Developing Risk of Mortality and Early Warning Score Models using Routinely Collected Data

by

Tessy Badriyah

The thesis is submitted in partial fulfilment of the requirements for the award of the degree of Doctor Philosophy of the University of Portsmouth

September 2013

Abstract

Aim. The aim of this study was to contribute to the building of effective and efficient methods to predict adverse clinical outcome. It has been done by developing risk of mortality and early warning score models using routinely collected data that are available from hospital computer systems.

Methods. To predict risk of mortality, firstly we used logistic regression using (Biochemistry and Haematology Outcome Model - BHOM dataset) to generate a model, and the performance of each model was then compared using discrimination (AUROC or c-index) and calibration (the Hosmer-Lemeshow test). Secondly, we focused on decision trees (DT) to be compared with logistic regression (LR). In addition, we used cross validation to compare LR with other various machine learning methods. We developed early warning score algorithmically using decision trees (DTEWS) using vital sign dataset and compared the performance of DTEWS with other EWSs based on clinical expertise using c-index, early warning score efficiency curve and distribution score. We also compared DTEWS with another EWS based on statistics and applied DTEWS to BHOM dataset.

Results. In BHOM dataset, there were 9497 adult hospital discharges, and it was divided into four subsets. A model was built using one training set and then applied to three other testing data sets. The model in logistic regression satisfied both discrimination and calibration value when the c-index in the range 0.700-0.800 is reasonable discrimination and the p-value > 0.05 indicates there is no evidence of significant lack of fit. We also found that decision trees gave a satisfactory result followed by some other machine learning methods.Using a large vital signs dataset (n = 198,755 observation sets) from acute medical admissions, DTEWS can provide a discrimination (c-index) as good as other EWSs, has a better c-index, and also is better in other measurements including EWS efficiency curve, and distribution of score. We found DTEWS can also be applied to BHOM dataset with satisfactory results.

Conclusion. The results of this study support the idea that decision trees can be applied to medical problems. When we produced a model for risk of mortality, we have shown that the decision trees model has reasonable discrimination and could be considered as an alternative technique to logistic regression. We have shown that a structured methodology using decision trees to develop early warning score has satisfactory result and contributes additional evidence that suggests an algorithmical method can be employed to quickly produce EWSs for employment in particular types of medical purpose.

Declaration of authorship

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

.....

TESSY BADRIYAH

September 2013

Table of Contents

List of Tables

List of Figures

Abbreviation

AI:	Artificial Intelligence
AWTTS:	Aggregate Weighted Scoring Systems
BP_SYS:	Blood pressure systolic
BP_DIA:	Blood pressure diastolic
CA:	cardiac arrest
CART:	Classification Regression Trees
CHAID:	Chi-Square Automatic Interaction Detector
CONC_LEVEL:	Conscious Level
DM:	Data Mining
DT:	Decision Trees
EWS:	Early Warning Score
HR:	Heart Rate
ITU:	unanticipated ICU admission
KNN:	K-Nearest Neighbours
LR:	Logistic Regression
ML:	Machine Learning
NB:	Naïve Bayes
NEWS:	National Early Warning Score
NN:	Neural Networks
RCP:	Royal College of Physicians
SVM:	Support Vector Machines
UK:	United Kingdom
ViEWS:	VitalPAC™ Early Warning Score

Acknowledgement

First of all, all praises and thanks to Allah - God Almighty. It is so much blessing that I can present this thesis. And special thanks must go to my family, my husband, Iwan Syarif, my children Daisy, Defita and Pascal, my Dad and my Mum, Makhmud Mujab and Ilik Mizan, this work dedicated to you, thanks so much for all supports and never ending love for me.

I thank to my first supervisor, Dr. Jim Briggs who gives all the guidance, for his support in which ever form without which I possibly wouldn't have been able to achieve this. I sincerely appreciated it.

I would especially like to thank Dr. Tineke Fitch for her support in all those ways could make me have the opportunity to come and study in the UK.

I thank to my second supervisor Prof. Dave Prytherch for guidance and provision of data used in this study, my third supervisor Dr. Ivan Jordanov for discussion in the early of my study, and also Mrs. Deborah Prytherch, for the great proofreading and editing.

I thank to my PhD committee, Dr. Mohamed Gaber, Dr. Christine Urquhart and Dr. Chris Subbe for their valuable inputs and suggestions on my Thesis during my VIVA on 6th September 2013.

I thank to Prof. Gary Smith for provision of data and publishing paper, Dr. M Mohammed for useful advice and also all member of clinical outcome modelling team for cooperation. I would like to take this opportunity of thanking to all my colleagues in Electronic Engineering Polytechnic Institute of Surabaya (EEPIS) and all staff in the School of Computing, University of Portsmouth, thank you for your support. To all my friends, and all my fellow PhD students in the department, thank you for your support and friendship.

I am indebted to the Indonesian Government for the scholarship for my PhD study for 3.5 years and also School of Computing, University of Portsmouth for giving me a chance to conducting research activities.

Southampton, 23 September 2013

Tessy Badriyah

Dissemination

Journals :

1. Badriyah, T., Briggs, J., Prytherch, D., Mohammed, M. A., Meredith, P. et al. (2013). Decision-tree early waning score (DT-EWS) validates the design characteristics of the National Early Warning Score (NEWS), R*esuscitation*. (submitted)

2. Jarvis, S. W., Kovacs, C., Badriyah, T., Briggs, J., Mohammed, M. A., Meredith, P., et al. (2013). Development and validation of a decision tree early warning score based on routine laboratory test results for the discrimination of hospital mortality in emergency medical admissions. *Resuscitation*. (accepted)

Posters & Talk :

Badriyah, T., Briggs, J., Prytherch (2012). DTEWS: Developing EWS using Decision Trees, 2012, University of Portsmouth.
Badriyah, T., Briggs, J., Prytherch (2010, 2011). Comparison of Modelling Technique to Predict Clinical Outcomes using Routinely Collected Data, 2010, 2011, University of Portsmouth.

Chapter 1 Introduction

1.1 Background to the research

When people get seriously ill, they go to the hospital and get medical care. Hospital staff need to identify patients who are at high risk and respond to it appropriately. Unsafe hospital care will increase a patient's risk of death. A serious adverse event (SAE) was characterized by NICE (National Institute for Health and Clinical Excellence, 2007) as an unpleasant event that can prove fatal and can cause incapacity or disability. SAE can also extend the duration of patients admission in a hospital.

Knowing how to identify the 'sick' hospital patient at the earliest opportunity would be useful to identify patients at high or low risk of death or other serious adverse event. The hospital can then respond appropriately to this – perhaps through the nursing or medical staff directly responsible for the patient, or by means of some specialist facility such as a high-dependency or intensive care unit. The response typically provides action or advice on additional care required to prevent deterioration and, therefore, avoid an adverse outcome or other morbidities (Duckitt et al., 2007).

In this thesis, we discuss how predicting the risk of mortality or other adverse outcome can provide advantages where the information could be used by clinical staff and/or hospital management to implement more individualized treatment strategies. By investigating and developing models to predict risk of adverse outcome, and also by comparing methods using different tools and testing them on the real benchmarking dataset from hospital gives us knowledge of what is the appropriate way for predicting adverse clinical outcome.

Previous studies have developed a model to predict the risk of mortality by using routinely collected data that are available in the first few hours following admission to hospital. For the evaluation of in-hospital mortality, a study conducted by Silke, Kellett, Rooney, Bennett, & O'Riordan (2010) formulated a system of scoring which consists of the basic clinical and variable of laboratory which are present when a patient is admitted in the hospital. This will facilitate validation of the system in an independent sample. Their study was based on the assumption that, for acutely ill patients, the most important period is the initial few hours.

An assumption that one can model the risk of in-hospital mortality among general medical patients through administrative data and laboratory items were tested by a study conducted by Prytherch together with his co-workers (Prytherch et al., 2005). These items are available soon after the patient is admitted in the hospital. They used the Biochemistry and Haematology Outcome Model (BHOM), based on data that are available routinely from hospital pathology and administrative computer systems.

For the management of the ill patients, hospital is supposed to be the most appropriate place since it provides the supportive environment for effective treatment. The concept of the patient at risk when their condition deteriorates, requiring critical care, means that the first most important thing is how to recognize the patient's condition. What value of each physiological variable can be categorized as abnormal, and how can we use that to provide an overall picture of the patient's condition? To recognize when a patient starts to deteriorate, clinical staff need to recognize which patient will deteriorate so that they can provide additional care. This is normally done by monitoring a standard set of vital signs. In many hospitals, an early warning score (EWS) system is used to convert the vital signs into a decision as to whether an action (e.g. call a doctor) should be triggered due to the patient's condition.

The very first early warning score system was formulated by Morgan and his co-workers in 1997. It was developed through use of aggregate use of weighted scoring of vital signs to warn the physicians about the deteriorating condition of the patient. Since then, several modifications have been made in the system (Gao et al., 2007). The system developed was simple enough to be used in the wards, using the observations recorded by nursing staff routinely (Morgan & Wright, 2007). Review paper carried out by Smith, Prytherch, Schmidt, & Featherstone (2008) on the use of 33 unique aggregates weighted scoring of vital signs. They found that there were only 12 out of the 33 unique systems (36%) discriminated reasonably well between survivors and non-survivors. Further, Prytherch, Smith, Schmidt, & Featherstone (2010) developed a new system called as ViEWS and compared the performance with 33 other previously presented system that are referenced in (Smith, Prytherch, Schmidt, & Featherstone, 2008). They found that ViEWS performed better than 33 others unique systems.

An EWS typically assigns a small integer score (e.g. 0, 1, 2 or 3) to a given physiological variable. 0 is assigned to values in the normal range and 1, 2 or 3 are given as the variable becomes more abnormal. The EWS is the total of the individual scores. The EWS is then used to determine what, if any, further action is required, following a pre-determined "escalation protocol". The EWS is primarily intended as an aid for more junior, less experienced members of staff. The choice of threshold at which action should be triggered, and the choice of action, is very important. Too low a threshold could mean that the response is swamped by lots of low-level cases. Too high a threshold could mean that deteriorating patients are detected too late to do anything for them. During the initial stages of physiological weakening, simple response is required. If the response is delayed the treatment required can be significantly more complex and need intensive resource.

According to the research by Goldhill, McNarry, Mandersloot, & McGinley (2005), it was found that there is an association between higher number of

physiological abnormalities with greater hospital mortality. Mortality rate in patients with no abnormality was found to be 0.4%, whereas, it was 51.9% in patients with five or more physical abnormalities.

National Institute for Health and Clinical Excellence (2007) reported there was a relation between physiological abnormalities and higher hospital mortality. There is also a need to categorise patients by early risk assessment. They also recommended that six variables: heart rate, respiratory rate,systolic blood pressure, Conscious Level, oxygen saturation and temperature, should be used by track and trigger system as the system's warning about the condition of patients who needed additional care.

Things that we have been discussed at the beginning and the end of the previous paragraph brings us to the conclusion that there are two facts that are needed in this case: there is a need to categorize patients by early risk assessment and there is a need to know when the patient's condition began to deteriorate. It is closely related to delivering better care to patients as the main purpose that we want to achieve in this thesis, and this can be done by using routinely collected data that are available from hospital computer systems. To achieve that goal, then there are two points that we want to accomplish. The first point is how to predict the risk of mortality of patient by categorized it in risk assessment. The result can be used by hospital, especially to determine which patients are in high-risk and thus require additional care. The second point is how to develop early warning score system that can facilitate the hospital to detect the situation when the patient's condition needs more serious treatment due to deterioration.

1.2 The aim of the study

The primary aim of this study was to investigate modelling techniques to predict risk of adverse clinical outcome. Part of this was to develop a structured methodology to generate an early warning score model. Our approach is based on using routinely collected data that is available in the first few hours of a patient hospital episode.

This was achieved by addressing the following objectives:

1. Designing a system to predict clinical outcome by using data mining techniques that can be applied to the problem area.
2. Investigating and implementing a comparison study to predict risk of mortality and testing the candidates on the Biochemistry and Haematology Outcome Models (BHOM) dataset.
3. Assessing the performance of the Risk of Mortality models developed using discrimination and calibration. Validating the model with work by Prytherch, Sirl, et al. (2005) on BHOM dataset in objective 2.
4. Developing a new structured methodology to generate an early warning score algorithmically based on decision trees, assessing the performance of the model, and validating the model with the previous study by (Prytherch, et al., 2010).
5. Validating the Early Warning Score model generated in objective 4 with other adverse clinical outcomes, including cardiac arrest and unanticipated ICU admission that has been done by Smith, Prytherch, Meredith, Schmidt, & Featherstone (2013)
6. Different from score based on clinical judgment, we compare our new structured method in objective 4 with another system based on statistics by Tarassenko et al. (2011).
7. We are of the opinion that our objective number 4 can be applied to another kind of dataset for particular clinical situations. For that

purpose, we apply our new structure method to generate early warning score on BHOM dataset which was used in objective 2.

1.3 The rationale for this study

Risk of adverse clinical outcome can be modelled using routinely collected data, and the most important period is the initial few hours after admission (Prytherch, Briggs, Weaver, Schmidt, & Smith, 2005; Prytherch, Sirl, et al., 2005; Silke, et al., 2010)

The rationale for doing this is that it raises the possibility of categorizing patients based on an assessment of their risk of some outcome and responding appropriately. Mortality (i.e. death) is the most extreme outcome and therefore one that bears particular study, but the techniques are similarly applicable to other outcomes.

Naturally, there are different models to predict risk of mortality based on different data sources, and it is depend on the availability of the data in the hospital. Whether more complete data will improve the predictions of death is still questionable. The research conducted by Pine, Jones, & Lou (1998) investigated the effectiveness of the different models in predicting mortality differing by source of data and by medical condition. They showed that models based exclusively on administrative data didn't predict death as well as did models that were based on clinical factors. Adding laboratory values to administrative data improved predictions of death. However, the selection of the data that can be used depends on the availability of existing data in the hospital administrative computer systems and clinical judgment after analysing the results of the model.

There is a variety of different statistical and machine learning techniques have been used in the literature. Among them, logistic regression is the most

popular of the modelling techniques that have been developed over the last few decades to provide risk stratification. Several studies of these models have shown good external validation with respect to both calibration and discrimination (Pine, et al., 1998; Prytherch, Sirl, et al., 2005; Prytherch et al., 2003; Tang et al., 2007). Logistic Regression (LR) is the current "standard" technique for predicting risk of mortality and when looking at alternative methods, they compare their performance with LR. The two references in the next paragraph below support this assertion.

Asiimwe et al. (2011) used Classification and Regression Tree (CART) methods to analyse routinely collected laboratory data to identify prognostic factors for inpatient mortality with Acute Chronic Obstructive Pulmonary Disease (ACOPD). He showed that CART could be considered as an alternative technique to logistic regression and produced effective models. Another paper (Verplancke et al., 2008) compared logistic regression with support vector machines (SVM), and came to the conclusion that both the LR and SVM models were good. They compared the accuracy of predicting hospital mortality in patients admitted to an intensive care unit (ICU) with haematological malignancies. They concluded that the LR and SVM models were equally effective.

Both methods: CART and SVM are existing methods in machine learning. Apart from those methods, there are still a lot of methods that could be used in machine learning to predict adverse clinical outcome. In this thesis, we will focus on decision trees as one of the machine learning method to develop risk of mortality and early warning score model. The reason to choose decision trees is due to the logic of the modelling results. When people need to make a decision, they then compose a number of rules to solve the problem. In this thesis, we will investigate decision trees as a base method to predict risk of mortality and early warning score model.

From a clinical perspective, early risk assessment would be very useful to facilitate clinician decision making, in particular identifying patients at high or low risk. Especially for those high risk patients, it can allow them to receive more individualized treatment, for example: care in the emergency unit. However, Goldhill, White, & Sumner (1999) found that in an ICU, patients of the wards showed more deaths in contrast to the other individuals who were admitted in the operating/recovery, emergency and accident department. This means that there is a need to recognise the state where the patient who was not categorized as high risk in the beginning, can suddenly need more serious treatment due to deterioration. This was the reason why Morgan, as a founder of early warning score (EWS), further developed this as a "track and trigger system" (TT). This was meant to track physiological variables and then raise a 'trigger' for those patients who deteriorate and need further treatment.

Morgan emphasized that EWS was designed solely to create a safe environment, to ensure that skilled clinicians could be called in a good time to help patients who exhibit signs of physiological deterioration (Morgan & Wright, 2007). Therefore in the original EWS developed in 1997 was not initially designed to predict an outcome. Even so, in the end, for the prompt knowing of potential or established critical illness, the use of track and trigger systems (TTs) was proved as a tool to identify patients in high risk. Since then, most TTs use scores based on the judgement and experience of a clinician (either singly or collectively in a committee or working party). (Gao, et al., 2007) identified that majority of the scoring systems developed so far are based on local modifications of either the original Early Warning Score (EWS) which was discovered by Morgan (Morgan & Wright, 2007), or a later modification of this (Stenhouse, Coates, Tivey, Allsop, & Parker, 2000). Modification of the original early warning score called modified EWS (MEWS) was investigated by Subbe, Kruger, Rutherford, & Gemmel (2001) to identify patients who have a deteriorating condition. They showed that the

MEWS score associated with increased mortality and may help to identify patients at risk of worsening condition. Another research which is also a modification of the original EWS and also calling itself modified EWS (MEWS) but has a slightly different score conducted by Gardner-Thorpe, (2006). The authors discovered that MEWS was useful to be implemented.

Review paper identified different types of track and trigger systems (TTs) that can be classified as: single-parameter systems, multiple-parameter systems, aggregate weighted scoring systems (AWTTS) and combination systems (Gao, et al., 2007). Detailed explanation about those different TTs can be found in Chapter 2 (Literature Review).

Of the four types of TTs, Cuthbertson & Smith, (2007) report that aggregate weighted scoring systems are the most widely used system in the UK. Prytherch, et al. (2010) gave the definition of AWTTS as systems which allocate points in a weighted manner, based on the derangement of patients' vital signs variables from an arbitrarily agreed 'normal' range. The sum of the allocated points is known as the early warning score (EWS).

Regarding AWTTS as the most widely track and trigger system (TTS) used in UK, Smith, Prytherch, Schmidt, & Featherstone, (2008) reviewed a wide range of unique, but very similar, AWTTS in clinical use and there is no consistency regarding their physiological components, but the majority differ only in minor variations in the weightings for physiological derangement and/or the cut-off points between physiological weighting bands. The performance of most systems tested was poor when used to discriminate between survivors and non-survivors, although from 33 unique systems, there are only 12 systems (36%) that discriminated reasonably well. Their results support the argument that physiology can be used to predict outcome, but that further work is required to improve the AWTTS models.

Not only is further work required to improve AWTTS models, but also to evaluate the effectiveness of the early warning scores. Prytherch, et al. (2010)

provided a new concept in thinking about AWTTSs – the EWS efficiency curve. For each AWTTS, there is a relative measure of the number of "triggers" that would be generated at different values of EWS, and this permits the comparison of the workload generated by different AWTTSs. In the same paper, the authors also develop the VitalPAC™ EWS (ViEWS) by utilising an iterative, realistic 'trial and error' method intentionally being altered to increase its capability to predict internal hospital mortality and ability to discriminate patients at higher risk of mortality within 24 hours of the observation. By using large-scale vital signs data (nearly 200,000 observation sets), performance of ViEWS compared with 33 unique systems as presented in (Smith, Prytherch, Schmidt, & Featherstone, 2008). The performance of ViEWS was better than 33 unique systems in the term of discrimination (c-index) and also the most efficient system when measured by early warning score efficiency curve.

As opposed to most of the previous systems which were based on the opinions of medical experts, (Duckitt, et al., 2007) develop scoring system derived based on multivariate logistic regression analysis. They made a derivation and validation study of 4384 patients in a Medical Assessment Unit, and described important physiological variables related to in-hospital mortality. They then applied them in the new scoring system, which was validated against a different cohort. The new scoring system *Worthing PSS* was obtained from the regression coefficient for each variable. They stated that the new scoring system had reasonable accuracy and was more accurate than most other scoring systems. Higher score is associated with higher mortality and a longer length of stay in hospital.

Different from clinical judgment and structured methodology, (Tarassenko, et al., 2011) developed an early warning score (EWS) system based on the statistical properties of a dataset comprising 64,622 hours' worth of continuous vital-sign data, acquired from 863 acutely ill in-hospital patients using bedside monitors. Normalised histograms and cumulative distribution

functions were plotted for each physiological variable (heart rate, respiration rate, oxygen saturation and systolic blood pressure). Their system, Centile-based alerting system, was constructed as follows: an EWS score of 3 was assigned when a vital sign is lower than 1st centile or greater than 99th centile for that variable (in case of double–sided distribution). When a vital sign is between 1st and 5th centile or between the 95th and 99th centile, then this represents score 2. Score 1 refers to the vital sign between 5th and 10th centile or between the 90th and 95th centile. For the appropriate characterization of vital signs the EWS system based on the above approach was found to be good but, yet it needs to be studied to find if this has brought any positive outcome on patient treatment.

We conclude that there is a need to produce a robust methodology to develop an early warning score model. The use of scores with parameters and cut-off points that are not appropriate is unhelpful, and there is therefore a need for an EWS that derives its thresholds systematically, based on actual data. Prytherch, et. al. (2010) achieved this by brute force trial and error. Until now, scoring systems have been developed using different and various clinical assessments, staff expertise, and personal experience. We now need to produce a structured methodology to generate early warning scores algorithmically, which can then be evaluated by clinical expert knowledge. This should also properly evaluate the scoring system. This will provide the real results for delivering better care to patients.

Chapter 2 Literature Review

This chapter starts from the idea that it might be possible to use one of the methods in machine learning to predict adverse clinical outcome.

This chapter can be divided into three parts:

1. The foundation of data mining or knowledge discovery;
2. The history of predictive modelling of the risk of mortality using routinely collected data; and
3. Recognising and responding to patient deterioration.

The first part will be covered in six sections. Firstly, we discuss the importance of extracting useful knowledge from the data as the foundation of data mining or knowledge discovery. We then go on to describe classification and prediction as the method that has been chosen in this thesis. We show how to generate decision trees as a model and also discuss logistic regression. After we have reviewed some alternative models of decision trees and logistic regression, the next step is to show how to assess the performance of the model. The second part discusses the history of predictive modelling of the risk of mortality using routinely collected data. The third part will then review the literature about how deterioration in the patient's condition is recognised and responded to.

2.1 Extracting useful knowledge from the data

At present, the importance of gathering valuable information and acting in accordance with the gathered data is greatly increasing. In the age of digital information, the problem of data overloads increases. There exist a gap between the capacities for data-organization and data-collection and also the capacity for data analysis. Regrettably, this gap is widening. We need to "mine" the data to extract something useful that could be used as knowledge.

Han, Kamber, & Pei (2006) identify that what motivates this data mining is the present situation in which we are often faced with the fact that the data is rich but the information poor. Consequently, decision makers often make a decision based on their intuition rather than based on information-rich data stored in data repositories. This is because they don't have the tools to extract the knowledge which exists in the large amount of data. Data mining is intended to solve this problem. Using data mining, decision makers have a tool that performs data analysis, covers data patterns and contributes to the strategic solution of any problem.

Various authors described data mining in different terms. Han, et al. (2006) define data mining as extracting valuable information from a large sized data. While Kantardzic (2002) gives a definition for data mining as finding distinct models, gathered values or summaries from a defined data.

In this thesis, we would like to give a formal definition of data mining as the process of extracting valuable information from the data collected at hospital using innovative techniques and computer based procedures.

According to Kantardzic (2002), prediction and description is thought to be the main aim of data mining. Description is based on the discovering patterns

for the description of data that can be interpreted by the humans whereas, prediction makes uses of different variables or fields in a set of data to predict the undefined or future values for the variable of interest. The activities in this thesis will fall into the predictive data mining category, producing a model of risk of mortality and early warning score using a given dataset.

Data mining systems should be able to discover patterns of knowledge based on the premise that data can be useful if it is turned into information, and then data mining proceeds to extract information from large amounts of data to produce knowledge. The resulting knowledge can be used to solve the problem. Han, et al. (2006) provide some functionalities or techniques used in data mining, the most popular of which are correlation, association, classification, prediction, cluster analysis and outlier analysis. The techniques that we used and focused on in this thesis are classification and prediction.

Classification facilitates in finding a model (or function) that helps in defining and separating distinct classes or concepts; it also enables to make the proper use of model in order to find out the unknown class label. The obtained model is helpful for the analysis of the training data (i.e. data objects whose class label is well defined). Classification also plays a key role in predicting the categorical (discrete, unordered), predicting models and their constant-valued functions and labels. This can be used to recognize the numerical data values which are not available or is missing other than the class labels (Han, et al., 2006).

Before commencing an analysis, it is essential to gain an understanding of the data. Therefore in the next sub section we will explain about how the data are defined and collected.

2.1.1 Data definition and collection

The definition of data is a factual type of information, especially information organized for analysis or used to reason or to make decisions. In the sort of application that we are considering, data are collected on a sample from a much larger group called the population. The sample itself is of interest not in its own right, but from what it can tell us about the population. Because of chance, different samples from the population might have slightly different characteristics from the general characteristics of the population and this must be taken into account when using a sample to make inferences about the population (Kirkwood & Sterne, 2003).

In more complex systems, data are stored in a database system (also called Database Management System (DBMS) that consists of a collection of interrelated data. Here, we will not discuss further about the methods of data storage, but however we will highlight that the data are usually stored in the form of tables.

Each table consists of a set of attributes (columns or fields) and usually stores a large set of tuples (records or rows). Each tuple in a table represents an object and is described by a set of attribute values. Because it involves a set of data, in this case, we can call the data a *dataset.*

There are two main types of data:

1. Qualitative or categorical: measurement expressed by natural language description. Categorical data can be nominal or ordinal. Nominal has no ranking or order, for example, colour, gender, etc. Ordinal describes an order or ranking between the items measured, such as high, medium or low for salary rate.

2. Quantitative or numeric: numerical measurement. There are two types of numeric data, discrete and continuous.

   a. Any kind of data which has finite number of possible values is known as discrete (for instance student id number can be defined as a number which cannot be added or subtracted).

   b. Continuous data is the one which can have any value (e.g. height, length, mass).

In Table 2.1, we try to make an example of a dataset, it was as adapted from data mining course materials collaborating with our colleagues (Basuki, Badriyah, & Ridho, 2009) in Politeknik Elektronika Negeri Surabaya (PENS), Indonesia. From the dataset, we want to determine whether a person has a risk of hypertension (or not) based on age, weight, and gender attributes.

Table 2.1 Dataset Hypertension

Name	Age	Weight	Gender	Hypertension
Andy	young	overweight	male	yes
Eddy	young	underweight	male	no
Annie	young	average	female	no
Boby	old	overweight	male	no
Harley	old	overweight	male	yes
Dody	young	underweight	male	no
Ruth	old	overweight	female	yes
George	old	average	male	no

Attribute identity — Independent attributes — Target attribute / class / outcome

In Table 2.1, each record is characterized by an *identity attribute,* such as the name of the patient, followed by a fixed number of measurements, or attributes, along with a *target attribute*, denoting its *class* or in another term its *outcome*. Attributes that are not target attributes (class) are called *independent attributes* as their values are not dependent on other values.

There are some term we use from both data mining and statistics. When discussing machine learning techniques in data mining, we use such terms as *target attribute* or *class.* When discussing regression in statistics, we use the term *outcome* variable.

From the table as the basic representation of data, we would like to discover something useful that can be used to predict target attributes. Hence in the above example, we would like to discover how to predict the risk of hypertension from an unknown dataset based on the known dataset in Table 2.1.

The known dataset is called a training dataset and can be used to produce a model as the result from extracting something useful from the dataset. We used a model to predict target attributes in the unknown dataset. The unknown dataset is called a testing dataset, to test the model. The process to predict target attributes using data testing is called *validation.*

2.1.2 Knowledge discovery from data

We have a table that contains data definitions. We need to produce a model to extract something useful from the dataset. This can be done by using data mining. The term Knowledge Discovery from Data or KDD and data mining are used interchangeably by many people. (Han, et al., 2006) identified data mining as a process of knowledge discovery consisting of the following steps:

1. cleaning and integration of data
2. data selection and transformation
3. learning data to discover knowledge/pattern
4. evaluation and presentation of knowledge/pattern
5. knowledge needed

With those 5 steps, machine learning techniques take the role to discover knowledge hidden in the data in step 3.

Before we discuss further about machine learning, we need to differentiate between data mining, machine learning, artificial intelligence and statistics. Sometimes there is considerable overlap in these terms. We would say that they are all related, but they are all different things. We would like to briefly define each of these terms:

- Statistics is a discipline mainly based in mathematical methods, which can be used for the same purpose as data mining, especially in classifying or grouping things. It is also concerned with probabilistic models, specifically inference on models using data, and makes some assumptions about data properties, such as distribution of data.

- Data mining consists of some steps in building models in order to detect the patterns that allow us to classify or predict something from a given dataset. Data mining is applied machine learning that can help to understand the important things that were previously unknown in the data.

- Machine learning is the task of finding knowledge and storing it in some form that can be mathematical models, algorithms, tree, or anything that can help to present knowledge.

- Artificial intelligence is the branch of computer science concerned with making computers behave like humans or to emulate how the brain works. Artificial Intelligence encompasses other areas apart from machine learning, including natural language processing, planning, robotics. We see machine learning as a part of Artificial Intelligence.

Kantardzic (2002) identifies machine learning as a combination of artificial intelligence and statistics, spawning a number of different problems and algorithms for their solution. These algorithms vary in their goals, in the

available training datasets, and in the learning strategies and representation of data.

Inductive machine learning is considered to be the basic machine-learning task in which a set of samples can be helpful in making generalization. It is designed by utilizing different methods and models. They can be further classified into supervised learning and unsupervised learning. In order to calculate the unknown dependence from a known input-output, sample supervised learning can be used. This kind of inductive learning is helpful in regression and classification. The term "supervised" denotes that the output values for training samples are known (i.e., provided by a "teacher") (Kantardzic, 2002).

Within the learning scheme which is not supervised, only the samples which had the input values are defined to a learning system. In the process of unsupervised learning there is no notion of the output. It eliminates the class or target attribute in the dataset and requires that the “learner” forms and evaluates the model on its own.

There are many data-mining methodologies and corresponding computer-based tools is available. One of the methodologies is decision trees. Typical techniques in *decision trees* include the ID3 algorithm and C4.5 algorithm.

2.2 Classification and Prediction

As mentioned in section 2.1., this thesis focuses on the predictive data mining category, producing a model of risk of mortality and early warning score using a given dataset.

As a part of predictive data mining category, classification and prediction techniques are related to produce a model. In this thesis, we will focus on

decision trees as one of the machine learning method to develop a model. The reason to choose decision trees is due to the logic of the modelling results. When people need to make a decision, they then compose a number of rules to solve the problem. In this thesis, we will investigate decision trees as a base method to predict risk of mortality and early warning score model.

To clarify the definition of the classification and prediction, in the first part of this section will discuss distinguish between both techniques. And then went on to explain basic techniques for data classification, such as how to build decision tree classifiers.

The two kinds of data analysis namely classification and prediction are useful tool in extracting models which describes important data classes and helps in determining the unknown values while checking the outcome of data. The difference between classification and prediction can be shown by the following two examples.

An example of classification is: suppose a medical researcher wants to analyse the factors that influence the risk of breast cancer. In order to predict whether the person is at risk the outcome attributes will be, have a risk (yes) or don't have a risk (no). Model or classifier is formulated for the prediction of categorical labels in task of classification. This model is a classifier. The special class of classification which the target attribute has only two possible values (e.g. yes or no, true or false) is called binary classification.

An example of prediction is: suppose that the medical researcher wants to predict the *level* of risk of breast cancer. Level of risk can be expressed in numerical numbers ranging from 0 to 1. Risk 0 indicates that someone does not have a risk of cancer at all, while risk 1 indicates that a person has a 100% risk of cancer. This sort of data explains the numerical prediction in which a

designed model predicts a continuous-valued function or ordered value as opposite to a categorical label.

2.2.1 Classification by Decision Trees Induction

Decision trees are one of the most well-established classification methods. A decision tree can be described as a kind of tree structure in which every node (non-leaf node) determines a test; each branch corresponds to a value of the test, and each leaf node (or terminal node) which has a class label.

Figure 2.1 gives an example of a decision tree built from a hypertension dataset (Basuki, et al., 2009). The tree begins with what is termed as root node, considered to be the "parent" of every other node. Decision tree can play a role in classification by routing from the root node till it arrives at the leaf of the node.

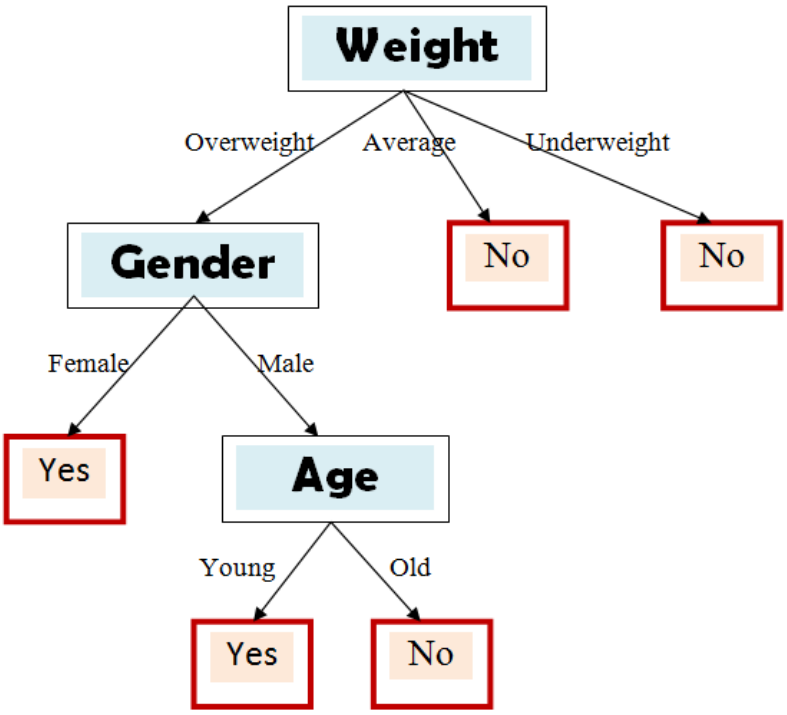


Figure 2.1 Decision Tree for hypertension dataset

Decision trees are built using an attribute selection measure to put attribute on the tree node. An attribute selection measure is a way of selecting the splitting criterion that "best" separates a given data partition that discriminates different classes (e.g. class1='yes', class2='no'). Ideally, all of the records that fall into a given partition would belong to the same class.

A very well-known decision tree classifier, called the ID3 algorithm, was invented by ***Ross Quinlan*** in 1979 (Quinlan, 1993). It uses information gain as its attribute selection measure. Attribute selection measure means the mechanism to select attributes that will be placed as a node in the trees. The measurement of information gain based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages (Shannon, 1948). This is measured in *bits* – the number of binary digits that would be required to store the information in its purest form.

All the formulas that are used to generate decision trees using ID3 algorithm has been taken from data mining book by Han, et al. (2006). There are various kind of decision trees depend on attribute selection used to select an attribute that will placed in a node in decision trees. ID3 algorithm uses information gain as attribute selection that describe in formulas in Equation 2.1-Equation 2.3.

In ID3 algorithm, the expected information needed to classify a record in dataset *D* is given by:

Equation 2.1 :

$$Info(D) = -\sum_{i=1}^{m} p_i \, log_2(p_i)$$

Where:

- $Info(D)$ is the original information requirement, which here means the expected information needed to classify a record in dataset *D* (based on just the proportion of classes)
- *D* is the hypertension dataset
- p_i is the proportion of classes in the dataset (there are two classes in the outcome variable in Table 2.1: "yes" and "no")

To get an exact classification, we need to know how much further information do we still need after the partitioning by measured:

Equation 2.2 :

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} x Info(D_j)$$

Where:

- $Info_A(D)$ is the expected information needed to classify a record in *D* if the records are partitioned according to *A* (obtained after partitioning on *A*)
- *A* is the selected attribute
- *D* is the dataset
- D_j is the dataset partitioned according to *A*.

In the following equation, the term information gain is defined as a difference between the original information requirement (based on the proportion of classes) and new requirement (gained when A is partitioned).

Equation 2.3 :

$$Gain(A) = Info(D) - Info_A(D)$$

Example 2.1 :

To get a clear picture of the formation of the decision tree, we will give a complete illustration from hypertension dataset in Table 2.2 using the ID3 algorithm as all the formula has been described previously. FromTable 2.2, the outcome is hypertension variable. It has 2 classes: "yes" means a person has a risk of hypertension and "no" means a person doesn't have a risk of hypertension. Data was taken with 8 samples, denoted as ***D***. There are two steps used to convert data into a tree: to determine the selected node, and to develop a tree.

Table 2.2 Hypertension dataset

Name	Age	Weight	Gender	Hypertension
Andy	young	overweight	male	yes
Eddy	young	underweight	male	no
Annie	young	average	female	no
Boby	old	overweight	male	no
Harley	old	overweight	male	yes
Dody	young	underweight	male	no
Ruth	old	overweight	female	yes
George	old	average	male	no

The proportion of datasets that predict hypertension as the outcome is 3/8. The proportion that predicts no hypertension is correspondingly 5/8. We first use Equation 2.1 to compute the expected information needed to classify a record in ***D***:

$$\text{Info(D)} = -\frac{3}{8}log_2\left(\frac{3}{8}\right) - \frac{5}{8}log_2\left(\frac{5}{8}\right) = 0.53 + 0.42 = 0.95$$

Using Equation 2.2, the expected information needed to classify a record in ***D*** if the records are partitioned according to *age* (4/8 old and 4/8 young) is:

$$Info_{age}(D) = \frac{4}{8} * \left(-\frac{1}{4}log_2\frac{1}{4} - \frac{3}{4}log_2\frac{3}{4}\right) + \frac{4}{8} * \left(-\frac{2}{4}log_2\frac{2}{4} - \frac{2}{4}log_2\frac{2}{4}\right) = 0.90$$

From Equation 2.3, hence, the gain in information from such a partitioning would be:

$$Gain(age) = Info(D) - Info_{age}(D) = 0.95 - 0.90 = 0.05$$

Similarly, we can compute *Gain*(*weight*) = 0.5 bits, and *Gain*(*gender*) = 0.04 bits. Because *weight* has the highest information gain among the attributes, it is selected as the splitting attribute.

Node *N* is labelled with *weight*, and branches are grown for each of the attribute's values.

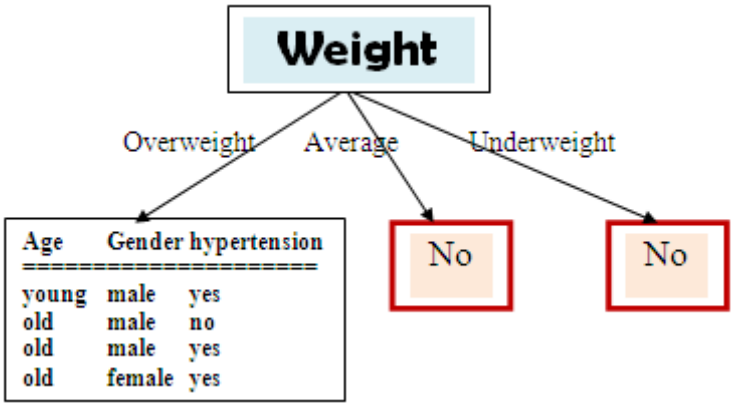


Figure 2.2 Split at the root node of the decision tree

The data in any branch has a homogenous value if the branch has the same value/class for the target attribute. As we can see in the above picture, when *Weight* = *average*, all the data have hypertension = "No", and also when *Weight* = *underweight*, all data values have hypertension = "No" as well. Therefore, these parts of the tree can be represented by leaf nodes. Leaf nodes are nodes with no branches to lower nodes. Consequently, these leaf nodes cannot continue to process the next attribute in a lower branch.

In the next step, having divided the tree by the attribute Weight, we will focus on the branch where *Weight* = *overweight:*

Name	Age	Gender	Hypertension
Andy	young	male	yes
Boby	old	male	no
Harley	old	male	yes
Ruth	old	female	yes

The expected information needed to classify a record in the table when *Weight=overweight:*

Info(D) = $-\frac{3}{4}log_2\left(\frac{3}{4}\right)-\frac{1}{4}log_2\left(\frac{1}{4}\right)$ = 0.31 +0.50 = 0.81

Using Equation 2.2, the expected information needed to classify a record in *D* if the records are partitioned according to age attribute is

$Info_{age}(D) = \frac{1}{4} * \left(-\frac{1}{1}log_2\frac{1}{1}-0\right)+\frac{3}{4} * (-\frac{2}{3}log_2\frac{2}{3}-\frac{1}{3}log_2\frac{1}{3})$ = 0.043

The gain ratio for age attribute would be :

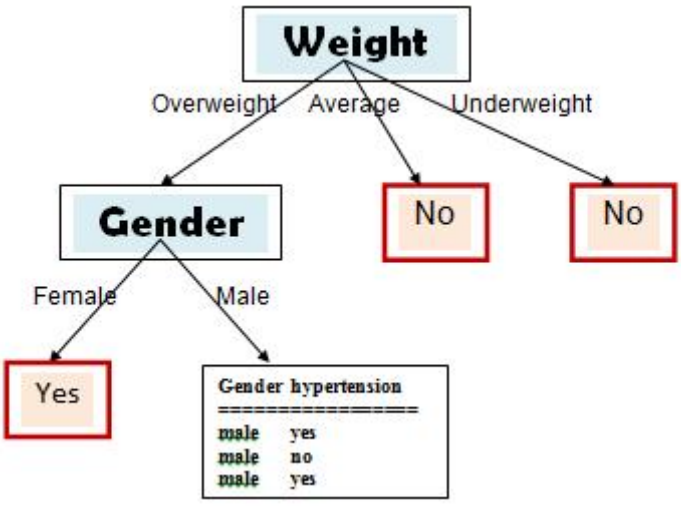
$$Gain(age) = Info(D) - Info_{age}(D) = 0.81 - 0.043 = 0.767$$

Similarly, the expected information needed to classify a record in *D* if the records are partitioned according to *gender* is :

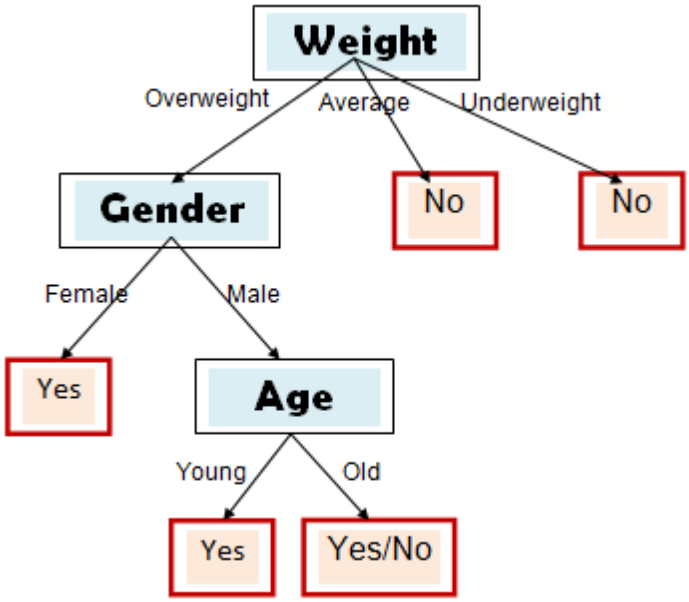
$Info_{gender}(D) = \frac{3}{4} * \left(-\frac{2}{3}log_2\frac{2}{3}-\frac{1}{3}log_2\frac{1}{3}\right)+\frac{1}{4} * \left(-\frac{1}{1}log_2\frac{1}{1}-0\right)$ = 0.043

Attribute *gender* has the same gain ratio as attribute age = 0.767

Therefore, there is no way to determine the next branches except by using expert knowledge or random selection. If we choose *Gender* to be the next attribute, the tree can be developed as follows:



From the leaf that contains mixed values (yes and no), we can continue the calculation of entropy. Fortunately, age is the last attribute left and we can directly choose the age attribute without calculating the entropy value. The next tree obtained is as follows:



As we can see above, if age=old, there is still a mix of (Yes) and (No) as there is one record with a Yes value and one with a No value. However, we must choose one value; there is no way except using expert knowledge (if available) or using random selection. In Figure 2.1, we choose 'No' value if age=old.

These illustrations of generating decision trees from the dataset are intended to provide a clear picture about the processes that exist in the decision trees. Besides many other attribute selection measures that have been proposed, we use two methods of attribute selection measures: Chi-Square Automatic Interaction Detector (CHAID) in SPSS and the Classification Regression Trees (CART) in MATLAB. Both methods are used in chapter 3 to develop risk of mortality and in chapter 4 to develop early warning score.

CHAID method was designed in South Africa by (Kass, 1980). This was a decision tree which used a measure based on the statistical chi-test χ^2).

Breiman et al., (1984) developed a CART algorithm for obtaining the binary decision trees in which every node has two branches. It uses an attribute selection measure called the *Gini Index*.

The Gini index measures the impurity of *D*, a data partition or set of training records, as:

Equation 2.4:

$$Gini(D) = 1 - \sum_{i=1}^{m} {p_i}^2$$

Where p_i is the probability that a record in *D* belongs to class *Ci* and is estimated by $|C_{i,D}|/|D|$. The sum is computed over *m* classes.

The Gini index considers a binary split for each attribute. Let's first consider the case where *A* is a discrete-valued attribute having *v* distinct values, [a_1, a_2, ... ,a_v] occurring in *D*.

In case of binary split, there was a computation of all impurity of every division. For instance, if a binary split on A divides D into D1 and D1, the Gini index of D says that the given partition is in equation 2.5.

Equation 2.5:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

By the binary split on a discrete-or continuous valued attribute A can lead to lowering in impurity:

Equation 2.6:

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

2.2.2 Transformation of a decision tree into decision rules

Decision Trees provide a representation that is intuitive and easily understandable by humans. But to actually generate the output of the decision trees model, in a form that is suitable for human application or implementation in a computer program, it needs to be transformed into decision rules. This is particularly important for with trees of many nodes, which quickly become too complex to be easily read.

To transform a decision tree into decision rules, each leaf or final node in the decision tree needs to be transferred into an IF-THEN production rule. Therefore, the number of rules will be equal to the number of leaf nodes.

The IF part in the decision rules comprises of all tests on a path and the THEN part is the final classification then the rules in this form are known as decision rules. The samples will be classified for all the collection of decision rules in the same way as leaf nodes in a tree.

Decision rules look much like human decision making, with yes/no questions and "if, then" conditions.

An example of the transformation of a decision tree into a set of decision rules is given in Figure 2.3, from the example derived from Table 2.2, to determine whether a person has the risk of hypertension.

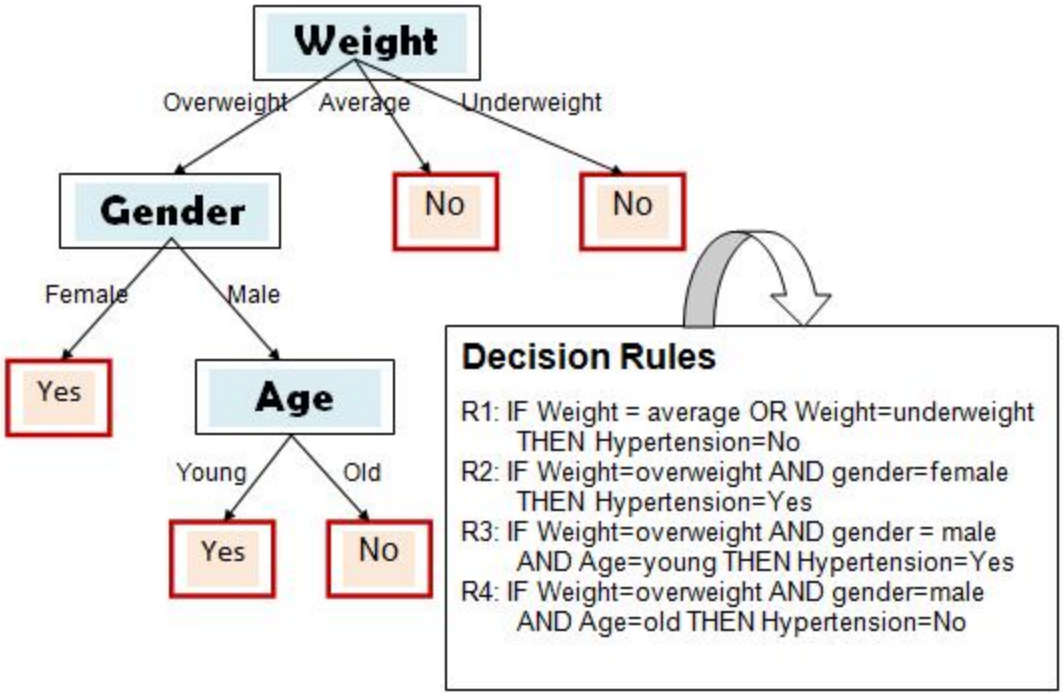


Figure 2.3 Transformation of a decision tree into decision rules

2.2.3 Pruning to overcome the limitation of decision trees

There is simplicity, readability and intuitively in tree based models. Unlike many statistical approaches, they do not require assumptions to be made about distribution of attribute values or independence of attributes. But, there are some serious downsides of the logical approach and data mining which must be taken into consideration by the analysts.

Decision rules production increases the complexity of a rule-based model when the number of rules gets bigger. Therefore, a data-mining analyst has to be very careful in applying decision trees, especially for nonlinear problems (Kantardzic, 2002).

When a decision tree is growing and the number of rules is getting bigger, many branches of the tree will produced, and make the tree more

complicated. We need to cut or remove some branches that are not really required to solve the problem. Such methods to remove the least reliable branches are called *pruning*. The intention behind decision-tree pruning is to remove sections of the tree (subtrees) that make no contribution to solving the problem under investigation. This action produces a tree that is less complex and easier to understand.

(Han, et al., 2006) identifies two common approaches to tree pruning: *prepruning* and *postpruning*.

In the prepruning approach, a tree is "pruned" by halting its construction early (e.g., by deciding not to further split or partition the subset of training records at a given node). The second and more common approach is postpruning, which removes subtrees from a "fully grown" tree. Post-pruning demands more calculation than pre-pruning; however it usually results in a more reliable tree. It is worth nothing that no single pruning method can be considered preferable to all others.

In the postpruning approach, usually the process of pruning is based on estimated error rates. In the prepruning approach, called the recursive-partitioning method, there is a stopping criterion. A stopping criterion means deciding not to divide a set of samples any further under some conditions. The stopping criterion for pruning is normally based on some statistical tests. The χ^2 test is an example of this and is based on the following: If there are not any major differences in classification accuracy pre- and post-division, then represent a current node as a leaf. Therefore, the decision is made *prior* to splitting, leading to the definition of this approach as *pre-pruning*.

We will not here investigate which is the best method of pruning. We are focused on decision trees as one of the techniques in Data Mining, and use it to generate the model by the tools at hand. The process to generate a model is made as efficient as possible without the need to enter any parameter. We also handle the result in the case where the model created by the decision tree

is growing to be complex and produces so many nodes, that a process such as pruning is needed. The focus in this case is how to make the process run easily and efficiently to obtain reliable results. Reliable means that the resulting model can provide good results. The condition of the decision trees growing to be complex is called as overfitted where the model result has too many values that are not always too related with the outcome. According to the complexity of the model result, the decision trees have large number of nodes trees.

2.2.4 Handling continuous attribute values in decision trees

In section 2.1.1, we have illustrated ID3 algorithm using categorical attributes, but how can we compute the information gain of an attribute that has a continuous value? For such a scenario, we must determine the "best" split-point for A, where the split-point is a threshold on A.

First, the A values are sorted in increasing order. Generally, the midpoint of each pair of adjacent values is identified as a potential split-point. Therefore, given v values of A, evaluation of v-1 possible splits is undertaken; e.g. the midpoint between the values a_i and a_{i+1} of A is:

$$\frac{a_i + a_{i+1}}{2}$$

2.2.5 Handling unknown (missing) values in decision trees

ID3 algorithm developed by Quinlan (1993) is based around the assumption that all values for all attributes have been determined. However, in a data set, some attribute values for a number of samples can be absent; this lack of attribute values is common in real-world situations.

There are several reasons why this may happen. The value may not be relevant to a specific sample, or it may not have been recorded when the data were collected. Another possibility is human error when entering data into the database.

To address the issue of missing values, (Kantardzic, 2002) proposes two options:

- Ignore all samples that contain missing data within a database, or
- Create a new algorithm, or alter an existing one, that will function with missing data

Quinlan (1993) developed C4.5 algorithm as the extension of ID3 algorithm that including the arrangement of missing values, continuous attributes, and pruning mechanism (Han &Kamber, 2006).

2.3 Prediction by regression methods

In section 2.2., we discussed about the definition of classification and prediction. Classification using decision trees has been described in section 2.2.1, and in this section, we will describe prediction. Prediction is the task of

predicting continuous (or ordered) values for a given input. For example, we may wish to predict the stratified risk of a disease. We might wish to stratify the risk in five distinct bands, where the lowest risk is represented as number 1 and the highest risk represented as number 5 (or we can set lowest risk=0.2 and the highest risk=1.0).

One method commonly used to perform prediction is regression analysis. We can look at regression analysis is a way to show the relationship between at least one independent attribute or predictor variables, and a dependent or response variable or goal attribute. There are a number of software programs for solving regression problems, such as SAS and SPSS, among others.

2.3.1 Simple linear regression and correlation

In this section, we will discuss the relationship between a numerical outcome (target attribute) and a numerical independent attribute using simple linear regression in which only one independent attribute is considered.

Linear regression can be used to suggest the best-fitting straight line to illustrate the relationship. This particular methodology also gives an estimate of the correlation coefficient, which is a measure of the closeness (strength) of the linear relationship (Kirkwood & Sterne, 2003).

Linear regression with one input variable is the simplest form of regression, and can be expressed as:

$$Y = \alpha + \beta.X$$

Where α and β are regression coefficients.

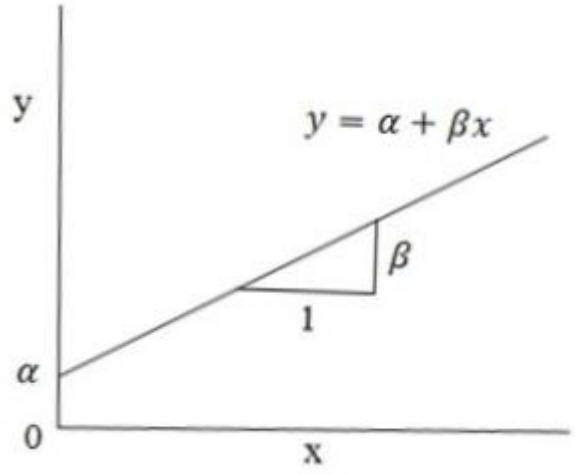


Figure 2.4 : The intercept and slope of the regression equation

The intercept (α) is the point where the line crosses the y axis and gives the value of y for x=0. The slope (β) is the increase in y corresponding to a unit increase in x.

The best fitting line is derived using the least squares by finding the values for the parameter α and β that minimize the sum of squared vertical distances of the points from the line.

The formula for regression coefficients is:

Equation 2.7:

$$\beta\beta = \left[\sum_{i=1}^{n}(x_i - mean_x).(y_i - mean_y)\right] / \left[\sum_{i=1}^{n}(x_i - mean_x)^2\right]$$

$$\alpha = mean_y - \beta.mean_x$$

The quality of the linear-regression model can be estimated. The aim of correlation analysis is to attempt to judge the strength of the relationship between two distinct variables; in this case, the linear regression equation expresses the relationship.

We can quantify the strength of the linear association between a pair of variables via the correlation coefficient r. From a given set of observations

(x_1, y_1), (x_2, y_2), ... (x_n, y_n), we can compute the correlation coefficient by using the formula:

Equation 2.8:

$$r = S_{xy} / \sqrt{(S_{xx}.S_{yy})}$$

Where:

$$S_{xx} = \sum_{i=1}^{n} (x_i - mean_x)^2$$

$$S_{yy} = \sum_{i=1}^{n} (y_i - mean_y)^2$$

$$S_{xy} = \sum_{i=1}^{n} (x_i - mean_x)(y_i - mean_y)$$

The correlation coefficient is always a number between -1 and 1.

2.3.2 Multiple regression

If there are more than one dependent attributes, we must use multiple regression.

The general form of a multiple regression model for the effects of two or more independent variables (X_1, X_2, ..X_n) on an outcome variable (Y) is:

Equation 2.9:

$$Y = \alpha + \beta_1.X_1 + \beta_2.X_2 + \ ... + \beta_n.X_n$$

2.3.3 Logistic regression

Logistic regression is the method most commonly used for the analysis of binary outcome variables (Kirkwood & Sterne, 2003).

Logistic regression is only used if the output variable of the model used is defined as a binary categorical. It is, however, possible for all the inputs to also be quantitative; thus, logistic regression backs up a more generic input dataset. Let us suppose, for example, that there are two possible categorical values for output Y, coded as 0 and 1. In the equation below, p_j is the predicted probability of the event which is coded with 1, and $(1 - p_j)$ is the predicted probability of the other decision which is coded with 0.

Equation 2.10:

$$\log\left(\frac{p_j}{1-p_j}\right) = \alpha + \beta_1.X_{1j} + \beta_2.X_{2j} + \ldots + \beta_n.X_{nj}$$

Where:

α is the intercept

X_{1j} ... X_{nj} are independent attributes in the record-j

β_1 ... β_n are slopes for independent attributes

n is the number of independent attributes

j is the number of records in the dataset

2.3.4 Which regression method do we need to use?

We have described simple and multiple linear regression for the analysis of numerical outcome variables, logistic regression for the analysis of binary outcome variables. In Chapter 3, we will show how all these types of regression modelling can be used to estimate a linear effect on an outcome of a continuous or ordered categorical (ordinal) or non-ordered categorical (nominal) on independent attributes.

We often have a selection of applicable regression models that we can use. This is dependent on how the outcome variable is represented.

For example, outcome death may be expressed as a continuous variable (e.g. a number between 0-1 to represent the probability of the outcome death), or as a binary variable (true or false, 0 or 1, where true and 1 represent death), in which case we would use multiple linear regression or logistic regression respectively.

The consideration is not only related to the outcome, but also the choice of how to represent type of variable of independent attributes. For example, attribute *gender* could be represented as 0 and 1 where 1 means female and 0 means male. Alternatively, it could be represented as a categorical variable as '*female*' and '*male*'. Or even, attribute *gender* could be represented as an ordered categorical variable, as 0 and 1 where 1 has a higher level than 0.

To sum up, in making such choices we need to balance two considerations:

It is advantageous to choose the regression model that uses as much of the information in the data as possible. For example, an outcome which is a continuous attribute such as blood pressure would favour using multiple linear regressions with blood pressure as a continuous variable, since categorizing or dichotomizing it would discard some of the information

collected. A binary outcome such as death allows more choice. We can set the outcome as 0 and 1 and choose to use multiple linear regressions. Or we can set the outcome as TRUE and FALSE or just keep the outcome 0 and 1, and for both we can use logistic regression.

It is often rational to use simpler models before moving on to more complex ones. For example we first use logistic regression before arranging to use ordinal logistic regression to analyse the original outcome variable. We could then check whether the results of the two models are consistent.

The experiment and our analysis of using various types of variables on independent attributes will be discussed in Chapter 3 when predicting risk of mortality using logistic regression.

2.4 Assessing performance of a model

Once we have developed a model, we need to assess the performance of it to assess its feasibility and estimate how accurately it can predict the future or unknown target attribute of the testing dataset.

In general, the performance of a model is measured in two ways: discrimination and calibration. Discrimination is a main measurement refers to how good the performance of the model. Whereas calibration is complimentary to discrimination and refers to how the model agrees with the actual value.

Corresponding to mortality as an outcome, we can give the definition of discrimination as the ability to discriminate between survivor and non-survivor. While the definition of calibration is the degree of correspondence between the estimated probability produced by the model and the actual observed probability in each risk bands. We are conducting an experiment that considering these aspects of measurements (discrimination and calibration) when developing risk of mortality model in Chapter 3.

After we have used different methods to build more than one model, we need to compare the performance of several models. In the following sections, we discuss some of these different measurements to assess the performance of a model, both discrimination and calibration.

As a main measurement to assess how good the model is, we will discuss several methods to assess discrimination. The discrimination measures which will be discussed in the next section are accuracy, sensitivity and the area under ROC curve (c-index). After that, we will discuss about the calibration using chi-test and statistical inference to evaluate the difference between two methods using t-test distribution.

2.4.1 Accuracy (accuration rate)

The most popular method for assessing discrimination is the accuration rate or accuracy. We will revisit the decision tree example from section 2.1.1 as an illustration of how it is used.

Example 2.2:

From the training data in Table 2.2, we can generate a decision tree as in Figure 2.1. In this example, we use that tree to find the predicted outcome for each row of the dataset, and add an additional column *predicted value* into the hypertension dataset as shown in Table 2.3.

Table 2.3 Hypertension dataset, comparing predicted and observed value

Name	Age	Weight	Gender	Hypertension (observed value)	Hypertension (predicted value)
Andy	young	overweight	male	yes	yes
Eddy	young	underweight	male	no	no
Annie	young	average	female	no	no
Boby	old	overweight	male	no	no
Harley	old	overweight	male	yes	no
Dody	young	underweight	male	no	no
Ruth	old	overweight	female	yes	yes
George	old	average	male	no	no

We can measure the effectiveness of a model on a specific test set by looking at the percentage of test set records that have been proved correct by the model. In this example, 7 out 8 records have the observed outcome predicted correctly, and 1 (Harley) has an incorrect prediction.

Equation 2.11:

$$Acc(M) = 1 - Err(M)$$

Where:

- M is a model
- Acc(M) is the accuracy of M
- Err(M) is the error rate or misclassification rate of M.

From Equation 2.11 therefore, we can calculate that the accuracy rate is: 1 – misclassification = 1 – (1/8) = 0.875 or as a percentage: Accuracy = 87.5%.

Using the training data to predict the error rate of a particular model produces a quantity known as the rebsubsitution error. This error estimate provides an optimistic view of the true error rate (likewise, the corresponding

accuracy estimate is also optimistic). This is because the model is only tested against samples it has already seen. Generally we use another dataset (called the testing data) to evaluate the model. The process to estimate the value in the testing data using a model is validation.

If we have standard classification problems, it is possible that there are $m^2 - m$ types of errors, with m being the quantity of classes. In the event of there being just two classes, which would be positive/negative samples, represented by either T and F or by 1 and 0, then it is only possible for there to be two types of error:

- The result is predicted to be T, but observed (actual) outcome is actually F. This is a false negative error. Likewise,
- The result is predicted to be F, but observed (actual) outcome is actually T. This is a false positive error.

A confusion matrix is a useful tool for analysing how well the model can recognize data of different classes. A confusion matrix for two classes is shown in Figure 2.5.

		Predicted Class	
Observed Class/ Actual Class		**P_1**	**P_2**
	O_1	TP (True Positives)	FN (False Negatives
	O_2	FP (False Positives)	TN (True Negatives)

Figure 2.5 A confusion matrix for positive and negative records

In the case of the hypertension dataset, we can assigning O_1 as disease positive, O_2 as disease negative, P_1 as test positive and P_2 as test negative.

In Figure 2.5, false negatives and false positives are such type of errors. If there are N classes, NxN confusion matrix can review types of errors present. For example, if the number of classes m = 3, there are six types of errors ($m^2 -$

m = 3^2-3 = 6), the resulting confusion matrix table is represented in Table 2.4. Every class contains 25 samples in this example, and the total is 75 testing samples.

Table 2.4 Confusion matrix for three classes

	Classification model			Total
True class	**0**	**1**	**2**	
0	20	**2**	**3**	25
1	**3**	22	0	25
2	**1**	**1**	23	25
Total	24	25	26	75

In this example, misclassifications have been highlighted in bold type, and there are (2+3+3+1+1) = 10 misclassifications. Therefore the error rate for this example is:

$$Err = 10/75 \ = 0.13$$

and the corresponding accuracy is

$$Acc = 1 - R = 1 - 0.13 = 0.87 \ (or\ as\ a\ percentage: Accuracy = 87\%)$$

2.4.2 Sensitivity, Specificity, and precision

Accuracy is commonly used as a measurement to assess the performance of the model. It is very clear and easy to understand to assess the model using accuracy.

Despite of the clarity of accuracy, we might be faced with a situation where accuracy may not fully represent the performance of a model. For instance, suppose we have a dataset where the target attribute is either "cancer" or "not cancer." An accuracy rate of, say, 90% may make the classifier seem quite accurate, but what if only very few records in the dataset actually have that outcome? If only 1–2% of the training data are actually "cancer", then clearly an accuracy rate of 90% would probably not be acceptable - the classifier could be good at correctly labelling only the "not cancer" records. This

situation can be described as an *unbalanced dataset*, when the number of records in one class ("cancer") is very few compared to another class ("not cancer).

Instead of using accuracy as the test, we would like to be able to assess how well the classifier can recognize *"cancer"* records (the positive records) and how well it can recognize *"not cancer"* records (the negative records). The measures *sensitivity* and *specificity* can be used, respectively, for this purpose.

Another descriptor for 'sensitivity' is the *true positive (recognition) rate;* that is,the percentage of positive records that are identified accurately. 'Specificity', on the other hand, is the *true negative rate*; that is, the percentage of negative records that are accurately identified.The definition of sensitivity and specificity would be easier using a confusion matrix as shown in Figure 2.5.:

Equation 2.12:

$$sensitivity = \frac{TP}{pos} = \frac{TP}{(TP+FN)}$$

$$specificity = \frac{TN}{neg} = \frac{TN}{(TN+FP)}$$

Where:

- *TP* is the number of true positives (*"cancer"* records that were correctly classified)
- *pos* is the number of positive (*"cancer"*) records, is the sum of true positives (TP) and false negatives (FN)
- *TN* is the number of true negatives (*"not cancer"* records that were correctly classified),
- *neg* is the number of negatives (*"not cancer"*) records, is the sum of true negatives (TN) and false positives (FP)

- *FP* is the number of false positives (*"not cancer"* records that were incorrectly labelled as *"cancer"*)
- *FN* is the number of false negatives (*"cancer"* records that were incorrectly labelled as *"not cancer"*)

The accuracy can be expressed as a function of sensitivity and specificity:

Equation 2.13 :

$$accuracy = sensitivity\frac{pos}{(pos + neg)} + specificity\frac{neg}{(pos + neg)}$$

Furthermore, it is possible to use 'precision' to assess the percentage of records labelled as "*cancer*" that in actual fact are "cancer" records. And we also use other measurements that are related to the confusion matrix, such as positive predicted value (PPV) and negative predicted value (NPV). The following is the formula of precision, PPV and NPV as the following formula:

Equation 2.14 :

$$precision = \frac{TP}{(TP + FP)}$$

$$positive\ predictive\ value\ (PPV) = \frac{TP}{(TP + FP)}$$

$$negative\ predictive\ value\ (NPV) = \frac{TN}{(TN + FN)}$$

The true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are also useful in assessing the costs and benefits (or risks and gains) associated with a classification model. The cost associated with a false negative (such as incorrectly predicting that a patient with high risk is not predicted as having high risk) is often far greater than that of a false positive (incorrectly labelling a patient who has low risk as having high risk). In such cases, we can assign a different cost to each measurement. These costs may take into account the risk to the patient, or other costs incurred by the

hospital. Likewise, the advantages of a true positive decision could differ from those of a true negative decision.

It is also not always reasonable to assume that all records are uniquely classifiable. Rather, it is more probable to assume that each record may belong to more than one class. Therefore, even where the dataset has a binary outcome (for example, 0 and 1 representing alive or dead), the result of the model could be a continuous number between 0 and 1. This value might represent the risk of a particular outcome, and this might better represent what is known about the patient than simply predicting the most likely outcome. We will discuss this further when we implement logistic regression and other machine learning methods to predict risk of mortality in Chapter 3. In these circumstances, accuracy is not an appropriate measure because it does not take into account the possibility of records belonging to more than one class.

2.4.3 Area under ROC Curves

In this thesis, we need an adequate and appropriate measurement to assess the performance of the model, and we consider the area under the receiver operating characteristic (ROC or AUROC) curve is the primary measurement for evaluating our models. It is commonly used in studies of medical decision making.

ROC curves can be considered a helpful visual tool for evaluating and contrasting classification models. ROC stands for Receiver Operating Characteristic, and ROC curves originate from Signal Detection Theory that was developed during the Second World War. In this military context it was used for the analysis of radar images. A ROC curve describes the compromise between the true positive rate or sensitivity (percentage of positive records that have been correctly identified) and the false-positive rate (percentage of

negative records that have been wrongly identified as positive), for a specific model.

The vertical axis of an ROC curve represents *sensitivity*, and the horizontal axis represents *1-specificity*. A ROC curve is plotted based on a cut-off point. The term of area under ROC curve (AUROC) is also called as the c-statistic or c-index (Cook, 2008). The area under the ROC curve, summarised by the c-index, can range from 0.5 (no predictive ability) to 1 (perfect discrimination). Reasonable discrimination is indicated by c-index values of 0.700-0.800, and good discrimination by values exceeding 0.800.

In the following example, we make our own example to illustrate the process to calculate area under ROC curve (AUROC) or c-index.

Example 2.3 :

To illustrate how to calculate the c-index, suppose we have 50 data records, where score is the predicted attribute and outcome is the observed attribute. Score is discrete with a number range from 1 to 5 [1, 2, 3, 4, 5]. Outcome is binary with 0 and 1 as the values.

Table 2.5 Sample dataset to demonstrate area under ROC curve

Record	predicted	observed	Record	predicted	observed	Record	predicted	observed
1	1	0	18	2	0	35	4	1
2	1	0	19	2	0	36	4	1
3	1	0	20	2	0	37	4	1
4	1	0	21	3	0	38	4	1
5	1	0	22	3	0	39	4	1
6	1	0	23	3	0	40	4	1
7	1	0	24	4	0	41	5	1
8	1	0	25	5	0	42	5	1
9	1	0	26	1	1	43	5	1
10	1	0	27	2	1	44	5	1
11	1	0	28	2	1	45	5	1
12	2	0	29	3	1	46	5	1
13	2	0	30	3	1	47	5	1
14	2	0	31	3	1	48	5	1
15	2	0	32	4	1	48	5	1
16	2	0	33	4	1	50	5	1
17	2	0	34	4	1			

From Table 2.5, for each outcome, we can choose a score threshold *N* where the predicted value will be "1" if the score is greater than or equal to *N (N* is a value from the range of the score attribute). We then generate a new table based on Table 2.5 to calculate: FP, FN, TP, FN, sensitivity and (1-specificity) as shown in Table 2.6.

Table 2.6 Set of points in the *Sensitivity* and *1-Specificity* to form ROC curve

Positive if greater than or equal to	TP	TN	FP	FN	Sensitivity	Specificity	1-Specificity
1	25	0	25	0	1.00	0.00	1.00
2	24	11	14	1	0.96	0.44	0.56
3	22	20	5	3	0.88	0.80	0.20
4	19	23	2	6	0.76	0.92	0.08
5	10	24	1	15	0.40	0.96	0.04

Figure 2.6 shows the ROC curve derived from plotting the set of points in Table 2.6.

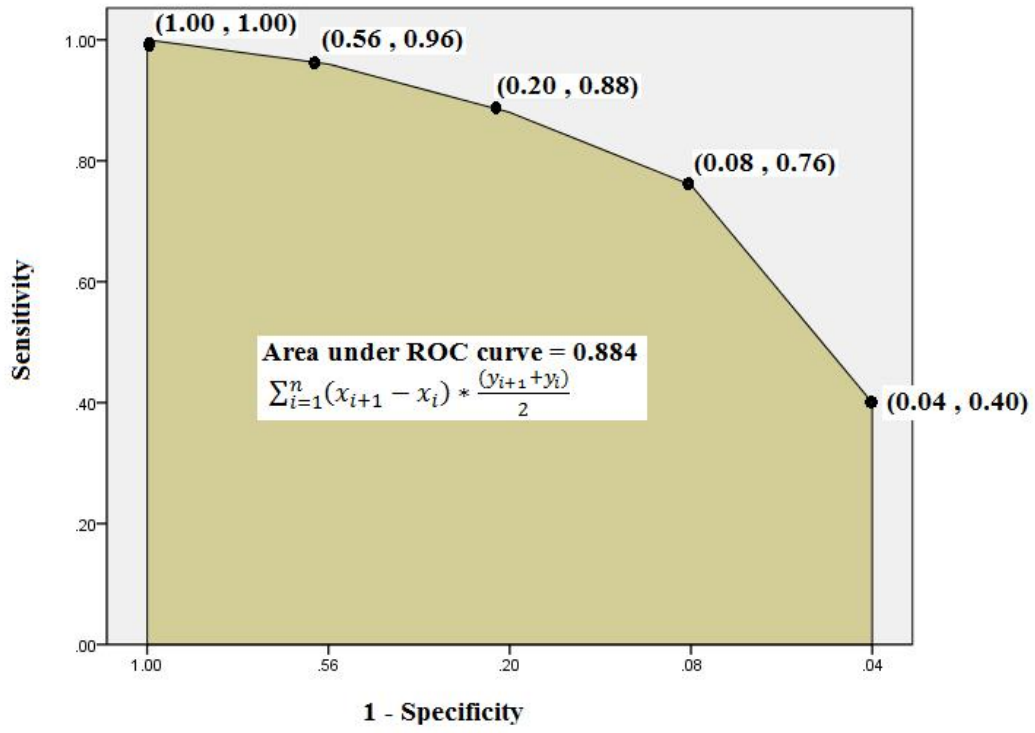
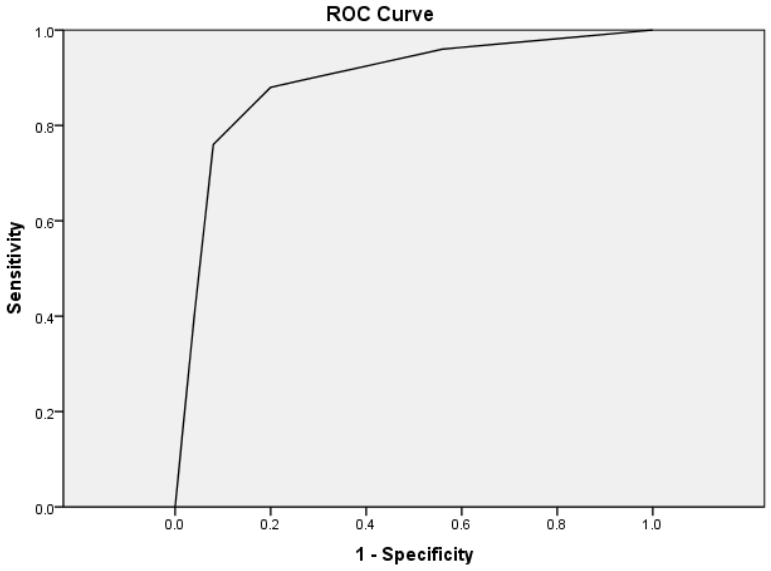


Figure 2.6 ROC Curve from table 2.8

Figure 2.7 shows the area under the ROC curve as derived by SPSS. SPSS uses a different cut-off value. Instead of using the range of values in the score attribute (1,2,3,4,5), SPSS uses the set of cut-off values of (0, 1.5, 2.5, 3.5, 4.5, 6).

Consequently, it gives a similar but slightly different result (0.892 compared with 0.884). The way to determine our cut-off points can be different with SPSS, but the most important thing is the formula in Figure 2.6 to calculate the area has been proven right. If we put the value of cut off points used by SPSS: (0, 1.5, 2.5, 3.5, 4.5, 6) in the formula in Figure 2.6, then the results will be same 0.892.



Coordinates of the Curve

Test Result Variable(s):score

Positive if Greater Than or Equal To[a]	Sensitivity	1 - Specificity
.00	1.000	1.000
1.50	.960	.560
2.50	.880	.200
3.50	.760	.080
4.50	.400	.040
6.00	.000	.000

Area Under the Curve

Test Result Variable (s):score

Area
.892

Figure 2.7 ROC curve, cut-off values and calculation of area under the curve using SPSS

So far, we have discussed discrimination (accuracy, sensitivity, specificity, c-index) to assess the performance of the model. In the next few sections, we will discuss some other point of view to analyse the data using statistical terminology, including estimate p-value and confidence intervals, and calibration using chi-square (λ^2) statistic. These measurements are complementary.

2.4.4 Using p-values and confidence intervals to interpret results

The provision of discrimination using c-index is often accompanied by a confidence interval. And also using calibration involved p-values. In this section, we will discuss confidence interval and p-values.

When we discussed data definition in section 2.1.1, we mentioned that different samples from the population might have slightly different characteristics from those of the population, and this must be taken into account when using a sample to make inferences about the population (Kirkwood & Sterne, 2003). For that reason, we employ *confidence intervals* as a means of assessing those differences. A confidence interval comprises a range of values within which we can be confident to a reasonable extent that our population difference lies.

'Sample' is the term usually used in statistics for what in computer science (and more specifically data mining) is known as the dataset. In statistical analysis, we need to test the hypothesis about the sample/dataset. We will describe the scenario of testing the hypothesis in the following example.

Example 2.4 :

We take the following example from statistic book by Kirkwood & Sterne (2003). If we hypothesise that everyone who lives to the age of 90 or over is a non-smoker, this can be investigated in two ways:

1. Validate the hypothesis by sourcing every person aged 90 or over, and confirming that they are all non-smokers.

2. Invalidate the hypothesis by identifying there is one person is a smoker and he/she is 90 years old or over

In general, it is much easier to find evidence against a hypothesis than to be able to prove that it its correct.

Hence, in this case, the *null hypothesis* is:

There is someone aged 90 or over who is a smoker.

The p-value is related to the evidence against the null hypothesis. The smaller the p-value, the stronger an evidence against the null hypothesis.

In particular, a p-value less than 0.05 is often reported as 'statistically significant'. This is why hypothesis tests have often been called significance tests. A p-value less than 0.05 signals that there is less than a 5% likelihood that the result obtained is wrong and was obtained by chance.

Before a confidence interval is constructed, let's talk about some important aspects of statistics. In statistics, normal distribution is commonly used and important. The normal distribution is relevant because it plays a central role in statistical analysis techniques. Specifically, by means of an appropriate change of units, any normally distributed variable can be matched to the standard normal distribution (SND); where the mean is zero, and the standard deviation is 1.

Equation 2.15:

Standard normal distribution (SND), $= \frac{x-\mu}{\sigma}$,

Where: x is the original variable, with mean μ and standard deviation σ.

Example 2.5 :

We take the following example from statistic book by Kirkwood & Sterne (2003). Study case: The heights of adult men in the UK, which is approximately normal with mean (μ) =171.5 cm and standard deviation (σ) =6.5 cm. The normal distribution can be used to estimate, for example, the proportion of men taller than 180 cm. The corresponding SND is: $z =$

$\frac{180-171.5}{6.5}$=1.31. By using normal distribution that occupies a central role, the approximate normal curve shown in Figure 2.8.

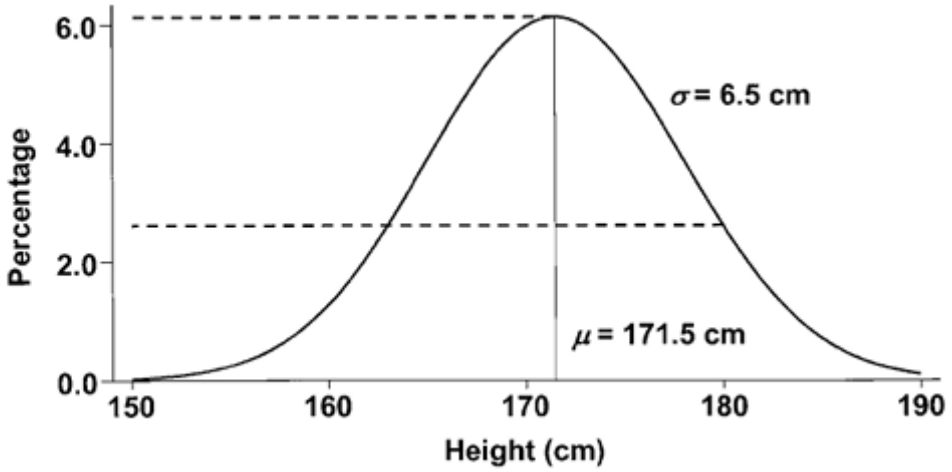


Figure 2.8 The approximate normal curve describing the distribution of height of adult men

In Figure 2.8, the area above 1.31 is given in statistic table is 0.0951 (9.51%). Therefore, we conclude that 9.51% of adult men are taller than 180 cm.

Confidence intervals provide information about statistical significance. This also allows a decision about the clinical relevance of the results.

In all cases the confidence interval is constructed as the sample estimate:

Equation 2.16:

$$95\%\ Confidence\ Interval = estimate - (1.96 * standard\ error)\ to\ estimate + (1.96 * standard\ error)$$

or

$$95\%\ Confidence\ Interval = estimate\ \pm (1.96 * standard\ error)$$

It should be acknowledged that the 95% confidence level is based on the same arbitrary value as the 0.05 threshold: a z value of 1.96 corresponds to a p-

value of 0.05. This means that if P<0.05 then the 95% confidence interval will not contain the null-value. It is also important to appreciate that the size of the p-value depends on the size of the sample.

To conclude, we can confidently state that p-values and confidence intervals both hold substance as statistical concepts, and do not contradict each other. The two statistical concepts are complementary, and both are helpful in interpreting the results of medical research.

2.4.5 Calibration using chi-square statistic

Calibration (or reliability) refers to whether the predicted probabilities agree with observed probabilities. Calibration is most suited to a problem where we would like to predict risk in the future. This is because calibration measures how well the predicted probabilities correctly estimate a future event.

The Hosmer-Lemeshow statistic is an appropriate test and the most popular measure of calibration (Lemeshow & Hosmer, 1982). Individual records in the validation subset are separated in groups defined by risk range. Looking at each individual risk, the predicted number of deaths is compared against the actual number observed. Goodness-of-fit can be analysed using the χ^2 test (chi-test). Due to the fact that this is a null hypotheses test, it must be stated that p values of less than 0.05 indicates that there is evidence of significant lack of fit.

The chi-squared test compares observed (actual) and expected (predicted) frequencies. The form of the test is:

Equation 2.17:

$$\chi^2 = \sum_{i=1}^{n} \frac{(observed(i) - predicted(i))^2}{predicted(i)}$$

$$degrees\ of\ freedom(df.)$$
$$= (number\ of\ groups) - (number\ of\ parameter) - 1$$

The chi-squared distribution depends on the degrees of freedom.

Example 2.6 :

We use our own example of a risk of mortality table here. Suppose we want to calculate the agreement of observed distribution with the predicted values using the chi-squared goodness of fit test. There are 10 levels of risk, risk 1 is the lowest and risk 10 is the highest level. In this case, there are ten frequencies and no parameters have been estimated. In the last row of the table, we calculate the total number of deaths predicted, the total number of deaths reported and the value of χ^2 (chi-test).

Table 2.7 Predicted and observed risks in bands

Risk bands	**Predicted** (P)	**Observed** (O)	χ^2 $\left(\frac{(O-P)^2}{P}\right)$
1	22	16	1.44
2	18	17	0.13
3	21	22	0.09
4	22	27	1.00
5	20	20	0.01
6	30	31	0.04
7	22	22	0.01
8	22	12	5.97
9	18	17	0.09
10	9	7	0.71
All	203	191	9.48

Based on the results shown in Table 2.7, the chi-test (χ^2) =9.48, with degrees of freedom (df.) = 10, therefore p-value = 0.487 which (because it is > 0.05) indicates there is no evidence of significant lack of fit. We can found the p-value in the statistic table or built-in function in the software tools. For example in Excel we can use built in function =CHIDIST(x, degrees of freedom), where x is calculated chi-squared.

2.4.6 Statistical inference to evaluate the differences between two methods

If a method exists has and has proved to be reliable, we may have an alternative method to contrast with it. We would like to find out if variations exist between the two models formed by the two methods. From both methods we have some result, each result come from two using same dataset. We obtained some results from two methods after applying any sampling method (as explained in the next section 2.5). Our purpose is to evaluate between two methods, to determine if there is any "real" difference in the c-index. For that purpose, we need to employ a *test of statistical significance*.

The factors needed to perform the statistical test must be identified and defined. As an example, let us say that for each model we perform a 10-fold cross-validation ten times. For each cross-validation, we would use a different 10-fold partitioning of the gathered data.

Cross-validation is where we choose different partitions (sample sets) from the population to use as the training set for our model, and then apply them using the remaining data as the testing set. Cross-validation can be used to show whether or not the choice of sample has had an unintended effect on the model. If each model results in a test with a similar performance, we can be confident that the choice of sample was not important.

Every individual partitioning is drawn independently. We can take ten of c-index gathered for each method (which we will term M1 and M2), respectively, with the result being the generation of the c-index for every individual model. Looking at a selected model, the specific c-index determined in the cross-validations could be measured as different, unique samples from a probability distribution.

Kirkwood & Sterne (2003) suggest t distribution can be used in situations where there is only a small sample size. The authors have measured the comparison of birth weights of children born to 14 heavy smokers with those of children born to 15 non-smokers. They set the null hypothesis that there was no difference between birth weights of children born to heavy smokers and non-smokers. In their calculation result using t distribution, the p-value of 0.0064 (<0.05) provides strong evidence against the null hypothesis. It indicated that birth weight of children born from heavy smoker is lower than that from non-smokers.

We follow a measurement of comparison using t distribution was performed by Kirkwood & Sterne (2003) with k-1 degrees of freedom where, here, k = sample size = 10. Hypothesis tests (also known as significance tests) and p-values are used to assess the strength of the evidence against the null hypothesis that there is no true association in the population from which the sample was drawn.

We hypothesise that the two individual models are identical; or to put this in another context, the difference in mean c-index between our two selected models is zero. Alternatively, if we are able to reject this hypothesis (i.e. the null hypothesis), then we will be able to successfully argue that the difference between the two models is statistically significant.

Looking now at the t distribution, which calculates the t-statistic with k-1 degrees of freedom for k samples, in our example we have k = 10 since, our c-index obtained from 10-fold cross validations for each specific model.

Example 2.7 :

In this example, we make our own data in Table 2.8. It shows the c-index of two methods, M1 and M2 using 10-fold cross validation to predict risk of mortality. In section 2.5, we will discuss in detail the resampling method using cross-validation.

Table 2.8 : c-index obtained from applying two methods, using 10-fold cross validation

M1 (X_1)	M2 (X_0)
0.748	0.708
0.765	0.859
0.783	0.842
0.663	0.756
0.771	0.788
0.718	0.803
0.759	0.790
0.699	0.731
0.761	0.820
0.785	0.859
$\overline{X_1}$ = 0.7452	$\overline{X_0}$ = 0.7956
S_1 = 0.0397	S_0 = 0.0520
n_1 = 10	n_0 = 10

The calculations needed to derive the confidence interval are: the difference between means ($\overline{x1}$ - $\overline{x0}$) =0.7452-0.7956 = -0.0504

The standard deviation (s) is:

$$s = \sqrt{\frac{10*0.0397^2+10*0.0520^2}{10+10-2}} = 0.046246$$

Standard error of the difference (s.e.):

$$s.e. = 0.046246 * \sqrt{\left(\frac{1}{10}+\frac{1}{10}\right)} = 0.020682$$

Degrees of freedom (d.f.) = 10 + 10 - 2 = 18; t'=2.10

Regarding to 95% confidence interval, the 5% percentage point of the t distribution with 18 degrees of freedom is 2.10. Therefore, the 95% confidence interval for the difference between the mean c-index is :

= -0.0504 - (2.10*0.020682) to -0.0504 - (2.10*0.020682)

= -0.09383 to -0.00697

In small samples we allow for the sampling variation in the standard deviation by using the t distribution. This is called the t-test, and is calculated as:

Equation 2.18:

$$t = \frac{\overline{x_1} - \overline{x_0}}{s.e.} = \frac{\overline{x_1} - \overline{x_0}}{s\sqrt{(\frac{1}{n_1} + \frac{1}{n_0})}} \quad , \quad d.f. = n_1 + n_0 - 2$$

The corresponding p-value is derived in exactly the same way as for the z distribution using this following formula:

Equation 2.19:

$$s = \sqrt{(\frac{(n_1 - 1){s_1}^2 + (n_0 - 1){s_0}^2}{(n_1 + n_0 - 2)})}$$

Example 2.8 :

The calculations for the t-test to compare the c-index value of logistic regression with those of c-index value of decision trees, as shown in Equation 2.18 as follows:

$$t = \frac{(0.7452 - 0.7956)}{0.046246\sqrt{\left(\frac{1}{10} + \frac{1}{10}\right)}} = -2.43$$

As the test is two-sided, the p-value corresponding to minus 2.43 is the same as that corresponding to plus 2.43. The statistic table shows that the p-value corresponding to t=2.4 with 18 degrees of freedom is 0.027 (<0.05). A p-value of 0.027 provides fairly strong evidence against the null hypothesis. These data therefore suggest that M2 generally performs better than M1.

2.5 Re-sampling method

In the earlier section where we discussed data definition, we define training data as the dataset that is used to make a model (section 2.1.1). Using training data to derive a model (classifier) and using the same data to estimate the performance of the model can result in misleading, overoptimistic estimates due to overspecialization of the learning algorithm to the data. Where there are different samples of training data from the same population, the model derived from each could be quite different.

With the various dataset provided to use as training data, which one is the best to generate a sample dataset as training data to produce the best model? This section addresses these questions, how to do that with a re-sampling method. Below is a brief explanation of the various resampling methods. A selection will be required based on the characteristics of the data and the problem in hand:

Resubstitution Method - The easiest method. All available data are used for training and testing. That is, both testing and training sets are the same.

Holdout Method– Half or two thirds, of the data, are used for training purposes with the remaining data used for testing. The training and testing sets are independent from each other, and separate partitioning results in different estimates.

Leave-one-out Method -A model has been designed using (n-1) samples for training, following which it is evaluated against the remaining sample. This process is repeated n times using alternative training sets, always of size (n-1). The computational requirement of this particular method is very high, due to the fact that n different models must be constructed and then compared.

Rotation Method (n-fold cross validation) - This method can be considered as conciliation between holdout and leave-one-out methods. To summarise its

structure, it divides all available samples into P disjoint subsets, where $1 \leq P \leq n$. Subsequently, (P-1) subsets are selected for training, whilst the rest of the subsets are used for testing. In practice, this is the most common method, particularly for problems with a small number of samples.

Bootstrap method - The bootstrap method works by resampling the available data with substitutes to create a number of replication data sets of identical size to the original data set. Ordinarily, several hundred of these new sets will be generated. The newly generated training sets are used to identify what are called 'bootstrap' estimates of the error rate. Test results have identified that the bootstrap estimates perform better than cross-validation estimates. Situations which have limited data sets are ideally suited to this method.

Among the five commonly used re-sampling method, in our opinion, the rotation method or cross validation is the right approach for doing fair comparison. If we take n as 10, therefore we take 10-fold cross validation, the system is trained and tested for 10 iterations. This means there will be a total of 10 different training datasets, each independent from its testing data.

2.6 A brief history of physiological outcome modelling

2.6.1 The history

The history of physiological outcome modelling has been developed since Copeland, Jones, & Walters (1991) first introduced POSSUM (Physiological and Operative Severity Score for the enumeration of Mortality and Morbidity) as a system for standardising patients' data, the outcome being that direct comparisons of patient outcome were able to be drawn in spite of a range of patterns of referral and population.

In POSSUM, Copeland originally assessed 48 physiological factors and 14 operative and postoperative factors for each patient. Sagar et al. (1994) have described the use of POSSUM for comparative audit purposes. Khuri et al. (1997) proposed a model which requires up to 34 separate physiological and operative data items for mortality while Daley et al. (1997) proposed a risk of morbidity model that requires the collection of up to 55 data items.

Prytherch et al, (1998) proposed a modification of POSSUM called P-POSSUM that uses multivariate analysis techniques. P-POSSUM reduces the complexity of the technique by using 12 physiological and six operative factors. The original POSSUM (using logistic regression) had actually over-predicted the risk of death. Whereas, the P-POSSUM equation gave far better results, with a close fit with the observed in hospital mortality rate

Moving from surgery to medicine, Prytherch, Sirl, et al. (2005) have shown the prediction of the patients outcome for general medical patients using standard routinely obtained data. This includes non-surgical instances. This

study allows the future possibility for the treatment and surveillance of patients to be classified by early risk assessment.

The pathology data items studied can be identified as being from the first routinely gathered haematology and biochemistry blood tests. These items are haemoglobin, white cell count, serum levels of urea, albumin, creatinine, sodium and potassium. The administrative data collected were patient age at time of admission, mode of admission (elective or emergency), sex of the patient, and outcome (survival or non-survival) at time of hospital discharge. From this, a model was constructed using a specific training set (Q1). The application of the model to the validation sets gave *c*-indices as follows: 0.779 (Q2), 0.764 (Q3) and 0.757 (Q4), respectively. This suggests a reasonably good level of discrimination. Hosmer-Lemeshow analysis produced results of χ^2 = 9.43 (Q2), χ^2 = 7.39 (Q3) and χ^2 = 8.00 (Q4) (*p*-values of 0.307, 0.495 and 0.433) for 8 degrees of freedom, which would indicate strong calibration.

In another paper, Prytherch et al. (2007) developed Vascular Biochemistry and Haematology Outcome Models (VBHOM), which embraced the idea of using a minimum data set, in order to model outcome. This particular approach was targeted to test this type of model on a group of patients undergoing open elective and non-elective abdominal aortic aneurysm (AAA) repair. This new model, created from recent national vascular database (NVD) data, assumed the approach of using a minimum set of data to create a model for outcomes. It uses only data items that can be obtained before an operation from hospital pathology and patient administration computer systems. These data items are only routinely gathered within usual pathways of clinical care. Thus, the application can be generic and data collection no longer poses a burden for the care providers.

In VBHOM, Prytherch, et al. (2007) used a training sample of 327 patients. Of these, 208 had elective AAA repair, and 119 had emergency repair carried out on a ruptured AAA. The outcome following elective and non-elective AAA

repair could be accurately described by applying the same model. The overall mean predicted risk of death measured at 14.13%, and the number of deaths predicted was 48. In actual terms, the number of deaths was 53 ($\chi 2$ = 8.40, 10 d.f., p-value = 0.590; no evidence of lack of fit). A solid discrimination was also shown by the model (c-index = 0.852).

2.6.2 Logistic regression is the most popular method to predict risk of mortality

All the literature reviewed in the previous section (2.6.1) used logistic regression as the modelling method. Much other previous research showed that logistic regression gives a good result in predicting the risk of mortality.

Other authors used logistic regression with various local datasets, some of them using only clinical datasets, others combining both clinical and administrative data. (Pine, et al., 1998) investigated the effectiveness of the different models in predicting mortality differing by source of data and by medical condition. Administrative models (c-index=0.834) didn't predict death as well as did clinical models (c-index=0.875). Adding laboratory values to administrative data improved predictions of death (c-index=0.860) and improved its average correlation of patient-level predicted values with those of the clinical model from 0.86 to 0.95. However, the selection of the data that can be used depends on the availability of existing data in the hospital administrative computer systems, and clinical judgment used to analyse the results of the model.

2.6.3 Developing risk of mortality using methodology in machine learning

In our literature review, Logistic Regression (LR) was described as the current standard to predict risk of mortality and when looking at an alternative method, the author still compared their performance with LR. Asiimwe, et al. (2011) analysed routinely collected laboratory data to identify prognostic factors for inpatient mortality with Acute Chronic Obstructive Pulmonary Disease (ACOPD) using Classification and Regression Tree (CART) analysis compared with Logistic Regression. Performance of CART was c-index=0.734 on the training set and 0.701 on the validation set, both could be considered to indicate good discrimination.

Verplancke, et al. (2008) compared logistic regression and Support Vector Machines (SVM), and the result was that both the LR and SVM models were good. They compared the accuracy of predicting hospital mortality in patients with haematological malignancies admitted to the ICU between models based on LR and SVM. They concluded that the discriminative power of both the LR and SVM models was good. No statistically significant differences were found in discriminative power between both models for prediction of hospital mortality.

Macrina et al. (2010) compared two methods in machine learning, Support Vector Machine and neural networks, with the motivation that a model such as NN or SVM may show a higher discriminatory potency than standard multivariable models (logistic regression). However, they didn't include logistic regression in the comparison. Their result showed that both NN and SVM can predict risk of mortality with good discrimination. They argued that their work was the first to adopt neural networks and support vector machines, with the intention of assessing the somewhat long-term predictive task of a reasonably significant series of potential risk factors which include

pre-operative, operative and immediately post-operative variables found in patients with acute aortic dissection (AAD) type A.

There is still a wide possibility that a methodology in machine learning is worth to looking at and can be compared with logistic regression to predict the risk of mortality. Therefore, in Chapter 3 we will compare the performance of logistic regression with other methods in machine learning to see whether different techniques could be used alternatively to solve a problem that has previously been solved satisfactorily by logistic regression. Especially, we will compare decision trees with the performance of logistic regression. To be compared with logistic regression, we also use some other methods in machine learning to bridge the existing gap in the research literature. And we also conducted experiments to assess the stability of the models by using 10-fold cross validation method.

2.7 Recognising and responding to clinical deterioration

Ensuring that patients who are the sickest in hospital receive proper and timely care is the key to meeting safety and quality challenges. All patients should receive the same level of comprehensive care regardless of their location in the hospital or the time of day. In the previous section (2.6), we have discussed the rationale for investigating the risk of mortality. Early risk assessment would be very useful to facilitate clinician decision making, in particular identifying patients at high or low risk. Especially for those high risk patients it can allow them to receive more individualized treatment, for example care in the intensive care unit (ICU).

Goldhill, et al. (1999), however, found that patients admitted from the wards to the ICU have a greater mortality rate in contrast to patients admitted from the operating/recovery and accident and emergency departments. This means there is a need to recognise the state where the patient who was not originally categorized as high risk, can suddenly need more serious treatment due to deterioration. For this reason, hospital staff (e.g. nurses) need to identify patients whose condition has deteriorated such that they need additional care (e.g. from a doctor). Resources are limited so that the selection of patients who might benefit from critical care is crucial. Identifying medical in-patients at risk of deterioration at an early stage may reduce the number of pre-ICU resuscitations (Subbe, Kruger, Rutherford, & Gemmel, 2001).

2.7.1 Systems for recognising and responding to clinical deterioration

Contributing factors to the failure to recognize and respond to deteriorating patients are complex and overlapping. Goldhill, et al. (1999) identified some issues including: not monitoring vital signs consistently; not detecting changes to vital signs; and lack of knowledge about the signs and symptoms that may indicate deterioration.

It is important to note that specific systems have been implemented to deal with these issues, to provide a structure for dealing with patients whose conditions worsen in hospital. "Rapid Response System" is the generic term often used for these systems.

A significant number of hospitals have launched rapid response systems (RRS) to enable the early identification of adult patients whose conditions are

worsening, and to support the delivery of enhanced care to the patient's bedside (Duckitt, et al., 2007; Gao, et al., 2007; Subbe, et al., 2001).

A rapid response system incorporates a system of early identification of warning signs that could indicate deterioration in a patient. It also includes processes to ensure a timely response to these signs, in order to reduce the chances of further deterioration of events.

2.7.2 Rapid Response System: recognising and responding to clinical deterioration

Most RRSs use a set of predetermined, largely objective, ''calling criteria'' as indicators of the need to call for more expert help. These sets of calling criteria, also known as ''track and trigger'' systems, can be categorised as single-parameter systems, multiple-parameter systems, aggregate weighted scoring systems or combination systems (Smith, Prytherch, Schmidt, & Featherstone, 2008). A scoring system based on a single parameter, called the MET calling criteria, was first developed in Australia (Cuthbertson & Smith, 2007).

A trigger can be thought of as sets of calling criteria to identify when the patient's condition has deteriorated. This recognition must then be used to deal with factors that could increase deterioration in the following hours. This action is called a response to the trigger. In the following sub section we will describe the process to recognize and respond to clinical deterioration.

2.7.2.1 Early Warning Score system for recognising to clinical deterioration

Subbe, *et al.* (2001) investigated the ability of a trigger score to identify the risk of catastrophic deterioration and found that there is association between raised score with increased mortality. That is also discovered by Quarterman, Thomas, McKenna, & McNamee (2005) who showed that there is a significant relationship between trigger score and patient outcome.

The major components of a rapid response system's *trigger* are: processes for monitoring vital signs; what criteria need to be met, including changes to vital signs; how the call for assistance is made; and observation charts and methods of recording vital signs.

The trigger itself is raised by the early warning score (EWS) system. An EWS typically assigns a score of 0, 1, 2 or 3 to a given physiological variable. 0 is assigned to values in the normal range and 1, 2 or 3 are given as the variable becomes more abnormal. The EWS is the total of the individual scores. The EWS is then used to determine what, if any, further action is required, following a pre-determined "escalation protocol". The EWS is primarily intended as an aid for more junior, less experienced members of staff.

In other words, we can define an early warning, or "track and trigger", system as a structured process to measure basic vital signs and act on the results. Once the criterion is reached (the *trigger*) an action must be initiated. Gao, et al. (2007) classified track and trigger systems as:

1. Single-parameter systems – periodic observation of selected vital signs which are compared to a simple set of criteria with predefined thresholds, with a response algorithm being activated when any criterion is met;

2. Multiple-parameter systems – where the response algorithm involves more than one criterion being met or differs according to the number of criteria met;

3. Aggregate weighted scoring systems – where weighted scores are assigned to physiological values and compared to predefined trigger thresholds;

4. Combination systems – involving single- or multiple-parameter systems in combination with aggregate weighted scoring systems.

Early warning systems differ in terms of the vital signs that they measure. Also the weighting of these measures, the way measures are combined, and finally the cut-off criterion used to trigger a response or action may vary (Smith, Prytherch, Schmidt, & Featherstone, 2008; Smith, Prytherch, Schmidt, Featherstone, & Higgins, 2008)

Two systematic reviews have inferred that the performance of the majority of early warning systems has been labelled as poor by two contemporary systematic reviews. It has been shown that there is little evidence to support the claim that they are reliable, valid, and serve a practical purpose (Gao, et al., 2007; Smith, Prytherch, Schmidt, & Featherstone, 2008).

However, contrary to the lack of evidence, the UK's National Institute for Health and Clinical Excellence (NICE) issued guidelines recommending that the physiological track and trigger systems should be implemented to monitor adult patients who are in acute hospital situations (National Institute for Health and Clinical Excellence, 2007).

2.7.2.2 Responding to clinical deterioration

There are a number of different models to reflect different circumstances in which to respond to those patients who are deteriorating.

The three most common models can be summarised as follows:

1. MET

   Australia is the first country to use MET (medical emergency team) and it was developed at Liverpool Hospital in Sydney and first introduced in 1990 (Lee, Bishop, Hillman, & Daffurn, 1995)

2. Rapid Response Team

   The terms “rapid response team” and “medical emergency team” tend to be used interchangeably in Australia, however it has been implemented widely in the United States (DeVita, Bellomo, & Hillman, 2006)

3. Critical care outreach

   Critical care outreach teams have been primarily established in United Kingdom, and generally include critical care services provided to patients on general wards and follow-up of patients from ICU (McDonnell et al., 2007)

Chapter 3 Developing a model of risk of mortality using routinely collected data

3.1 Introduction

Chapter 3 describes predictive modelling of the risk of mortality. This chapter uses administrative and laboratory data which has been obtained from Portsmouth NHS Hospitals Trust, the Biochemistry and Haematology Outcome Model (BHOM) dataset. The dataset obtained from 1 January to 31 December 2001 and divided into four subsets. One subset of training data was used to generate a model, and the model obtained was then applied to three testing datasets.

There are four main things that will be done in this chapter:

- We follow the research that has been done by Prytherch, et.al. (2005) - the use of routine laboratory data to predict in-hospital death in medical admissions. We follow the same track by using the same data (BHOM dataset) and then use the same method (logistic regression) to generate a model. The performance of each model was then compared using the same analysis methods; calibration (the $\chi 2$ test or chi-test) and discrimination (area under ROC curve or c-index).

- To bridge the existing gap in the research literature, we will focus on using decision trees as the potential method and then we compare their performance with that of logistic regression. We also consider some

techniques in machine learning in order to find alternative methods to predict risk of mortality.

- We investigate whether the comparison is "fair". We conducted experiments to assess the stability of the models by using a 10-cross validation method. We use t-test statistics to assess models from cross validation.
- We also propose a new measurement, exhaustive method, to assess the performance of the method for predicting risk of mortality.

3.2 Design of a System to Predict Clinical Outcomes

In this section, we will start from the design of the system as shown in Figure 3.1. At the beginning, the 'Generate Model' section loads training data to the system to generate a model. The process to generate a model can be achieved using SPSS as a tool or by developing a program in MATLAB to produce a model which generates results from the training data.

The resulting model is then saved and loaded again at the time of the 'Applied Model' stage when the model is used to obtain the outcome of the testing data. The results of applying this model to testing data then go further into the 'Assess Performance' stage to assess the discrimination and calibration of the result.

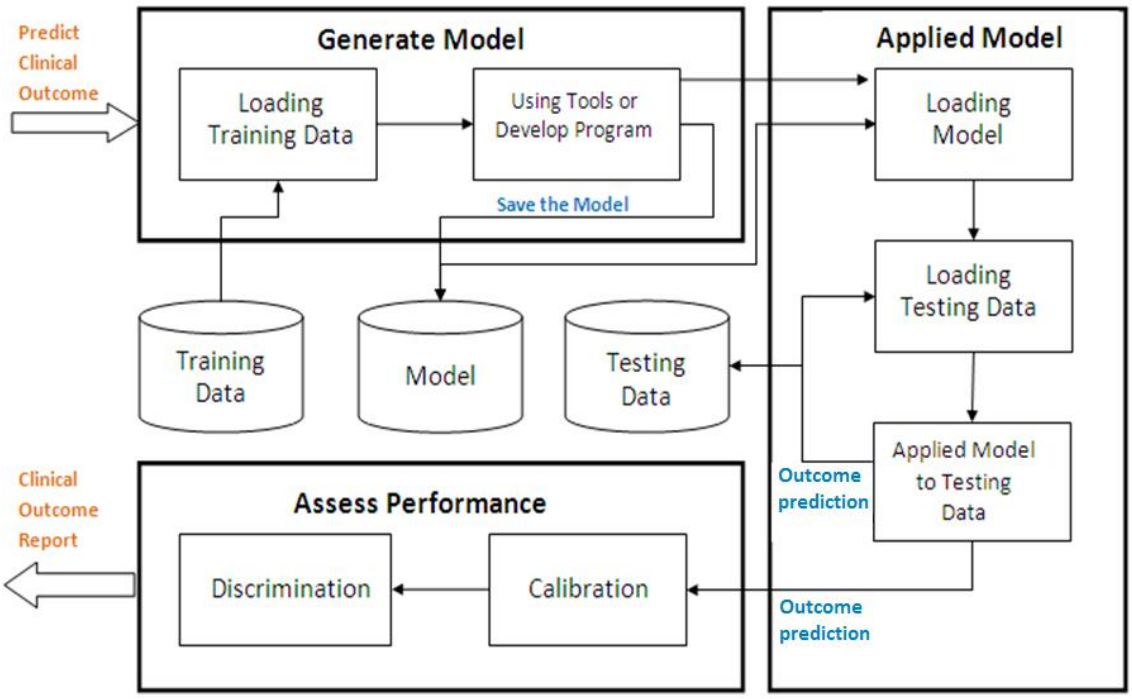


Figure 3.1 Design of System to Predict Clinical Outcome

3.3 Ethical Considerations

The dataset used in this thesis is covered by our second supervisor Prof. David Prytherch's existing ethical approval. The title for the purpose of the Research Ethics Committee (REC) is "Case-mix adjusted predictive models of adverse clinical outcomes". The name of the Research Ethics Committee is Isle of Wight, Portsmouth and South East Hampshire Research Ethics Committee. The REC reference number is 8/2/1394.

3.4 Data Description

The study in this chapter focuses on predicting the risk of an adverse clinical outcome - mortality for all general admissions to a hospital. This is done using routinely collected data.

This chapter uses administrative and laboratory data which has been obtained from the hospital pathology and administrative computer systems at Portsmouth NHS Hospitals Trust. This particular dataset was the Biochemistry and Haematology Outcome Model (BHOM) dataset, which contains 9497 adult hospital discharges, and it was divided into four subsets:

- Q1 dataset – n1 = 2257 - data from 1 January to 31 March 2001
- Q2 dataset – n2 = 2335 - data from 1 April - 30 June 2001
- Q3 dataset – n3 = 2361 - data from 1 July - 31 September 2001
- Q4 dataset– n4 = 2544 - data from 1 October - 31 December 2001

A model was built using a training set (Q1) corresponding to three months' worth of patients. The model obtained from Q1 dataset was then applied to three testing data sets: Q2, Q3 and Q4.

The fields in the dataset are:

- death - at discharge - F=alive, T =dead (class attribute)
- Age at admission
- mode of Admission – (mostly emergency, but some elective)
- Gender (male or female)
- Haemoglobin (unit of measurement is mmol/l)

- White cell count (unit of measurement is WBC count (10^9/l))
- Urea (unit of measurement is mmol/l)
- Serum sodium (unit of measurement is mmol/l)
- Serum potassium (unit of measurement is mmol/l)
- Creatinine (unit of measurement is mmol/l)
- Albumin (unit of measurement is mmol/l)
- Urea / creatinine ratio (unit of measurement is mmol/l)

Where sex and mode of admission are categorical attributes coded, "F" for female, "M" for male, "Elec" for elective admission and "Emer" for emergency admission.

In the dataset, *death* attribute is the target attribute or dependent attribute. There are 11 (eleven) independent attributes which determine the value of the dependent attribute (death) in the dataset.

3.5 The characteristics of the dataset

The characteristics of the dataset are provided in Appendix 2. In that appendix, the Q1 dataset is shown in Table 1. This dataset was used as the training data to generate a model. The characteristics of the other datasets (Q2, Q3, Q4) are shown in Tables 2, 3 and 4. The percentage of hospital mortality in each dataset is under 10%. It means very few patients are known as dead. Therefore, we can categorize the dataset as an unbalanced dataset. For such kinds of datasets, as we mentioned in Chapter 2, the accuracy rate is not suitable to assess the performance of the model. The reason is that even if

the model fails to predict all cases in the minority class (i.e. death), the accuracy rate would still be good, being around 90%.

As can be seen by comparing the characteristics among datasets, from Appendix 2, in Tables 1-4, each of the four datasets has similar characteristics, e.g. percentage of the gender have the balanced proportion.

There are also balanced percentage of hospital mortality between male and female. From tables in Appendix 2, we can see that the number of emergency admissions much exceeds elective admission. In other words, only a few patients were admitted as "elective".

3.6 Assessing performance of a model

3.6.1 Discrimination using area under ROC curve (AUROC)

As discussed in chapter 2 (section 2.4.1), the c-index (also known as the area under ROC curve (AUROC)) is the most appropriate method for assessing discrimination in the healthcare area, especially for an unbalanced dataset.

In our dataset, discrimination refers to the ability to accurately discriminate between two conditions. 'Survivor' and 'non-survivor' have been selected in this instance. The discriminatory ability of each model can be analysed by using receiver-operating characteristics (ROC) curves. Referring to the area under the ROC curve, it is summarised by the c-index, and it has a range of between 0.500 (no predictive ability) and 1 (perfect discrimination). Between these, a c-index value of 0.700-0.800 would indicate a reasonable

discrimination. Anything above 0.800 can be considered as a good discrimination

3.6.2 Calibration using chi-test

In addition to evaluating the performance of the model using c-index as the measurement of discrimination, we need to develop stratification models that can help us calibrate. For this purpose, the outcome of the model is divided into bands of risk of mortality. The bands range from the lowest risk to the highest risk.

In chapter 2 we described calibration as a degree of correspondence between the estimated risk produced by the model and the actual observed risk. In our experiment, chi-test used to evaluate the observed and predicted case in each risk band in the stratification model.

The Hosmer-Lemeshow statistic is an appropriate test and the most popular measure of calibration (Lemeshow & Hosmer, 1982). Individual records in the validation subset are grouped by risk range. The risk bands are divided from the lowest level until the highest risk band level, with the following risk bands adjustment:

- ≥ 0 to < 5 (lowest),
- ≥ 5 to < 7.5,
- ≥ 7.5 to < 10,
- ≥ 10 to < 12.5,
- ≥ 12.5 to < 15,
- ≥ 15 to < 20,
- ≥ 25 to < 33,
- ≥ 33 to < 50,
- ≥ 50 to ≤ 100 (highest).

For each risk band, the predicted number of deaths is compared to the number observed. Therefore, in each risk band, we need to specify:

- Number of cases
- Mean predicted risk
- Number of predicted
- Number of observed
- Chi-test

Goodness-of-fit is assessed using the χ^2 test (chi-test). As this is a null hypothesis test, p values less than 0.05 indicate evidence of significant lack of fit. The form of chi-test as shown in Equation 2.20.

Section 3.8.3 will implement the calibration using chi-test with the detail explanation to calculate those values in each risk band.

3.6.3 Exhaustive method

We propose a new measurement to assess the performance of the model using what we term the exhaustive method. The basic idea of this method is to compare the risk of mortality of each record (episode) with the other records in the dataset.

As an illustration of the exhaustive method, we describe the following scenario. For example, we have record A_i which we will compare with another record A_j.

If the risk of mortality A_i *greater than* A_j AND the outcome of A_i is dead and the outcome of A_j is alive, we count this as a success. In other words, for person A_i that has the risk of mortality greater than person A_j, the outcome of

A_i is dead and the outcome of A_j is alive, indicates success of the method to predict risk of mortality.

If the risk of mortality A_i is *less than* the risk A_j, AND the outcome of A_i is alive and the outcome of A_j is dead, we also count this as a success.

If the outcome of A_i *is equal to* A_j, (dead or alive) then we don't do anything in this condition.

Otherwise, if none of the above condition is fulfilled, then we indicate that the method has failed to predict risk of mortality. In this case, we count it as a failure.

Once all of the records have been compared with each other record, we calculate discrimination of exhaustiveas:

Discrimination=success/ (success+failure).

We can affirm that the exhaustive method is a method of discrimination. The algorithm to implement exhaustive method will be presented in the section 3.8.4.

3.6.4 t-test statistics to assess models from cross validation

In Chapter 2 section 2.4.3, we illustrated the t-test statistic to evaluate the differences between two methods. When conducting cross validation experiments in section 3.10, we use the t-test statistic to evaluate the c-index produced by each method.

For all the measurements we used in section 3.6 (c-index, chi-test, exhaustive method and t-test statistic), a method can be regarded as the best method if the result of calibration indicates no evidence of significant lack of fit and the

result of discrimination (c-index) has the largest value when compared to all other methods. Specifically for cross validation experiments, we evaluate the c-index and t-test statistics to find out which method is superior to other methods. By conducting cross validation experiments, we can evaluate the stability of the method when dealing with dataset sampled in different ways.

3.7 Developing a Risk of Mortality Model using SPSS

In this section, we describe how an outcome model was constructed from logistic regression and decision trees using IBM SPSS statistical software version 19. We used Q1 dataset as training data to build a model, and compared the performance of logistic regression and decision trees model when applied to three testing (Q2,Q3,Q4) datasets.

3.7.1 Logistic Regression Model

3.7.1.1 Building model

In Chapter 2 (2.3.3), we explained that logistic regression can be used when the output variable of a model is defined as a binary categorical. In this section, we describe how we used the Q1 (as specified in the section 3.5) to build a model using the logistic regression facilities in SPSS.

Open the Q1 dataset file. Click *Analyse, Regression, Binary Logistic*. Put *death* variable into the Dependent box and all other variables except *id* into the Covariates box. The dialog box should now look like Figure 3.2:

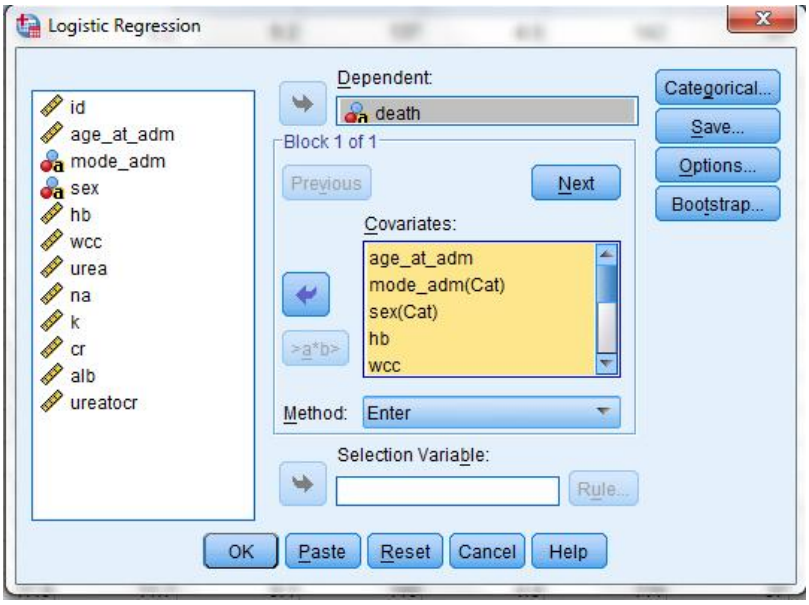


Figure 3.2 Generate Logistic Regression Model using SPSS

The **Variables in the Equation** output in Figure 3.3 shows us that the logistic regression model produced by SPSS (where R is the risk of mortality) is:

Equation 3.1 :

$$\text{Ln}\left(\frac{R}{1-R}\right) = \text{-4.493+(0.013 × gender)+)+ (-0.037 × haemoglobin) +(0.067 × white cell count)+ (0.018 × urea)+ (-18.714 × mode of admission)+ (0.053 × age on admission)+ (-0.001 × Serum sodium)+ (-0.101 × Serum potassium)+ (0.001 × creatinine) +(-0.047 × albumin)+ (2.744 ×urea/creatinine ratio).}$$

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)	
Step 1[a]	age_at_adm	.053	.007	64.749	1	.000	1.054	
	mode_adm(1)	-18.714	5072.945	.000	1	.997	.000	
	sex(1)	.013	.174	.006	1	.941	1.013	
	hb	-.037	.038	.984	1	.321	.963	
	wcc	.067	.013	25.930	1	.000	1.069	
	urea	.018	.025	.528	1	.467	1.018	
	na	-.001	.018	.007	1	.935	.999	
	k	-.101	.131	.595	1	.440	.904	
	cr	.001	.002	.472	1	.492	1.001	
	alb	-.047	.015	10.017	1	.002	.954	
	ureatocr	2.744	4.192	.428	1	.513	15.547	
	Constant	-4.493	2.693	2.783	1	.095	.011	

Variable(s) entered on step 1: age_at_adm, mode_adm, sex, hb, wcc, urea, na, k, cr, alb, ureatocr.

Figure 3.3 Variables in the Equation Output

We can now use this model to **predict the odds** of a subject dying. The odds prediction equation is DEATHS = $e^{\alpha+\beta_1.X_{1j}+\beta_2.X_{2j}+\cdots+\beta_n.X_{nj}}$ where the coefficients in the equation are taken from the table above.

In the **Categorical Variables Codings** output (shown in Figure 3.4), we can see that SPSS has coded the categorical variables itself. As we can see, sex and mode of admission are coded, female = 1, male = 0, elective = 1, and emergency = 0, respectively.

Categorical Variables Codings

		Frequency	Parameter coding (1)
sex	F	1118	1.000
	M	1139	.000
mode_adm	Elec	55	1.000
	Emer	2202	.000

Figure 3.4 Categorical Variables Codings output

Knowing all the variable coefficients in Figure 3.3 are not directly we can get the risk of mortality. We need to create a new syntax in SPSS to express categorical variables codings and produce the odds prediction of DEATHS. Click *File, New, Syntax* and type the following syntax as shown in Figure 3.5.

```
DO IF (sex EQ "M").

COMPUTE         Gender=0.
ELSE IF (sex EQ "F").
COMPUTE         Gender=1.
END IF.

DO IF (mode_adm EQ "Emer").
COMPUTE         Adm=0.
ELSE IF (mode_adm EQ "Elec").
COMPUTE         Adm=1.
END IF.

COMPUTE TOTAL=-4.493+age_at_adm*0.053+ureatocr*2.744+alb*-0.047+
cr*                0.001+k*-.101+na*-0.001+urea*0.018+wcc*0.067+hb*-0.037+
Gender*0.013+Adm*-18.714.
COMPUTE PROB=EXP(TOTAL)/(1+EXP(TOTAL)).

EXECUTE.
```

Figure 3.5 Developing syntax to calculate the probability attribute

Run syntax and now, we have a new attribute *PROB* which expresses the probability of the patient being likely to die. With PROB as probability attribute of patients will die and DEATH as the target attribute, we can calculate the performance of the model using a ROC Curve.

Click *Analyse, ROC curve*. Put *PROB* variable into the Test Variable box and *death* into the State Variable box. Give the value of 'T' in the Value of State Variable. Check in the option *Standard error and confidence interval*. The dialog box should now look like this:

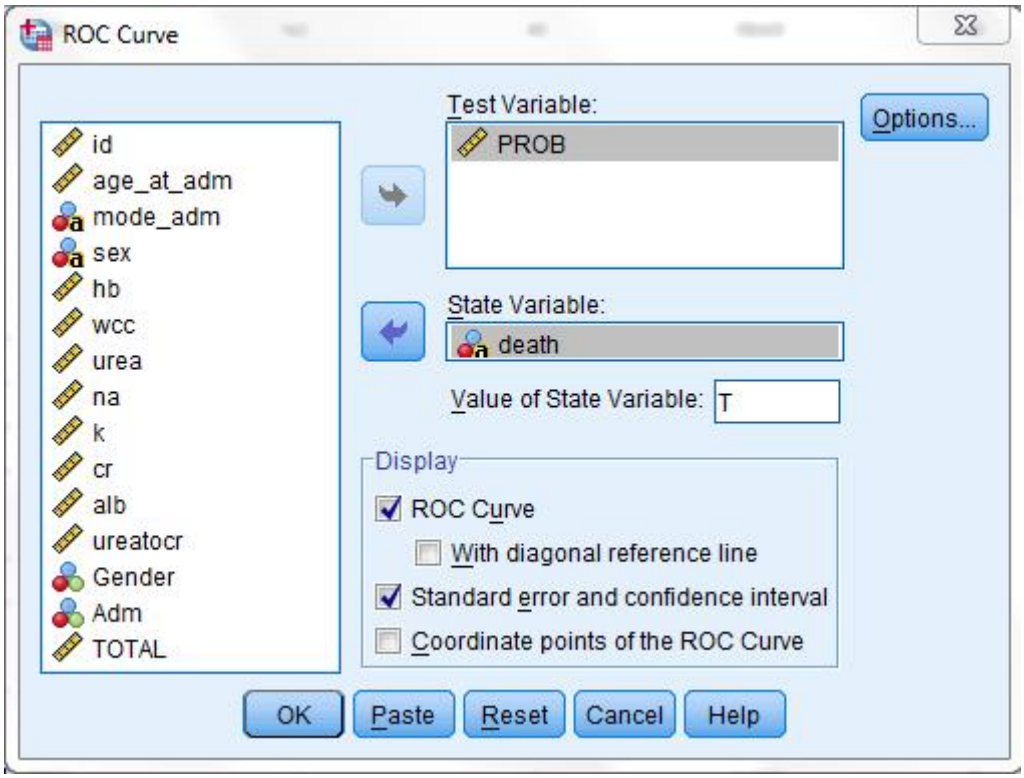
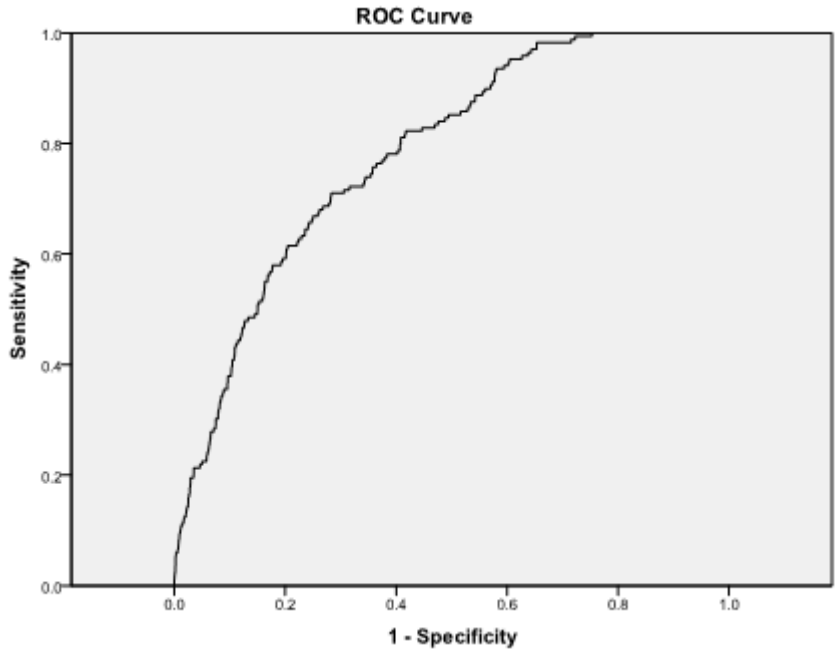


Figure 3.6 Generate area under ROC curve for Logistic Regression model

As we explained in Chapter 2 (2.4.1.3), the vertical axis of an ROC curve represents *sensitivity*, and the horizontal axis represents *1-specificity*. The plotting of area under the ROC curve is shown in Figure 3.7.



Area Under the Curve

Test Result Variable(s):PROB

Area	Std. Error[a]	Asymptotic Sig.[b]	Asymptotic 95% Confidence Interval	
			Lower Bound	Upper Bound
.781	.016	.000	.749	.814

a. Under the nonparametric assumption

b. Null hypothesis: true area = 0.5

Figure 3.7 Area under ROC curve for Q1 dataset model

In Figure 3.7, we can see the performance of the logistic regression model using Q1 dataset is 0.781 with the confidence interval (CI) 0.749 to 0.814. This performance (0.781) is in the range between 0.700 and 0.800, indicating reasonable discrimination, as we have explained in the previous section (3.6.1). In the next section we will validate the Logistic Regression model produced by Q1 dataset into three other datasets Q2, Q3, and Q4.

3.7.1.2 Applied model to testing data

In the previous section, a model that was built using logistic regression on a training set (Q1) produced c-indices of 0.781(CI: 0.749 to 0.814) indicating reasonable discrimination. In this section, we will validate the three other testing datasets (Q2, Q3 and Q4) and will see the performance of the logistic regression model when validating other datasets.

Logistic regression using SPSS tools based on the BHOM Q1 training set produced the outcome model in Equation 3.1.To validate other three testing datasets (Q2, Q3 and Q4), we need to open a testing dataset and use the same syntax as we used before in Figure 3.5 to produce *prob*attribute as probability of patient likely to die.

In the same way, we obtained the area under ROC curve for all three testing data as follows in Table 3.1.

Table 3.1 The performance of model when validating other datasets

No.	Dataset	Area under ROC curve (c-index)
1	Q2	0.779 (Confidence Interval : 0.748-0.810)
2	Q3	0.764 (Confidence Interval : 0.729-0.799)
3	Q4	0.758 (Confidence Interval : 0.725-0.790)

All the results in Table 3.1 indicate reasonable discrimination as the result of c-index between 0.700 and 0.800. In the next section, we will evaluate the performance of the model using decision trees model in SPSS and we will see whether the model does as well or better than the model produced by logistic regression.

3.7.2 Decision Trees Model

In Chapter 2, we illustrated how to generate a decision trees model from the dataset. We already mentioned that there are some attribute selection measures that are used for building decision trees. One of them is CHAID, which stands for Chi-Square Automatic Interaction Detector. This is a decision tree algorithm that uses an attribute selection measure based on the statistical χ^2 test for independence (Han, et al., 2006).

3.7.2.1 Building Model

In this section, we will show the use of CHAID method in SPSS to predict risk of mortality model. Using the same training data as previous section, open Q1 dataset. To run a decision tree, from the menus choose: *Analyse, Classify, Tree*. Put *death* variable into the Dependent box and select all the remaining variables except *id* as Independent variables. The dialog box should now look like Figure 3.8.

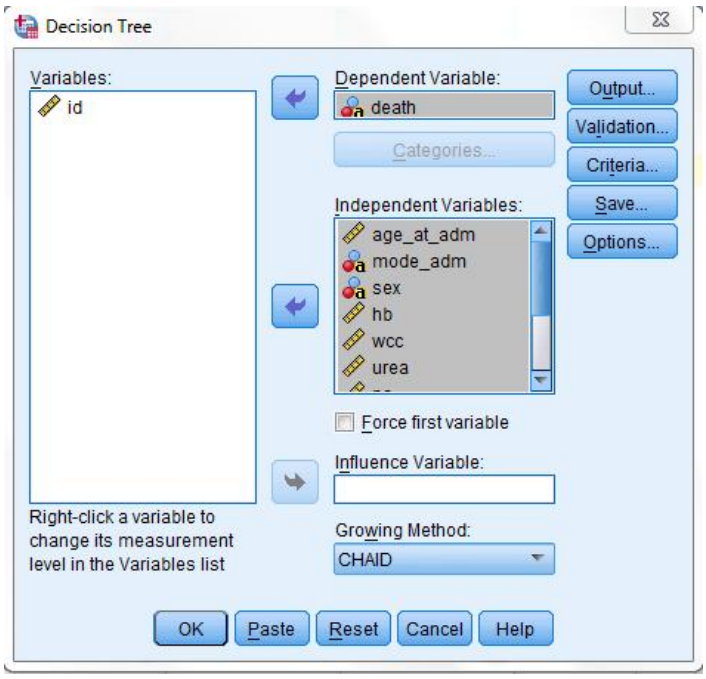


Figure 3.8 Generate Decision Trees model using SPSS

When we build a model, we can calculate the probabilities in each record. To save the value of probabilities, click button *Save* in Figure 3.8, and then check the option *Predicted probabilities,* the dialog box should look like Figure 3.9.

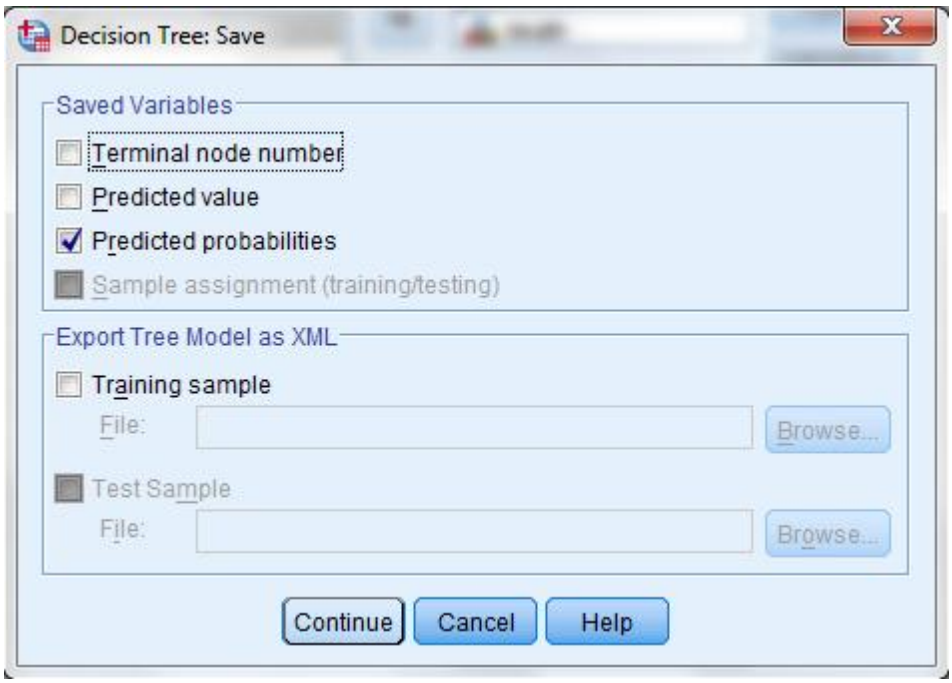


Figure 3.9 The option to save predicted probabilities in Decision Trees model

Click OK to run the procedure. Figure 3.10 shows the resulting decision trees model. We also have an additional variable which expresses the predicted probabilities of patients when death='F' and death='T'.

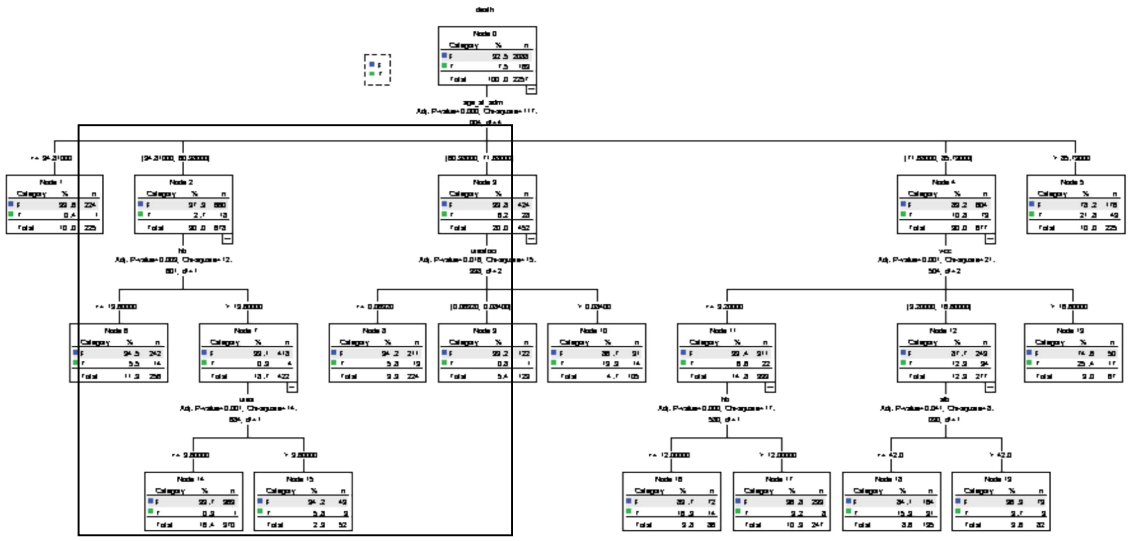


Figure 3.10 Complete Decision Trees model

Figure 3.11 is the zoom-out from rectangle area in Figure 3.10.

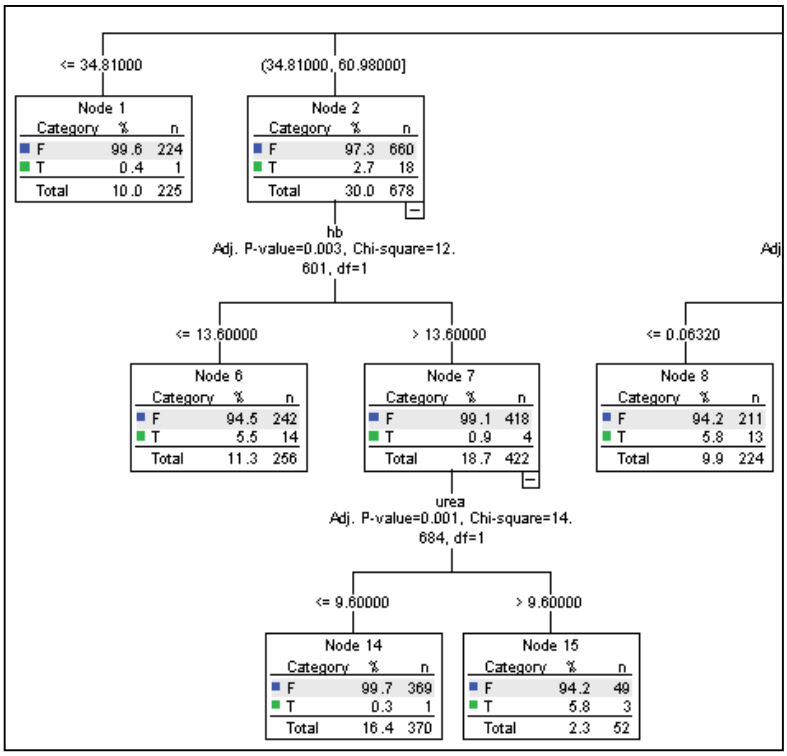


Figure 3.11 Zoom-out from Decision Trees model in Figure 3.10

The number of risk bands in decision trees is determined by the number of terminal nodes (leaves) that exist on the tree. Based on the modelling results in Figure 3.10, we can see that there are as many as 13 risk bands.

From the zoom-out decision trees model in Figure 3.11, there are five risk bands as assigned with node 1, node 6, node 8, node 14 and node 15. Among the five nodes, node 14 is the lowest level of risk band with the probability of risk of mortality of only 0.3%: only one person is reported dead from a total of 370 people in this node. The percentage of people who fall into this node is 16.4% of the total number of patients. Whereas if we look at the complete decision trees model in Figure 3.10, we found that the highest level of risk band that is at node 13, the probability of risk of mortality is 25.4%: 17 people are reported dead out of a total of 50 people in this node.

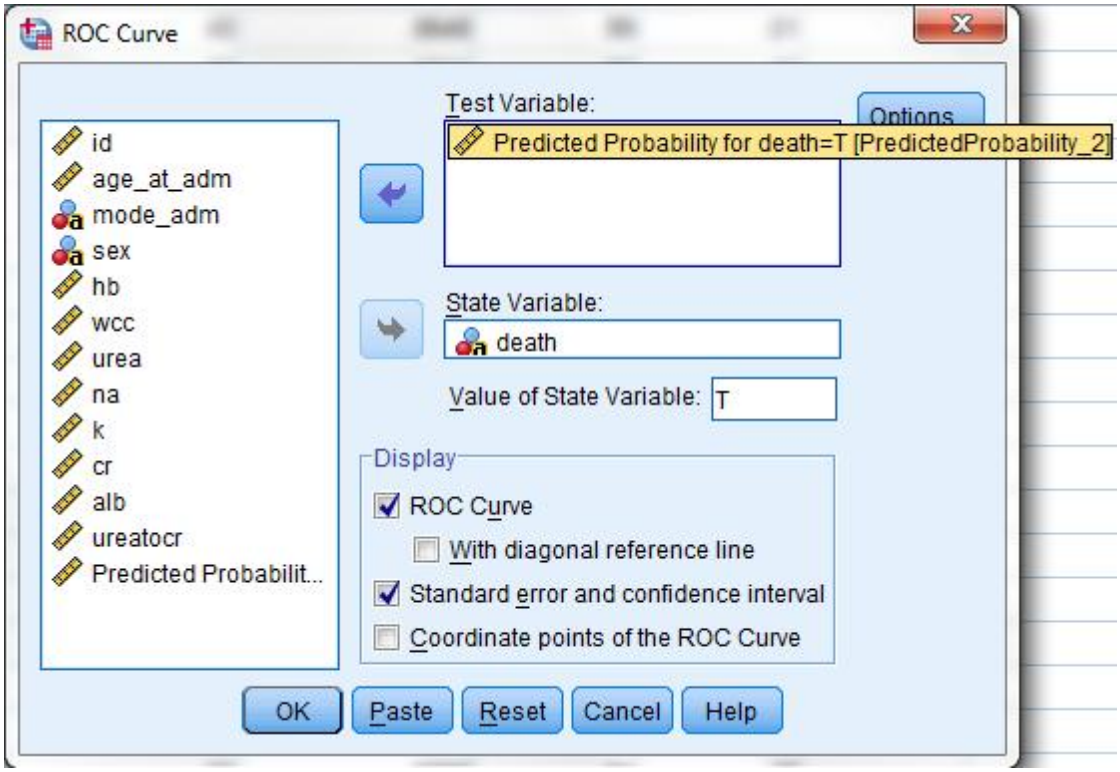
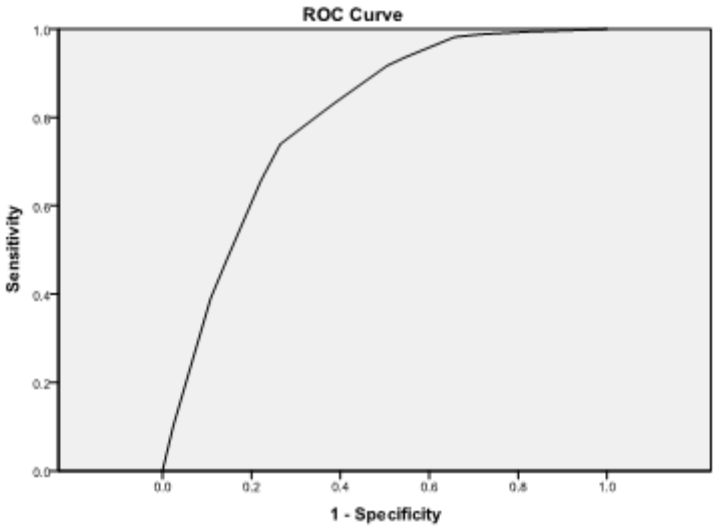


Figure 3.12 Generate area under ROC curve for Decision Trees model

To assess the performance of the model, we need to calculate area under ROC curve. Click *Analyse, ROC curve*. Put *Predicted Probability for death=T* variable into the Test Variable box and *death* into the State Variable box. Give the value of 'T' in the Value of State Variable. Check in the option Standard error and confidence interval. The dialog box should now look like Figure 3.12.

From Figure 3.13, we can see the performance of the decision trees model in terms of discrimination (c-index) using Q1 dataset is 0.796 with the confidence interval (CI) 0.767 to 0.825, indicating good discrimination and slightly better than c-index produced by logistic regression model (0.781 with CI= 0.749 to 0.814).



Area Under the Curve

Test Result Variable(s):Predicted Probability for death=T

Area	Std. Error[a]	Asymptotic Sig.[b]	Asymptotic 95% Confidence Interval	
			Lower Bound	Upper Bound
.796	.015	.000	.767	.825

Figure 3.13 Area under ROC curve for Q1 dataset using Decision Trees model

In the next section we will validate the decision trees model produced by Q1 dataset into three other datasets Q2, Q3, and Q4.

3.7.2.2 Applied model to testing data

In the previous section, a decision trees model was built using a training set (Q1) and produced c-index of 0.796 (CI 0.767 to 0.825) indicating reasonable discrimination. In this section, we will validate three other testing datasets (Q2, Q3 and Q4) and will see the performance of the decision trees model when validating other datasets.

To enable the decision trees model to be validated with other datasets, we need to generate classification rules and need to save into a file in order to load the classification rules when we evaluate the other datasets. To do this, click button *Output* in Figure 3.8, and click option *Generate classification rules* and *Export rules to a file* and then specify the file name. The dialog box should look like Figure 3.14.

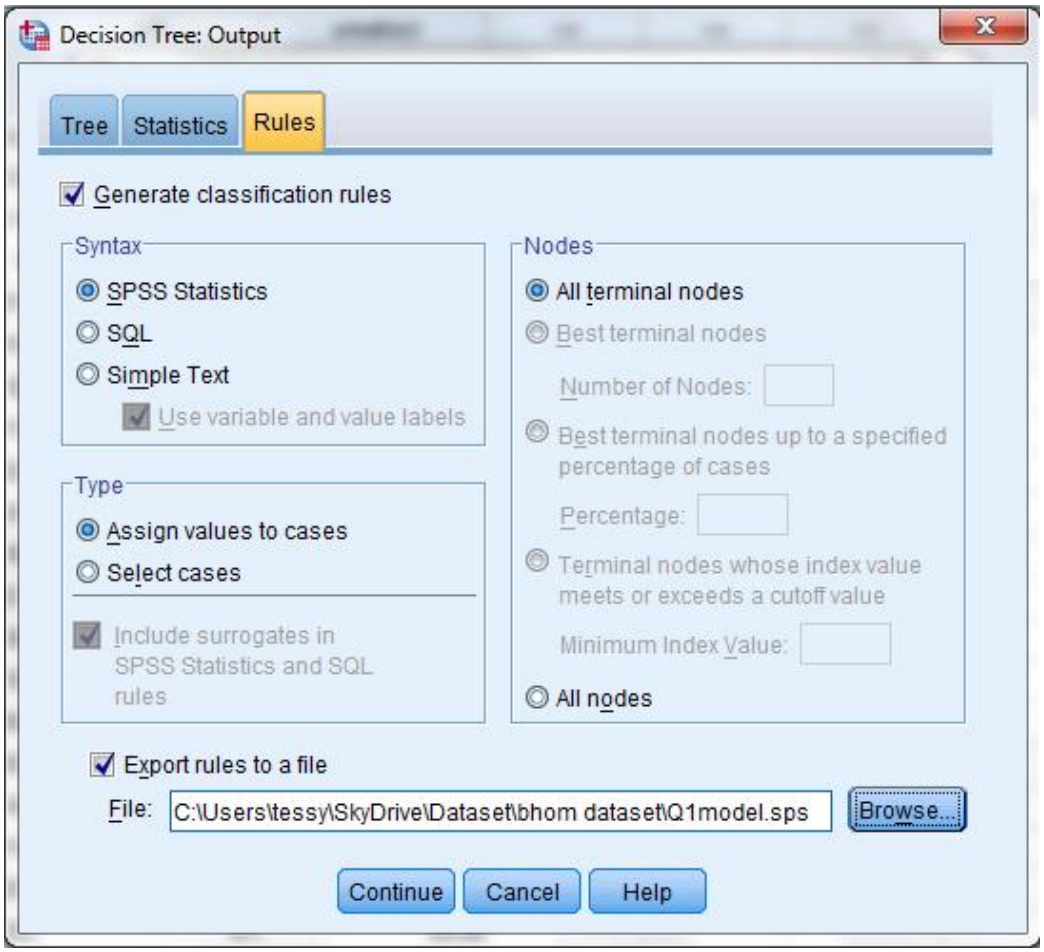


Figure 3.14 Saving Decision Trees model

The decision trees model will be saved into *Q1model.sps*. After having classification rules files as model results, we can now apply that model to other data files containing same structure as testing data (Q2, Q3 and Q4) and then generate a new variable predicted probabilities for each record in that file.

Open testing dataset Q2. From the menus choose: *File, New,* and *Syntax*. In the command syntax window, type:

INSERT FILE='C:*File directory*\Q1model.sps'.

After running the syntax, we obtained *Predicted value, Terminal node number* and *Predicted probabilities.* Predicted probabilities which we get is predicted probabilities for death='F', because what we need is predicted probabilities for death = T, then we need to compute a new variable as 1-predicted probabilities.

To obtain area under ROC curve (c-index), use the same dialog box with Figure 3.12. We obtained c-index for all three testing data as follows in Table 3.2.

Table 3.2 The performance of Decision Trees model when validating other datasets

No.	Dataset	Area under ROC curve (c-index)
1	Q2	0.735(Confidence Interval : 0.701-0.770)
2	Q3	0.721(Confidence Interval : 0.684-0.759)
3	Q4	0.700(Confidence Interval : 0.666-0.735)

3.7.3 The effects of changes of the type of data in the independent attributes

In the original Q1 dataset, independent attributes sex and mode of admission have 'categorical' as type of data. When the logistic regression model was built, under the *Categorical Variables Codings,*these attributes have been coded automatically by SPSS, based on the following coded values: female = 1, male = 0, elective = 1, and emergency = 0, respectively.

The regression equation obtained from original Q1 dataset was in Equation 3.1:

$\text{Ln}\left(\frac{R}{1-R}\right)$= -4.493+(0.013 × gender)+)+ (-0.037 × haemoglobin) +(0.067 × white cell count)+ (0.018 × urea)+ (-18.714 × mode of admission)+ (0.053 × age on admission)+ (-0.001 × Serum sodium)+ (-0.101 × Serum potassium)+ (0.001 × creatinine) +(-0.047 × albumin)+ (2.744 ×urea/creatinine ratio).

In this section, we are conducting an experiment to change the type of data in the independent attributes of Q1 dataset, from categorical to numeric data type. We purposely give different coded values: female = 0, male = 1, elective = 0, and emergency = 1, respectively. And further, we give the name of the new dataset as *Q1new* dataset.

The regression equation obtained from Q1new dataset is:

Equation 3.2 :

$\text{Ln}\left(\frac{R}{1-R}\right)$= -23.194+(-0.013 × gender)+)+ (-0.037 × haemoglobin) +
(0.067 × white cell count)+ (0.018 × urea)+ (18.714 × mode of admission)+ (0.053 × age on admission)+ (-0.001 × Serum sodium)+
(-0.101 × Serum potassium)+ (0.001 × creatinine) +(-0.047 × albumin)+
(2.744 ×urea/creatinine ratio).

Comparing Equation 3.1 from Q1 dataset and Equation 3.22 from Q1new dataset, there are some differences as follows:

- The intercept has been changed from -4.493 to -23.194.
- The slopes of gender and mode of admission have been changed only in the sign, where (-0.013) in the previous one for gender, change to 0.013. While slope of mode of admission from 18.714, change to (-18.714).

Even though they have some differences in the intercept and in the sign of two attributes, for those two models we obtained exactly the same area under ROC curve (c-index) as 0.781(CI: 0.749 to 0.814). And also under Model Summary we see that the -2 Log Likelihood statistics is exactly the same for those two datasets as 1025.827. This statistic measures how poorly the model

predicts the decisions. With the same value of c-index and also -2 Log likelihood, we can conclude that even though the two datasets have some differences on intercept and slopes, they are exactly similar on the results and we should not worry about categorical data type because SPSS will code it into numerical attributes automatically.

The same results were also obtained in the decision trees model. There are no effects of changes from categorical data type to numeric data type and the same result obtains as well for nominal and ordinal data types. The resulting tree model is exactly the same for all those type of attributes.

3.7.4 Discussion of the Results

Table 3.3 compares the performance between decision trees and logistic regression using SPSS in the case of discrimination. We used the Q1 dataset as training data to build a model and then applied this model to three other datasets (Q2,Q3,Q4).

Table 3.3 Comparison discrimination between Logistic Regression and Decision Trees model using SPSS

Dataset	No. of cases	The area under ROC curve (c-index)	
		Logistic Regression	Decision Trees
Q1	2257	0.781(CI : 0.749- 0.814)	0.796 (CI 0.767- 0.825)
Q2	2335	0.779 (CI : 0.748-0.810)	0.735(CI : 0.701-0.770)
Q3	2361	0.764 (CI : 0.729-0.799)	0.721(CI : 0.684-0.759)
Q4	2544	0.758 (CI : 0.725-0.790)	0.700(CI : 0.666-0.735)

As mentioned in the introduction in this chapter, we follow the same track as Prytherch, et.al. (2005) by using the same data and method. The results obtained from our experiment are exactly the same as that paper, producing c-index 0.781 (Q1), 0.779 (Q2), 0.764 (Q3). We are slightly different with c-index 0.758 for Q4 while Prytherch, et.al. (2005) obtained 0.757 for Q4.

Looking at the results in Table 3.3, it is obvious that the differences between logistic regression and decision trees are trivial and also both in the range of 0.700 – 0.800, therefore we can conclude that both models have reasonable performance in terms of discrimination.

From Table 3.3, building models using training data and then applying those to training data itself, obtained the resulting model of decision trees (0.796) which is slightly better than the performance of the logistic regression model (0.781). However, when applied to three other datasets, the logistic regression model does slightly better than decision trees for all testing datasets.

Between logistic regression and decision trees the models have different forms. While the logistic regression model expresses as equation of intercept and the slopes of independent attributes, the decision trees model has the form of a tree as the model result.

The representation of the tree model allows intuitive understanding of the equation. From the tree, we can see that the *Age at admission* attribute is the root of the tree model, so we can say that age attribute is the most influential attribute. When we built a decision trees model in SPSS, we used CHAID method. In each step, CHAID always selects the independent variable which shows the strongest relationship to the dependent variable. Assuming that the dependent variables are reasonably similar and then the categories of each predictor are merged. From the resulting model, only 6 of 11 independent attributes were found to be relevant with dependent variable (death) and appear in the decision trees model. Those attributes are age at admission, haemoglobin, creatinine ratio, urea, white cell count and albumin.

We already know from the result of our experiments that the changes made in the type of data of being categorical into numeric, doesn't make any difference in the resulting model. The only differences in the logistic regression model are the value of the intercept and the sign of slopes in the regression equation. In the decision trees model, on the other hand, the

changes of data type, give no change at all in the resulting model. Both the logistic regression and decision trees models obtained exactly the same results, in terms of probability value and the performance of the model. Therefore we can conclude that we should not worry about the type of categorical data, and do not need to recode the categorical data as numeric.

3.8 Developing a risk of mortality model using MATLAB

In this section, we developed code in MATLAB (Ver. R2011b) to construct an outcome model of logistic regression and decision trees. We used the Q1 dataset as training data to build a model, and compared the performance of the logistic regression and decision trees models when applied to three testing data (Q2,Q3,Q4) datasets.

3.8.1 Logistic regression model

3.8.1.1 Building the model

We used glmfit as a built-in function in MATLAB to generate logistic regression. Developing function in MATLAB to derive and applied logistic regression models as shown in the following algorithm:

Algorithm 3.1 Logistic Regression in MATLAB

1: Load training data
2: Get independent attributes *X* and dependent attribute *Y* from training data
4: Get coefficient estimates *B* using glmfit
5: B = glmfit(X, [Y ones(xx,1)], 'binomial', 'link', 'logit')
6: Load testing data to applied model
7: **for all** records in testing data **do**
8: Get odds of deaths using coefficient *B*
9: Z=B(1) + X(1)*(B(2)+X(2)*B(3)+ ... X(n)*B(n+1);
10: prob=(exp(Z))/(1+exp(Z));
11: **end for**
10: Save *prob* into file to be evaluated

Where:

B(1) = the intercept of logistic regression

B(2) ... B(n) = coefficient of slope of independent attributes

From Algorithm 3.1, , glmfit function returns a coefficient estimate for a generalized linear regression of the target variable Y on the independent attributes in X.

Logistic regression with developing MATLAB function produced the following outcome model:

Equation 3.3 :

$$\text{Ln}\left(\frac{R}{1-R}\right) = -4.493 + (0.013 \times \text{gender}) + (-0.037 \times \text{haemoglobin}) + (0.067 \times \text{white cell count}) + (0.018 \times \text{urea}) + (-100.057 \times \text{mode of admission}) + (0.053 \times \text{age on admission}) + (-0.0015 \times \text{Serum sodium}) + (-0.101 \times \text{Serum potassium}) + (0.001 \times \text{creatinine}) + (-0.047 \times \text{albumin}) + (2.744 \times \text{urea/creatinine ratio}).$$

The intercept of the logistic regression model produced by MATLAB is exactly the same as the intercept produced by SPSS (-4.493). All the slopes of independent attributes produced by MATLAB are the same as those produced by the SPSS except mode of admission attribute. Using SPSS, we got (-18.714) for mode of admission, and using MATLAB, we got (-100.057) for mode of admission. We are not sure of the reason for the discrepancy, but suspect it is related to minor algorithmic differences in the two implementations.

In the previous section, we evaluated the performance of the logistic regression model by calculating the area under ROC curve using SPSS. To keep consistency, the model produced by MATLAB will also be evaluated using ROC curve in SPSS.

The performance of logistic regression by MATLAB for Q1 dataset is 0.781 with the confidence interval (CI) 0.748 to 0.810, indicating reasonable discrimination. This discrimination is exactly the same with the c-index produced by SPSS (0.781), only slightly different in the confidence interval, while in SPSS the confidence interval (CI) is 0.749 to 0.814.

In the next section we will validate the logistic regression model produced by the Q1 dataset against three other datasets Q2, Q3, and Q4.

3.8.1.2 Applied Model to Testing Data (Validation)

In the previous section, a logistic regression model was built using a training set (Q1) in the MATLAB produced c-indices of 0.781 (CI : 0.748 to 0.810) indicating reasonable discrimination. In this section, the logistic regression model produced by MATLAB is validated against three other testing datasets (Q2, Q3 and Q4) and the performance of the testing dataset evaluated using area under ROC curve in SPSS. We obtained the area under ROC curve for all three testing datasets as follows in Table 3.4.

Table 3.4 The performance of the logistic regression model using MATLAB when validating other datasets

No.	Dataset	Area under ROC curve (c-index)
1	Q2	0.779 (Confidence Interval : 0.748-0.810)
2	Q3	0.765 (Confidence Interval : 0.729-0..800)
3	Q4	0.757 (Confidence Interval : 0.724-0.790)

All the results in indicate reasonable discrimination, and look similar to the result of the logistic regression model using SPSS in Table 3.1.

In the next section, we will evaluate the performance of the model using the decision trees model in MATLAB and we will see whether the model performs as well as or better than the model produced by logistic regression.

3.8.2 Decision Trees Model

3.8.2.1 Building the Model

We used the built-in function *classregtree* in MATLAB to generate the decision trees model. Function *classregtree* uses CART method as described in section 2.2.1. The model that has been generated then applied to testing datasets using the built-in function *eval*. Developing function in MATLAB to derive and applied decision trees models as shown in the following algorithm:

Algorithm 3.2 Decision Trees in MATLAB

1: Load training data
2: Get independent attributes *X* and dependent attribute *Y* from training data
4: Train classification decision tree using*classregtree*
5: t = classregtree(X, Y, 'method', 'classification')
6: Applied to training data itself
7: [yPredicted, leafnode] = eval(t, X);
8: Calculate risk of mortality (*prob)* for all terminate node

9: **for all** terminate node in the model **do**
10: calculate risk of mortality (*probs)*
11: **end for**
12: Load testing data
13: Get independent attributes *X* and dependent attribute *Y* from testing data
14: Applied model (t) to testing data *X*
15: [yPredicted, leafnode]= eval(t, X);
16: **for all** records in testing data **do**
17: refer risk of mortality of leafnode based on *probs*in the model
18: **end for**
19: Save risk of mortality of leafnode (*prob)* into file to be evaluated

---

Function *classregtree*in returns classification rules as follows:

t =

Decision tree for classification

1 if age_at_adm<88.075 then node 2 elseif age_at_adm>=88.075 then node 3 else 0

2 ifwcc<22.75 then node 4 elseif wcc>=22.75 then node 5 else 0

3 if age_at_adm<88.9 then node 6 elseif age_at_adm>=88.9 then node 7 else 0

.....

222 if age_at_adm<75.175 then node 224 elseif age_at_adm>=75.175 then node 225 else 0

223 class = 0

224 class = 0

225 class = 1

From the above decision rules, almost all independent attributes (predictor variables) involved in the trees, 9 of 11 attributes, (all except mode of admission and gender) do not involve the trees. Number of rules indicates the number of leaf nodes in the decision trees. To know the number of leaf nodes in the trees, we can use :

m=size(unique(hasiltest))

>>m=99

Therefore, there are 99 terminal nodes in the trees. MATLAB does not provide the risk of mortality for each terminal node when we use the built-in function *eval*, therefore we have to calculate it by using the formula :

Risk of mortality = $\frac{ndie}{nnode}$

Where:

ndie is number of patients dead in terminal node

nnode is number of patients who fall in the terminal node

Due to the size of picture, we cannot display the resulting tree.

We also evaluated the performance of the decision trees model produced by MATLAB using ROC curve in SPSS. The performance of the decision trees model by MATLAB for the Q1 dataset is 0.982 with the confidence interval (CI) 0.976 to 0.988, indicating very good discrimination. This result is much better than the previous result of the decision trees model by SPSS (0.796 with (CI) 0.767 to 0.825).

In chapter 2, we mentioned the danger of being overoptimistic when the result of decision trees is quite good due to using the same dataset for both training and testing. In the case overoptimistic, when the model applied to other dataset, the result is not so good. Decision trees producing by MATLAB has a large tree with the number of terminate node is 113. When the model of

Q1 apply to other datasets, as previously mention before, the result of discrimination have poor discrimination of the following (0.647, 0.591, 0.607) for (Q2, Q3, Q4) dataset, respectively.

To prevent overoptimistic, we applied pruning strategy to simplify decision trees. In MATLAB, pruning strategy is implemented by using built-in function *prune*:

T = prune(tree, 'LEVEL')

The value LEVEL=0 means no pruning.

We choose to set level therefore the number of terminate nodes in the tree around 20. After pruning, the performance of the decision trees model by MATLAB for the Q1 dataset is 0.780 (CI: 0.742 to 0.818) indicating reasonable discrimination.

3.8.2.2 Applied Model to Testing Data (Validation)

In this section we will check, when the model using pruning is applied to other datasets, whether the discrimination is still good or not. We will validate the decision trees model by the Q1 dataset against three other datasets Q2, Q3, and Q4.

To enable the decision trees model to be validated with other datasets, we also use the built-in function *eval* and put the Q2, Q3, and Q4 as testing data. The following Table 3.5 shows the performance of the decision trees model using MATLAB when validated against other datasets. Only dataset Q2 in the discrimination results in Table 3.5 is more than 0.700 indicating reasonable discrimination. Other datasets have discrimination below 0.700.

Table 3.5 The performance of Decision Trees model using MATLAB when validating other datasets

No.	Dataset	Area under ROC curve (c-index)
1	Q2	0.705 (Confidence Interval : 0.668-0.742)
2	Q3	0.681 (Confidence Interval : 0.641-0.721)
3	Q4	0.688 (Confidence Interval : 0.653-0.724)

When applied to other datasets, we got discrimination of (0.705, 0. 681, 0.688) for (Q2, Q3, Q4), respectively. However, these results are better compared to the result before pruning.

3.8.3 Implementation of stratification model and calibration using chi-test

This section will implement the calibration using chi-test. We use Hosmer-Lemeshow statistics which grouped individual records by 10 risk ranges, from the lowest level ($0 \leq$ risk bands < 5) up to the highest risk band level ($50 \leq$ risk bands ≤ 100). In the logistic regression model, each individual record has its own risk based on the calculation of independent attributes put into the regression equation. Therefore, we can implement the logistic regression model using Hosmer-Lemeshow statistics. In the decision trees model, on the other hand, the risk of individual records is based on the risk of mortality of each leaf node. Therefore the number of bands of risk in the decision trees depends on the number of leaf nodes. Due to the differences between the two methods, we will implement the Hosmer-Lemeshow grouping of risk bands into the logistic regression model and treat the decision trees model differently, based on the existing leaf nodes in the trees.

As we obtained before in the decision trees model using SPSS, there are 13 leaf nodes, therefore there are 13 unique risks of mortality. In the decision trees model using MATLAB, we used pruning, therefore the number of leaf

nodes (terminate nodes) as many as 21, each leaf node has their own risks of mortality (Figure 3.15).

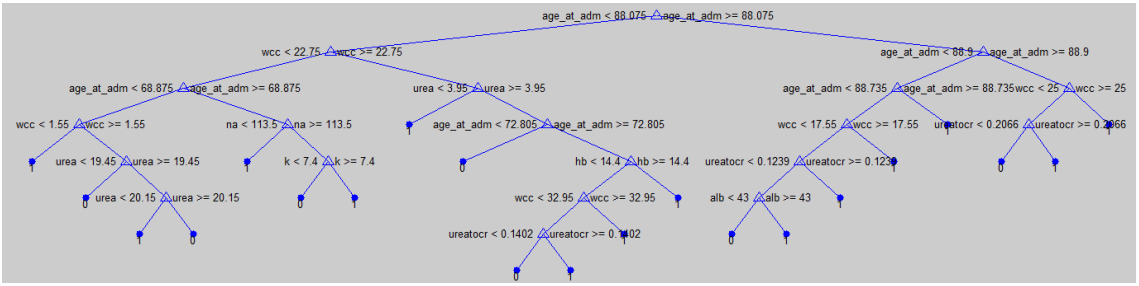


Figure 3.15 Decision trees model produced by MATLAB

Generally, for each risk band, we need to specify: number of cases, mean predicted risk, number of predicted, number of observed and chi-test value. First, we will implement the calculation of number of cases, mean predicted risk, number of predicted, number of observed in.

Algorithm 3.3 Calculation the mean predicted risk, number of predicted and observed

1: set predicted=0, numberofcases=0
2: **for all** records in testing data **do**
3: **if** risk in the range of risk bands **then**
4: predicted=predicted+risk;
5: numberofcases= numberofcases+1;
6: **if** (deaths attribute ==’Y’) **then**
7: observed=observed+1;
8: **end if**
9: **end if**
10: **end for**
11: for all index=1 to number of risk bands do
12: mean predicted risk(index)= $\frac{predicted(index)}{numberofcases(index)} * 100$
13: **end for**

Secondly, we will use the calculation from the first step to calculate the chi-test for each risk band in Algorithm 3.4.

Algorithm 3.4 Calculation chi-test for each risk bands

1: **for** each risk band **do,**
2: % Calculate chi-value for death
3: $chideath = \frac{(observed-predicted)^2}{predicted}$
4: % Calculate chi-value for alive
5: expalive=number of cases - predicted;
6: actalive=number of cases - observed;
7: $chialive = \frac{(expalive-actalive)^2}{expalive}$
8: % Calculate chi-test
9: chitest=chideath+chialive;
10: **end for**

Equation 3.11 was obtained when the logistic regression model was built using code automatically generated by SPSS, based on these following coded values: female = 1, male = 0, elective = 1, and emergency = 0, respectively. The regression equation obtained from the original Q1 dataset was in :

$\text{Ln}\left(\frac{R}{1-R}\right)$ = -4.493+(0.013 × gender)+ (-0.037 × haemoglobin) +
(0.067 × white cell count)+ (0.018 × urea)+ (-18.714 × mode of admission)+
(0.053 × age on admission)+ (-0.001 × Serum sodium)+
(-0.101 × Serum potassium)+ (0.001 × creatinine) +
(-0.047 × albumin)+ (2.744 ×urea/creatinine ratio).

Equation 3.22 was obtained when the logistic regression model was built using SPSS and the coded value was based on the paper by Prytherch, et.al. (2005), with the following coded values: female = 0, male = 1, elective = 0, and emergency = 1, respectively. The regression equation was in:

$\text{Ln}\left(\frac{R}{1-R}\right)$= -23.194+(-0.013 × gender)+)+ (-0.037 × haemoglobin) +(0.067 × white cell count)+ (0.018 × urea)+ (18.714 × mode of admission)+ (0.053 × age on admission)+ (-0.001 × Serum sodium)+ (-0.101 × Serum potassium)+ (0.001 × creatinine) +(-0.047 × albumin)+ (2.744 ×urea/creatinine ratio).

We obtained exactly same results of stratified modelling as shown in Table 3.6 by using Equation 3.1 and Equation 3.2. In both equations, we used Q1 as training data and applied it to the Q2 dataset. In the last row of this table, we then calculate the total of mean predicted risk, the total number of death predicted, the total number of death reported and the value of χ^2 (chi-test).

Table 3.6 Stratification of Logistic Regression Model using SPSS, based on Equation 3.1/ Equation 3.2
Goodness-of-fit by Hosmer-Lemeshow χ2 statistic for (Q2) data covering period 1 April –30 June 2001

Risk bands	No. of cases	Mean predicted risk (%)	Predicted	Observed	χ^2
≥ 0 to < 5	1037	2.07	22	16	1.44
≥ 5 to < 7.5	298	6.21	18	17	0.13
≥ 7.5 to < 10	240	8.65	21	22	0.08
≥ 10 to < 12.5	202	11.15	23	27	1.00
≥ 12.5 to < 15	150	13.62	20	20	0.01
≥ 15 to < 20	174	17.23	30	31	0.04
≥ 20 to < 25	97	22.18	22	22	0.01
≥ 25 to < 33	77	28.09	22	12	5.97
≥ 33 to < 50	46	39.10	18	17	0.09
≥ 50 to ≤ 100	14	61.00	9	7	0.71
≥ 0 to ≤ 100	2335	8.96	203	191	9.48

Calibration: χ^2 = 9.48; 8 d.f.; H–L *p*-value =0.303; *discrimination: c*-index = 0.779 (CI: 0.748-0.810)

In Table 3.6, those different coded values have exactly the same stratification model. Therefore the different coded values do not cause a change in calibration.

To be compared, in Table 3.7 is shown the stratification model by Prytherch, et.al. (2005) for the same dataset as used in Table 3.6. From those tables, there is no an evidence of significant lack of fit demonstrated by p-values greater than 0.05.

Table 3.7 Stratification of Logistic Regression Model using SPSS by Prytherch, et.al. (2005) Goodness-of-fit by Hosmer-Lemeshow χ2 statistic for (Q2) data covering period 1 April –30 June 2001

Risk bands	No. of cases	Mean predicted risk (%)	Predicted	Observed	χ^2
≥ 0 to < 5	1037	2.24	26	21	1.03
≥ 5 to < 7.5	298	6.47	16	19	0.53
≥ 7.5 to < 10	240	8.95	25	32	1.92
≥ 10 to < 12.5	202	11.50	18	19	0.13
≥ 12.5 to < 15	150	13.92	20	22	0.22
≥ 15 to < 20	174	17.67	27	33	0.50
≥ 20 to < 25	97	22.75	17	14	0.81
≥ 25 to < 33	77	28.92	17	13	1.53
≥ 33 to < 50	46	40.46	15	14	0.11
≥ 50 to ≤ 100	14	64.89	6	4	1.65
≥ 0 to ≤ 100	2335	8.05	188	191	9.43

Calibration: χ^2 = 9.43; 8 d.f.; H–L *p*-value =0.307; *discrimination: c*-index = 0.779

From those tables Table 3.6 and Table 3.7, we can conclude that our experiment and that of Prytherch, et.al. (2005) in logistic regression, satisfied both discrimination and calibration value when the c-index in the range 0.700-0.800 is reasonable discrimination and the p-value > 0.05 indicates there is no evidence of significant lack of fit.

Table 3.8 shows the stratification results of the logistic regression model using MATLAB.

Table 3.8 Stratification of Logistic Regression model using MATLAB Goodness-of-fit by Hosmer-Lemeshow χ2 statistic for (Q1) data covering period 1 January–31 March 2001

Risk bands	No. of cases	Mean predicted risk (%)	Predicted	Reported	χ^2
≥ 0 to < 5	1113	2.06	23	27	0.72
≥ 5 to < 7.5	320	6.15	20	20	0.01
≥ 7.5 to < 10	240	8.70	21	16	1.25
≥ 10 to < 12.5	170	11.28	19	19	0.00
≥ 12.5 to < 15	122	13.65	17	18	0.13
≥ 15 to < 20	127	17.21	22	29	2.82
≥ 20 to < 25	75	22.36	17	11	2.56
≥ 25 to < 33	57	27.91	16	15	0.07
≥ 33 to < 50	25	40.77	10	9	0.24
≥ 50 to ≤ 100	8	61.30	5	5	0.00
≥ 0 to ≤ 100	2257	9.37	169	169	7.80

Calibration: χ^2 = 7.80; 8 d.f.; H–L *p*-value = 0.303434; *discrimination: c*-index = 0.781 (CI: 0.748-0.810)

In the same way, Table 3.9 below evaluates the result of stratification model of logistic regression using the Q1 dataset as training data applied to testing data (Q2,Q3 and Q4) using SPSS and MATLAB.

Table 3.9 Stratification model of Logistic Regression model using SPSS and MATLAB

Dataset	SPSS			MATLAB		
	c-index	χ^2	p-value	c-index	χ^2	p-value
Q1	0.781	9.24	0.32	0.781	7.80	0.30
Q2	0.779	9.48	0.30	0.779	9.09	0.33
Q3	0.764	23.36	0.0029	0.765	11.84	0.16
Q4	0.758	6.62	0.58	0.757	5.89	0.66

Our stratification model of logistic regression using SPSS produced $\chi^2 = 9.48$ (Q_2), $\chi^2 = 23.36$ (Q_3) and $\chi^2 = 6.62$ (Q_4) (p-values of 0.30, 0.0029, and 0.01) for 8 degrees of freedom while the stratification model by Prytherch, et.al. (2005) gave $\chi^2 = 9.43$ (Q_2), $\chi^2 = 7.39$ (Q_3) and $\chi^2 = 8.00$ (Q_4) (p-values of 0.307, 0.495 and 0.433) for 8 degrees of freedom. All the results indicate good calibration, except that our model for the Q3 dataset as $\chi^2 = 23.36$ (p-value =0.0029 < 0.005) indicates there is evidence of significant lack of fit. However, the discrimination for Q4 has reasonable discrimination (0.758).

In the previous, we compared the calibration using the logistic regression model and the decision trees model in SPSS with detail of illustration. In decision trees model using MATLAB, due to the size of trees (there are 21 terminate nodes), we only report the value of chi-test and p-value. Table 3.10 shows the result of stratification model of the decision trees model using the Q1 dataset as training data applied to testing data (Q2,Q3 and Q4) using SPSS and MATLAB.

Table 3.10 Stratification model of Decision Trees model using SPSS and MATLAB

Dataset	SPSS			MATLAB		
	c-index	χ^2	p-value	c-index	χ^2	p-value
Q1	0.796	0	>0.05	0.780	0	>0.05
Q2	0.735	67.23	<0.05	0.705	112.52	<0.05
Q3	0.721	159.87	<0.05	0.681	120.17	<0.05
Q4	0.700	133.19	<0.05	0.688	101.03	<0.05

In the term of calibration, decision trees cannot be successfully generated using both SPSS and MATLAB, it has reasonable discrimination but there is an evidence of significant lack of fit for all datasets which indicates that the calibration is poor. However, when using SPSS to get the model of decision trees, both training and testing datasets have reasonable discrimination.

3.8.4 Implementation of Exhaustive method to assess performance of the model

This section will implement our new proposed measurement, exhaustive method, to assess the performance of the model.

Algorithm 3.5 is a step by step procedure to calculate the measurement using exhaustive method as explained in section 3.6.3.

Algorithm 3.5: Exhaustive method to assess performance of the model

```
1: for A=1 to number of records do,
2:     for B=1 to number of records do,
3:         % If index of A not equal index of B, compare the risk and outcome
4:             if (A not equal B) then
5:    % If risk of A greater than risk of B
6:    if (risk(A)>risk(B) AND outcome(A)=dead AND outcome(B)=alive) then
7:             success=success+1
8:    % If risk of A less than risk of B
9:    elseif(risk(A)<risk(B) AND outcome(A)=alive AND outcome(B)=death)
      then
10:            success=success+1
11:   % If outcome of A equal outcome of B
12:   elseif(outcome(A)=outcome(B) then
13:            ; // don't do anything
```

14: % otherwise (if not satisfied all above condition)

15: **Else** fail=fail+1

17: **End if**

18: **End if**

31: **End for**

32: **End for**

We then applied exhaustive method to the model using SPSS and MATLAB for the two methods; logistic regression and decision trees.

Table 3.11 demonstrates the value of exhaustive method most likely the same with c-index. However, when using Q3 as testing data by SPSS, there is an evidence of significant lack of fit indicated by χ^2 = 23.36 (p-value < 0.05) while the exhaustive value is still has reasonable discrimination (0.764).

Table 3.11 Performance of discrimination, calibration and exhaustive method of Logistic Regression model using SPSS and MATLAB

Chapter 4	Chapter 5		**SPSS**	Chapter 6		**MATLAB**
	c-index	**χ^2, p-value**	**Exhaustive method**	**c-index**	**χ^2, p-value**	**Exhaustive method**
Q1	0.781(CI: 0.749-0.814)	9.24	0.781	0.781(CI : 0.748-0.810)	7.80, 0.30	0.781
Q2	0.779 (CI : 0.748-0.810)	9.48, 0.30	0.779	0.779 (CI : 0.748-0.810)	9.09, 0.33	0.779
Q3	0.764 (CI : 0.729-0.799)	23.36, 0.0029	0.764	0.765 (CI : 0.729-0.800)	11.84, 0.16	0.764
Q4	0.758 (CI : 0.725-0.790)	6.62, 0.01	0.757	0.757 (CI : 0.724-0.790)	5.89, 0.66	0.757

Table 3.12 shows the performance of c-index, χ^2 (chi-test) and exhaustive method using SPSS and MATLAB. From that table we can see that the value of exhaustive method is different than c-index. By using SPSS as a tool, when the calibration of the model is poor, the discrimination of exhaustive method also has poor discrimination, all the values are less than 0.700 for dataset

Q2,Q3 and Q4 by SPSS. However, the values of c-index for those results are still in reasonable discrimination (between 0.700-0.800), but we need to be careful with Q4 dataset by SPSS which only has c-index of 0.700 - nearly a poor discrimination, with the smallest of the exhaustive method values (0.660). From this we can conclude that exhaustive method measurement still consistently follows the value of discrimination with c-index and calibration with p-value.

Table 3.12 Performance of c-index, $\chi 2$ (chi-test) and exhaustive method of decision trees model using SPSS and MATLAB

Dataset	SPSS	MATLAB				
	c-index	χ^2, p-value	Exhaustive method	c-index	χ^2, p-value	Exhaustive method
Q1	0.796 (CI : 0.767-0.825)	0, >0.05	0.759	0.780 (CI : 0.742-0.818)	0, >0.05	0.658
Q2	0.735 (CI : 0.701-0.770)	67.23, <0.05	0.696	0.705 (CI : 0.668-0.742)	112.52, <0.05	0.559
Q3	0.721 (CI : 0.684-0.759)	159.87, <0.05	0.685	0.681 (CI : 0.641-0.721)	120.17, <0.05	0.524
Q4	0.700 (CI : 0.666-0.735)	133.19, <0.05	0.660	0.688 (CI : 0.653-0.724)	101.03, <0.05	0.527

3.8.5. Discussion of the Results

In section 3.8, we developed MATLAB for some of the following tasks:

- To generate models of logistic regression and decision trees
- To develop a stratification model by calculating the mean predicted risk, number of predicted and observed for each risk band, and also calculating chi-test for each risk band
- To develop the implementation of our new measurement, exhaustive method.

All models of logistic regression generated from MATLAB satisfied both discrimination and calibration. As seen in Table 3.9, in the terms of discrimination, all the results of c-index were around 0.700-0.800 indicating a reasonable discrimination. Those results also have similar discrimination with logistic regression model produced in MATLAB.

However, the results of c-index for the decision trees model using MATLAB as seen in Table 3.10 is not as favourable. Our model when utilised on Q4 dataset has c-index = 0.750, two other testing datasets (Q3 and Q4) have poor performance with c-index under 0.700, comprising (0.681 and 0.688) for Q3 and Q4 correspondingly. All the testing dataset (Q2, Q3, Q4) have p-value < 0.05, indicating that there is an evidence of significant lack of fit.

From the results of Table 3.11, the implementation of a new proposed measurement, exhaustive method, is found to be most likely consistent with c-index and also p-value. Only when applied to the Q3 dataset using logistic regression in SPSS, is there evidence of significant lack of fit indicated by χ^2 = 23.36 (p-value < 0.05) while exhaustive value still has reasonable discrimination (0.764) and also has the same value with c-index.

Decision tree model both using SPSS and MATLAB show poor performance when applied to testing test, all of them below 0.700. When the result of discrimination in logistic regression is similar in the term of discrimination by c-index and exhaustive method and also calibration, the results of model in decision trees are difference between SPSS and MATLAB. The reason of this is because SPSS and MATLAB implement different methods of decision trees. While SPSS uses CHAID method, MATLAB uses CART method instead.

3.9 Developing a risk of mortality model using RapidMiner

In the previous section, we used MATLAB and SPSS to develop a risk of mortality model using Q1 as training data to build a model that was then applied to three testing datasets (Q2,Q3,Q4). In this section, with the same dataset, we will use RapidMiner tools to construct an outcome model from various machine learning methods. RapidMiner is one of open-source system for data mining. It is by Ralf Klinkenberg, Ingo Mierswa, and Simon Fischer in 2001 at the Artificial Intelligence Unit of the Dortmund University of Technology.

We decided to use RapidMiner, because it provides a lot of flexibility in the choice of method and its use. RapidMiner's framework consists of all the processes including (1) loading training data, (2) build a model (3) applied model to testing datasets, (4) calculating performance and (5) save the result, combined in one go, without the need for programming at all. With this ease of use, it will be easy to make a comparison among many methods to decide which method is worthwhile looking at.

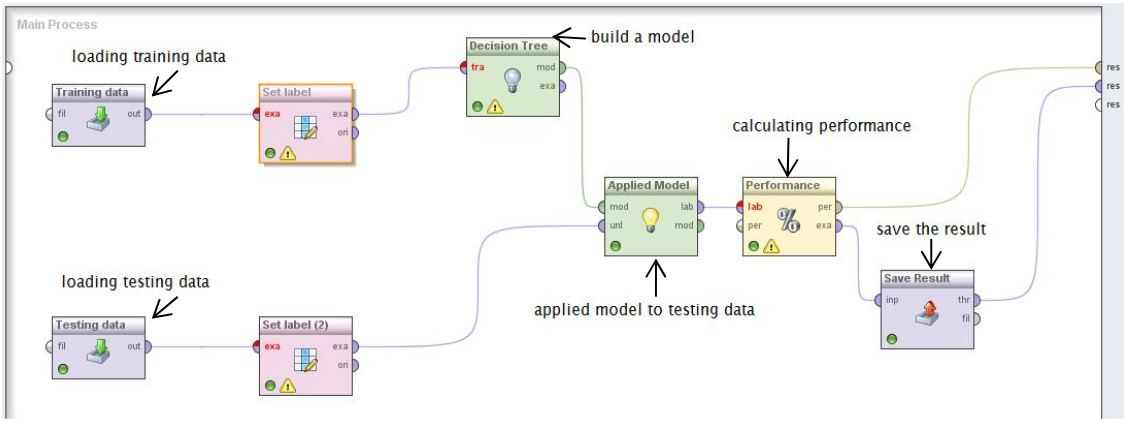


Figure 3.16 Main Process in RapidMiner's framework

The experiment in this section use Q1 dataset as training data to build a model, and compared the performance of various methods in machine learning when applied to three testing data (Q2, Q3, Q4) datasets. There are 6 methods used, logistic regression, decision trees(gini-index), neural networks (feed-forward), naïve bayes, support vector machines (libSVM) and k-nearest neighbours (based on explicit similarity measures).

Table 3.13 Comparison Stratified Modelling by RapidMinerFrameWork using Q1 as training data, Q2,Q3,Q4 as testing data

Method	c-index			
	Q1	Q2	Q3	Q4
LOGISTIC REGRESSION	0.781	0.779	0.765	0.757
DECISION TREES	0.865	0.660	0.684	0.678
NEURAL NETWORKS	0.807	0.732	0.706	0.704
NAIVE BAYES	0.762	0.761	0.753	0.727
SUPPORT VECTOR MACHINES	0.828	0.590	0.596	0.627
K-NEAREST NEIGHBOURS	0.777	0.757	0.722	0.701

Table 3.13 demonstrates the value of c-index of Logistic Regression produced by RapidMiner have consistently the same value with c-index produced by SPSS and MATLAB in Table 3.11.

Among 6 methods, LR give consistently reasonable discrimination. Following by NB and NN also give good results when applied to testing datasets. Even though NN gives a good discrimination when applied to training data (0.807), but still can give reasonable discrimination when applied to testing dataset. However LR performance in the term of discrimination applied to testing data is still better than NN and NB for all testing datasets.

DT and SVM give a good discrimination and much better than LR when trained Q1 dataset, however when applied to testing data, DT and SVM give a poor performance, this probably due to 'overfitting' as we discussed in the previous section. In RapidMiner, the c-index values are very good only when implemented into training data and becomes poor performance when implemented into testing data is called overfitting also occur in this such case

(the same thing as before in MATLAB when implemented DT using CART method).

3.10 Cross Validation

As mentioned before in section 3.6.4., we conducted an experiment using cross validation to evaluate the stability of the method when dealing the datasets in different ways. By looking at the possibility of resampling using cross validation, different sample of training and testing datasets could have different results. A good method always consistently gives good results for any sample data.

In this experiment, we used cross validation. There are two issues relating to the use of cross validation in this experiment which measure the stability of an algorithm and see the impact of the use of cross validation of the c-index values obtained.

Looking at k-fold cross-validation, the method is for the dataset to be partitioned into k sub-sets. One of the K subsets is then used as the validation data with the purpose of testing the model. The remaining (k-1) subsets are used as training data. The process of cross-validation is then repeated k times (the folds), with each of the k sub-sets being used just once as the validation data. Then, in a rotation system, each specific subset of data becomes the testing set in precisely one iteration.The performance of the method is the average of area under ROC curve (c-index) over the k iterations. In this study, we have used 10-fold cross validation.

3.10.1 Generate Dataset for Cross-Validation

We developed a program in MATLAB to generate the dataset for cross validation. We then used MATLAB to generate the model from the training dataset (cross(1) .. cross (10)) and then applied the model to the testing data (fold(1) .. fold(10)). The risk of mortality obtained in the dataset was then evaluated using SPSS to get the area under ROC curve (c-index) and the 95% confidence interval.The following algorithm generates cross validation from the whole dataset (Q1,Q2,Q3,Q4). The steps that should be done: (1) merge all the datasets into one combined dataset; (2) generate data randomly from the whole dataset to get 10 independence subsets of folds (fold(1) ... fold(10)); (3) for each one independence subset (fold(i)), the remaining records in the whole dataset (after data taken randomly) are saved to the cross dataset which will serve as training data (cross(i)).

Algorithm 3.6 : Generate file for doing cross validation

1: ALL = Q1 + Q2 + Q3 + Q4
2: **for** times=1 to 10 **do**
3: % Get data randomly from ALL into 10 independence subset $\sum_{i=1}^{10} fold(i)$
4: **for** I=1 to 10 **do**
5: **for** J=1 to 10 **do**
6: **If** (I not equal J) **do**
7: Merge $\sum_{i=1}^{10} fold(i)$ when i not equal j, saved into cross(i)
8: **End If**
9: **End for**
10: **End for**
11: % In the end of the process, we get $\sum_{i=1}^{10} cross(i)$ and $\sum_{i=1}^{10} fold(i)$
12: % Back to for loop *times* to get different subset of cross and fold
13: **End for**

The next section will present the performance of the method using cross validation. Experiment using cross validation was repeated 10 times to avoid bias due to the formation of the data splits.

3.10.2 Cross Validation among methods in Machine Learning

Appendix 3 shows the performance of 6 methods: logistic regression, decision trees (gini-index), neural networks (feed-forward), naïve bayes, support vector machines (libSVM) and k-nearest neighbours (based on explicit similarity measures) in the term of c-index. The experiment was conducted 10 (ten) times using 10-fold cross validation in order to avoid bias due when the whole dataset (merging Q1,Q2,Q3,Q4) can be split in the different ways.

For each subsets formed 10-fold cross validation called as $subset_1$, $subset_2$, … $subset_{10}$

One of ten (10) subsets in Appendix 3: $subset_1$, shows in Table 3.14. From the table, we can see that 3 methods: LR, NB, and NN consistently have c-index in around 0.700 – 0.800, even one time each of methods have c-index > 0.800 indicating good discrimination.

Table 3.14. The performance of six (6) methods in subset1 formed 10-fold cross validation.

No.	Fold	c-index					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.756	0.718	0.617	0.710	0.738	0.655
2	Fold2	0.786	0.765	0.615	0.824	0.796	0.712
3	Fold3	0.778	0.781	0.615	0.707	0.759	0.652
4	Fold4	0.806	792	0.701	0.768	0.783	0.685
5	Fold5	0.756	0.688	0.583	0.727	0.756	0.604
6	Fold6	0.788	0.779	0.649	0.758	0.761	0.707
7	Fold7	0.776	0.786	0.620	0.754	0.769	0.629
8	Fold8	0.776	794	0.629	0.768	0.825	0.647
9	Fold9	0.773	0.742	0.562	0.744	0.760	0.607
10	Fold10	0.759	0.722	0.598	0.720	0.759	0.648

LR = logistic regression, DT = decision trees, SVM = support vector machine, NB = naïve bayes, NN = neural networks, KNN = K-nearest neighbours.

In the chapter 1: Introduction, we mention that logistic regression (LR) is widely used to predict clinical outcome and used as standard, it means when some authors looking at an alternative method, they used LR as the base for comparison. By the such reason, we use logistic regression to be compared with other five (5) methods using t-test statistic as discussed in section 2.4.3 to measure the c-index performance.

Table 3.15 shows the results of LR performance to be compared with 5 other methods using t-test statistics. In the repeating ten (10) times 10-fold cross validation, subsets called as $subset_1$, $subset_2$, … $subset_{10}$.

Table 3.15 The performance of LR to be compared with 5 other methods (DT, SVM, NB, NN, KNN) using t-test statistics.

	(95% CI of the difference of c-index mean), p-value				
	DT	**SVM**	**NB**	**NN**	**KNN**
Subset1	(-0.009 to 0.047), 0.18	(0.128 to 0.185), 9.92E-10	(0.00046 to 0.054), 0.047	-0.016 to 0.025), 0.63	(0.092 to 0.15), 4.88E-08
Subset2	(-0.008 to 0.036), 0.22	(0.11 to 0.16), 9.28E-10	(0.008 to 0.049), 0.009	(0.004 to 0.042), 0.020	0.083 to 0.12), 3.85E-09
Subset3	(-0.026 to 0.04), 0.65	(0.12 to 0.17), 7.11E-09	(-0.011 to 0.045), 0.23	(-0.016 to 0.047), 0.32	(0.091 to 0.15), 2.16E-07
Subset4	(0.0015 to 0.041), 0.036	(0.12 to 0.17), 7.06E-10	(0.011 to 0.048), 0.0038	(0.0043 to 0.055), 0.024	(0.10 to 0.14), 2.33E-10
Subset5	(-0.0008 to 0.043), 0.058	(0.13 to 0.17), 1.25E-10	(0.013 to 0.049), 0.002	(0.0004 to 0.055), 0.047	(0.11 to 0.149), 3.30E-10
Subset6	(-0.003 to 0.046), 0.082	(0.11 to 0.18), 1.27E-07	(-0.003 0.044), 0.087	(0.002 to 0.046), 0.032	(0.11 to 0.16), 2.90E-09
Subset7	(0.004 to 0.049), 0.025	(0.12 to 0.18), 5.65E-09	(0.017 to 0.045), 0.0002	(0.009 to 0.048), 0.007	(0.101 to 0.138), 6.48E-11
Subset8	(-0.004 to 0.045), 0.095	(0.110 to 0.170), 8.28E-09	(-0.007 to 0.048), 0.130	(-0.007 to 0.057), 0.112	(0.080 to 0.136), 2.058E-07
Subset9	-0.017 to 0.054), 0.295	0.096 to 0.168 4.066E-07	-0.008 to 0.042), 0.177	-0.015 to 0.060), 0.217	0.100 to 0.156), 1.446E-08
Subset10	0.0021 to 0.054), 0.036	0.113 to 0.160), 3.77E-10	-0.0053 to 0.0370), 0.133	0.0052 to 0.0496), 0.018	0.096 to 0.138), 7.53E-10

The null hypothesis of the experiment is that there is no difference between the results of two methods. The p-value provides fairly strong evidence against the null hypothesis if p-value < 0.05.

LR considered being convincingly good if the p-value obtained is less than 0.05 and 95% CI for the difference between the mean of c-index contains the positive values for both sides. For example, in the Table 3.15, the 95% CI for the difference between the mean of c-index to NB is 0.00046 to 0.054 with a p-

value = 0.047 (<0.05), it means that the p-value provides fairly strong evidence against the null hypothesis and that LR has convincingly get bigger c-index than NB.

In the cross validation experiments were repeated 10 times, compared to other methods, LR is the most superior to KNN and SVM, because all subsets gives p-value < 0.05 indicates that there is strong evidence against the null hypothesis. And those subsets give 95% CI for the difference mean of c-index with the values > 0, its means that in all the cases, the mean of c-index owned by LR greater than KNN and SVM.

While the NB and NN, only 5 out of 10 which produces p-value > 0.05 with 95% CI values mean of c-index greater than 0. While the DT, there are only 3 out of 10 yielding p-value > 0.05 with 95% CI values mean of c-index greater than 0. This means, DT can quite compete with LR to obtain the value c-index.

3.11 The summary of results and overall discussion

This section will summarize our investigation of developing models to predict risk of mortality.

Our stratification model of logistic regression using SPSS is exactly the same as that paper by Prytherch, et.al. (2005) in the term of discrimination. However, in the term of calibration, all of our results to generate logistic regression model from SPSS indicate good calibration, except that our model for the Q4 dataset as χ^2 = 23.36 (p-value =0.0029 < 0.005) indicates there is evidence of significant lack of fit. However, the discrimination for Q4 still has reasonable discrimination (0.758).

There is no effect of changing the type of the data. The model obtained exactly the same result of c-index. Even though they have differences on intercept and also some slopes (only difference on the sign), for those two models we obtained exactly the same area under ROC curve (c-index), therefore we should not worry about categorical data type because SPSS will code it into numerical attributes automatically.

We investigated and modelled system to predict risk of mortality using different tools and methods to gain knowledge of what is the appropriate alternative in the various methods in machine learning that are worth looking at. The logistic regression model produced in SPSS, MATLAB, and RapidMiner have similar result in c-index. However, our experiments that considering decision trees as the potential method to be compared with logistic regression do not always give good results. The CHAID method in SPSS (when compared with logistic regression) had reasonable performance in terms of discrimination. In the opposite, CART method in MATLAB gave an 'overfitted' as a very complex condition when decision trees has large number of nodes trees. However, we can tackle 'overfitting' using pruning and the discrimination is significantly improved. From the experiment, we can see that the performance of a decision trees model not always better than logistic regression. However, decision trees are advantageous as the representation of the tree model is simple enough, intuitive and understandable.

We propose a new measurement (exhaustive method) to assess the performance of the model. From our experiment, the analysis of exhaustive method to assess the performance of model was most likely consistent with c-index. In the opposite, the analysis of calibration using chi-test were not always consistent with c-index. When the discrimination was reasonable and

exhaustive method confirmed with a good result, in some cases there still was evidence of significant lack of fit means calibration performs poor.

Using RapidMiner as a tool, in order to find alternative methods other than decision trees, we used various methods in machine learning: naïve bayes (NB), neural networks (NN), k-nearest neighbours (KNN), support vector machine (SVM). Without tuning parameter of those methods, we found that NN provide a worthwhile result when implemented Q1 as training data into testing datasets (Q2, Q3, Q4). Even the performance to build a model using Q1 give a good discrimination (c-index > 0.800), there are still have a reasonable discrimination when applied to testing datasets (c-index between 0.700 - 0.800).

Further, we used cross validation in order to evaluate various methods in more fair way when all records are tested and they contribute to the overall performance of the algorithm. Intuitively, the suitable method will produce more consistent results and demonstrates stability when the range of c-index obtained is not so wide. From the experiments using 10-cross validation on BHOM dataset, we found that logistic regression and decision trees is a method that gives a pretty good result, following by neural network and naive bayes. Whereas k-nearest neighbours and support vector machine give a poor performance. We noted the drawback of cross validation is that it takes a longer time compared to the other methods because the testing process has to be carried out *K* times for *K*-fold cross-validation.

4. A Structured methodology for developing early warning score using decision trees (DTEWS)

4.1. Introduction

Early identification of vital signs abnormalities or physiological deterioration, and responding to these appropriately, should result in an improvement in clinical outcome (for instance, mortality rate). With few exceptions, the method to assign weighting values of vital sign variables has been derived from expert experience and clinical judgement alone.

In this chapter, we develop a methodology that can be used to devise a new Early Warning Score (EWS) using decision trees applied to individual vital signs recorded from hospital data. Our proposed method DTEWS to develop early warning scores simplifies, speeds up, and radically reduces the effort required to develop the scores.

There are four main things that will be done in this chapter :

1. We describe the previous study by Prytherch, et al. (2010) as our main reference to develop early warning score. We explain about the characteristic of the dataset, the method to develop and the performance.
2. We propose the new methodology to develop early warning score algorithmically using decision trees (DTEWS).
3. We compare the performance of DTEWS and ViEWS using the same dataset and the performance of EWSs assessed using the area under

ROC curve (c-index), early warning score efficiency curve and distribution score.

4. In addition, we extend the implementation of DTEWS using the multiple % (percentage of death), relative risks, different thresholds and different number of risk bands.

In the next section, we will describe previous research into developing early warning scores.

4.2. Previous Study

We use previous study by Prytherch, et al. (2010) as our main reference to develop early warning score. There are two issues regarding this work: the characteristics of the dataset and the method to develop an early warning score.

4.2.1. The characteristics of the dataset

Prytherch, et al. (2010) started with an EWS that included all six of the essential vital signs recommended by The National Institute for Health and Clinical Excellence, 2007). It is recommended that building early warning score should measure 6 (six) vital signs: heart rate, respiratory rate, systolic blood pressure, conscious level, oxygen saturation and temperature.

Prytherch, et al. (2010) also investigated the impact of adding fractional inspired oxygen concentration (SpO2). Therefore, the number of physiological parameters included there are 7 (seven) vital signs. After that, they developed a vital signs database (n = 198,755 observation sets) from clinical data obtained from completed 35,585 consecutive admissions to beds

in the Medical Assessment Unit (MAU) of Portsmouth Hospital between 8 May 2006 and 30 June 2008. Those people who were well enough to be discharged from hospital before midnight on the day of admission were excluded. This generated a database consisting of complete vital signs observation sets from those patients who died in hospital or who stayed in hospital past midnight on their admission day. The dataset was recording vital signs data using the VitalPAC™ device in order to produce an early warning score (EWS).

The demographics and physiological characteristics of the vital signs dataset are shown in Table 4.1 By using a large vital signs dataset (n = 198,755 observation sets) , at hospital discharge, 196756 (98.994%) people admitted were alive and 1999 (1.006%) were dead.

Table 4.1 The characteristics of the patients in the study

Admission: Who died in hospital or who stayed in hospital past midnight on the day of admission	
Numbers (%)	
Male	94376 (47.5)
Female	104379 (52.5)
Total	198755 (100)
Mean age in years	
Male	66.7 (17.9)
Female	69.3 (19.5)
Total	68.1 (18.8)
Hospital mortality (%)	
Male	1.023
Female	1.009
Total	1.006
Vital signs (mean ± S.D.)	
Heart rate (beats min^{-1})	82 (20)
Breathing rate (breaths min^{-1})	17 (4)
Systolic BP (mmHg)	126 (22)
Temperature (^{o}C)	36.6 (0.5)
Vital signs (Frequency, %)	
Conscious level = ALERT	182307 (91.7%)
Conscious level = Not ALERT	16448 (8.3%)
Oxygen saturation (SpO2) = NORMAL	153167 (77.1%)
Oxygen saturation (SpO2) = Not NORMAL	45588 (22.9%)

4.2.2. The method to develop early warning score

In the previous research, Prytherch and colleagues (Prytherch, et al., 2010) identified 33 track and trigger systems from the literature, most of them based on clinical judgement only. Then they devised an early warning score called ViEWS (an early warning score based on the VitalPAC™ dataset). This was built utilising an iterative, realistic 'trial and error' method intentionally being altered to increase its capability to predict internal hospital mortality within 24 hours of a vital signs observation.

In his iterative development process, Prytherch chose seven bands in which to group values of each vital sign. These bands were arbitrarily given weightings of 3, 2, 1, 0, 1, 2, 3 because that is common in other EWS.

Early warning score of ViEWS as follows in Table 4.2:

Table 4.2 ViEWS early warning score by (Prytherch, et al., 2010)

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate	≤8		9-11	12-20		21-24	≥25
S_PO_2	≤91	92-93	94-95	≥96			
Any Supplemental Oxygen?				No			Yes
Temperature (°C)	≤35.0		35.1-36.0	36.1-38.0	38.1-39.0	≥39.1	
Systolic BP (mmHg)	≤90	91-100	101-110	111-249	≥250		
Heart Rate		≤40	41-50	51-90	91-110	111-130	≥131
Conscious Level				Alert (A)			Voice (V) Pain (P) Unresponsive (U)

4.2.3. The performance

The AUROC (95% CI) for ViEWS early warning scores using in-hospital mortality within 24 hours of the observation set was 0.888 (0.880–0.895). The AUROCs (95% CI) for the other 33 aggregate weighted scoring systems (AWTTSs) using this outcome ranged from 0.803 (95% CI: 0.792–0.815) to 0.850 (95% CI: 0.841–0.859) (Prytherch, et al., 2010). From this result, we can conclude that ViEWS early warning scores performed better than the 33 others for all outcomes tested.

Early warning score efficiency curve introduced by (Prytherch, et al., 2010) provides a relative measure of the number of "triggers" that would be generated at different values of EWS. EWS efficiency curve consistently follow the c-index when ViEWS early warning score compared with the worse and the best of other systems in the term of c-index, the curve displayed that ViEWS is the most efficient among them.

4.3. Methodology to generate early warning score

4.3.1. Data used and Description

We use the same dataset as used by Prytherch, et al. (2010) described in section 4.2.1. and named as *vital signs1* dataset. We named *vital signs1* dataset because in the next section we will use two other vital signs datasets that are different to this.

Each vital signs observation set in the database records of vital sign dataset holds the following data:

a) the date/time of observation set,
b) respiration rate,
c) peripheral oxygen saturation (SpO_2).
d) Categorical data to determine whether the patient was breathing air or oxygen.
e) body temperature,
f) systolic and diastolic blood pressure,
g) heart rate,
h) neurological status using the Alert-Verbal-Painful-Unresponsive (AVPU) scale

Now, we need to determine which vital sign variables that we need to take from database records of vital sign dataset and what is the name in the early warning score table in Table 4.2.

We take b) as *Respiratory rate*, c) as $\mathbf{S_pO_2}$, d) as *Any Supplemental Oxygen?* , e) as *Temperature (^oC)*, we only take systolic blood pressure in f) as *Systolic BP (mmHg)*, g) as *Heart Rate*, and h) as *Conscious Level*.

To simplify the issue and facilitate the recording of data in order to avoid mistakes, in this thesis we converted neurological or conscious level assessment into two values - ALERT (alert/responds) and NotALERT (not alert/ does not respond). We also specified oxygen saturation (SpO_2) as NORMAL when the patient was breathing air and NotNORMAL when the patient was breathing any increased level of oxygen.

4.3.2. Assessing performance of a model

There are three (3) things required to assess performance of a model, we use area under ROC curve (AUROC) or c-index, early warning score efficiency curve and distribution score.

The area under ROC curve (AUROC) or c-index has been explained before in Chapter 2. Therefore, in this chapter we will explain early warning score efficiency curve, distribution of the score with associated mortality and distribution score for different age groups.

4.3.2.1. Area under ROC curve (AUROC)

As well as predicting risk of mortality in Chapter 3, we also use the area under ROC curve (AUROC) or c-index to assess the performance of our model.

4.3.2.2. Early warning score efficiency curve

Prytherch, et al. (2010) used an efficiency curve to compare the relative performance of AWTTSs. The EWS efficiency curve provides a relative measure of the number of "triggers" that would be generated at different values of EWS and permits the comparison of the workload generated by each of them.

We take the following figure from Prytherch, et al. (2010).

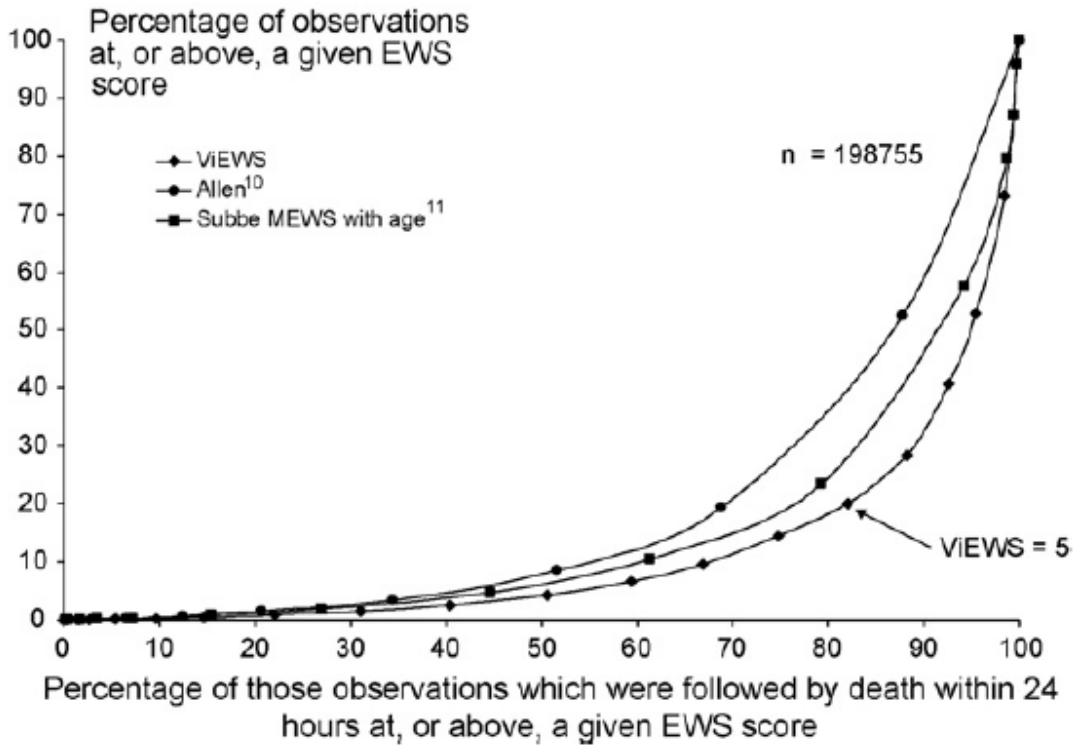


Figure 4.1 Early Warning Score efficiency curve comparison amongs EWS score by (Prytherch, et al., 2010), (Subbe, et al., 2001) and (Allen, 2004)

Figure 4.1 compared EWS efficiency curve performance of VIEWS by Prytherch, et al. (2010) with the best score among 33 AWTTSs by Subbe, et al. (2001) and that of the worst by Allen (2004). From the figure, we can see that VIEWS outperform 33 other of EWS score.

4.3.2.3. Distribution of score with associated mortality

The distribution of scores has a relevance to the percentage of mortality. The existing hypothesis is the higher the score, the higher the percentage of deaths occurring. If the result of EWS score can follow this hypothesis, it can be assured that the EWS score has reliable result.

We take Figure 4.2 from Prytherch, et al. (2010) which shows the distribution of score of their system. We can see that the figure of VIEWS values can follow the hypothesis means VIEWS has reliable result.

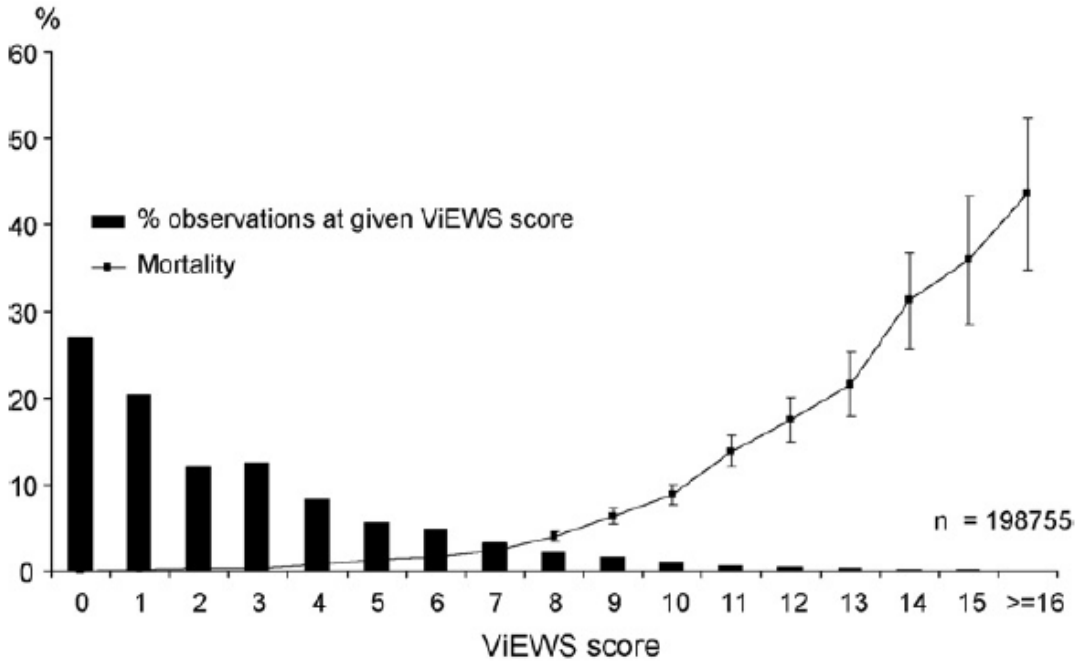


Figure 4.2 The distribution of VIEWS score (Prytherch, et al., 2010) and associated mortality

4.3.2.4. Distribution score for different age groups

Smith and his colleagues examined whether there was a link between the higher values of the EWS score and the older age of the patient (Smith et al., 2008) by showing the distribution scores for different age groups. The general assumption is known that the higher the person's age, the greater the likelihood that risk score had a high value. The model of our system will also be assessed by this assumption.

4.4. A new structured methodology to develop early warning score using decision trees (DTEWS)

In this section, we will describe our methodology to generate Early Warning Scores using decision trees (DTEWS). As described in section 4.3.1, we have 7 vital signs variables to put in the early warning score table (e.g. Table 4.2.) as follows : **Respiratory rate**, **S_pO_2**, **Any Supplemental Oxygen?**, **Temperature (^oC)**, **Systolic BP (mmHg)**, **Heart Rate**, and **Conscious Level.**

DTEWS methodology can be developed using SPSS, MATLAB, R, or any kind of tools/programming. In this section, we choose SPSS to describe the process in DTEWS because of the simplicity.

As there are seven vital signs variables, we need to take vital sign variable, one by one to generate weighting score for each variable.

Decision trees model for heart rate variable can be shown in Figure 4.3.

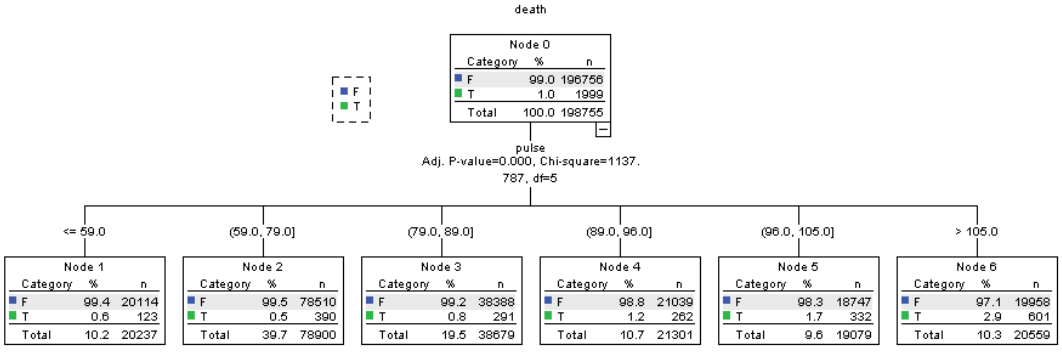


Figure 4.3 Decision trees model for heart rate variable

Each node in the tree is associated with the value in the tree table as follows:

Table 4.3. Tree table for heart rate variable

	F		T		Total		
Node	N	Percent	N	RISK	N	Percent	Split Values
0	196756	99.0%	1999	1.0%	198755	100.0%	
1	20114	99.4%	123	.6%	20237	10.2%	<= 59.0
2	78510	99.5%	390	.5%	78900	39.7%	(59.0, 79.0]
3	38388	99.2%	291	.8%	38679	19.5%	(79.0, 89.0]
4	21039	98.8%	262	1.2%	21301	10.7%	(89.0, 96.0]
5	18747	98.3%	332	1.7%	19079	9.6%	(96.0, 105.0]
6	19958	97.1%	601	2.9%	20559	10.3%	> 105.0

From Table 4.3, split values <=59.0 associated with Node 1 in the trees, there were in total 20237 people, 20114 (99.4%) people admitted were alive and 123 (0.6%) were dead. In the split values (59.0, 79.0), there were in total 78900 people, 78510 (99.5%) people admitted were alive and 390 (0.5%) were dead.

Mapping RISK (in Table 4.3) onto scores is an arbitrary process. We use a simple algorithm as follows :

Algorithm 4.1 Mapping risk onto scores

1: **If** percentage of death < 1
2: **then** score=0
3: **Elseif** percentage of death $\geq$ 1 and percentage of death < 2
4: **then** score=1
5: **Elseif** percentage of death $\geq$ 2 and percentage of death < 3
6: **then** score=2
7: **Elseif** percentage of death $\geq$ 3
8: **then** score=3
9: **End if**

By using Algorithm 4.1. to map risk onto scores, we obtained SCORE column in the tree table as follows :

Table 4.4 Converting percentage of death into the score for heart rate variable

	F		T			Total		
Node	N	Percent	N	Percentage of death	SCORE	N	Percent	Split Values
0	196756	99.0%	1999	1.0%		198755	100.0%	
1	20114	99.4%	123	.6%	0	20237	10.2%	<= 59.0
2	78510	99.5%	390	.5%	0	78900	39.7%	(59.0, 79.0]
3	38388	99.2%	291	.8%	0	38679	19.5%	(79.0, 89.0]
4	21039	98.8%	262	1.2%	1	21301	10.7%	(89.0, 96.0]
5	18747	98.3%	332	1.7%	1	19079	9.6%	(96.0, 105.0]
6	19958	97.1%	601	2.9%	2	20559	10.3%	> 105.0

The next process then clusters groups of similar score together.

Split values : <= 59.0, (59.0, 79.0] , and (79.0, 89.0] clusters into SCORE 0.

Split values : (89.0, 96.0] and (96.0, 105.0] clusters into SCORE 1.

Split values : > 105.0 clusters into SCORE 2.

If we join all of them, we obtain the decision rules as in the following:

If heart rate $\leq$ 89.0 then SCORE=0

Elseif heart rate > 89.0 and heart rate $\leq$ 105.0 then SCORE=1

Elseif heart rate > 105.0 then SCORE=2

The last process is to determine the weighting score for each variable to build early warning system. The following is the weighting scores for the *heart rate* variable, generated from the above decision rules:

Table 4.5 Score for *heart rate* variable

SCORE						
3	2	1	0	1	2	3
			$\leq$ 89	90-105	$\geq$106	

In the same way for other variables in *vital signs1* dataset, we get the early warning score systems as shown in Table 4.6.

Table 4.6 Decision trees SPSS early warning score using *vital signs1* dataset

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate				≤18	19-21		≥22
S_pO_2	≤ 92		93-94	95-99	≥100		
Any Supplemental Oxygen?				No			Yes
Temperature (°C)		≤ 36.2	36.3-36.5	36.6-37.1	≥37.2		
Systolic BP (mmHg)	≤ 99		100-114	≥115			
Heart Rate				≤ 89	90 - 105	≥106	
Conscious Level				Alert (A)			Voice (V) Pain (P) Unresponsive (U)

To apply early warning score system as in Table 4.6, if we have one record in the dataset that has the following attribute values:

respiratory rate = 22, S_pO_2 = 100, any supplemental oxygen=Yes,
temperature = 36.9, systolic BP=114, heart rate = 83,
conscious level =ALERT
Trigger value for that patient is :

score(respiratory rate) + score (S_pO_2) + score(any supplemental oxygen) + score(temperature) + score(systolic BP) + score(heart rate) + score(conscious level) = 3 + 1 + 3 + 0 + 1 + 0 + 0 = 8

Therefore, 8 (eight) is the trigger (total score) for that patient with the data as mentioned above.

The area under ROC curve (c-index) for model obtained from developing early warning score using decision trees in SPSS is 0.888 (95% CI : 0.880 – 0.895). This is exactly the same value with c-index of ViEWS which is 0.888 (95% CI: 0.880 – 0.895).

In the next section, we will describe how to generate early warning score using decision trees in MATLAB.

4.5. Develop early warning score using MATLAB

In this section, we will describe DTEWS methodology to develop an early warning score in MATLAB. MATLAB does not provide tree table as SPSS, therefore we will move all the calculations obtained in SPSS to be implemented in MATLAB. The need to develop DTEWS in MATLAB is because that it allows all processes can be done automatically if it is developed in MATLAB. We will review all of DTEWS process to develop an early warning score as follows:

- Build tree model
- Build tree table based on tree model
- Generate scores from tree table
- Determine the weights for the main vital signs

4.5.1. Illustration of DTEWS methodology

Figure 4.4. summarises the DTEWS process for generating an early warning score for each parameter in the dataset. In this case, the *heart rate (pulse)* variable is used as the example.

Score for *heart rate* variable

SCORE						
3	2	1	0	1	2	3
≤38		39-46	47-89	90-100	≥101	

(3)

If (heart rate <38.5) then SCORE = 3
Elseif (heart rate >=38.5 and heart rate <46.5) then SCORE = 1
Elseif (heart rate >=46.5 and heart rate <89.5) then SCORE = 0
Elseif (heart rate >=89.5 and heart rate <100.5) then SCORE = 1
Elseif (heart rate >=100.5) then SCORE = 2

(1) (2)

Tree table and Generating score for *heart rate* variable

No.	Total record in split values		Total record where death = T		SCORE	Split Values
	N totalinsplit	Percent totalpercent	N diedinsplit	Percent riskpercent		
1	198755	100	1999	1.005		ALL
2	181	0.09	16	8.84	3	<38.5
3	1349	0.67	25	1.85	1	[38.5, 46.5]
4	43833	22.05	191	0.44	0	[46.5, 66.5]
5	24664	12.41	138	0.56	0	[66.5, 72.5]
6	29110	14.65	143	0.49	0	[72.5, 79.5]
7	38679	19.46	291	0.75	0	[79.5, 89.5]
8	31771	15.99	427	1.34	1	[90.5, 100.5]
9	29168	14.68	768	2.63	2	>=100.5

Figure 4.4 Illustration of DTEWS process when it generates EWS for each variable

In the next following section, we will explain in detail each of the five steps in DTEWS.

4.5.2. Decision trees model and generating cut points

As explained in the previous section (section 4.5.1.), determining split values only works on a continuous variable, so the first action is building a decision tree for each continuous variable. There follows the algorithm to generate decision trees from MATLAB using build-in function *classregtree* as in Algorithm 4.2.

Algorithm 4.2 Building decision trees for each such field in the dataset

[1] Input:

x=continuous fields = x1, x2, ..., xn

y=dependent attribute (field DEATH)

vars=name of fields

[2] for each continuous field do the following (step 3 until 5)

[3] % build classification decision trees

t = classregtree(x, y, 'method', 'classification', 'names', vars);

[4] % display the tree

view(t);

The output from this algorithm is a tree, as depicted in Figure 4.5.

Figure 4.5 shows the results of the decision trees generated in MATLAB models for *heart rate* variable. From this figure, we can see the difference between the formations of decision trees that exist in MATLAB (Figure 4.5) with SPSS formations (Figure 4.3).

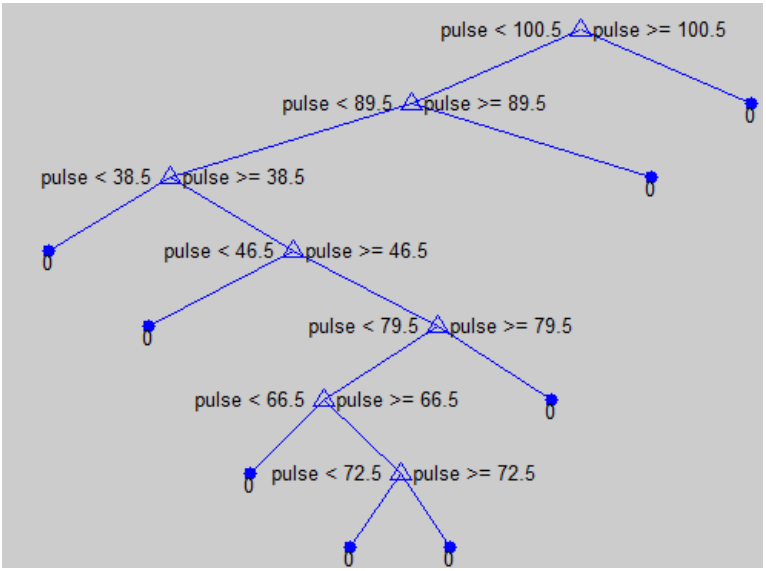


Figure 4.5 Decision trees for heart rate (pulse) variable

As described in Chapter 2, section 2.2.1, decision trees in MATLAB by using CART method will generates binary decision trees, where each node is

divided exactly into two branches. As we see in Figure 4.5, each leaf in the decision trees will split into two branches (so it is called binary).

In case with decision trees in MATLAB, we cannot get the split of values explicitly. Then we have to develop a program to catch cut off points. As shown in Figure 4.5, we obtained the value of cut points as follows: [<38.5, (38.5, 46.5), (46.5, 66.50), (66.50, 72.5), (72.5, 79.5), (79.5, 89.5), (89.5, 100.5), ≥100.5]. The value of the split values then go further into the next process, building tree table.

4.5.3. Building tree table

The reason why we need to develop tree table is because MATLAB doesn't provide it like SPSS. This following algorithm can be used to build a tree-table in MATLAB:

Algorithm 4.3 Algorithm to build tree-table in MATLAB

[1] Input:
x=continuous field
y=dependent attribute (field DEATH)
N=number of records (row) in the dataset
t=split values (t_1,t_2, ..., t_{max}) from lowest to highest
max=number of threshold values
[2] Initialisation
totalinsplit=0;
totalinsplit is the number of records that have specific split values (when x equal one of split values, increment the value of totalinsplit)
diedinsplit=0;
diedinsplit is the number of death in variable totalinsplit
[3] for i = 1 to N do the following
[4] % use all threshold values in x field

```
        if (x(i)< t1) then do
            % adding up number of record that has x field = split
             value, saved into totalinsplit variable
            totalinsplit(1)=totalinsplit(1)+1;
            % calculate number of death in totalinsplit variable
            if (death(i)=1) then diedinsplit(1)=diedinsplit(1)+1;
              elseif (x(i)>=t1 && x(i)<t2) then do
            totalinsplit(2)=totalinsplit(2)+1;
            if (death(i)=1) then diedinsplit(2)=diedinsplit(2)+1;
.....................
          elseif (x(i)>=tmax) then do
            totalinsplit(max)=totalinsplit(max)+1;
            if (death(i)=1) then diedinsplit(max)=diedinsplit(max)+1;
          endif
      endfor
[5] for i = 1 to max do the following
[6]     % calculate the percentage of non survival in each split value
        riskpercent(i)=diedinsplit(i)*100/totalinsplit(i);
[7]     % calculate number of records in split values over the entire dataset
        totalpercent(i)=totalinsplit(i)*100/N;
      endfor
```

Using the same variable (*heart rate* variable) as in the previous step, we obtained the following tree table:

Table 4.7 Tree table for *Heart rate* field

No.	Total record in split values		Total record where death = T		Split Values
	N totalinsplit	Percent Totalpercent	N diedinsplit	Percent Riskpercent	
1	198755	100	1999	1.005	ALL
2	181	0.09	16	8.84	<38.5
3	1349	0.67	25	1.85	[38.5 , 46.5]
4	43833	22.05	191	0.44	[46.5 , 66.5]
5	24664	12.41	138	0.56	[66.5 , 72.5]

No.	Total record in split values		Total record where death = T		Split Values
	N totalinsplit	Percent Totalpercent	N diedinsplit	Percent Riskpercent	
6	29110	14.65	143	0.49	[72.5 , 79.5]
7	38679	19.46	291	0.75	[79.5 , 89.5]
8	31771	15.99	427	1.34	[90.5 , 100.5]
9	29168	14.68	768	2.63	>=100.5

The term of Riskpercent column in Table 4.7 is similar with RISK column in Table 4.3.

4.5.4. Generating Score

In the third step, using the tree table for each field in the dataset, we generate scores based on the risk of death associated with each range of values of the parameter. "Risk" here is the riskpercent column in the tree table.

Mapping risk onto scores is an arbitrary process. We can use simple algorithm in Algorithm 4.1. for that purpose.

We obtained scores from tree table as shown in Table 4.8.

Table 4.8 Generating score for *heart rate* variable

No.	Total record in split values		Total record where death = T		SCORE	Split Values
	N totalinsplit	Percent totalpercent	N diedinsplit	Percent riskpercent		
1	198755	100	1999	1.005		ALL
2	181	0.09	16	8.84	3	<38.5
3	1349	0.67	25	1.85	1	[38.5 , 46.5]
4	43833	22.05	191	0.44	0	[46.5 , 66.5]
5	24664	12.41	138	0.56	0	[66.5 , 72.5]
6	29110	14.65	143	0.49	0	[72.5 , 79.5]
7	38679	19.46	291	0.75	0	[79.5 , 89.5]
8	31771	15.99	427	1.34	1	[90.5 , 100.5]
9	29168	14.68	768	2.63	2	>=100.5

To generate a score from tree table, the first thing to do is to determine the position of the score 0 where the risk was less than 1%. Then from this position, score will grow up consistently to score 1, score 2 until score 3. Using *heart rate* variable as an example, we got risks: 0.44, 0.56, 0.49 and 0.75 for the risk was less than 1%, therefore we can set score of 0 for these risks. Risks 1.34 and 1.85 as a risk equal to or greater than 1% and less than 2%, can be set as score 1. Whereas we can set score of 3 for risks 2.63 and risk 8.84.

Why we need to decide the position of score 0 in the tree table and have to keep the score growing up consistently? Because it is related to the establishment of the rule. An example of inconsistent risk can be exemplified at *temperature* variable in the following table:

Table 4.9 Tree table and generating score for *temperature* variable

No.	Total record in split values		Total record where death = T		SCORE	Split Values
	N totalinsplit	Percent totalpercent	N diedinsplit	Percent riskpercent		
1	198755	100	1999	1.005		ALL
2	13	0.01	0	0.00	3	≤32.1
3	2	0.00	2	100.00	3	[32.2 , 32.30]
4	655	0.33	87	13.28	3	[32.4 , 35.4]
5	1173	0.59	75	6.39	3	[35.5 , 35.8]
6	6530	3.29	183	2.80	2	[35.9 , 36.0]
7	14328	7.21	183	1.28	1	[36.1 , 36.2]
8	29561	14.87	312	1.06	1	[36.3 , 36.4]
9	24686	12.42	225	0.91	0	[36.5 , 36.5]
10	24557	12.36	167	0.68	0	[36.6 , 36.6]
11	25838	13.00	148	0.57	0	[36.7 , 36.7]
12	45551	22.92	317	0.70	0	[36.8 , 37.0]
13	6727	3.38	36	0.54	0	[37.1 , 37.1]
14	15776	7.94	176	1.12	1	[37.2 , 37.9]
15	3358	1.69	88	2.62	2	≥ 38.0

In Table 4.9, we can set risk on 0.91, 0.68, 0.57, 0.70, 0.54 to score 0. From this position, the score will grow up consistently of being bigger to the next level. Cell with marking in red colour in Table 4.9 shown that to be consistent, we set risk 0.00 on split values [≤32.1] to be score 3, not being of score 0.

4.5.5. Determine weighting scores

We continue the process of heart rate variable from generating score into determining weighting score. From Table 4.8, the next process then clusters groups of similar score together.

Split values: <38.5 into SCORE 3

Split values: [38.5 , 46.5] into SCORE 1.

Split values: [46.5,66.5], [66.5,72.5],[72.5,79.5],[79.5,89.5] clusters into SCORE 0.

Split values: [90.5 , 100.5] into SCORE 1

Split values: >= 105.0 into SCORE 2.

If we join all of them, we obtained the decision rules as in the following:

if (heart rate <38.5) then SCORE = 3

Elseif (heart rate >=38.5 and heart rate <46.5) then SCORE = 1

Elseif (heart rate >=46.5 and heart rate <89.5) then SCORE = 0

Elseif (heart rate >=89.5 and heart rate <100.5) then SCORE = 1

Elseif (heart rate >=100.5) then SCORE = 2

The last process is to determine the weighting score for each variable to build early warning system. The following is the weighting scores for the *heart rate* variable, generated from the above decision rules:

Table 4.10 Score for *heart rate* variable in MATLAB

SCORE						
3	2	1	0	1	2	3
≤38		39-46	47-89	90-100	≥101	

4.5.6. Building early warning score system

In the same way with *heart rate* variable, all independent variable in the *vital signs1* dataset establish early warning score as in Table 4.11.

Table 4.11 DTEWS early warning score using *vital signs1* dataset

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate				≤18	19-20	21-24	≥25
S_pO_2	≤89	90-92	93-94	95-99	≥100		
Any Supplemental Oxygen?				No			Yes
Temperature (°C)	≤35.8	35.9-36.0	36.1-36.4	36.5-37.1	37.2-37.9	≥38.0	
Systolic BP (mmHg)	≤89		90-116	117-272			≥273
Heart Rate	≤38		39-46	47-89	90-100	≥101	
Conscious level				Alert (A)			Voice (V) Pain (P) Unresponsive (U)

Our system DTEWS using early warning score in Table 4.11 has c-index=0.889 (95% CI : 0.881-0.896) perform better than ViEWS (c-index=0.888 (95% CI : 0.880–0.895)) in the term of discrimination.

The performance of our model DTEWS compared to ViEWS will be evaluated in the next section (section 4.6).

4.6. Evaluation of DTEWS methodology

In this section, we evaluate the performance of our system DTEWS with VIEWS by Prytherch, et al., (2010) using vital sign dataset (n=198,755) as described in the section 4.3.1. To assess the performance of a model, we use the area under ROC curve, distribution of score with associated mortality, EWS efficiency curve and distribution score for different age groups as described in section 4.3.3.

In evaluating between DTEWS and ViEWS, we posed 4 questions. And if the DTEWS is feasible, then it should give expected answers as in the following table:

Table 4.12 Research question and expected answer

Question	Expected answer
1. Can DTEWS, as a new methodology to generate early warning scores algorithmically, provide a performance as well as or better than an EWS developed using the technique described in the VIEWS paper?	DTEWS early warning scores can provide discrimination (AUROC or c_index) as good as or better than ViEWS early warning scores.
2. Is the EWS efficiency curve of DTEWS acceptable?	The efficiency curve for DTEWS should be similar to that for ViEWS.
3. What is the distribution of the score values of EWS and associated mortality within 24 hours of a given vital signs observation set?	There is a monotonically increasing relationship between the DTEWS score and the risk of death (mortality rate).
4. What is the risk of death by score for each patient age group?	As mortality rates usually increase with patient's age, the group of patients with higher EWS scores should be older, and thus more likely to have a higher mortality rate.

As described previously in section 4.2.3., as our main reference to develop early warning score, (Prytherch, et al., 2010) identified 33 track and trigger systems, most of them based on clinical judgement only. The 33 track and trigger systems have a performance of c-index ranged from 0.803 (0.792–

0.815) to 0.850 (0.841–0.859). Then they devised a new early warning score: ViEWS that performed better than the 33 others with c-index was 0.888 (0.880–0.895). Our system DTEWS has c-index 0.889 (95% CI : 0.881-0.896) perform better than ViEWS in the term of discrimination.

The ability of each model to discriminate between survivors and non-survivors was evaluated using the area under the receiver-operating characteristics (AUROC) curve. We also measured the efficiency of the two models using the EWS efficiency curve and we also evaluate distribution of the score values of EWS.

4.6.1. Comparing score values and the performance

Table 4.13 shows the sensitivity and specificity that performing the area under ROC curve (c-index) for ViEWS and DTEWS. From that table, we plot it into the graphics in Figure 4.6.

Table 4.13 Sensitivity and Specificity that performing ROC curve for ViEWS and DTEWS

ViEWS		DTEWS	
Sensitivity	1-Specificity	Sensitivity	1-Specificity
0.000	0.000	0.000	0.000
0.000	0.001	0.000	0.001
0.000	0.002	0.000	0.003
0.000	0.003	0.000	0.007
0.000	0.014	0.000	0.021
0.000	0.026	0.001	0.038
0.001	0.054	0.001	0.080
0.002	0.096	0.003	0.125
0.004	0.147	0.005	0.196
0.007	0.220	0.010	0.277
0.013	0.310	0.018	0.372
0.023	0.403	0.031	0.457
0.038	0.506	0.051	0.568
0.060	0.593	0.080	0.646
0.091	0.669	0.119	0.722

ViEWS		DTEWS	
Sensitivity	1-Specificity	Sensitivity	1-Specificity
0.139	0.748	0.171	0.800
0.195	0.820	0.242	0.855
0.276	0.882	0.335	0.906
0.402	0.926	0.448	0.942
0.524	0.954	0.601	0.968
0.729	0.984	0.827	0.990
1.000	1.000	1.000	1.000

The ROC curve for the DTEWS and ViEWS together performing AWTTSs using in-hospital mortality within 24 h of the observation set as the outcome is shown in Figure 4.6.

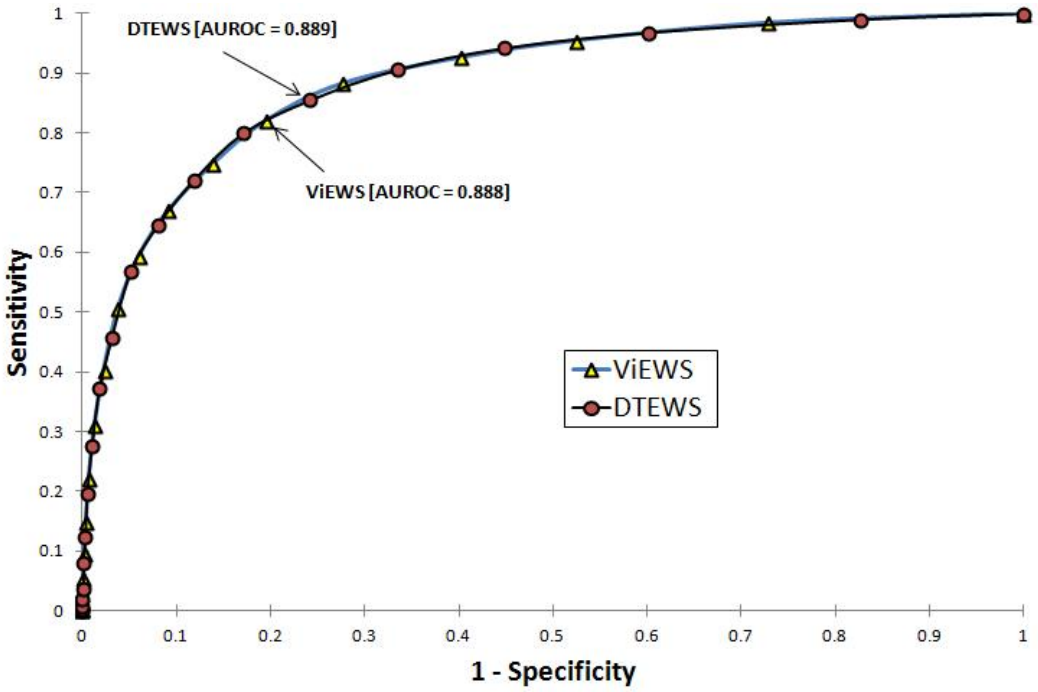


Figure 4.6 Area under ROC curve (c-index) between ViEWS and DTEWS

DTEWS performed better than ViEWS, in terms of discrimination. This is the answer to question 1 in Table 4.12.

4.6.2. Evaluating the efficiency using EWS efficiency curve

(Prytherch, et al., 2010) used an efficiency curve to compare the relative performance of AWTTSs. The EWS efficiency curve for ViEWS and DTEWS in Figure 4.7 provides a relative measure of the number of "triggers" that would be generated at different values of EWS and permits the comparison of the workload generated by both of them. Table 4.14 is formed from that definition.

As an example, we can see from Table 4.14, a ViEWS score of 5 would generate a trigger in 20% of observations, and this would be sufficient to "detect" 82% of all deaths within 24 h of the observation set. To detect the same proportion of deaths would require 1.25 times the workload (25%/20%) if the organisation used DTEWS and this would be sufficient to "detect" 86% of all deaths within 24 h of the observation set.

Table 4.14 EWS Efficiency curve between ViEWS and DTEWS

A=Percentage of those observations which were followed by death within 24 hours at, or above, a given EWS score

B= Percentage of observations at, or above, a given EWS score

	Score	**0**	**1**	**2**	**3**	**4**	**5**	**.....**	**20**
ViEWS	**A**	100	98	95	93	88	82	.....	0.1
	B	100	73	53	41	28	20	.....	0.001
DTEWS	**A**	100	99	97	94	91	86	.....	0.050
	B	100	83	60	45	34	25	.....	0.001

From the Figure 4.7, the EWS efficiency curve of ViEWS and DTEWS performs similarly. This is the answer to question 2 in Table 4.12.

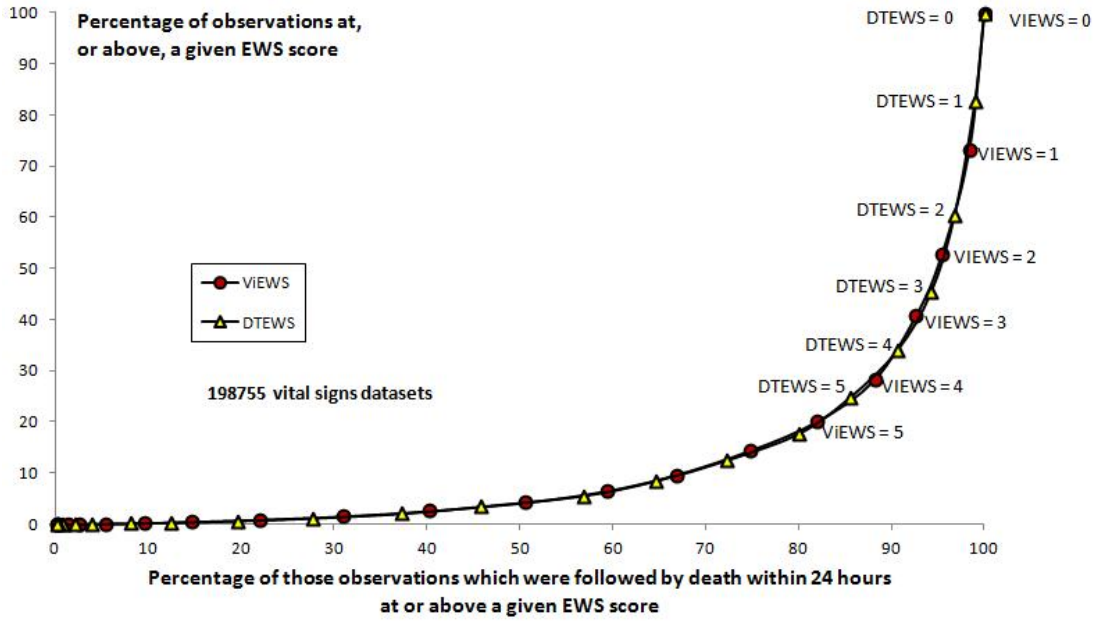


Figure 4.7 the EWS efficiency curves for DTEWS and NEWS using *vital sign1*

4.6.3. Distribution of early warning score and their relationship with mortality

The distribution of EWSs and their relationship with mortality at 24 h post-observation between ViEWS and DTEWS is shown in Figure 4.8.

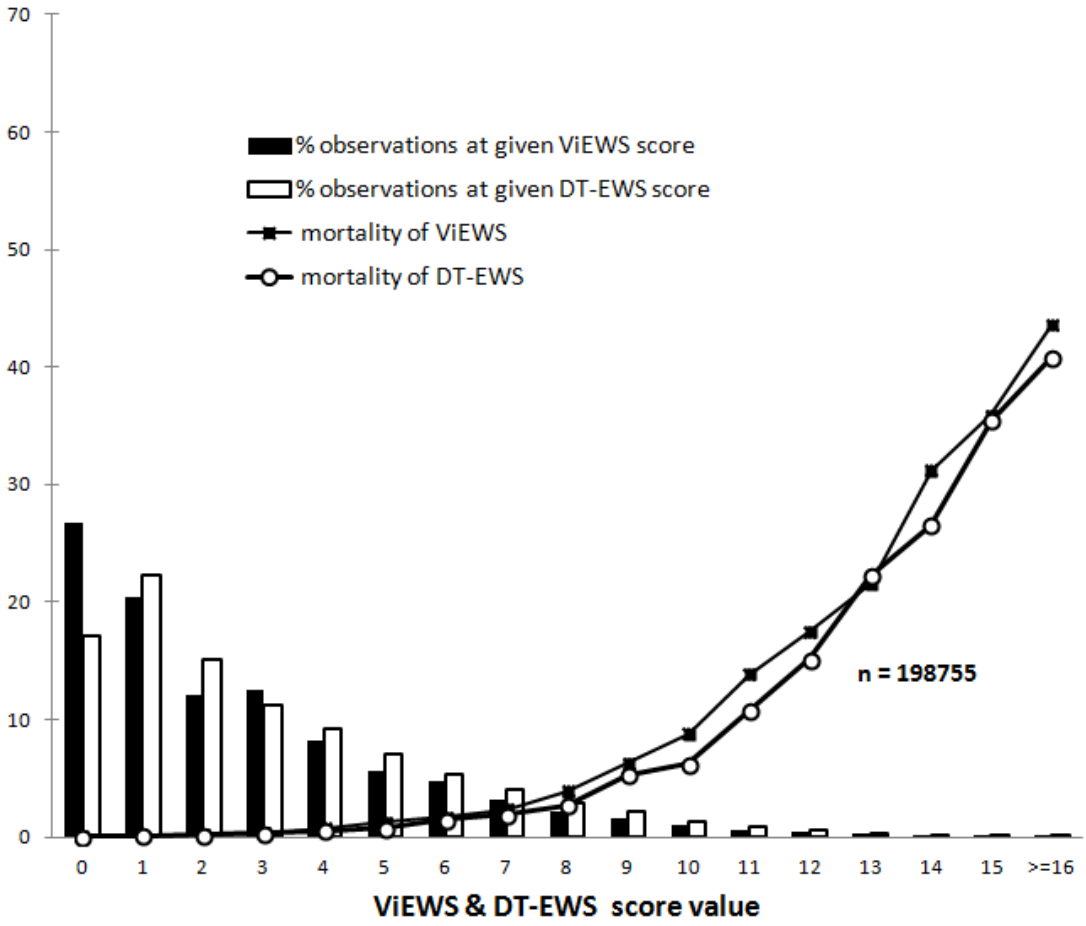


Figure 4.8 Distribution of scores generated by VIEWS and DTEWS and associated mortality within 24h of a given vital signs observation set using *vital signs1* dataset

As seen in Figure 4.8, there is a monotonically increasing relationship between the DTEWS score and the risk of death (mortality rate). This is the answer to question 3 in Table 4.12.

4.6.4. Distribution score for different age groups

As mortality rates are usually related to age, the percentages of likelihood of deaths by score for different age groups were compared. The age groups were determined using the same method that was used from previous research by Smith (Smith et al, 2008) . Using only the integer of patients' ages, vital signs

data were grouped as follows: 16—39; 40—64; 65—79, and ≥80 years. For each age group, we calculated in-hospital mortality across ranges of all physiological variables.

Percentage of deaths by ViEWS and DTEWS score for each group of age is presented in Figure 4.9.

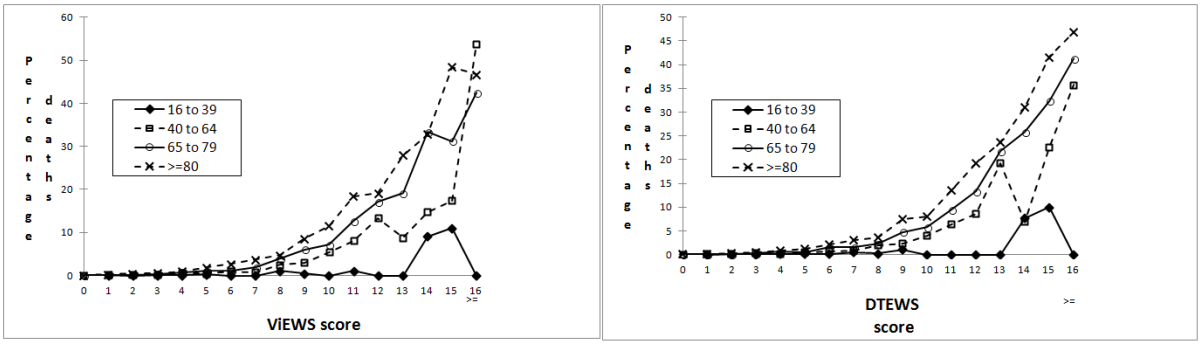


Figure 4.9 Percentage deaths by ViEWS & DTEWS score for each age group

We can see from Figure 4.9, as mortality rates increase with patient's age, the older of the group of patients thus more likely they have a higher mortality rate. This is the answer to question 4 in Table 4.12.

4.7. Extending DTEWS

Our dataset (*vital signs1* dataset) implemented in the previous section is exactly the same data used by (Prytherch, et al., 2010). It had a percentage of death of 1.006% (1999 patients died in total out of 198,755 patients). The value of this percentage (1.006%) is close to 1%, so implementing DTEWS methodology can get satisfactory results when applied using score 0-3.

However, what happens if the percentage value is much higher than 1%, such as 10%, or even slightly different, such as 0.6%? Can the results obtained with the use of score 0-3 give a satisfactory value as well? To investigate this, in

this section we applied the DTEWS methodology to two datasets that have a percentage of death of 0.68% and 10%, called *vitalsign2* and *vitalsign3* dataset, respectively.

4.7.1. Score using multiple % (percentage of death)

Previously, we used *vital signs1* dataset that has 1.006% percentage of death, (i.e. 1999 patients died out of a total of 198,755 patients). In this section, we first discuss experiments using the dataset that has a 0.68% percentage of death (i.e. 406 patients died out of a total of 5937 patients). We named the dataset as *vital signs2* dataset. This value (0.68%) is actual percentage of patients died in the *vital sign2* dataset and this value is less than 1%. Hereafter, we called the term percentage of death as actual percentage.

Once implemented in the DTEWS methodology, we found that there is a fairly significant difference in scores obtained on the different settings of the scoring system between score 0-3 and score using multiple % (percentage of death), as exemplified in the *heart rate* variable in Table 4.15.

Table 4.15 Tree table and generating score for *heart rate* variable using *vital sign2* dataset

No.	Total record in split values		Total record where death = T		Score 0,1,2,3	Score 0, 0.68, 1.36, 2.04	Split values
	N totalinsplit	Percent totalpercent	N diedinsplit	Percent riskpercent			
1	59357	100	406	0.68			ALL
2	5	0.01	1	20.00	3	3	<39.5
3	45	0.08	0	0.00	3	3	[39.5 , 44.5]
4	11	0.02	2	18.18	3	3	[44.5 , 45.5]
5	43680	73.59	200	0.46	0	0	[45.5 , 100.5]
6	12692	21.38	121	0.95	0	1	[100.5 , 119.5]
7	1447	2.44	26	1.80	1	2	[119.5 , 125.5]
8	843	1.42	22	2.61	2	3	[125.5 , 133.5]
9	112	0.19	8	7.14	3	3	[133.5 , 135.5]
10	221	0.37	8	3.62	3	3	[135.5 , 140.5]

No.	Total record in split values		Total record where death = T		Score 0,1,2,3	Score 0, 0.68, 1.36, 2.04	Split values
	N totalin split	Percent totalpercent	N diedin split	Percent riskpercent			
11	42	0.07	5	11.90	3	3	[140.5 , 142.5]
12	20	0.03	0	0.00	3	3	[142.5 , 143.5]
13	50	0.08	5	10.00	3	3	[143.5 , 146.5]
14	163	0.27	5	3.07	3	3	[146.5 , 171]
15	9	0.02	3	33.33	3	3	[171 , 174.5]
16	17	0.03	0	0.00	3	3	>=174.5

As shown in Table 4.15, there are some differences in the two types of this score. For example, when riskpercent = 0.95 on split value [100.5 , 119.5], using actual percentage would categorize this risk as score 1, because 0.95 > 0.68, while using the score (0-3) would categorize this risk as score 0, because 0.95 is less than 1.

The following table describes the differences between score 0-3 and actual score with percentage of death in the dataset equal to 0.68%.

Table 4.16 Different scoring system between score 0-3 and actual score

Score 0-3		Actual score with percentage of death=0.68	
riskpercent	score	riskpercent	score
Riskpercent<1	0	Riskpercent<0.68	0
Riskpercent≥1 and Riskpercent<2	1	Riskpercent≥0.68 and Riskpercent<(2*0.68)	1
Riskpercent≥2 and Riskpercent<3	2	Riskpercent≥(2*0.68) and Riskpercent<(3*0.68)	2
Riskpercent≥3	3	Riskpercent≥(3*0.68)	3

Based on Table 4.16, we can generate weighting score for the *heart rate* variable using score 0-3 and score 0, 0.68, 0.36, 2.04 as follows:

If we did the same as DTEWS and set thresholds at risks of 1,2,3% we would get weighting score as in Table 4.17.

Table 4.17 Weighting score for *heart rate* variable in *vital signs2* dataset using score 0-3

SCORE 0-3						
3	2	1	0	1	2	3
≤45			46-119	120-125	126-133	≥134

However, if instead we choose thresholds at multiples of 0.68% (actual percentage) we get weighting score as in Table 4.18.

Table 4.18 Weighting score for *heart rate* variable in *vital signs2* dataset using multiple % (percentage of death)

SCORE 0, 0.68, 1.36, 2.04						
3	2	1	0	1	2	3
≤45			46-100	101-119	120-125	≥126

Finally, the results obtained for score 0-3 and score using actual percentage (0, 0.68, 1.36, 2.04) for all independent variables are shown in Table 4.19.

Table 4.19 Early warning score of *vital sign2* dataset using score 0,1,2,3 and score using actual percentage

PHYSIOLOGICAL PARAMETERS	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Respiratory rate (#)				**≤21**	**22-31**		**≥32**
Respiratory rate (*)				**≤21**	**22-25**	**26-31**	**≥32**
S_pO_2 (#)	**≤84**	**85-86**	**87-90**	**≥91**			
S_pO_2 (*)	**≤86**		**87-90**	**≥91**			
Any Supplemental Oxygen? (#)				**No**		**Yes**	
Any Supplemental Oxygen? (*)					**No**		**Yes**
Temperature (°C) (#)	**≤35.5**		**35.6-35.9**	**36.0-37.5**	**37.6-39.6**		**≥39.7**
Temperature (°C) (*)	**≤35.5**	**35.6-35.9**		**36.0-37.5**		**37.6-39.6**	**≥39.7**
Systolic BP (mmHg) (#)	**≤80**		**81-100**	**≥100**			
Systolic BP (mmHg) (*)	**≤80**	**81-100**		**≥100**			
Heart Rate (#)	**≤45**			**46-119**	**120-125**	**126-133**	**≥134**
Heart Rate (*)	**≤45**			**46-100**	**101-119**	**120-125**	**≥126**
Conscious Level (#)				**Alert (A)**			**Voice (V) Pain (P) Unresponsive (U)**
Conscious Level (*)				**Alert (A)**			**Voice (V) Pain (P) Unresponsive (U)**

Where : (#) = using score 0-3;
(*) = using score multiple % (percentage of death in the dataset)

We point out the differences between two models in Table 4.19 as follows. The difference between the two scores is the difference in the displacement range for each score. On the *respiratory rate* variable, actual score is more sensitive than score 0-3, as indicated by the definition of more places that fill the blank in score 0, 0.68, 1.36, 2.04 compared to score 0-3. In contrast to SpO_2, even score 0-3 is more sensitive than actual score, with four (4) placements entered in table 5.2 compared with three (3) placements entered in table 5.3 for SpO_2 variable. In general we can say that there is a change in score and displacement range (the score) for all physiological parameters, except for conscious level which is the same for both scores.

Now, we can compare the performance of these models in Table 4.20 using discrimination.

Using different scores will lead to differences in the results obtained. The selection of score 0, 0.68, 1.36, 2.04 referred to the actual score for using the percentage of death in the dataset rather than an arbitrary 0,1,2,3. Not only different result obtained, but also different performance. There is slightly different performance of these models as shown in Table 4.20.

Table 4.20 Different performance of c-index between two different scores using vital sign2 dataset

Score	c-index
DTEWS score 0-3	0.788 (95% CI : 0.761-0.815)
DTEWS score 0.68,1,36,2.04	0.792 (95% CI : 0.766-0.819)

The performance of score 0, 0.68, 1.36, 2.04 is better than the performance of score 0-3. The selection of score 0, 0.68, 1.36, 2.04 referred to the actual score for using multiple % (the percentage of death) in the dataset rather than an arbitrary 0,1,2,3.

In the next experiment, we use *vital signs3* dataset that has a 10% percentage of death (i.e. 5269 patients died out of a total of 55683 patients). This value

(10%) is actual percentage of patients died in the *vital sign3* dataset and this value is much bigger than 1%.

Table 4.21 Tree table and generating score for *heart rate* variable using *vital sign3* dataset using actual percentage

No.	Total record in split values		Total record where death = T		Score 0,10,20,30	Split values
	N Totalinsplit	Percent Totalpercent	N diedinsplit	Percent Riskpercent		
1	55683	100	5269	9.46		ALL
2	2955	5.31	339	11.47	2	<64.5
3	25618	46.01	2147	8.38	2	[64.5 , 90.5]
4	3719	6.68	272	7.31	2	[90.5 , 93.5]
5	13388	24.04	1222	9.13	2	[93.5 , 105.5]
6	6533	11.73	728	11.14	2	[105.5 , 116.5]
7	2811	5.05	418	14.87	2	[116.5 , 131.5]
8	545	0.98	128	23.49	2	[131.5 , 148.5]
9	68	0.12	11	16.18	1	[148.5 , 156.5]
10	27	0.05	0	0.00	0	[156.5 , 169.5]
11	12	0.02	3	25.00	2	[169.5 , 177.5]
12	1	0.00	1	100.00	3	[177.5 , 178.5]
13	6	0.01	0	0.00	3	≥178.5

As shown in Table 4.21, it is impossible to generate score using score 0-3 for *vital sign3* dataset that has 10% of percentage of death. Except risk = 0.00 (row 10), all the risk get score 3, making the rule obtained does not make sense. Using score multiple % (percentage of death) rather than arbitrary score 0-3 can be a solution of this problem.

Generally, score multiple % (the percentage of death) is more reasonable than the score 0-3 in that it better represents the true condition of the data. It is reasonable to assume that we will encounter difficulties in generating a set of rules for the dataset that has a value of 10% percentage of death by applying the score 0,10,20,30.

4.7.2. Using relative risks

In the previous section, generating score using multiple % (percentage of death in the dataset) has fairly significantly improved when using vital sign2 dataset that has 0.68% of percentage of death and encounter the difficulties to generate score when the dataset has actual percentage much higher than 1% when using vital sign3 dataset that has 10% of percentage of death.

In this section, we also use actual percentage (percentage of death in the dataset) to get the relative risk before generating score using arbitrary 0,1,2,3. We got the value of relative risks by dividing Riskpercent column in Table 4.15 by percentage risk of death in the dataset.

To illustrate the process of generating score using relative risks, we use *vital sign3* dataset that has a percentage of death in the dataset of 10%. By using the same variable: *heart rate*, the calculation of relative risks can be shown in the Table 4.22.

Table 4.22 Tree table and generating score for *heart rate* variable using *vital sign3* dataset using relative risks

No.	Total record in split values		Total record where death = T		Relative risk	Score 0 - 3	Split values
	N totalinsplit	Percent totalpercent	N diedinsplit	Percent riskpercent			
1	55683	100	5269	9.46			ALL
2	2955	5.31	339	11.47	1.147	2	<64.5
3	25618	46.01	2147	8.38	0.838	2	[64.5 , 90.5]
4	3719	6.68	272	7.31	0.731	2	[90.5 , 93.5]
5	13388	24.04	1222	9.13	0.913	2	[93.5 , 105.5]
6	6533	11.73	728	11.14	1.114	2	[105.5 , 116.5]
7	2811	5.05	418	14.87	1.487	2	[116.5 , 131.5]
8	545	0.98	128	23.49	2.349	2	[131.5 , 148.5]
9	68	0.12	11	16.18	1.618	1	[148.5 , 156.5]
10	27	0.05	0	0.00	0.000	0	[156.5 , 169.5]
11	12	0.02	3	25.00	2.500	2	[169.5 , 177.5]
12	1	0.00	1	100.00	10.000	3	[177.5 , 178.5]
13	6	0.01	0	0.00	0.000	3	≥178.5

The results from score using multiple % (percentage of death) in Table 4.15 same with the results from score using relative risks in Table 4.22.

For the next section, we are conducting the experiment using multiple % (percentage of death) (section 4.7.1) or relative risks as we found they got a better result rather than using arbitrary score 0-3.

4.7.3. Using different thresholds

There is of course no single score 0-3 system for any set of data - it depends where we choose (arbitrarily) to set the thresholds. When we used vital sign1 dataset as the incidence of death in the dataset is 1.006% - close to 1%, there is a question arises is: What happens if the thresholds of score are closer to percentage of death (e.g. 0, 0.5, 1.0 1.5 or 0, 0.75, 2.25,) ? What happens if the thresholds of score are further apart ? (e.g. 0, 1.5, 3.0 4.5 or 0, 2.0, 4.0, 6.0) ?

The use of different threshold of score can affect the result of early warning score. Table 4.23 shows the use of different threshold of score between score 0,1,2,3 and score 0, 2.0, 4.0, 6.0.

Table 4.23 Different threshold scores of DTEWS on *vital sign1* dataset (using score 0,1,2,3 and score 0, 2, 4, 6)

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate (#)				≤18	19-20	21-24	≥25
Respiratory rate (*)				≤20	21-24	25-28	≥29
S_pO_2 (#)	≤89	90-92	93-94	95-99	≥100		
S_pO_2 (*)	≤89		90-92	≥93			
Any Supplemental Oxygen? (#)				No			Yes
Any Supplemental Oxygen? (*)				No	Yes		
Temperature (^oC) (#)	≤35.8	35.9-	36.1-36.4	36.5-	37.2-	≥38.0	

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
		36.0		37.1	37.9		
Temperature (°C) (*)	≤35.8		35.9-36.0	36.0-37.9	≥38.0		
Systolic BP (mmHg) (#)	≤89		90-116	117-272			≥273
Systolic BP (mmHg) (*)	≤75	76-89		90-272			≥273
Heart Rate (#)	≤38		39-46	47-89	90-100	≥101	
Heart Rate (*)	≤38			39-100	≥101		
Conscious Level (#)				Alert (A)			Voice (V) Pain (P) Unresponsive (U)
Conscious Level (*)				Alert (A)		Voice (V) Pain (P) Unresponsive (U)	

Where : (#) = using score 0-3;
(*) = using score 0, 2, 4, 6

As shown in Table 4.24, in generally, the more threshold values are getting away from percentage of death, the performance will be diminished. For example, we compare between multiple 2% (score 0, 2.0, 4.0, 6.0) as threshold and multiple 0.9% (score 0, 0.9, 1.8, 2.7) as threshold. The value of 2% is getting far away from 1.006% instead of 0.9%, therefore c-index of score multiple 2% (0.872) has less performance rather than c-index of score multiple 0.9% (0.890).

Table 4.24 The performance of early warning score using different threshold

Scores	c-index
Score 0, 0.5, 1.0, 1.5	0.875 (95% CI : 0.867-0.883)
Score 0, 0.75, 1.5, 2.25	0.888 (95% CI : 0.881-0.896)
Score 0, 0.9, 1.8, 2.7	0.890 (95% CI : 0.882-0.897)
Score 0, 1, 2, 3	0.889(95% CI : 0.881-0.896)
Score 0, 1.25, 2.5, 3.75	0.886 (95% CI : 0.878-0.894)
Score 0, 1.5, 3.0, 4.5	0.880 (95% CI : 0.872-0.888)
Score 0, 2.0, 4.0, 6.0	0.872 (95% CI : 0.863-0.880)

4.7.4. Using different number of risk bands

In the preceding experiment, we used risk bands 0, 1, 2, 3 to generate the score. The question that arises is what if the scoring system is simplified by using only a risk band 0-1 or a risk band 0-2? Conversely, what would happen if the score is made more complicated by using a wider range of scores, such as 0-4 or 0-5 instead of using 0-3 as before? To investigate this, we applied the DTEWS methodology using *vital sign1* dataset with percentage of death 1.006% which described in section 4.3.1.

The experiment on early warning score additionally tested an extended range of possible scores, both reduced (0-1, 0-2) and expanded (0-4, 0-5, 0-6).

In this section we describe experiments using simplified early warning scores, score 0-1 results can be seen in Table 4.25 and score 0-2 results can be seen in Table 4.26.

Table 4.25 *vital signs1* dataset, score 0-1

PHYSIOLOGICAL PARAMETERS	**1**	**0**	**1**
Respiratory rate		≤18	≥19
S_pO_2	≤94	95-99	≥100
Any Supplemental Oxygen?		No	Yes
Temperature (°C)	≤36.4	36.5-37.1	≥37.2-
Systolic BP (mmHg)	≤116	117-272	≥273
Diastolic BP(mmHg)	≤60	61-105	≥106
Heart Rate	≤46	47-89	≥90
Conscious Level		Alert (A)	Voice (V) Pain (P) Unresponsive (U)

By using simpler scoring the process carried out will be simplified, which in turn will simplify the calculation of scores, especially if it has to be done manually (e.g. by nursing staff).

Table 4.27 shows the result of scoring using a more complex score, which is score 0-4. The difference between table 5.6 and table 4.2 in the previous chapter is, that some threshold values that are included in the score 3 when using score 0-3, will become part of the score 4 when using score 0-4.

Table 4.26 *vital signs*1 dataset, score 0-2

PHYSIOLOGICAL PARAMETERS	2	1	0	1	2
Respiratory rate			≤18	19-20	≥21
S_pO_2	≤92	93-94	95-99	≥100	
Any Supplemental Oxygen?			No		Yes
Temperature (°C)	≤36.0	36.1-36.4	36.5-37.1	37.2-37.9	≥38.0
Systolic BP (mmHg)	≤89	90-116	117-272		≥273
Diastolic BP(mmHg)	≤54	55-60	61-105	106-140	≥141
Heart Rate	≤38	39-46	47-89	90-100	≥101
Conscious Level			Alert (A)		Voice (V) Pain (P) Unresponsive(U)

The use of a more complex score results in increasingly complex calculations, particularly if the calculation is done manually based on the early warning scores.

Table 4.27 *vital signs1* dataset, score 0-4

PHYSIOLOGICAL PARAMETERS	4	3	2	1	0	1	2	3	4
Respiratory rate					≤18	19-20	21-24		≥25
S_pO_2	≤89		90-92	93-94	95-99	≥100			
Any Supplemental Oxygen?					No			Yes	
Temperature (°C)	≤35.8		35.9-36.0	36.1-36.4	36.5-37.1	37.2-37.9	≥38.0		
Systolic BP (mmHg)	≤89			90 - 116	117-272				>273
Diastolic BP(mmHg)	≤46		47-54	55 - 60	61-105	106-140			≥141
Heart Rate	≤38			39-46	47-89	90-100	≥101		
Conscious Level					Alert (A)				Voice (V) Pain (P) Unresponsive (U)

To analyse the performance of these new forms of early warning score, both simplification and extension of the previous score value (score 0-3), the value of c-index for each range is shown in Table 4.28.

Table 4.28 The performance of different number of risk bands to generate early warning scores using *vital sign1* dataset

Score to be used	Area under ROC curve (C-index)
Score 0-1	0.868
Score 0-2	0.887
Score 0-3	0.889
Score 0-4	0.890
Score 0-5	0.888
Score 0-6	0.888
Score 0-7	0.888

Where the simpler scores are used (score 0-1 and score 0-2), then the performance will decrease (indicated by lower c-index). This is because, with the simple score, there is a merger between the scores' level with a score above these levels thus resulting in a system that is not very sensitive to the scoring of a value.

In the case of score expansion, from score 0-3 to score 0-4, we find that the performance of a larger score (0-4) had a better c-index, even though the difference is very little (from 0.889 to 0.900). This is understandable because the wider score has more detail and consequently has obtained a better c-index. However, we need to examine further whether the expanded scoring system is always guaranteed to get a better c-index.

In the case of score 0-5, the c-index value obtained is 0.888. This value is smaller than the c-index value before the expansion (c-index = 0.889 using score 0-4).

As with the score 0-5, in the score 0-6 the value of the c-index is the same as that was obtained before, of 0.888; in addition, 0.888 was obtained for score 0-

7. This can be explained by the fact that the c-index value reaches the optimal score when we use score 0-4, and then decreases as the scores become higher.

4.8. The summary of results and overall discussion

In this chapter, we develop a new structured methodology that can be used to devise a new early warning score using decision trees (DTEWS). We took the previous study by Prytherch, et al. (2010) as our main reference. Prytherch, et al. (2010) identified 33 track and trigger systems from the literature, most of them based on clinical judgement only. Then they devised an early warning score called ViEWS (an early warning score based on the VitalPAC™ dataset) by using a large vital signs dataset (n = 198,755 observation sets) obtained from completed consecutive admissions to beds in the Medical Assessment Unit (MAU) of Portsmouth Hospital between 8 May 2006 and 30 June 2008. At hospital discharge after midnight on the day of admission, there were 196756 (98.994%) people admitted were alive and 1999 (1.006%) were dead. The AUROC (95% CI) for ViEWS early warning scores using in-hospital mortality within 24 hours of the observation set was 0.888 (0.880–0.895) performed better than 33 other EWSs score ranged from 0.803 (0.792–0.815)10 to 0.850 (0.841–0.859). When evaluated with ViEWS, our structured method DTEWS can provide discrimination c-index=0.889 (95% CI : 0.881-0.896) which is slightly better than ViEWS. Other measurements including the EWS efficiency curve, distribution of scores and distribution of score in age group, also show that DTEWS has as good a performance as ViEWS. We can conlude that our structured methodology DTEWS validates the EWS developed by Prytherch, et.al. (2010).

The first time we developed DTEWS, we arbitrarily chose 1%, 2% and 3% as the risk thresholds. This seemed to work quite well on the vital sign dataset, as the incidence of death in the dataset is 1.006% - close to 1%. We then noticed that those thresholds didn't result in as effective a model when applied to another dataset, vital signs with a percentage of death equal to 0.68%. We observed that by using multiple 0.68% (score 0.68%, 1.32%, 2.04%) as the risk threshold, we can get more reasonable result and get better c-index. The performance of score 0, 0.68, 1.36, 2.04 with c-index = 0.792 (95% CI : 0.766-0.819) is better than the performance of score 0-3 with c-index=0.788 (95% CI : 0.761-0.815).

By using multiple % (percentage of death) as the risk threshold, we also encounter the difficulties in generating a set of rules for the dataset that has a value of 10% percentage of death by applying score 10%, 20%, 30% as the risk threshold. We also noticed, generating a score using multiple % (the percentage of death) and relative risks has exactly the same result. The difference between them is the way they use percentage of death. In relative risks, column riskpercent (in Table 4.22) divided by percentage of death before mapping into the score using arbitrary 0-3, whereas multiple % (percentage of death) using column riskpercent and directly mapping into the score using multiple % (percentage of death) as the risk threshold instead of an arbitrary 0-3.

Further, we investigated different thresholds than percentage of death (mean risk or actual percentage) and found that changing the thresholds a bit did not make much difference.

We also investigated different number of risk bands using score 0-1, 0-2, ... 0-7, and found that, using simpler scoring, the process carried out will be

simplified, which in turn will simplify the calculation of scores, especially if it has to be done manually (e.g. by nursing staff).

From the experiment results in this chapter, we can conclude that DTEWS can be used as a tool to provide early warning scores algorithmically and it clearly involves much less effort.

5. Validating and comparing decision tree early warning score (DTEWS)

5.1. Introduction

We have shown in Chapter 4 that our structured methodology DTEWS validates the ViEWS developed by Prytherch, et al. (2010).

Subsequent to this, the Royal College of Physicians London (RCPL) employed ViEWS as the foundation for the recently publicised NEWS (National Early Warning Score), making slight alterations to the weightings outlined in ViEWS.

Smith, et al. (2013) published a paper in which they show the ability of NEWS to discriminate patients at 4 adverse clinical outcomes. We will use the dataset in this paper to show that DTEWS can validate NEWS.

There are three main things that will be done in this chapter:

1. We want to show that DTEWS as a structured methodology to generate early warning score can validate National Early Warning Score (NEWS) using 4 adverse clinical outcome datasets in a paper by Smith, et al. (2013).

2. DTEWS will be compared with another system based on statistics named as Centile proposed by Tarassenko, et al. (2011) and we want to know the comparison of performance between both systems.

3. We are of the opinion that DTEWS can be applied to another kind of dataset for employment in particular clinical situations. In section 5.3,

DTEWS will be applied to BHOM dataset which was already used in Chapter 3.

5.2. DTEWS validates National Early Warning Score (NEWS)

Prytherch, et al. (2010) developed ViEWS (VitalPAC Early Warning Score) for use in the early recognition and response to patient deterioration. ViEWS was constructed using an iterative, pragmatic, 'trial and error' approach, with the cut-offs for its scoring bands being deliberately adjusted to maximise its ability to predict in-hospital death within 24 hours of a vital signs observation. Subsequent to this, the Royal College of Physicians London (RCPL) employed ViEWS as the foundation for the recently publicised NEWS (National Early Warning Score), making slight alterations to the weightings outlined in ViEWS. Smith, et al. (2013) have thus illustrated the capability of NEWS to discriminate patients at 4 other adverse clinical outcomes as following: at risk of cardiac arrest, unanticipated intensive care unit admission or death within 24 hours, and any of these.

5.2.1. Minor changes between ViEWS and NEWS

Comparing ViEWS as the foundation for the recently publicised NEWS, both of them mostly have the same score with only slight alterations. The following table describe the value of weighting variable to be compared.

Table 5.1 Comparison of early warning score of ViEWS and NEWS

Where : (#) = ViEWS; (*) = NEWS

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate (#)	≤8		9-11	12-20		21-24	≥25
Respiratory rate (*)	≤8		9-11	12-20		21-24	≥25
S_pO_2 (#)	≤91	92 -93	94-95	≥96			
S_pO_2 (*)	≤91	92 -93	94-95	≥96			
Any Supplemental Oxygen? (#)				No			Yes
Any Supplemental Oxygen? (*)				No		Yes	
Temperature (°C) (#)	≤35.0		35.1-36.0	36.1-38.0	38.1-39.0	≥39.1	
Temperature (°C) (*)	≤35.0		35.1-36.0	36.1-38.0	38.1-39.0	≥39.1	
Systolic BP (mmHg) (#)	≤90	91-100	101-110	111-249	≥250		
Systolic BP (mmHg) (*)	≤90	91-100	101-110	111-219			≥220
Heart Rate (#)		≤40	41-50	51-90	91-110	111-130	≥131
Heart Rate (*)	≤40		41-50	51-90	91-110	111-130	≥131
Conscious Level (#)				Alert (A)			Voice (V) Pain (P) Unresponsive (U)
Conscious Level (*)				Alert (A)			Voice (V) Pain (P) Unresponsive (U)

We can point out that there are three differences between two scores:

1. Any Supplemental Oxygen

   ViEWS give score 3 for patients who need any supplemental oxygen, whereas NEWS give score 2 for patients who has the same condition.

2. Systolic BP

   ViEWS give a scores 0 for patients with systolic BP in the range of 111-249 and give a score of 1 for value≥ 250. NEWS give score 0 for patients with systolic BP values within a narrower range 111-219, and give a score of 3 for the value ≥ 220.

3. Heart rate
   For patients who has the value of heart of <=40, ViEWS give a score 2, whereas NEWS give a score 3.

5.2.2. Data used and description

This section will describe 4 other adverse clinical outcome datasets as used in NEWS paper by (Smith, et al., 2013) and it is also will used in this section to show that DTEWS validates NEWS.

The original ViEWS dataset had 198755 observation sets on 35,585 individual patients. 1999 of these observation sets were followed by death within 24 hours of the observation sets - irrespective of any other adverse clinical outcome. This is the data set that was used in the ViEWS paper by Prytherch, et al. (2010).

Later, these 198755 observation sets were matched to other adverse clinical outcomes - specifically cardiac arrest (CA) and unanticipated ICU admission (ICU).

It is possible that any individual observation set is followed by multiple adverse clinical outcomes within 24 hours. So, to facilitate the analysis we defined some precedence rules for the analysis of these extra adverse clinical outcomes, as follows:
(As noted, in all cases the outcomes are within 24 hours of the observation set)

- if CA was followed by ICU admission the outcome is defined to be CA
- if CA was followed by ICU admission and then death, the outcome is defined to be CA
- if CA was followed by death the outcome is defined to be CA
- if ICU was followed by CA (there were none of these as it happens) the outcome is defined to be ICU

- if ICU was followed by CA and then death the outcome would have been defined as ICU
- if ICU was followed by death the outcome is defined to be ICU consequently, outcomes can only be defined as death if they were not preceded by either of CA or ICU.

This gave the other datasets where the outcomes were:

- CA (199 outcomes), named as *CA_PRECEDENCE* dataset
- Unanticipated ICU admission (1161 outcomes) named as *ITU_PRECEDENCE* dataset
- death (with precedence - 1789), named as *DEATH_PRECEDENCE* dataset
- and the final dataset where the outcome was defined to be any of these outcomes - that is CA or ICU or death (within 24 hours, as always). There were 3149 of these, named as *ANY* dataset.

The following table summarises the percentage of outcome of four other adverse clinical outcomes:

Table 5.2 Four others adverse clinical outcomes dataset and the percentage of death

Dataset	Number of patients with outcome in total out of 198,755 patients	Percentage of death in the dataset
ANY	3149	1.58%
DEATH_PRECEDENCE	1789	0.90%
ITU_PRECEDENCE	1161	0.58%
CA_PRECENDENCE	199	0.10%

According to the number of percentage of death for each dataset inTable 5.2, we will use score multiple 1.58% (score 1.58%, 3.16%, 4.74%) for ANY dataset, score multiple 0.90% (score 0.90%, 1.8%, 2.7%) for DEATH_PRECEDENCE dataset, score multiple 0.58% (score 0.58%, 1.16%, 1.74%) for ITU_PRECEDENCE, and score multiple 0.1% (score 0.10%, 0.2%, 0.3%) for CA_PRECEDENCE dataset

5.2.3. Generating score from 4 other adverse clinical outcomes

We develop an early warning score for 4 other adverse clinical outcomes using DTEWS methodology that use score using multiple % percentage of death (section 4.7.1) or relative risks (section 4.7.2).

Table 5.3 shows the early warning score for any of 3 other adverse clinical outcomes.

Table 5.3 Early warning score for any of 3 other adverse clinical outcomes (ANY dataset)

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate	<3		4-11	12-18	19-20	21-24	≥25
S_pO_2	≤89	90-92	93-94	95-99	≥100		
Any Supplemental Oxygen?				No			Yes
Temperature (°C)	≤35.8	35.9-36.0		36.1-37.1	37.2-37.5	37.6-38.9	≥39.0
Systolic BP (mmHg)	≤89		90-116	117-230			≥231
Heart Rate	≤38		39-49	50-89	90-100	≥101	
Conscious level				Alert (A)			Voice (V) Pain (P) Unresponsive (U)

Table 5.4 shows the early warning score for death with precedence that looks similar with EWS for any of 3 other adverse outcomes in Table 5.3. In both EWSs, the new thing that is very different and not found when DTEWS develop EWS in the ViEWS paper is the respiratory rate given score 3 if its value is ≤3.

Table 5.4 Early warning score for death with precedence (DEATH_PRECEDENCE dataset)

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate	≤3			4 -18	19 -20	21 -22	>=23
S_pO_2	≤90	91 -92	93 -94	95 -99	>=100		
Any Supplemental Oxygen?				No			Yes
Temperature (°C)	≤35.8	35.9-36.0	36.1 - 36.4	36.5 - 37.9		38.0-40.0	>=40.1
Systolic BP (mmHg)	≤89	90 - 95	96 -116	>=117			
Heart Rate	≤38		39-46	47-89	90-100	101 - 118	>=119
Conscious level				Alert (A)			Voice (V) Pain (P) Unresponsive (U)

Table 5.5 Early warning score for unanticipated ICU admission precedence (ITU_PRECEDENCE dataset)

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate				≤17	18 -21	22 -24	≥25
S_pO_2	≤89	90-90	91 -93	94 -98	99- 99	≥100	
Any Supplemental Oxygen?				No			Yes
Temperature (°C)	≤35.9		36.0-36.0	36.1-37.3	37.4-37.5		≥37.6
Systolic BP (mmHg)	≤89		90- 118	119 - 239			≥240
Heart Rate				≤95	96 -100	101 - 111	≥112
Conscious level				Alert (A)			Voice (V) Pain (P) Unresponsive (U)

Table 5.5 shows early warning score for unanticipated ICU admission precedence, whereas Table 5.6 shows early warning score for cardiac arrest precedence that has a significant difference with 3 other early warning scores

in the determination of the score value for any supplemental oxygen and concious level variable.

Table 5.6 Early warning score for cardiac arrest precedence (CA_PRECEDENCE dataset)

PHYSIOLOGICAL PARAMETERS	3	2	1	0	1	2	3
Respiratory rate				≤17	18-21	22 - 33	≥34
S_PO_2	≤88		89- 96	≥97			
Any Supplemental Oxygen?				No		Yes	
Temperature (°C)	≤35.8	35.9-36.0		36.1 - 36.4	36.5-39.0		≥39.1
Systolic BP (mmHg)	≤107		108-114	115-171			≥172
Heart Rate	≤38		39-42	43-59		60-128	≥129
Conscious level				Alert (A)	Voice (V) Pain (P) Unresponsive (U)		

5.2.4. Comparing performance amongs EWSs

Further, we compare the performance amongs DTEWS, ViEWS and NEWS. Firstly, we use vital sign dataset in the ViEWS paper described in section 4.3.1. Secondly, we use 4 other adverse clinical outcome datasets as described in section 5.2.1.

5.2.4.1. Using vital sign dataset in the ViEWS paper

The configurations of DTEWS and NEWS were quite comparable, in spite of the different procedures bringing about their development. The AUROC (95% CI) for DTEWS using vital sign1 dataset was 0.889 (0.881-0.896) compared to 0.886 (0.878-0.893) for NEWS, as shown in Figure 5.1.

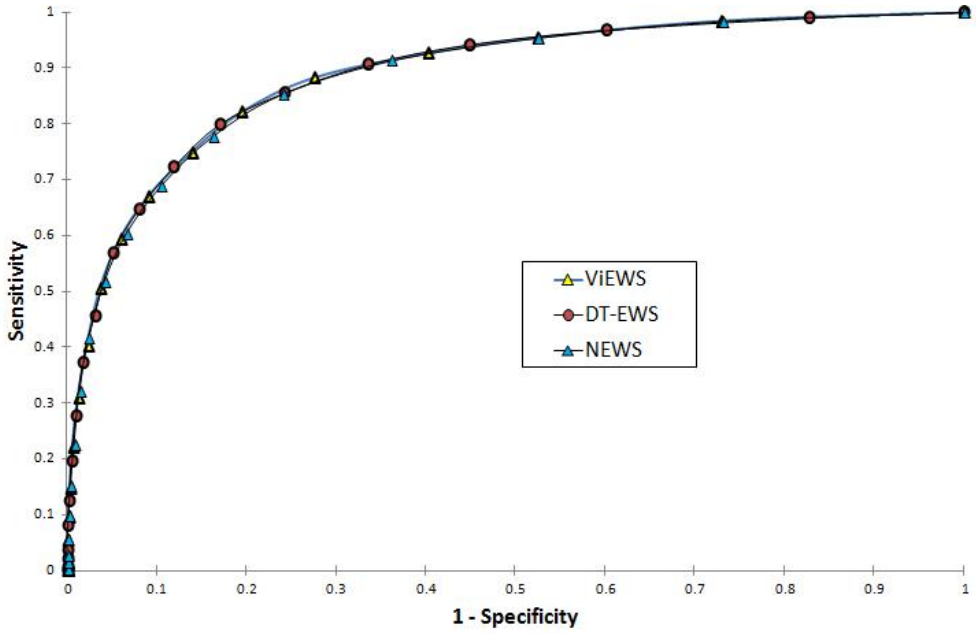


Figure 5.1 The area under ROC curve (c-index) amongs ViEWS, DTEWS and NEWS using vital sign1 dataset

From Table 5.7, we can see the c-index of ViEWS, DTEWS and NEWS are very similar.

Table 5.7 The area under ROC curve (c-index) amongs ViEWS, DTEWS and NEWS using *vital sign1* dataset

EWS Score	The area under ROC curve (c-index)
ViEWS	0.888 (95% CI : 0.880-0.895)
DTEWS	0.889 (95% CI : 0.881-0.896)
NEWS	0.886 (95% CI : 0.878 -0.893)

Figure 5.2 shows distributed of score belong to 3 EWS scores using *vital sign1* dataset. All EWS scores has a distribution of score which shows the association between the higher score with the higher mortality.

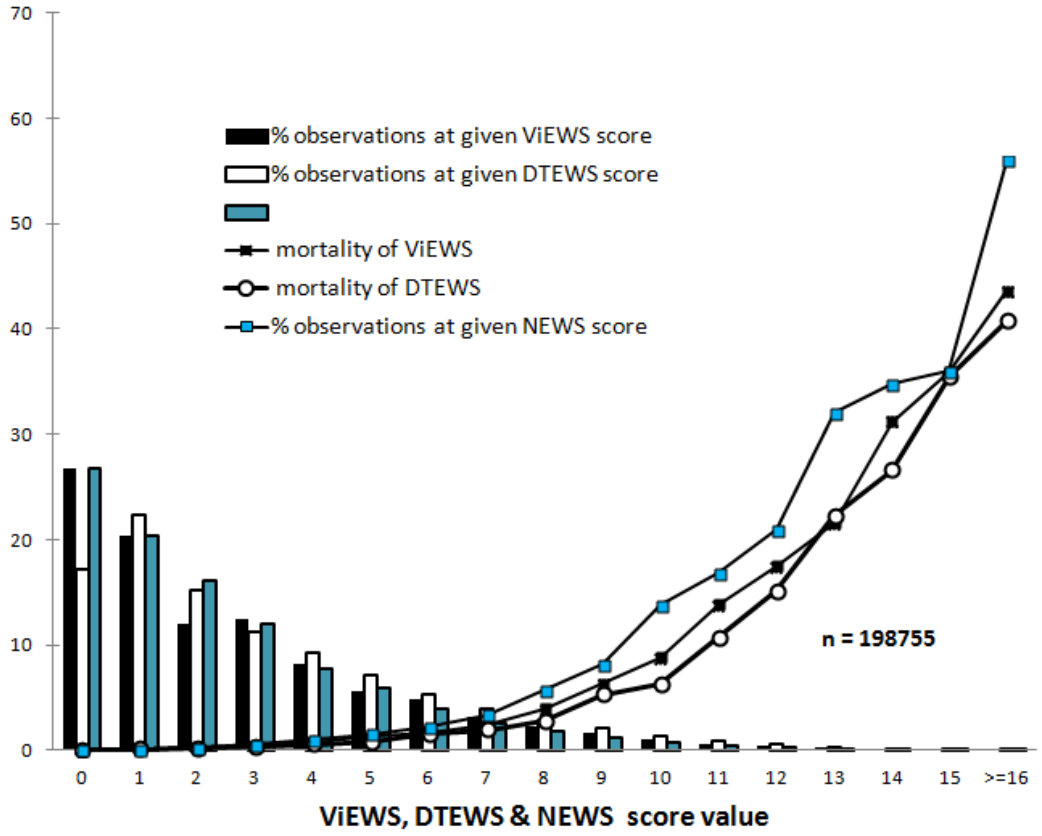


Figure 5.2 Distributed score of ViEWS, DTEWS and NEWS on *vital sign1* dataset

In terms of efficiency, there is a minor difference between DTEWS and NEWS, but both look similar as shown in Figure 5.3.

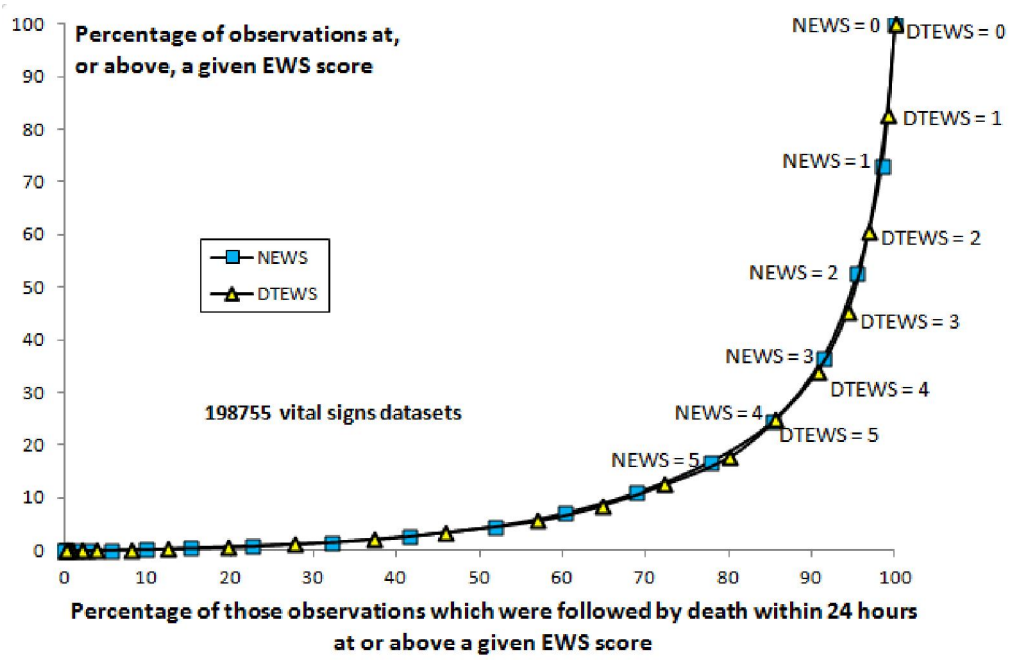


Figure 5.3 EWS efficiency curve between DTEWS and NEWS on *vital sign1* dataset

In Figure 5.4 between ViEWS and NEWS nearly have exactly the same efficiency curve which can be seen in the picture, especially for trigger score 0, 1 and 2.

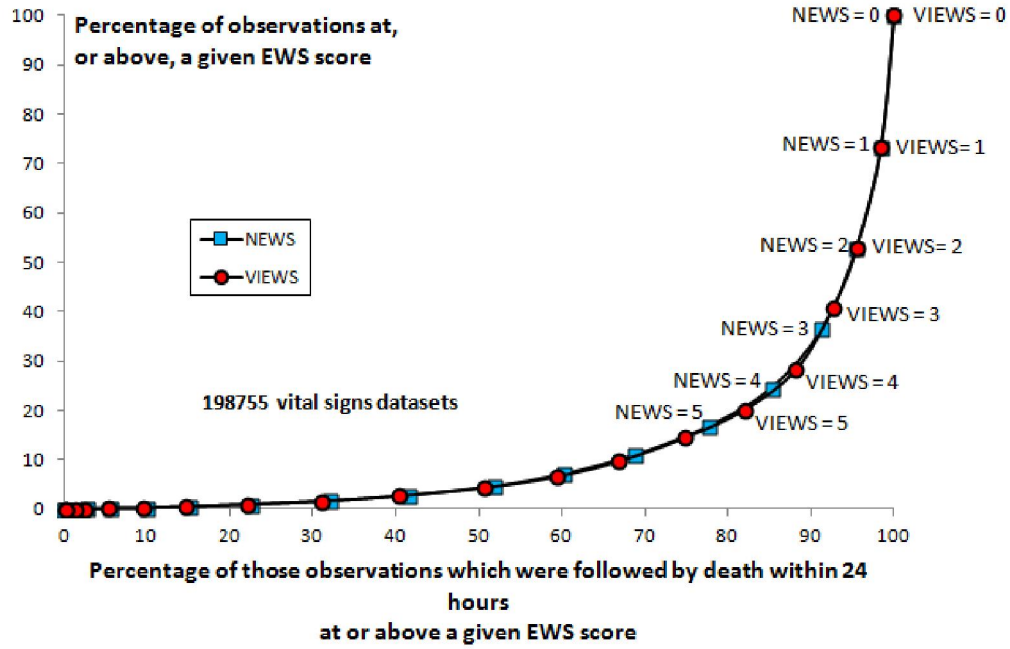


Figure 5.4. EWS efficiency curve between ViEWS and NEWS on *vital sign1* dataset

5.2.4.2. Using 4 other adverse clinical outcomes

Figure 5.5 shows the area under ROC curve belong to 3 EWS scores on 4 other adverse clinical outcomes.

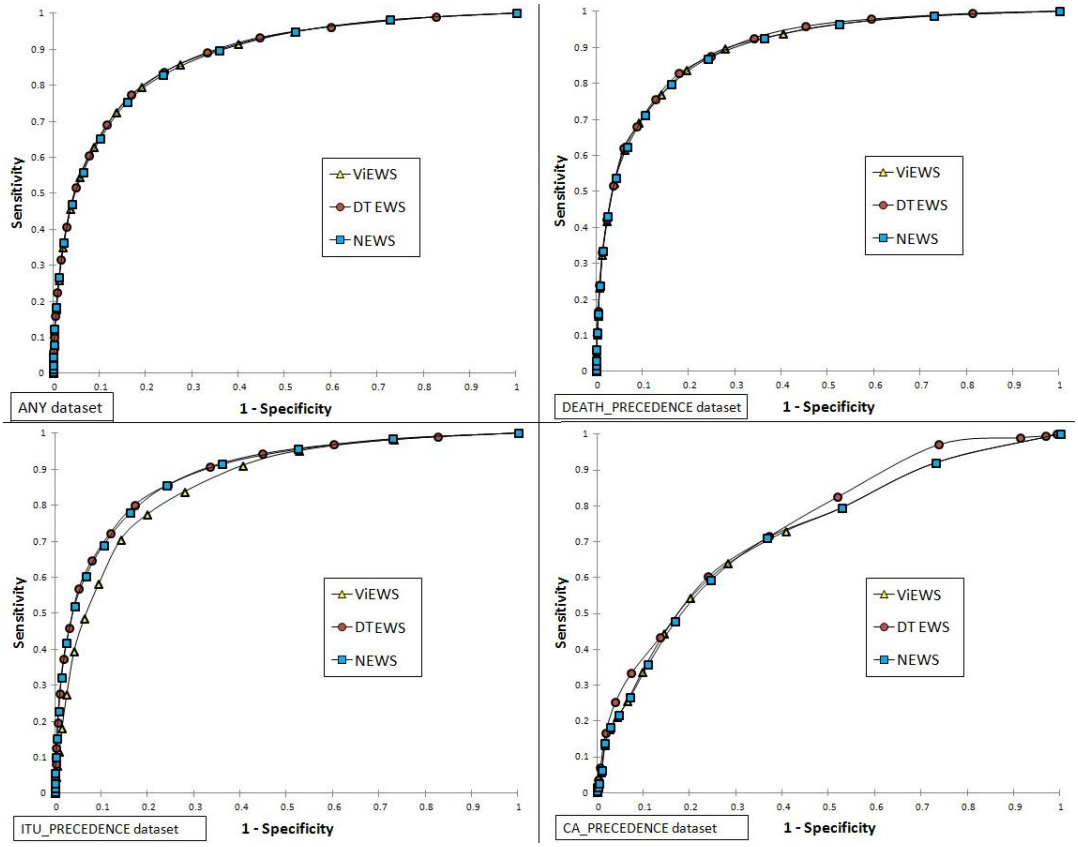


Figure 5.5 The comparison of the area under ROC curve among 3 EWS scores on 4 other adverse clinical outcomes

Table 5.8 below completes the calculation of c-index in Figure 5.5.

Table 5.8 The area under ROC curve for 4 other adverse clinical outcomes amongs 3 EWS scores

EWS Score	The area under ROC curve (c-index)			
	ANY dataset	DEATH_PRECEDENCE dataset	ITU_PRECEDENCE dataset	CA_PRECEDENCE dataset
ViEWS	0.875 (95% CI : 0.869-0.882)	0.897 (95% CI : 0.889-0.904)	0.860 (95% CI : 0.850-0.870)	0.724 (95% CI : 0.687-0.761)
DTEWS	0.877 (95% CI : 0.870-883)	0.900 (95% CI : 0.893-0.908)	0.870 (95% CI : 0.860-0.880)	0.749 (95% CI : 0.715-0.782)
NEWS	0.873 (95% CI : 0.866-0.879)	0.894 (95% CI : 0.887-0.902)	0.857 (95% CI : 0.847-0.868)	0.722 (95% CI : 0.685-0.759)

Distribution of score in Figure 5.6 looks good for any dataset belonging to 3 EWS scores, except in CA_PRECEDENCE not clearly seen due to the small number of records in the dataset.

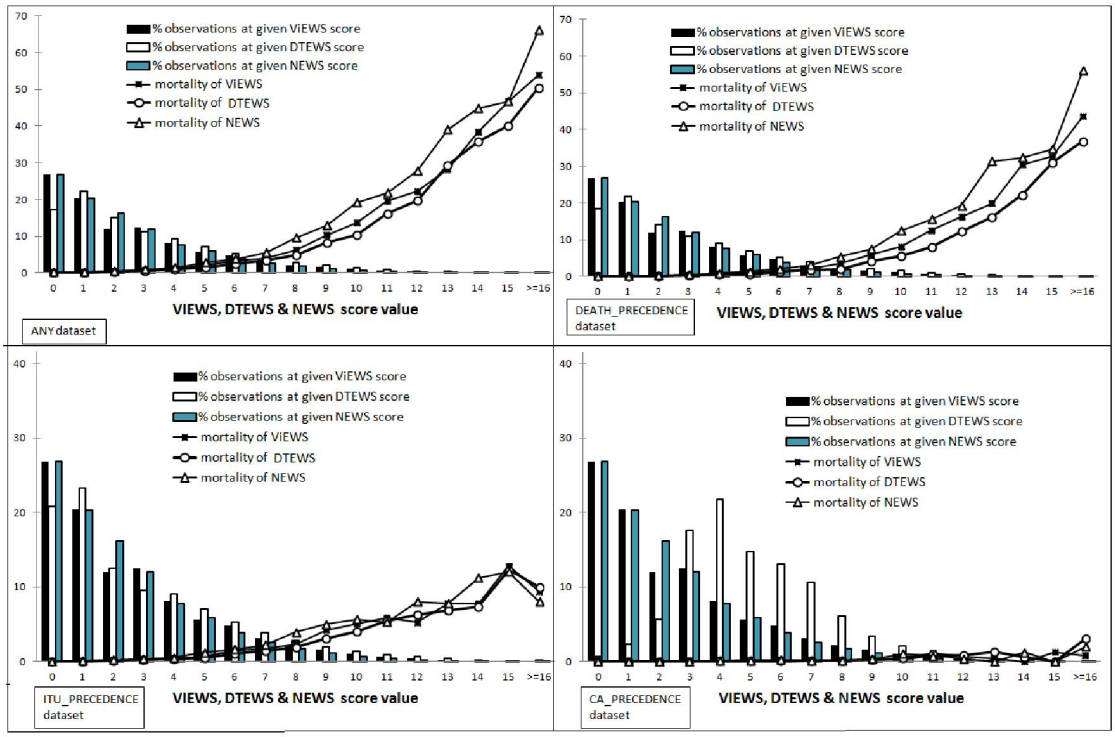


Figure 5.6 Distribution of scores of ViEWS, DTEWS and NEWS on 4 other clinical outcomes dataset

In Figure 5.7, the efficiency amongs 3 EWS scores look similar, except for the ITU_PRECEDENCE and CA_PRECEDENCE shows that DTEWS is more efficient than others.

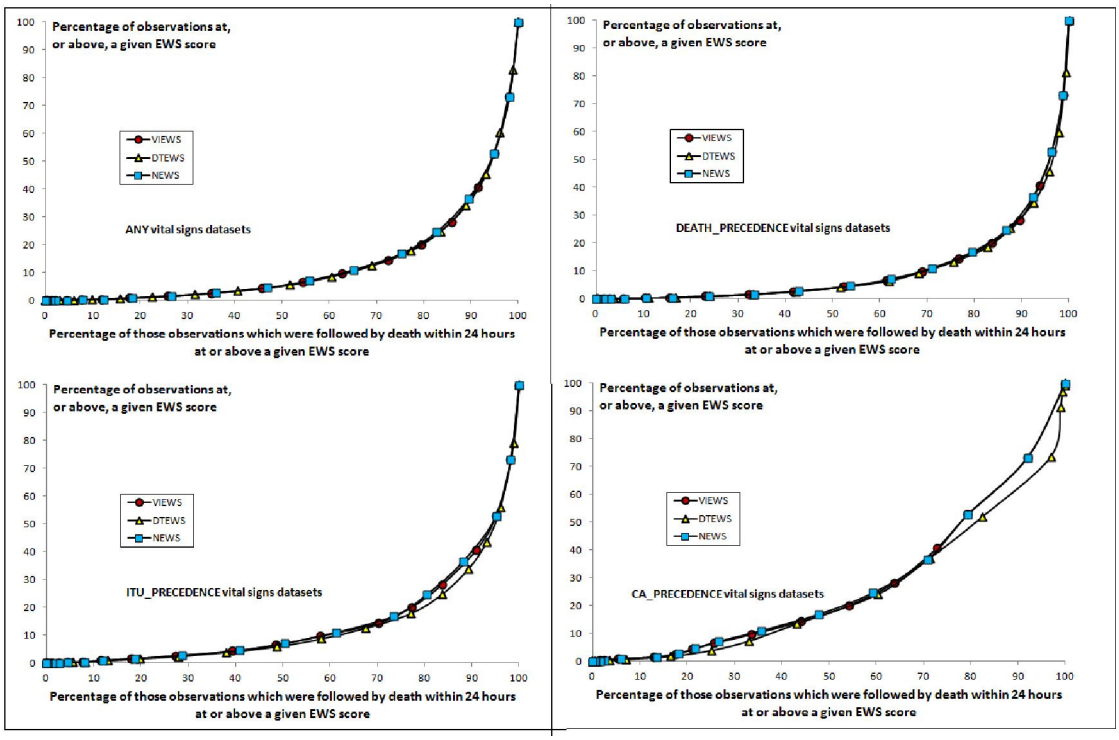


Figure 5.7 EWS efficiency curve of ViEWS, DTEWS and NEWS on 4 other adverse clinical outcome dataset

Based on the result of assessing its performance, DTEWS independently can build the score and do learning on the dataset and quickly provide an almost identical EWS to NEWS.

5.3. Comparing DTEWS with other system based on statistics (Centile)

In chapter 4 and the earliest section in Chapter 5, we discussed developing an early warning score based on clinical judgement (VIEWS and NEWS). And then we compared their performance with our proposed structured methodology DTEWS. In this section, different from clinical judgment and structured methodology as we discussed before, we will discuss an early warning score (EWS) system based on the statistical properties developed by Tarassenko, et al. (2011) called Centile.

In their paper, the authors used a dataset comprising 64,622 hours' worth of continuous vital-sign data, acquired from 863 acutely ill in-hospital patients using bedside monitors. Normalised histograms and cumulative distribution functions were plotted for each physiological variable (heart rate, respiration rate, oxygen saturation and systolic blood pressure). Their system is named Centile due to developing an alerting system based on percentile in statistics.

5.3.1. Generating score of vital sign dataset using Centile

Centile was constructed as follows: an EWS score of 3 was assigned when a vital sign is lower than 1st centile or greater than 99th centile for that variable (in case of double–sided distribution). When a vital sign is between 1st and 5th centile or between the 95th and 99th centile, then this represents score 2. Score 1 refers to the vital sign between 5th and 10th centile or between the 90th and 95th centile.

In the following, we will describe how we generated Centile score from SPSS using *vital sign1* dataset (as described in section 4.3.1.).

After open the dataset file. Click *Descriptive Statistics, Frequencies ...*

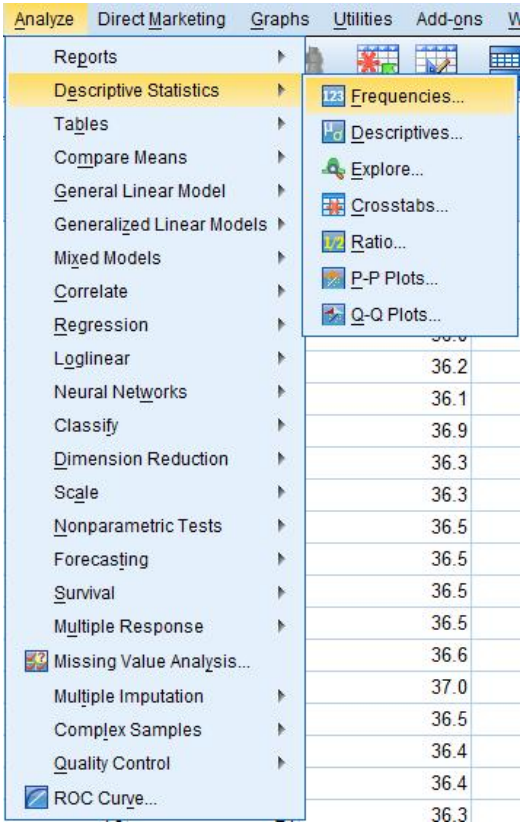


Figure 5.8 Generate Centile score

Select *heart_rate* variable move into the next box. The dialog box should now look like Figure 5.9.

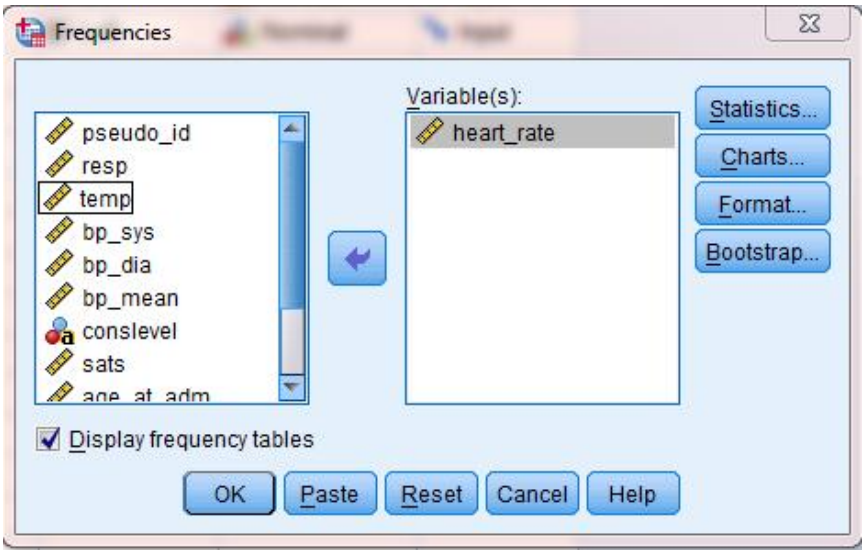


Figure 5.9 Choose heart rate variable as an example

As shown in Figure 5.9, click *Statistics...* button, the following dialog box will open after that:

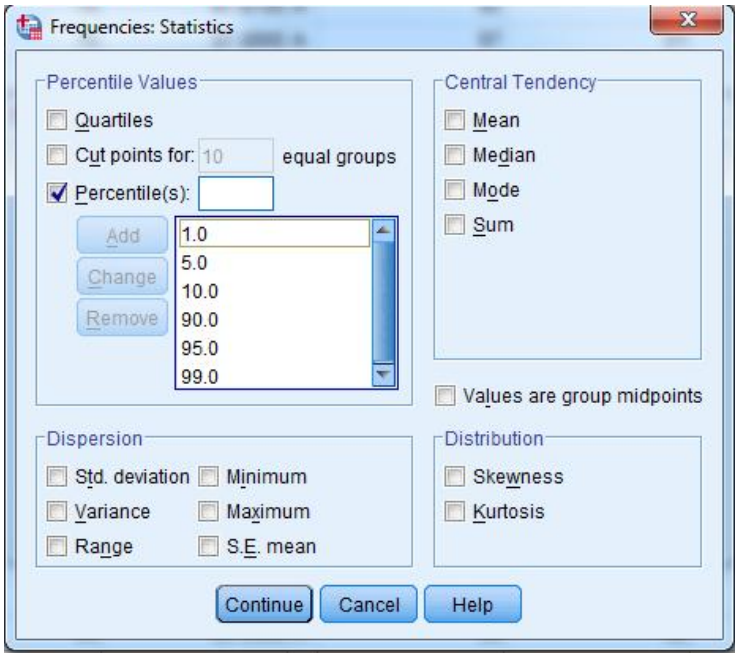


Figure 5.10 Deciding perCentile

Put the following number: 1.0, 99, 0.5, 95, 10, 90 into the *Percentile(s)* box as shown in Figure 5.10.

The result for Percentile should look like in Figure 5.11.

Statistics

PULSE

N	Valid	198755
	Missing	0
Percentiles	1	48.00
	5	55.00
	10	59.00
	90	106.00
	95	115.00
	99	135.00

Figure 5.11 Obtained percentile scores

We interpreted the result in Figure 5.11 to the range score as follows:

3 : for *heart rate*≥135 or *heart rate*≤48

2 : for 47≤*heart rate*≤55 or 115<*heart rate*<=135

1 : for 54≤*heart rate*<59 or 106<*heart rate*<=115

0 : for 60≤*heart rate*≤105

The following is the weighting for the *heart rate* variable from the above range score:

Table 5.9 Weighting scores for *heart rate* variable using Centile

SCORE						
3	2	1	0	1	2	3
≤48	47-55	56-59	60-105	106-114	115-134	≥135

In the same way with *heart rate* variable, all independent variable in the *vital signs1* dataset establish early warning score as in Table 5.10.

Table 5.10 Centile early warning score using *vital signs1* dataset

PHYSIOLOGICAL PARAMETERS	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Respiratory rate	≤12	13	14	15 -20	21-23	24-31	≥32
S_pO_2	≤86	87-90	91-92	93-99		100	≥101
Any Supplemental Oxygen?				**No**			**Yes**
Temperature (°C)	≤35.9	36.0-36.1	36.2	36.3-37.0	37.1-37.3	37.4-38.1	≥38.2
Systolic BP (mmHg)	≤81	82-93	94-100	101-153	154-162	163-188	≥189
Heart Rate	≤48	47-55	56-59	60-105	106-114	115-134	≥ 135
Conscious level				**Alert (A)**			**Voice (V) Pain (P) Unresponsive (U)**

In the following section, we compare the performance of Centile with our structured methodology DTEWS.

5.3.2. Comparing score values between DTEWS and Centile

5.3.2.1. The performance on vital sign with percentage 1,006%

In term of the area under ROC curve, DTEWS WITH c-index = 0.889 (95% CI: 0.881-0.896) outperform Centile with c-index = 0.853 (95% CI: 0.844-0.863), as shown in figure Figure 5.12.

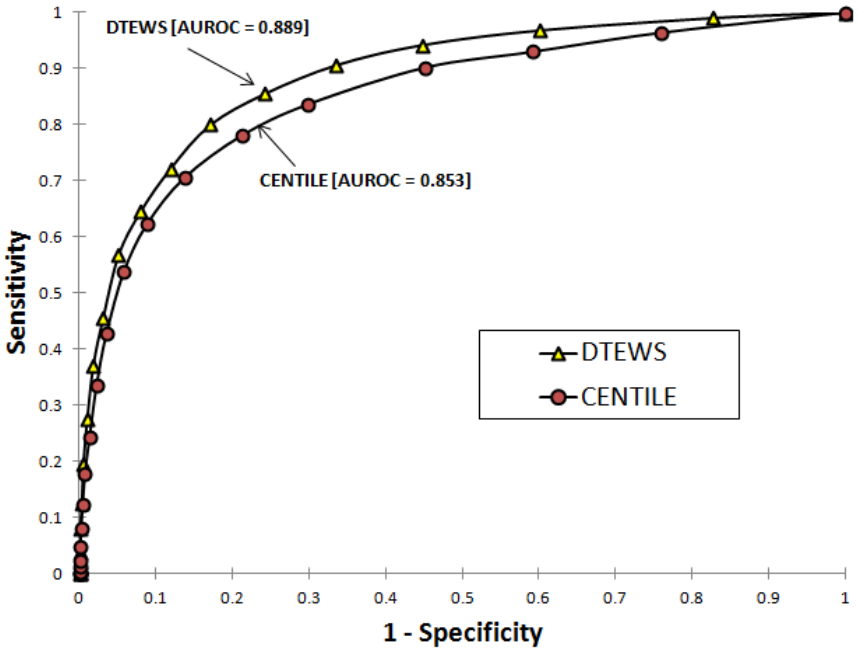


Figure 5.12 Comparison between AUROC of DTEWS and Centile

Having a look at the distribution of scores in Figure 5.13, both early warning scores have the same trend, only it should be noted that the scores of 14 and 15 and also score 16 and 17 in Centile associates with the same percentage of mortality.

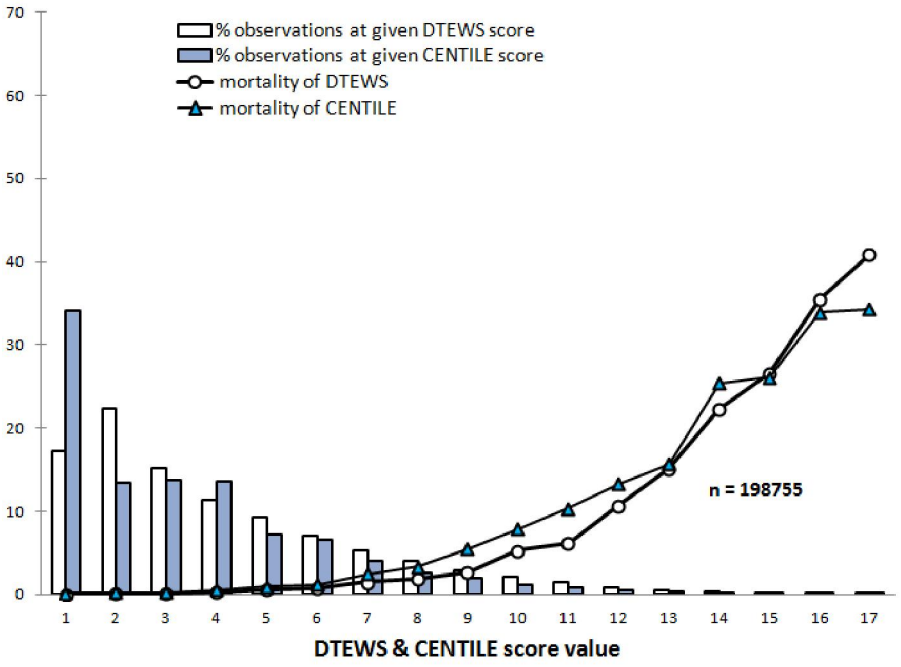


Figure 5.13 Distribution of score between DTEWS and Centile

Whereas in terms of the EWS efficiency curve, DTEWS is clearly more efficient than Centile as shown in Figure 5.14.

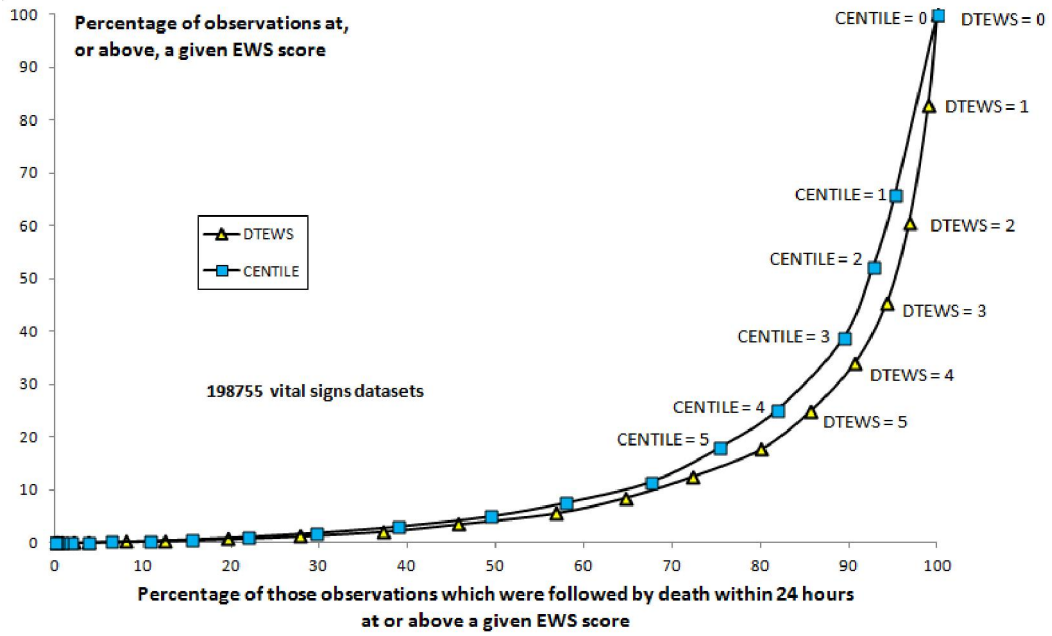


Figure 5.14 Comparison of EWS efficiency curve between DTEWS and Centile

5.3.2.2. The performance on 4 other adverse clinical outcomes

In this section, we are conducting an experiment that compares the performance between DTEWS and Centile using 4 other adverse clinical outcome datasets which are described in section 5.1.1.

In terms of the area under ROC curve, DTEWS outperforms Centile for all datasets as shown in Figure 5.15.

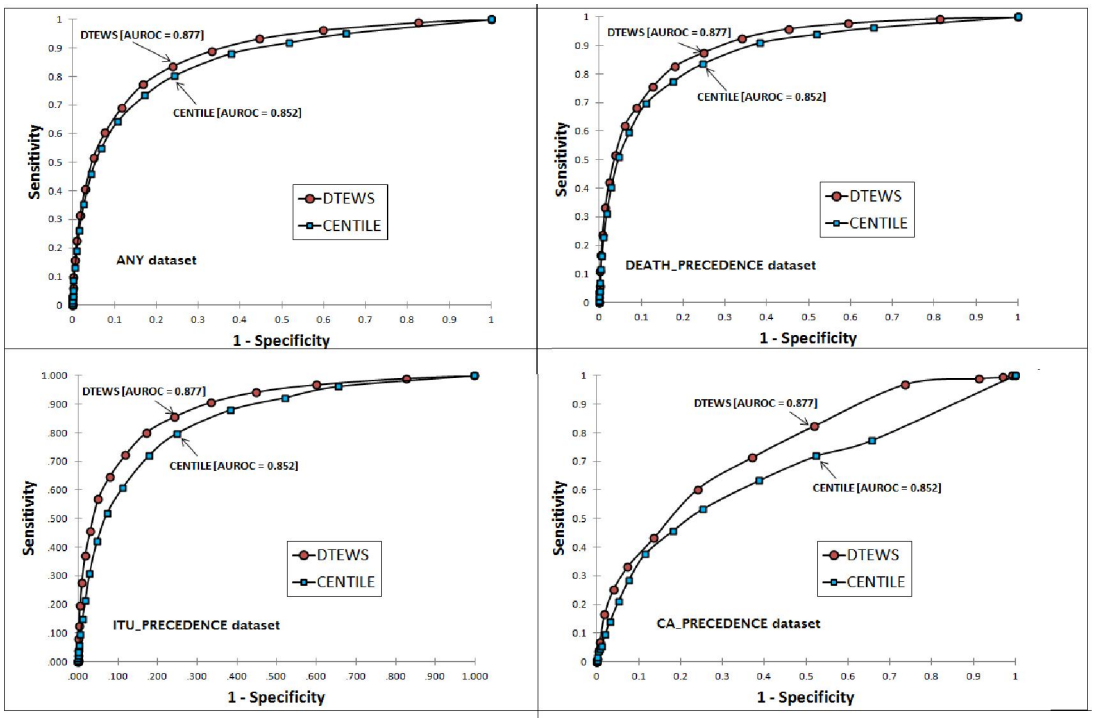


Figure 5.15 Area under ROC curve (c-index) between DTEWS and CENTILE using 4 adverse clinical outcome datasets

The results of c-index between DTEWS and Centile for each dataset can be summarized in Table 5.11:

Table 5.11 The area under ROC curve (c-index) amongs ViEWS, DTEWS and NEWS using 4 other adverse clinical outcomes datasets

Dataset	EWS score			
	VIEWS	DTEWS	NEWS	CENTILE
ANY	0.875 (95% CI : 0.869-0.882)	0.877 (95% CI : 0.870-883)	0.873 (95% CI : 0.866-0.879)	0.852 (95% CI : 0.844 - 0.859)
DEATH_PRECEDENCE	0.897 (95% CI : 0.889-0.904)	0.900 (95% CI : 0.893-0.908)	0.894 (95% CI : 0.887-0.902)	0.872 (95% CI : 0.863 - 0.881)
ITU_PRECEDENCE	0.860 (95% CI : 0.850-0.870)	0.870 (95% CI : 0.860-0.880)	0.857 (95% CI : 0.847-0.868)	0.845 (95% CI : 0.833 - 0.856)
CA_PRECEDENCE	0.724 (95% CI : 0.687-0.761)	0.749 (95% CI : 0.715-0.782)	0.722 (95% CI : 0.685-0.759)	0.662 (95% CI : 0.619-0.706)

Figure 5.16 shows the distribution of scores between DTEWS and Centile. Both early warning score show the relation between distributed of score with mortality, except in CA_PRECENCE dataset due to the very small number of records in the dataset make cannot be clearly delineated.

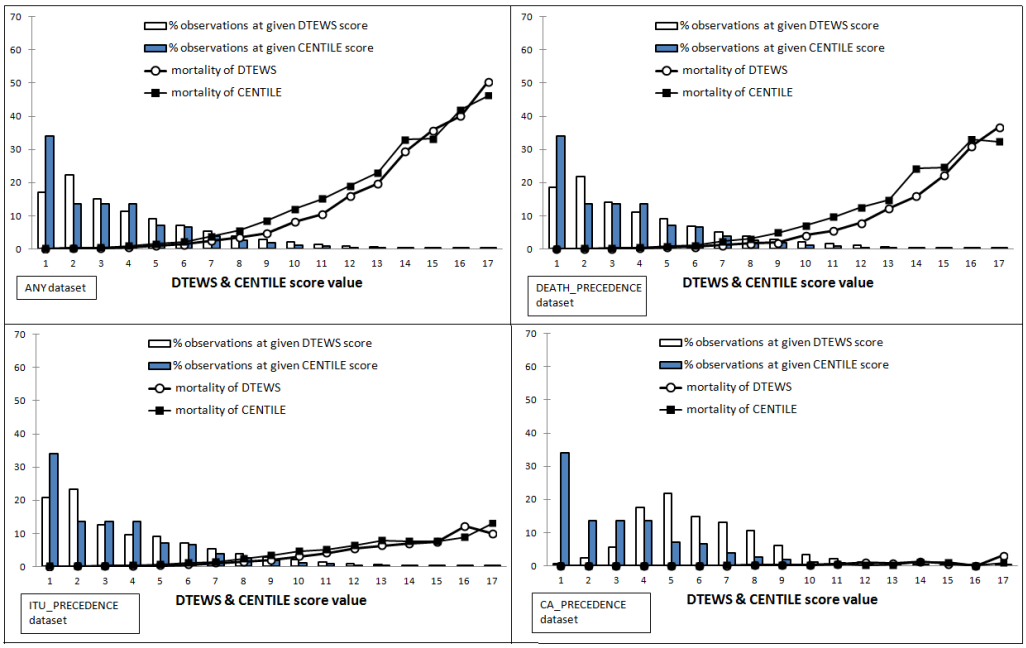


Figure 5.16 Distribution score of DTEWS and CENTILE using 4 other adverse clinical outcome datasets

From Figure 5.17, we can see that in the term of efficiency, based on EWS efficiency curve, DTEWS is more efficient than Centile, even on CA_PRECEDENCE dataset the differences between the two EWS scores are very significant

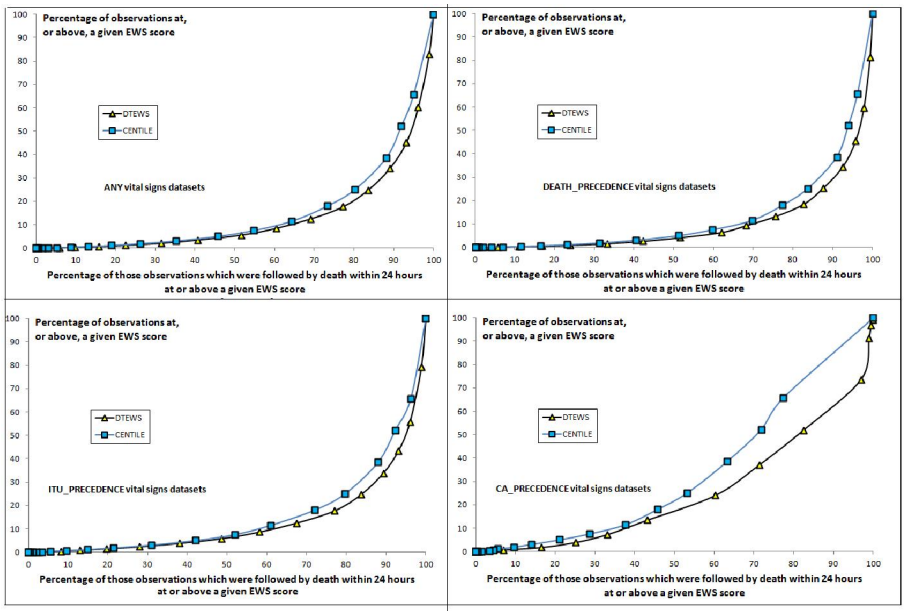


Figure 5.17 EWS efficiency curve between DTEWS and CENTILE using 4 other adverse clinical outcome datasets

5.4. Modelling BHOM dataset using DTEWS methodology

In this section we want to show that DTEWS as a structured methodology to generate an early warning score can also perform well if implemented using a different kind of dataset. Therefore we will build BHOM dataset as used in Chapter 3 (described in section 3.4) using DTEWS methodology. The performance of a model will then be evaluated using the area under ROC curve (c-index).

In the development of an early warning score for BHOM dataset, we use the CHAID method in SPSS and the CART method in MATLAB.

When we build early warning score of BHOM using CART method in MATLAB, we had difficulty in grouping similar observations together.

We take wcc variable as an example, using the CART method in MATLAB obtained the following tree table:

Table 5.12 Generating score for *wcc* variable using CART method

No.	Total record in split values		Total record where death = T		Split Values
	N totalinsplit	Percent totalpercent	N diedinsplit	Percent riskpercent	
1	2257	100	169	7.48	ALL
2	5	0.22	3	60.00	<2.20
3	2197	97.34	147	6.69	[2.2, 22.75]
4	6	0.27	3	50.00	[22.75, 23.30]
5	4	0.18	0	0.00	[23.30, 23.85]
6	1	0.04	1	100.00	[23.85, 23.95]
7	9	0.40	1	11.11	[23.95, 24.90]
8	1	0.04	1	100.00	[24.90. 25.05]
9	9	0.40	4	44.44	[25.05, 26.25]
10	3	0.13	0	0.00	[26.25, 27.50]
11	8	0.35	6	75.00	[27.50, 29.65]
12	10	0.44	2	20.00	[29.65, 41.35,
13	1	0.04	1	100.00	[41.35, 45.45],
14	3	0.13	0	0.00	≥45.45

As we see in the highlight column, we need to look at this column to mapping from riskpercent column into score. Looking at the result of riskpercent column in Figure 5.12, we can see that it will be difficult for us to determine where score 0 will be placed, whether in the split value [23.30, 23.85] or in the split value [26.25, 27.50].

If we use CHAID method in SPSS, it will obtained tree table for wcc variable as shown in Table 5.13 and will be easily for us to convert mean risk in the highlight column into Score column. As the percentage of death in the dataset is 7.5%, therefore we can use score 0, 7.5%, 15%, 22.5% instead of arbitrary score 0-3.

Table 5.13 Generating score for *wcc* variable using CHAID method

T		Total		Split Values	Score 0, 7.5%, 15%, 22.5%
N	Percent	N	Percent		
169	7.5%	2257	100.0%		
58	5.2%	1125	49.8%	<= 9.20000	0
78	8.6%	910	40.3%	(9.20000, 16.60000]	1
33	14.9%	222	9.8%	> 16.60000	3

We obtained early warning score of BHOM dataset as follows in Table 5.14.

Table 5.14. Early warning score for BHOM dataset

PHYSIOLOGICAL PARAMETERS	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Age at admission				<=71.68	71.68 - 85.73	>85.73	
Mode at admission				Elec	Emer		
Hb			<=12	>12			
Wcc				<=9.2	9.2 - 16.6	>16.6	
Urea				<=9.6	>9.6		
Cr				<=156	>256		
Alb			<=37	>37			
Urea to cr				<=0.0748	>0.0748		

We then evaluate the performance of a model using the area under ROC curve (c-index). We obtained c-index for Q1, Q2, Q3 and Q4 datasets as follows:

Table 5.15 Discrimination of BHOM model developed by DTEWS methodology

Dataset	No. of cases	The area under ROC curve (c-index)
Q1	2257	0.751 (95% CI : 0.716-0.785)
Q2	2335	0.756 (95% CI : 0.725-0.788)
Q3	2361	0.747 (95% CI : 0.713-0.782)
Q4	2544	0.709 (95% CI : 0.672-0.746)

All the results are indicating reasonable discrimination with c-index between 0.700 and 0.800. In addition, these results are better compared to the model from original decision trees using CHAID method in SPSS in Table 3.2.

5.5. The summary of results and overall discussion

Royal College of Physicians London (RCPL) employed ViEWS as the foundation for the recently publicised NEWS (National Early Warning Score), making slight alterations to the weightings outlined in ViEWS. By using 4 other adverse clinical outcomes in NEWS paper by Smith 2013, DTEWS performed slightly better than NEWS, in terms of discrimination. And there was similarity between DTEWS and NEWS for other measurements including EWS efficiency curve and distribution of score.

Further if we explore more deeply applying DTEWS on 4 other clinical outcomes, we can see that the validation of DTEWS on cardiac arrest dataset (CA_PRECEDENCE) is unclear, since the number of collected data were too small. However, the performance of result from 3 EWSs score (DTEWS,

ViEWS and NEWS) on CA_PRECEDENCE still gives reasonable discrimination with c-index between 0.700 and 0.800, and DTEWS with c-index = 0.749 outperforms ViEWS (c-index=0.724) and NEWS (c-index=0.722).

It clearly demonstrates that DTEWS methodology can be used to produce a good model if the dataset used for modelling is quite representative. The very small proportion of percentage of death usually could not adequately represent the overall characteristic of the dataset and therefore cannot produce a good model.

Based on the satisfactory results obtained when using 4 other adverse clinical outcomes, we are of the opinion that DTEWS can be employed to promptly generate EWSs for employment in particular clinical situations.

In order to prove this, we conducted the experiment using BHOM dataset and got the satisfactory result. All the results indicate reasonable discrimination with c-index between 0.700 and 0.800. In addition, these results are better compared to the model that was built by the original decision trees model using CHAID method in SPSS in Table 3.2.

As well as comparing DTEWS with early warning scores based on clinical expertise such as ViEWS and NEWS , we also compared it to the system based on statistical properties developed by Tarassenko, et al. (2011), the Centile method. In the experiment, DTEWS outperforms the Centile method for all datasets including the vital sign dataset in ViEWS paper and 4 other adverse clinical outcome datasets in NEWS paper.

6. Overall Discussion and Conclusion

The primary aim of this study was to investigate modelling techniques to predict risk of adverse clinical outcome. To achieve this aim, there are two points we need to accomplish. The first point is how to predict the risk of mortality of patient by categorized risk assessment. And the second point is how to develop an early warning score system that can facilitate a hospital in detecting the situation when the patient's condition needs more serious treatment due to deterioration. Our approach is based on using routinely collected data that are available from hospital computer systems.

6.1. Study Outcome

The following is a summary of the work in this thesis. For the work related to predicting risk of mortality, we follow the research that has been done by Prytherch, et.al. (2005) by using the BHOM dataset obtained from 1 January to 31 December 2001 and divided into four subsets (Q1, Q2, Q3 and Q4). The model generated from the Q1 dataset was then applied to the three other (Q2,Q3,Q4) testing datasets. Our stratification model of logistic regression using SPSS is exactly the same as in that paper by Prytherch, et.al. (2005) in terms of discrimination (section 3.7.4., Table 3.3).

We investigated changing the type of data when logistic regression was used as a method to predict risk of mortality. Our experiment shows that there was no effect from changing the type of the data. Even though the two models have some differences in the intercept and in the sign of attributes, for those two models we obtained exactly the same area under ROC curve (c-index) (section 3.7.3).

We investigated and modelled systems to predict risk of mortality using different tools and methods to gain knowledge of what is the appropriate alternative in the various methods in machine learning that are worth looking at. We first focused on decision trees to be compared with logistic regression. We used different tools: SPSS and MATLAB. The CHAID method in SPSS (when compared with logistic regression) had reasonable performance in terms of discrimination (section 3.7.4, Table 3.3). On the other hand, CART method in MATLAB gave an 'overfitted' model as a very complex condition when decision trees have a large number of nodes. However, we can tackle 'overfitting' using pruning and the discrimination is significantly improved (section 3.8.2.1). From the experiment, we can see that the performance of a decision trees model is not always better than logistic regression. However, decision trees are advantageous as the representation of the tree model is simple enough, intuitive and understandable.

We propose a new measurement (exhaustive method) to assess the performance of the model (section 3.6.3.). From our experiment, the analysis of exhaustive method to assess the performance of model was most likely consistent with c-index (section 3.8.5, Table 3.11 and Table 3.12). Conversely, the analysis of calibration using chi-test was not always consistent with c-index. When the discrimination was reasonable and exhaustive method confirmed with a good result, in some cases there still was evidence of significant lack of fit meaning that calibration performs poorly (section 3.8.5, Table 3.11 and Table 3.12).

In addition to comparing logistic regression (LR) with decision trees (DT), we also compared LR with various machine learning methods including neural networks, naïve bayes, support vector machines and k-nearest neighbours. We used RapidMiner tools to construct an outcome model from various

machine learning methods. Using BHOM dataset, we found that the performance of LR outperform all other methods (section 3.9, Table 3.13). Further, we used cross validation in order to evaluate various methods in a more fair way. From the experiments using 10-cross validation on BHOM dataset, we found that logistic regression and decision trees are methods that give reasonable results, followed by neural network and naive bayes. Whereas k-nearest neighbours and support vector machine give a poor performance (section 3.10.2., Table 3.15).

In the work related with developing early warning score model, we used the previous study by Prytherch, et al. (2010) as our main reference to develop early warning score. We then evaluated the performance of our structured method DTEWS with ViEWS by Prytherch, et al. (2010) using vital sign dataset (n=198,755) as described in section 4.3.1. DTEWS early warning scores can provide discrimination (AUROC or c_index) slightly better than ViEWS (section 4.6.1.). Other measurements including EWS efficiency curve, distribution score and distribution score in different age groups, between DTEWS and ViEWS have similar results (section 4.6.2-4.6.4).

The decision tree process in DTEWS clusters groups of similar observations together. For each cluster we can calculate the risk. Mapping risk onto scores is an arbitrary process. The first time we did it, we arbitrarily chose 1%, 2% and 3% as the risk thresholds. This seemed to work quite well on the vital sign dataset, as the incidence of death in the dataset is 1.006% - close to 1%. We then noticed that those thresholds didn't result in as effective a model when applied to another dataset, vital signs with an average percentage of death equal to 0.68%. We observed that by using multiple 0.68% (score 0.68%, 1.32%, 2.04%) as the risk threshold, we can get more reasonable result and get better c-index. We also encounter the difficulties in generating a set of rules

for the dataset that has a value of 10% percentage of death by applying score 10%, 20%, 30% as the risk threshold (section 4.7.1). We also noticed, doing relative risk also give exactly the same result with choosing score multiple % of death as the risk threshold (section 4.7.2)

Further, we investigated different thresholds than percentage of death (mean risk or actual percentage) and found that changing the thresholds a bit did not make much difference (section 4.7.3). We also investigated different number of risk bands in section 4.7.4. and found that, using simpler scoring, the process carried out will be simplified, which in turn will simplify the calculation of scores, especially if it has to be done manually (e.g. by nursing staff).

The Royal College of Physicians London (RCPL) employed ViEWS as the foundation for the recently publicised NEWS (National Early Warning Score), making slight alterations to the weightings outlined in ViEWS. By using 4 other adverse clinical outcomes in NEWS paper, DTEWS performed slightly better than NEWS, in terms of discrimination. And there was similarity between DTEWS and NEWS for other measurements including EWS efficiency curve and distribution of score (section 5.1).

As well as comparing DTEWS with early warning scores based on clinical expertise, such as ViEWS and NEWS, we also compared it to the system based on statistical properties developed by Tarassenko, et al. (2011), the Centile method. In the experiment, DTEWS outperforms the Centile method for all datasets including the vital sign dataset in ViEWS paper and 4 other adverse clinical outcome dataset in NEWS paper (section 5.2).

Based on the satisfactory results obtained when using 4 other adverse clinical outcomes, we are of the opinion that DTEWS can be employed to promptly generate EWSs for employment in particular clinical situations. In order to

prove this, we conducted the experiment using BHOM dataset and achieved a satisfactory result (section 5.3).

6.2. Original Contribution to Knowledge and limitation of the study

The original contribution to knowledge of this work is:

1. We have shown that an early warning score can be developed using a decision tree algorithm, coupled with a pragmatic selection of risk thresholds based on multiples of the overall risk of outcome.

2. Secondly, we have shown that an EWS developed by such a mechanism is as effective as those developed by Prytherch, et.al. (2010) via brute force, and the EWS adopted by the RCP that was an adaptation of that. A structured methodology, DTEWS, can provide discrimination (c-index) as good as ViEWS. Other measurements including the EWS efficiency curve, distribution of scores and distribution of score in age group, also show that DTEWS has asgood performance as ViEWS. We can conclude that our structured methodology DTEWS validates the EWS developed by Prytherch, et.al. (2010).

3. The RCP employed ViEWS as the foundation for the recently publicised NEWS (National Early Warning Score), making slight alterations to the weightings outlined in ViEWS. By using 4 other clinical adverse outcome dataset in NEWS, we show that DTEWS also provides a similar score to NEWS and gives a performance as good as

NEWS. We can conclude that DTEWS validates the design characteristics of the National Early Warning Score (NEWS).

The study had some limitations regarding the task for developing risk of mortality and also developing early warning scores.

1. When we explored the use of various machine learning techniques to be compared with logistic regression, we didn't use tuning parameter to get the optimization of method which can give the best result. We only used default parameter for methods in machine learning that we used to be compared with logistic regression. By using tuning parameter, the result could have been better. However, this kind of work needs the design of the algorithm to tune parameter so that the tuning process can work effectively.

2. DTEWS had some limitations most of which concerned the representation of the dataset. Validation by clinical experts is required to confirm and revise a score that has been generated by this methodology. It is obviously important to make sure that the dataset used to generate DTEWS model is capable and representative enough to get weighting score for each vital sign variable.

6.3. Reflection on the Results in Clinical Context

To improve outcomes, we need to develop prediction models that can be used to facilitate clinicians in identifying patients at high or low risk of mortality. We hope that the EWS systems we have discussed here can make it easier for patients to get timely appropriate intervention. This will deliver better care to patients, which ultimately is the main purpose that we want to achieve in this thesis.

6.4. Suggestions for future work

In the research for establishing early warning score, some studies stated that in all studies the patients were monitored intermittently and it could occur that a patient displaying normal vital sign, during the intermittent appraisal displayed considerable abnormalities (DeVita et al., 2010; Smith, Prytherch, Schmidt, & Featherstone, 2008). There is no existing research that has explored the changes in value of vital signs between the various subsequent measurements. However, this motivates us to carry out further studies concerning this.

In this thesis, we focus on decision trees as a base method to predict risk of mortality and early warning score model. The reason to choose decision trees is due to the logic of the modelling results. When people need to make a decision, they then compose a number of rules to solve the problem. Apart from decision trees method, there are still a lot of methods that could be used in machine learning to predict adverse clinical outcome. Their result could improve what is already provided by decision trees.

6.5. Overall conclusion

The results of this study support the idea that decision trees are one of the methods in computer science that can be applied to problems in the medical area.

When we produced a model for risk of mortality, we showed that decision trees model have reasonable discrimination and could be considered as an alternative technique to logistic regression. Secondly, we have shown that a structured methodology using decision trees to develop early warning score

confirms previous findings and contributes additional evidence that suggests an algorithmical method to produce EWS system. We would expect that the outcome of these works would be acknowledged, and promote more extensive employment of machine learning techniques in particular types of medical purpose.

Appendices

Appendix 1. Honorary Contract from Portsmouth Hospital NHS trust

Portsmouth Hospitals NHS
NHS Trust

Wednesday, 25 November 2009

Employee Resourcing Department
The Old Gymnasium Building
Fort Southwick (NHS)
James Callaghan Drive
Fareham
Hampshire
PO17 6AR

Tel: 02392 322000
Fax: 02392322078
Hannah.Stevens@porthosp.nhs.uk

PRIVATE & CONFIDENTIAL
Tessy Badriyah
204 Burgess Road
Southampton
Hampshire
SO16 3AY

Dear Tessy

Re: Arrangements for placement at Portsmouth Hospitals NHS Trust

I am writing to confirm the details of your placement and arrangements for undertaking duties at Portsmouth Hospitals NHS Trust.

The arrangements set out in this letter will begin on 1 November 2009 ("the placement") and are subject to the following.

1. Unless terminated sooner, you will undertake the placement until 31 October 2013.
2. Whilst undertaking the placement you will be investigating clinical outcomes from data stored in hospital clinical information systems.
3. Notwithstanding paragraph 2 above the Trust may terminate the placement at any time by giving you not less than 1 week's written notice. Either party may terminate the placement with immediate effect by serving written notice on the other party in the event of serious or persistent un-remedied breach of the other party's obligations under these arrangements.
4. You will not disclose to any person any information of a confidential nature received or acquired by you in connection with the placement including without prejudice to the generality of the foregoing:
    a. financial or other confidential information about or relating to the Trust, and
    b. any Personal Data or Sensitive Personal Data (within the meaning of the Data Protection Act 1998)

    other than to the extent expressly permitted by the Trust.
5. For the duration of the placement you will not be an employee of Portsmouth Hospitals NHS Trust and not entitled to receive any remuneration from it.
6. During the placement, as you will be working on the Trust's premises, you will be required to comply with the Trust's policies on:
    a. Health and Safety,
    b. Infection Control, including naked below the elbow,

c. Security,

d. Dress,

e. Confidentiality, and

f. Any other policy notified to you as applicable to the work or any other activities you carry out during the placement on Trust premises or with Trust patients/clients.

7. Whilst undertaking your placement and duties at Portsmouth Hospitals NHS Trust, you will be covered by the Trust's insurance schemes.

Should you have any queries regarding the content of this letter, please contact the Employee Resourcing Department as detailed above.

Please sign and return one copy of this letter as acknowledgement of receipt.

Yours sincerely

pp Hannah Stevens
Employee Resourcing Team Leader
For and on behalf of Portsmouth Hospitals NHS Trust

DO NOT DETACH

I acknowledge receipt of the above letter.

Signed.....Date.....08-12-2009

Appendix 2. The characteristics of Biochemistry and Haematology Outcome Model (BHOM) dataset

Table 1. The characteristics of Q1 dataset (n1 = 2257), data covering period 1 January to 31 March 2001

Categorical attributes		Number of records	Percentage(%)	Hospital mortality(%)
Gender	Male	1139	50.5	3.3
	Female	1118	49.5	4.2
Mode of Admission	Emergency	2202	97.6	7.5
	Elective	55	2.4	0
Continuous attributes	Range		Mean	Std. Deviation
	Min	Max		
Age	16.0	104.5	63.3	80.8
Haemoglobin	3.4	23.8	13.5	2.2
White cell count	0.1	51.3	10.4	4.9
Urea	0.9	71.7	8.0	6.7
Serum sodium	112	163	137.8	4.4
Serum potassium	2.1	8.3	4.3	0.6
Creatinine	39	1167	114.3	80.5
Albumin	13	58	39.7	5.7
Urea/creatinine ratio	0.02	0.27	0.07	0.03

Table 2. The characteristics of Q2 dataset (n2 = 2335), data covering period 1 April--30 June 2001

Categorical attributes		Number of records	Percentage(%)	Hospital mortality(%)
Gender	Male	1164	49.9	4.0
	Female	1171	50.1	4.2
Mode of Admission	Emergency	2280	97.6	8.1
	Elective	55	2.4	0.1
Continuous attributes	Range		Mean	Std. Deviation
	Min	Max		
Age	16.9	102.6	63.7	18.9
Haemoglobin	4.1	19.6	13.3	2.0
White cell count	0.4	70.1	10.7	5.3
Urea	1.0	71.7	8.3	6.8
Serum sodium	108	165	138.0	4.7
Serum potassium	2.1	8.1	4.3	0.6
Creatinine	39	1204	116.9	93.0
Albumin	12	58	39.7	5.6
Urea/creatinine ratio	0.02	0.27	0.07	0.03

Table 3. The characteristics of Q3 dataset (n3 = 2361), data covering period 1 July - 31 September 2001

Categorical attributes				
		Number of records	Percentage(%)	Hospital mortality(%)
Gender	Male	1148	48.6	3.2
	Female	1213	51.4	4.1
Mode of Admission	Emergency	2308	97.8	7.2
	Elective	53	2.2	0.1
Continuous attributes				
	Range			
	Min	Max	Mean	Std. Deviation
Age	16.3	105.5	63.0	19.2
Haemoglobin	3.6	19.3	13.3	2.1
White cell count	1.2	80.1	10.6	5.2
Urea	0.7	56.7	7.9	6.2
Serum sodium	108	162	137.9	4.7
Serum potassium	2.0	162	4.2	0.7
Creatinine	37	1204	116.7	97.0
Albumin	12	60	39.5	5.9
Urea/creatinine ratio	0.01	0.3	0.07	.03

Table 4. The characteristics of Q4 dataset (n4 = 2544), data covering period 1 October - 31 December 2001

Categorical attributes				
		Number of records	Percentage(%)	Hospital mortality(%)
Gender	Male	1312	51.6	3.8
	Female	1232	48.4	4.4
Mode of Admission	Emergency	2479	97.4	8.1
	Elective	65	2.6	0.1
Continuous attributes				
	Range			
	Min	Max	Mean	Std. Deviation
Age	5.3	99.2	64.3	19.2
Haemoglobin	3.1	21.8	13.3	2.1
White cell count	1.4	66.2	10.5	5.4
Urea	1.0	70.0	8.0	6.9
Serum sodium	103	176	137.9	4.7
Serum potassium	1.7	8.3	4.3	0.6
Creatinine	36	1204	118.2	98.1
Albumin	12	56	39.7	5.8
Urea/creatinine ratio	0.01	0.33	0.07	0.03

Appendix 3. The performance of six (6) methods using 10-fold Cross Validation that was repeated 10 times

In the table, LR = logistic regression, DT = decision trees, SVM = support vector machine, NB = naïve bayes, NN = neural networks, KNN =K-nearest neighbours.

Subset$_1$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.756	0.718	0.617	0.710	0.738	0.655
2	Fold2	0.786	0.765	0.615	0.824	0.796	0.712
3	Fold3	0.778	0.781	0.615	0.707	0.759	0.652
4	Fold4	0.806	0.792	0.701	0.768	0.783	0.685
5	Fold5	0.756	0.688	0.583	0.727	0.756	0.604
6	Fold6	0.788	0.779	0.649	0.758	0.761	0.707
7	Fold7	0.776	0.786	0.620	0.754	0.769	0.629
8	Fold8	0.776	0.794	0.629	0.768	0.825	0.647
9	Fold9	0.773	0.742	0.562	0.744	0.760	0.607
10	Fold10	0.759	0.722	0.598	0.720	0.759	0.648

Subset$_2$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.792	0.774	0.636	0.778	0.756	0.683
2	Fold2	0.769	0.759	0.655	0.739	0.753	0.658
3	Fold3	0.782	0.775	0.678	0.767	0.755	0.666
4	Fold4	0.812	0.787	0.612	0.758	0.798	0.694
5	Fold5	0.804	0.794	0.604	0.777	0.761	0.679
6	Fold6	0.787	0.741	0.639	0.751	0.775	0.681
7	Fold7	0.780	0.750	0.613	0.767	0.741	0.709
8	Fold8	0.781	0.804	0.685	0.733	0.749	0.702
9	Fold9	0.752	0.722	0.668	0.712	0.738	0.642
10	Fold10	0.748	0.763	0.660	0.736	0.753	0.658

Subset$_3$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.810	0.776	0.660	0.784	0.768	0.657
2	Fold2	0.777	0.769	0.685	0.756	0.776	0.691
3	Fold3	0.783	0.793	0.603	0.749	0.786	0.674
4	Fold4	0.824	0.838	0.668	0.818	0.820	0.656
5	Fold5	0.770	0.756	0.639	0.748	0.718	0.608
6	Fold6	0.738	0.726	0.647	0.749	0.730	0.648
7	Fold7	0.763	0.786	0.630	0.750	0.774	0.664
8	Fold8	0.766	0.743	0.631	0.776	0.744	0.632
9	Fold9	0.801	0.817	0.613	0.754	0.779	0.714
10	Fold10	0.836	0.766	0.629	0.791	0.794	0.674

Subset$_4$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.784	0.774	0.571	0.742	0.709	0.679
2	Fold2	0.741	0.742	0.626	0.707	0.712	0.655
3	Fold3	0.776	0.754	0.645	0.744	0.757	0.687
4	Fold4	0.771	0.737	0.693	0.754	0.754	0.657
5	Fold5	0.783	0.769	0.615	0.767	0.725	0.646
6	Fold6	0.797	0.807	0.640	0.772	0.772	0.616
7	Fold7	0.796	0.748	0.675	0.764	0.794	0.693
8	Fold8	0.790	0.749	0.622	0.771	0.746	0.677
9	Fold9	0.801	0.754	0.642	0.767	0.790	0.659
10	Fold10	0.796	0.787	0.622	0.752	0.782	0.650

Subset$_5$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.796	0.745	0.619	0.781	0.794	0.649
2	Fold2	0.768	0.783	0.672	0.764	0.744	0.659
3	Fold3	0.763	0.711	0.594	0.749	0.716	0.611
4	Fold4	0.788	0.761	0.667	0.748	0.731	0.682
5	Fold5	0.795	0.793	0.625	0.749	0.788	0.641
6	Fold6	0.784	0.776	0.662	0.756	0.758	0.649
7	Fold7	0.781	0.780	0.647	0.749	0.693	0.665
8	Fold8	0.764	0.744	0.598	0.729	0.747	0.687
9	Fold9	0.816	0.767	0.608	0.760	0.799	0.673
10	Fold10	0.755	0.739	0.626	0.718	0.761	0.622

Subset$_6$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.793	0.755	0.624	0.765	0.755	0.665
2	Fold2	0.751	0.725	0.576	0.724	0.737	0.618
3	Fold3	0.795	0.796	0.666	0.768	0.784	0.639
4	Fold4	0.760	0.747	0.630	0.736	0.736	0.609
5	Fold5	0.784	0.766	0.629	0.755	0.726	0.636
6	Fold6	0.781	0.758	0.659	0.764	0.764	0.619
7	Fold7	0.807	0.778	0.700	0.769	0.771	0.652
8	Fold8	0.771	0.765	0.641	0.754	0.743	0.649
9	Fold9	0.736	0.699	0.554	0.727	0.730	0.638
10	Fold10	0.795	0.767	0.684	0.809	0.784	0.713

Subset$_7$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.759	0.767	0.598	0.734	0.753	0.685
2	Fold2	0.777	0.768	0.639	0.754	0.771	0.660
3	Fold3	0.786	0.761	0.638	0.754	0.791	0.644
4	Fold4	0.803	0.739	0.538	0.757	0.773	0.632
5	Fold5	0.789	0.763	0.642	0.746	0.745	0.661
6	Fold6	0.784	0.719	0.667	0.748	0.723	0.696
7	Fold7	0.784	0.732	0.640	0.748	0.773	0.687
8	Fold8	0.799	0.740	0.649	0.789	0.781	0.650
9	Fold9	0.785	0.811	0.700	0.767	0.715	0.698
10	Fold10	0.800	0.803	0.621	0.759	0.756	0.655

Subset$_8$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.818	0.769	0.607	0.781	0.772	0.661
2	Fold2	0.756	0.752	0.689	0.765	0.754	0.692
3	Fold3	0.777	0.736	0.622	0.760	0.761	0.690
4	Fold4	0.759	0.776	0.618	0.729	0.755	0.626
5	Fold5	0.747	0.735	0.617	0.724	0.760	0.619
6	Fold6	0.805	0.768	0.670	0.793	0.786	0.688
7	Fold7	0.724	0.721	0.613	0.723	0.671	0.635
8	Fold8	0.767	0.726	0.587	0.718	0.697	0.684
9	Fold9	0.784	0.762	0.614	0.758	0.752	0.667
10	Fold10	0.747	0.736	0.644	0.725	0.723	0.639

Subset$_9$

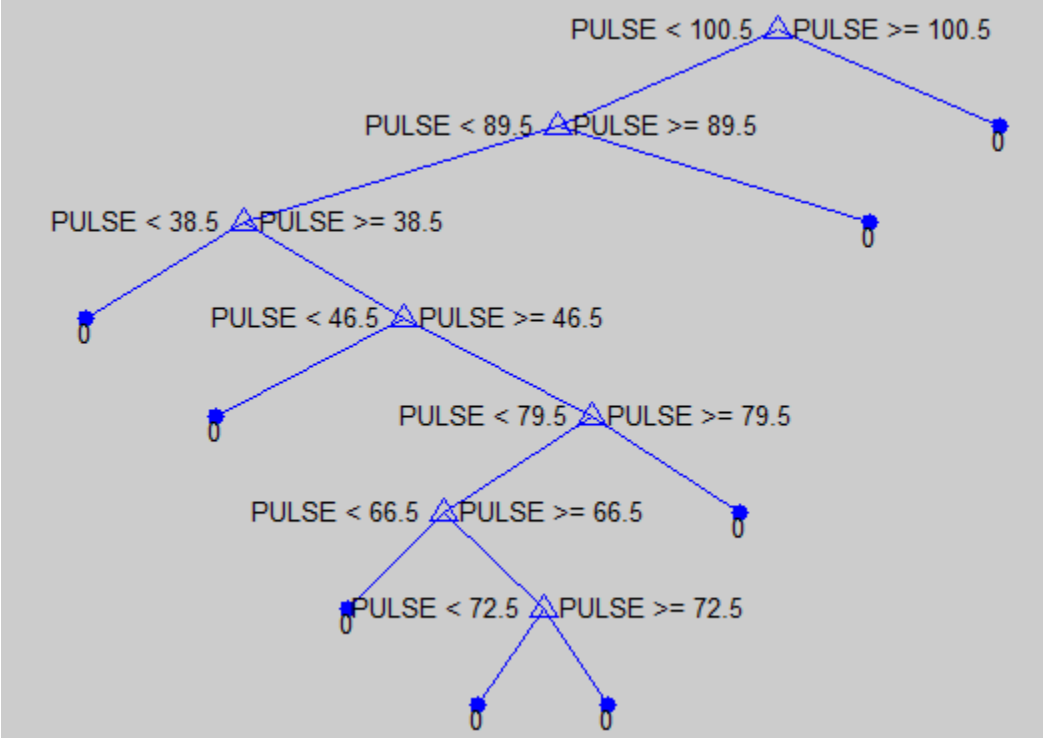
No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.763	0.719	0.617	0.740	0.764	0.583
2	Fold2	0.748	0.740	0.596	0.736	0.680	0.614
3	Fold3	0.787	0.766	0.619	0.765	0.789	0.625
4	Fold4	0.789	0.802	0.671	0.761	0.780	0.648
5	Fold5	0.736	0.707	0.711	0.747	0.677	0.645
6	Fold6	0.803	0.775	0.611	0.769	0.769	0.647
7	Fold7	0.697	0.673	0.584	0.726	0.725	0.638
8	Fold8	0.754	0.718	0.658	0.730	0.720	0.630
9	Fold9	0.797	0.761	0.638	0.750	0.780	0.658
10	Fold10	0.748	0.777	0.598	0.728	0.710	0.653

Subset$_{10}$

No.	*Fold*	*c-index*					
		LR	**DT**	**SVM**	**NB**	**NN**	**KNN**
1	Fold1	0.807	0.725	0.685	0.803	0.749	0.671
2	Fold2	0.794	0.802	0.632	0.764	0.782	0.695
3	Fold3	0.790	0.723	0.627	0.761	0.761	0.652
4	Fold4	0.782	0.760	0.618	0.768	0.756	0.650
5	Fold5	0.741	0.726	0.674	0.739	0.707	0.631
6	Fold6	0.807	0.788	0.663	0.788	0.795	0.676
7	Fold7	0.769	0.772	0.606	0.778	0.758	0.707
8	Fold8	0.786	0.769	0.664	0.765	0.773	0.670
9	Fold9	0.770	0.716	0.669	0.790	0.737	0.667
10	Fold10	0.799	0.782	0.648	0.731	0.753	0.659

Appendix 4. Developing DTEWS on vital sign dataset in VIEWS paper by (Prytherch, et al., 2010)

PULSE (heart rate)



Tree Table for PULSE (heart rate) variable

No.	Death = T		Total		Percent of event=T / total percentage of event (**1.006**)	Split values	**Score**
	Number of event =T on split values	Percent of event= T on split values	Number of records on split values	Percent of records on split values			
1	1999	**1.006**	198755	100			
2	16	8.84	181	0.09	8.79	≤38	3
3	25	1.85	1349	0.68	1.84	[39 , 46]	1
4	191	0.44	43833	22.05	0.43	[47 , 66]	0
5	138	0.56	24664	12.41	0.56	[67 , 72]	0
6	143	0.49	29110	14.65	0.49	[73 , 79]	0
7	291	0.75	38679	19.46	0.75	[80 , 89]	0
8	427	1.34	31771	15.99	1.34	[90 , 100]	1
9	768	2.63	29168	14.68	2.62	>=101	2

Rule set for PULSE attribute

if (pulse<38.5) then score=3

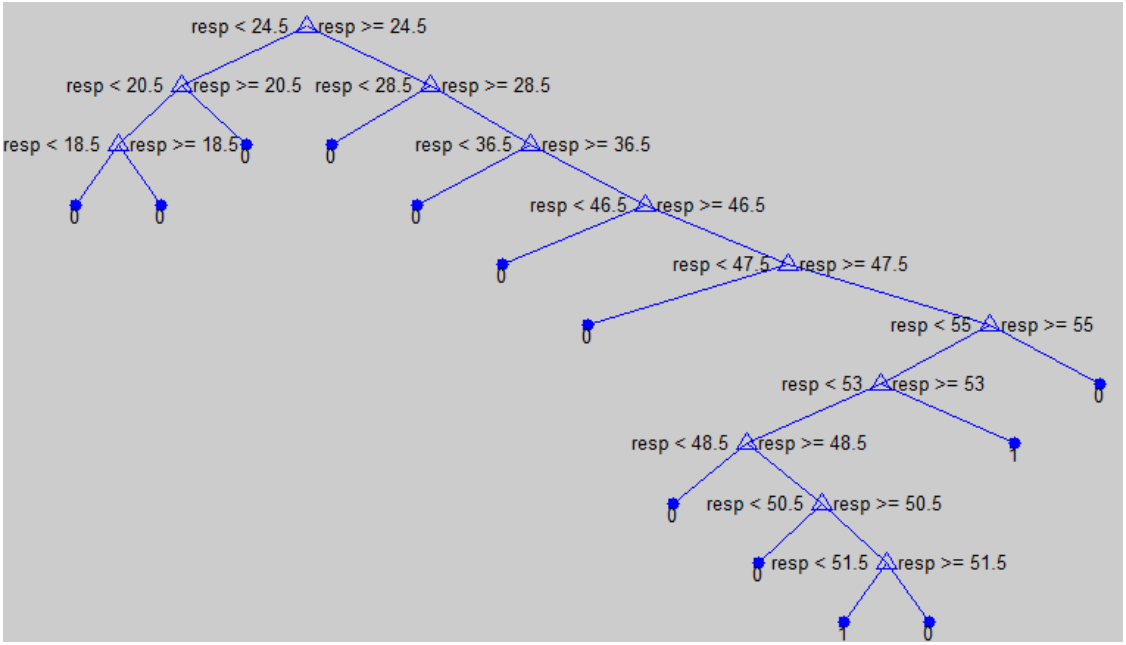
Elseif (pulse>=38.5 and pulse<46.5) then score=1

Elseif (pulse>=46.5 and pulse<89.5) then score=0

Elseif (pulse>=89.5 and pulse<100.5) then score=1

Elseif (pulse>=100.5) then score=2

RESP (respiratory rate)



Tree Table for RESP (respiratory rate) variable

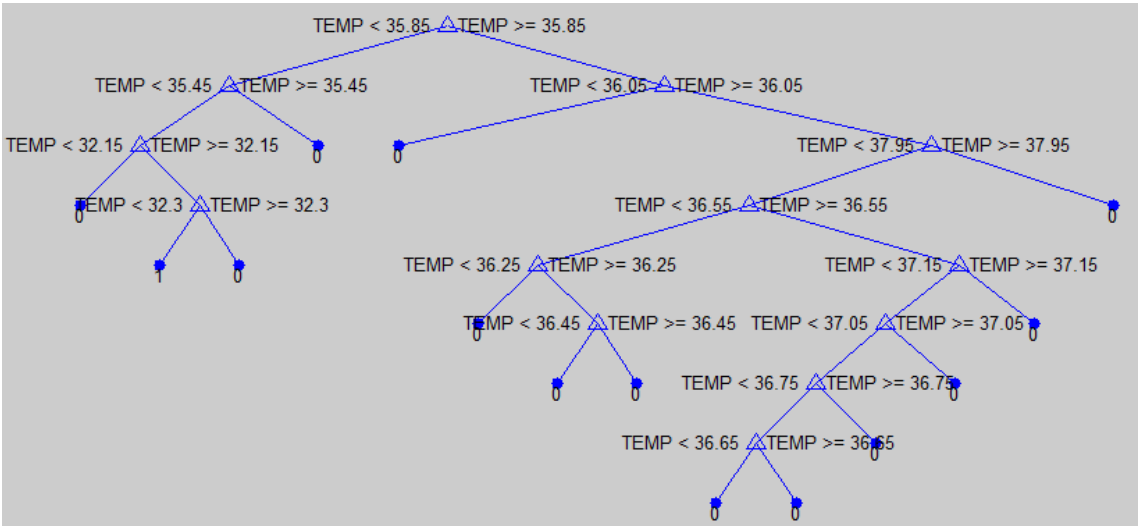
No.	Death = T		Total		Percent of event=T / total percentage of event (**1.006**)	Split values	**Score**
	Number of event =T on split values	Percent of event=T on split values	Number of records on split values	Percent of records on split values			
1	1999	**1.006**	198755	100			
2	628	0.42	148910	74.92	0.42	≤18	0
3	352	1.28	27598	13.89	1.19	[19 , 20]	1
4	426	2.97	14334	7.21	2.95	[21 , 24]	2
5	277	5.63	4919	2.47	5.60	[25 , 28]	3
6	220	9.51	2313	1.16	9.45	[29 , 36]	3
7	82	13.42	611	0.31	13.34	[37 , 46]	3
8	2	50.00	4	0	49.71	[47 , 47]	3
9	6	20.69	29	0.01	20.57	[48 , 48]	3

No.	Death = T		Total		Percent of event=T / total percentage of event (**1.006**)	Split values	**Score**
	Number of event =T on split values	Percent of event=T on split values	Number of records on split values	Percent of records on split values			
10	1	9.09	11	0.01	9.03	[49 , 50]	3
11	1	100.00	1	0	99.42	[51, 51]	3
12	2	14.29	14	0.01	14.20	[52 , 52]	3
13	1	100.00	1	0	99.42	[53 , 54]	3
14	1	10.00	10	0.01	10.00	≥55	3

Rule set for RESP (respiratory rate) variable

if (resp<18.5) then score=0
Elseif (resp>=18.5 and resp<20.5) then score=1
Elseif (resp>=20.5 and resp<24.5) then score=2
Elseif (resp>=24.5) then score=3

TEMP



Tree Table for TEMP (temperature) variable

No	Death = T		Total		Percent of event=T / total percentage of event (**1.006**)	Split values	**Score**
	Number of event =T on split values	Percent of event=T on split values	Number of records on split values	Percent of records on split values			
1	1999	**1.006**	198755	100			
2	0	0.00	13	0.01	0.00	≤32.1	3
3	2	100.0	2	0.00	99.43	[32.2 , 32.30]	3
4	87	13.28	655	0.33	13.21	[32.4 , 35.4]	3
5	75	6.39	1173	0.59	6.36	[35.5 , 35.8]	3
6	183	2.80	6530	3.29	2.79	[35.9 , 36.0]	2
7	183	1.28	14328	7.21	1.27	[36.1 , 36.2]	1
8	312	1.06	29561	14.87	1.05	[36.3 , 36.4]	1
9	225	0.91	24686	12.42	0.91	[36.5 , 36.5]	0
10	167	0.68	24557	12.36	0.68	[36.6 , 36.6]	0
11	148	0.57	25838	13.00	0.57	[36.7 , 36.7]	0
12	317	0.70	45551	22.92	0.69	[36.8 , 37.0]	0
13	36	0.54	6727	3.38	0.53	[37.1 , 37.1]	0
14	176	1.12	15776	7.94	1.11	[37.2 , 37.9]	1
15	88	2.62	3358	1.69	2.61	≥ 38.0	2

Rule set for TEMP (temperature) variable:

if (temp<35.85) then score=3

Elseif (temp>=35.85 and temp<36.05) then score=2

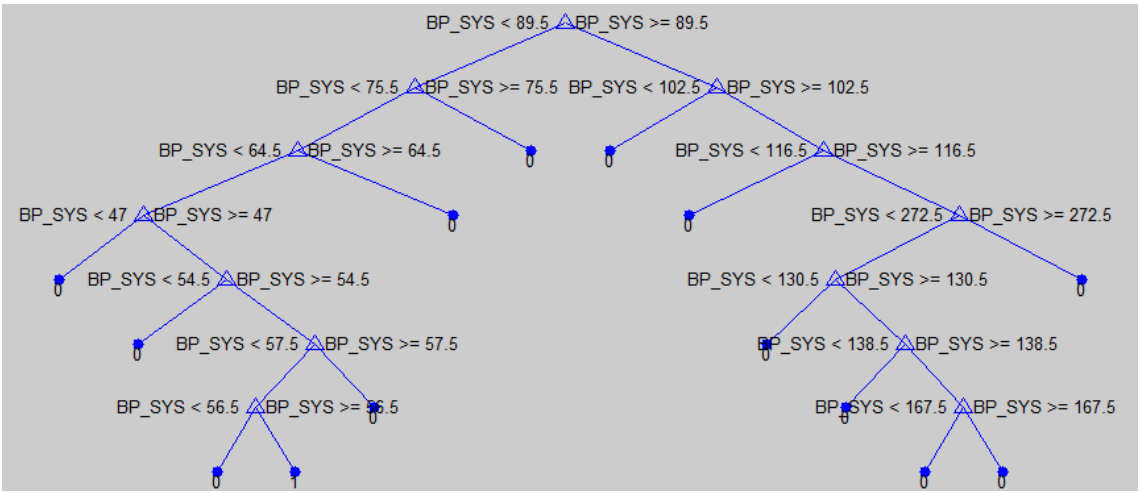
Elseif (temp>=36.05 and temp<36.45) then score=1

Elseif (temp>=36.45 and temp<37.15) then score=0

Elseif (temp>=37.15 and temp<37.95) then score=1

Elseif (temp>=37.95) then score=2

BP_SYS (BP systolic)



Tree Table for BP_SYS (BP systolic) variable

No.	Death = T		Total		Percent of event=T / total percentage of event (**1.006**)	Split values	**Score**
	Number of event =T on split values	Percent of event=T on split values	Number of records on split values	Percent of records on split values			
1	1999	**1.006**	198755	100			
2	2	50	4	0.00	49.71	<46.0	3
3	2	13.33	15	0.01	13.26	[47.0 , 54]	3
4	4	40.00	10	0.01	39.77	[55 , 56]	3
5	3	60.00	5	0.00	59.66	[57 , 57]	3
6	23	25.27	91	0.05	25.13	[58 , 64]	3
7	95	13.25	717	0.36	13.17	[65 , 75]	3
8	255	5.03	5073	2.55	5.00	[76 , 89]	3
9	389	1.94	20082	10.10	1.93	[90 , 102]	1
10	429	1.02	41942	21.10	1.02	[103 , 116]	1
11	334	0.67	49901	25.11	0.67	[117 , 130]	0
12	122	0.47	25783	12.97	0.47	[131 , 138]	0
13	306	0.64	47449	23.87	0.64	[139 , 167]	0
14	34	0.44	7681	3.86	0.44	[168 , 272]	0
15	1	50.00	2	0.00	49.71	>=273	3

Rule set for BP_SYS (BP systolic) variable:

if (bp_sys<89.5) then score=3

Elseif (bp_sys>=89.5 and bp_sys<116.5) then score=1

Elseif (bp_sys>=116.5 and bp_sys<272.5) then score=0

Elseif (bp_sys>=272.5) then score=3

Tree Table for CONS_LEVEL (conscious level) variable:

No.	Death = T		Total		Percent of event=T / total percentage of event (**1.006**)	Split values	**Score**
	Number of event =T on split values	Percent of event=T on split values	Number of records on split values	Percent of records on split values			
1	1999	**1.006**	198755	100			
2	1170	0.64	182307	91.72	0.64	ALERT	0
3	829	5.04	16448	8.28	5.01	NOT ALERT	3

Rule set for CONS_LEVEL (conscious level) variable:

if (CONS_LEVEL="ALERT") then score=0

Elseif then score=3

Tree Table for O2_CONC (any supplemental oxygen?) variable

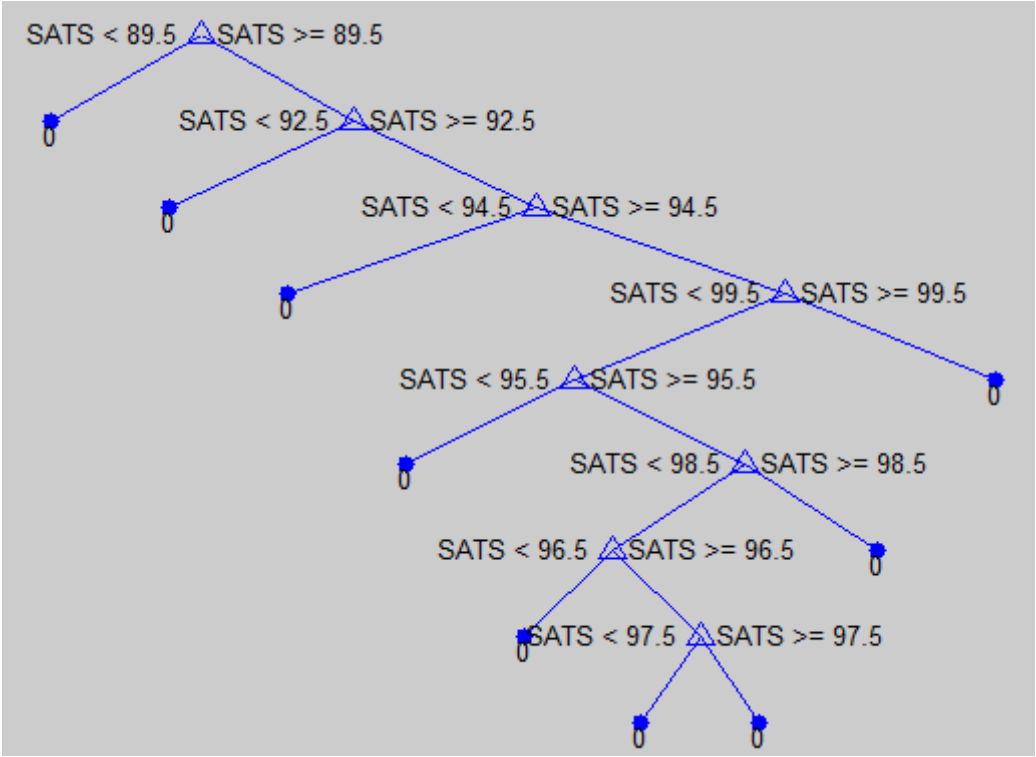
No.	Death = T		Total		Percent of event=T / total percentage of event (**1.006**)	Split values	**Score**
	Number of event =T on split values	Percent of event=T on split values	Number of records on split values	Percent of records on split values			
1	1999	**1.006**	198755	100			
2	478	0.31	153167	77.06	0.31	Inspired O2=NO	0
3	1521	3.34	45588	22.94	3.32	Inspired O2=YES	3

Rule set for O_2CONC (any supplemental oxygen?) variable

if (O2_CONS=21) then score=0

Else then score=3

SATS (S_pO_2) variable



Tree Table for SATS (S_pO_2) variable

No.	Death = T		Total		Percent of event=T / total percentage of event (**1.006**)	Split values	**Score**
	Number of event =T on split values	Percent of event=T on split values	Number of records on split values	Percent of records on split values			
1	1999	**1.006**	198755	100			ATS
2	394	9.00	4376	2.20	8.95	≤89	3
3	292	2.41	12127	6.10	2.39	[90 , 92]	2
4	273	1.12	24306	12.23	1.12	[93 , 94]	1
5	203	0.79	25670	12.92	0.79	[95 , 95]	0
6	207	0.59	35017	17.62	0.59	[96 , 96]	0
7	146	0.45	32796	16.50	0.44	[97 , 97]	0
8	179	0.58	30929	15.56	0.58	[98 , 98]	0
9	150	0.71	21135	10.63	0.71	[99 , 99]	0
9	155	1.25	12399	6.24	1.24	≥100	1

if (SATS<89.5) then score=3

Elseif (SATS>=89.5 and SATS<92.5) then score=2

Elseif (SATS>=92.5 and SATS<94.5) then score=1

Elseif (SATS>=94.5 and SATS<99.5) then score=0

Elseif (SATS>=99.5) then score=1

References

Allen, K. (2004). Recognising and managing adult patients who are critically sick. *Nurs Times, 100*(35), 34-37.

Asiimwe, A. C., Brims, F. J., Andrews, N. P., Prytherch, D. R., Higgins, B. R., Kilburn, S. A., et al. (2011). Routine laboratory tests can predict in-hospital mortality in acute exacerbations of COPD. *Lung, 189*(3), 225-232.

Basuki, A., Badriyah, T., & Ridho, A. (2009). *Data Mining course materials*. Surabaya, Indonesia: Politeknik Elektronika Negeri Surabaya (PENS).

Cook, N. R. (2008). Statistical evaluation of prognostic versus diagnostic models: beyond the ROC curve *Clin Chem* (Vol. 54, pp. 17-23). United States.

Copeland, G. P., Jones, D., & Walters, M. (1991). POSSUM: a scoring system for surgical audit. *Br J Surg, 78*(3), 355-360.

Cuthbertson, B. H., & Smith, G. B. (2007). A warning on early-warning scores! *Br J Anaesth* (Vol. 98, pp. 704-706). England.

Daley, J., Khuri, S. F., Henderson, W., Hur, K., Gibbs, J. O., Barbour, G., et al. (1997). Risk adjustment of the postoperative morbidity rate for the comparative assessment of the quality of surgical care: results of the National Veterans Affairs Surgical Risk Study. *J Am Coll Surg, 185*(4), 328-340.

DeVita, M. A., Bellomo, R., & Hillman, K. (2006). Introduction to the rapid response systems series. *Jt Comm J Qual Patient Saf, 32*(7), 359-360.

DeVita, M. A., Smith, G. B., Adam, S. K., Adams-Pizarro, I., Buist, M., Bellomo, R., et al. (2010). "Identifying the hospitalised patient in crisis"--a consensus conference on the afferent limb of rapid response systems *Resuscitation* (Vol. 81, pp. 375-382). Ireland: 2010 Elsevier Ireland Ltd.

Duckitt, R. W., Buxton-Thomas, R., Walker, J., Cheek, E., Bewick, V., Venn, R., et al. (2007). Worthing physiological scoring system: derivation and validation of a physiological early-warning system for medical admissions. An observational, population-based single-centre study. *Br J Anaesth, 98*(6), 769-774.

Gao, H., McDonnell, A., Harrison, D. A., Moore, T., Adam, S., Daly, K., et al. (2007). Systematic review and evaluation of physiological track and trigger warning systems for identifying at-risk patients on the ward. *Intensive Care Med, 33*(4), 667-679.

Goldhill, D. R., McNarry, A. F., Mandersloot, G., & McGinley, A. (2005). A physiologically-based early warning score for ward patients: the association between score and outcome. *Anaesthesia, 60*(6), 547-553.

Goldhill, D. R., White, S. A., & Sumner, A. (1999). Physiological values and procedures in the 24 h before ICU admission from the ward *Anaesthesia* (Vol. 54, pp. 529-534). England.

Han, J., Kamber, M., & Pei, J. (2006). *Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)*: Morgan Kaufmann.

Kantardzic, M. (2002). *Data Mining: Concepts, Models, Methods, and Algorithms*: Wiley-IEEE Press.

Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 29*(2), 119-127.

Khuri, S. F., Daley, J., Henderson, W., Hur, K., Gibbs, J. O., Barbour, G., et al. (1997). Risk adjustment of the postoperative mortality rate for the comparative assessment of the quality of surgical care: results of the National Veterans Affairs Surgical Risk Study. *J Am Coll Surg, 185*(4), 315-327.

Kirkwood, B. R., & Sterne, J. A. C. (2003). *Essential Medical Statistics*: Blackwell Science.

Lee, A., Bishop, G., Hillman, K. M., & Daffurn, K. (1995). The Medical Emergency Team. *Anaesth Intensive Care, 23*(2), 183-186.

Lemeshow, S., & Hosmer, D. W., Jr. (1982). A review of goodness of fit statistics for use in the development of logistic regression models. *Am J Epidemiol, 115*(1), 92-106.

Macrina, F., Puddu, P. E., Sciangula, A., Totaro, M., Trigilia, F., Cassese, M., et al. (2010). Long-term mortality prediction after operations for type A ascending aortic dissection *J Cardiothorac Surg* (Vol. 5, pp. 42). England.

McDonnell, A., Esmonde, L., Morgan, R., Brown, R., Bray, K., Parry, G., et al. (2007). The provision of critical care outreach services in England: findings from a national survey *J Crit Care* (Vol. 22, pp. 212-218). United States.

Morgan, R. J., & Wright, M. M. (2007). In defence of early warning scores *Br J Anaesth* (Vol. 99, pp. 747-748). England.

National Institute for Health and Clinical Excellence. (2007). *National Institute for Health and Clinical Excellence: Acutely ill patients in hospital: recognition of and response to acute illness in adults in hospital.*

Pine, M., Jones, B., & Lou, Y. B. (1998). Laboratory values improve predictions of hospital mortality. *Int J Qual Health Care, 10*(6), 491-501.

Prytherch, D. R., Briggs, J. S., Weaver, P. C., Schmidt, P., & Smith, G. B. (2005). Measuring clinical performance using routinely collected clinical data *Med Inform Internet Med* (Vol. 30, pp. 151-156). England.

Prytherch, D. R., Sirl, J. S., Schmidt, P., Featherstone, P. I., Weaver, P. C., & Smith, G. B. (2005). The use of routine laboratory data to predict in-hospital death in medical admissions *Resuscitation* (Vol. 66, pp. 203-207). Ireland.

Prytherch, D. R., Sirl, J. S., Weaver, P. C., Schmidt, P., Higgins, B., & Sutton, G. L. (2003). Towards a national clinical minimum data set for general surgery. *Br J Surg, 90*(10), 1300-1305.

Prytherch, D. R., Smith, G. B., Schmidt, P. E., & Featherstone, P. I. (2010). ViEWS--Towards a national early warning score for detecting adult inpatient deterioration. *Resuscitation, 81*(8), 932-937.

Prytherch, D. R., Tang, T., Walsh, S. R., Lees, T., Varty, K., & Boyle, J. R. (2007). VBHOM, a data economic model for predicting the outcome after open abdominal aortic aneurysm surgery. *Br J Surg, 94*(6), 717-721.

Prytherch, D. R., Whiteley, M. S., Higgins, B., Weaver, P. C., Prout, W. G., & Powell, S. J. (1998). POSSUM and Portsmouth POSSUM for predicting mortality. Physiological and Operative Severity Score for the enUmeration of Mortality and morbidity. *Br J Surg, 85*(9), 1217-1220.

Quarterman, C. P., Thomas, A. N., McKenna, M., & McNamee, R. (2005). Use of a patient information system to audit the introduction of modified early warning scoring. *J Eval Clin Pract, 11*(2), 133-138.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*: Morgan Kaufmann Publishers Inc.

Sagar, P. M., Hartley, M. N., Mancey-Jones, B., Sedman, P. C., May, J., & Macfie, J. (1994). Comparative audit of colorectal resection with the POSSUM scoring system. *Br J Surg, 81*(10), 1492-1494.

Shannon, C. E. (1948). A Mathematical Theory of Communication. Retrieved 9 July, 2013

Silke, B., Kellett, J., Rooney, T., Bennett, K., & O'Riordan, D. (2010). An improved medical admissions risk system using multivariable fractional polynomial logistic regression modelling *QJM* (Vol. 103, pp. 23-32). England.

Smith, G. B., Prytherch, D. R., Meredith, P., Schmidt, P. E., & Featherstone, P. I. (2013). The ability of the National Early Warning Score (NEWS) to discriminate patients at risk of early cardiac arrest, unanticipated intensive care unit admission, and death. *Resuscitation, 84*(4), 465-470.

Smith, G. B., Prytherch, D. R., Schmidt, P. E., & Featherstone, P. I. (2008). Review and performance evaluation of aggregate weighted 'track and trigger' systems. *Resuscitation, 77*(2), 170-179.

Smith, G. B., Prytherch, D. R., Schmidt, P. E., Featherstone, P. I., & Higgins, B. (2008). A review, and performance evaluation, of single-parameter "track and trigger" systems. *Resuscitation, 79*(1), 11-21.

Stenhouse, C., Coates, S., Tivey, M., Allsop, P., & Parker, T. (2000). Prospective evaluation of a modified Early Warning Score to aid earlier detection of patients developing critical illness on a general surgical ward. *British Journal of Anaesthesia, 84*(5), 663.

Subbe, C. P., Davies, R. G., Williams, E., Rutherford, P., & Gemmell, L. (2003). Effect of introducing the Modified Early Warning score on clinical outcomes, cardio-pulmonary arrests and intensive care utilisation in acute medical admissions. *Anaesthesia, 58*(8), 797-802.

Subbe, C. P., Gao, H., & Harrison, D. A. (2007). Reproducibility of physiological track-and-trigger warning systems for identifying at-risk patients on the ward. *Intensive Care Med, 33*(4), 619-624.

Subbe, C. P., Kruger, M., Rutherford, P., & Gemmel, L. (2001). Validation of a modified Early Warning Score in medical admissions. *QJM, 94*(10), 521-526.

Tang, T., Walsh, S. R., Prytherch, D. R., Lees, T., Varty, K., & Boyle, J. R. (2007). VBHOM, a data economic model for predicting the outcome after open abdominal aortic aneurysm surgery. *Br J Surg, 94*(6), 717-721.

Tarassenko, L., Clifton, D. A., Pinsky, M. R., Hravnak, M. T., Woods, J. R., & Watkinson, P. J. (2011). Centile-based early warning scores derived from statistical distributions of vital signs. *Resuscitation, 82*(8), 1013-1018.

Verplancke, T., Van Looy, S., Benoit, D., Vansteelandt, S., Depuydt, P., De Turck, F., et al. (2008). Support vector machine versus logistic regression modeling for prediction of hospital mortality in critically ill patients with haematological malignancies *BMC Med Inform Decis Mak* (Vol. 8, pp. 56). England.