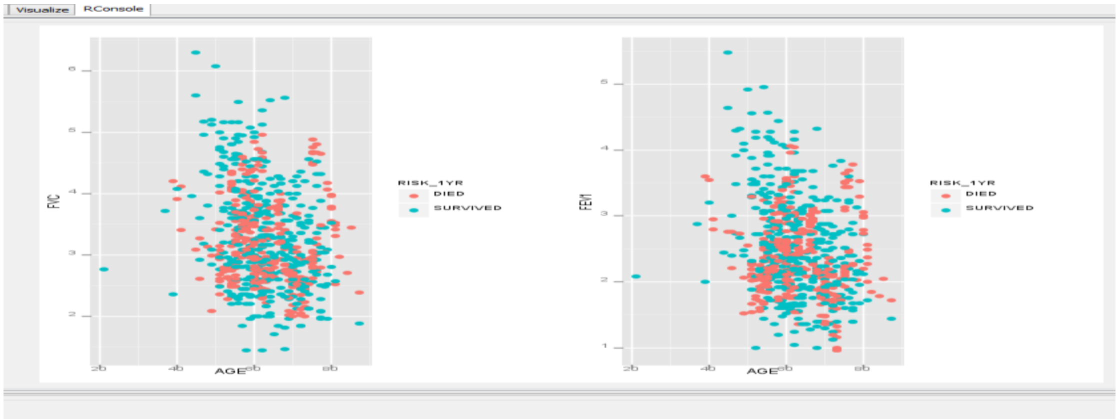


Figure 3.14 A scatter plot of the FVC and FEV1 for the patients.

3.5. Analysis of algorithms by data mining

Machine learning techniques used in data mining includes supervised and unsupervised learning. Supervised learning is used for building models to describe the relations within a dataset while unsupervised learning is used for pattern discovery (Donalek, 2014). Classification is a form of supervised learning used for predictive modelling on nominal variables while regression is applied on numerical attributes. For unsupervised learning, clustering, probability distribution and association rules are used for descriptive modelling to find hidden patterns in a data set. The data set used for this analysis is dedicated to a classification problem. The first approach to applying the classification algorithm will be to check the characteristics and performance of the various classifiers by selection, training and testing in order to choose the best algorithm for building the model (Kuncheva, 2004). Weka has several in-built algorithm designed for this purpose;

3.5.1. Classification

Classification is a very significant area of data mining that has involved a lot of studies in the machine learning and statistics community. It is the most widely used area of data mining and it works by using an algorithm to make predictions for a target class from the information obtained within an independent variable (Freitas, 2003). Classification uses supervised learning algorithms to assign tags to observations or instances known as training set and uses this tag on an unobserved data known as the test set (Dogan and Tanrikulu, 2013). For example, the goal of classification may be to predict a customer as good or bad for obtaining credit facility based on predicting attributes such as the age, salary, account balance, current running facilities etc.

When classification is done, an algorithm is first used to create a data-driven model that learns an unknown function by mapping several input variables to an output target. Classifiers are usually evaluated either by supplying a test set on a classifier model, by a hold out method otherwise known as percentage split and cross- validation technique. Since we do not have an independent test set, the cross validation and percentage hold out test option was used.

3.5.1.1 Cross validation

Cross validation is used for evaluating the performance of a machine learning algorithm. It is a systematic way of improving upon repeated holdouts by reducing the variance of the dataset in order to get more accurate predictions. Generally, cross validation works by randomly dividing input variables into a number of equally sized segments and repeatedly training the applied classifier each time on all but one segment which will be used for validation purposes. This segment is usually denoted by k, thus, in a k-fold cross validation process, k-1 segments are used for training while 1 is used for testing (Granholm et al., 2012). The formula for cross validation is denoted below;

$$\boldsymbol{CV = 1/k \sum_{i=1}^{k} PM_i}$$

Where $\boldsymbol{CV}$:cross-validation, $\boldsymbol{k}$::number of folds , and $\boldsymbol{PM}$: performance measure for each fold (Olson and Delen, 2008).

For example, when a 10 fold cross validation is specified, the data is divided once into 10 segments and each of the pieces is held out in turn for training and testing at a ratio of 9:1 respectively. The process is repeated 10 times, and average result for the 10 times is displayed as the final result for evaluation. Weka does a stratified cross-validation by default in which an 11th round of validation is done but this time on the entire data set and the output result is used for evaluation (Witten and Frank, 2005).

3.5.1.2 Percentage split

The percentage split also known as the simple hold out method works by partitioning a dataset into two mutually exclusive segments known the test set and the training set. The training set is first used to build a model on which the test set would be applied. By default, Weka allocates two-thirds of the data set to the training set and one-thirds to the test set however; there are options for varying the allocation. For artificial neural network algorithms, the dataset is usually partitioned into three mutually exclusive segments known as the training, validation and test set (Olson and Delen, 2008).

3.5.2. Using single classifiers.

We have different learning schemes for making predictions for numeric and categorical classes but since we are dealing with categorical classes, the focus was on algorithms that work with them. Weka is known to have a variety of classification algorithms and its classification characteristic is generally recognised as one of the major strength of Weka (Achtert et al., 2009). The first step in classification employed was to check the performance of various algorithms provided by the Weka for nominal classes. The algorithms employed includes , ZeroR for baseline accuracy, One R, Naïve Bayes, Bayes Net, J48, Ibk, PART, Random tree, PART, ADTree, conjunctive Rule, decision table and support vector machines (SMO and LibSVM). Their characteristics are explained below;

3.5.2.1 Characteristics of the single classifiers.

(a) IBk: This is known as Instance Based learning and it works by choosing a majority class form several k neighbours. It uses a similarity function such as the Euclidean distance to find observations that is most like the training set and does classification based on this. Where more than one neighbour is specified, the prediction of the neighbours are weighed by their distance to the test instance (Witten and Frank, 2005).

(b) PART: This is used for the pattern of building rules from a decision tree learner. This learner work by recursively generating set of rules for training instances and applying it to classify test instances. In Weka, the PART applies a divide and conquer strategy of RIPPER in combination with the decision tree approach of C4.5 to form a partial decision tree for the instances being considered and forms a new rule from the leaf with the largest coverage (Berger et al., 2006).

(c) J48: The J48 is Weka's version of the C 4.5 algorithm and it also implements a decision tree learner. It is one of the most widely applied learning algorithms for classification problems. Its distinctive characteristic also lies in its ability to work with numeric and nominal class attributes, deal sensible with missing data and its pruning option that leverages its ability to deal with noisy data. Like lost decision tree algorithm, it can apply a top-down recursive divide and conquer technique by picking root node attributes and splitting the instances into subsets using a recursive repeated process for each branch, (Witten and Frank, 2005).

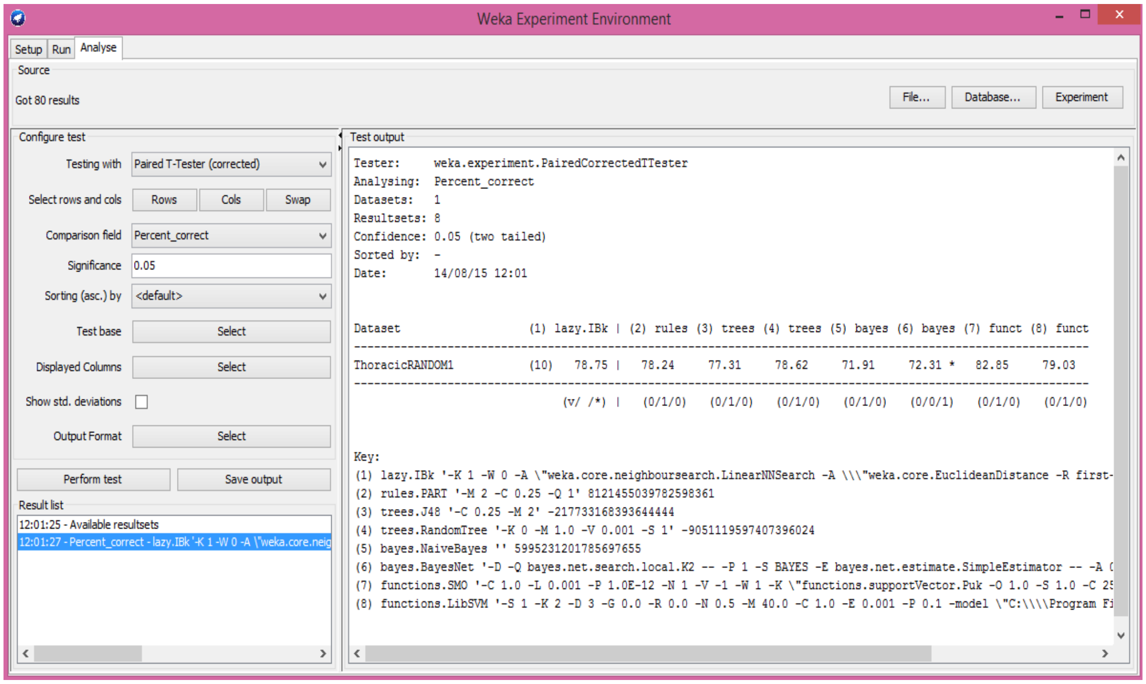
(d) Random tree: This algorithm can deal with both classification and regression i.e. classification of numeric class problem. It constructs a tree using a K- randomly chosen attribute at the tree nodes. It can be considered as a subset of random forest and bagging ensembles of a random tree would produce a random forest classification (Breiman, 2001).

(e) Naive Bayes: This algorithm applies the probabilistic Naïve Bayes classifier by assuming that a conditional independence exists between attributes given a particular class value. Thus, it achieves the overall class probability by multiplying each attribute conditional probabilities together taking the prior probabilities of the class into consideration. Implementing the Naïve Bayes classifier on Weka would require the discretization which involves changing numeric values to nominal or factor variables. This is achieved by setting 'useSupervisedDiscretization' to True which would result in a higher receiver operating characteristic area and a more comprehensive visualization (Witten et al., 1999).

(f) Bayes Net: Like the Naïve Bayes algorithm, when the Bayesian network (Bayes Net) is applied for classification problems in Weka, all the variables need to be discretized by applying the class 'weka.unsupervised.attribute.discretize' function. It also assumed that the instance has no missing value (Bouckaert, 2004). It has four different algorithms namely the local score metrics, the conditional independence tests, the global score metrics and the fixed structure for assessing the conditional probabilities tables of the network. The result of a classifer can be improved by applying a booster algorithm that has the capaicty to boost the performance of the algorithm that was used. Usually, such algorithm works by comibining classifiers and Weka provides this option from the meta classifier and tree options.

(g) Support Vector Machines: Abbreviated as SVM, this is an advanced machine learning algorithm developed by (Cortes and Vapnik, 1995) for binary classifiactions. An advantage of the SVM is that though it constructs complex models, it is simple enough to be analysed mathemetically (Hearst et al., 1998) . It works by detecting the optimal seperating hyperplane also known as a boudary visualizer in Weka between two classes and then searches out for two critical points from each classs known as support vectors . A perpedicular line links the support vectors and the middle of the line become the optimal seperating hyperplane (Meyer and Wien, 2014). An advantage of the SVM is that It is very resilient to overfitting because it doesn't depend on all the points in the data set but a few critical points known as the support vectors. Weka has some functions that executes the SVM such as the sequential minimal optimization (SMO) developed by (Platt, 1999) which is restricted to two classes and the faster and more sophisticated LibSVM developed by (Chih-Chung Chang and Lin, 2011).

The process of employing these classifers is illustrated in Figure 3.15 from the Weka experiment environment;



Source

Got 80 results

Configure test

Testing with: Paired T-Tester (corrected)

Select rows and cols: Rows | Cols | Swap

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Test base: Select

Displayed Columns: Select

Show std. deviations: □

Output Format: Select

Result list

12:01:25 - Available resultsets

12:01:27 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig

Test output

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 8
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       14/08/15 12:01


Dataset                   (1) lazy.IBk | (2) rules (3) trees (4) trees (5) bayes (6) bayes (7) funct (8) funct
--------------------------------------------------------------------------------------------------------------
ThoracicRANDOM1           (10)   78.75 |   78.24     77.31     78.62     71.91     72.31 *   82.85     79.03
--------------------------------------------------------------------------------------------------------------
                               (v/ /*) |   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)   (0/0/1)   (0/1/0)   (0/1/0)


Key:
(1) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-
(2) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
(3) trees.J48 '-C 0.25 -M 2' -217733168393644444
(4) trees.RandomTree '-K 0 -M 1.0 -V 0.001 -S 1' -9051119597407396024
(5) bayes.NaiveBayes '' 5995231201785697655
(6) bayes.BayesNet '-D -Q bayes.net.search.local.K2 -- -P 1 -S BAYES -E bayes.net.estimate.SimpleEstimator -- -A
(7) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVector.Puk -O 1.0 -S 1.0 -C 25
(8) functions.LibSVM '-S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model \"C:\\\\Program Fi
```

Figure 3.15 Classification process from the Weka Experimenter Interface

3.5.3. Combining multiple classifiers

The use of multiple classifiers for data mining is becoming a rapidly growing practice and it is gaining more attention in the machine learning community (Kuncheva, 2004). Combining multiple classifiers is a method used to achieve a more accurate classification decision by building a more reliable and sophisticated model. Due to the level of complexity that is involved, it is often proposed that the best classification tool at the disposal of the researcher should be first used before attempting to explore classifier combinations. Though combining classifiers can produce a better model, it is argued that it suffers all the draw backs of the various classifiers at the same time which is not desirable or cause over fitting problems shams (Rushdi, 2012). This can be attributed to the fact that a typical dataset often contain different kinds of variables such continuous, discrete, nominal and ordinal and it would be difficult to lump them together into a single classifier to optimize the performance of the classifier (Xu et al., 1992). In a published IEEE journal, (Xu et al., 1992) suggests four possible ways of combining classifiers to deal with these problems. He proposed a first approach which would deal with combining distance classifiers together, while the other three approaches could be used for combining any kind of classifier. It was concluded by (Dietterich, 2000) that multiple classifiers can be more advantageous as taking averages of various classifiers outputs can eliminate the risks of using an inadequate single classifier, generate better approximation of values, and increase accuracy. Among the various methods, this research considered standalone multiple commonly known as ensemble learners and also attempted to combine classifiers using Weka nested dichotomies.

3.5.3.1 Ensemble learning

Ensemble learning methods also called model combiners takes advantage of the influence of multiple models to give better prediction accuracy by improving the performance of learning algorithms. They are one of the main current focuses in machine learning and have been applied to many real life problems. Though, an integrated theory on ensemble methods is not available, there are theoretical justifications and empirical evidence of its efficiency (Valentini and Masulli, 2002). Weka offers various standalone ensemble learning methods that can effectively deal with binary class problems. These method ensures diversity by presenting each algorithms with different subset of training sets (Oza, 2008). They are explained below:

(a) Bagging: Bagging is an acronym for the classification procedure known as "**B**ootstrap **AGG**regat**ING"** and it works by combining a popularity vote based on the output of each copy of learning algorithms that are built on a bootstrap with samples of training set (Kuncheva, 2004). Independent classifiers generate a variety of training set by sampling with replacement such that small change would lead to large changes in the classifier output.

(b) AdaBoost: AdaBoost is an acronym for the classification procedure known as "**ADA**ptive **BOOST**ing", aboosting implementation in Weka. It is the most commonly used boosting algorithm which has proven successful because of its ability to quickly drive ensemble training error to zero with the initial iterations. It works by creating a sequence of base models and allocating weights to each set which would be used to predict the outcome of a certain event (Oza, 2008). AdaBoost was originally designed for binary classifications on nominal attributes but later extended to work on multiple classes. It applies the techniques of reweighting and resampling of classifiers by developing a group of classifiers and incrementally adding them one after the other (Kuncheva, 2004). The performance of the boosting and bagging algorithm was studied differently by (Freund and Schapire, 1997) and it was proved that the boosting performs significantly better that bagging especially when applied on a good performing single classifier.

(c) Random Forest: This is a classifier that comprises a group of tree – structured classifiers with each tree depending in the result of an independently sampled random vector, having the equal distribution across the trees in the forest (Breiman, 2001). When compared to bagging there is a minimal increase in diversity and reduction in classification errors. Research has proven random forest as an effective tool for delivering high classification performance among the various tools available today (Svetnik et al., 2003) as it creates diverse classifiers that are accurate.

3.5.3.2 Nested dichotomies

The Weka nested dichotomies allow handling of multiclass problems having an order of two-class classifiers. They are meta algorithms that are wrapped around base classifiers in order to increase their usability and performance. They form a binary tree that offers the option of inducing a single model from more than one classification algorithm by recursively breaking down a set of classification problems into smaller subsets (Dong et al., 2005). Amongst the options available in Weka, the stacking and voting which has been found to be effective in classification problem was applied in this research.

(a) Voting: Voting works by combining different machine learning algorithms producing classifiers for the same problem and letting them vote for test instances. It produces outputs that may be hard to analyse but gives good performance. Voting offers some combination rules that forms the criteria for combining the classifier output which are majority voting rule, median, maximum, minimum, average, and product of probabilities. For the voting approach, voting is done according to the weight of the classifier accuracy while for the probabilities, a probabilistic distribution vector for all the relevant classes is produced by each classifier based on the specified probability (Sigletos et al., 2005). One drawback that voting has is that it does not perform well when there are poorly performing classifiers among the base classifiers.

(b) Stacking: Stacked generalization otherwise known as stacking is used to combine predictions of multiple base learners using a meta classifier. Stacking lacks general acceptance because it has no defined method and it is difficult to analyse it (Hall et al., 2009). For example, stacking has no defined procedure to arrange or stack the selected base classifiers yet, this affects the classification accuracy. Stacking overcomes the limitation of handling poorly performing classifiers by searching for the reliable classifiers and using another learning algorithm known as the base classifiers to discover the best way of combining output of the base learners (Hall et al., 2009). It is also mentioned that stacking with meta decision trees gives better performance than bagging and boosting with decision trees (Džeroski and Ženko, 2004; Sigletos et al., 2005).

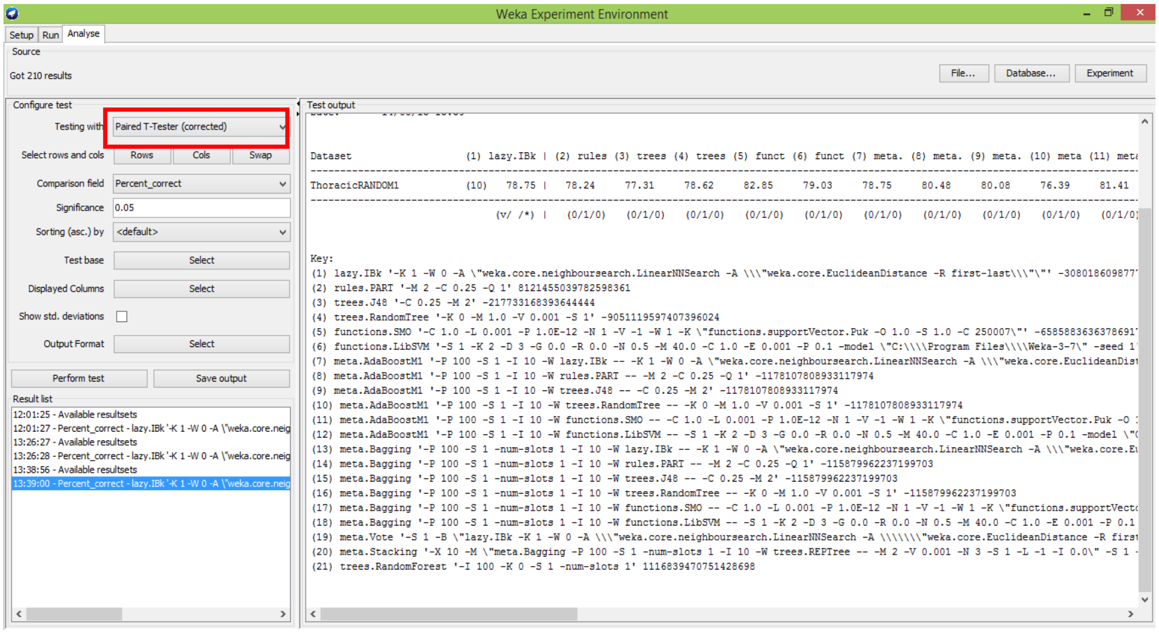
3.5.4. Evaluating the classifiers

Assessing the performance of various classifiers algorithms is a major step in both empirical and applied machine learning. It involves various methods for comparing and selecting a suitable model with best classification performance. As a traditional approach, the accuracy of the classifiers was checked. To guide against errors, the classifiers was evaluated by plotting the receiver operating characteristics (ROC) curve and the precision recall curve (PRC).

3.5.4.1 Percentage of correctly classified instances.

The result of the correctly classified instances is a measure of the instances that have been accurately classified as either positive of negative against the total number of instances in a data set. It is usually derived from the confusion matrix; however, it may be highly misleading for evaluating the performance of an algorithm based on this output alone especially when the number of positive cases in far lesser that the number of negative cases (Kubat et al., 1998). This difference in class level is very typical in a health data set. For example, in a case of 1000 patients with 995 testing negative to HIV and 5 testing positive, if the system classifies all the patients as negative, it would give an accuracy of 95.5 percent even though this is misleading and very costly as it has misclassified all the HIV positive patients.

For this test, the default percentage split on the randomized data, a 10 fold cross validation and a repeated 10 fold cross validation is applied to the data set. The result is then compared using the single classifier algorithms and the ensemble algorithms on the Weka experimental Interface. The top six performing classifiers from the initial selection criteria was selected for the single algorithms and the Adaboost and bagging Meta classifier was applied on each to enhance their performance. In addition, the bagging meta classifier was used to stack the Ibk, randomforest, PART and J48 while voting was applied on SMO and the same algorithms used for stacking. Figure 3.16 gives a pictorial representation of the process.



Source

Got 210 results

Configure test

Testing with: Paired T-Tester (corrected)

Select rows and cols

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Test base

Displayed Columns

Show std. deviations ☐

Output Format

Result list

```
12:01:25 - Available resultsets
12:01:27 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
13:26:27 - Available resultsets
13:26:28 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
13:38:56 - Available resultsets
13:39:00 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
```

Test output

```
Dataset                   (1) lazy.IBk | (2) rules (3) trees (4) trees (5) funct (6) funct (7) meta. (8) meta. (9) meta. (10) meta (11) meta
------------------------------------------------------------------------------------------------------------------------------------------
ThoracicRANDOM1           (10)   78.75 |   78.24     77.31     78.62     82.85     79.03     78.75     80.48     80.08     76.39     81.41
------------------------------------------------------------------------------------------------------------------------------------------
                               (v/ /*) |   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)   (0/1/0)


Key:
(1) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -308018609877
(2) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
(3) trees.J48 '-C 0.25 -M 2' -217733168393644444
(4) trees.RandomTree '-K 0 -M 1.0 -V 0.001 -S 1' -9051119597407396024
(5) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVector.Puk -O 1.0 -S 1.0 -C 250007\"' -6585883636378691
(6) functions.LibSVM '-S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model \"C:\\\\Program Files\\\\Weka-3-7\" -seed 1
(7) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W lazy.IBk -- -K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDist
(8) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W rules.PART -- -M 2 -C 0.25 -Q 1' -1178107808933117974
(9) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974
(10) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.RandomTree -- -K 0 -M 1.0 -V 0.001 -S 1' -1178107808933117974
(11) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVector.Puk -O 1
(12) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W functions.LibSVM -- -S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model \"C
(13) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W lazy.IBk -- -K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.Eu
(14) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W rules.PART -- -M 2 -C 0.25 -Q 1' -115879962237199703
(15) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703
(16) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.RandomTree -- -K 0 -M 1.0 -V 0.001 -S 1' -115879962237199703
(17) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVecto
(18) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W functions.LibSVM -- -S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1
(19) meta.Vote '-S 1 -B \"lazy.IBk -K 1 -W 0 -A \\\"weka.core.neighboursearch.LinearNNSearch -A \\\\\\\"weka.core.EuclideanDistance -R first
(20) meta.Stacking '-X 10 -M \"meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0\" -S 1 -
(21) trees.RandomForest '-I 100 -K 0 -S 1 -num-slots 1' 1116839470751428698
```

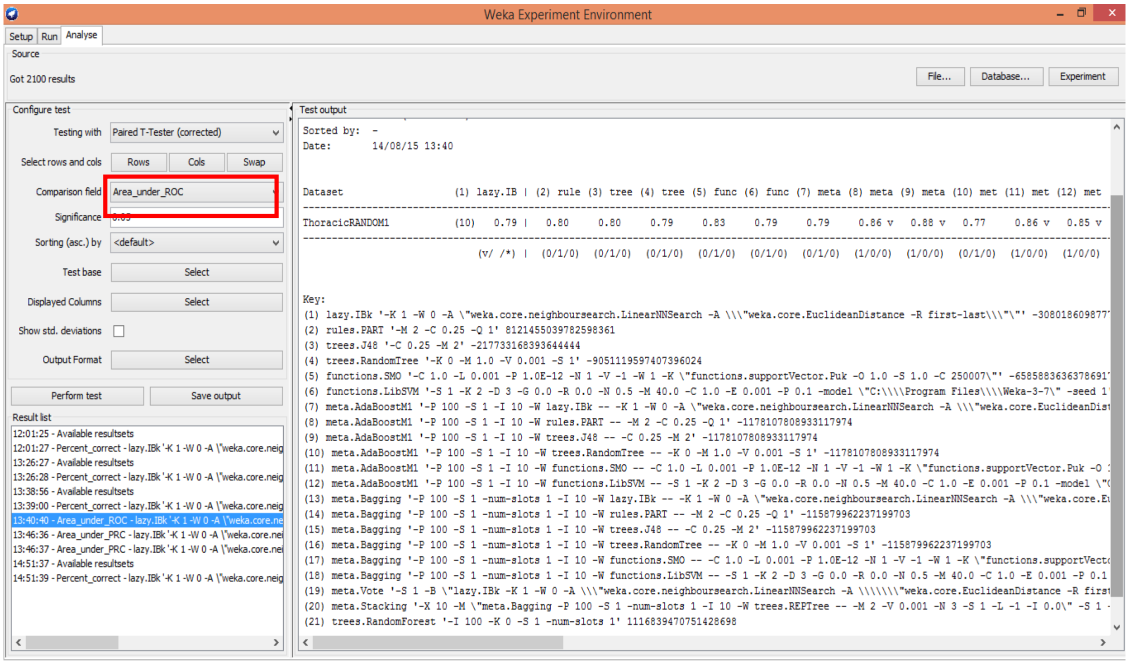
Figure 3.16 The procedure for generating the percentage accuracy of algorithms.

Weka provides an option for ranking the algorithms according to the comparison filed criteria that have been specified but this can only be done for single test criteria. For example, ranking according to the percent correct gives an output with the highest ranked algorithm at the top and the lowest ranked at the bottom which is achieved by selecting ranking as the test base. Since three test criteria are considered, the ranking was done based on the average classifier performance.

3.5.4.2 The receiver operating characteristics (ROC) area

The receiver operating characteristics (ROC) plot is one of the best-known graphical tools that enable testing of the performance of models applied to a data set for classification. It is used to visualize the performance of a binary classifiers i.e. a classifier with two possible output classes. For this data set, the possible classes are either 'DIED' or 'SURVIVED'. The medical professionals have developed an extensive literature that proves the use of ROC curves as one of the primary methods for diagnostic testing (Olson and Delen, 2008). The ROC curve gives a plot of the true positive rate (TPR) against the false positive rate (FPR) as the definition of the positive test is varied. The sensitivity of the test refers to the TPR while the specificity points to one minus the FPR.

The area under the ROC curve is the best composite measurement in ROC analysis (Centor and Keightley, 1989). The closer it gets to the value of one, the better the classifier algorithm which means that a perfect curve would give an area of one. This also implies that the classifier can completely separate the patients that survived from those that died. Figure 3.17 illustrates the process of generating the area under the ROC curve using the same testing algorithms for correctly classified instances from the Weka experimenter interface.



Got 2100 results

Testing with: Paired T-Tester (corrected)

Comparison field: Area_under_ROC

Significance: 0.05

Sorting (asc.) by: <default>

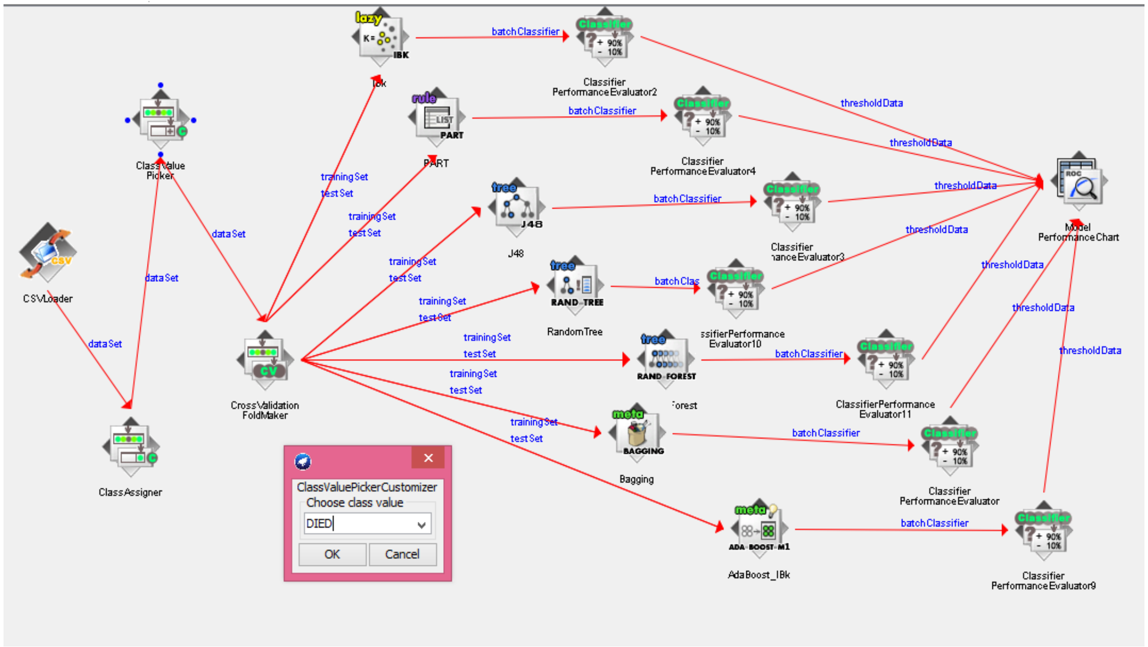
Show std. deviations: □

```
12:01:25 - Available resultsets
12:01:27 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
13:26:27 - Available resultsets
13:26:28 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
13:38:56 - Available resultsets
13:39:00 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
13:40:40 - Area_under_ROC - lazy.IBk '-K 1 -W 0 -A \"weka.core.ne
13:46:36 - Area_under_PRC - lazy.IBk '-K 1 -W 0 -A \"weka.core.nei
13:46:37 - Area_under_PRC - lazy.IBk '-K 1 -W 0 -A \"weka.core.nei
14:51:37 - Available resultsets
14:51:39 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
```

```
Sorted by:  -
Date:       14/08/15 13:40


Dataset                 (1) lazy.IB | (2) rule (3) tree (4) tree (5) func (6) func (7) meta (8) meta (9) meta (10) met (11) met (12) met
----------------------------------------------------------------------------------------------------------------------------------------
ThoracicRANDOM1         (10)   0.79 |   0.80     0.80     0.79     0.83     0.79     0.79     0.86 v   0.88 v   0.77      0.86 v   0.85 v
----------------------------------------------------------------------------------------------------------------------------------------
                            (v/ /*) |  (0/1/0)  (0/1/0)  (0/1/0)  (0/1/0)  (0/1/0)  (0/1/0)  (1/0/0)  (1/0/0)  (0/1/0)   (1/0/0)  (1/0/0)


Key:
(1) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -308018609877
(2) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
(3) trees.J48 '-C 0.25 -M 2' -217733168393644444
(4) trees.RandomTree '-K 0 -M 1.0 -V 0.001 -S 1' -9051119597407396024
(5) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVector.Puk -O 1.0 -S 1.0 -C 250007\"' -6585883636378691
(6) functions.LibSVM '-S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model \"C:\\\\Program Files\\\\Weka-3-7\" -seed 1
(7) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W lazy.IBk -- -K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDist
(8) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W rules.PART -- -M 2 -C 0.25 -Q 1' -1178107808933117974
(9) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974
(10) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.RandomTree -- -K 0 -M 1.0 -V 0.001 -S 1' -1178107808933117974
(11) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVector.Puk -O 
(12) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W functions.LibSVM -- -S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model \"
(13) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W lazy.IBk -- -K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.E
(14) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W rules.PART -- -M 2 -C 0.25 -Q 1' -115879962237199703
(15) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703
(16) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.RandomTree -- -K 0 -M 1.0 -V 0.001 -S 1' -115879962237199703
(17) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVect
(18) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W functions.LibSVM -- -S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1
(19) meta.Vote '-S 1 -B \"lazy.IBk -K 1 -W 0 -A \\\"weka.core.neighboursearch.LinearNNSearch -A \\\\\\\"weka.core.EuclideanDistance -R first
(20) meta.Stacking '-X 10 -M \"meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0\" -S 1 -
(21) trees.RandomForest '-I 100 -K 0 -S 1 -num-slots 1' 1116839470751428698
```

Figure 3.17 The procedure for generating the AUC curve.

The process of plotting the ROC curve involves a graphical representation of the relatonship between the TPR and the FPR. In this research,the ROC curve was generated for the best performing ensemble algorithm which is the random forest and six other classifiers based on the binary classes ‘DIED’ and ‘SURVIVED’. The process is depicted in Figure 3.18.



ClassValuePickerCustomizer

Choose class value

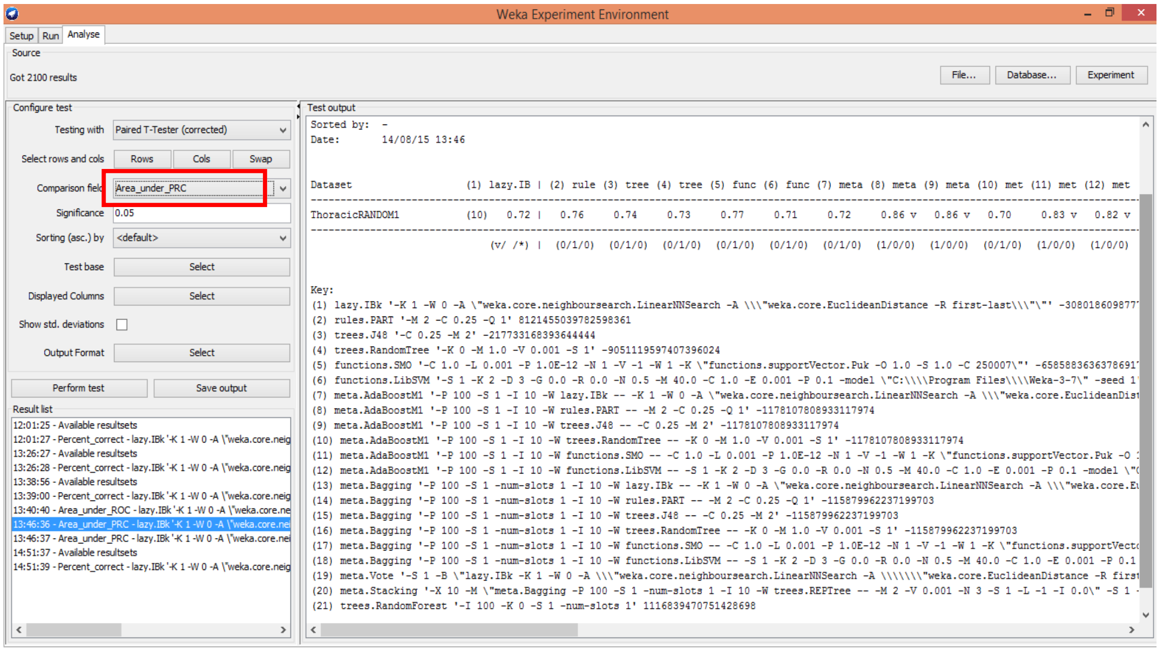
DIED

OK Cancel

Figure 3.18 Plotting multiple ROC curves of all the algorithms on the knowledge flow interface.

3.5.4.3 Precision recall curve (PRC).

Over the last few years, the use of the precision recall curve has gained more attention as a tool for evaluating the performance of classifiers in such fields as information retrieval, medicine, computational biology and processing of natural language (Keilwagen et al., 2014). It works well with binary classifiers and gives a plot of the precision (positive predictive value) against the recall (true positive rate). One major difference between the ROC curve and the PRC curve is that while the optimum performance of the classifier is depicted on the ROC curve when the curve is closer to the upper- left-hand-corner of the graph, the PRC has its optimum performance depicted when the curve is closer to the upper-right-hand-corner. This gives two kinds of visual representation for both curves. Figure 3.19 illustrates the process of generating the PRC curve.



Got 2100 results

Testing with: Paired T-Tester (corrected)

Comparison field: Area_under_PRC

Significance: 0.05

Sorting (asc.) by: <default>

Show std. deviations: □

```
12:01:25 - Available resultsets
12:01:27 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
13:26:27 - Available resultsets
13:26:28 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
13:38:56 - Available resultsets
13:39:00 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
13:40:40 - Area_under_ROC - lazy.IBk '-K 1 -W 0 -A \"weka.core.ne
13:46:36 - Area_under_PRC - lazy.IBk '-K 1 -W 0 -A \"weka.core.nei
13:46:37 - Area_under_PRC - lazy.IBk '-K 1 -W 0 -A \"weka.core.nei
14:51:37 - Available resultsets
14:51:39 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.neig
```

```
Sorted by:  -
Date:       14/08/15 13:46


Dataset                   (1) lazy.IB | (2) rule (3) tree (4) tree (5) func (6) func (7) meta (8) meta (9) meta (10) met (11) met (12) met
----------------------------------------------------------------------------------------------------------------------------------------------
ThoracicRANDOM1           (10)   0.72 |   0.76     0.74     0.73     0.77     0.71     0.72     0.86 v   0.86 v   0.70      0.83 v   0.82 v
----------------------------------------------------------------------------------------------------------------------------------------------
                                (v/ /*) |  (0/1/0)  (0/1/0)  (0/1/0)  (0/1/0)  (0/1/0)  (0/1/0)  (1/0/0)  (1/0/0)  (0/1/0)  (1/0/0)  (1/0/0)


Key:
(1) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -308018609877
(2) rules.PART '-M 2 -C 0.25 -Q 1' 8121455039782598361
(3) trees.J48 '-C 0.25 -M 2' -217733168393644444
(4) trees.RandomTree '-K 0 -M 1.0 -V 0.001 -S 1' -9051119597407396024
(5) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVector.Puk -O 1.0 -S 1.0 -C 250007\"' -6585883636378691
(6) functions.LibSVM '-S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model \"C:\\\\Program Files\\\\Weka-3-7\" -seed 1
(7) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W lazy.IBk -- -K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDist
(8) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W rules.PART -- -M 2 -C 0.25 -Q 1' -1178107808933117974
(9) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974
(10) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.RandomTree -- -K 0 -M 1.0 -V 0.001 -S 1' -1178107808933117974
(11) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVector.Puk -O
(12) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W functions.LibSVM -- -S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model \"
(13) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W lazy.IBk -- -K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.E
(14) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W rules.PART -- -M 2 -C 0.25 -Q 1' -115879962237199703
(15) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703
(16) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.RandomTree -- -K 0 -M 1.0 -V 0.001 -S 1' -115879962237199703
(17) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K \"functions.supportVect
(18) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W functions.LibSVM -- -S 1 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1
(19) meta.Vote '-S 1 -B \"lazy.IBk -K 1 -W 0 -A \\\"weka.core.neighboursearch.LinearNNSearch -A \\\\\\\"weka.core.EuclideanDistance -R first
(20) meta.Stacking '-X 10 -M \"meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0\" -S 1 -
(21) trees.RandomForest '-I 100 -K 0 -S 1 -num-slots 1' 1116839470751428698
```

Figure 3.19 The procedure for generating the PRC Curve.

The PRC curve is a plot of the precision versus the recall. During the process of plotting the PRC curve, the same classifers used for the ROC curve was applied. Again, the more the curve points to the upper right corner, the better the curve.

3.5.5. Choosing the classifier model

There is no perfect tool that can be used to measure the performance of classifiers because this is largely dependent several factors like the kind of data set being analysed, the purpose of the classification, etc. Though the area under the ROC has a shortfall with respect to factoring the error cost or class distribution it has, it remained the best criteria used for measuring classifier performance for medical diagnosis since the 1970's (Keilwagen et al., 2014). The PRC also suffers the shortcoming in the area of interpolating the points of the variables however, it is very effective on data sets with skewed class distribution (Davis and Goadrich, 2006). In addition, research has proven that evaluating a classifier by simply checking their classification accuracy can be misleading and using the area under the ROC gives a better results (Huang and Ling, 2005).

3.6. Data post-processing

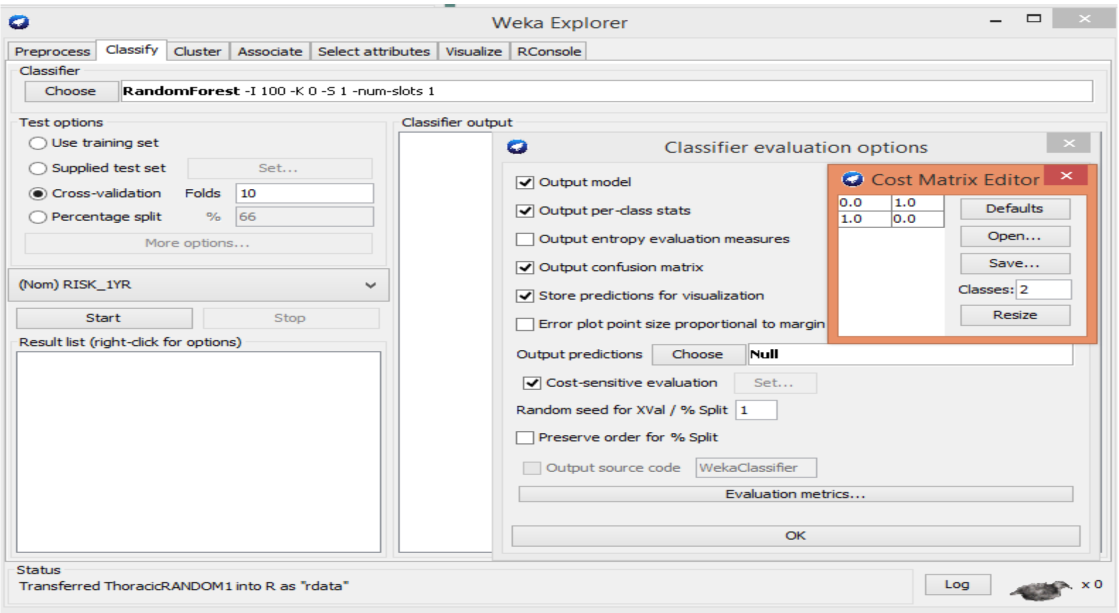
This is a knowledge discovery process that involves getting the true value of the result of the machine learning algorithm. The percentage accuracy, the area under the receiver operating characteristic (ROC) curves and the precision recall curve (PRC) was used as a graphical tool to evaluate the performance of classifiers. This was done in order to select the best performing classifier which was eventually used to build the model for decision making. Data post-processing generally involves the following (Bruha and Famili, 2000);

1. The filtering of knowledge by post-pruning the nodes and branches of decision tree.
2. Presentation of knowledge gained in a visualizable format.
3. Specifying criteria to evaluate and test the model
4. Integration of knowledge gained with decision support systems.

3.6.1. The loss matrix

Studies show that most of the algorithms used today assume equal cost for all errors which is rarely true in the real world of knowledge discovery. For example, when a financial institution wants to offer a credit facility, it may be considered a higher risk when a customer with a bad credit record is misclassified as good than when a good profiled customer is misclassified as bad because the former case has the potential of ruining the business. Similarly, in a medical diagnosis, it would be more costly to inaccurately misclassify a sick customer as healthy than to misclassify a customer that is healthy as being sick. It has been mentioned that the inclusion of costs into machine learning and classification would be one of the most important topics of future research (Brefeld et al., 2003). This has unavoidably led to an increased interest ways to enhance the cost sensitivity of classification algorithms (Domingos, 1999) but only a few studies reveal how it works with essemble classifiers.

For this research, a higher cost for wrongly predicting patients that die within one year to have survived (FNR) than for wrongly predicting the patients that survived to have died (FPR) is recommended. This would mean specifying a threshold that increases the TPR against the TNR. In other words, it is better to wrongly predict a patient to die within a year and the patient goes on to live some more years that to wrongly give a patient a high hope of surviving a thoracic surgery operation and the patient dies within a short period. Specifying a loss matrix can be set from the cost-sensitive evalutaion options on the weka classify panel or by using a cost sensitve classifier but more information would be needed to assign the correect class weights. This is illustrated in Figure 3.20;



Classifier
RandomForest -I 100 -K 0 -S 1 -num-slots 1

Test options
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) RISK_1YR

Classifier evaluation options
☑ Output model
☑ Output per-class stats
□ Output entropy evaluation measures
☑ Output confusion matrix
☑ Store predictions for visualization
□ Error plot point size proportional to margin
Output predictions Null
☑ Cost-sensitive evaluation
Random seed for XVal / % Split 1
□ Preserve order for % Split
□ Output source code WekaClassifier

Cost Matrix Editor

0.0	1.0
1.0	0.0

Classes: 2

Status
Transferred ThoracicRANDOM1 into R as "rdata"

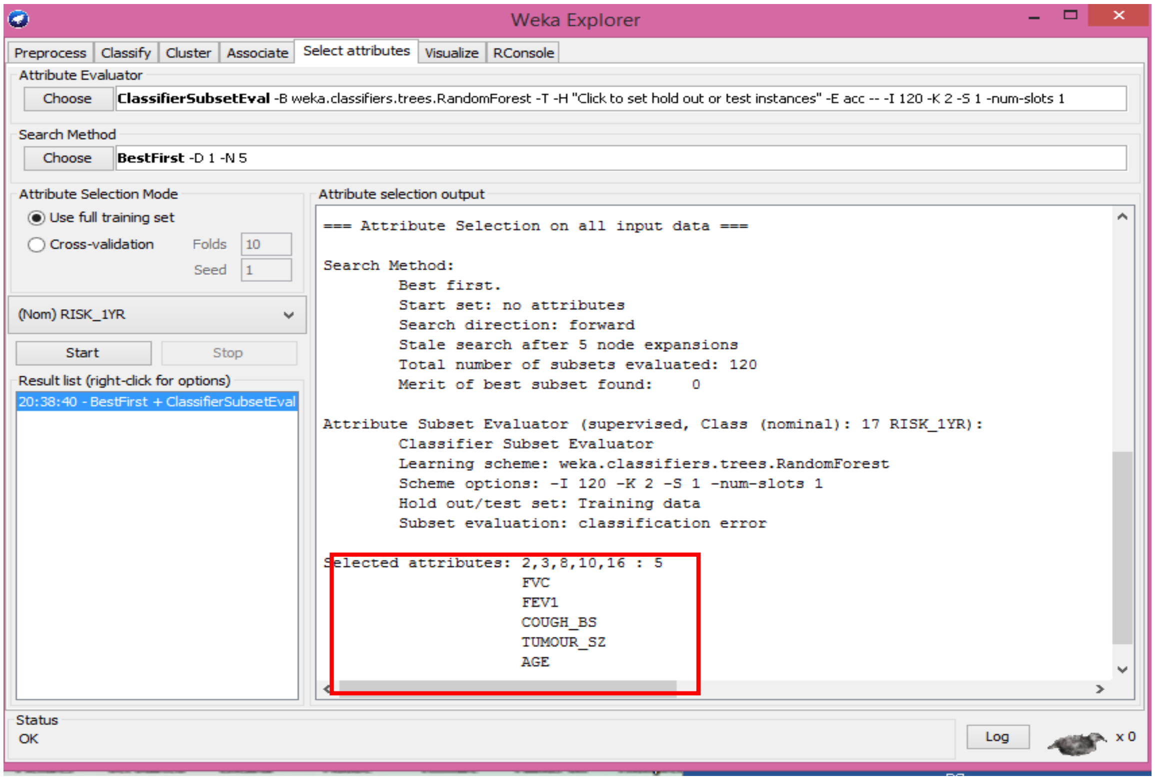
Figure 3.20 Specifying a Loss Matrix.

3.6.2. Knowledge filtering

This involved the methods applied on the model to improve the performance of the model in terms of ROC, PRC and percentage classification accuracy of the random forest model. (Breiman, 2001) did an extensive work on random forest and recommend some ways of improving the performance of the classifier and at the same time avoid overfitting. He suggested using a random selection of features to split each node noting that reducing the number of features selected can give a near optimum result. He also mentioned that increasing the number of trees do not cause overfitting and combining boosting on random features with a small data set would not yield any significant change. An attempt was made to improve the random forest output by trying several options for adjusting the number of attributes to be used in random selection and the generated number of trees At the end, setting the number of features to 2 and number of tree to 120 seemed to produce a better accuracy but the default setting was used because it was more effective in correctly classifying the true positive rate.

3.6.3. Feature selection by wrapper method with RandomForest.

(Fayyad et al., 1996) emphasised that data mining is an iterative process and researchers often see the need to visit some steps performed during the earlier stages of the data mining process to get more interpretation and for visualization the data set. The wrapper method uses a subset evaluator to create subsets from the feature vector and then applies a classifiers such as nearest neighbour classifiers, decision trees to select features based on their classification performance (Rangarajan, 2010). The evaluator uses a search technique such as random search, hybrid search, depth first search and breath first search to find the subsets. Since the random forest ensemble algorithm outperformed other classifiers, it was applied to gain more understanding of the most relevant attributes that influenced the classification results and for visualization. The process and output is indicated in Figure 3.21 below from the Weka explorer Interface:



Attribute Evaluator

ClassifierSubsetEval -B weka.classifiers.trees.RandomForest -T -H "Click to set hold out or test instances" -E acc -- -I 120 -K 2 -S 1 -num-slots 1

Search Method

BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Nom) RISK_1YR

Result list (right-click for options)

20:38:40 - BestFirst + ClassifierSubsetEval

Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 120
        Merit of best subset found:    0

Attribute Subset Evaluator (supervised, Class (nominal): 17 RISK_1YR):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.trees.RandomForest
        Scheme options: -I 120 -K 2 -S 1 -num-slots 1
        Hold out/test set: Training data
        Subset evaluation: classification error

Selected attributes: 2,3,8,10,16 : 5
                     FVC
                     FEV1
                     COUGH_BS
                     TUMOUR_SZ
                     AGE
```

Status

OK

Figure 3.21 Attribute selection procedure.

The random forest classifier has selected five out of the 17 attributes in the data set as the most relevant features for building the model. These are the numeric attributes FVC, FEV1 and AGE with nominal attributes COUGH_BS and TUMOUR_SZ.

3.6.4. Classifying with Weka packages on Java

Like the Rapid Miner, Hadoop and Mallet, Weka is written in Java, an object oriented programming language and it works by calling the Java Virtual Machine to execute the various algorithms embedded as packages. (Abeel et al., 2009) highlighted that implementing new algorithms and comparing classifiers and clustering for Java machine learning (ML) can be relatively easy and straight forward . He also stated that Java ML allows a wide-range of instance based techniques well suited to manage high- dimensional domains often experienced in bioinformatics and biomedical applications (Abeel et al., 2009). This would exclude the conventional use of the graphical user interface that is offered by the the data analysis applications. Classification of the data set using the random forest classifer on the eclipse IDE is demonstrated below;

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.Random;

import weka.classifiers.evaluation.*;
import weka.classifiers.trees.J48;
import weka.classifiers.trees.RandomForest;
import weka.classifiers.trees.RandomTree;
import weka.core.Instances;

public class StartWeka {
        public static void main(String[]args)throws Exception{

        BufferedReader  br  =  new  BufferedReader(new  FileReader("C:/Users/Username/Desktop/
        ProjFile/Thoracic_surgery.arff"));

        Instances train = new Instances(br);
        train.setClassIndex(train.numAttributes()-1);//Returns the last attribute.
        br.close();

        RandomForest tree = new RandomForest();
        tree.buildClassifier(train);
        Evaluation eval = new Evaluation(train);
        eval.crossValidateModel(tree, train, 10, new Random(1));
        System.out.println(eval.toSummaryString("\nResults\n=====\n", true));
        System.out.println("Fmeasure: "+ eval.fMeasure(1)+ "\nPrecision: " + eval.precision(1)
        + "\nRecall: " +eval.recall(1) + eval.toMatrixString());
                }
}
```

CHAPTER 4

4. RESULTS AND ANALYSIS

In this chapter, the results from the data mining and post-processing stages of patient datasets in the previous chapter is stated and analysed and the performance of the model evaluated. Elements of the misclassification table known as the confusion matrix that is generated by the model would be discussed, showing how they help to answer the research questions. It was noted by (Fayyad et al., 1996) that the process of knowledge discovery is an iterative process which may require the researcher to revisit some steps that have previously been explored in the earlier stages of the data mining process. Graphical exploration is applied again using the features generated by the model to gain more understanding about the operation of the model.

4.1. The pre-processing and exploration

This section presents the output of the applied algorithms on Weka as illustrated in the previous chapter in a tabular form for ease of interpretation.

4.1.1. Evaluation of single classification

The result of the single classification as applied in section 3.5.2 yielded the results in the Table 4.1. It is observed that sequential minimal optimization (SMO) algorithm performed best with an accuracy of 82.85 percent followed by the library for support vector machine (libSVM) and the instance based learner (ibk). The parameters of the SMO was tweaked against its default settings to yield optimum results. The naïve bayes classifiers performed least with an output of 71.91 percent which is most likely due to the fact that it works by assuming feature independence (Rish, 2001), conversely, health data sets often exhibit a lot of dependencies among the variables.

Table 4.1 The single classifier output.

Algorithms	IBk	PART	J48	Random Tree	NaiveBayes	BayesNet	SMO	lIBsvm
Percent Correct	78.75	78.24	77.31	78.63	71.91	74.67	82.85	79.02

4.1.2. Classifier accuracy

The Table 4.2 shows the accuracy based on percentage split on the data, a 10 fold cross validation and a repeated 10 fold cross validation using the top six classifiers in Table 4.1 with some multiple classifiers. The result indicates a good performance for all the classifiers with the lowest accuracy value at 76.3 percent. The performance of majority of the single classifiers were boosted by the ensemble learners except for the random tree, the libSVM and Lazy Ibk where no significant change was recorded possibly due to overfitting problems. The best algorithm according to accuracy was achieved by stacking Ibk, randomforest, PART and J48 classifiers together.

Table 4.2 Results of the Percentage of Correctly classified Instances

Classifers (Percent Correct)	Percentage Split	10 Fold Cross Validation	10 Fold Cross Validation(10 repetition)	Average	Rank
Stacking	81.95	84.30	83.23	83.160	1
SMO	82.80	82.85	83.45	83.033	2
Bagging (SMO)	82.76	82.85	83.18	82.930	3
Random Forest	82.61	82.72	83.06	82.797	4
Voting	82.14	81.92	82.48	82.180	5
AdaBoost (SMO)	81.13	81.41	82.13	81.557	6
Bagging (LazyJ48)	79.93	82.19	80.96	81.027	7
Bagging (Random Tree)	80.09	81.53	81.17	80.930	8
Bagging (Lazy PART)	80.13	80.21	81.58	80.640	9
AdaBoost (Lazy PART)	79.62	80.48	79.79	79.963	10
AdaBoost (libSVM)	79.74	80.62	79.50	79.953	11
AdaBoost (LazyJ48)	79.47	80.08	80.12	79.890	12
LIBSVM	78.14	79.03	79.71	78.960	13
Bagging (Lazy Ibk)	78.14	78.89	78.87	78.633	14
Lazy Ibk	78.34	78.75	78.69	78.593	15
AdaBoost (Lazy Ibk)	78.34	78.75	78.69	78.593	16
PART	78.07	78.24	78.53	78.280	17
Bagging (libSVM)	76.78	78.24	78.43	77.817	18
J48	76.71	77.31	78.12	77.380	19
Random Tree	75.70	78.62	77.14	77.153	20
AdaBoost (Random Tree)	76.01	76.39	76.58	76.327	21

4.1.3. The area under the receiver operating characteristics (ROC) curve.

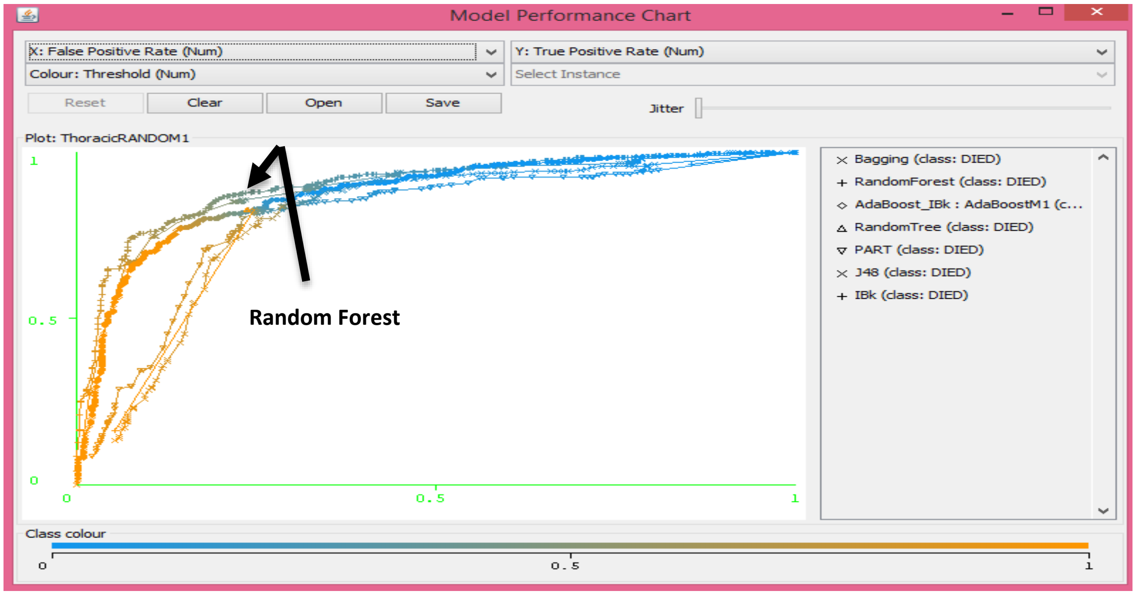
As mentioned in the previous chapter, an ROC area with a value of one indicates a perfect classification. Table 4.3 shows a list of the various classifiers evaluated on the dataset and their respective average ROC areas with the last column representing their ranks.

Table 4.3 Results of the area under the ROC curve.

Classifers (ROC Area)	Percentage Split	10 Fold Cross Validation	10 Fold Cross Validation(10 repetition)	Average	Rank
Random Forest	0.89	0.90	0.90	0.897	1
Voting	0.88	0.89	0.89	0.887	2
Stacking	0.87	0.89	0.89	0.883	3
Bagging (Random Tree)	0.87	0.89	0.88	0.880	4
Bagging (LazyJ48)	0.86	0.88	0.88	0.873	5
AdaBoost (LazyJ48)	0.86	0.88	0.87	0.870	6
Bagging (Lazy PART)	0.86	0.87	0.88	0.870	6
AdaBoost (Lazy PART)	0.86	0.86	0.87	0.863	8
AdaBoost (SMO)	0.85	0.86	0.86	0.857	9
Bagging (SMO)	0.85	0.85	0.85	0.850	10
AdaBoost (libSVM)	0.84	0.85	0.85	0.847	11
Bagging (libSVM)	0.84	0.84	0.85	0.843	12
SMO	0.83	0.83	0.83	0.830	13
Bagging (Lazy Ibk)	0.82	0.83	0.84	0.830	14
PART	0.79	0.80	0.80	0.797	15
J48	0.78	0.80	0.80	0.793	16
LIBSVM	0.78	0.79	0.80	0.790	17
Lazy Ibk	0.78	0.79	0.79	0.787	18
AdaBoost (Lazy Ibk)	0.78	0.79	0.79	0.787	19
Random Tree	0.76	0.79	0.78	0.777	20
AdaBoost (Random Tree)	0.76	0.77	0.77	0.767	21

From the results, it is observed that the random forest classifier performed best with an average ROC area of 0.897, but this time, the stacked algorithms seems to have been demoted to the third place. This further raises the concern of over-dependency on accuracy of classifiers as a tool for classifier evaluation as this may be deceptive (Provost et al., 1998). The random tree still maintained its position in the last two rows as the worst classifier. Finally, it is observed that the classifiers performed better when boosted with ensemble learners.

The ROC curve for patients who died and those who survived is illustrated in Figure 4.1A-B for the binary classes ‘DIED’ and ‘SURVIVED’ respectively for ease of comparison and visualization using the Weka knowledege flow Interface illustrated in Figure 3.18. The best classifier curve points towards upper-left-corner.



Model Performance Chart

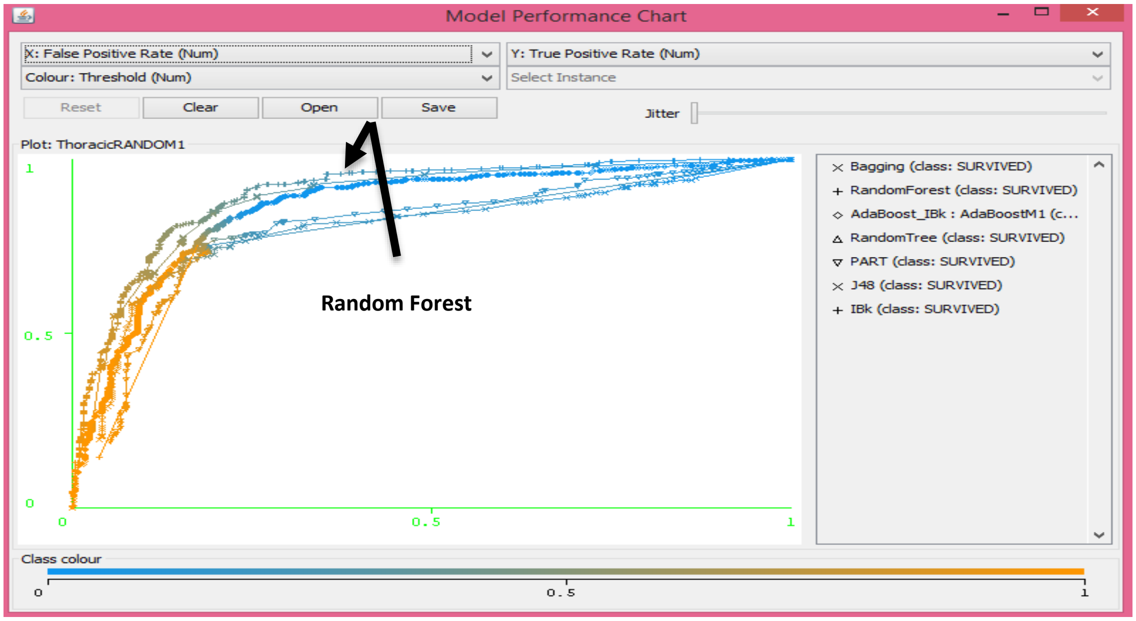
X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Plot: ThoracicRANDOM1

Figure 4.1A The model performance chart of the ROC curves for the class of patients that died.



Model Performance Chart

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Plot: ThoracicRANDOM1

Figure 4.1B The model performance chart of the ROC curves for the class of patients that survived.

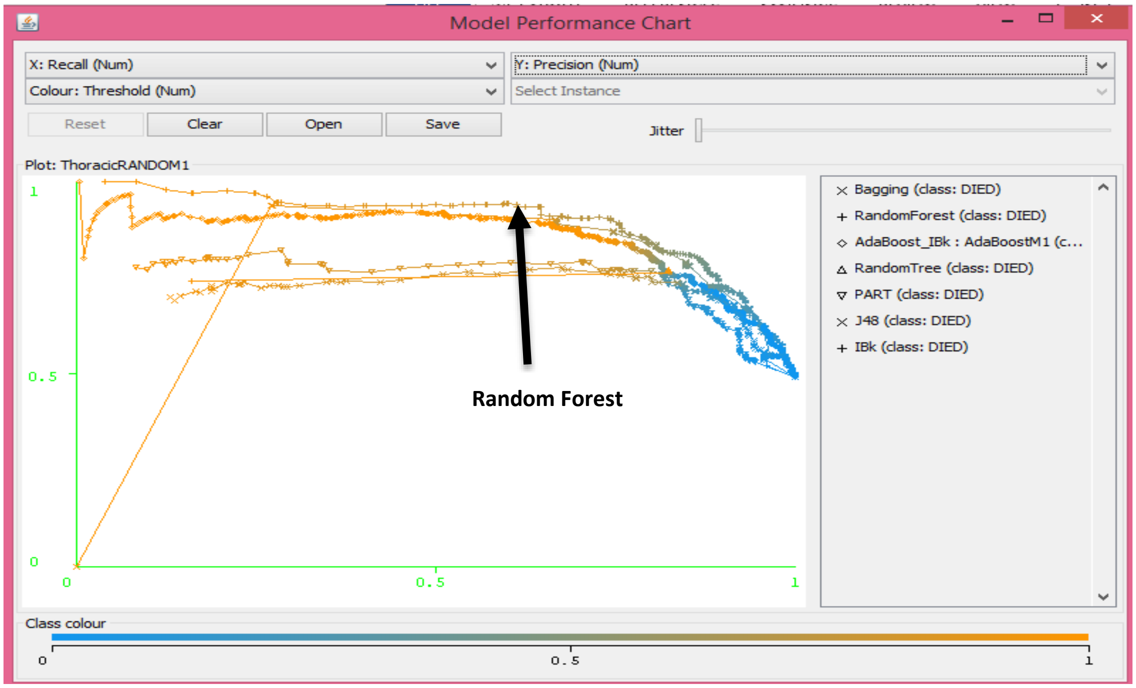
The classifiers that were plotted is indicated on the left panel of the graph in Figure 4.1A-B. It should be noted that Weka does not arrange the line graphs according to how it is listed on panel on the left hand side but the lines can be recognised by the signs or by clicking on any of the points across the line. The random forest has been labelled depicting the best ROC curve.

4.1.4. The area under the precision recall curve (PRC).

Table 4.4 indicates the performance results of the classifiers ranked according to the average area under the PRC. It is observed that the random forest classifier dropped to the second place while the voting method performed best with a PRC area of 0.900. Again, the classifiers performed better after boosting them with the J48 algorithm having the most improved performance. Plotting the PRC curve according to the same criteria used for the ROC curve in indicated in Figure 4.2A-B.

Table 4.4 Results of the Precision recall curve.

Classifers (PRC)	Percentage Split	10 Fold Cross Validation	10 Fold Cross Validation(10 repetition)	Average	Rank
Voting	0.89	0.91	0.90	0.900	1
Random Forest	0.88	0.90	0.90	0.893	2
Stacking	0.86	0.89	0.89	0.880	3
Bagging (LazyJ48)	0.85	0.88	0.88	0.870	4
Bagging (Lazy PART)	0.85	0.87	0.88	0.867	5
Bagging (Random Tree)	0.85	0.87	0.87	0.863	6
AdaBoost (LazyJ48)	0.85	0.86	0.86	0.857	7
AdaBoost (Lazy PART)	0.84	0.86	0.86	0.853	8
AdaBoost (SMO)	0.82	0.83	0.83	0.827	9
AdaBoost (libSVM)	0.81	0.82	0.83	0.820	10
Bagging (SMO)	0.81	0.80	0.81	0.807	11
Bagging (Lazy Ibk)	0.77	0.80	0.80	0.790	12
Bagging (libSVM)	0.78	0.79	0.80	0.790	13
SMO	0.78	0.77	0.78	0.777	14
PART	0.74	0.76	0.75	0.750	15
J48	0.72	0.74	0.74	0.733	16
Lazy Ibk	0.71	0.72	0.72	0.717	17
AdaBoost (Lazy Ibk)	0.71	0.72	0.72	0.717	18
LIBSVM	0.71	0.71	0.72	0.713	19
Random Tree	0.69	0.73	0.71	0.710	20
AdaBoost (Random Tree)	0.70	0.70	0.70	0.700	21

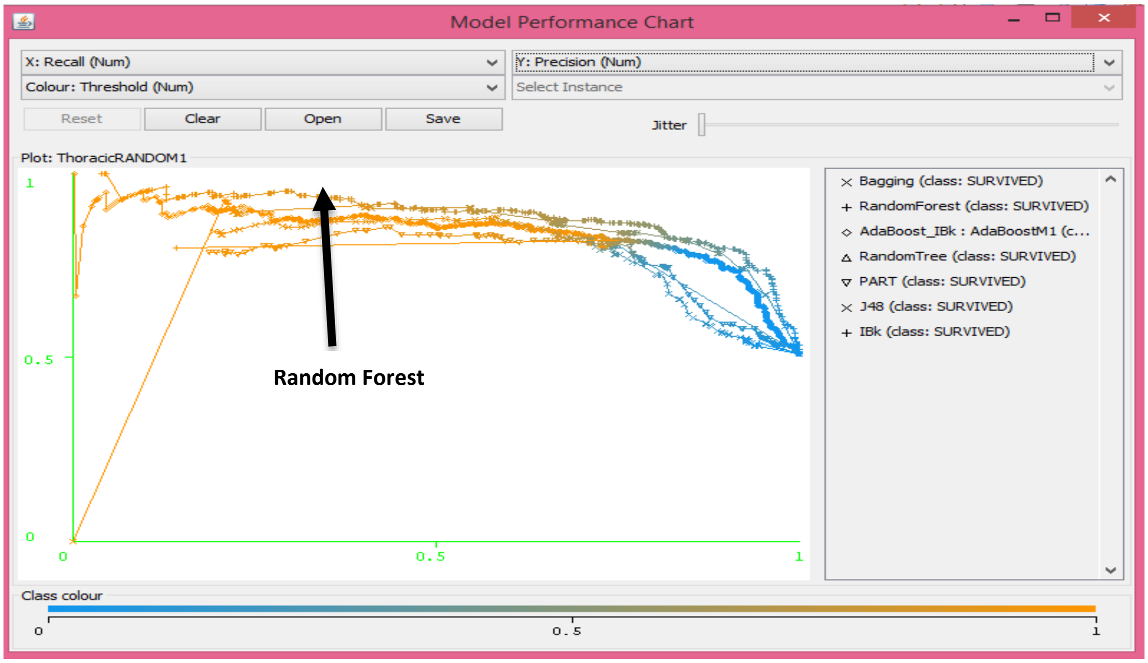

X: Recall (Num)

Y: Precision (Num)

Colour: Threshold (Num)

Plot: ThoracicRANDOM1

Figure 4.2A The model performance chart of the PRC curves for the class of patients that died.


X: Recall (Num)

Y: Precision (Num)

Colour: Threshold (Num)

Plot: ThoracicRANDOM1

Figure 4.2B: The model performance chart of the PRC curves for the class of patients that survived.

As seen in Tables 4.2A-B, it was observed that the algorithms behaved differently under the three test measures with Stacking, Random Forest and Voting ranking first for the accuracy, the area under the ROC and the area under the PRC respectively. The Random Forest is the preferred algorithm for the classification owing to the fact that it is more consistent among the top ranks and it has the highest area under the ROC curve.

4.2 Evaluating the Random Forest Performance.

So far, the performance of various classifiers were compared and the random forest was selected as the classifier for building the model. It is no surprise that the random forest classifier outperformed other classifiers as many research has proven the effectiveness of the random forest algorithm as a classifier in the medical domain. (Lavanya and Rani, 2011) mentioned that decision tree classifiers are broadly used for medical diagnosis such as ovarian cancer, breast tumour and heart sound diagnosis. Generally, a good classifier for medical diagnosis should be very effective in terms of performance, ability to deal with missing and noisy data , measure of transparency to the medical practitioner , ease of description and should have a reduced amount of test carried out on patients (Kononenko, 2001). In addition to these attributes, (Breiman, 2001) mentioned that the random forest works faster than bagging and boosting ; highlighting its simplicity and ease of parallelization. This are further discussed below;

4.2.1. Performance measure

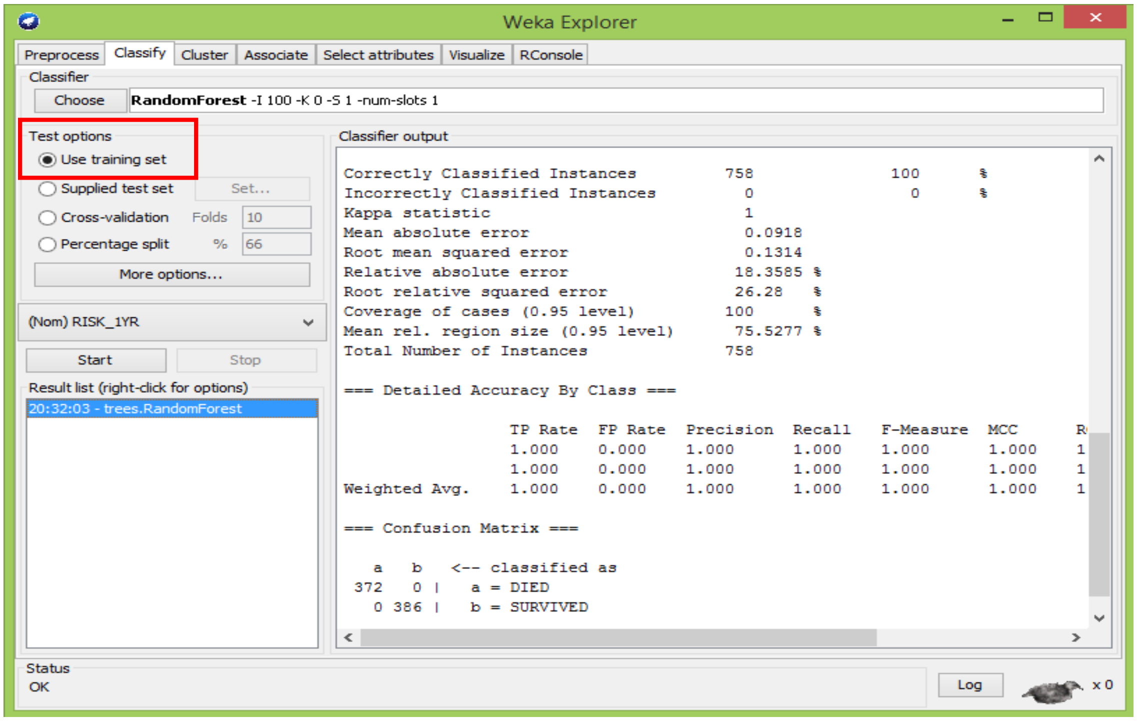
The random forest was the best performing classifier based on the average ranking criteria with an ROC area of 0.897, a PRC area of 0.893 and an accuracy of 82.797%. This indicates a significantly impressive performance when compared to the base line or minimum accuracy of 50.92% generated by ZeroR classifier. Some other classifiers also came very close such as the voting, stacking, bagging and the boosted J48 algorithm which are all ensemble learners; further emphasizing the efficacy of using multiple classifiers over single classifiers. The accuracy result for the support vector machines was very impressive but the ROC and the PRC areas were quite low in comparison with other algorithms. The PART rule based learner only performed well after it was boosted and the Bayesian classifiers were not employed at the final test stage due to the poor result generated at the initial test stage.

The better performance of the random forest may be attributed to the fact that while single decision trees are subject to high variance and bias owing to tuning issues, the random forest combines the benefit of two dominant machine learning techniques; bagging and feature selection to give the average output of individual classification trees (Pal, 2005). Also, while other classifiers uses methods for classification like the probability distributions from the Bayes network and naïve Bayes, nearest neighbours from Ibk and linear combination of attributes from neural networks like the SVM, tree classifiers works by memorizing a set of TRUE/FALSE decision rules to classify attributes. The decision rules tends to fit well with medical diagnosis features which typically answers a true or false question as to whether a patient has a disease or not. For example, the thoracic data set has 17 attributes and 11 of them are nominal binary attributes including the class attribute. This would make rule decisions to be less complex and easily interpretable.

4.2.2. Dealing with noise and overfitting

Overfitting is a subtle problem that plagues machine learning methods both at the theoretical and practical levels. A good algorithm should eliminate a low bias and high variance during classification. A major characteristic of the random forest is that it creates more complex decision boundaries from the multiple decision trees making it less susceptible to overfitting. This is achieved by adding a randomness criteria to the prediction probability returned by each tree which is then averaged. One way it does this is by a method known as bootstrap aggregation which uses random subsets to stabilize predictions and reduce overfitting (Witten and Frank, 2005). The randomness is also achieved by the combination of different tree predictors which uses a feature selection criteria to randomly select the features by splitting the tree nodes. The feature selection process helps the algorithm to reduce the noise that could be created from less relevant features.

An example of running the random forest on the training set alone is depicted in Figure 4.3. This gives an imperfect boundary which splits all the variables to their appropriate classes with a 100 percent accuracy. This means that each specified tree makes the exact same prediction, returning an average prediction probability of 100 percent but when a cross validation technique is employed, this average is reduced to 82.72 percent reflecting a good margin when compared to the overemphasised result which occurred due to overfitting. It is therefore misleading to concentrate classification efforts on training set alone.



Classifier: RandomForest -I 100 -K 0 -S 1 -num-slots 1

Test options
● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) RISK_1YR

Result list (right-click for options)
20:32:03 - trees.RandomForest

Classifier output

```
Correctly Classified Instances         758               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0918
Root mean squared error                  0.1314
Relative absolute error                 18.3585 %
Root relative squared error             26.28   %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      75.5277 %
Total Number of Instances              758

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      R
                 1.000    0.000    1.000      1.000    1.000      1.000    1
                 1.000    0.000    1.000      1.000    1.000      1.000    1
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1

=== Confusion Matrix ===

   a   b   <-- classified as
 372   0 |   a = DIED
   0 386 |   b = SURVIVED
```

Status
OK

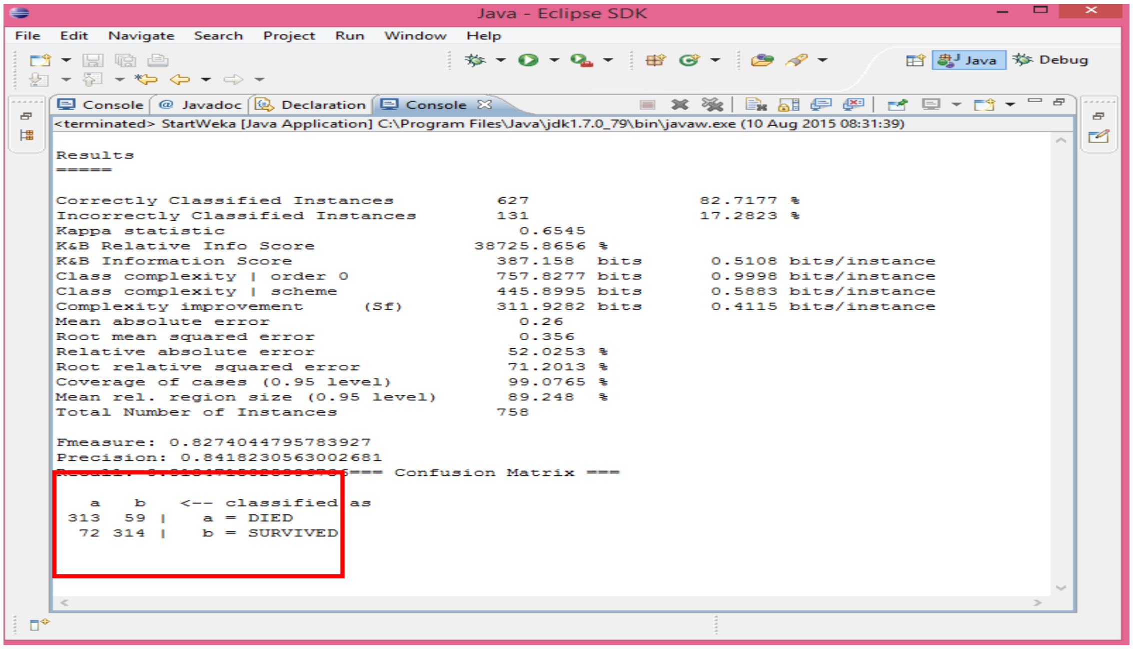
Figure 4.3 Output of classifying with the training set by random forest.

4.2.3. Measure of Transparency and ease of description

The measure transparency refers to the ability to easily analyse and understand the knowledge generated from the application of a particular learning algorithm (Kononenko, 2001). In other words, the physician should be able to interpret classification output for making an informed decision. One great advantage of the random forest is that it produces human readable rules for classification using nodes and leaves to position of each attribute thus, building the tree quickly and at the same time yielding better accuracy (Lavanya and Rani, 2011). On the other hand, other algorithms such as neural networks produces a good classification result but applies multiple weighting and voting technique that is hard to analyse theoretically, The nearest neighbour has a poor transparency of knowledge presentation.

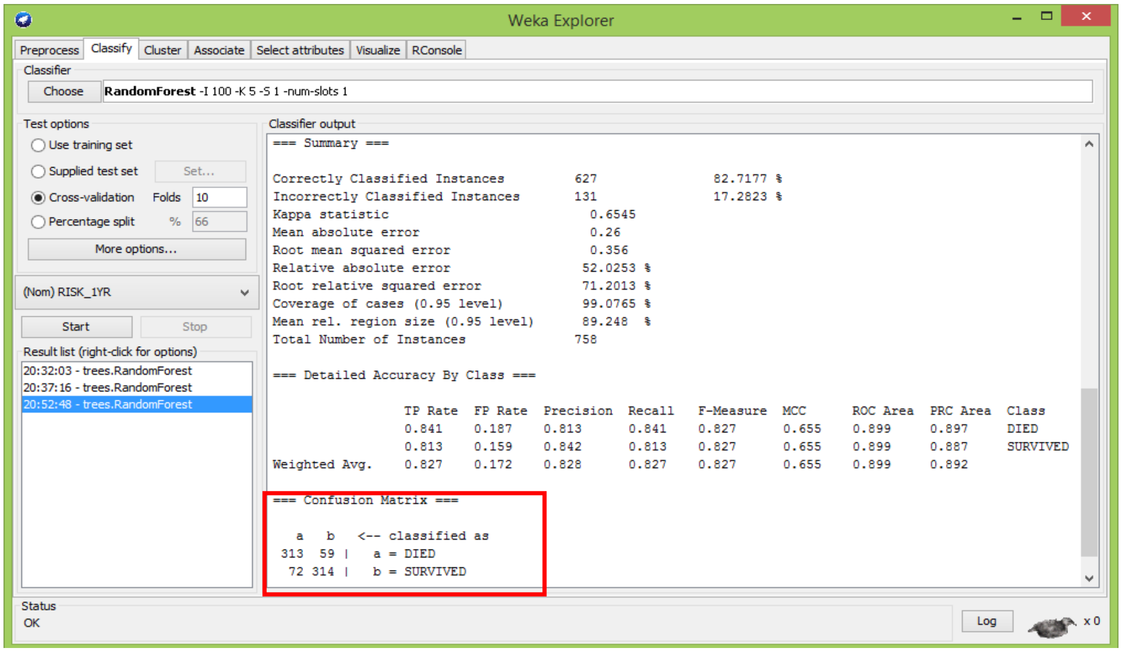
4.3. Classification result on Weka explorer interface.

Figure 4.4A and B indicates the output of the random forest classification from eclipse console as specified in section 3.6.4 and on Weka respectively.



```
<terminated> StartWeka [Java Application] C:\Program Files\Java\jdk1.7.0_79\bin\javaw.exe (10 Aug 2015 08:31:39)

Results
======

Correctly Classified Instances         627               82.7177 %
Incorrectly Classified Instances       131               17.2823 %
Kappa statistic                          0.6545
K&B Relative Info Score              38725.8656 %
K&B Information Score                  387.158  bits      0.5108 bits/instance
Class complexity | order 0             757.8277 bits      0.9998 bits/instance
Class complexity | scheme              445.8995 bits      0.5883 bits/instance
Complexity improvement     (Sf)        311.9282 bits      0.4115 bits/instance
Mean absolute error                      0.26
Root mean squared error                  0.356
Relative absolute error                 52.0253 %
Root relative squared error             71.2013 %
Coverage of cases (0.95 level)          99.0765 %
Mean rel. region size (0.95 level)      89.248  %
Total Number of Instances              758

Fmeasure: 0.8274044795783927
Precision: 0.8418230563002681
Recall: 0.8104715[illegible]6=== Confusion Matrix ===

   a   b   <-- classified as
 313  59 |   a = DIED
  72 314 |   b = SURVIVED
```

Figure 4.4A Classification result on Eclipse Console



RandomForest -I 100 -K 5 -S 1 -num-slots 1

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) RISK_1YR

20:32:03 - trees.RandomForest
20:37:16 - trees.RandomForest
20:52:48 - trees.RandomForest

```
=== Summary ===

Correctly Classified Instances         627               82.7177 %
Incorrectly Classified Instances       131               17.2823 %
Kappa statistic                          0.6545
Mean absolute error                      0.26
Root mean squared error                  0.356
Relative absolute error                 52.0253 %
Root relative squared error             71.2013 %
Coverage of cases (0.95 level)          99.0765 %
Mean rel. region size (0.95 level)      89.248  %
Total Number of Instances              758

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.841    0.187    0.813      0.841    0.827      0.655    0.899     0.897     DIED
                 0.813    0.159    0.842      0.813    0.827      0.655    0.899     0.887     SURVIVED
Weighted Avg.    0.827    0.172    0.828      0.827    0.827      0.655    0.899     0.892

=== Confusion Matrix ===

   a   b   <-- classified as
 313  59 |   a = DIED
  72 314 |   b = SURVIVED
```

Status
OK

Figure 4.4B The random forest classification output on Weka.

An important output known as the confusion matrix in Figure 4.4A-B from which the parameters for measuring the classifier performance is derived is further explored.

4.3.1. The confusion matrix

The confusion matrix otherwise known as the contingency table represents the outcome of a set of instances when a classification algorithm is applied to a data set(Fawcett, 2004). It contains information about the actual and predicted classes and the data in the matrix is used to evaluate the classifier performance. In order words, the percent accuracy, ROC area, PRC area and other performance measures are evaluated from the confusion matrix.

Table 4.5 shows a two-by-two matrix confusion matrix for the binary class attributes ‘DIED’ and ‘SURVIVED’ and the results generated from the classification exercise with the diagonal results representing the correct decisions made while the error is represented on either side of the diagonal. The confusion matrix compares the number of correct and incorrect predictions made by the random forest model to the actual outcomes in the data set.

Table 4.5 Confusion matrix generated from the random forest classifier.

Confusion Matrix		**Predicted (Random Forest)**			
		‘DIED'	‘SURVIVED'		
True Class	'DIED'	**TP** = 313	**FN** = 59	***Sensitivity (True Positive rate)***	0.841
	‘SURVIVED'	**FP** = 72	**TN** = 314	***Specificity (True Negative Rate)***	0.813
Precision		0.813		***Accuracy*= 0.827**	

TP: True Positive class; FP: False Positive class; TN: True Negative class; FN: False Negative class.

True Positive Rate (TPR): The true positive rate is also referred as the recall or sensitivity in terms of how the classifier has performed. It is the proportion of correctly predicted positive cases which is calculated with the equation below;

$$\mathrm{TPR}= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

For this research, it measures the performance of the classifier to correctly identify a patient that died within one year of having a thoracic surgery. The model generated by the random forest classifier indicates that it correctly classified 84.1 percent of the patients in the ‘DIED’ class.

True Negative Rate (TNR): This is also known as the specificity. It is the proportion of the negative classes that is correctly predicted as negative which is calculated with the equation below;

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

The TNR in the confusion matrix indicates that 81.3 percent of patients in the 'SURVIVED' class were correctly predicted by the classifier.

False Positive Rate (FPR): This is the proportion of negative cases wrongly predicted as positive i.e. the patients that survived wrongly classified as died. It is calculated as;

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

The confusion matrix indicates that 18.7 percent of patients in the 'SURVIVED' class was wrongly predicted as 'DIED'.

False Negative Rate (FNR): This is the proportion of positive cases wrongly predicted as negative, in this case, the proportion of people that died wrongly classified as survived.

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}$$

The confusion matrix indicates that 15.9 percent of patients in the 'DIED' class was wrongly predicted as 'SURVIVED'. This is the same as 1 minus the TPR.

Accuracy: The accuracy is defined as the proportion of all the classifier prediction that were correct, both 'SURVIVED' and 'DIED'. This implies that the error rate would be 1 minus accuracy. This is solved as;

$$\textbf{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}$$

The confusion matrix indicates that 82.7 percent of all the patients were correctly classified into their respective classes.

Precision: This is given as the proportion of the predicted positive instances that were correct. In this situation, the proportion of predicted death cases that were correct. This is solved as;

Precision = $\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$

F- Measure: This is also referred to as the F –score and is mostly used for single number measures in Information Retrieval and machine learning (Powers, 2014). It computes the average of precision and recall metrics in Information retrieval but this average is known as the harmonic mean. Mathematically, this can be written as;

$$\frac{1}{\mathrm{F}} = \frac{1}{2}\left(\frac{1}{\mathrm{R}} + \frac{1}{\mathrm{P}}\right) = \frac{\mathrm{P} + \mathrm{R}}{2\mathrm{PR}}$$

Where $\mathbf{F}$**:** F- measure**,** $\mathbf{P}$: Precision and $\mathbf{R}$**:** Recall.

In Table 4.1, the confusion matrix showed that that the classifier had an accuracy of 0.835 (83.51%).

4.3.2. Interpretation on the prediction results.

The prediction in Weka is achieved by exporting the predictions in a comma-separated value (CSV) format from the classifier evaluation options on the Weka experimental interface. Table 4.6 summarises the result of the prediction for the tenth fold created by cross validation of the dataset. It would be recalled from chapter 3 that after the cleaning phase, we were left with a total of 758 instances. By specifying 10 folds for cross-validation, the dataset was split into ten folds with one-tenth of the fold used for testing and the remaining nine-tenth for training and building the model.

Table 4.6 Table of prediction on test data.

Instances	Actual	Predicted	Error	Prediction
1	2:SURVIVED	2:SURVIVED		0.83
2	2:SURVIVED	2:SURVIVED		0.65
3	2:SURVIVED	2:SURVIVED		0.885
4	2:SURVIVED	2:SURVIVED		0.88
5	2:SURVIVED	2:SURVIVED		0.81
6	2:SURVIVED	2:SURVIVED		0.939
7	2:SURVIVED	1:DIED	+	0.64
8	2:SURVIVED	2:SURVIVED		0.934
9	2:SURVIVED	2:SURVIVED		0.626
10	2:SURVIVED	2:SURVIVED		0.97
11	2:SURVIVED	2:SURVIVED		0.988
12	2:SURVIVED	2:SURVIVED		0.76
13	2:SURVIVED	1:DIED	+	0.54
14	2:SURVIVED	2:SURVIVED		0.731
15	2:SURVIVED	1:DIED	+	0.87
16	2:SURVIVED	2:SURVIVED		0.85
17	2:SURVIVED	2:SURVIVED		0.893
18	2:SURVIVED	2:SURVIVED		0.8
19	2:SURVIVED	2:SURVIVED		0.988
20	2:SURVIVED	2:SURVIVED		0.75
21	2:SURVIVED	2:SURVIVED		0.6
22	2:SURVIVED	2:SURVIVED		0.898
23	2:SURVIVED	2:SURVIVED		0.68
24	2:SURVIVED	2:SURVIVED		1
25	2:SURVIVED	2:SURVIVED		0.831
26	2:SURVIVED	2:SURVIVED		0.78
27	2:SURVIVED	2:SURVIVED		0.9
28	2:SURVIVED	2:SURVIVED		0.904
29	2:SURVIVED	2:SURVIVED		1
30	2:SURVIVED	2:SURVIVED		0.73
31	2:SURVIVED	2:SURVIVED		0.9
32	2:SURVIVED	2:SURVIVED		0.79
33	2:SURVIVED	2:SURVIVED		0.818
34	2:SURVIVED	2:SURVIVED		0.8
35	2:SURVIVED	1:DIED	+	0.63
36	2:SURVIVED	2:SURVIVED		0.588
37	2:SURVIVED	1:DIED	+	0.668
38	2:SURVIVED	1:DIED	+	0.65
39	1:DIED	1:DIED		0.86
40	1:DIED	1:DIED		0.7
41	1:DIED	1:DIED		0.91
42	1:DIED	1:DIED		0.95
43	1:DIED	1:DIED		0.595
44	1:DIED	1:DIED		0.81
45	1:DIED	1:DIED		0.99
46	1:DIED	1:DIED		0.55
47	1:DIED	1:DIED		0.54
48	1:DIED	1:DIED		1
49	1:DIED	1:DIED		0.795
50	1:DIED	1:DIED		0.7
51	1:DIED	1:DIED		0.97
52	1:DIED	1:DIED		0.77
53	1:DIED	1:DIED		0.8
54	1:DIED	1:DIED		1
55	1:DIED	2:SURVIVED	+	0.97
56	1:DIED	1:DIED		0.87
57	1:DIED	1:DIED		0.84
58	1:DIED	1:DIED		0.97
59	1:DIED	1:DIED		0.9
60	1:DIED	1:DIED		0.94
61	1:DIED	1:DIED		0.53
62	1:DIED	1:DIED		0.85
63	1:DIED	2:SURVIVED	+	0.56
64	1:DIED	1:DIED		0.93
65	1:DIED	1:DIED		0.95
66	1:DIED	1:DIED		0.8
67	1:DIED	1:DIED		0.98
68	1:DIED	1:DIED		0.72
69	1:DIED	1:DIED		1
70	1:DIED	2:SURVIVED	+	0.849
71	1:DIED	1:DIED		0.71
72	1:DIED	1:DIED		0.9
73	1:DIED	1:DIED		0.82
74	1:DIED	2:SURVIVED	+	0.53
75	1:DIED	1:DIED		0.95

Actual: *indicates the true class of the instance before the classifier was applied.* ***Predicted:*** *indicates the class as detected by the classifier.* ***Error:*** *a plus symbol (+) indicate an incorrect prediction.* ***Prediction:*** *gives the probability of accuracy for the predicted class*

In Table 4.6, it is observed that there are instances with outcomes of the actual class similar to the outcome of the predicted class indicating a correct prediction while a few instances have different outcomes for the actual and predicted class indicating an incorrect prediction. The degree of prediction accuracy of the classifier is specified by the probability. For example, on the 7th row, the classifier wrongly predicted a patient that survived as died with a 64% probability of accuracy. Also from the table, 10 of the instances were wrongly classified out of the 75 instances.

Sorting the probability distribution (prediction column) in Table 4.6 in descending order results in a cluster of the wrongly classified instances at the bottom of the table. Sorting the last 15 rows as seen in Table 4.7 reveal 7 out of the 10 wrongly classified instances. This elaborates the classification threshold set by the random forest classifier as it can be seen that 70% of errors had low probabilities.

Table 4.7 Sorting the prediction results of the last 15 rows

Instances	Actual	Predicted	Error	Prediction
37	2:SURVIVED	1:DIED	+	0.668
2	2:SURVIVED	2:SURVIVED		0.65
38	2:SURVIVED	1:DIED	+	0.65
7	2:SURVIVED	1:DIED	+	0.64
35	2:SURVIVED	1:DIED	+	0.63
9	2:SURVIVED	2:SURVIVED		0.626
21	2:SURVIVED	2:SURVIVED		0.6
43	1:DIED	1:DIED		0.595
36	2:SURVIVED	2:SURVIVED		0.588
63	1:DIED	2:SURVIVED	+	0.56
46	1:DIED	1:DIED		0.55
13	2:SURVIVED	1:DIED	+	0.54
47	1:DIED	1:DIED		0.54
61	1:DIED	1:DIED		0.53
74	1:DIED	2:SURVIVED	+	0.53

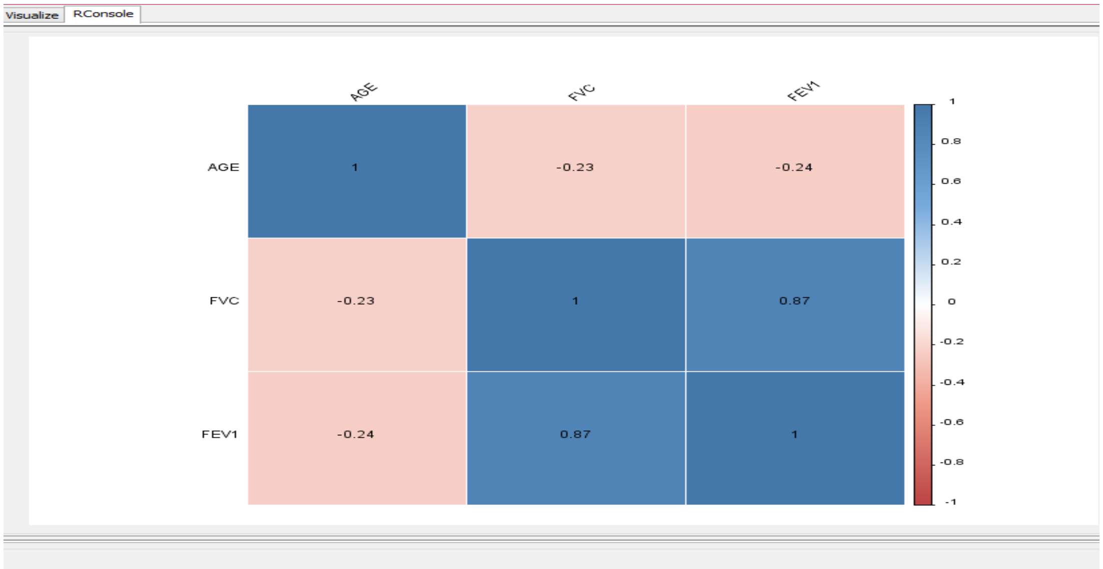
4.4. More exploratory analysis

More exploration on the five attributes selected from the feature selection by wrapping the random forest as indicated in section 3.6.3 was carried out to give a deeper understanding of the relation within the features and why they are the best features for classification as indicated by the classifier. This is done by data exploration on the Rconsole environment of Weka. The results are described below:

4.4.1. Correlation Matrix

A correlation test is a measure of dependence that reveals the relationship between numeric variables in a data set. It helps to search the dependence between multiple variables in a data set by giving the correlation coefficients between the variable data (Winston Chang, 2012). R presents different methods for calculating the correlation coefficients. This includes the Pearson parametric correlation test which is obtained by measuring the linear dependence between two numeric variables, Spearman correlation test and Kendall rank-based correlation tests for a non-parametric rank-based correlation test. The Pearson's correlation test is set as default. A visual representation of the correlation matrix was given by the script below with the output displayed in Figure 4.5:

```
library(corrplot);
Thorcor <- cor(Thoracic[, c(2,3,16)]);
Thorcor <- round(Thorcor,2);
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"));
corrplot(Thorcor, method="shade", shade.col=NA, tl.col="black", tl.srt=45,col=col(2
00), addCoef.col="black", addcolorlabel="no", order="AOE");
```



	AGE	FVC	FEV1
AGE	1	-0.23	-0.24
FVC	-0.23	1	0.87
FEV1	-0.24	0.87	1

Figure 4.5 Correlation matrix with correlation coefficients of the numeric attributes.

Figure 4.5 indicates the positive correlations in blue while the negative correlations are indicated in red. The colour intensity of the boxes are proportional to the correlation coefficients while the colour legends on the right side shows how the value of the correlation coefficients matches different colours. Figure 4.5 can be modified to show the scatterplots, the distribution and the correlation coefficient of the variables with the script below;

```
library(PerformanceAnalytics);
Thor1 <- Thoracic[, c(2,3,16)];
chart.Correlation(Thor1, histogram=TRUE, pch=19)
```

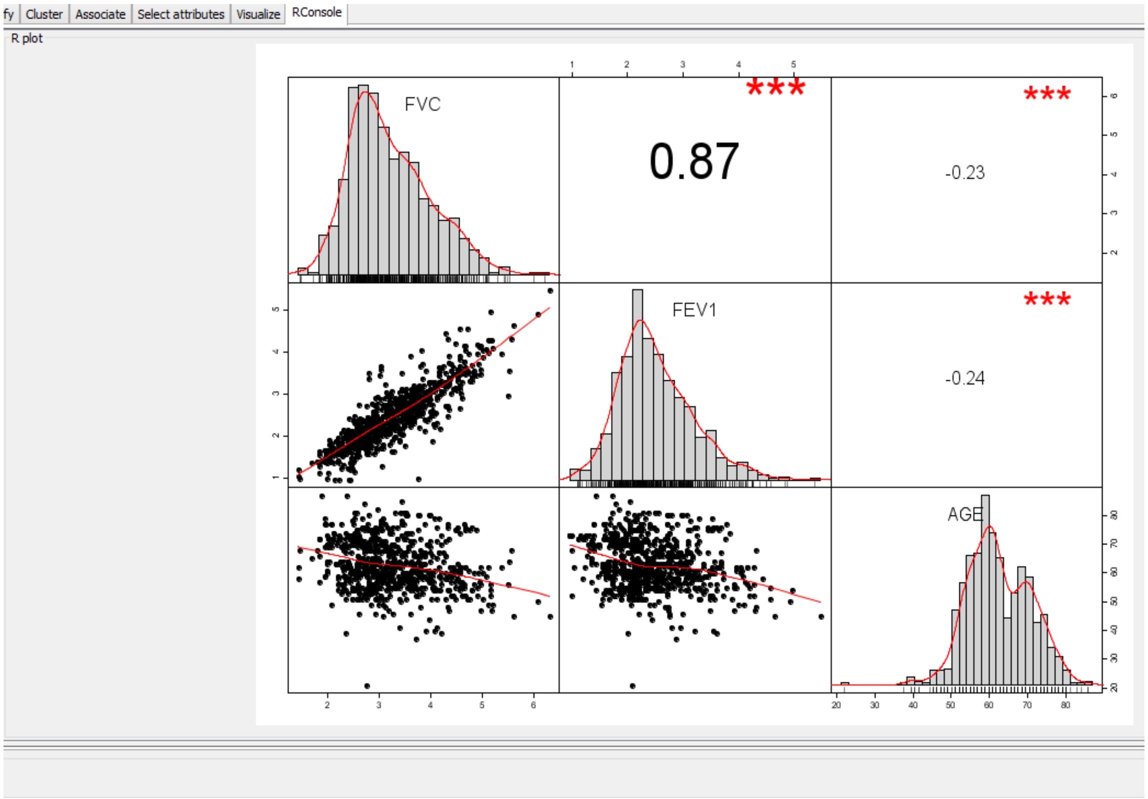


Figure 4.6 The correlation chart

The diagonal line in the chart shows the variable distribution of the numeric attributes. It can be seen that the variables follows a normal distribution. The bottom of the diagonal indicates the bivariate scatter plots with a line of best fit plotted on the graphs while the top of the diagonal indicates the correlation values.

The value of a correlation coefficient is usually between 1 and -1. For the scatter plots indicated at the bottom of the diagonal, a positive slope shows a correlation coefficient between 0 and +1 while a negative slope has a coefficient between 0 and -1. A straight horizontal line has correlation coefficient equal to zero. In addition, for a positive correlation, as the value of the variable on the y axis increases, the corresponding value on the x axis increases and vice-versa while for a negative correlation, as the variable on the y axis decreases, the variable on the x axis increases and vice-versa. This is usually indicated by a positive and negative slope respectively (Tomar and Agarwal, 2014).

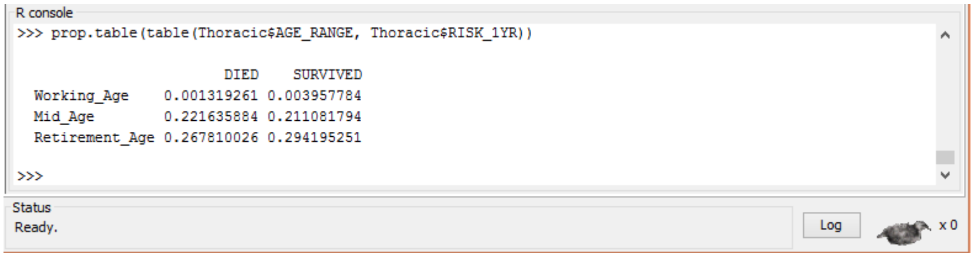
From Figure 4.6, the FVC and FEV1 has a positive slope of 0.87. This implies that as the FVC of the patients increases, the FEV 1 also increase. This relationship therefore influenced the model built by the classifier.

4.4.2. Bar Graphs

These are one of the most used graph for data visualization. Bar graphs are generated by plotting numeric values on the y axis for different discretised categories on the x axis. A discretized attribute known as AGE_RANGE is generated in order to understand more about the age distribution and its relationship with other variables. The age is binned into three groups comprising of Working_Age (21-39), Mid_age (40-60) and Retirement_age (61 and above) categories. This is written with the script below;

```
Thoracic$AGE_RANGE <- ifelse (Thoracic$AGE<40, 'Working_Age', 'Mid_Age');
Thoracic$AGE_RANGE <- ifelse (Thoracic$AGE>60, 'Retirement_Age',
Thoracic$AGE_RANGE);
Thoracic$AGE_RANGE <- ordered (Thoracic$AGE_RANGE, levels = c
("Working_Age","Mid_Age",   "Retirement_Age"));
```

To determine the age range that is most affected by the thoracic disease and the category of age at higher risk of death, the script below gives an aggregate of class attribute with respect to the age_range.



```
>>> prop.table(table(Thoracic$AGE_RANGE, Thoracic$RISK_1YR))

                      DIED    SURVIVED
  Working_Age    0.001319261 0.003957784
  Mid_Age        0.221635884 0.211081794
  Retirement_Age 0.267810026 0.294195251

>>>
```

Figure 4.7 Probability distributions of the age range.

This can be further understood by visualizing a bar plot of this output. This is depicted by running the scripts below on the Weka R console;

```
library(‘plyr’, ‘dplyr’, ‘ggplot2’ );
```

```
Thor <- Thoracic %>% group_by(RISK_1YR,AGE_RANGE)%>%  summarise(Total = n());
Thor <- ddply(Thor, "AGE_RANGE", transform, label_y= cumsum(Total));
g1  <-  ggplot(Thor,  aes(AGE_RANGE,Total,  fill=RISK_1YR))  +  geom_bar(stat="identity")
+geom_text(aes(y=label_y,label=Total),vjust=1.5,                         colour="white")
+scale_size(range=c(3,6))
g2 <- ggplot(Thor, aes(x=AGE_RANGE, y=Total, fill=RISK_1YR)) +
geom_bar(stat="identity") +geom_text(aes(y=label_y, label=paste(Total,",",
round(Total/sum(Total)*100, 1), "%")), vjust=1.5, colour="white")
+scale_size(range=c(3,6))

library(gridExtra);
grid.arrange(g1, g2)
```

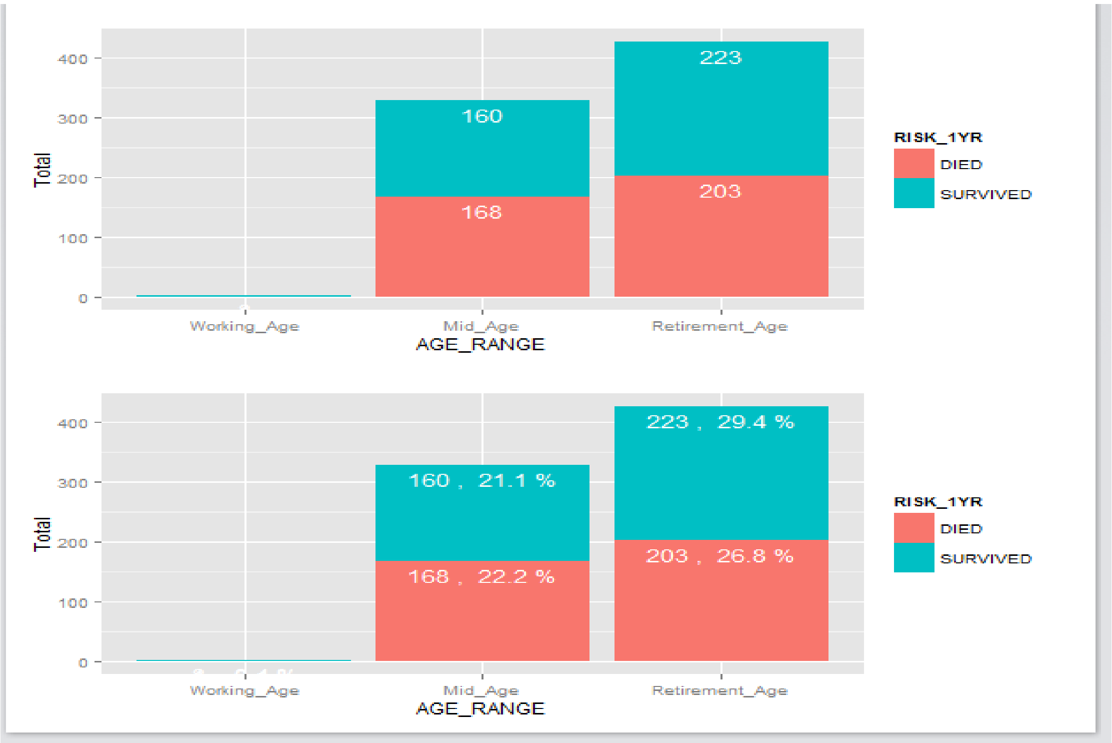


Figure 4.8 Bar chart of different age ranges.

The bar chart in Figure 4 .8 indicates that patients in the working age category are hardly affected by lung cancer and if affected at all, they are not likely to have a surgery until they have reached the mid_age category. Comparing these percentages in the chart of patients that survived and died in the mid_age and retirement_age categories, though we have a higher percentage of retired patients that died, It is interesting to see that after the operation, they had a higher probability of surviving. This is because the percentage of survival rate outnumbered the percentage of death rate unlike the mid_age categories. More research may need to be done to understand possible reason for this trend. Suggestion of research focus are monitoring the recuperation of patients with respect to factors such as stress and which is not included in the dataset. Other attribute and their relationship with the age range is given by the output below;

```
f1 <-ggplot( Thoracic,aes(AGE_RANGE))+geom_bar(aes(fill=Z.SCALE))+ geom_text(stat=
'bin' ,aes (label= ..count..) ,vjust=-1, size=4);
f2 <-ggplot( Thoracic,aes(AGE_RANGE))+geom_bar(aes(fill=COUGH_BS))+ geom_text(stat=
'bin' ,aes (label= ..count..) ,vjust=-1, size=4);
f3 <-ggplot( Thoracic,aes(AGE_RANGE))+geom_bar(aes(fill=SMOKING))+ geom_text(stat=
'bin' ,aes (label= ..count..) ,vjust=-1, size=4);
f4 <-ggplot( Thoracic,aes(AGE_RANGE))+geom_bar(aes(fill=TUMOUR_SZ))+ geom_text(stat=
'bin' ,aes (label= ..count..) ,vjust=-1, size=4);
f5 <-ggplot( Thoracic,aes(AGE_RANGE))+geom_bar(aes(fill=DGN))+
geom_text(stat='bin',aes (label= ..count..) ,vjust=-1, size=4);
f6 <-ggplot( Thoracic,aes(AGE_RANGE))+geom_bar(aes(fill= HAEMOPTYSIS_BS))+
geom_text(stat= 'bin',aes (label =..count..),vjust=-1, size=4);
library(gridExtra);
grid.arrange(f1,f2,f3,f4,f5,f6, ncol= 3)
```

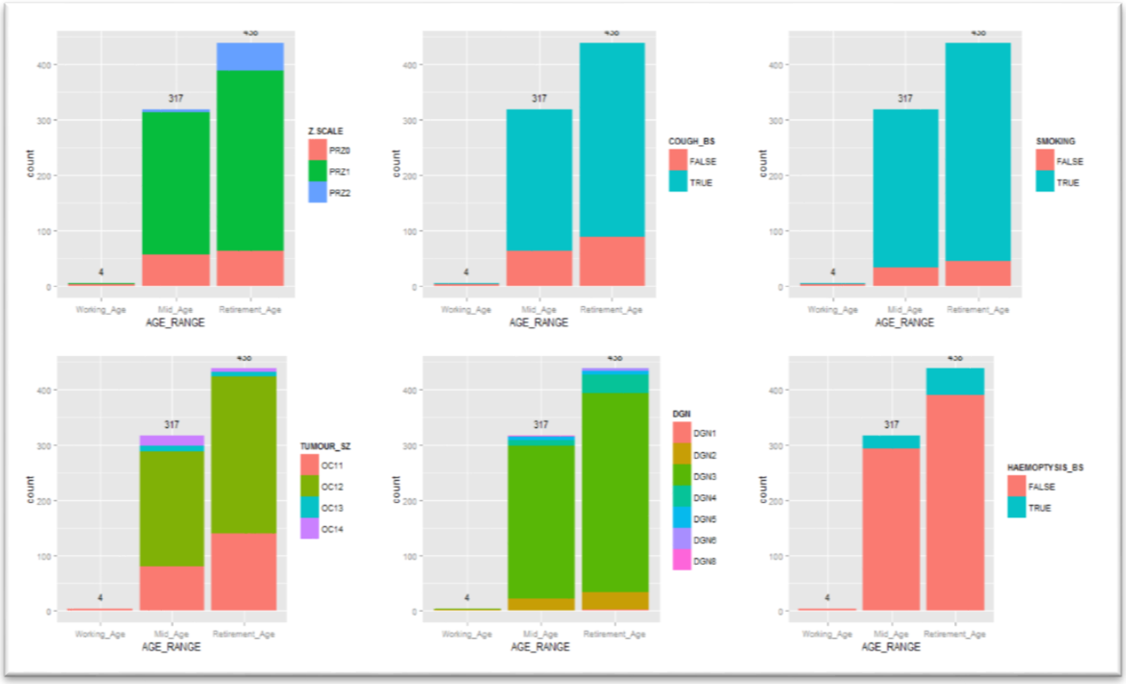


Figure 4.9 Bar chart of the age range and other attributes.

From figure 4.9, certain relationships of the distribution of the variables according to the age_range of the patients are observed. Examples includes; on the Z_SCALE it is seen that the performance status zubrod scale of 1 (PRZ1) is the most occurring among the patients. Also, it can be seen that even though we have OC11 as the smallest tumour size and OC14 as the largest, OC12 seem to be the most occurring size of tumour found in patients that underwent surgery.

4.4.3. Frequency polygon

The frequency polygon gives the same information as the histogram and it is mainly used to extract more information from the graph. Plotting all the binary attributes versus the age is done with the script below;

```
a1<- ggplot(Thoracic, aes(AGE))+geom_freqpoly(aes(color= PAIN_BS),binwidth=1)+ scale_x_
continuous(breaks =seq(0,100, 5));
a2<-  ggplot(Thoracic,  aes(AGE))+geom_freqpoly(aes(color=  HAEMOPTYSIS_BS),binwidth=1)+
scale_x_continuous(breaks =seq(0,100, 5));
a3<-    ggplot(Thoracic,    aes(AGE))+geom_freqpoly(aes(color=DYSPNOEA_BS),binwidth=1)+
scale_ x_continuous(breaks =seq(0,100, 5));
```

```
a4<- ggplot(Thoracic, aes(AGE))+geom_freqpoly(aes(color= COUGH_BS),binwidth=1) +scale_x
_continuous(breaks =seq(0,100, 5));
a5<-   ggplot(Thoracic,   aes(AGE))+geom_freqpoly(aes(color=   WEAKNESS_BS),binwidth=1)
+scale _ x_continuous(breaks =seq(0,100, 5));
a6<-    ggplot(Thoracic,    aes(AGE))+geom_freqpoly(aes(color=    DIABETES),binwidth=1)+
scale_x_ continuous(breaks =seq(0,100, 5));
a7<-   ggplot(Thoracic,   aes(AGE))+geom_freqpoly(aes(color=   MI_6MONTHS),binwidth=1)+
scale_x_ continuous(breaks =seq(0,100, 5));
a8<- ggplot(Thoracic, aes(AGE))+geom_freqpoly(aes(color= PAD), binwidth=1)+ scale_x_
continuous(breaks =seq(0,100, 5));
a9<- ggplot(Thoracic, aes(AGE))+geom_freqpoly(aes(color= SMOKING),binwidth=1)+ scale_x_
continuous(breaks =seq(0,100, 5));
a10<- ggplot(Thoracic, aes(AGE))+geom_freqpoly(aes(color=ASTHMA),binwidth=1)+ scale_x_
continuous(breaks =seq(0,100, 5));
grid.arrange(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10)
```

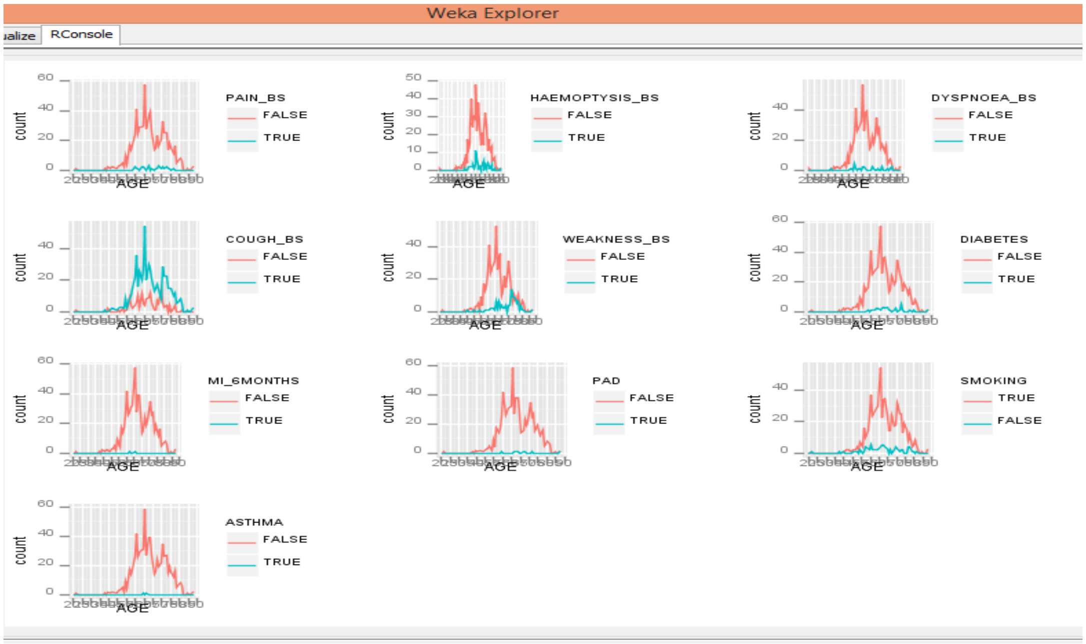


Figure 4.10 A frequency polygon of the age and the binary attributes.

This results presented in Figure 4.10 shows that of all the ailments, cough was most prevalent across all the age groups with the blue line surpassing the red line. This supports why the random forest algorithm identifies this attribute amongst the relevant features.

4.5. Applications of the classification results.

Though the use of machine learning tools for medical diagnosis often outperform the judgements of experts in the medical field in terms of specificity, sensitivity and accuracy, they are rarely used in the medical practice (Kukar, 2001). One very important element of any machine learning process is establishing a reliability estimation of the classification model i.e. establishing a probability that the classification result would correctly classify the patients. The model that was built with the random forest algorithm has an overall accuracy indicating an 82.7 percent probability of correctly separating the patients that would die from those that survive within a year after undergoing thoracic surgery. Also, the prediction table includes a prediction column that indicates the accuracy probability of each predicted instance. This can improve medical practice in the following ways;

1. Provides an accurate research forecasts on evidence-based based diagnosis and treatments and improve patient outcomes.
2. Serve as a guide for medical practitioners in deciding the risk factors associated with a surgery type that can be performed on a patients with lung cancer based on the variables of previous records. For example, the classifier output like that in Table 4.6 may indicate a higher risk for smokers that undergo a particular kind surgery to die within a year given other factors like their age, tumour size, zuboid scale etc. Thus, given a higher risk of surgery, other possible treatment options may be considered.
3. Serve as a guide for the health sector to advice citizens on risk factors associated with a certain kind of disease and help government with policies to restrict access to some products.
4. Could guide in medical practitioners in counselling their patients on available treatment options available to them so that they can both agree on the best choice of treatment procedure.
5. Having the right diagnosis can help reduce tests and pains undergone from unnecessary treatment procedures, save a lot of time and money for the patients and the health sector.

CHAPTER 5

CONCLUSION

This chapter presents a summary of the research by highlighting how it answers the research questions. This chapter also identifies the limitation experienced during this study; stating the learning outcome and recommendations for future work.

5.1. Key findings

The following findings were made during this research:

1. The performance of the Weka software tool can be boosted by integrating the R data analysis software.
2. Data cleaning and dimension reduction as a pre-processing steps in data analysis are very vital as they help remove outliers that could distort the prediction results and simplify the overall analysis.
3. The random forest classifier was the most suitable model for classification on the dataset based on evaluation of the outcome of the ROC area, the PRC area and the percentage accuracy.
4. The insufficiency of adapting the classifier accuracy as the only basis for measuring classifier performance was affirmed as the rank of classifiers that seemed to perform well when their accuracy was measured dropped after the ROC and PRC area was investigated.
5. To avoid weakening the performance of classification models on an imbalanced dataset, sampling could be done at the data level or applying a cost-sensitive boosting at the algorithmic level.
6. The best features of the dataset as recommended by the model for making decision of the health of a thoracic surgery patient were FVC, FEV1 AGE, COUGH_BS and TUMOUR_SZ of the lung cancer.
7. Applying exploratory analysis on a dataset is an iterative process that exposes hidden relationships between features of a dataset and helps the researcher to make analytical decisions.

8. Based on this study, the random forest outperformed other algorithms by correctly diagnosis 84.1 and 81.3 percent of patients that died and survived respectively with an overall accuracy of 82.72 percent.

5.2. Limitations of study

In the course of building the model, some limitations were experienced from the software, applying the algorithms and working with the dataset. With respect to software use, limitations experienced include:

1. The use of other available R packages such as the 'R markdown' package which would have enabled easy export of the plotted graphs to Microsoft word document or pdf file was not possible from the Weka environment and thus reducing the visibility of some of the graphs.
2. Integrating R and Weka caused a minute reduction in the processing time of Weka which may be have been significant if a much larger data set was used for this research.
3. There was limited time to learn the use of the packages available in both R and Weka which would have improved the outcome of the research.

With respect to algorithm use, the challenges experienced includes:

1. The application of a cost matrix was introduced but the making a decision on the threshold to specify would need more input from medical professionals.
2. The practice of health analytics would need more interaction between the analyst and the medical specialist especially at the post-processing stage for some feedback mechanism to improve the initial model that was built but this was not available.

While with respect to dataset used:

1. The data set was already structured and cleaned and as a result did not fully portray a typical health data set which often contains some inconsistencies and missing values. This limited the demonstration the various aspects of the pre-processing stage.
2. The obtained result is limited to the region where the data was collated and the time frame.
3. The data set is small and may not sufficiently demonstrate the effectiveness of other sophisticated classification algorithms.

5.3. Learning outcomes

The project has helped the researcher to:

1. Gain exposure to the R data analysis software and an improved the knowledge of Weka.
2. Understand the operations of the various single classification algorithms and multiple algorithms and how to apply them methodologically in predictive analytics.
3. Understand the various ways of measuring the performance of classification algorithms.
4. Understand the practical application data visualization in business analysis.

5.4. Suggestions for future work

Several limitations that were experienced in this study from areas that may be considered for future study. These include:

1. Application of a cost- sensitive classifier to build the predictive model.
2. Exploring the use of other combinations of analytical software to enhance the knowledge discovery process.
3. Considering the possibility of boosting the performance of the random forest classifier with ensemble learners or by using stacking or voting on other available classifiers.
4. Development and enhancement of the random forest classifier for a large scale data set by applying feature engineering on the attributes as a form of improvement and validation of the classifier performance.
5. Consideration of new methods of dealing with the imbalanced class.

REFERENCES

Abeel, T., Van de Peer, Y. & Saeys, Y. (2009) 'Java-ML: A machine learning library'. *The Journal of Machine Learning Research,* 10 pp.931-934.

Achtert, E., Bernecker, T., Kriegel, H.-P., Schubert, E. & Zimek, A. (2009) '*ELKI in time: ELKI 0.2 for the performance evaluation of distance measures for time series'*. *Advances in Spatial and Temporal Databases.* Springer, pp. 436-440.

Adam, A., Ivaylo, B. & Jia, P. (2014) '*Life Expectancy Post Thoracic Surgery'*. Stanford University.

Amadio, W.J., Pelletteri, B.M. & Krall, I.I.I.J.S. (2014) 'MODERN DATA ANALYTICS FOR DECISION MAKING'. *Global Conference on Business & Finance Proceedings,* 9 (2), pp.247-252.

Bauer, E. & Kohavi, R. (1999) 'An empirical comparison of voting classification algorithms: Bagging, boosting, and variants'. *Machine learning,* 36 (1-2), pp.105-139.

Berger, H., Merkl, D. & Dittenbach, M. (2006) Published. 'Exploiting partial decision trees for feature subset selection in e-mail categorization'. *Proceedings of the 2006 ACM symposium on Applied computing*, 2006. ACM, pp.1105-1109.

Blagus, R. & Lusa, L. (2013) 'SMOTE for high-dimensional class-imbalanced data'. *BMC Bioinformatics,* 14 pp.106.

Bolon-Canedo, V., Sanchez-Marono, N. & Alonso-Betanzos, A. (2013) 'A review of feature selection methods on synthetic data'. *Knowledge and Information Systems,* 34 (3), pp.483-519.

Bose, R. & Sugumaran, V. (1999) 'Application of intelligent agent technology for managerial data analysis and mining'. *Database for Advances in Information Systems,* 30 (1), pp.77-94.

Bouckaert, R.R. (2004) *Bayesian network classifiers in weka.* Department of Computer Science, University of Waikato.

Brefeld, U., Geibel, P. & Wysotzki, F. (2003) '*Support vector machines with example dependent costs'*. *Machine Learning: ECML 2003.* Springer, pp. 23-34.

Breiman, L. (2001) 'Random forests'. *Machine learning,* 45 (1), pp.5-32.

Bruha, I. & Famili, A. (2000) 'Postprocessing in machine learning and data mining'. *ACM SIGKDD Explorations Newsletter,* 2 (2), pp.110-114.

Carslaw, D.C. & Ropkins, K. (2012) 'openair - An R package for air quality data analysis'. *Environmental Modelling & Software,* 27-28 pp.52-61.

Centor, R.M. & Keightley, G.E. (1989) '*Receiver Operating Characteristics (ROC) curve area analysis using the ROC ANALYZER*'.

Chang, C.-C. & Lin, C.-J. (2011) 'LIBSVM: A library for support vector machines'. *ACM Transactions on Intelligent Systems and Technology (TIST),* 2 (3), pp.27.

Chang, W. (2012) *R graphics cookbook.* " O'Reilly Media, Inc.".

Chawla, N.V., Japkowicz, N. & Kotcz, A. (2004) 'Editorial: special issue on learning from imbalanced data sets'. *ACM Sigkdd Explorations Newsletter,* 6 (1), pp.1-6.

Chopra, K.N. (2014) 'Modeling and Technical Analysis of Electronics Commerce and Predictive Analytics'. *Journal of Internet Banking & Commerce,* 19 (2), pp.1-10.

Corrigan, H.B., Craciun, G. & Powell, A.M. (2014) 'How Does Target Know So Much About Its Customers? Utilizing Customer Analytics to Make Marketing Decisions'. *Marketing Education Review,* 24 (2), pp.159-166.

Cortes, C. & Vapnik, V. (1995) 'Support-vector networks'. *Machine learning,* 20 (3), pp.273-297.

Danjuma, K.J. (2015) 'Performance Evaluation of Machine Learning Algorithms in Post-operative Life Expectancy in the Lung Cancer Patients'. *arXiv preprint arXiv:1504.04646*.

Das, S., Sismanis, Y., Beyer, K.S., Gemulla, R., Haas, P.J. & McPherson, J. (2010) Published. 'Ricardo: integrating R and Hadoop'. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010. ACM, pp.987-998.

Dasu, T. & Johnson, T. (2003) *Exploratory data mining and data cleaning.* John Wiley & Sons.

Dattero, R., White, E.M. & Janson, M.A. (1991) 'Methods for the Identification of Data Outliers in Interactive SQL'. *Journal of Database Management (JDM),* 2 (1), pp.7-18.

Davenport, T.H. (2006) 'Competing on analytics'. *harvard business review,* (84), pp.98-107, 134.

Davis, J. & Goadrich, M. (2006) Published. 'The relationship between Precision-Recall and ROC curves'. *Proceedings of the 23rd international conference on Machine learning*, 2006. ACM, pp.233-240.

Dietterich, T.G. (2000) '*Ensemble methods in machine learning'. Multiple classifier systems.* Springer, pp. 1-15.

Dogan, N. & Tanrikulu, Z. (2013) 'A comparative analysis of classification algorithms in data mining for accuracy, speed and robustness'. *Information Technology and Management,* 14 (2), pp.105-124.

Domingos, P. (1999) Published. 'Metacost: A general method for making classifiers cost-sensitive'. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999. ACM, pp.155-164.

Donalek, C. (2014) '*Supervised and Unsupervised learning'*.

Dong, L., Frank, E. & Kramer, S. (2005) '*Ensembles of balanced nested dichotomies for multi-class problems'*. *Knowledge Discovery in Databases: PKDD 2005.* Springer, pp. 84-95.

Fawcett, T. (2004) 'ROC graphs: Notes and practical considerations for researchers'. *Machine learning,* 31 pp.1-38.

Fayyad, U.M., Piatetsky-Shapiro, G. & Smyth, P. (1996) Published. 'Knowledge Discovery and Data Mining: Towards a Unifying Framework'. *KDD*, 1996. pp.82-88.

Freitas, A.A. (2003) '*A survey of evolutionary algorithms for data mining and knowledge discovery'*. *Advances in evolutionary computing.* Springer, pp. 819-845.

Freund, Y. & Schapire, R.E. (1997) 'A decision-theoretic generalization of on-line learning and an application to boosting'. *Journal of computer and system sciences,* 55 (1), pp.119-139.

Gandomi, A. & Haider, M. (2015) 'Beyond the hype: Big data concepts, methods, and analytics'. *International Journal of Information Management,* 35 (2), pp.137-144.

Goadrich, M., Oliphant, L. & Shavlik, J. (2006) 'Gleaner: Creating ensembles of first-order clauses to improve recall-precision curves'. *Machine Learning,* 64 (1-3), pp.231-261.

Gobble, M.M. (2013) 'Big data: The next big thing in innovation'. *Research-Technology Management,* 56 (1), pp.64.

Goes, P.B. (2014) 'Big Data and IS Research'. *MIS Quarterly,* 38 (3), pp.iii-viii.

Granholm, V., Noble, W.S. & Käll, L. (2012) 'A cross-validation scheme for machine learning algorithms in shotgun proteomics'. *BMC Bioinformatics,* 13 (Suppl 16), pp.S3.

Grossman, R.L. & Siegel, K.P. (2014) 'ORGANIZATIONAL MODELS FOR BIG DATA AND ANALYTICS'. *Journal of Organization Design,* 3 (1), pp.20-25.

Gupta, M. & Palmer, R.J. (2007) 'Government Efficiency versus Accountability: How an Emerging Control Model for Purchase Card Use May Enable US Government Agencies to Achieve Both Goals'. *Public Contract Law Journal,* pp.175-201.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I.H. (2009) 'The WEKA data mining software: an update'. *ACM SIGKDD explorations newsletter,* 11 (1), pp.10-18.

Han, J., Kamber, M. & Pei, J. (2011) *Data mining: concepts and techniques: concepts and techniques.* Elsevier.

Hand, D.J. (2006) 'Classifier technology and the illusion of progress'. *Statistical science,* 21 (1), pp.1-14.

Hartwig, F. & Dearing, B.E. (1979) *Exploratory data analysis.* Sage.

Harun, A.U. & Alam, N. (2015) 'Predicting Outcome of Thoracic Surgery by Data Mining Techniques'. *International Journal of Advanced Research in Computer Science and Software Engineering,* 5 (1), pp.7-10.

Hearst, M.A., Dumais, S.T., Osman, E., Platt, J. & Scholkopf, B. (1998) 'Support vector machines'. *Intelligent Systems and their Applications, IEEE,* 13 (4), pp.18-28.

Hickey, S.J. (2013) 'Naive Bayes Classification of Public Health Data with Greedy Feature Selection'. *Communications of the IIMA,* 13 (2), pp.87.

Hornik, K., Buchta, C. & Zeileis, A. (2009) 'Open-source machine learning: R meets Weka'. *Computational Statistics,* 24 (2), pp.225-232.

Horvath, A.O. & Symonds, B.D. (1991) 'Relation between working alliance and outcome in psychotherapy: A meta-analysis'. *Journal of counseling psychology,* 38 (2), pp.139.

Hsinchun, C., Chiang, R.H.L. & Storey, V.C. (2012) 'BUSINESS INTELLIGENCE AND ANALYTICS: FROM BIG DATA TO BIG IMPACT'. *MIS Quarterly,* 36 (4), pp.1165-1188.

Hu, F., Liu, X., Dai, J. & Yu, H. (2014) 'A Novel Algorithm for Imbalance Data Classification Based on Neighborhood Hypergraph'. *The Scientific World Journal,* 2014.

Huang, J. & Ling, C.X. (2005) 'Using AUC and accuracy in evaluating learning algorithms'. *Knowledge and Data Engineering, IEEE Transactions on,* 17 (3), pp.299-310.

James Martin (2015) Healthcare analytics market – *global industry analysis, size, share, growth, trends and forecast published by a leading research firm* [WhaTech] Available from: http://www.whatech.com/market-research/medical/81610-healthcare-analytics-market-global-industry-analysis-size-share-growth-trends-and-forecast-published-by-leading-research-firm [Last accessed 17th August 2015]

Karegowda, A.G., Manjunath, A.S. & Jayaram, M.A. (2010) 'Feature Subset Selection Problem using Wrapper Approach in Supervised Learning'. *International Journal of Computer Applications,* 1 (7).

Kdnuggets (2012) *Poll results: Top languages for analytics/data mining programming.* Available from: http://www.kdnuggets.com/2012/08/poll-analytics-data-mining-programming-languages.html [Last accessed 10th July 2015]

Keilwagen, J., Grosse, I. & Grau, J. (2014) 'Area under precision-recall curves for weighted and unweighted data'. *PloS one,* 9 (3), pp.e92209.

Kononenko, I. (2001) 'Machine learning for medical diagnosis: history, state of the art and perspective'. *Artificial Intelligence in medicine,* 23 (1), pp.89-109.

Kubat, M., Holte, R.C. & Matwin, S. (1998) 'Machine learning for the detection of oil spills in satellite radar images'. *Machine learning,* 30 (2-3), pp.195-215.

Kumar, A., Niu, F. & Ré, C. (2013) 'Hazy: Making it easier to build and maintain big-data analytics'. *Communications of the ACM,* 56 (3), pp.40-49.

Kukar, M. (2001) *Making reliable diagnoses with machine learning: A case study.* Springer.

Kukuyeva, I. (2009) '*Graphics for Exploratory Data Analysis in R'*. UCLA Department of Statistics R Bootcamp.

Kuncheva, L.I. (2004) 'Combining Pattern Classifiers'. *Methods and Algorithms. Wiley, Chichester*.

Laurikkala, J. (2001) *Improving identification of difficult small classes by balancing class distribution.* Springer.

Lavanya, D. & Rani, K.U. (2011) 'Performance evaluation of decision tree classifiers on medical datasets'. *IJCA) International Journal of Computer Applications,* 26 (4).

Maindonald, J. (2006) '*Data mining methodological weaknesses and suggested fixes*'. *Proceedings of the fifth Australasian conference on Data mining and analystics - Volume 61*. Sydney, Australia, 1273810: Australian Computer Society, Inc.

Maroco, J., Silva, D., Rodrigues, A., Guerreiro, M., Santana, I. & de Mendonça, A. (2011) 'Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests'. *BMC Research Notes,* 4 pp.299.

Meyer, D. & Wien, F.T. (2014) 'Support vector machines'. *The Interface to libsvm in package e1071*.

Muellner, M.K., Duernberger, G., Ganglberger, F., Kerzendorfer, C., Uras, I.Z., Schoenegger, A., Bagienski, K., Colinge, J. & Nijman, S.M.B. (2014) 'TOPS: a versatile software tool for statistical analysis and visualization of combinatorial gene-gene and gene-drug interaction screens'. *BMC Bioinformatics,* 15 (1), pp.98.

Mike Miliard (2015) *Clinical decision support system: no longer just a nice-to-have 2014.*[Healthcare IT News] Available from: http://www.healthcareitnews.com/news/clinical-decision-support-no-longer-just-nice-have [Last accessed 26th August 2015]

Muenchen, R.A. (2011) *R for SAS and SPSS users.* Springer Science & Business Media.

Olson, D.L. & Delen, D. (2008) *Advanced data mining techniques.* Springer Science & Business Media.

Oza, N.C. (2008) 'Ensemble data mining methods'. *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications,* 3.

Oztekin, A., Delen, D. & Kong, Z.J. (2009) 'Predicting the graft survival for heart–lung transplantation patients: An integrated data mining methodology'. *international journal of medical informatics,* 78 (12), pp.e84-e96.

Pal, M. (2005) 'Random forest classifier for remote sensing classification'. *International Journal of Remote Sensing,* 26 (1), pp.217-222.

Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S. & Stonebraker, M. (2009) Published. 'A comparison of approaches to large-scale data analysis'. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, pp.165-178.

Platt, J. (1999) 'Fast training of support vector machines using sequential minimal optimization'. *Advances in kernel methods—support vector learning,* 3.

Powers, D.M. (2014) 'What the F--measure doesn’t measure…'.

Pramokchon, P. & Piamsa-nga, P. (2014) 'Content-Adaptive Feature Selection for Classifying Class-Imbalanced Data'. *International Journal of Advancements in Computing Technology,* 6 (5), pp.66.

Provost, F.J., Fawcett, T. & Kohavi, R. (1998 Published. 'The case against accuracy estimation for comparing induction algorithms'. *ICML*, 1998. pp.445-453.

Quinlan, J.R. (1994) Published. 'Comparing connectionist and symbolic learning methods'. *Computational Learning Theory and Natural Learning Systems: Constraints and Prospects*, 1994. Citeseer.

Raghupathi, W. & Raghupathi, V. (2014) 'Big data analytics in healthcare: promise and potential'. *Health Information Science and Systems,* 2 (1), pp.3.

Rangarajan, L. (2010) 'Bi-level dimensionality reduction methods using feature selection and feature extraction'. *International Journal of Computer Applications,* 4 (2).

Rish, I. (2001) Published. 'An empirical study of the naive Bayes classifier'. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001. IBM New York, pp.41-46.

Rushdi, S. (2012) '*Weka Tutorial 13: Stacking Multiple Classifiers (Classification)*'. Available from https://www.youtube.com/watch?v=Nje8mblA7bs [Last accessed 20th June 2015]

Sabate, S., Mazo, V. & Canet, J. (2014) 'Predicting postoperative pulmonary complications: implications for outcomes and costs'. *Curr Opin Anaesthesiol,* 27 (2), pp.201-209.

Sahin, Y., Bulkan, S. & Duman, E. (2013) 'A cost-sensitive decision tree approach for fraud detection'. *Expert Systems with Applications,* 40 (15), pp.5916-5923.

SAS Institute (2009) *Analytics in healthcare.* Available from http://www.sas.com/en_au/whitepapers /analytics-healthcare-102465.html [Last accessed 21st July 2015]

Sigletos, G., Paliouras, G., Spyropoulos, C.D. & Hatzopoulos, M. (2005) 'Combining information extraction systems using voting and stacked generalization'. *The Journal of Machine Learning Research,* 6 pp.1751-1782.

Sindhu.V, S.A. Sathya Prabha, S.Veni & M.Hemalatha (2014) 'Thoracic Surgery Analysis using Data Mining Techniques'. *International Journal of Computer Technology and Applications.,* 5 (2), pp.578-586.

Sun, Y., Kamel, M.S. & Wang, Y. (2006) Published. 'Boosting for learning multiple classes with imbalanced class distribution' Proceedings of the *Sixth International Conference on Data Mining (ICDM'06*). IEEE, pp.592-602.

Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P. & Feuston, B.P. (2003) 'Random forest: a classification and regression tool for compound classification and QSAR modeling'. *Journal of chemical information and computer sciences,* 43 (6), pp.1947-1958.

Tim Morin (2012) VigiLanz Predicts U.S. Supreme Court Decision Upholding Affordable Care Act to Result in Rapid Adoption of Real-Time Decision-Making in American Health Care [Reuters] Available from: http://www.reuters.com/article/2012/06/29/idUS181140+29-Jun-2012+BW20120629 [Last accessed 29th July 2015]

Tippmann, S. (2015) 'Programming Tools: Adventures with R'. *Nature,* 517 (7532), pp.109-110.

Tomar, D. & Agarwal, S. (2014) 'A survey on pre-processing and postprocessing techniques in data mining'. *International Journal of Database Theory & Application,* 7 (4).

Valentini, G. & Masulli, F. (2002) '*Ensembles of learning machines'*. *Neural Nets.* Springer, pp. 3-20.

Van Valkenhoef, G., Tervonen, T., Zwinkels, T., De Brock, B. & Hillege, H. (2013) 'ADDIS: a decision support system for evidence-based medicine'. *Decision Support Systems,* 55 (2), pp.459-475.

Veeranjaneyulu, N., Bhat, M.N. & Raghunath, A. (2014) 'Approaches for Managing and Analyzing Unstructured Data'. *International Journal on Computer Science and Engineering,* 6 (1), pp.19-24.

WebMD (2010) COPD Diagnostic Tests: Pulmonary Function, Spirometry, and More..Available from: http://www.webmd.com/lung/copd/diagnostic-tests [Last accessed 23rd August 2015]

Wever, A. & Maiden, N. (2011) Published. 'What are the day-to-day factors that are preventing business analysts from effective business analysis?'. *Requirements Engineering Conference (RE), 2011 19th IEEE International*. pp.293-298.

Wickham, H. (2014) 'Tidy data'. *Under review*.

Witten, I.H. & Frank, E. (2005) *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann.

Witten, I.H., Frank, E., Trigg, L.E., Hall, M.A., Holmes, G. & Cunningham, S.J. (1999) 'Weka: Practical machine learning tools and techniques with Java implementations'.

Womack, R. (2014) 'Data Visualization and R'.

Xu, L., Krzyżak, A. & Suen, C.Y. (1992) 'Methods of combining multiple classifiers and their applications to handwriting recognition'. *Systems, man and cybernetics, IEEE transactions on,* 22 (3), pp.418-435.

Yang, H., Fong, S., Wong, R. & Sun, G. (2013) 'Optimizing classification decision trees by using weighted naïve bayes predictors to reduce the imbalanced class problem in wireless sensor network'. *International Journal of Distributed Sensor Networks,* 2013.

Yang, P., Xu, L., Zhou, B.B., Zhang, Z. & Zomaya, A.Y. (2009) 'A particle swarm based hybrid system for imbalanced medical data sampling'. *BMC genomics,* 10 (Suppl 3), pp.S34.

Ying, L. (2014) 'Big Data and Predictive Business Analytics'. *Journal of Business Forecasting,* 33 (4), pp.40-42.

Zhou, Z.-H. & Liu, X.-Y. (2006) 'Training cost-sensitive neural networks with methods addressing the class imbalance problem'. *Knowledge and Data Engineering, IEEE Transactions on,* 18 (1), pp.63-77.

Zięba, M., Tomczak, J.M., Lubicz, M. & Świątek, J. (2014) 'Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients'. *Applied Soft Computing,* 14 pp.99-108.

Übeyli, E.D. & Güler, İ. (2005) 'Improving medical diagnostic accuracy of ultrasound Doppler signals by combining neural network models'. *Computers in Biology and Medicine,* 35 (6), pp.533-554.