THE UNIVERSITY OF HULL

Machine Learning Based Data Pre-processing for the Purpose of Medical Data Mining and Decision Support

being a Thesis submitted for the Degree of

Doctor of Philosophy

in the University of Hull

by

M. Mostafizur Rahman, BSc (Hons), MSc

March 2014

ACKNOWLEDGEMENTS

During the work on this thesis, I have benefited from a lot of communication with many people, who have provided valuable input to my studies. First of all, I would like to thank my supervisor Dr. Darryl N Davis for giving in valuable feedback and advice during my work. His engagement and knowledge have inspired me a lot. I would also like to thank Dr. Chandrasekhar Kambhampati and Prof. Yiannis Papadopoulos. They have helped me with their insights and advice. I would like to thank SEED software of Department of Computer Science of The University of Hull for providing the financial supports during my period of study.

I would also like to thank my family. A specific 0appreciation goes to my parents for supporting and believing in me during all these years. I would especially express my thanks to my wife, Asma Khanam, my son Ahnaf and my daughter Maliha for their love and support.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ABSTRACT

Building an accurate and reliable model for prediction for different application domains, is one of the most significant challenges in knowledge discovery and data mining. Sometimes, improved data quality is itself the goal of the analysis, usually to improve processes in a production database and the designing of decision support. As medicine moves forward there is a need for sophisticated decision support systems that make use of data mining to support more orthodox knowledge engineering and Health Informatics practice. However, the real-life medical data rarely complies with the requirements of various data mining tools. It is often inconsistent, noisy, containing redundant attributes, in an unsuitable format, containing missing values and imbalanced with regards to the outcome class label.

Many real-life data sets are incomplete, with missing values. In medical data mining the problem with missing values has become a challenging issue. In many clinical trials, the medical report pro-forma allow some attributes to be left blank, because they are inappropriate for some class of illness or the person providing the information feels that it is not appropriate to record the values for some attributes. The research reported in this thesis has explored the use of machine learning techniques as missing value imputation methods. The thesis also proposed a new way of imputing missing value by supervised learning. A classifier was used to learn the data patterns from a complete data sub-set and the model was later used to predict the missing values for the full dataset. The proposed machine learning based missing value imputation was applied on the thesis data and the results are compared with traditional Mean/Mode imputation. Experimental results show that all the machine learning methods which we explored outperformed the statistical method (Mean/Mode).

The class imbalance problem has been found to hinder the performance of learning systems. In fact, most of the medical datasets are found to be highly imbalance in their class label. The solution to this problem is to reduce the gap between the minority class samples and the majority class samples. Over-sampling can be applied to increase the number of minority class sample to balance the data. The alternative to over-sampling is under-sampling where the size of majority class sample is reduced. The thesis proposed one cluster based under-sampling technique to reduce the gap between the majority and minority samples. Different under-sampling and over-sampling techniques were explored as ways to balance the data. The experimental results show that for the thesis data the new proposed modified cluster based under-sampling technique performed better than other class balancing techniques.

In further research it is found that the class imbalance problem not only affects the classification performance but also has an adverse effect on feature selection. The thesis proposed a new framework for feature selection for class imbalanced datasets. The research found that, using the proposed framework the classifier needs less attributes to show high accuracy, and more attributes are needed if the data is highly imbalanced.

The research described in the thesis contains the flowing four novel main contributions.

a) Improved data mining methodology for mining medical data

b) Machine learning based missing value imputation method

c) Cluster Based semi-supervised class balancing method

d) Feature selection framework for class imbalance datasets

The performance analysis and comparative study show that the use of proposed method of missing value imputation, class balancing and feature selection framework can provide an effective approach to data preparation for building medical decision support.

ABBREVIATIONS

ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
FIS	Fuzzy Inference System
FL	Fuzzy Logic
FURIA	Fuzzy Unordered Rule Induction Algorithm
KNN	K-nearest Neighbour Algorithm
MA	Majority
MI	Minority
MLP	Multilayer Perception
RAIDOR	Ripple Down Rule
SVM	Support Vector Machine
SMOTE	Synthetic Minority Over-sampling Technique
KDD	Knowledge Discovery and Data Mining
CDS	Clinical Decision Support
IBL	Instance Based Learning
ECG	Electrocardiogram
MCR	Missing Completely at Random

PUBLIC OUTPUT

The work undertaken in producing this thesis has led to the following publications. They inform the thesis and form a part of it.

1. Rahman, M. M. and Davis, D. N. (2014) Semi Supervised Under-Sampling: A solution to the class imbalance problem for classification and feature selection. IAENG Transactions on Engineering Technologies, Springer.
2. Rahman, M. M. and Davis, D. N. (2013). Addressing the Class Imbalance Problem in Medical Datasets. International Journal of Machine Learning and Computing.
3. Rahman, M. M. and Davis, D. N. (2013) Cluster Based Under-Sampling for Unbalanced Cardiovascular Data. The International Conference of Data Mining and Knowledge Engineering, World Congress on Engineering 2013, London.
4. Rahman, M. M. and Davis, D. N. (2012) Machine Learning Based Missing Value Imputation Method for Clinical Datasets. IAENG Transactions on Engineering Technologies, Springer. Rahman, M. M. and Davis, D. N.
5. Fuzzy Unordered Rules Induction Algorithm Used as Missing Value Imputation Methods for K-Mean Clustering on Real Cardiovascular Data. The 2012 International Conference of Data Mining and Knowledge Engineering, World Congress on Engineering 2012 (July 4-6, 2012, London)
6. Rahman, M. M. and Davis, D. N. (2012) "Cluster Based Under-Sampling Technique for Unbalanced Datasets", Poster presented on 4th Annual Departmental Conference for Postgraduate Research, The University of Hull, December 2012.
7. Rahman, M. M. and Davis, D. N. (2011) "Machine Learning Techniques for Missing Value Imputation", Poster presented on 4th Annual Departmental Conference for Postgraduate Research, The University of Hull, December 2011.
8. Rahman, M. M. and Davis, D. N. (2010) "Data Mining in Medicine Using Fuzzy Theory", Poster presented on 3rd Annual Departmental Conference for Postgraduate Research, The University of Hull, January 2011.

CHAPTER 1 : INTRODUCTION

1.1 Motivation of the research

Information retrieval and data mining are two components of the same problem, the search for information and extraction of knowledge from large amounts of data, very large databases or data warehouses. As medicine moves forward there is a need for sophisticated decision support systems that make use of data mining to support more orthodox knowledge engineering and Health Informatics practice. In general data mining and machine learning models can be used to discover knowledge or rules from a data set, which can then be used to solve different problems, such as classification, prediction, clustering and mining association rules etc. A suitable data mining methodology can give a better way of solving many complex problems.

Real-life data rarely complies with the requirements of various data mining tools. It is often inconsistent, noisy, containing redundant attributes, in an unsuitable format, and containing missing values. The problem with missing attribute values is a very important issue in Data Mining. In general, methods to handle missing values belong either to sequential methods or parallel methods (Maimon and Rokach 2010). Clinical data often contains many missing values and typically the datasets are imbalanced with regard to the class label of interest. A well balanced training dataset is very important in creating a good training set for the development of classifiers. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced, because they aim to optimize the overall accuracy without considering the relative distribution of each class (Liu et al. 2011).

The need for a good data preparation methodology for the purpose of medical data mining and decision support was the primary motivation of the research.

1.2 Research problem overview

The adoption of clinical governance in the NHS has mandated the development of appropriate and reliable clinical data-sets for use in comparative audit (Scally 1998). These data-sets will be useless without the ability to interrogate and analyse them in a meaningful way. A validated data mining model would allow them to set achievable national standards and thereby to improve quality of care throughout clinical units in the UK by implementing guidelines and allowing comparative audit using local and national data-sets. Most of the existing predictive models, usually based on linear statistical analysis, have proved disappointing. Statistical methods such as statistical regression, Cox proportional-hazards regression, logistic regression, inverse variance weighted method; or groups' comparison have been commonly used in different studies (Bellamy et al. 2007, Howard et al. 2006, Ruijter W 2009). However, these methods are usually used to explain the data and to model the progression of the disease rather than to make predictions for populations or individual patients.

Many researchers have identified several important and challenging issues (Sittig et al. 2008, Fox et al. 2010, Bellazzi and Zupan 2008) for clinical decision support. In ‘‘Grand challenges for decision support” Sittig et. all (2008) set out 10 critical problems for ‘‘designing, developing, presenting, implementing, evaluating, and maintaining all types of clinical decision support capabilities for clinicians, patients and consumers”. However Sittig et al.’s identification covers little about data pre-processing. Sometimes, improved data quality is itself the goal of the analysis, usually to improve processes in a production database (Dasu and Johnson 2003) and the design of decision support.

Typically, two types of databases are available in medical domains (Dasu and Johnson 2003). The first is the dataset acquired by medical experts, which are collected for a special research topic where data collection is triggered by the hypothesis of a clinical trial. The other type is a huge dataset retrieved from hospital information systems. These data are stored in a database automatically without any specific research purpose. These data records are often used for further analysis and building clinical decision support. These types of datasets are often very complex where the numbers of records are very huge, with a large number of attributes for each record. This data often contains many missing values and typically the datasets are imbalanced with regard to the class label of interest.

Many real-life data sets are incomplete, with missing values. The problem with missing attribute values is a very important issue in data mining, and has become a challenging issue in medical data mining. In many clinical trials, the medical report pro-forma allow some attributes to be left blank, because they are inappropriate for some class of illness or the person providing the information feels that it is not appropriate to record the values for some attributes (Almeida et al. 2010).

Most medical datasets are also not balanced with regard to their class labels. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced. This is because they aim to optimize the overall accuracy without considering the relative distribution of each class (Liu et al. 2011). Therefore, there is a need of a good sampling technique for such datasets where the target classes are not balanced and the given labels are not always appropriate. Indeed in some cases it has been noticed that the given class labels do not accurately represent characteristics of the data record and do not accurately reflect the nature of a patient. For example, some patients are registered as dead but may have died for some other reason than the target cause and some patients are alive by chance or may have died later.

The aim of this project is to investigate suitable data preparation techniques for feature cleaning and reduction and making ready to prepare a good decision support model for medical data mining. Clinical data from cardiovascular medicine and other domains (Merz 1996) are available for use in this project.

1.3 Research questions

There are a wide range of research issues to be addressed in this project. These can be summarised at a high level as the following set of over-arching questions:

a) How can data pre-processing be improved for medical data mining?

b) What forms of techniques (and metrics) are useful for determining data cleansing, feature reduction and classification?

These questions are addressed via specific objectives as detailed in the following section.

1.4 Aim and objectives

The research associated with this project addresses data mining, principled methodologies for its application, and the development of new data preparation methodologies. A set of specific aims can be defined as:

a) Investigation of systematic data preparation techniques for data cleansing and feature reduction.

b) Investigation and Development of metrics for underpinning missing value imputation.

c) Investigation and Development of metrics for underpinning class imbalance problem.

d) Investigation and Development of metrics for underpinning feature selection for class imbalanced dataset.

e) Compare the performance of different classifiers and clustering algorithms on thesis cardiovascular data.

1.5 Thesis structure

The research aims stated above will be dealt within the next eight chapters. Chapter two describes decision support, data mining and the issues with medical data. Chapter three describes different machine learning algorithms for classification and clustering used in this research. Chapter four introduces the process of dimension reduction and feature selection in data mining. The chapter five is the overview of the research case studies. Chapter six, chapter seven and chapter eight present the experiments and analysis of the outcome of the case studies discussed in chapter five. Chapter 9 unites the work of the previous chapters in a practical setting. The thesis concludes in this chapter with an analysis and a discussion of the research outlined in the previous chapters. The final chapter ends with conclusions and suggestions for future work and possible extensions to the research outlined in this thesis.

CHAPTER 2 : INFORMATION MANAGEMENT AND DECISION SUPPORT IN MEDICINE

2.1 Introduction

While evidence-based medicine has increasingly broad-based support in health care (Bates et al. 2003), it remains difficult to get physicians to actually practice it. Information systems can provide decision support to users at the time they make decisions, which results in improved quality of care. Clinical decision support systems have been coined as active knowledge systems, which use two or more items of patient data to generate case-specific advice. This chapter presents a background for decision support, clinical decision support, and methodologies for data mining. The issues and challenges with medical data mining are also discussed in this chapter.

2.2 Decision support

The term Decision Support is used often and in a variety of contexts related to decision making. Decision Support is utilizing computer-based systems that facilitate the use of data, models, and structured decision processes in decision making. Some key words associated with Decision Support are: Decision Theory, Decision Analysis, Operations Research, Management Science, and Artificial Intelligence (Mladenić 2003). Decision Support is a broad field concerned with supporting people in making decisions. It is a part of Decision Sciences, which it shares with normative and descriptive approaches to decision making (Bouyssou 2010). Decision Support encompasses a number of disciplines, including operations research, decision analysis, Decision Support Systems, data warehousing, and group decision support. The major future contributions to decision support are expected in relation with data warehouses, integration with data mining, developments in qualitative modelling and "soft "computing, and networking (Mladenić 2003).

2.3 Clinical decision support

Clinical decision supports are computer systems designed to impact clinician decision making about individual patients at the point in time that these decisions are made (Berner 2007). A typical decision support system consists of five components: the data management, the model management, the knowledge engine, the user interface, and the user(s).

2.4 Data mining

There are many definitions of data mining. Hand et al (2001) produced a general definition as follows:

"Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel

ways that are both understandable and useful to the data owner." (Hand et al. 2001) Chapter 1; Page 1)

Data mining is the way to get information buried in the data, enabling the extraction of hidden patterns from large and complex collections of data. In medical domains, data mining can be seen as the production of a pattern recognition system to predict future patient risks from existing patient records.

2.4.1 Example data mining methodology

There are two popular existing data mining methodologies for the "knowledge discovery from data" process (kdnuggets 2007): CRISP_DM (Catley et al. 2009), and SEMMA (Matignon and Institute 2007).

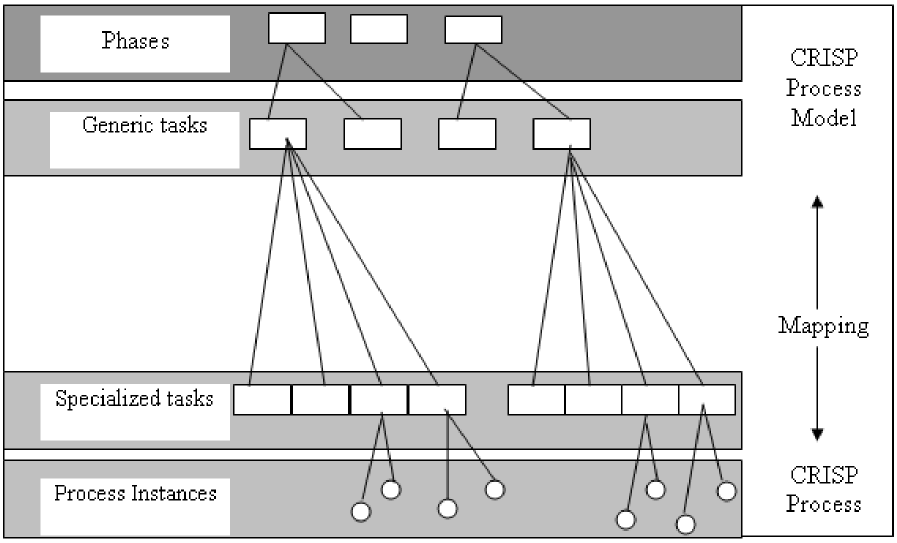


Figure 2-1: The methodology of CRISP-DM (Inc 2000)

CRISP-DM is being developed by an industry led consortium as the Cross-Industry Standard Process Model for Data Mining (see Figure

2.1). It consists of a set of tasks described at four levels from general to specific (Inc 2000). At the top level, the data mining process is organized into a number of phases where each phase consists of several generic tasks at the second level. The second level includes generic tasks which can cover all possible data mining situations such as the process tasks, possible data mining applications, and techniques. In the third level, the specialized task shows detailed actions in generic tasks for certain specific situations. For example, if the generic task is a "dealing with missing data", the more detailed tasks in the third level will be a category of specialized missing data tasks namely "dealing with missing numeric values"; "dealing with missing categorical values"; and so on. The fourth level, as the process instance, is a record of the actions, decisions, and results of an actual data mining engagement.

An example of the use of CRISP-DM methodology of data mining in multidimensional medical data streams can be found in the work of Catley (2009).

SEMMA is a data mining methodology derived from the Statistical Analysis Software Institute (Matignon and Institute 2007) consisting of the five steps: Sample, Explore, Modify, Model, and Assess (SEMMA). All cases from a data set are taken and partitioned into the training, validation and test sets in the Sample step. The Explore step allows data sets to be visualised statistically and graphically. The

Modify step allows the transformation of the data or deals with missing values in the data set. The Model step requires the fitting of the data mining and machine learning techniques such as Decision Tree and Neural Networks. Lastly, the Assess step means using alternative partitions of test sets to validate the derived model in order to estimate how well the data mining process performs.

According to (Fayyad 1996), the Knowledge Discovery in Database (KDD) process is interactive and iterative, involving numerous steps with many decisions made by the user. The process has nine interactive and iterative steps presented in figure 2.2.

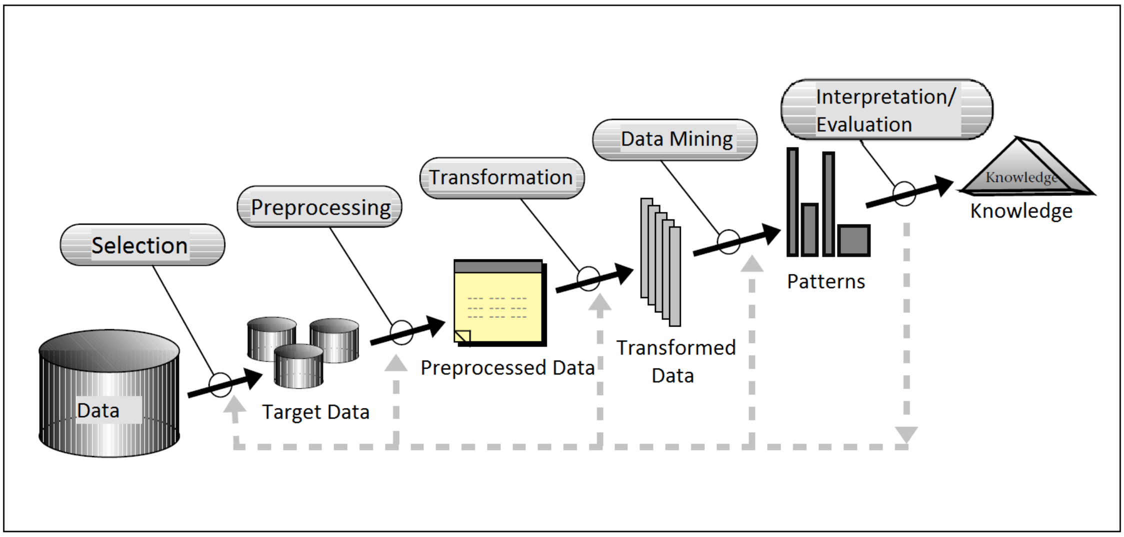


Figure 2-2: An overview of the steps of KDD process (Fayyad 1996)

First is developing an understanding of the application domain and the relevant prior knowledge and identifying the goal of the KDD process from the customer’s viewpoint.

Second is creating a target data set: selecting a data set, or focusing on a subset of variables or data samples, on which discovery is to be performed.

Third is data cleaning and pre-processing. Basic operations include removing noise if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time-sequence information and known changes.

Fourth is data reduction and projection; the finding of useful features to represent the data depending on the goal of the task. With dimensionality reduction or transformation methods, the effective number of variables under consideration can be reduced or in-variant representations for the data can be found.

Fifth is matching the goals of the KDD process (step 1) to a particular data-mining method. For example, summarization, classification, regression, clustering, and so on.

Sixth is exploratory analysis and model and hypothesis selection: choosing the data-mining algorithm(s) and selecting method(s) to be used for searching for data patterns. This process includes deciding which models and parameters might be appropriate (for example, models of categorical data are different than models of vectors over the reals) and matching a particular data-mining method with the overall criteria of the KDD process (for example, the end user might be more interested in understanding the model than its predictive capabilities).

Seventh is data mining: searching for patterns of interest in a particular representational form or a set of such representations, including classification rules or trees, regression, and clustering. The users can significantly aid the data mining method by correctly performing the preceding steps.

Eighth is interpreting mined patterns, possibly returning to any of steps 1 through 7 for further iteration. This step can also involve visualization of the extracted patterns and models or visualization of the data given the extracted models.

Ninth is acting on the discovered knowledge: using the knowledge directly, incorporating the knowledge into another system for further action, or simply documenting it and reporting it to interested parties. This process also includes checking for and resolving potential conflicts with previously believed (or extracted) knowledge.

2.4.2 Thesis data mining methodology

The data mining methodology adopted for this thesis can be seen in Figure 2.2 with little modification in the steps see figure 2.3. This methodology is derived from the knowledge discovery in database process of (Fayyad 1996). The following are reasons for using this specific methodology:

The existing methodologies are not suitable for the thesis data. The CRISP-DM and SEMMA methodologies are too big and too complicated for use with the thesis data domain. For example, CRISP-DM contains the "Business Understanding" and "Deployment" phases whereas the thesis methodology does not. SEMMA includes the task of representing data sets statistically and graphically, again not required for the purposes of this thesis. The existing methodologies also do not cover the class balancing and which is an important part of medical data mining.

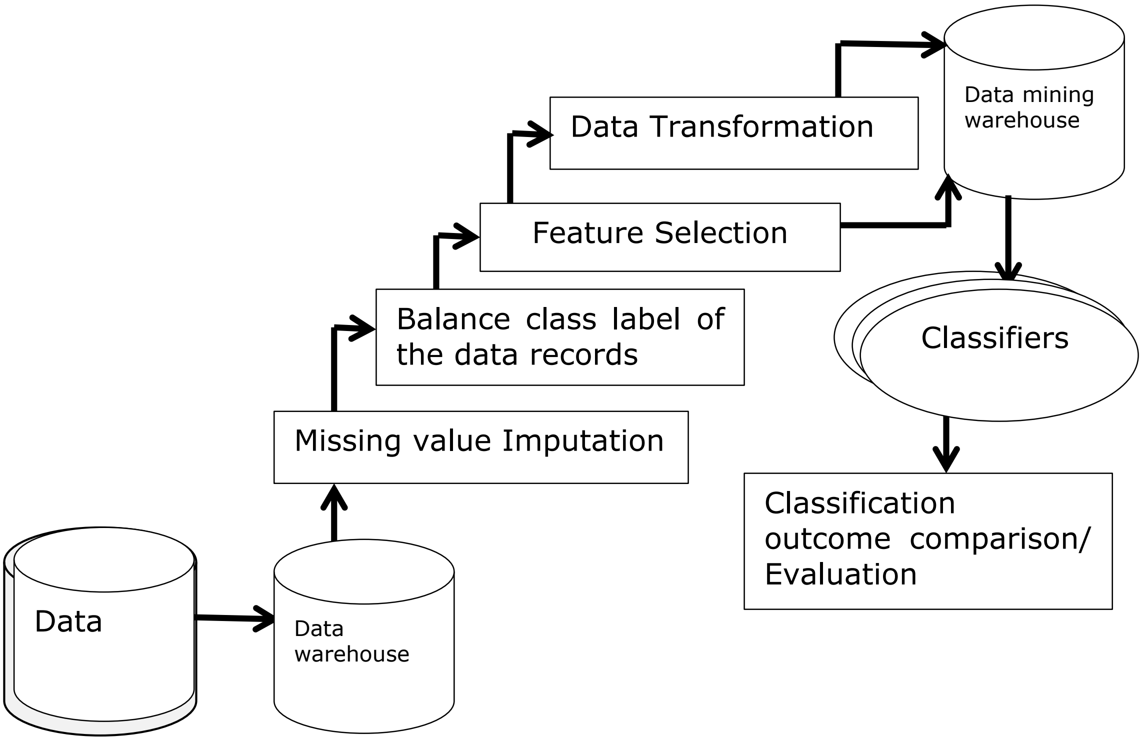


Figure 2-3: data mining methodology

Thesis Methodology:

Step 1 (Selection)**:** The data set relevant to the experiments will be chosen from different data files (Hull Site Data, Dundee Site Data) to make the data mining warehouse.

Step 2 (Data Preparation and feature selection)**:** Data will be analyzed by using data mining methods in order to define how the data is to be made more meaningful and useable for the classification techniques used in later steps.

> **Step 2.1:** Data will be cleaned by supplying missing values with conventional statistical methods and also machine learning techniques will be used to predict missing values.
>
> **Step 2.2:** Class distributions of the data records of the dataset will be balanced. For example, a cluster based under-sampling technique can be used to under sample the majority class to reduce the ratio gap between the classes.
>
> **Step 2.3:** Data will be transformed to more appropriate value types such as numerical as per the need of the classifier.
>
> **Step 2.4:** Different techniques of feature selection technique will be used to select relevant features. For experimentation, this phase will be done in different ways. Feature selection will be done on original data and also data balanced by different balancing techniques.

Step 3 (Data Mining Techniques): Different classifier and clustering algorithms will be applied on the data sets prepared in step 2. The outcome will be stored for further analysis.

Step 4 (Comparison/ Evaluation): The classified results are compared or evaluated based on standard measures such as mean square error, confusion matrix, sensitivity and specificity rates, the positive predictive value, and negative predictive value.

Step 5 (Building New Models): The data set is then stored in the "Data Mining Warehouse" for further prediction and analysis processes.

2.5 Data mining in medicine

In the past twenty years there has been a transformation in patient record management, with medical information being stored electronically as Hospital Information System or medical database (e.g. see European Institute for EHealth Records (Eurorec 2014)). In order to get knowledge out of the data, more intelligent techniques such as data mining and classical statistical methods have been used (Shusaku 2000). Medical decision support systems are designed to support clinicians in their diagnosis by mostly the use of linear statistical analysis, which have in the main produced disappointing results (Bellamy et al. 2007); (Howard et al. 2006); (Ruijter W 2009); (Wang et al. 2006).

Machine learning is a branch of artificial intelligence, concerns the construction and study of systems that can learn from data. The use of Machine Learning techniques has increased in the last ten years, to overcome the weakness of the classical statistical models. The core of machine learning deals with representation and generalization. Representation of data instances and functions evaluated on these instances are part of all machine learning systems. Generalization is the property that the system will perform well on unseen data instances; the conditions under which this can be guaranteed are a key object of study in the subfield of computational learning theory. Most of the cases of the data mining process classifiers are used to solve the prediction problem and which is a subset of machine learning techniques. Many variations of classical K-mean clustering algorithm (discussed in section 3.2.2) are used in the medical domain such as KMIX (Davis & Nguyen, 2007), and K-Mean (Thangavel 2006).

The design of Artificial Neural Network (Schalkoff 1997) was originally motivated by the phenomena of learning and recognition (Bishop 1995). Neural network techniques can be divided into two alternative ways of learning: supervised and unsupervised. Many different types of neural network are used in medical domain such as Multilayer Perceptrons (Aeinfar et al. 2009), Support Vector Machine (SVM)

(Guyon I 2002), and (unsupervised) Self-Organising Maps (Haykin 2009).

2.6 Fuzzy logic in data mining

The term "fuzzy logic" was introduced with the 1965 proposal of fuzzy set theory by Lotfi A. Zadeh (Zadeh 1965), is now widely used in data mining in different domains. Looking for too strict a relation between variables may be impossible because of the variability of descriptions in the data, while looking for an imprecise relation between variables or to a crisp relation between approximate values of variables may lead to a better solution. Fuzzy logic provides means to represent approximate knowledge, where other expert systems use knowledge base generated from a set of crisp samples. Fuzzy logic has been used in medical domains, for example Fuzzy ART based classification (Benkaci et al. 2010), fuzzy clustering (Mirkin and Nascimento 2012, Tutmez 2012, Wu 2012), Fuzzy-Rough sets (Jensen and Qiang 2009b, Ren and Qiang 2011, Jensen and Qiang 2009a) and much more.

Tez (2008) proved that Neuro-Fuzzy (Jang et al. 1997) systems can incorporate data from many clinical and laboratory variables to provide better diagnostic accuracy in the prediction of acute appendicitis. The basic idea of combining fuzzy systems and neural networks is to design an architecture that uses a fuzzy system to represent knowledge in an interpretable manner and the learning ability of a neural network to optimize its parameters. Adaptive Neuro-Fuzzy Inference Systems (or ANFIS) (Jang et al, 1997, chapter 12) have been used for breast cancer detection (Übeyli 2009), the diagnosis of Aphasia (Fazeli et al. 2008) and few more applications of data mining in the area of medicine (Yardimci 2009, Benkaci et al. 2010, Kochurani et al. 2007, Ubeyli and Guler 2005).

2.6.1 Application of fuzzy logic in data mining process

Fuzzy logic can be applied in different phases of data mining, for example:

Problem understanding phases: In these phases, fuzzy set methods can be used to formulate, for example, the background domain knowledge in vague manner, which can be used for the subsequent modelling phases (Bai et al. 2006).

Data preparation step: Fuzzy methods can be used to detect outliers. Fuzzy clustering can be used to cluster the data and find those data points that lay far away from the cluster. Fuzzy rules based algorithm can also be used as a missing values imputation method (see section 5.2.2).

Modelling phase:

As neural networks tend to do badly since the domain knowledge cannot be incorporated into the neural networks (Hongjun et al.

1996), fuzzy logic based models utilize the domain knowledge in coming up with rules of data selection and extraction. Fuzzy data analysis approaches can be applied to build classifiers. One kind of application is to analyze fuzzy data, which are derived from imprecise measurement instruments or from the descriptions of human domain experts.

Evaluation phase: Fuzzy modeling methods are interpretable systems. Therefore, they can easily be checked for plausibility against the intuition and expectations of human experts (Bai et al. 2006).

2.7 Issues and challenges with clinical data mining and decision support

The application of data mining, knowledge discovery and machine learning techniques to medical and health data is challenging and intriguing (Cios & Moore, 2002). The datasets usually are very large, complex, heterogeneous, and hierarchical and vary in quality. Data pre-processing and transformation are required even before mining and discovery can be applied. Sometimes the characteristics of the data may not be optimal for mining or analytic processing. The challenge here is to pre-process the data into appropriate form before any learning or mining can begin (Hosseinkhah et al. 2009).

Many researchers have identified several important and challenging issues (Sittig et al. 2008, Fox et al. 2010, Bellazzi and Zupan 2008) for clinical decision support. In ‘‘Grand challenges for decision support” Sittig et al. (2008) set out 10 critical problems for ‘‘designing, developing, presenting, implementing, evaluating, and maintaining all types of clinical decision support capabilities for clinicians, patients and consumers”.

Sittig et. all (2008) placed the grand challenges into the following three large categories:

A) Improve the effectiveness of CDS interventions

- Improve the human-computer interface
- Summarize patient-level information
- Prioritize and filter recommendations to the user
- Combine recommendations for patients with co-morbidities
- Use free-text information to drive clinical decision support

B) Create new CDS interventions

- Prioritize CDS content development and implementation
- Mine large clinical databases to create new CDS

C) Disseminate existing CDS knowledge and interventions

- Disseminate best practices in CDS design, development, and implementation
- Create architecture for sharing executable CDS modules and services
- Create internet-accessible clinical decision support repositories

However Sittig et al.’s identification covers very little about data pre-processing. Sometimes, improved data quality is itself the goal of the analysis, usually to improve processes in a production database (Dasu and Johnson 2003) and designing of decision support.

Many other researchers also mention several other issues of clinical decision support related to clinical data and data pre-processing. The ones most relevant to this thesis are as follows.

High volume of data:

Due to the high volume of the medical databases, current data mining tools may require extraction of a sample from the database (Cios and Moore 2002, Maimon and Rokach 2010). Hence good sampling techniques are needed to select data records for further analysis.

Update:

Medical databases are updated constantly by adding new results for lab tests and new ECG signals for patients (Hosseinkhah et al. 2009).

Inconsistent data representation:

Inconsistencies due to data entry errors are common problems. Inconsistencies due to data representation can exist if more than one model for expressing a specific meaning exists. Additionally, the data type does not always reflect the true data type (Cios and Moore 2002). If we consider the cardiovascular risk based on dead or alive of previous patient’s records, some patients have smellier properties of most of the dead patients but still alive and some patients may have died for some other cause.

Number of variables:

High-level information is essential to support medical diagnostic and decisions for the clinicians. The computational complexity is not linear for certain data mining techniques. In such cases, the time required may become infeasible as the number of variables grow (Tsang-Hsiang et al. 2006).

Missing/Incomplete data:

Clinical database systems do not often collect all the data required for analysis or discovery. In many clinical trials, the medical report pro-forma allow some attributes to be left blank, because they are inappropriate for some class of illness or the person providing the information feels that it is not appropriate to record the values for some attributes (Almeida et al. 2010). For example the main thesis data have 23 attributes and 823 records where 18 attributes have a missing value frequency from 1% to 30% and out of 823 records, 613 records have 4% to 56% missing values in their attributes (See appendix A).

Noise:

Medical databases include noise. This can include abbreviations in categorical attributes and outlier values in numerical attributes. As an example of numerical outlier values in the current research data, the attribute "PACK YRS" has a big gap between the maximum value of 160, and the minimum value of 2. This affects the transformation process as it unduly changes the mean of the attribute values. For different kind of example, the attribute “CAROTID_DISEASE” includes a mixture of abbreviated and fully specified values such as “asymptomatic carotid disease”, “Asx”, and so on. In fact, both these values have the same meaning (i.e. are homonyms, See appendix A). Therefore, data mining techniques should be improved to make them less sensitive to noise (Tsumoto 2000, Han and Kamber 2011).

Class Imbalance:

Most medical datasets are not balanced in regards to their class labels. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced. This is because they aim to optimize the overall accuracy without considering the relative distribution of each class (Liu et al. 2011). Therefore, there is a need of a good sampling technique for such datasets where the target classes are not balanced.

Validity of Class Label:

In some medical datasets it has been noticed that the given class labels do not accurately represent characteristics of the data record. Indeed in many cases the class labels do not accurately reflect the nature of a patient. Some patients may have same properties to be in the group of high risk patient but they are labelled as low risk as they are alive. On the other hand some patients may have died with other non-target cause. A quantitative preliminary study shows that sets acquired from clustering does not match to given outcome. This suggests that the given labels are not always appropriate.

2.8 Summary

This chapter provides a general background of decision support, data mining and data mining methodology. There are two popular existing data mining methodologies, CRISP_DM, and SEMMA. But the existing methodologies are not suitable for the thesis data. The data mining methodology adopted for this thesis is derived from the process of knowledge discovery in database (KDD) of Fayyad (1996).

The application of data mining to medical and health data is very challenging. The datasets usually are very large, complex, heterogeneous, and hierarchical and vary in quality. Sittig et. all (2008) placed the grand challenges of clinical data mining into three large categories: Improve the effectiveness of Clinical Decision

Support interventions, Create new Clinical Decision Support interventions and Disseminate existing Clinical Decision Support knowledge and interventions. However Sittig et al.'s identification covers very little about data pre-processing.

Sometimes, improved data quality is itself the goal of the analysis, usually to improve processes in a production database and designing of decision support. Many other researchers also mention several other issues of clinical decision support related to clinical data and data pre-processing. The ones most relevant to this thesis are high volume of data, data update, inconsistent data representation, number of variables, missing/incomplete data, and class imbalance. The rest of the thesis will be dealing with some of the above mention issues with data pre-processing. Problem with missing value, class imbalance and feature selection for imbalance data are discuss in details in chapter 5; over view of research case studies.

The next chapter will introduce some data mining and machine learning techniques which are used for most of the experiments in subsequent chapters.

CHAPTER 3 : CLUSTERING, CLASSIFIERS AND ALGORITHMS

3.1 Introduction

This chapter presents a background on the use of clustering and classification for data mining. Some basic methods of machine learning (see section 2.5) are introduced. Alternative evaluation metrics for classification performance are introduced in this chapter. They are: confusion matrix; accuracy; the rates of sensitivity and specificity as well as the positive predictive value and the negative predictive value. These rates are used in all thesis experiments in later chapters for discussions and comparisons.

3.2 Clustering

Clustering and classification are both fundamental tasks in Data Mining. Clustering is a form of unsupervised learning whereby a set of observations (i.e., data points) is partitioned into natural groupings or clusters of patterns in such a way that the measure of similarity between any pair of observations assigned to each cluster minimizes a specified cost function (Haykin 2009). Clustering groups data instances into subsets in such a manner that similar instances are grouped together in a cluster. Formally, the clustering structure is representation of a sets $C = C_1, \dots \dots . C_k$ of S, such that: $S = U_{i=1}^{K} C_i$ and $C_i \cap C_j = \emptyset$ for $i \neq j$ (Maimon and Rokach 2010).

Clustering is essentially the production of a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Clustering can be roughly distinguished as: *Hard clustering (Kanzawa et al. 2011)*, where each object belongs to a cluster or not; and *Soft clustering*, where each object belongs to each cluster to a certain degree (e.g. a likelihood of belonging to the cluster). Finer distinctions are possible, for example:

Strict partitioning clustering: here each object belongs to exactly one cluster

Strict partitioning clustering with outliers: an object can belong to no cluster, and so be considered an outlier.

Overlapping clustering (also: alternative clustering, multi-view clustering)(Becker et al. 2012): where objects may belong to more than one cluster.

Hierarchical clustering (Iida et al. 2010): any objects that belong to a child cluster also belong to the parent cluster.

Subspace clustering (Chen et al. 2012): where clusters are not expected to overlap.

3.2.1 Distance matrix

Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. For the objects x_i and x_j the distance d (x_i , x_j) can be determined by different distance measures. The most commonly used distance measure is known as Euclidian Distance (Lele and Richtsmeier 1995). The Euclidean distance between point's x_i and x_j is the length of the line segment connecting them.

$$d(x_i, x_j) = \sqrt{\sum_{i=1}^{n}(x_i - \quad x_j)^2} \tag{3-1}$$

Different measures are used for different types of attributes. For attributes of type binary, nominal and mixed type attributes the distance between points can be defined as follows (Maimon and Rokach 2010):

Distance Measures for Binary Attributes

$$d(x_i, x_j) = \frac{r+s}{q+r+s+t} \tag{3-2}$$

Where q is the number of attributes that equal 1 for both objects; t is the number of attributes that equal 0 for both objects and s and r are the number of attributes that are not equal for both objects.

Distance Measures for Nominal Attributes

$$d(x_i, x_j) = \frac{p-m}{p} \tag{3-3}$$

Where p is the total number of attributes and m is the number of matches.

Distance Metrics for Mixed-Type Attributes

Distance of mixed type attributes can be calculated by combining the matrixes described above. The dissimilarity $d(x_i, x_j)$ of two instances, containing *p* attributes of mixed types, can be defined as (Maimon and Rokach 2010):

$$d(x_i, x_j) = \frac{\sum_{n=1}^{p} \delta_{ij}^{(n)} d_{ij}^{(n)}}{\sum_{n=1}^{p} \delta_{ij}^{(n)}} \qquad (3\text{-}4)$$

Where the indicator $\delta_{ij}^{(n)} = 0$ if one of the values is missing. The contribution of attribute n to the distance between the two objects $d^{(n)}(x_i, x_j)$ is computed according to its type:

- If the attribute is binary or categorical $d^{(n)}(x_i, x_j) = 0$ if $x_{in} = x_{jn}$, otherwise $d^{(n)}(x_i, x_j) = 1$.
- If the attribute is continues-valued, $d_{ij}^{(n)} = \frac{|x_{in} - x_{jn}|}{\max_h x_{hn} - \min_h x_{hn}}$, where *h* runs over all non-missing objects for attribute *n*.

More on different distance measures can be found in (Maimon and Rokach 2010).

3.2.2 K-Means clustering

K-Means is one of the simplest unsupervised learning algorithms, first proposed by Macqueen in 1967. It is used by many researcher to solve some well-known clustering problems (Han and Kamber 2001). The clustering procedure follows a simple algorithm to classify a given data set through a certain number of clusters (assume *k* clusters). The algorithm first randomly initializes the clusters center. The next step is to calculate the distance (discussed in the above section) between an object and the centroid of each cluster; then take each point belonging to a given data set and associate it to the nearest centre and recalculate the cluster centres. The process is repeated with the aim of minimizing a squared error objective function given by:

$$J(v) = \sum_{i=1}^{K} \sum_{j=1}^{C_i} (||x_{ij} - v_j||)^2 \qquad (3\text{-}5)$$

Where,

x_{ij} *is the j*th *point in the cluster*

v_i *is the i*th *cluster*

$||x_{ij} - v_j||$ *is the Euclidean distance between* x_{ij} *and* v_j

C_i *is the number of data points in* i^{th}*cluster.*

K is the number of cluster.

Algorithm 3.1: k-means clustering

```
Let $x = \{x_1,,,,,,,x_n\}$ be the set of data points and $V = \{v_1,,,,,,v_n\}$ be the set of centers.
Step 1: randomly select the cluster centres c1….ck
Step 2: calculate the distance between each data point and the cluster
        centres using some distance matrix (commonly Euclidean distance is used).
Step 3: assign the data points to the cluster centre whose distance from the cluster centre is minimum of all the cluster centres.
Step 4: recalculate the new cluster centre using
        $v_i = ({}^{1}/_{c_i}) \sum_{j=1}^{c_i} x_i$        (3-6)
Step 5: recalculate the distance between each data point and newly
        obtained cluster centres.
Step 6: if no data point was reassigned then stop, otherwise repeat from
        step 3.
```

3.3 Classification

Classification is a supervised learning process (Kononenko and Kukar 2007) that maps the input space into predefined classes. For instance, a classifier can be used to classify a cardiovascular patient as high risk patient or low risk patient based on given information. There are many alternative classifiers, for example, Decision Tree, k-nearest neighbour algorithm, artificial neural networks, support vector machine etc.

3.3.1 Decision Tree

A decision tree is a classifier expressed as recursive partition of the instance space. The decision tree consists of nodes that branch within a rooted tree. It starts with a root at the top that has no incoming edges. A node with outgoing edges is called an internal node, and all the other nodes are called leaves, also known as decision nodes. Each leaf is assigned to one class representing the majority target value at that node.

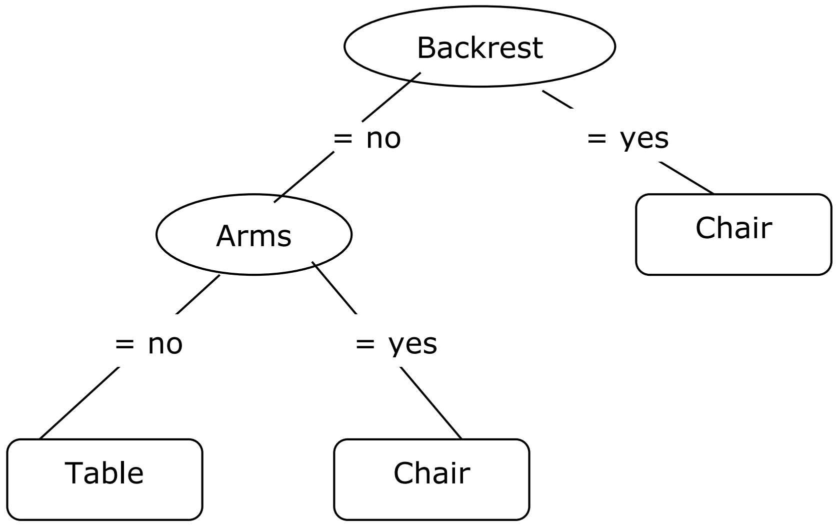


Figure 3-1: Decision tree presenting the classification of chair and table.

Decision tree inducers are algorithms that automatically construct a decision tree from a given dataset. Typically the goal is to find the optimal decision tree by minimizing the generalization error (Maimon and Rokach 2010) The earliest decision trees were ID3 and subsequently C4.5. The algorithms introduced by Quinlan (1985,

1993) have proved to be an effective and popular method for finding a decision tree to express information contained implicitly in a data set. WEKA (Bouckaert et al. 2010) that was used for thesis experiments has the implementation of C4.5 algorithm called J48. A brief description of the WEKA is given in the section 3.7.

Algorithm 3.2 represents a typical algorithm for generating a decision tree (Han and Kamber 2001). The algorithm is called with three parameters: *D*, *attribute_list*, and *Attribute_selection_method*. *D* is referred to as a data partition. Initially, *D* is the complete set of training tuples and their associated class labels (input training data). The parameter attribute_list is a list of attributes describing the tuples. *Attribute_selection_method* specifies a heuristic procedure for selecting the attribute that "best" discriminates the given tuples according to class. *Attribute_selection_method* procedure employs an attribute selection measure, such as Information Gain (Quinlan 1985) or the Gini Index (Park and Kwon 2011). Information gain is an impurity based criterion that uses the entropy measure as the impurity measure. The expected information gain is the change in information entropy H from a prior state to a state that takes some information. Let T denote a set of training examples, each of the form $(X,y) = (x_1, x_2, x_3, \ldots, x_k, y)$ where $x_a \in vals\ (a)$ is the value of the a^{th} attribute of example X and y is the corresponding class label. The information gain for an attribute a is defined in terms of entropy $H( )$ as follows:

$$IG(T,a) = H(T) - \sum_{v\in vals(a)} \frac{|\{\mathrm{X} \in T|x_a = v\}|}{|T|} . H\ (\{X \in T|x_a = v\}) \quad (3\text{-}7)$$

Where: $H(T)$ of discrete random variable *T* with possible values (t$_1$, t$_2$,…t$_n$} and probability mass function $P(T)$ as:

$$Entropy = H(T) = \sum_i P(t_i)I(t_i) = -\sum_i P(t_i)\log\ P(t_i) \quad (3\text{-}8)$$

Algorithm 3.2: Decision Tree Induction

```
Input: Data partition, D; Attribute_list: the set of candidate attributes
Attribute_selection_method: a procedure to determine the splitting criterion
that "best" partitions the data tuples into individual classes.
Step 1: Create a node N;
Step 2: If tuples in D are all of the same class, C then return N as a leaf
          node labelled with the class C;
Step 3: If attribute_list is empty then return N as a leaf node labelled with
          the majority class in D;
Step 4: Apply attribute_selection_method (D, arrtibute_list) to find the
          "best" splitting_criterion;
Step 5: Label node N with splitting_criterion;
Step 6: If splitting_attribute is discrete-valued and Multi way splits allowed
          then not to binary trees and remove splitting_attribute.
Step 8: Partition the tuples and grow sub-tees for each partition
Step 9: Let Dj be the set of a data tuples in D satisfying outcome j;
Step 10: If Dj is empty then
Step 11: Attach a leaf labelled with the majority class in D to node N;
Step 12: Else attach the node returned by Generate_decision_tree (Dj,
           attribute list) to node N;
Step 13: Return N;
```

3.3.2 K Nearest Neighbour algorithm (KNN)

K Nearest Neighbour Algorithm (KNN) is a method for classifying objects based on closest training examples in the feature space. KNN is a type of instance-based learning (Aha et al. 1991), or lazy learning where the function is only approximated locally and all computation is deferred until classification. The K Nearest Neighbour algorithm is amongst the simplest of all machine learning algorithms where an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its k nearest neighbours (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbour.

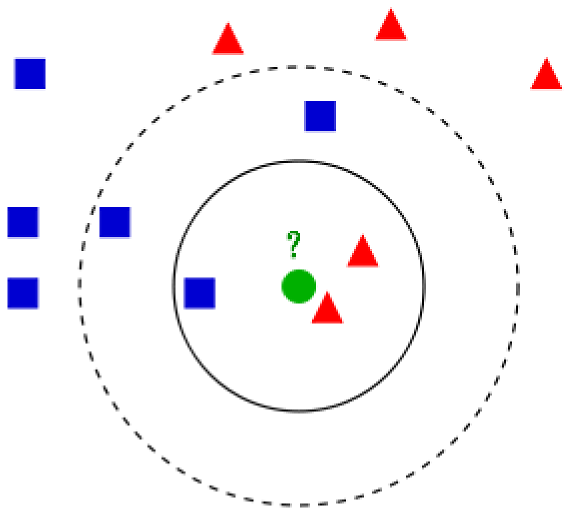


Figure 3-2: KNN Classification

IBL algorithms introduced by Aha et al. (1991) are the early de veloped instance based nearest neighbour algorithms. The IBL algorithms assume that similar instances have similar classifications.

This leads to their local bias for classifying novel instances according to their most similar neighbour's classification.

IBL algorithms differ from many other supervised learning methods: they do not construct explicit abstractions such as decision trees or rules. Most learning algorithms derive generalizations from instances when they are presented and use simple matching procedures to classify subsequently presented instances. Three versions of IBL algorithm IBL1, IBL2, and IBL3 proposed by Aha et al. (1991).

The IB1 algorithm, described below, is the simplest instance-based learning algorithm. The similarity function used here is:

$$Similarity\,(\mathrm{x},\mathrm{y}) = -\sqrt{\sum_{i=1}^{n} f(x_i, y_i)} \qquad \text{(3-9)}$$

Where the instance described by n attributes and $f(x_i, y_i) = (x_i - y_i)^2$ *for numeric-valued attributes and* $f(x_i, y_i) \neq (x_i \neq y_i)$ *for Boolean and symbolic-valued attributes.*

Algorithm 3.3: The IB1 algorithm

```
Step 1: Concept Description (CD)← ∅
Step 2: For each $x \epsilon$ Training set do
Step 3:     For each $y \epsilon$ CD do
Step 4:         Sim[y] ← Similarity(x,y)
Step 5:         y_max ← some $y \epsilon$ CD with maximal Sim[y]
Step 6:         if class(x) = class(y_max)
Step 7:             then classification ← correct
Step 8:         else classification ← incorrect
Step 9:       CD ← CD U $\{x\}$
```

Some advance work on k Nearest neighbour algorithm can be found in the work of Rasheed et al. (2006), Liao and Li (1997), Meesad and Hengpraprohm (2008) and Kissiov and Hadjitodorov (1976).

3.3.3 Artificial neural network (ANN)

An Artificial Neural Network (ANN) is an information processing paradigm that is a mathematical model inspired by the way biological nervous systems work, such as the human brain. In 1943, McCulloch and Pitts first introduced the idea of neural networks as computing machine and in 1958 Rosenblatt proposed supervised learning artificial neural network model by proposing *perceptron.*

The perceptron is the simplest form of a neural network used for classification of patterns. Basically, it consist of inputs, adjustable synaptic weights, bias and activation function (Haykin 2009).

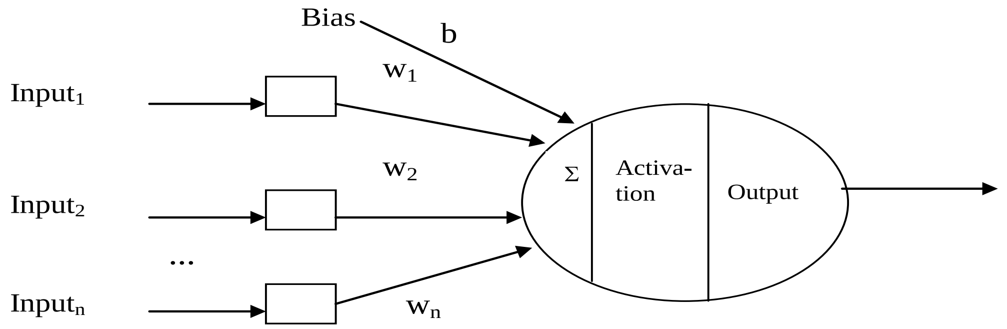


Figure 3-3: Single-flow graph of the perceptron

The synaptic weights of the perceptron are denoted by $w_1, w_2, \dots \dots w_n$. Correspondingly, the inputs applied to the perceptrons are denoted by $x_1, x_2, \ldots \ldots x_n$. The externally applied bias is denoted by *b.* The decision boundary for a perceptron is given by:

$$y = f\left(\sum_{j=1}^{n} x_j w_j + b\right) \tag{3-10}$$

Single layer perceptron are only capable of learning linearly separable patterns; in 1969 a famous book entitled Perceptrons by Marvin Minsky and Seymour Papert (Minsky 1969) showed that it was impossible for these classes of network to learn an XOR function. It is often believed that they also conjectured (incorrectly) that a similar result would hold for a multi-layer perceptron network. However, this is not true, as both Minsky and Papert already knew that multi-layer perceptrons were capable of producing an XOR function.

A multilayer perceptron is formed through the combination of multiple perceptrons in separate layers. There are three main points that highlights the basic feature of multilayer perceptron are as follows:

- The model of each neuron in the network includes a nonlinear activation function.
- The network contains one or more layers that are hidden from both the input and output.
- The network exhibits a high degree of connectivity, the extent of which is determined by synaptic weight of the network.

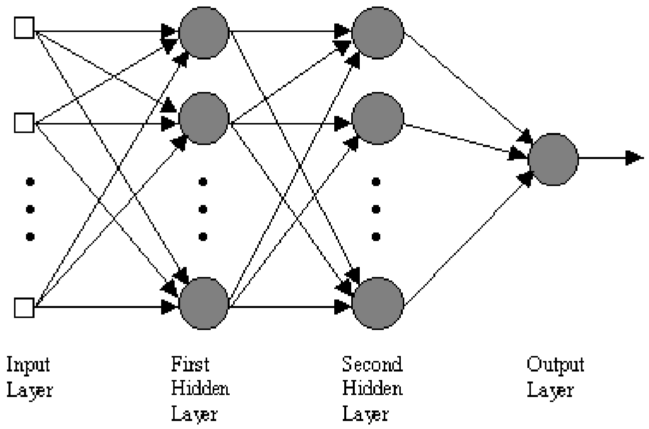


Figure 3-4: An example of a multilayer perceptron with two hidden layers.

The back propagation algorithm is one of the well-known learning algorithms for multilayer perceptron artificial neural network models. In order to understand the algorithm let us take an example of one connection initially, between a neuron in the output layer and one in the hidden layer shown in Figure 3.4 and the algorithm is presented in Algorithm 3.4.

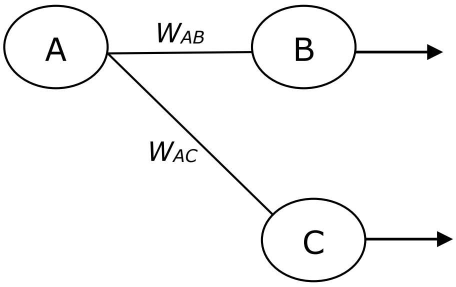


Figure 3-5: Single connection learning in a Back Propagation network

Algorithm 3.4: Back Propagation Algorithm

Step 1: First apply the inputs to the network and work out the output – where the initial output could be anything, as the initial weights are random numbers.

Step 2: Next work out the error for neuron B. The error is What we want – What we actually get, in other words: ErrorB = OutputB (1-OutputB)(TargetB – OutputB)

Step 3: Change the weight. Let W+AB be the new (trained) weight and WAB be the initial weight. W+AB = WAB + (ErrorB x OutputA)

Step 4: Calculate the Errors for the hidden layer neurons. Back Propagate the error from the output layer.

Step 5: Having obtained the Error for the hidden layer neurons now proceed as in step 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers.

3.3.4 Support vector machine (SVM)

Support Vector Machines implement complex decision rules by using a nonlinear function ϕ to map training points to a high dimensional feature space where the labelled points are separable (Haykin 2009). A separating hyperplane is found which maximizes the distance between itself and the nearest training points - this distance is called the margin. Assume that a pattern data set can be described in an m dimensional feature space. The idea of a support vector machine is to build a hyperplane to separate the positive and the negative patterns in a given data set. This hyperplane can be seen as a decision surface. The training points that are nearest to this hyperplane can be seen as support vectors (see Figure 3.6).

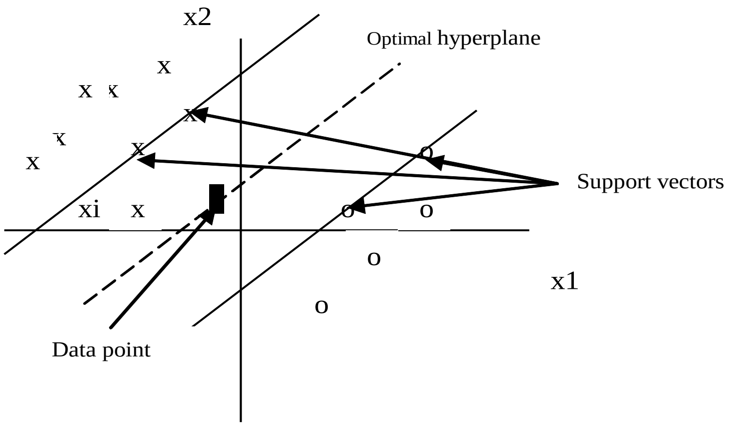


Figure 3-6: The description of support vectors

The key to understanding support vector machines is to see how it produces optimal hyperplanes to separate the patterns. According to Haykin (2009), two operations to build a support vector machine can be summarized as:

- Map data to higher dimensional space: It is a nonlinear mapping based on Cover's theorem (Haykin 2009). This means the following two conditions need to be satisfied:
  - The transformation is nonlinear;
  - And the dimensionality of the feature space is high enough.
- Construct an optimal hyperplane to separate the patterns: This construction is based on the use of an inner-product kernel to define a linear function separating the vectors in feature space. Therefore, the hyperplane can be formed as:

$$\sum_{j=1}^{m} w_j \varphi_j(x) + b = 0 \tag{3-11}$$

Where, x is a vector in input space, $\{\varphi_j(x)\}_{j=1}^{m}$ *is a set of nonlinear transformation vectors in feature space,* w_j *are the vector weights, and b is the bias.*

More details on how support vector machine works and simple SVM algorithm can be find in Vishwanathan et al (2002).

3.3.5 Ripple-down rule (Ridor)

The ripple-down rule (Ridor) method is an expert system methodology with its origin in the medical expert system GARVAN-ESI (Horn 1990). The basis of the method is the maintenance and retrieval of cases. When a case is incorrectly retrieved, the expert identifies how a case stored in a Knowledge Base System (KBS) differs from the present case. The ripple-down rule technique creates a two way dependency relation between rules such that rule activation is investigated only in the context of other rule activation. If the premise of a parent rule is true for a particular individual then, if has no dependents, its conclusion will be asserted for that individual. If, however, it has an 'if-true' dependent then that rule, and its dependents, will be tested, and the original conclusion will only be asserted if the premises of none of them are true for the entity. Conversely, if the premise of a parent rule is false for a particular individual, then not only will its conclusion not be asserted but also, if it has an ‘if false’ dependent then it, and its dependents, will be tested (Brian et al. 1995).

3.4 Fuzzy logic and fuzzy based classifiers

3.4.1 Fuzzy logic

Fuzzy Logic was initiated in 1965, and developed as a completely new, elegant approach to vagueness called *fuzzy set theory (Zadeh 1994)*. In this approach an element belongs to a set to a degree *k* (0 ≤ k ≤ 1), in contrast to classical set theory where an element must definitely belong or not to a set. For example, a classical set “*A*” is a fuzzy set and which is a set without a crisp boundary. The transition from “belong to a set” to “not belong to a set” is gradual with fuzzy sets, and this smooth transition is characterized by membership functions (Jang et al. 1997).

3.4.2 Fuzzy sets and membership functions

Let *X* be a space of objects and *x* be a generic element of *X*. By defining a characteristic function for each element *x* in *X*, a classical set A can be represented by a set of ordered pairs (*x,* 0) or (*x, 1*), which indicates $x \notin A$ or $x \in A$, respectively. On the other hand the fuzzy set (Zadeh 1996) express the degree to which an element belongs to a set. Hence the characteristic function of a fuzzy set is allowed to have values between 0 and 1, which denotes the degree of membership of an element in a given set.

If X is a collection of objects denoted generically by *x,* then a fuzzy set *A* in *X* is defined as a set of ordered pairs:

$$A = \{x, \mu_A(x) | x \in X\} \qquad (3\text{-}12)$$

Where $\mu_{A(x)}$ is called the membership function for the fuzzy set A. The membership function maps each element of x to a membership grade between 0 and 1. Various types of membership functions are used, including triangular, trapezoidal, generalized bell shaped, Gaussian curves, polynomial curves, and sigmoid functions more details Jang et al. (1997).

3.4.3 Fuzzy set operations

Fuzzy set operations are similar to crisp set operations. The elementary crisp set operations are union, intersection, and complement, which in effect correspond to *OR, AND,* and *NOT* operators, respectively (Rosen 1999).

Union:

The union of two fuzzy sets *A* and *B* is a fuzzy set *C*, written as *C* $= A \cup B$ or C= $A \text{ OR } B$, who's membership function can be written as:

$$\mu_C(x) = max(\mu_A(x), \mu_B(x)) = \mu_A(x) V \mu_B(x) \qquad (3\text{-}13)$$

Intersection:

The intersection of two fuzzy sets *A* and *B* is a fuzzy set *C*, written as *C* $= A \cap B$ or C= A AND B, who's membership function can be written as:

$$\mu_C(x) = min\big(\mu_A(x), \mu_B(x)\big) = \mu_A(x) \wedge \mu_B(x) \qquad (3\text{-}14)$$

Complement:

The complement of a fuzzy sets *A*, denoted by Ā and membership function can be defined as:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \qquad (3\text{-}15)$$

T-norm (triangular norm):

T-norm (also t-norm or, unabbreviated, triangular norm) is a kind of binary operation used in the framework of probabilistic metric spaces and in multi-valued logic, specifically in fuzzy logic. A t-norm generalizes intersection in a lattice and conjunction in logic. The intersection of two fuzzy sets A and B is specified in general by a function

$\mathrm{T} : [0,1] \mathrm{x}\, [0,1] \rightarrow [0,1]$, which aggregates two membership grade as follows (Jang et al. 1997)

$$\mu_{A \cap B}(x) = T\ (\mu_A(x), (\mu_B(x) * \mu_A(x) \qquad (3\text{-}16)$$

3.4.4 Fuzzy logic based classifiers

A classical fuzzy classifier consists of rules, each one describing one of the classes; in some cases each rule can represent more than one class with different probabilities (Marsala 2009). More recently, fuzzy classifier methods based on if-then rules have been applied to solve classification problems by constructing multi-model structures, which yield a class label for each vector in the given space. Let X be a vector in an n-dimensional real space R^n (the feature space), and $\Omega = \{w_1, \dots \dots, w_{c1}\}$ be a set of class lables. A (crisp) classifier is given by the mapping,

$$D: R^n \to \Omega \tag{3-17}$$

A fuzzy classifier is any classifier which uses fuzzy sets either during its operation and will have the label mapping (Kuncheva 2000):

$$D_P: R^n \to [0,1]^c - 0 \tag{3-18}$$

Instead of assigning a class label from Ω, D_p assigns to $x \in R^n$ a soft class label with degrees of membership in each class.

There are many fuzzy logic based classifiers proposed by different researchers and most of them are fuzzy ruse based classifiers such as Fuzzy Decision Tree (Papageorgiou et al. 2008), Fuzzy Unordered Rule Induction Algorithm (FURIA) (Lotte et al. 2007), and Fuzzy Lattice Reasoning Classifier (FLR) (Kaburlasos et al. 2009).

3.4.5 Fuzzy unordered rule induction algorithm (FURIA)

The fuzzy rule-based classification method called Fuzzy Unordered Rule Induction Algorithm (Lotte et al. 2007), or FURIA for short, is a modification and extension of the state-of-the-art rule learner RIPPER (Brian et al. 1995).

Representation of fuzzy rules in FURIA:

A selector constraining a numerical attribute A_i (with domain D_i = R) in a RIPPER rule can obviously be expressed in the form $(A_i \in I\ )$, where $I \subseteq$ R is an interval: $I = (-\infty, v]$ if the rule contains a selector $(A_i \leq v)$, $I = [u,\infty)$ if it contains a selector $(A_i \geq u)$, and $I = [u, v]$ if it contains both (in the last case, two selectors are combined).

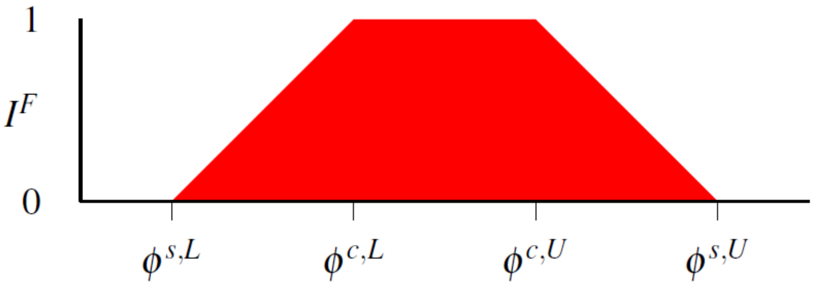


Figure 3-7: A fuzzy interval I^F

Essentially, a fuzzy rule is obtained through replacing intervals by fuzzy intervals, namely fuzzy sets with trapezoidal membership function. A fuzzy interval of that kind is specified by four parameters and will be written:

I F = ($\varphi^{s,L}$, $\varphi^{c,L}$, $\varphi^{c,U}$, $\varphi^{s,U}$):

$$I^F(v) \overset{\text{def}}{=} \begin{cases} 1 & \varphi^{c,\mathrm{L}} \le v \le \varphi^{c,U} \\ \frac{v-\varphi^{s,L}}{\varphi^{c,L}-\varphi^{s,L}} & \varphi^{s,\mathrm{L}} < v < \varphi^{c,\mathrm{L}} \\ \frac{\varphi^{s,U}-v}{\varphi^{s,U}-\varphi^{c,U}} & \varphi^{c,\mathrm{U}} < v < \varphi^{s,U} \\ 0 & else \end{cases} \tag{3-19}$$

$\varphi^{c,L}$ and $\varphi^{c,U}$ are, respectively, the lower and upper bound of the core (elements with membership 1) of the fuzzy set; likewise, $\varphi^{s,L}$ and $\varphi^{s,U}$ are, respectively, the lower and upper bound of the support (elements with membership > 0).

Note that, as in the non-fuzzy case, a fuzzy interval can be open to one side ($\varphi^{s,L} = \varphi^{c,L} = -\infty$ or $\varphi^{c,U} = \varphi^{s,U} = \infty$).

A fuzzy selector *($A_i \in I^F_i$) covers* an instance ***x*** = *(x*1 . . . *xn)* to the degree *I^F_i (x_i)*.

A fuzzy rule *r F* involving *k* selectors *($A_i \in I^F_i$) i* = 1 . . . *k*, covers ***x*** to the degree

$$\mu_{rF}(x) = \prod_{i=1\ldots k} I_i^F(x_i) \tag{3-20}$$

Rule fuzzification:

To obtain fuzzy rules, the idea is to fuzzify the final rules from the modified RIPPER algorithm. More specifically, using the training set *D*T ⊆ D for evaluating candidates, the idea is to search for the best fuzzy extension of each rule, where a fuzzy extension is understood as a rule of the same structure, but with intervals replaced by fuzzy intervals. Taking the intervals L_i of the original rules as the cores [*φc,Lᵢ , φc,Uᵢ*] of the sought fuzzy intervals L_F , the problem is to find optimal bounds for the respective supports, i.e., to determine *φs,Lᵢ* and *φs,U* .

For the fuzzification of a single antecedent *(Ai ∈ Ii)* it is important to consider only the relevant training data *Di*T, i.e., to ignore those instances that are excluded by any other antecedent.

$$D_T^i\{x = (x_1 \ldots\ldots x_k) \in D_T \left|I_j^F(x_j)\right. > 0\, for\, all\, j \neq i\} \subseteq D_T \qquad (3\text{-}21)$$

D^i_T is partitioned into the subset of positive instances, D^i_T+, and negative instances, D^i_T −. To measure the quality of a fuzzification, the rule purity will be used.

Rules are fuzzified in a greedy way, as presented in algorithm 3.5. In each iteration, a fuzzification is computed for every antecedent, namely the best fuzzification. This is done by testing all values.

Classifier output:

Suppose that fuzzy rules $r^{(j)}_1 \ldots r^{(j)}_k$ have been learned for class λ_j . For a new query

instance ***x***, the support of this class is defined by:

$$s_j(x) \overset{\text{def}}{=} \sum\nolimits_{i=1\ldots k} \mu_{r_i^{(j)}}(x).CF(r_i^{(j)}) \qquad (3\text{-}22)$$

where CF($r^{(j)}_i$) is the certainty factor of the rule $r^{(j)}_i$. It is defined as follows:

$$CF\left(r_i^{(j)}\right) = \frac{2\frac{|D_T^{(j)}|}{D_T} + \sum_{x \in D_T^{(j)}} \mu_{r_i^{(j)}}(x)}{2 + \sum_{x \in D_T^{(j)}} \mu_{r_i^{(j)}}(x)} \qquad (3\text{-}23)$$

where D(j)T denotes the subset of training instances with label λj

The class predicted by FURIA is the one with maximal support. In the case where *x* is not covered by any rule, which means that *s j (x)* = 0 for all classes λ_j , a classification decision is derived in a separate way. In the case of a tie, a decision in favor of the class with highest frequency is made.

Algorithm 3.5: The Antecedent Fuzzification Algorithm For A Single Rule r

```
Step 1: Let A be the set of numeric antecedents of r
Step 2: while A = ∅ do
Step 3:     $a_{max}$ ←null [$a_{max}$ denotes the antecedent with the highest
                purity]
Step 4:     $pur_{max}$ ←0 [$pur_{max}$ is the highest purity value, so far]
Step 5:     for $i$ ← 1 to size(A) do
Step 6:         compute the best fuzzification of A[i ] in terms of purity
Step 7:         $pur_{A[i]}$ ←be the purity of this best fuzzification
Step 8:         if $pur_{A[i]}$ > $pur_{max}$ then
Step 9:             $pur_{max}$ ← $pur_{A[i]}$
Step 10:        $a_{max}$ ← A[i]
Step 11:        end if
Step 12: end for
Step 13: A ← A \ $a_{max}$
Step 14: Update $r$ with $a_{max}$
Step 15: end while
```

3.5 Training testing and validation

For classification problems, the performance of a model is measured in terms of its error rate: percentage of incorrectly classified instances in the data set. A model is built because it can be used for classify new data. Hence we are chiefly interested in model performance on new (unseen) data.

A *training set* (seen data) is used to build the model (determine its parameters) and the *test set* (unseen data) to measure its performance (holding the parameters constant). Sometimes, a validation set is also needed to tune the model (e.g., for pruning a decision tree). The validation set cannot be used for testing (as it is not unseen). All three data set have to be representative samples of the data that the model will be applied to.

Cross Validation:

k-Fold cross validation is used to minimize the bias associated with random sampling of training and test data samples in comparing predictive accuracy of two or more methods (Olson and Delen 2008). Here the whole data set is randomly split into *'k'* mutually exclusive subsets of approximately equal size. Classification model is trained and tested k times. Each time it is trained on all but one fold. For example, if we use 10-fold cross validation, data will be spited into 10 mutually exclusive subsets using stratified sampling (Tantan et al.

2010). Each of these 10 folds will be used once to test performance of the classifier, while other 9 are used for training. Cross validation estimate of classifier's overall accuracy is calculated by simply taking the mean of '*k*' individual accuracy measures.

3.6 Classifier performance

This section will describe the classifier performance evaluation criteria used in most of the thesis case studies and experiments. They are: confusion matrix; accuracy *(ACC)*; sensitivity *(Sen)*; specificity *(Spec)* rates, and the positive predicted value *(PPV)* and negative predicted value *(NPV).*

3.6.1 Confusion matrix

In the field of machine learning (see section 2.5), a confusion matrix (Witten and Frank 2011), also known as a contingency table or an error matrix (Minsky 1969) , is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

Assume that the cardiovascular classifier output set includes two risk prediction classes: “High risk”, and “Low risk”. Each pattern x_i *(i=1, 2..n)* is allocated into one element from the set (*P, N)* of the risk prediction classes.

		Predicted classes	
Expected/ Actual Classes		High risk	Low risk
	High risk	TP	FN
	Low risk	FP	TN

Figure 3-8: Confusion Matrix.

Hence, each input pattern are mapped into one of four possible outcomes such as t*rue positive- true high risk (TP)*- when the outcome is correctly predicted as *High risk; true negative- true low risk (TN)*- when the outcome is correctly predicted as *Low risk; false negative-false Low risk (FN)*- when the outcome is incorrectly predicted as *Low risk*, when in fact it is *High risk* (positive); or *false positive- false high risk (FP) -* when the outcome is incorrectly predicted as *High risk*, when in fact it is *Low risk* (negative). The set of (*P, N)* and the predicted risk set can be built as a *confusion matrix* (Witten and Frank 2011).

From the confusion matrix in figure 3.7, the number of correct or incorrect (misclassification) patterns can be derived. The numbers along the major diagonal (from left to right) represent the correct while the rest represent the errors (confusion between the various classes).

Performance measures:

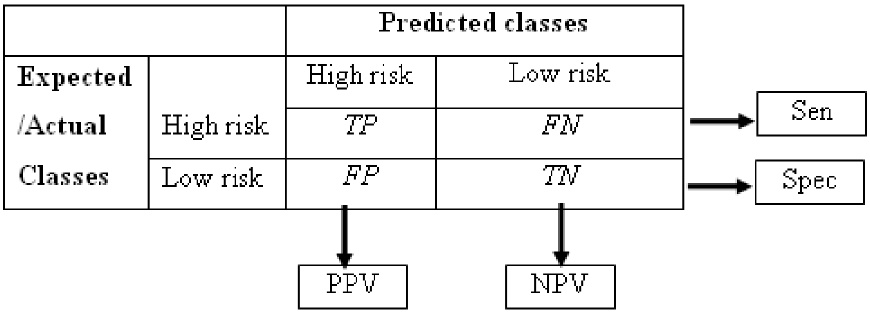
Accuracy (ACC), sensitivity (*SEN*), specificity (*SPEC*) rates, and the positive predictive value (PPV or precision), and the negative predictive value (NPV) can all be built from the confusion matrix (Witten and Frank 2011). These rates are used to evaluate and discuss classification performance.

The accuracy of a classifier is calculated by the total number of correctly predicted "High risk" (*true positive- true High risk*) and correctly predicted "Low risk" (*true negative- true Low risk)* over the total number of classifications. It is given by:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{3-24}$$

The error rate of performance, or misclassification rate, can be referred from this accuracy rate as: *1- ACC.*

However, the accuracy does not show how well the classifier can predict the positive ("High risk") and the negative ("Low risk") for the classification process. Therefore, the sensitivity, specificity, positive predictive value, and negative predictive value need to be calculated.



		Predicted classes	
Expected /Actual Classes		High risk	Low risk
	High risk	*TP*	*FN*
	Low risk	*FP*	*TN*

Figure 3-9: Classification Performance Rates.

The sensitivity is the rate of number correctly predicted "High risk" (*true positive- true high risk*) over the total number of correctly predicted "High risk" and incorrectly predicted "Low risk" (*false negative- false Low risk*). This rate can be seen as the rate of correctly predicted "High risk" over the total of expected/actual "High risk".

$$Sen = \frac{TP}{TP + FN} \tag{3-25}$$

The specificity rate is the rate of correctly predicted "Low risk" over the total number of expected/actual "Low risk". It is given by:

$$Spec = \frac{TN}{TN + FP} \tag{3-26}$$

The positive predictive value is the proportion of correct "High risk" over the total number of predicted "High risk" (including correct "High risk" and incorrect "High risk" after classification process). It is given by:

$$PPV = \frac{TP}{TP + FP} \tag{3-27}$$

The negative predictive value is the proportion of correct "Low risk" over the total number of predicted "Low risk" (including correct "Low risk" and incorrect "Low risk" after classification process). It is given by:

$$NPV = \frac{TN}{TN + FN} \tag{3-28}$$

3.7 WEKA software tool

WEKA (WEKA 1999) is an open source machine learning workbench developed using Java programming that supports many activities of machine learning practitioners. WEKA contains implementations of algorithms for classification, clustering, and association rule mining, along with graphical user interfaces and visualization utilities for data exploration and algorithm evaluation. Bouckaert et al. (2010) give an overview of the system; more comprehensive sources of information are Witten and Frank's book *Data Mining (2011)* and the user manuals included in the software distribution. Online sources, including the WEKA Wiki pages2 and the API, provide the most complete coverage. The *wekalist* mailing list is a forum for discussion of WEKA related queries, with nearly 3000 subscribers.

3.8 Summary

A general background on different data mining and machine learning techniques are given in this chapter. The literature is very useful and will help to understand the case studies described in chapter 5. The next chapter will introduce dimension reduction and feature selection.

CHAPTER 4 : DIMENSION REDUCTION AND FEATURE SELECTION

4.1 Introduction

Data mining algorithms are computationally intensive and the functional cost is correlated with the time required to run the algorithm and the size of the data set. Data mining processes bring a high computational cost when dealing with large datasets. Reducing dimensionality can effectively cut this cost (Maimon and Rokach 2010). There are two categories of techniques for dimensionality reduction: (a) feature extraction and (b) feature selection. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the original features. In this chapter two methods for dimension reduction are discussed: an instance base approach to feature selection algorithm *Relief*; and a feature selection method based on information gain. Both are used in our case study.

4.2 Feature extraction

Feature extraction is a process of finding new features that are calculated as a function of the original features. In this context the dimensionally reduction is a mapping from a multidimensional space into space of fewer dimensions (Kononenko and Kukar 2007) - see figure 4.1. Feature extraction methods can be classified as linear or nonlinear (Blum et al. 2013).

$$\begin{bmatrix}X_1\\X_2\\.\\:\\.\\X_N\end{bmatrix} \xrightarrow{\textit{Feature Extraction}} \begin{bmatrix}Y_1\\Y_2\\.\\:\\.\\Y_M\end{bmatrix} = f\left(\begin{bmatrix}X_1\\X_2\\.\\:\\.\\X_N\end{bmatrix}\right)$$

Figure 4-1: Feature extraction

where M << N.

The problem of feature extraction can be stated as:

Given a feature space $X_i R^N$ find a mapping $Y = f(x): R^N \to R^M$ with M<N such that the transformed feature vector $Y_i \in R^M$ preserves (most of) the information or structure in R^N.

Principal component analysis (PCA) is the most commonly used feature extraction technique. PCA is a linear transformation that chooses a new coordinate system for the data such that the greatest variance by any projection of the data set lies on the first axis, the second greatest variance on the second axis, and so on (Kononenko and Kukar 2007). The number of principal components is less than or equal to the number of original variables (Jolliffe 2002). There have been several other successful methods for feature extraction developed and implemented (Preece et al. 2009) such as, independent component analysis (Hyvarinen 2001), factor analysis (Schiepers et al. 2002), and nonlinear component analysis (Reddy et al. 2012). Other methods can be found in Guyon (2006).

However, in some situations feature extraction may not be feasible. This often happens when the attributes are strongly correlated amongst themselves. Moreover feature extraction processes do not provide a meaningful result, as it maps a multidimensional space into space of fewer dimensions. Furthermore, the results often cannot be used for devising a decision support system, and often it is not appropriate to reduce the dimension for creating the prediction models, where the labels associated with the attributes are just as important as the final result.

4.3 Feature selection

Feature selection, is a process closely related with dimension reduction. The objective of feature selection is to identify features in the dataset as important and discard any other feature as irrelevant, providing only redundant information.

Given a specific classification analysis task, the features employed to describe each training instance may be relevant or irrelevant to the target task. Thus, an important yet challenging issue is the selection of an appropriate subset of the available features so that the selected subset can adequately model the target task (i.e., relationships between feature values and classes)(Guyon 2006). However, the advantages of feature selection techniques come at a certain price, as the search for a subset of relevant features introduces an additional layer of complexity in the modelling task. Instead of just optimizing the parameters of the model for the full feature set, the need is to find the optimal model parameters for the optimal feature subset, as there is no guarantee that the optimal parameters for the full feature set are equally optimal for the optimal feature subset (Daelemans et al. 2003).

Generally, the approaches of feature selection can be divided into three types: filters, wrappers and embedded methods. These approaches differ in three ways i.e. search strategies, evaluation criterion definition (e.g. relevance index or prediction of classifiers), evaluation and criterion estimation (statistical test or cross validation/performance bounds) (Guyon et al. 2003).

4.3.1 Filter Method

Filters independently measure the relevance of feature subsets to classifier outcomes where each feature is evaluated with a measure such as the distance to outcome classes. All features in the data set are then ranked according to these measures. The first *m* features, from the ranked list, can be chosen by the user (Lazar et al. 2012). Filters estimate a relevance index for each feature to measure how relevant a feature is to the target. Then filters rank features by their relevance indices and perform search according to the ranks or based on some statistical criterion e.g. significance level. The most distinguishing characteristic of filters is that the relevance index is calculated based solely on a single feature without considering the values of other features (Lazar et al. 2012). Such implementation implies that filters assume orthogonally between features which usually is not true in practice. Therefore, filters omit any conditional dependence (or independence) that might exist, which is known to be one of the weaknesses of filters, since they might miss optimal subsets of features. However, filters are efficient and prove to be robust to overfitting (Ng 1998).

There are various heuristics to design relevance indices for filters, including correlation based (e.g. Pearson coefficient, signal to noise ratio) (Guyon et al. 2003), univariate prediction error rate (i.e. evaluate the relevance of a feature as how accurate the prediction is using only itself) (Pourahmadi 1993), information theory (mutual information, Minimum Description Length (MDL)) (Peng et al. 2005), and Relief (Kira and Rendell 1992). Most of the heuristics are derived from their relations to the bounds of Bayes errors (Tumer and Ghosh 1996) of a single feature. On the other hand, they differ in how to use data to evaluate the usefulness of a single feature.

Advantages of filter techniques are that they easily scale to very high dimensional datasets, they are computationally simple and fast, and they are independent of the classification algorithm (Saeys et al. 2007). As a result, feature selection needs to be performed only once, and then different classifiers can be evaluated.

4.3.2 Wrapper Method

Instead of ranking every single feature, wrappers rank feature subsets by the prediction performance of a classifier on the given subset. The wrapper method is used as an inductive algorithm to estimate the value of a given feature subset (Hernandez et al. 2013). The wrapper methodology offers a simple and powerful way to address the problem of variable selection, regardless of the chosen learning machine. In fact, the learning machine is considered a perfect black box and the method lends itself to the use of off-the-shelf machine learning software packages (Guyon et al. 2003). In its most general formulation, the wrapper methodology consists in using the prediction performance of a given learning machine to assess the relative usefulness of subsets of variables. Wrappers are often criticized because they seem to be a “brute force” method requiring massive amounts of computation, but in some cases it is not necessarily so. Efficient search strategies may be devised to overcome the problem (Guyon et al. 2003).

Feature selection by wrapper methods often achieve better results than filter due to the fact that they are tuned to the specific interaction between an induction algorithm and its training data.

(Maimon and Rokach 2010) However, they tend to be much slower than filters because they must repeatedly call the induction algorithm and must be rerun when a different induction algorithm is used.

4.3.3 Embedded methods

Embedded methods select features based on criteria that are generated during the learning process of a specific classifier. In contrast to wrappers, they do not separate the learning from the feature selection part, i.e. the selected features are sensitive to the structures of the underlying classifiers (Guyon 2006). Embedded methods have the advantage that they include the interaction with the classification model (Saeys et al. 2007), while at the same time being far less computationally intensive than wrapper methods.

4.4 An Instance Base Approach to Feature selection RELIEF

Kira and Rendell (1992) describe an algorithm called Relief that uses instance based learning to assign a relevance weight to each feature. Relief is a simple yet efficient procedure to estimate the quality of attributes. The key idea of Relief is to estimate the quality of attributes according to how well their values distinguish between instances that are near to each other. Given a randomly selected instance R_i from class L, Relief searches for k of its nearest neighbours from the same class called nearest hits H, and also k

nearest neighbours from each of the different classes, called nearest misses *M*. It then updates the quality estimation W_i for ith attribute based on their values for R_i, *H*, *M*. If instance R_i and those in *H* have different values on the i^{th} attribute, then the quality estimation W_i is decreased. On the other hand, if instance R_i and those in *M* have different values on the i^{th} attribute, then W_i is increased. `

$$W[A]:= W[A] - diff(A,Ri,H)/m + diff(A,Ri,M)/m \qquad (4\text{-}1)$$

where A is the current attribute; W[A] is the weight of the currently considered attribute; Ri is the ith sample; H is the "hit"; M is the "miss"; diff() is the probability function; and m is number of the neighbours.

The Relief algorithm is limited to classification problems with two classes. The RELIEF-F algorithm (Robnik et al. 2003) is an extension of the Relief algorithm that can deal with multiclass problems. RELIEF-F is a simple yet efficient procedure to estimate the quality of attributes in problems with strong dependencies between attributes. In practice, RELIEF-F is usually applied in data pre-processing as a feature subset selection method. There are many other extensions of the Relief and RELIEF-F proposed by many researchers. Details about the algorithms and their application can be found in work of Robnik et al (2003).

```
Algorithm 4.1: RELIEF-F

Input: for each training instance a vector of attribute values and the class value.
Output: the vector W of estimations of the qualities of attributes
Step 1: Set all weights W[A] := 0.0;
Step 2: for i := 1 to m do begin
          randomly select an instance Ri;  find k nearest hits Hj;
Step 3: for each class C 6= class(Ri) do
          from class C find k nearest misses Mj (C);
Step 4: for A := 1 to a do
          W[A] :=W[A] - $\sum_{j=1}^{K} diff(A, Ri\,, Hi)/(m-k)$ +
          $\sum_{c\neq class(Ri)}[\frac{P(C)}{1-P(class(Ri)} \sum_{j=1}^{K} diff(A, Ri, Mj(C))]/(m-k)$;
Step 5: end;
```

4.5 Information theory based feature selection

The ranking methods based on information theory filter methods evaluate single features, neglecting possible interactions. Information gain is a simple feature ranking method used by many researchers (Lee and Lee 2006). The information gain is the expected reduction in the entropy caused by partitioning the examples according to a given attribute. Entropy is a measure of the uncertainty associated with a discrete random variable. In other words, entropy is a measure of the average information content of the missing recipients when the system does not know the value. For a set with *k* different values in it, the entropy can be calculated as follows:

$$entropy\ (X) = H(X) = -\sum_{i=1}^{k} P(x_i).\log(P(x_i)) \qquad (4\text{-}2)$$

Where $P(x_i)$ is probabilities of occurrence in a set of possible events x (i.e. the transaction in cardiovascular risk prediction), k is number of transactions.

Other entropy calculations can also be used, for example:

Joint entropy:

Suppose there are two discrete variables *X* and *Y*. *H(X,Y)* is the joint entropy, given by:

$$H(X,Y) = -c \sum_{i=1}^{n} \sum_{j=1}^{m} p_{i,j}\,(x,y) log\; p_{i,j}(x,y) \qquad (4\text{-}3)$$

Condition entropy:

The conditional entropy of *Y* is $H_X(Y)$ defined as average of the entropy of *Y* for each value of *x*, weighted according to the probability of that particular *x*.

$$H_x(Y) = -c \sum_{i,j=1}^{n} p_{i,j}(x,y) \log\; p_i\,(y) \qquad (4\text{-}4)$$

$$H_x(X) = -c \sum_{i,j=1}^{n} p_{i,j}(x,y) \log\; p_j\,(x) \qquad (4\text{-}5)$$

where $p_{i,j}(x,y)$ is the probability of the joint occurrence of x and y; and $p_i(y)$ and $p_j(x)$ are conditional probabilities of X, and Y.

Relative Entropy:

The relative entropy is a measure of the statistical distance between two distributions. It is also known as the Kullback Leibler distance; or Kullback Leibler divergence (Kullback and Leibler 1951).

$$K(p,q) = \sum_{x \in A} p(x) \log(\frac{p(x)}{q(x)}) \qquad (4\text{-}6)$$

where p(x), and q(x) are the distributions in the data set A.

Mutual information:

Mutual information is a basic concept in information theory. It is a measure of general interdependence between random variables. The mutual information between discrete random variables *X* and *Y*, *MI(X,Y)*, is a measure of the amount of information in *X* that can be predicted when *Y* is known. For the case where *X* and *Y* are discrete random variables, *MI(X,Y)* can be written as:

$$MI(X,Y) = H(X) - H(X|Y) = \sum_i \sum_j p_{i,j}(x,y) \log[\, p_{i,j}(x,y) | p_i(x) p_j(y)] \quad (4\text{-}7)$$

Where H(X) is the entropy of X, H(X|Y) (or $H_X(Y)$) is the conditional entropy, which represents the uncertainty in X after knowing Y.

4.6 Summary

Feature extraction and feature selection are the two main categories of techniques for dimensionality reduction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features.

Feature selection may not feasible for cases where attributes are strongly correlated amongst themselves and with the features. Feature extraction processes do not provide a meaningful result if the attribute names in the original dataset are of importance. Hence, the results often cannot be used for devising a decision support system and is not often appropriate to reduce the dimension for creating the predicting models, where the labels associated with the attributes are just as important as the final result. However, the aim of this research is to investigate a principled methodology for the use of data mining in developing a decision support system. Feature selection will be suitable for dimension reduction of the thesis data. Advantages of filter techniques of feature selection are that they easily scale to very high-dimensional datasets, they are computationally simple and fast, and they are independent of the classification algorithm. Feature selection by wrapper methods often achieve better results than filter due to the fact that they are tuned to the specific interaction between an induction algorithm and its training data, but method requiring massive amounts of computation.

The thesis has used two simple feature selection methods Relief and Information gain for the case study discussed in the next chapter and applied in chapter 8.

CHAPTER 5 : OVERVIEW OF RESEARCH CASE STUDIES

5.1 Introduction

This chapter introduces the case studies used in this thesis, with a justification for their use. Three major issues in clinical data mining are discussed. The proposed solutions are also presented in this chapter. The results obtained and subsequent discussions are given in the following chapters.

Two types of databases are available in medical domain (Dasu and Johnson 2003). The first is the dataset acquired by medical experts, which are collected for a special research topic where data collection is triggered by the hypothesis of a clinical trial. The other type is a huge dataset retrieved from hospital information systems. These data are stored in a database automatically without any specific research purpose. These data records are often used for further analysis and building clinical decision support. These types of datasets are often very complex where the numbers of records are very huge, with a large number of attributes for each record. This data often contains many missing values and typically the datasets are imbalanced with regard to the class label of interest. The issues with medical data mining are discussed in depth in chapter 2. As mentioned in section

2.7 there are many issues with medical data mining for the purpose of decision support, but for time limitations this thesis will only deal with missing values, the class imbalance problem and feature selection for class imbalance datasets.

5.2 Data model

The research presented in the thesis will mainly make use of two cardiovascular datasets, from Hull and Dundee clinical sites. Some of the experiments will also use the LifeLab (Appendix A) datasets for comparison.

The Hull site data includes 98 attributes and 498 cases of cardiovascular patients and Dundee site data includes 57 attributes, and 341 cases from cardiovascular patients. After combining the data from both sites, 23 matched attributes are found. The data displays redundancy, noise, inconsistency and many missing values for many of the attributes. After combining the data and removing redundant attributes we found that out of 23 attributes 18 attributes have a missing value frequency from 1% to 30% and out of 839 records, 613 records have 4% to 56% missing values in their attributes. We decided to remove 17 records that have missing value more than 50%. Among all the final 823 records 120 patients are dead and 703 patients are alive.

5.2.1 Data Pre-processing

Redundant attributes: For example, the attribute "*ADMISSION_DATE*" shows patient's operation date; or the two attributes "*Surgeon name1*" and "*Surgeon name2*" represents names of operating doctors. Their values might be helpful in a general evaluation, but offer little relevance to the specific purposes of this thesis.

Missing values: After combining the two data sites and removing the redundant attributes, out of 23 attributes 18 attributes have 1% to 30% missing values and out of 823 records 613 records have 4% to 56% missing values in their attributes (see details in Appendix A).

Noisy and inconsistent data: As an example of numerical outlier values, the attribute "*PACK YRS*" has a big gap between the maximum value of 160, and the minimum value of 2. This affects the transformation process as it unduly changes the mean of the attribute values. These are abbreviations in categorical attributes and outlier values in some numerical attributes. For example, the attribute "CAROTID_DISEASE" includes a mixture of abbreviated and fully specified values such as "asymptomatic carotid disease", "Asx", and so on. In fact, both these values have the same meaning (i.e. they are homonyms). Therefore, these inconsistent entries are harmonised as single values (as shown in Appendix A). Finally a combined dataset having 23 attributes with 823 records was prepared. Out of 823 records 605 records have missing values and 218 records do not have any missing values. Among all the records only 120 patients are dead and 703 patients are listed as alive.

5.3 Clinical risk prediction models

Different risk prediction models have been developed and used by previous researchers on the project data (Davis and Nguyen 2008, Nguyen 2009). One clinical model (CM) in the study of (Davis and Nguyen 2008, Nguyen 2009) heuristic model CM1 (Clinical Model 1) uses patient death within 30 days of an operation as the "High Risk" outcome, with other patients are labelled as "Low Risk". A further model (CM2) uses patient death or severe cardiovascular event (for example Stroke or Myocardial Relapse or Cardio Vascular Arrest) within 30 days of an operation as the "High Risk" outcome; other patients are labelled as "Low Risk". Both the CM1 and CM2 models use all attributes from the "cleaned" patient records, other than those aggregated to form the output labels, as inputs. Further models use only a limited set of attributes. CM1 model is used for most of the thesis case study experiments. Further details of the models can be found in Davis and Nguyen (2008).

5.4 Classification outcome of the thesis data using well known classifiers

This experiment was designed to compare classification outcomes and establish a baseline classification for the thesis data. For this, Decision Tree, Ripple-down rules (Ridor), KNN, FURIA and Neural Network classifiers were used. The details of the classifiers are presented in chapter 3. As described above the dataset has 23 attributes with record of 703 low risk patients and 120 high risk patients. For this experiment the missing values were replaced using a standard Mean/Mode missing imputation technique. No class label balancing technique or any other data pre-processing were used.

The purpose of these experiments was to set a baseline classification outcome for the thesis data. The results are presented in the table 5.1 and later compared with the results from other experiments in chapters 6, 7 and 8.

Table 5.1 presents the classification outcome using the introduced classifiers. Most of the classifiers are showing good accuracy (72% to 80%) but with very poor sensitivity (11% to 23%). Consider the sensitivity rate; the classification outcome of the imbalanced data is very poor because the classifiers give the same attention to the majority class (Low Risk) and the minority class (High Risk). As discussed earlier, when the imbalance level is huge, it is hard to build a good classifier using conventional learning algorithms. They aim to optimize the overall accuracy without considering the relative distribution of each class. For all the classifiers used in this experiment the results show that it is hardly possible to achieve an acceptable prediction rate for high-risk patients as they are a minority set in the case of this data. The highest value of sensitivity (23%) is found with the classifier FURIA, which is still very poor.

Table 5.1: Baseline classification

Classifiers	Confusion Matrix			In %				
	Actual Risk ↓	Classified Risk		ACC	SEN	SPEC	PPV	NPV
		High	Low					
Decision Tree (J48)	High	13	107	80.00	11.00	92.00	19.00	86.00
	Low	56	647					
Ripple-down rules (Ridor)	High	16	104	78.00	13.00	89.00	18.00	86.00
	Low	74	629					
SVM	High	18	102	78.00	15.00	89.00	19.00	86.00
	Low	79	624					
KNN	High	25	95	77.00	21.00	87.00	21.00	87.00
	Low	92	611					
FURIA	High	27	93	72.00	23.00	80.00	16.00	86.00
	Low	140	563					
Neural Network	High	20	100	78.13	16.67	88.62	20.00	86.17
	Low	80	623					

The aim is to find a better way of data pre-processing in order to achieve an acceptable classification outcome in terms of high sensitivity, specificity and accuracy.

Decision Tree (J48) and FURIA are chosen as the baseline classifier and are used for all purposes in classification experiments. Decision Tree is powerful and popular tools for classification and prediction. Decision Tree represents rules, which can be understood by humans and used in decision support.

In summary, the given data with little (albeit standard) pre-processing gives very poor results for all well-known classifiers. It is suggested that this is typical of many legacy databases, particularly in medicine. An improved treatment in data preparation may boost the classification outcome and so help to build a good decision support system. The following sections address some of the data mining issues introduced in chapter 2.

5.5 Addressing the problem with missing values

Many real-life data sets are incomplete. The problem with missing attribute values is a very important issue in Data Mining. In medical data mining the problem with the missing values has become a challenging issue. In many clinical trials, the medical report pro-forma allow some attributes to be left blank, because they are inappropriate for some class of illness or the person providing the information feels that it is not appropriate to record the values for some attributes (Almeida et al. 2010). Typically there are two types of missing data (Little and Rubin 2002) ; one is called Missing

Completely at Random (MCAR). Data is MCAR when the response indicator variables *R* are independent of the data variables X and the latent variables Z. The MCAR condition can be succinctly expressed by the relation $P_{MCAR}(R|X, Z,\mu) = P(R|\mu)$. The second category of missing data is called missing at random or MAR. The MAR condition is frequently written as $P_{MAR}(R|X, Z, \mu) = P(R|X^{obs}, \mu)$ for all X, Z and μ (Marlin 2008, Baraldi and Enders 2010).

5.5.1 Missing values imputation using machine learning methods:

Machine Learning Methods can be used for missing values imputation; for example by using rule induction algorithm in which rules are induced from the original data set, with missing attribute values considered to be ”do not care” conditions or lost values. The Decision Tree can be generate by splitting cases with missing attribute values into fractions and adding these fractions to new case subsets (Maimon and Rokach 2010). Other methods of handling missing attribute values while generating Decision Trees were presented in (Bruha 2004). Jerez et al. (2010) presented comparison results of missing data imputation using statistical and machine learning methods in a real breast cancer problem. They used imputation methods based on statistical techniques, e.g., mean, hot-decking and multiple imputations, and machine learning techniques, e.g., multi-layer perceptron (MLP), self-organising maps (SOM) and k-nearest neighbour (KNN) and applied to the cancer data. The results were then compared to those obtained from the list wise deletion (LD) imputation method (Jerez et al. 2010). SVMI uses an SVM (Support Vector Machines) regression-based algorithm to fill in missing values. It sets the decision attributes (output or class) as the condition attributes (input attributes) and the condition attributes to be addressed as the decision attributes, then SVM regression can be used to predict the missing condition attribute values (Honghai et al. 2005). K Nearest neighbour algorithm has been used by many researchers for imputing missing value (Gustavo Batista 2013, Gajawada and Toshniwal 2012, Batista and Monard 2003, Gustavo Batista 2003). Every time a missing value is found in a current instance, KNN computes the K nearest neighbours and a value from them is imputed. For nominal values, the most common value among all neighbours is taken, and for numerical values, the average value is used (Batista and Monard 2003).

Gajawada and Toshniwal (2012) proposed a modified version of imputing missing value with KNN. Here, the dataset is divided into two sets records with missing value and records without missing value. K-Means clustering is applied to the complete instances set to obtain clusters of complete instances. This was then used to impute the missing values in the incomplete dataset.

5.5.2 Proposed machine earning based missing value imputation method

The thesis proposes a new way of imputing missing value using machine learning methods. The original data set is first partitioned in to groups. The records having missing values in their attributes are in one group and the records without any missing values are placed in a separate group. The classifier is trained with the complete data sets, and later the incomplete data is given to the model for predicting the missing attribute values. The process is repeated for the entire set of attributes that have missing values. At the end of training, this training dataset and missing value imputed datasets are combined to make the complete data. The final dataset is then fed to the selected classifier for classification on the true outcome.

Experiments were performed (details are in chapter 6) with five classification algorithms; Decision Tree (Marsala 2009); k-Nearest Neighbour (KNN) (Latifoğlu et al. 2008); Support Vector Machine (SVM) (Devendran et al. 2008); Fuzzy Unordered Rule Induction Algorithm (Hühn and Hüllermeier 2009); and Ripple-down rules (Ridor) (Brian et al. 1995). Missing values imputation based on Mean/Mode is used as a statistical technique for comparison.

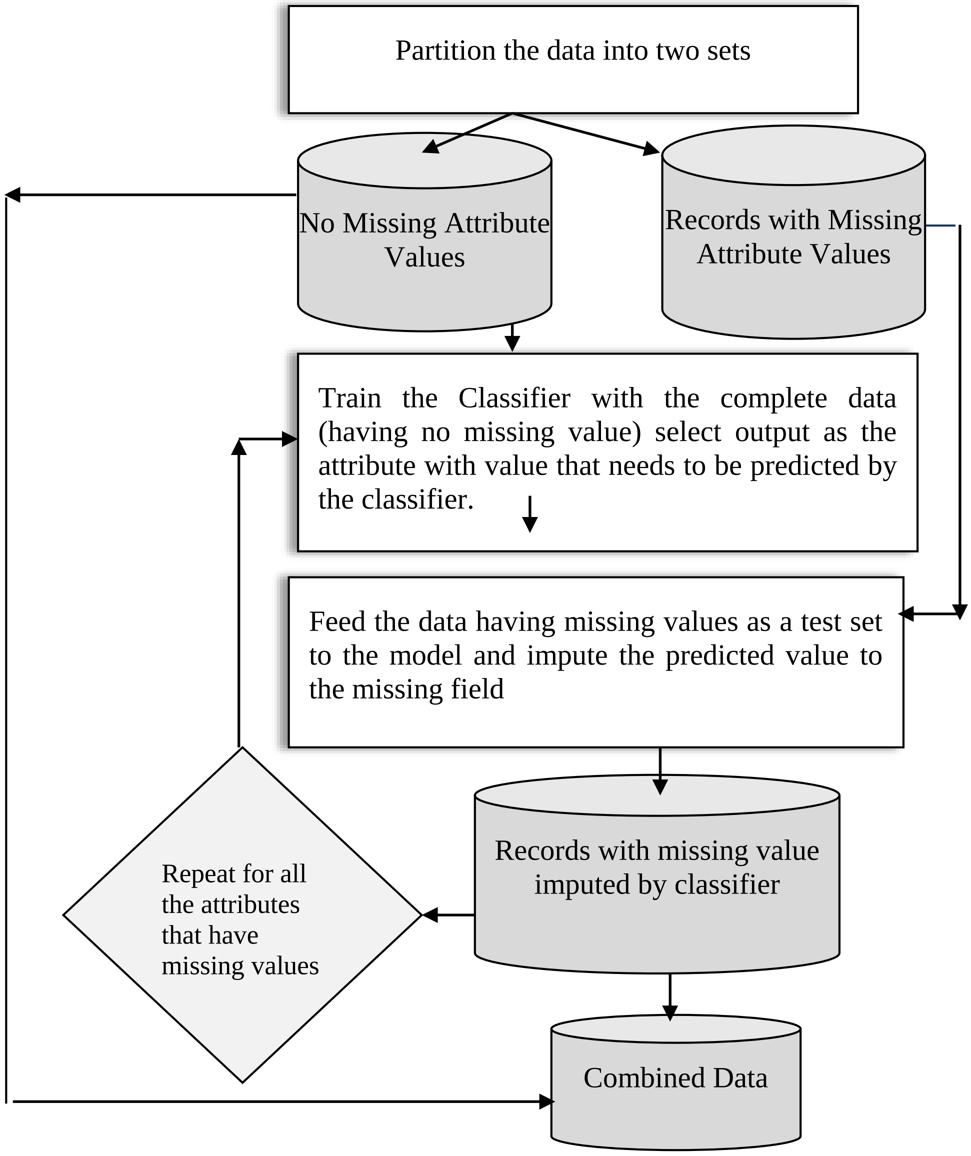

Partition the data into two sets
No Missing Attribute Values
Records with Missing Attribute Values
Train the Classifier with the complete data (having no missing value) select output as the attribute with value that needs to be predicted by the classifier.
Feed the data having missing values as a test set to the model and impute the predicted value to the missing field
Records with missing value imputed by classifier
Repeat for all the attributes that have missing values
Combined Data

Figure 5-1: Block Diagram of the Missing Value Imputation Technique

5.6 Addressing the class imbalance problem

A well balanced training dataset is very important in creating a good training set for the application of classifiers. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced, because they aim to optimize the overall accuracy without considering the relative distribution of each class (Liu et al. 2011). Typically real world data are usually imbalanced and it is one of the main causes for the decrease of generalization in machine learning algorithms (Kim 2007). Conventional learning algorithms do not take into account the imbalance of class. They give the same attention to the majority class and the minority class. When the imbalance level is huge, it is hard to build a good classifier (for the minority class) using conventional learning algorithms (Yan-Ping et al. 2010). Conventional classification algorithms like Neural Networks, Decision Tree, Native Bayes and K-Nearest Neighbour assume that all classes have a similar number of records in the training data and the cost derived from all the classes is equal. Actually, the cost in mispredicting minority classes is higher than that of the majority class for many class imbalance datasets. Therefore, if a classifier can make correct predictions on the minority class efficiently, it will be useful to solving many real applications (Yan-Ping et al. 2010). Sampling strategies have been used to overcome the class imbalance problem by either eliminating some data from the majority class (*under-sampling*) or adding some artificially generated or duplicated data to the minority class (*over-sampling*) (Laza et al. 2011).

5.6.1 Over-sampling

Over-sampling techniques increase the number of minority class members in the training set. The advantage of over-sampling is that no information from the original training set is lost since all members from the minority and majority classes are kept. However, the disadvantage is that we greatly increase the size of the training set. If we do not consider the time taken to resample, under-sampling beats over-sampling in terms of time and memory complexity (Liu 2004). Random over-sampling is the simplest approach to over-sampling, where members from the minority class are chosen at random; these randomly chosen members are then duplicated and added to the new training set (Zhai et al. 2011). Chawla et al. (2002) proposed an over-sampling approach called SMOTE in which the minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with duplication. Depending upon the amount of over-sampling required, neighbours from the k nearest neighbours of a record are randomly chosen. The implementation used in this thesis currently uses five nearest neighbours. For instance, if the amount of over-sampling needed is 200%, only two neighbours from the five nearest neighbours are chosen and one sample is generated in the direction of each. Synthetic samples are generated in the following way (Chawla et al. 2002) and algorithm is presented in algorithm 5.1.

- Take the difference between the feature vector (sample) under consideration and its nearest neighbour.
- Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration.

Algorithm 5.1: SMOTE over-sampling

```
SMOTE(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE N%; Number of nearest neighbors k; Output: (N/100)* T synthetic minority class samples
(∗ If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd. ∗)
if N <100
  then Randomize the T minority class samples
     T = (N/100) ∗ T
     N = 100
endif
N = (int)(N/100)( ∗ The amount of SMOTE is assumed to be in integral multiples of 100. ∗)
k = Number of nearest neighbors
numattrs = Number of attributes
Sample[ ][ ]: array for original minority class samples
newindex: keeps a count of number of synthetic samples generated, initialized to 0
Synthetic[ ][ ]: array for synthetic samples
for i ← 1 to T
     Compute k nearest neighbors for i, and save the indices in the nnarray
     Populate(N, i, nnarray)
endfor
Populate(N, i, nnarray) (∗ Function to generate the syn-thetic samples. ∗)
while N _= 0
     Choose a random number between 1 and k, call it nn.
     for attr ← 1 to numattrs
           Compute: dif = Sample[nnarray[nn]][attr] –  Sample[i][attr]
           Compute: gap = random number between 0 and 1
              Synthetic[newindex][attr] = Sample[i][attr] + gap ∗ dif
        endfor
     newindex++
     N = N − 1
endwhile  return (∗ End of Populate. ∗)
```

SMOTE blindly generates synthetic minority class samples without considering majority class samples and may cause overgeneralization (Yen and Lee 2009). Over-sampling may cause longer training time and over-fitting. Drummond and Holte (2003) showed that random under-sampling yields better minority prediction than random over-sampling.

5.6.2 Under-sampling

The alternative to over-sampling is under-sampling where the size of majority class sample is reduced from the datasets. Since there is much more samples of one class than the other class, to solve the imbalanced class distribution problem. Under-sampling is a technique to reduce the number of samples in the majority class. One simple method of under-sampling (random under-sampling) is to select a subset of majority class samples randomly and then combine them with minority class sample as a training set (Yen and Lee 2009).

Many researchers have proposed advanced ways of under-sampling the majority class data. According to (Chyi 2003) the under-sampling approach based on distance uses distinct modes: the nearest, the farthest, the average nearest, and the average farthest distances between minority and majority classes, as four standards to select the representative samples from the majority class. For every minority class sample in the dataset, the first method (‵‵nearest”) calculates the distances between all majority class samples and the minority class samples, and selects *k* majority class samples which have the smallest distances to the minority class sample. If there are *n* minority class samples in the dataset, the ‵‵nearest″ method would finally select *(k x n)* majority class samples *($k\geq1$)*. However, some samples within the selected majority class samples might be duplicated. The ‵‵farthest″ method selects the majority class samples which have the farthest distances to each minority class sample. For every majority class sample in the dataset, the third method (‵‵average nearest″) calculates the average distances between one majority class sample and all minority class samples. This method selects the majority class samples which have the smallest average distances. The last method ‵‵average farthest″ is similar to the ‵‵average nearest″ method; it selects the majority class samples which have the farthest average distances with all the minority class samples. The above under-sampling approaches based on distance in Chyi (2003) spend a lot of time selecting the majority class samples in the large dataset, and they are not efficient in real applications (Yen and Lee 2009).

Down-sizing the majority class results in a loss of information that may result in overly general rules (Zhang and Mani 2003). In order to overcome this drawback of the under-sampling approach (Yen and Lee 2009) proposed an unsupervised learning technique for supervised learning called cluster-based under-sampling. Their approach is to first cluster all the training samples into *K* clusters (they have run the experiment with different *K* values to observer the outcome) then chose appropriate training samples from the derived clusters. The main idea is that there are different clusters in a dataset, and each cluster seems to have distinct characteristics. If a cluster has more majority class samples and less minority class samples, it will behave like a majority class sample. On the other hand, if a cluster has more minority class samples and less majority class samples, it does not hold the characteristics of the majority class samples and behaves more like the minority class samples. Therefore, their approach selects a suitable number of majority class samples from each cluster by considering the ratio of the number of majority class samples to the number of minority class samples in the derived cluster (Yen and Lee 2009). They first cluster the full data to *K* clusters. A suitable number (*M*) of majority class samples from each cluster are then selected by considering the ratio of the number of majority class samples to the number of minority class samples in the cluster. The number *M* is determined by equation 5.1, and they randomly choose the *M* numbers of majority class samples from each cluster. In the *i* th cluster *(1≤ i ≥ K)* the $Size_{MA}^{i}$ will be:

$$Size_{MA}^{i} = (m \text{ x } Size_{MI}) \text{ x } \frac{Size_{MA}^{i} / Size_{MI}^{i}}{\sum_{i-1}^{K} Size_{MA}^{i} / Size_{MI}^{i}} \qquad (5\text{-}1)$$

(Yen and Lee 2009) also proposed five other approaches for under-sampling using clustering. First they cluster all samples into *K* ($K\geq 1$) clusters as well, and determine the number of selected majority class samples for each cluster by expression (5.1). For each cluster, the representative majority class samples are selected in different ways. The first method SBCNM-1 (sampling based on clustering with NearMisss-1) selects the majority class samples whose average distances to *M* nearest minority class samples (MP1) in the i^{th} cluster are the smallest. In the second method SBCNM-2 (sampling based on clustering with NearMisss-2), the majority class samples, whose average distances to *M* farthest minority class samples in the *i*-th cluster are the smallest, will be selected. The third method SBCNM-3 (sampling based on clustering with NearMisss-3) selects the majority class samples whose average distances to the closest minority class samples in the *i*-th cluster are the smallest. In the fourth method SBCMD (sampling based on clustering with Most Distance), the majority class samples, whose average distances to *M* closest minority class samples in the *i*-th cluster are the farthest, will be selected. The last proposed method, which is called SBCMF (sampling based on clustering with most far), selects the majority class samples whose average distances to all minority class samples in the cluster are the farthest.

5.6.3 Proposed Cluster Based Under-Sampling Technique

The thesis approach to under-sampling is different to the approach of Yen and Lee (2009). Where the data first separated into two sets; one subset has all the majority class samples and the other subset has the entire minority class sample. Then the clustering is applied to the majority class samples to *K* clusters (*K* > 1), then made *K* subsets of majority class samples, where each cluster is considered to be one subset of the majority class. The aim was not to produce a majority and minority class ratio of 1:1; but just reduce the gap between the numbers of majority class samples to the numbers of minority class samples.

All the subsets of majority class are separately combined with the minority class samples to make *K* different training data sets. All the combined datasets are classified with Decision Tree (J48) and Fuzzy Unordered Rule Induction Algorithm. The datasets giving the highest accuracy with majority of the classifiers were kept for further data mining processes. When there are so few members of the minority class, researchers are very hesitant to eliminate members of the minority class. Instead, the assumption is that each minority class member (target class) is very important. This may or may not be the case in practice since some minority class members may represent noise or extreme outliers. With so few data points, it is also difficult to differentiate between noise and minority class members (Liu

2004). Some experiments are done in order to evaluate the proposed method. The experimental results are presented and discussed in chapter 7.

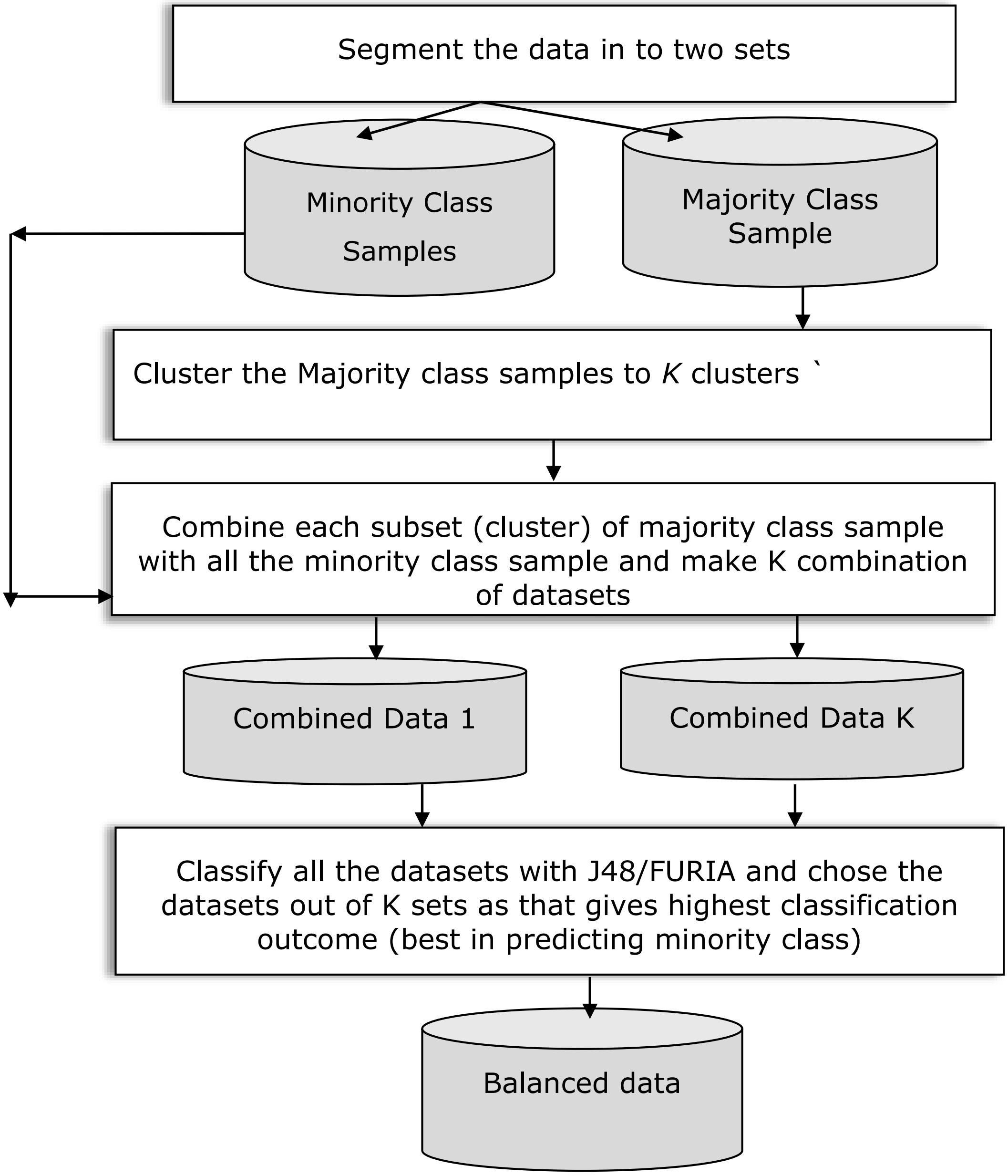

Segment the data in to two sets
Minority Class Samples
Majority Class Sample
Cluster the Majority class samples to *K* clusters `
Combine each subset (cluster) of majority class sample with all the minority class sample and make K combination of datasets
Combined Data 1
Combined Data K
Classify all the datasets with J48/FURIA and chose the datasets out of K sets as that gives highest classification outcome (best in predicting minority class)
Balanced data

Figure 5-2: Proposed Under-Sampling Process

5.7 Feature selection and class imbalance problem

Feature selection is the process of selecting a subset of relevant features for use in model construction. Feature selection is also useful as part of the data analysis process. By the help of some feature selection methods it can be seen that which features are important for prediction, and how these features are related. May researchers (Bunkhumpornpat et al. 2009, Chawla et al. 2003, Liu et al. 2011, Tong et al. 2009, Yan-Ping et al. 2010) found that most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced.

As discussed in the previous chapter, both the filter and wrapper methods of feature selection use the class label of the dataset to select the attribute subset. However, the class imbalance problem can also affect the feature selection process. Our research found that very few work is done in this area, to find and address the issues of class imbalance in feature selection. In the work of Al-Shahib et al. (2005), the author used random under-sampling to balance their data for SVM classification. The author also used feature selection on balanced data. They found that SVM performed well on the balanced data. Tian-yu (2009) tried some standard feature selection method on some class imbalance data and compare which method of feature selection good for their experimental data. Khoshgoftaar et al. (2010)

proposed repetitive feature selection method for imbalance data, where they used random under-sampling to balance the data.

But none of the paper provides any analysis of the effect of the class imbalance problem in feature subset selection.

5.7.1 Proposed feature selection framework for imbalanced dataset

A framework of feature selection for imbalanced clinical detests is proposed in this research. The framework is based on K-Means clustering and instance based feature selection algorithm.

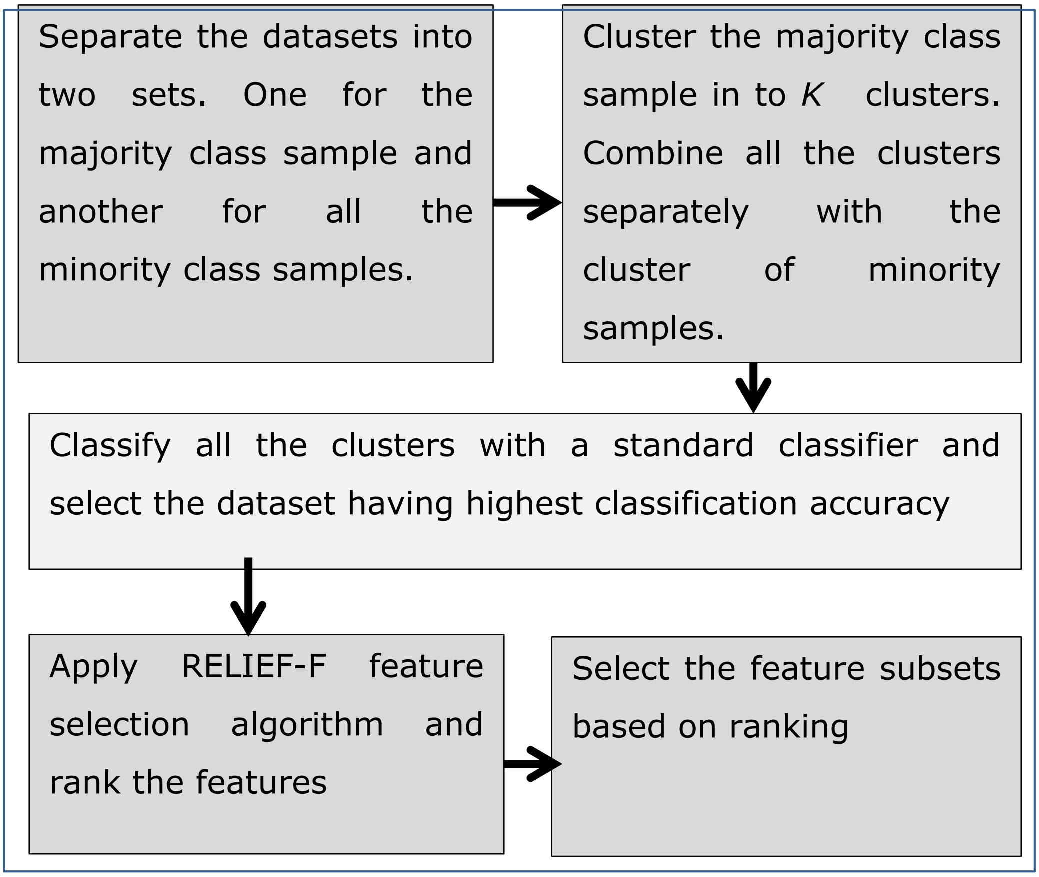

Separate the datasets into two sets. One for the majority class sample and another for all the minority class samples.
Cluster the majority class sample in to *K* clusters. Combine all the clusters separately with the cluster of minority samples.
Classify all the clusters with a standard classifier and select the dataset having highest classification accuracy
Apply RELIEF-F feature selection algorithm and rank the features
Select the feature subsets based on ranking

Figure 5-3: Proposed Framework of Feature Selection

First the cluster based under-sampling is used to balance the datasets; later the RELIEF-F algorithm is used for feature ranking. The steps of the feature selection process are given in Figure 5-3.

Some experiments are done to evaluate the framework. The experimental results are presented in chapter 8 section 8.4.

5.8 Summary

In summary, the given data with little (albeit standard) pre-processing gives very poor results for all well-known classifiers. It is suggested that this is typical of many legacy databases, particularly in medicine. A well treatment in data preparation can give better classification outcome.

In medical data mining the problem with the missing values has become a challenging issue. The proposed machine learning based missing value imputation method can be a new and better way of imputing missing values.

Typically real world medical data are usually imbalanced. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced. The proposed cluster based under-sampling technique not only solves the problem of the imbalance class but also can address the issue of validity of the class label discussed in section 2.7. Most of the feature selection methods use the class label of the dataset to select the attribute subset. However, the class imbalance problem can also affect the feature selection process. The chapter 8 will present some experiments to address the issue of feature selection for imbalanced datasets.

The following chapters address the issues raised in a series of case studies that tackle specific issues individually. The aim is to address these and so point to the development of a principled approach to medical data management for the purpose of data mining and decision support.

CHAPTER 6 : EXPERIMENTS ON MISSING VALUE IMPUTATION

6.1 Introduction

This chapter analyses the results of the case studies on missing value imputation discussed in the chapter 5. Experiments are designed on missing value imputation and evaluated using the classifiers introduced in chapter 3 and chapter 4. The classification results are measured and evaluated by using the standard measurements indicated in Chapter 3 such as Confusion Matrix, Accuracy (ACC), Sensitivity (SEN), Specificity (SPEC), Positive Predictive Value (PPV) and Negative Predicative (NPV).

6.2 Experiments on missing value imputation

Missing Values Imputation using proposed machine learning technique (see section 5.5.2) was used on the thesis data. The classifiers like decision tree (J48), KNN, Fuzzy Unordered Rule Induction Algorithm (FURIA), SVM and Ripple-down rules (Ridor) were used for predicting missing values. Dataset prepared by using all the classifiers are later classified using Decision Tree (J48), KNN, Fuzzy Unordered Rule Induction Algorithm (FURIA) and K-Mean clustering algorithm. As discussed in section 5.2, the thesis dataset have 23 attributes and 18 attributes have a missing value frequency from 1% to 30%. Furthermore, out of 832 records, 613 records have 4% to 56% missing values in their attributes. All the above classifiers are used separately to predict missing value and complete the incomplete data. Standard Mean/Mode missing imputation is also used for comparison. The experimental results are presented in Table 6.1 to Table 6.6.

Table 6.1 presents the Decision Tree (J48) classification outcome of the datasets prepared by different missing value imputation methods. First column of the table is the classifier used for training the model with the complete datasets and later used for predicting the missing field of the incomplete dataset. Accuracy (ACC), sensitivity (SEN), specificity (SPEC), positive predictive value (PPV) and negative predicative (NPV) values are calculated from the confusion matrix and presented in the last five columns of the table in percentages. The last row of the table is the classification outcome of the dataset prepared by the standard Mean/Mode missing value imputation method.

From the table 6.1 it can be observed that the Decision Tree (J48) classified accuracy of all the datasets of different missing values imputation methods are almost closed to each other (78% to 80%) and there is a big gap of sensitivity among all the imputation methods. The highest sensitivity (23%) was found with the use of Decision Tree (J48) as imputation method, and the lowest was by mean-mode (11%).

Table 6.1: Different Missing Imputation Methods with J48 Classification

Missing Imputation Methods	Confusion Matrix			In %				
	Actual Risk ↓	Classified Risk		ACC	SEN	SPEC	PPV	NPV
		High	Low					
Decision Tree (J48)	High	27	93	80	23	90	27	87
	Low	72	631					
KNN	High	20	100	80	17	90	23	86
	Low	68	635					
FURIA	High	24	96	80	20	90	25	87
	Low	72	631					
SVM	High	18	102	78	15	89	19	86
	Low	79	624					
Ripple-down rules (Ridor)	High	16	104	78	13	89	18	86
	Low	74	629					
Mean and Mode	High	13	107	80	11	92	19	86
	Low	56	647					

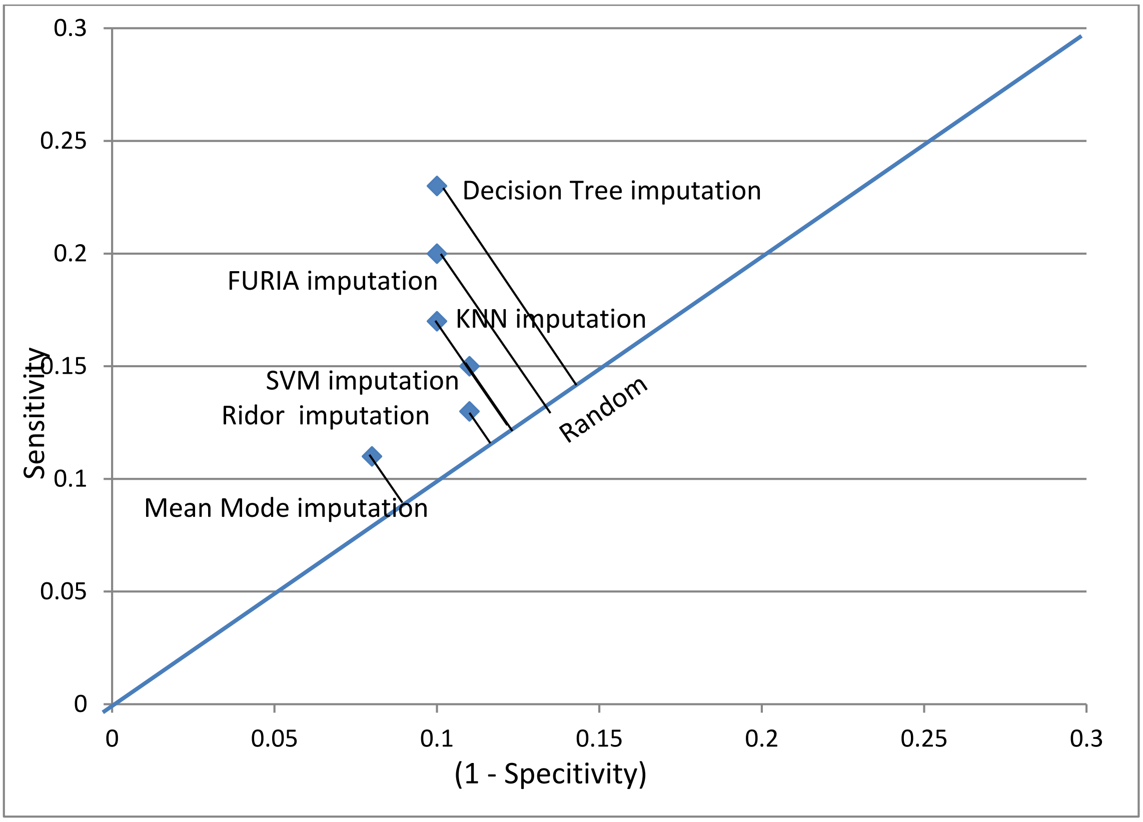


Figure 6-1: The ROC of Sensitivity Versus (1-Specificity) of Decision Tree (J48) Classification

Figure 6.1 shows the ROC distribution of J48 classification different dataset. The statistical method of missing values imputation (Mean/Mode) has high specificity (92%) and very low sensitivity (11%). Decision Tree (J48) imputation has specificity of 90% but has a higher sensitivity (23%) then the statistical method and also any of the other machine learning methods.

Table 6.2 presents the KNN classification outcome of all the datasets prepared by different missing value imputation methods. First column of the table is the classifier used for training the model with the complete datasets and later used for predicting the missing field of the incomplete dataset. Accuracy, sensitivity, specificity, positive predictive value and negative predicative values are calculated from the confusion matrix and in percentages. The last row of the table is the classification outcome of the dataset prepared by the standard Mean/Mode missing value imputation method.

Table 6.2: Different Missing Imputation Methods with KNN Classification

Missing Imputation Methods	Confusion Matrix			In %				
	Actual Risk	Classified Risk		ACC	SEN	SPEC	PPV	NPV
		High	Low					
Decision Tree (J48)	High	24	96	71	20	80	15	85
	Low	140	563					
KNN	High	29	91	81	24	91	32	88
	Low	63	640					
FURIA	High	25	95	79	21	89	24	87
	Low	79	624					
SVM	High	24	96	71	20	80	15	85
	Low	140	563					
Ripple-down rules (Ridor)	High	25	95	80	21	90	26	87
	Low	73	630					
Mean and Mode	High	25	95	77	21	87	21	87
	Low	92	611					

Table 6.2 shows the KNN classified accuracy of all the datasets of the different missing values imputation methods are from 71% to 81% and the highest sensitivity (24%) was found with the use of KNN as imputation method, and the lowest was by Decision Tree (J48) (20%).

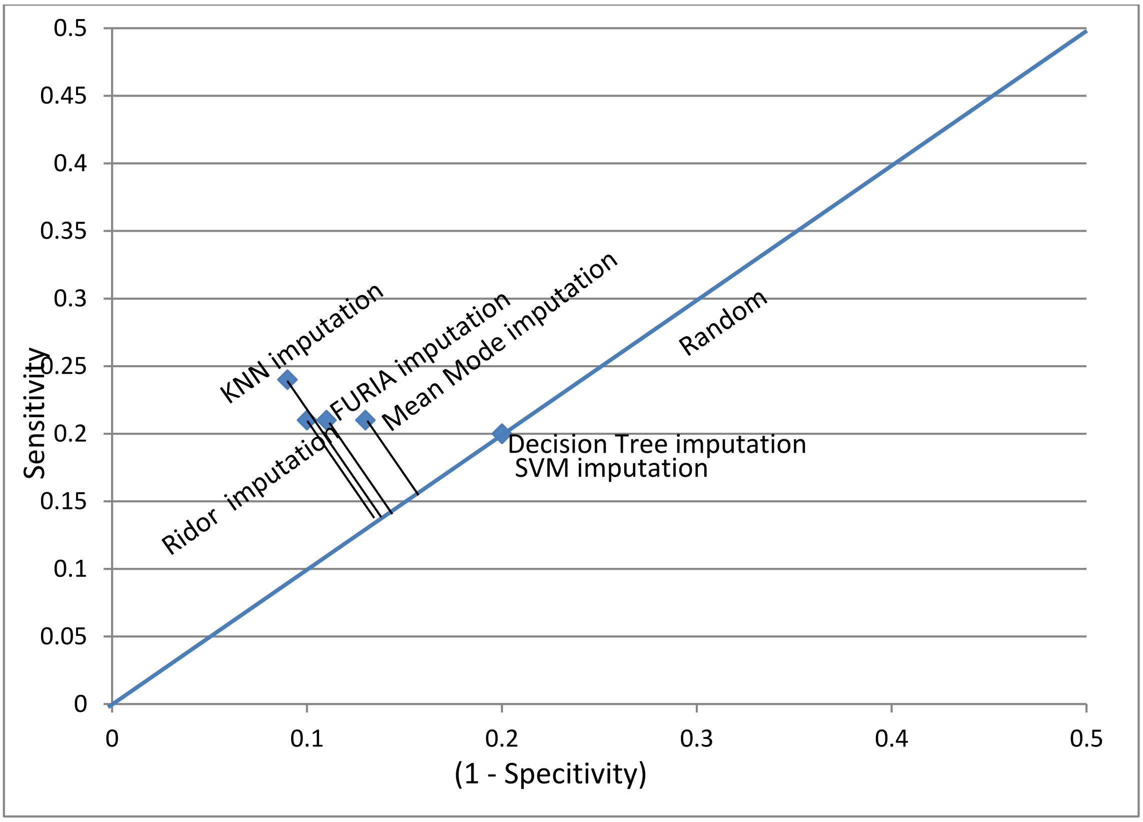


Figure 6-2: The ROC of Sensitivity versus (1-Specificity) of KNN Classification

Figure 6.2 shows the graph of sensitivity versus specificity. The use of KNN as missing imputation outperformed all the other methods. KNN has the highest sensitivity (24%), specificity (91%) and accuracy

(81%) among all the methods. The statistical method of missing values imputation (mean-mode) has slightly better sensitivity and accuracy then Decision Tree (J48) and SVM as missing imputation methods. The next table presents the classification outcome of Fuzzy Rule Induction Algorithm Classification of all the datasets prepared by the proposed missing value imputation method.

Table 6.3 presents the FURIA classification outcome of all the datasets prepared by different missing value imputation methods. First column of the table is the classifier used for training the model with the complete datasets and later used for predicting the missing field of the incomplete dataset. The last row of the table is the classification outcome of the dataset prepared by the standard Mean/Mode missing value imputation method.

Different machine learning algorithm were applied on the dataset to predict the missing values and the missing values imputed datasets are applied to Fuzzy Rule Induction Algorithm (FURIA) for classification. The classification results in table 6.3 shows that the use of Decision Tree (J48) has high sensitivity (40%).

Table *6.3*: Different Missing Imputation Methods with FURIA Classification

Missing Imputation Methods	Confusion Matrix			In %				
	Actual Risk	Classified Risk						
		High	Low	ACC	SEN	SPEC	PPV	NPV
Decision tree (J48)	High	48	72	63	40	67	17	87
	Low	230	473					
KNN	High	36	84	67	30	73	16	86
	Low	190	513					
Fuzzy Unordered Rule Induction Algorithm	High	36	84	67	30	73	16	86
	Low	190	513					
SVM	High	22	98	74	18	83	16	86
	Low	117	586					
Ripple-down rules (Ridor)	High	24	96	74	20	83	17	86
	Low	117	586					
Mean and Mode	High	27	93	72	23	80	16	86
	Low	140	563					

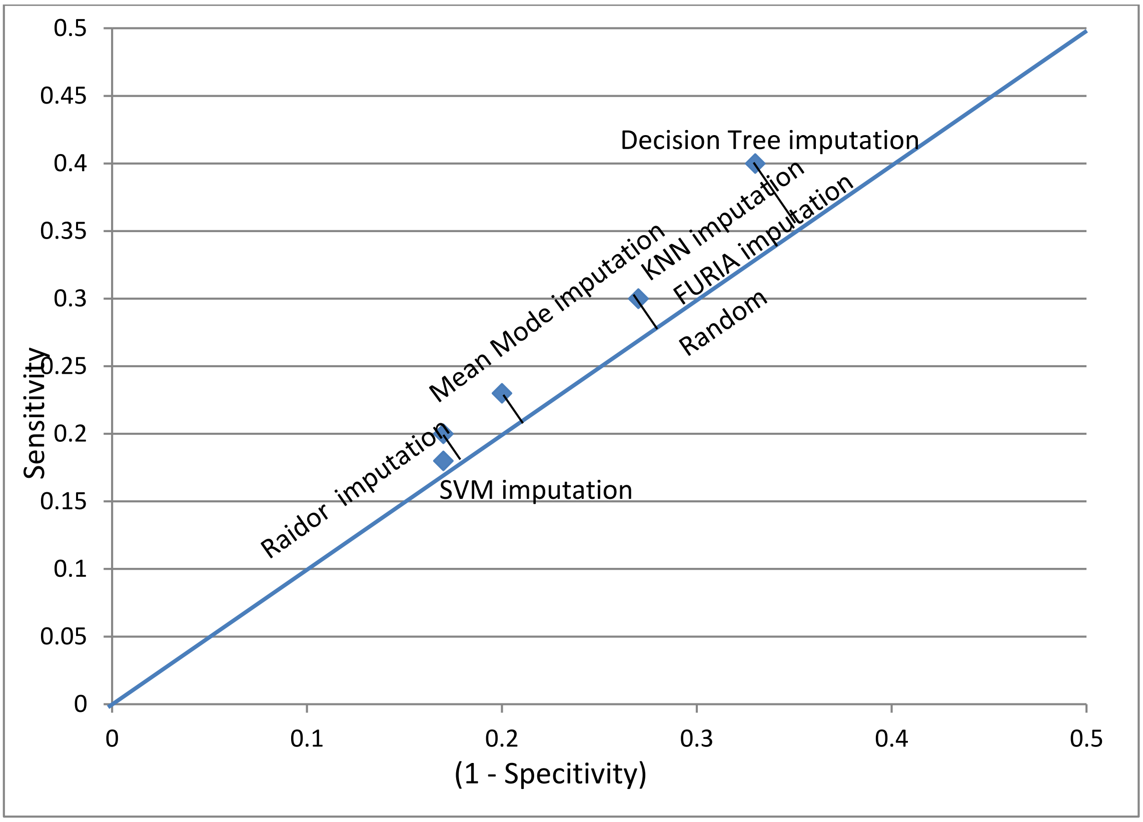


Figure 6-3: The ROC of Fuzzy Rule Induction Algorithm (FURIA) Classification

Figure 6.3 shows the graph of sensitivity versus specificity. The use of Decision Tree (J48) as missing imputation outperformed all the other methods. Decision Tree (J48) has the highest sensitivity (40%). Although SVM has the high specificity (83%), it shows very poor sensitivity (18%) compared to all the other imputation methods. Fuzzy Unordered Rule Induction Algorithm and KNN have the same sensitivity of 30%. For Fuzzy Rule Induction Algorithm (FURIA) the Decision Tree (J48) imputation method perform best for predicting the high risk patients. The next table presents the classification outcome of K-Means clustering of all the datasets prepared by the proposed missing value imputation method.

Table 6.4 presents the K-Means classification outcome of all the datasets prepared by different missing value imputation methods. First column of the table is the classifier used for training the model with the complete datasets and later used for predicting the missing field of the incomplete dataset.

Table 6.4: Different Missing Imputation Methods with K-Means Clustering

Missing Imputation Methods	Confusion Matrix			In %				
	Actual Risk	Classified Risk		ACC	SEN	SPEC	PPV	NPV
		High	Low					
Decision Tree (J48)	High	36	84	64	30	70	15	85
	Low	212	491					
KNN	High	51	69	53	43	54	14	85
	Low	321	382					
FURIA	High	52	68	58	43	60	16	86
	Low	281	422					
SVM	High	36	84	62	30	67	14	85
	Low	229	474					
Ripple-down rules (Ridor)	High	38	82	62	32	67	14	85
	Low	230	473					
Mean and Mode	High	35	85	63	29	69	14	85
	Low	219	484					

From table 6.4 we can see that Decision Tree (J48) imputation method shows the highest accuracy (64%), but there is a big gap between the highest sensitivity (43%) shown by both KNN and Fuzzy Unordered Rule Induction Algorithm and the Mean/Mode imputation method (29%). Although KNN and Fuzzy Unordered Rule Induction Algorithm display the same sensitivity (43%), the accuracy and positive predicted rate of Unordered Rule Induction Algorithm is higher than the KNN.

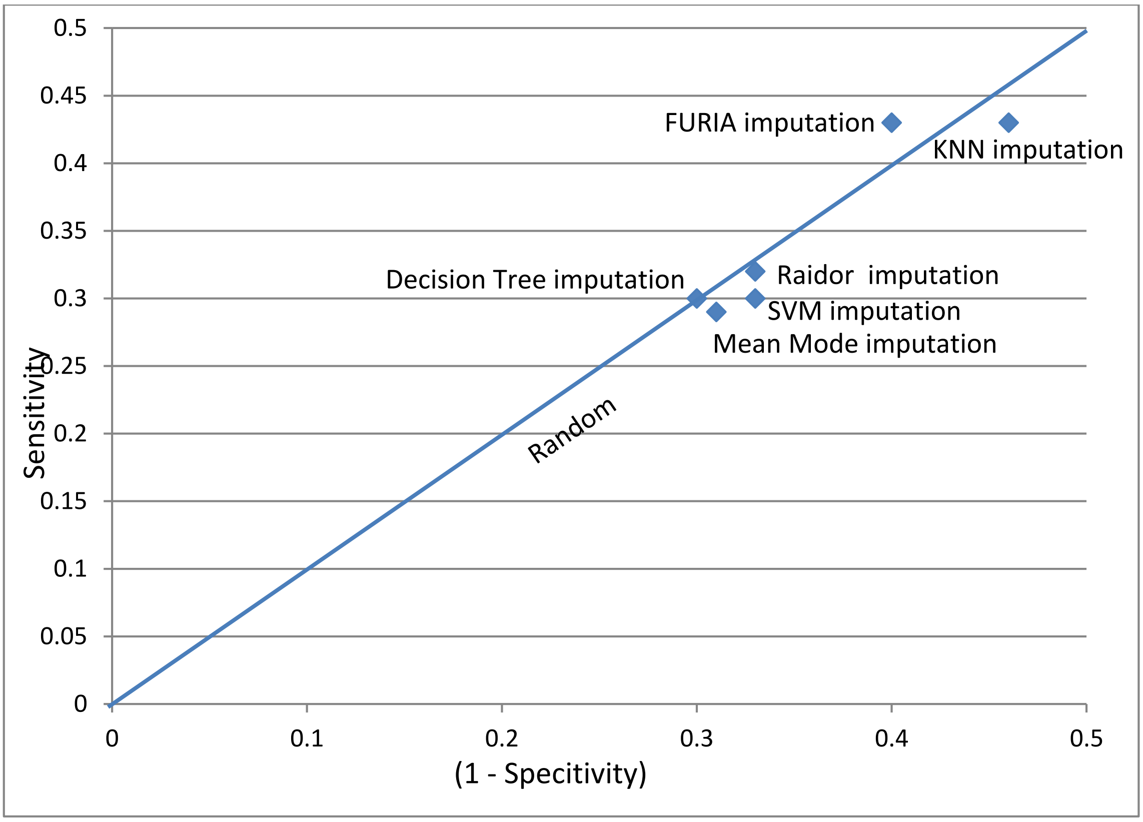


Figure 6-4: The ROC of Sensitivity versus (1-Specificity) of K-Mean Clustering for the different missing value imputation methods.

Figure 6.4 shows the graph of sensitivity versus specificity for K-Mean clustering. The use of Fuzzy Unordered Rule Induction Algorithm as missing imputation outperformed all the other methods. Although the Fuzzy Unordered Rule Induction Algorithm and KNN both have the highest sensitivity (43%), the accuracy and specificity of Fuzzy Unordered Rule Induction Algorithm is higher than KNN. The ROC curve for classification outcome of all combinations of missing value imputation methods and classifiers are presented in figure 6.5 and table 6.5 presents the highest sensitivity value obtained by different classifiers used as the proposed machine learning based missing value imputation method.

Figure 6.5 shows the ROC of different combination of the machine learning algorithms used for imputing missing values and classifying the final complete data. A random classification line was also drawn to see how much better the classification outcomes are over random. From the figure it can be seen that apart from the combination B and F all the combinations where machine learning algorithm were used, the classification performances are better than random classifier. The combination A (FURIA-K-Means), where FURIA was used to predict and impute the missing values and K-Mean was used to classify the final complete data has got the highest sensitivity.

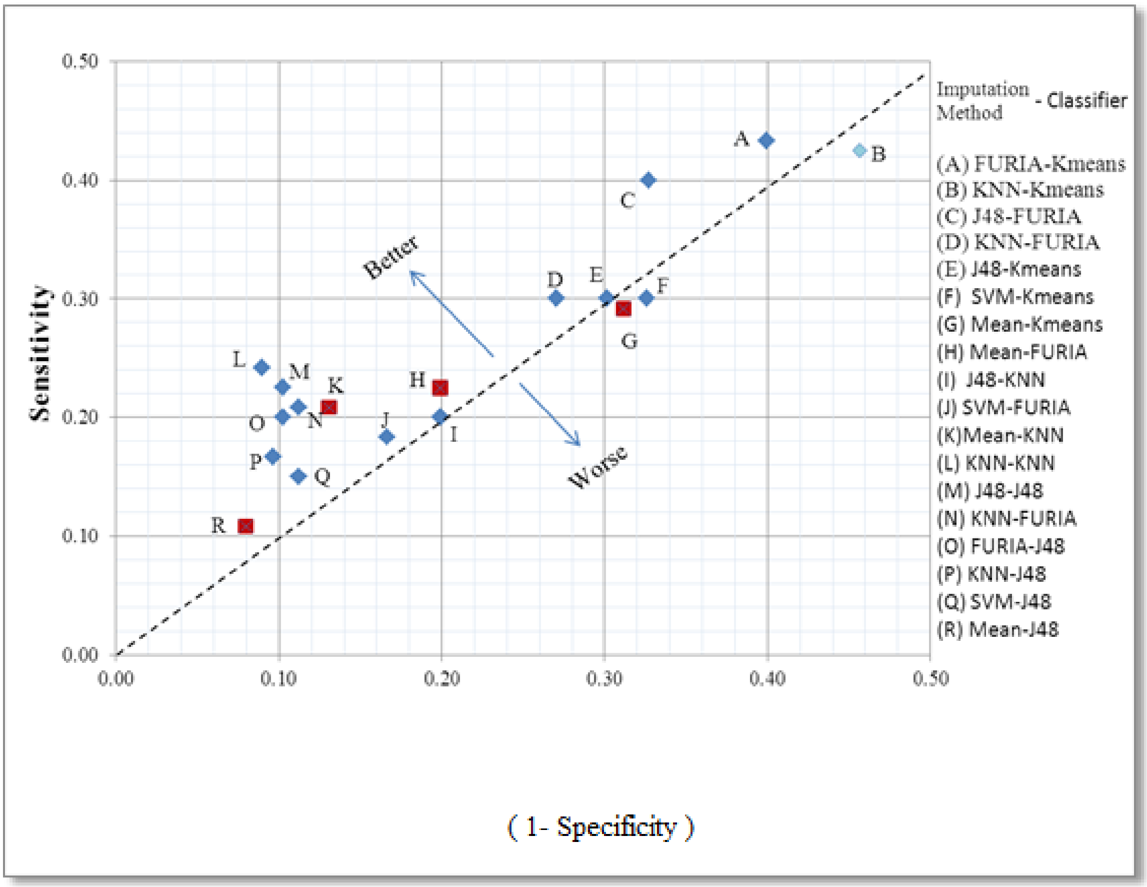


Figure 6-5: Sensitivity versus (1-Specificity) for All Imputation Methods. The data points A to R can be interpreted via the key with lists (Imputation Method-Classifier) pairings.

If we measure the perpendicular distance of the points from the random classification line the combination L and M are found to have the highest (best) distance from the random line. Some of the classification outcomes of classifiers where Mean/Mode was used to impute the missing vale also show better than random results. However most of them are very low compared to all the combinations where machine learning was used for missing value imputation. Out of the classifications where Mean/Mode was used as missing value imputation the combination K (Mean/Mode-KNN) found to be best. Table 6.5 presents the highest sensitivity found from the classifiers used as missing value imputation. First column of the table is the name of the classifier used for missing value imputation and last column is the name of the classifier use to classify the final complete datasets.

Table 6.5: The Highest Sensitivity Values of Different Missing Imputation Methods

Missing Imputation Methods	Highest Sensitivity	With the Accuracy	The Classifier Used
FURIA	43.3%	58%	K-Mean
KNN	42.5%	51%	K-Mean
Decision Tree (J48)	40%	63%	FURIA
Ripple-down rules (Ridor)	32%	62%	K-Mean
SVM	30%	62%	K-Mean
Mean and Mode	29%	63%	K-Mean

From the table 6.5 we can conclude that if the research aim is to achieve high sensitivity for unsupervised learning it is recommended to use FURIA as missing value imputation method and for supervised learning decision tree as missing value imputation method.

Table 6.6 presents a comparison of the classification outcome of the previous PhD research (Nguyen Thuy, T. T. 2009, Chapter 7, table 6.6).

Table 6.6: Results of K-Mix Clustering and K-mean Clustering with FURIA

Classifier with Different Missing Imputation Methods	Confusion Matrix			In %				
	Actual Risk	Classified Risk		ACC	SEN	SPEC	PPV	NPV
		High	Low					
Published Results of previous PhD Thesis	High	48	91	60	35	65	16	83
	Low	248	452					
K-Mean With FURIA missing value imputation method	High	52	68	58	43	60	16	86
	Low	281	422					

The same dataset was used and the results show that the K-Means performed better than the k-Mix if we use the proposed missing imputation method to address the issue of missing values.

6.3 Discussions

In this chapter the performance of machine learning techniques as missing value imputation was examined. The results are compared with traditional Mean/Mode imputation. Experimental results show that all the machine learning methods used in the experiment outperformed the statistical method (Mean/Mode), based on sensitivity and some cases accuracy. Figure 6.5 shows the ROC of classification outcome for the different classifiers used. Different data sets are prepared by the proposed imputation model using J48, KNN, FURIA, SVM, and K-Means. The classification outcome of the data prepared by the Mean/Mode imputation is also plotted for comparison.

The results show that with the data prepared using mean mode as missing value we can get maximum 29% sensitivity with 63% accuracy for the K-Means classification. On the other hand we can get 40%-43% sensitivity if we use machine learning methods to predict the missing value.

It is observed that in most of the cases if the same classifier is used for predicting the missing value and final classifier the performances are better than the other cases. This is likely because the bias of the classifiers in imputing missing values later benefits that classifier on the complete data. However, this is always not the case. We can also see some other combination of the imputation-classifier classification-classifier can produce good results. Some combinations are able to produce better sensitivity while some are producing better specificity. The appropriate selection of the classifier is an issue for this approach to missing value imputation. It is expected that selection will depend on the data and interests of the research. Preparing the data using Machine Learning algorithm *X* and achieving best results on that prepared data using the same Machine Learning algorithm *X* is also to be expected.

Using Mean-Mode we are imputing the unique value for the entire missing field but it is obvious that missing values cannot be unique. It is a big challenge to find the right value for the missing field. The proposed method uses pattern recognition technique to predict the value for the missing field by learning the pattern from the complete dataset. The experiments show that this method is giving an improved way of finding the best possible value for the missing fields.

6.4 Summary

Missing attribute values are common in real life datasets, which causes many problems in pattern recognition and classification. Researchers are working towards a suitable missing value imputation solution which can show adequate improvement in the classification performance. Medical data are usually found to be incomplete as in many cases on medical reports some attributes can be left blank, because they are inappropriate for some class of illness or the person providing the information feels that it is not appropriate to record the values. In this chapter the performance of machine learning techniques as missing value imputation was examined. The results were compared with traditional Mean/Mode imputation. Experimental results show that all the machine learning methods which were explored outperformed the statistical method (Mean/Mode), based on sensitivity and some cases accuracy.

The process of missing imputation with the proposed method can be computationally expansive for large numbers of records having missing values in their attributes. However, we know that data cleaning is part of the data pre-processing task for data mining which is not a real time task and neither a continuous process. Missing value imputation is a one-time task. With this extra effort a good quality data can be obtain for better classification and decision support.

The next chapter will present the experiments on the class imbalance problem of medical datasets.

CHAPTER 7 : EXPERIMENTS ON CLASS BALANCING

7.1 Introduction

Class imbalance is a common problem with most medical datasets. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced. Some commonly used class balancing methods are discussed in chapter 5. An improved cluster based under-sampling was also proposed in section 5.6.3. This chapter presents experiments on different class balancing techniques. Datasets prepared by different class balancing techniques are later classified using Decision Tree (J48) and FURIA. The classification results are measured and evaluated by using the standard measurements indicated in Chapter 3 such as confusion matrix, sensitivity, specificity, positive predictive value, and negative predictive value.

7.2 Experiments on Class Balancing

As discussed in section 5.2 the thesis dataset, with 823 records, has 22 input attributes and one class attribute. Among all the records 703 patients are classed as alive and 120 patients as dead. For this experiment according to clinical risk prediction model (CM1) (Davis and Nguyen 2008), patients with status “Alive” are consider to be

"Low Risk" and patients with status "Dead" are consider to be "High Risk". The data record with the label "Alive" are the majority class having 703 samples out of 823 and the "Dead" label records are the minority class having 120 samples out of 823. The ratio of majority and minority class is 6:1. The ratio gap between majority class and minority class was reduced using different methods discuss in chapter 5, section 5.6. Datasets are prepared using SMOTE over-sampling, random under-sampling, cluster based under-sampling proposed by Yen and Lee (2009) and our proposed under-sampling techniques. The details of the sampling techniques have discussed in chapter 5, section 5.6 and the descriptions of the balanced datasets are presented in table 7.1.

Besides sampling the majority class data by clustering, the minority class is also clustered and under-sampled. The experiment was performed to observe the outcome of different combination of data subsets of majority and minority class samples. The datasets were classified with J48 and FURIA, using. 10-Fold cross validation for training and testing sampling. The accuracy, sensitivity and specificity derived from confusion matrix (discussed in section 3.6), are used as classification performance measure. The results are presented in the table 7.2 to 7.6.

Table 7.1: Description of the data prepared by different balancing methods

Data	Majority : Minority Ratio	Description
D1	2 : 1	Data consist of all the minority class samples (“Dead”) and one cluster of majority class records out of three clusters made by K-Mean. (120 Dead, 213 Alive)
D2	2.4 : 1	Data consist of combination of two clusters of the minority class samples and one cluster of majority class samples. Clusters are made with simple k-mean for both of the classes (K=3). (89 Dead, 213 Alive)
D3	3 : 1	Data consist of combination of all the minority class samples with randomly (random cut 1) selected samples from majority class sample. (120 Dead, 350 Alive)
D4:	3: 1	Data consist of combination of all the minority class samples with randomly (random cut2) selected samples from majority class sample. (120 Dead, 353 Alive)
D5	6 :1	Original data with full samples. (120 Dead, 703 Alive)
D6	1.8 : 1	Majority samples of the data set D2 are clustered in to 3 cluster and each clusters are combined with the minority samples. (89 Dead & 160 Alive)
K3M1Yen	1: 1	Majority and minority ratio 1:1 (M=1) using Yen and Lee (2009)
K3M2Yen	2: 1	Majority and minority ratio 2:1 (M=2) using Yen and Lee (2009)
SMOTE	1:1	The data set was prepared using SMOTE over-sampling with the Majority and minority ratio 1:1.

Data described in the table 7.1 are later classified using Decision Tree (j48) and Fuzzy Unordered Rule Induction Algorithm (FURIA). Table 7.2 presents the FURIA classification outcome of all the datasets prepared by different class balancing techniques described in the table 7.1. The table 7.3 presents the Decision Tree (J48) classification outcome of all the datasets prepared by different class balancing techniques described in the table 7.1. First column of the tables is the name of the dataset and subsequent columns are the classification accuracy (ACC), sensitivity (SEN), specificity (SPEC), positive predictive value (PPV) and negative predicative (NPV) values calculated from the confusion matrix and presented as percentages.

Table 7.2*:* Classification outcome of FURIA (Fuzzy Rules) classification

	FURIA (Fuzzy Rules) classification (%)				
Data Sets	ACC	SEN	SPEC	PPV	NPV
D1	85.89	64.17	98.12	95.06	82.94
D2	92.11	79.78	97.21	92.21	92.07
D3	74.68	11.67	96.29	51.85	76.07
D4	70.82	15.83	89.52	33.93	75.78
D5	66.71	30.00	72.97	15.93	85.93
D6	96.39	91.01	99.38	98.78	95.21
K3M1Yen	61.48	67.50	55.65	59.56	63.89
K3M2Yen	60.39	22.50	79.66	36.00	66.90
SMOTE Over-Sampling	83.00	82.00	83.00	83.00	83.00

Table 7.3: Classification outcome of Decision Tree (J48)

	J48 (Decision Tree Classification) (%)				
Data Sets	ACC	SEN	SPEC	PPV	NPV
D1	84.08	67.50	93.43	85.26	83.61
D2	92.05	83.15	95.77	89.16	93.15
D3	67.66	35.83	78.57	36.44	78.13
D4	66.60	33.33	77.90	33.90	77.46
D5	79.59	20.00	89.76	25.00	86.80
D6	97.59	93.26	100.00	100.00	96.39
K3M1Yen	51.64	52.50	50.81	50.81	52.50
K3M2Yen	59.55	39.17	69.92	39.83	69.33
SMOTE Over-Sampling	85.78	84.21	87.34	86.93	84.69

From the table 7.2 and 7.3 it can be seen that the original imbalance dataset D5 has accuracy of 66.71% with FURIA classification and 79.59 % with Decision Tree (J48) classification. For both of the classifiers the sensitivity value is very poor (30% and 20%). The accuracy is high because the classifier was able to classify the majority class (*alive*) sample well (72.97% and 89.76%) but failed in classifying the target minority set. Dataset D1 where data are balanced by clustering the majority class samples and combining all the minority samples shows better classification outcome than the original imbalance data. With the FURIA and Decision Tree (J48) classification of the D1 dataset, the sensitivity value is 64.2% with the Decision Tree (J48) and 67.5% with the FURIA. The classification outcome of the D2 is 2 to 3 times higher than the original datasets. The dataset prepared by the method proposed by Yen and Lee (2009) shows some increase in the sensitivity value but the accuracy dropped and overall performance was not good. Under-sampling by random cut D3 and D4 also disappointed with its poor accuracy and sensitivity values.

The SMOTE over-sampling technique shows a good classification outcome for both the classifiers Decision Tree (J48) and FURIA. But the performance of under-sampling using the proposed approach is better in terms of classification accuracy and training time of the classifier. Out of 10 runs the average training time of the dataset prepared by SMOTE was 3.84 second and with our approach 0.1 to 0.31 second. ROC spaces of the two classification outcome of the under-sampled datasets are plotted in the figure 7.1 and 7.2.

If we analyse the ROC space for all datasets classified with Decision Tree (J48) plotted in figure 7.1 and FURIA plotted in figure 7.2, we will find that overall accuracy of all the datasets are above the random line and the datasets D1, D2 and D6 which are prepared by our proposed method display the highest accuracy among all the datasets. Accuracy of the datasets prepared by the method of Lee (2009) is just close to random and far worse than all the datasets prepared by the proposed method of under-sampling.

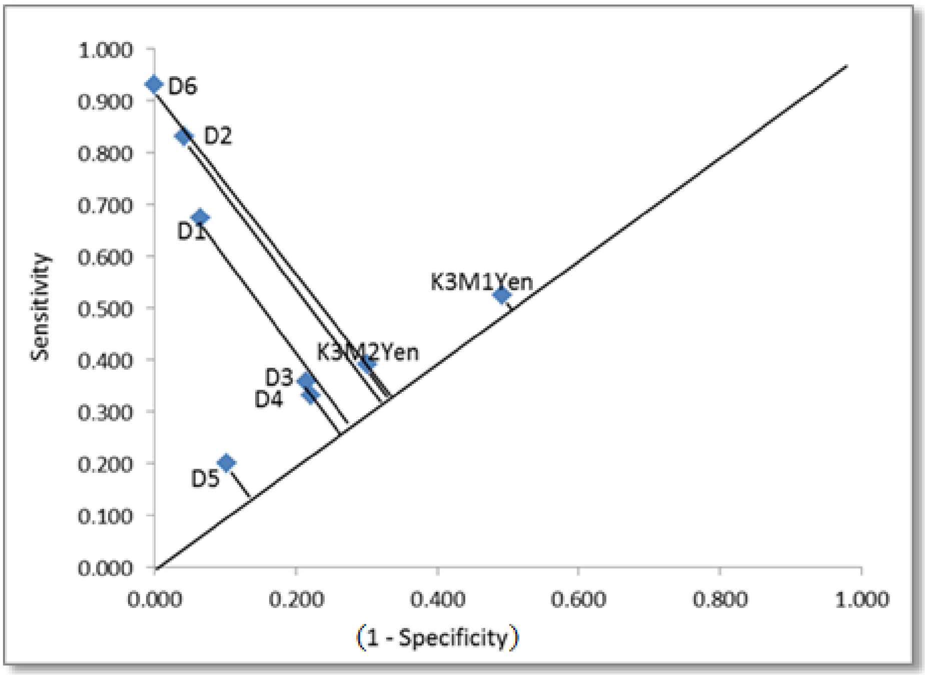


Figure 7-1: ROC of Decision Tree Classification of all balanced data by under-sampling

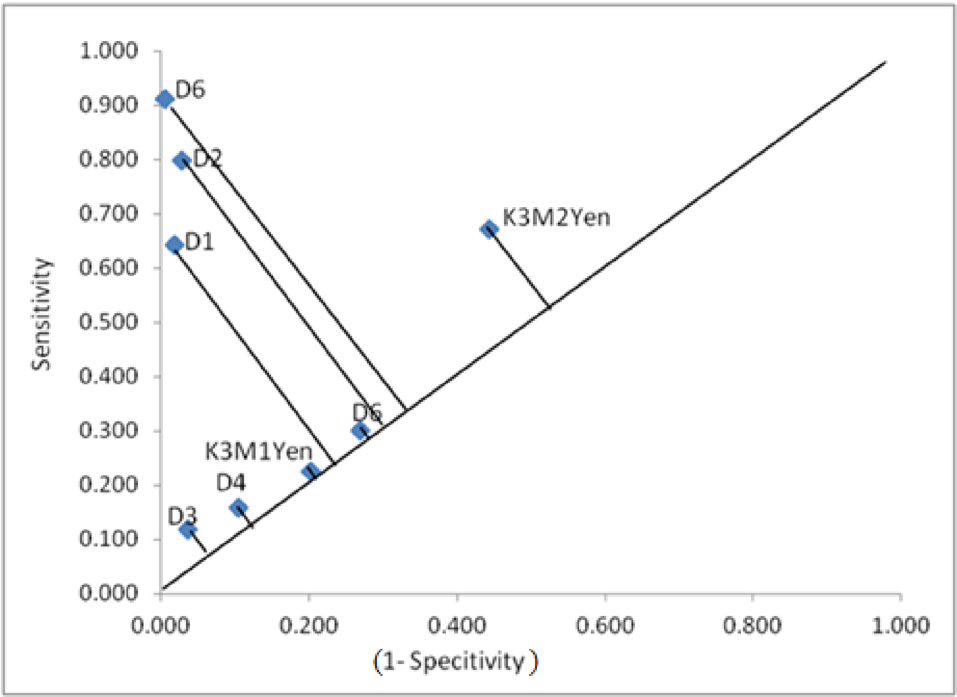


Figure 7-2: ROC of FURIA Classification of all balanced data by under-sampling

7.2.1 Experimental outcome of the proposed class balancing method with LifeLab datasets

In order to see if the performance of the proposed method of class balancing transfers to other datasets, the LifeLab datasets was also tested on the method. LifeLab is a prospective cohort study consisting of patients who were recruited from a community-based outpatient clinical based in England (the University of Hull Medical Centre, UK). This dataset presents the incidents, prevalence and persistence of heart failure, and the dataset routinely collected clinical data to be used for research purposes (appendix A).

Table *7.4*: Data description of datasets made from LifeLab by the proposed method

ID	Description	Number of Dead Records	Number of Alive
LD1	Original Data	520	1512
LD2	Alive records are classified into 3 clusters and each of the clusters is combined with all the dead records. Final dataset is LD2 which has the highest outcome using J48.	520	592
LD3	Dead records from the original data are clustered into 3 clusters and combined with the alive records from LD2. Final dataset is LD3 which has the highest outcome using J48.	282	592

The LifeLab dataset has 85 attributes and 2032 records where 520 records are of dead patients and 1512 are from alive patients. The data description can be seen in appendix A. Three data sets are prepared using the proposed method and described in table 7.4. LD1 is the original imbalanced datasets. The dataset LD2 a LD3 are prepared by proposed class balancing method. Details of the datasets are presented in table 7.4.

Table 7.5: Classification outcome of LifeLab datasets

Classifier and the datasets		Confusion Matrix			%				
		Actual Risk	Classified Risk		ACC	SEN	SPEC	PPV	NPV
			High	Low					
LD1	J48	High	182	338	67.86	35.00	79.17	36.62	77.98
		Low	315	1197					
	FURIA	High	163	357	70.92	31.35	84.52	41.06	78.17
		Low	234	1278					
LD2	J48	High	390	130	77.25	75.00	79.22	76.02	78.30
		Low	123	469					
	FURIA	High	377	143	82.73	72.5	91.72	88.50	79.15
		Low	49	543					
LD3	J48	High	243	39	92.91	86.17	96.11	91.35	93.59
		Low	23	569					
	FURIA	High	252	28	96.11	90.07	98.99	97.69	95.44
		Low	6	586					

All the datasets described in the table 7.4 are classified using the j48 and FURIA. Results are as presented in the table 7.5. First column of the table is the name of the dataset prepared by proposed class balancing technique. The second column is the names of the classifiers used to classify the datasets. The subsequent columns are the classification outcome present as percentages.

From table 7.5 it can be seen that the classification outcome of the original imbalanced data set LD1 is very poor compared to the other datasets prepared by the proposed class balancing technique. As seen in the table 7.4 the LD1 has more “low risk” sample than “high risk” due to class imbalance problem more “high risk” patients record were miss classified as “low risk”. The sensitivity of LD1 data was recoded as 30 % to 35 % with Decision Tree (J48) and FURIA classification. However the datasets LD2 and LD3 prepared by the proposed class balancing technique is showing very high accuracy and sensitivity which is more than double of the imbalanced dataset. The highest accuracy of 96.11 % and sensitivity of 90.07 % is found with the dataset LD3 for FURIA classification.

7.2.2 Experimental outcome of the proposed class balancing method with UCI datasets

The proposed method of class balancing was applied to the Indian-Liver data set from the UCI Machine Learning Repository and was classified with the Decision Tree (J48). The results are presented in the table 7.6.

Table 7.6: IndianLiver data balancing by the proposed method and classified by J48

Data	In %				
	ACC	SEN	SPEC	PPV	NPV
IndianLiver (Original data)	68.78	75.66	44.09	82.93	33.53
IndianLiver (Balanced by the proposed method)	72.1	77.74	59.74	80.86	55.09

IndianLiver data consist of 10 attributes with 586 records. Where there are 167 records are having class label of NonLiver and 419 records are having class label of Liver. The data set is highly imbalanced. It has more records of the target class than the non-target class. The data was balanced using the proposed method and later classified using Decision Tree (j48). The results are presented in table 7.6.

From table 7.6 it can be seen that for the original imbalanced data the Decision Tree (J48) classifier was good at classifying the liver disease records but failed to correctly classifying the non-liver disease patient’s records. When the data was balanced with our proposed method the prediction rate of the minority class was much better than the original imbalanced data.

7.3 Discussion

From the above experiments on class balancing it can be seen that the classification outcome of the balanced data is much better than the imbalanced dataset. Some well-known techniques were used for balancing the data records and the performance of our proposed method of class balancing was compared with them.

When the imbalance level is huge, it is hard to build a good classifier using conventional learning algorithms. They aim to optimize the overall accuracy without considering the relative distribution of each class, and the classification outcome of the imbalance data was always poor.

The over-sampling technique SMOTE was not found to be as good as the under- sampling technique. It is found that the SMOTE blindly generates synthetic minority class samples without considering majority class samples and may cause overgeneralization. Over-sampling also takes a longer training time.

It is also observed from the experiments that the majority and minority ratio is not the only issue in building a good prediction model. There is also a need for good training samples that display data properties consistent with the class label assigned to them. Most of the time the records of clinical datasets do not truly reflect data properties consistent with the target or outcome label. If we consider the cardiovascular risk based on dead or alive of previous patient's records, it may also happen that some of the patients may have died with some other cause than the target and some patients have more propriety of a high risk patient but still they are alive. It can be seen that the majority and minority ratio of D1 and D2 are very close but the classification outcomes are not similar. Although the majority minority ratio is almost same, there is a big difference in the classification accuracy, sensitivity and specificity of the datasets; as can be noticed in the table 7.2 and 7.3. The dataset "K3M1Yen" prepared by the method proposed by Yen and Lee (2009) has 1:1 ratio but still displays poorer classification outcome than the other datasets. Two more datasets (Lifelab and Indian Liver) were also tested on the proposed method. The data balanced by the proposed method for all the datasets was found to perform much better than all other methods tested here. For under-sampling the proposed method is found to be good in selecting the best samples from the minority class.

Further experiments were done with the aim to observe how good the classification models are to deal with the unknown records, when they are built with the data balanced by the proposed method. The datasets prepared by the proposed class balancing method was used as a training set to build a classifier model then the original imbalanced dataset was used as a testing set to test the model. Details of the experimental results are present in Appendix B. The thesis data and the LifeLab data were used for the experiments. From the results it is observed that for Decision Tree (j48) classification with 10 fold cross validation, the sensitivity of the imbalanced thesis data was 20% with 89.76% specificity. The sensitivity of 84% and 67.5% specificity was found with one of the dataset balanced by the proposed method. Moreover, 79% sensitivity and 40% specificity was found when the balanced data was used to train the classifier and the full dataset was used as testing set. It is observed that the specificity dropped down if the balanced data was used as a training set. The experimental results of the Lifelab were also very impressive. For Decision Tree (j48) classification with 10 fold cross validation, the sensitivity of the imbalanced LifeLab data was 35% and 75% with the data balanced by the proposed method. Furthermore, the 96% classification sensitivity was found when the balanced data was used as training and the full original imbalanced data was used as a testing set.

The classification model built by the data balanced by the proposed method was found to be reliable to classify the target class (high risk patients) but not good enough to classify the low risk patients. This is likely because that the class labels do not truly reflect the property of the patients (see section 2.7). The classification model built by the balanced data set was found to be good enough to classify the high risk (target class) records from the unknown dataset. Moreover, this is also the ultimate goal of a clinical risk prediction model.

The results show that the proposed method of under-sampling not only can balance the data for better classification but also can select good training samples for building reliable classification models.

7.5 Summary

This chapter presented experiments on the sampling techniques like SMOTE and some under-sampling techniques over the cardiovascular data and other datasets. The results were compared with the proposed cluster based under-sampling technique. It is found that the proposed modified cluster based under-sampling method not only can balance the data but also can generate good quality training sets for building classification models.

The outcome labels of most of the clinical datasets are not consistent with the underlying data. If we consider the principal data set used in the thesis, where cardiovascular risk is based on whether previous patients records display dead or alive, it appears some of the patients may have died due to causes other than cardiovascular risk; conversely some high risk cardiovascular patients appear to be alive. Both situations confound the class imbalance problem. The conventional over-sampling and under-sampling technique may not always be appropriate for such datasets.

The proposed method is found to be useful for such datasets where the class labels are not certain and can also help to overcome the class imbalance problem of clinical datasets and also for other data domains. The next chapter will present the experimental results on feature selection.

CHAPTER 8 : EXPERIMENTS ON FEATURE SELECTION

8.1 Introduction

This chapter will present the experimental results of feature selection for imbalanced datasets. As discussed in chapter 4, both the filter and wrapper methods of feature selection use the class label of the dataset to select the attribute subset. For highly imbalanced data, the class imbalance problem not only affects classification but also can affect the feature selection process. Our research found that very few work is done in this area. Tian-yu (2009) tried some standard feature selection method on some class imbalance data and compare which method of feature selection good for their experimental data. Khoshgoftaar et al. (2010) proposed repetitive feature selection method for imbalance data, where they used random under-sampling to balance the data. In the work of Al-Shahib et al. (2005), the author used random under-sampling to balance their data for SVM classification. They found that SVM performed well on the balanced data.

A case study was prepared to examine the effect of the class imbalance problem on feature selection. Experiments are done based on the feature selection framework proposed in section 5.7.1.

RELIEF-F feature selection is used for most of the experiments and the detail of the algorithm is described in section 4.4. Information Gain based feature ranking is also used for comparisons. The purposes of the experiments are not to compare the feature selection technique RELIEF-F and Information gain. The main purposes of the experiments are to study the effect of class imbalance problem on feature selection and evaluate the proposed feature selection framework discussed in section 5.7.1. The thesis cardio vascular data was used for all the experiments in this chapter. As discussed in chapter 5, the thesis data has 22 input attributes and 1 class attribute, which is the patient status "alive" or "death". The original data is highly imbalanced with regard to the class label. Out of 823 records 120 patients are dead and 703 patients are alive.

8.2 Attribute ranking of thesis data with RELIEF-F and Information Gain

This section demonstrates the attribute ranking of the thesis data. RELIEF-F and Information Gain are used to rank the attributes of the thesis data described in section 5.2. The details of the feature selection techniques are already discussed in chapter 4. Figures 8.1 presents the rank of attributes of the imbalanced thesis data set based on information gain based attribute selection and RELIEF-F feature ranking as discussed in section 4.4.

From the figure 8.1 it is observed that the RELIEF-F produces much higher values for some of the attributes over information gain. For example the RELIEF-F ranked the attribute "Carotid_status" as first whereas the first ranked by the information gain is the attribute "patch".

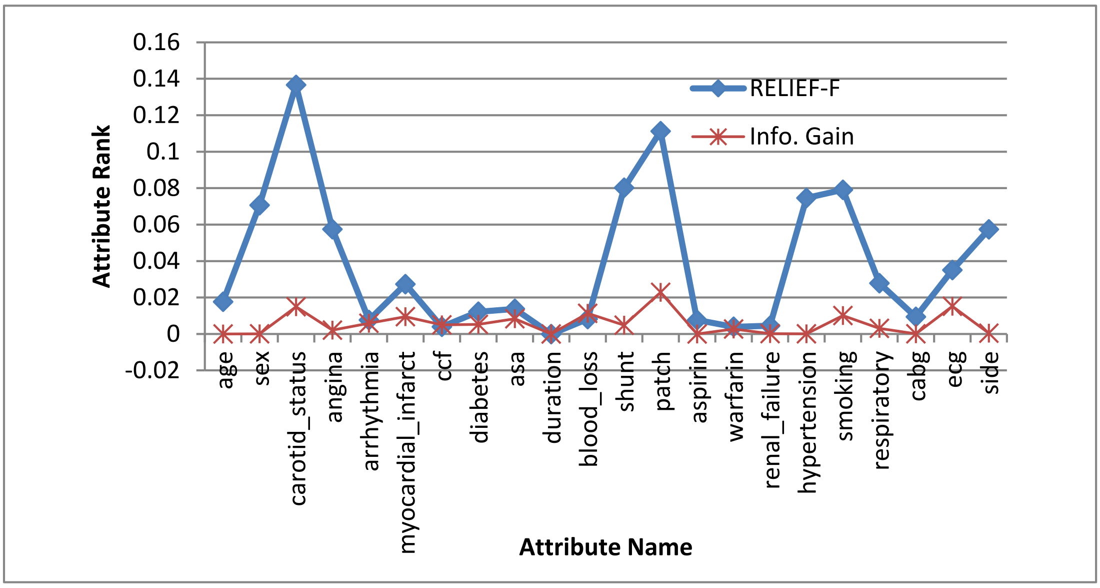


Figure 8-1: Attribute ranking of imbalanced data by information gain and RELIEF-F

The attribute subsets ranked by RELIEF-F and information gain were later classified using Decision Tree (J48). Different attribute subsets were considered; in each case one attribute was removed from the bottom of the list and a new subset was prepared for classification. The process was repeated and Decision Tree (J48) was run for each attribute subset containing 22 attributes down to 1 attribute. The sensitivity value (see section 3.6.1) was calculated and presented in figure 8.2.

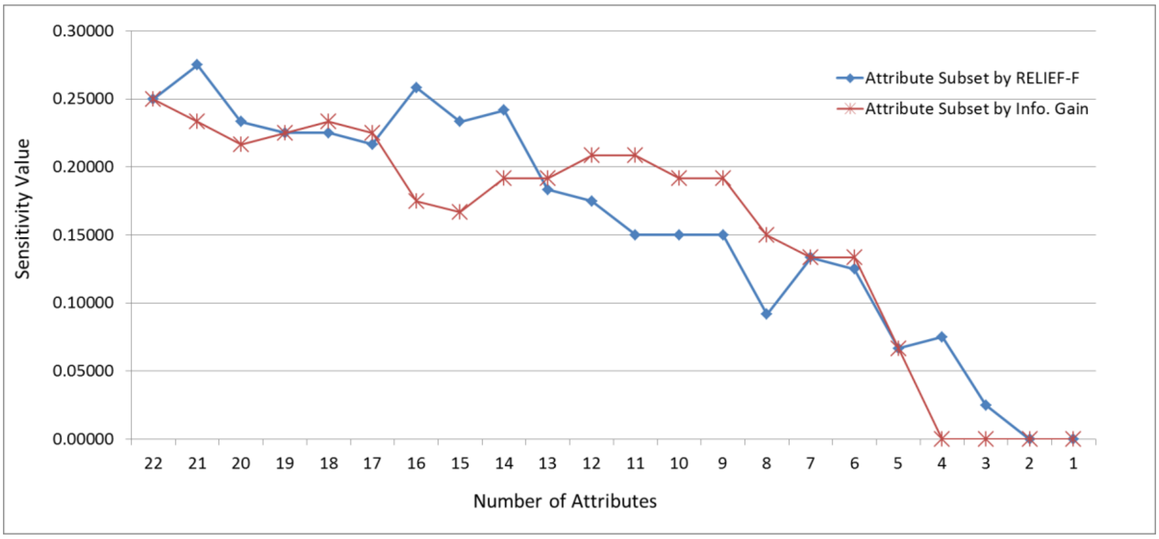


Figure 8-2: Sensitivity of Decision Tree (J48) classification of different attribute subset by RELIEF-F and Information Gain

Figure 8.2 shows the sensitivity value of different attribute subsets classified using J48. The attribute number 22 is the dataset contains the top (RELIEF-F ranked) 22 attributes and attribute number 1 is the dataset contain only one attribute which the top most attribute ranked by RELIEF-F. From the figure 8.2 it can be seen that for both of the datasets using all the attributes the sensitivity is 25%. The sensitivity value went down when one attribute was removed from bottom of the information gain ranked list. On the other hand the RELIEF-F ranked attributes goes high with one less attribute. For RELIEF-F ranked attribute only 14 attributes were needed to get the expected sensitivity (sensitivity value found using all the attributes), compared to 18 attributes needed for information gain ranked attributes. The results show that the RELIEF-F is good at ranking the attributes compared to information gain. Later experiments are designed to see the attribute rank of imbalanced data compared to data balanced by some class balancing techniques.

8.3 Attribute ranking of balanced datasets

This section details experiments on attribute ranking of the thesis data balanced by different class balancing techniques. SMOTE (see section 5.6.1) was used to balance the data by over-sampling the minority samples. Three other under-sampling techniques, random under-sampling, cluster based under-sampling by Lee (2009) and our proposed under-sampling techniques were used to under-sample the majority class data set. Details of the techniques are described in chapter 5, section 5.6 to 5.7. The attributes of the datasets are later ranked using RELIEF-F and information gain. The ranking of the attributes are given in table 8.1 and table 8.2. Figure 8.3 and figure 8.4 plot the attributes ranking compared to their average ranking. The average ranking of each attribute was calculated by making the average ranking of the individual attribute ranking in all the different datasets. This is done to see how the ranking values are deviating from their average ranking.

Table 8.1: RELIEF-F Attribute Ranking Of Different Balanced Data

Attribute Name	Ranking of the attributes					
	Original Data	Proposed Method	Under-sampling by Lee (1:1)	Random Under-sampling	SMOTE Over-sampling (1:1)	Average Ranking
shunt	3	5	1	3	3	3
carotid_status	1	4	9	1	2	3.4
patch	2	10	4	2	1	3.8
smoking	4	3	8	5	6	5.2
Sex	6	1	5	8	7	5.4
angina	7	7	2	6	5	5.4
hypertension	5	6	7	4	11	6.6
side	8	2	22	7	4	8.6
Ecg	9	14	3	10	9	9
respiratory	10	11	6	9	14	10
Age	12	8	11	13	8	10.4
diabetes	8	13	14	12	15	12.4
Asa	13	12	17	11	13	13.2
myocardial_infarct	11	9	21	17	12	14
blood_loss	16	21	12	14	10	14.6
cabg	15	19	13	21	16	16.8
Ccf	20	15	15	20	20	18
aspirin	18	22	18	15	17	18
duration	22	17	10	22	22	18.6
arrhythmia	21	18	19	16	19	18.6
renal_failure	19	20	16	19	21	19
warfarin	21	16	20	18	21	19.2

Table 8.2: Information Gain attribute ranking of different balanced data

Attribute Name	Ranking of the attributes					
	Original Data	Proposed Method	Under-sampling by Lee (1:1)	Random Under-sampling	SMOTE Over-sampling (1:1)	Average Ranking
patch	1	4	5	1	3	2.8
carotid_status	3	5	2	3	14	5.4
smoking	5	3	7	5	8	5.6
ecg	2	7	1	2	16	5.6
myocardial_infarct	6	9	8	6	10	7.8
shunt	11	16	3	8	5	8.6
angina	8	17	4	12	6	9.4
blood_loss	4	22	6	4	13	9.8
diabetes	9	14	9	11	9	10.4
sex	18	1	17	16	4	11.2
side	15	2	14	14	11	11.2
arrhythmia	8	11	11	7	19	11.2
hypertension	17	8	13	15	7	12
respiratory	12	10	15	10	15	12.4
ccf	10	19	10	9	20	13.6
asa	7	21	21	21	2	14.4
warfarin	13	15	12	13	21	14.8
age	21	6	20	20	12	15.8
cabg	19	12	18	18	17	16.8
renal_failure	16	13	16	17	22	16.8
duration	22	20	22	22	1	17.4
aspirin	20	18	19	19	18	18.8

Table 8.1 presents the RELEIEF-F attribute rank of all the datasets prepared by different class balancing methods. Table 8.2 presents the information gain based ranking of all the datasets. The topmost attribute is ranked as "1" and the bottom most attribute is ranked as "22". The first column of the tables is the name of the attributes and subsequent columns are the ranking value.

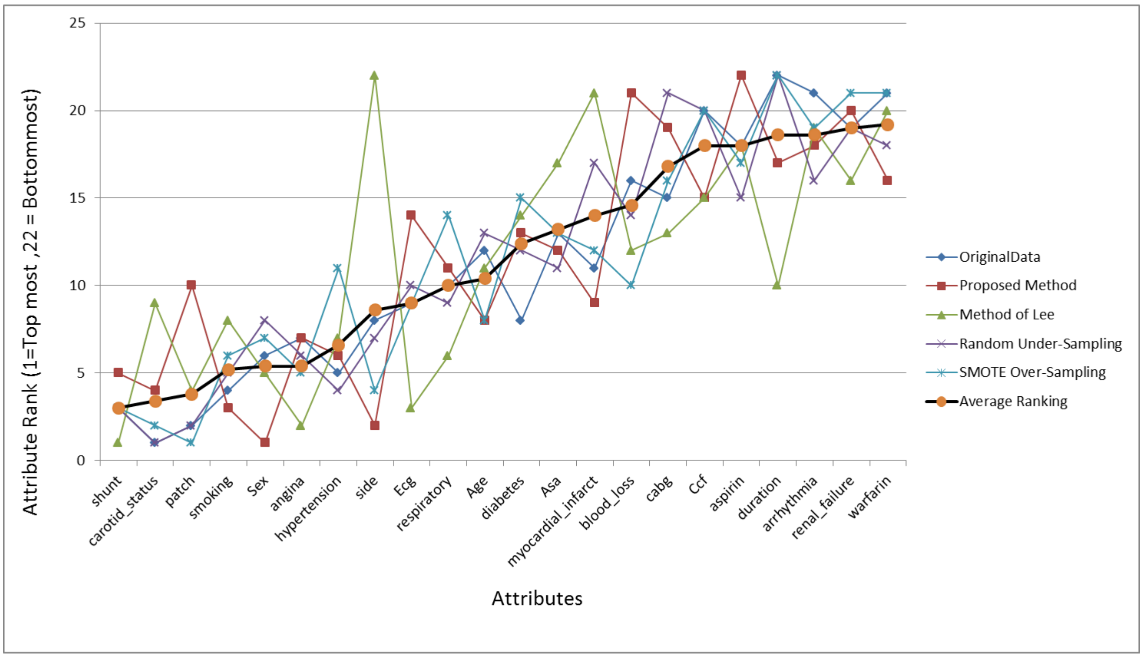


Figure 8-3: Attribute ranking of all the balanced and imbalanced datasets by RELIEF-F

Figure 8.3 presents the attribute rank of all the imbalanced and balanced datasets ranked by RELIEF-F. The average ranking of the five datasets is also plotted in the figure. From figure 8.3 we can see that the average top ranked attribute is "shunt" and average bottommost attribute is "Warfarin". For the original dataset which is highly imbalanced with regard to the class label, some attributes are ranked close to the average. However others have a big disagreement with the average ranking; for example, the attribute "Diabetes", which is ranked as 8 for the original dataset and the average ranking of that attribute for the other datasets is 12.4. When the dataset is prepared by the thesis proposed data balancing technique RELIEF-F ranked the "Sex" attribute as the top most attribute; however the average ranking of that attribute is 5.4. Apart from the attribute "Sex", the rank of attributes "Patch", "blood_loss" and "Side" have big disagreements with attribute ranking from other datasets. The average ranking of the attribute "Patch" is 3.4 however it is ranked as 10 for the dataset prepared by the proposed class balancing method. When RELIEF-F was used on the dataset prepared by SMOTE over-sampling, it is found that apart from the attributes "Hypertension", "Side", "Blood_loss" and "duration" most of the attributes ranking was close to the average. Furthermore, the attribute ranking of the balanced dataset prepared by the class balancing technique proposed by Lee (2009) is very disappointing. Most of its attributes ranking are far away from the average rank. For example, the attribute "Side" which has an average rank of 8.6 but it is ranked as 22 for the dataset prepared by the method of Lee (2009). The same is found for other attributes like "carotid_status", "ECG", "respiratory", "myocardial_infarct", "cabg" and "duration", which have big disagreement with their average ranking made by RELIEF-F from other balanced and imbalanced datasets.

Figure 8.4 presents the attribute rank of all the imbalanced and balanced datasets ranked by information gain. The average ranking of the five datasets is also plotted in the figure.

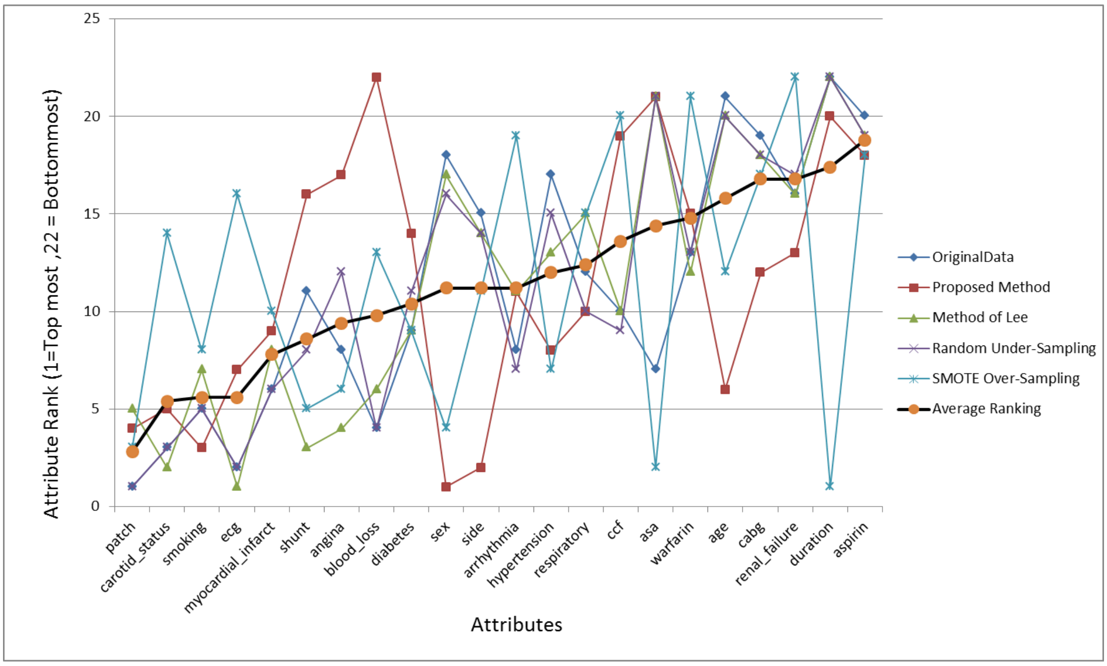


Figure 8-4: Attribute ranking of all the balanced and imbalanced datasets by information gain

From the figure 8.4 it can be seen that the average top ranked attribute is “patch” and average bottommost attribute is “aspirin”. For the original dataset (the highly imbalanced data) only a few attributes are close to their average rank made by information gain from other datasets. Moreover the figure shows high disagreement of attribute ranking made by information gain using all the five datasets prepared by different class balancing techniques. For example, the attribute duration which is ranked as 1 for the dataset prepared by SMOTE, however the average ranking of that attribute is 17.4. The attribute ranking using is information gain is truly disappointing as very less agreement found between all different datasets. The funding suggest that information gain based feature selection is not suitable for the thesis data.

8.4 Experimental results of the proposed feature selection framework for imbalanced data

This section presents the experimental outcome of the proposed feature selection framework. As described in the section 5.7.1 the framework first used a cluster based under-sampling method to reduce the gap between the majority and minority sample then the RELIEF-F algorithm is used to rank the attributes. Based on the ranking made by RELIEF-F, a total of 21 data subsets were prepared. Datasets are later classified using Decision Tree (J48) with 10 fold cross validation. The subsets were prepared by selecting the top (*n-1)* attributes, where *n* is the number of attributes. The Decision Tree (J48) was applied to all the data subsets and sensitivity value was calculated in each case. The same steps were also performed on the balanced data. This is done to compare the outcome of the balanced data with the original imbalanced data. The outcome of the experiment is plotted in the figure 8.5.

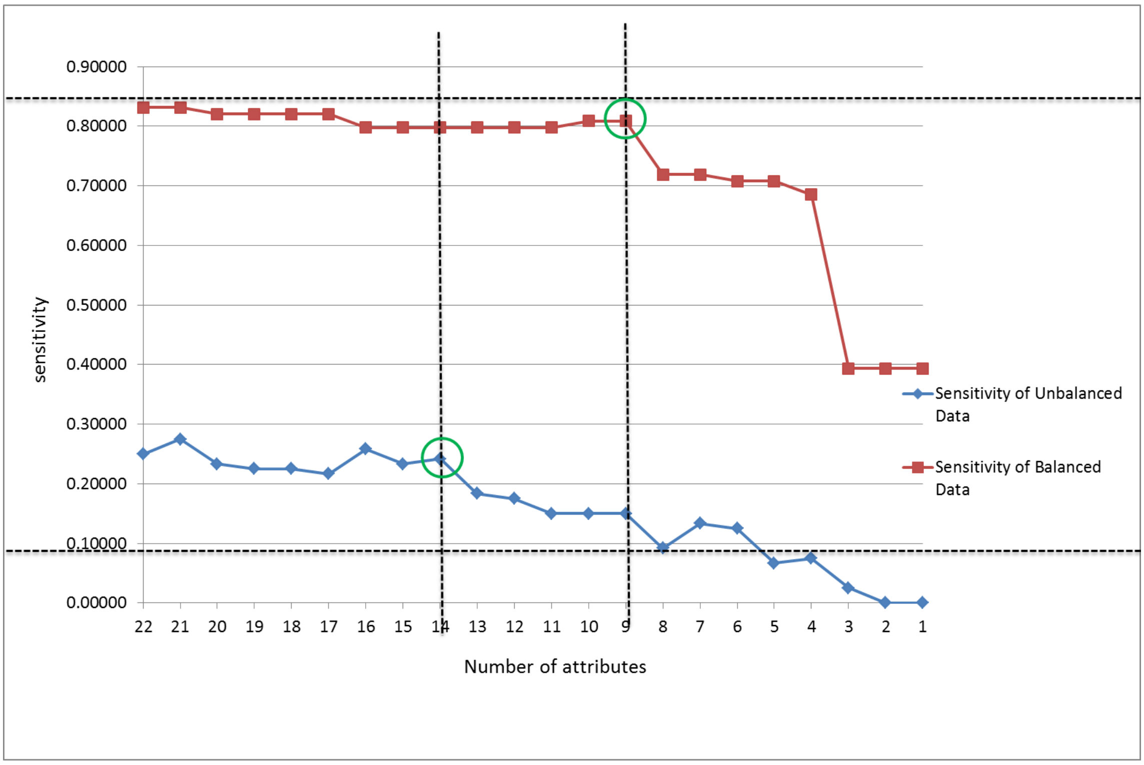


Figure 8-5: Sensitivity value of attribute subsets of the imbalanced data and balanced data ranked by RELIEF-F

Figure 8.5 presents the sensitivity value of Decision Tree (J48) classification of all the attributes ranked by the RELIEF-F. From the figure it can be seen that the balanced data by the proposed data balancing method is not only producing high sensitivity compared to the original imbalanced data but also using less attributes to keep a high sensitivity value. For Decision Tree (J48) classification of the imbalanced data, a minimum of 14 attributes are needed to get a good classification outcome. Only 9 attributes are required for the balanced data (upper line in Figure 8.5). A statistical test (F-Test) was made to see if the results are statistically significant different. An F-test (Harper 1984) is a statistical test in which the test statistic has an F-distribution under the null hypothesis. It is most often used when comparing statistical models that have been fitted to a data set, in order to identify the model that best fits the population from which the data were sampled. Exact "F-tests" mainly arise when the models have been fitted to the data using least squares. F-Test of the classification outcome of balanced and imbalanced dataset is presented in the table 8.3.

Table 8.3: F-Test of two ranges of sensitivity values

	Balanced Dataset	*Imbalanced Dataset*
Mean	0.73033708	0.158333333
Variance	0.02078862	0.00728836
Observations	22	22
df	21	21
F	2.85230444	
P(F<=f) one-tail	0.01010094	
F Critical one-tail	2.08418862	

From the table 8.3 it can be seen that the F value is higher than the “F Critical one-tail” which shows that the outcome of the two models are statistically significantly different.

A further experiment was done by replacing the RELIEF-F ranking with information gain ranking. This is done just to compare different feature ranking method for the proposed feature selection framework for imbalanced datasets. The experimental outcome is plotted in the figure 8.6, which presents the sensitivity value of Decision Tree (J48) classification of all the attributes ranked by the information gain. From the figure it can be seen that the balanced data by the proposed data balancing method is not only producing high sensitivity compared to the original imbalanced data but also using less attributes to keep its high sensitivity value. For imbalanced data, 17 attributes are needed to keep a high sensitivity. However, there is a need of only 7 attributes for the balanced data.

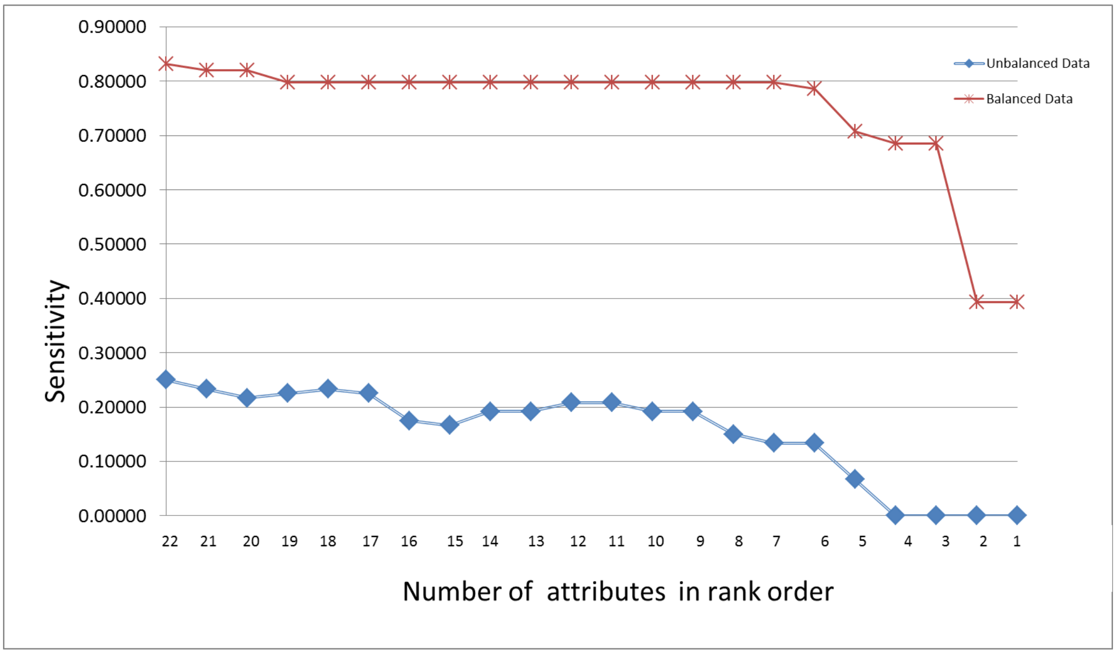


Figure 8-6: Sensitivity value of attribute subsets of the imbalanced data and balanced data ranked by information gain

8.5 Discussion

In machine learning problems that involve learning a "state-of-nature" (maybe an infinite distribution) from a finite number of data samples in a high-dimensional feature space with each feature having a number of possible values, an enormous amount of training data are required to ensure that there are several samples with each combination of values. With a fixed number of training samples, the predictive power reduces as the dimensionality increases, and this is known as the Hughes effect or Hughes phenomenon (Oommen et al. 2008). Feature selection, is a process closely related with dimension reduction and addresses the Hughes effect. The objective of feature selection is to identify features in the dataset as important and discard any other feature as irrelevant, where they provide only redundant information. As the feature selection methods use the class label of the dataset to select the attribute subset, the class imbalance problem is found to be an additional problem, particularly while working on dimension reduction for highly imbalanced datasets.

The experiments of this chapter show that distribution of the class label has a big impact on the feature ranking done by an algorithm. This is because most of the feature selection methods use the class label of the dataset to select the attribute subset. As it can be seen from the figure 8.3 and 8.4, attribute ranking made by RELIEF-F and information gain of different dataset with different distribution of the class label are not same. The research found that the classifier needs less attributes to show high accuracy for balanced data, and conversely more attributes are needed if the data is highly imbalanced. The F-test of the results also suggest that the outcome of the two feature selection frame work (feature selection before class balancing and feature selection after class balancing) are statistically significantly different. Based on the experiments of this chapter it found that for reliable classification the class balancing need to be performed before feature selection.

8.6 Summary

This chapter presented experiments on feature selection for the thesis data. Experiments were made to rank the attributes of the thesis data with RELIEF-F and information gain. The chapter also presented experiments on feature selection for imbalanced data with the proposed feature selection framework discussed in chapter 5. The study found that feature rankings are different for the balanced and imbalanced datasets and more attributes are required if imbalanced data is used. Most of the medical data are found to be imbalanced in their class label. Mining imbalanced data for the purpose of clinical decision support is a challenging issue. This research finding shows that the class imbalance is not only an issue for the classifier but also a big issue for feature selection. The findings show that class balancing enables reliable feature selection. The research suggests the class balancing needs to be performed before feature selection for reliable feature selection and classification of clinical data, and perhaps similarly so for other domain data.

CHAPTER 9 : CONCLUSION

9.1 Introduction

The application of data mining to medical and health data is very challenging. The datasets usually are very large, complex, heterogeneous, hierarchical and vary in quality. Sittig et. all (2008) placed the grand challenges of Clinical Decision Support into three large categories: Improve the effectiveness of Clinical Decision Support interventions, Create new Clinical Decision Support interventions and Disseminate existing Clinical Decision Support knowledge and interventions. However Sittig et al.'s identification covers little about data pre-processing. Without quality data these three aims cannot be realised.

Sometimes, improved data quality is itself the goal of the analysis, usually to improve processes in a production database and the designing of decision support. Many other researchers mention several other issues of clinical decision support related to clinical data and data pre-processing. The ones most relevant to this thesis were high volume of data, data update, inconsistent data representation, number of variables, missing/incomplete data, and class imbalance. As there are a number of issues with medical data mining, this research was limited to addressing the issues with missing value, issues with class imbalance and the effect of class imbalance for feature selection.

The aim of the thesis research was to investigate suitable data pre-processing techniques for medical data mining, in particular the following two research questions:

a) How can data pre-processing be improved for medical data mining?
b) What forms of techniques (and metrics) are useful for determining data cleansing, feature reduction and classification?

This chapter presents the conclusions for the research questions initially presented in Chapter 1. The thesis successfully addressed the above research questions and proposed three novel techniques to improve data preprocessing for the purpose of medical data mining. The thesis also proposed an improve data mining methodology for mining medical data.

9.2 Thesis Contributions

The research described in the thesis contains flowing four novel main contributions.

9.2.1 Contribution 1: Improved data mining methodology for mining medical data

A primary objective for this research was the investigation of systematic data preparation techniques for data cleansing and feature reduction. Chapter 2 introduced a data mining methodology that was used a guideline for the research described in the thesis. On the basis of the results obtained, it is perhaps prudent to revisit that methodology. The research findings suggest that the data pre-processing has an important part in the medical data mining methodology with proper order of the pre-processing tasks. An improved data mining methodology for the medical data mining is given section 2.4.4.

The research suggests that the “class balancing” needs to be performed before the feature selection. Through the experiments it is found that a subsequent feature selection method can select less number of attribute if the data is balanced.

9.1.2 Contribution 2: Machine learning based missing value imputation method

A machine learning based missing value imputation method has been proposed. A quantitative study shows that final classifier performance is improved when the machine learning algorithm is used to predict missing attribute values. In most cases, the proposed machine learning techniques were found to perform better than when the standard means imputation technique was used. The work was presented in a high quality conference at Imperial College, London and awarded the best paper award 2012 (IAENG 2012). The extended version of the work is published as a book chapter by Springer (M. M Rahman and D. N. Davis 2013).

9.2.3 Contribution 2: Cluster Based semi-supervised class balancing method

Most medical datasets are not balanced with regard to their class labels. Most existing classification methods tend not to perform well on minority class examples when the dataset is extremely imbalanced. This is because they aim to optimize the overall accuracy without considering the relative distribution of each class. A cluster based semi-supervised under-sampling technique was proposed, that solves the class imbalance problem for our cardiovascular data and also shows significant better performance than the existing methods. The work is presented in a conference (M. M. Rahman and D. N. Davis 2013b) and also published in a journal (M. M. Rahman and D. N. Davis 2013a).

9.7.4 Contribution 4: Feature selection framework for class imbalance datasets

An empirical study was made to examine the effect of class imbalance problem on the feature selection for the medical datasets. The findings show that the class distribution has an impact on the feature selection. Most of the feature selection techniques use the class label as one of the metrics to select the optimal attribute subset. The class imbalance has adverse effect on the feature selection process. The thesis proposed a feature selection framework for imbalanced clinical datasets. The work is accepted to be published as a book chapter by Springer. (See section "Public Output" for the complete reference).

9.8 Summary and future work

There are many issues with medical data mining and the data pre-processing is most challenging issue among them. Sometimes, improved data quality is itself the goal of the analysis, usually to improve processes in a production database and the designing of decision support. This research addressed the issues with missing value, class imbalance and feature selection for imbalance datasets. For each of the issues, the thesis proposed new methods to deal with them.

Many machine learning algorithms were used in the thesis for data cleansing (missing value imputation and class balancing), feature selection and classification of the thesis data. For the classification problem several classifiers like Multi-Layer Perceptron, Support Vector Machine (SVM), Decision Tree (J48), KNN, Ripple-down rules (Ridor) and Fuzzy Unordered Rule Induction Algorithm were used. For most of the cases it is found that Decision Tree (J48), Fuzzy Unordered Rule Induction Algorithm and KNN performed better than the other classifiers on the main thesis data. Decision Tree (J48) and Fuzzy Unordered Rule Induction Algorithm performed better when they were used as a classifier for the proposed missing value imputation technique. It is also found that in most cases the proposed machine learning based missing value imputation outperformed the standard Mean/Mode imputation.

SMOTE over-sampling, random over-sampling, random under-sampling, cluster based under-sampling by Lee (2009) and thesis proposed under-sampling were used to balance the imbalanced thesis data and two other datasets (Life_Lab and Indian_Liver). The experiments found that for the original imbalanced data the highest sensitivity was 30% with FURIA classification and 20% with Decisions Tree (J48) classification. SMOTE over-sampling was found to produce high accuracy (83% accuracy and 82% sensitivity), however taking longer to train. The under-sampling by Lee (2009) takes less training time and could produce better results than the original imbalanced data by 63% with FURIA classification. The proposed class balancing technique outperformed all the other techniques explored in the research. The highest sensitivity was found as 93.26% when data was balanced by the proposed method and classified using Decision Tree (J48).

For further research the proposed missing value imputation method can be tested on the data of other domains. The main limitation of the method is there is a need of an adequate number of complete records in the dataset for building the prediction model. Further research is required to find out a solution for the model to work with all the dataset even with unavailability of the complete data. It would also be interesting to see if there is any effect on the proposed method on the stratified dataset. Where, learning and imputation will be made separately on each class of data records.

Due to the limitation of time the proposed feature selection framework was only tried with the RELIEF-F and information gain feature selection techniques. Further research can also be made to observe the effect of class imbalance problem on the other well-known dimension reduction techniques with different domain datasets.

Clinical decision support are computer systems designed to impact clinician decision making about individual patients at the point in time that these decisions are made (Berner 2007). A typical decision support system consists of five major components: the data management, the model management, the knowledge engine, the user interface, and the user(s). Data management component is the base of the clinical decision support system. Knowledge engine is built through mining the clinical data. The datasets usually are very large, complex, heterogeneous, and hierarchical and vary in quality. Data pre-processing and transformation are required even before mining and discovery can be applied. The reliable data management and data preparation is very important for mining knowledge from the clinical datasets. This research addressed the issues with missing value, class imbalance and feature selection for class imbalance datasets and proposed new methods. The performance analysis and comparative study show that the proposed method of missing value imputation, class balancing and feature selection framework provide an effective approach to data preparation for building medical decision support.

REFERENCES

Aeinfar, V., Mazdarani, H., Deregeh, F., Hayati, M. and Payandeh, M. (2009) 'Multilayer Perceptron Neural Network with supervised training method for diagnosis and predicting blood disorder and cancer', in *IEEE International Symposium on Industrial Electronics*, 2075-2080.

Aha, D. W., Kibler, D. and Albert, M. K. (1991) 'INSTANCE-BASED LEARNING ALGORITHMS', *Machine Learning,* 6(1), 37-66.

Al-Shahib, A., Breitling, R. and Gilbert, D. (2005) 'Feature selection and the class imbalance problem in predicting protein function from sequence', *Appl Bioinformatics,* 4(3), 195-203.

Almeida, R. J., Kaymak, U. and Sousa, J. M. C. (2010) 'A new approach to dealing with missing values in data-driven fuzzy modelling', in *IEEE International Conference on Fuzzy Systems (FUZZ)*, Barcelona, IEEE, 1 - 7.

Bai, Y., Zhuang, H. and Wang, D. (2006) *Advanced fuzzy logic technologies in industrial applications,* London: Springer.

Baraldi, A. N. and Enders, C. K. (2010) 'An introduction to modern missing data analyses', *Journal of School Psychology,* 48(1), 5-37.

Bates, D. W., Kuperman, G. J., Wang, S., Gandhi, T., Kittler, A., Volk, L., Spurr, C., Khorasani, R., Tanasijevic, M. and Middleton, B.

(2003) 'Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality', *J Am Med Inform Assoc,* 10(6), 523-30.

Batista, G. and Monard, M. C. (2003) 'An analysis of four missing data treatment methods for supervised learning', *Applied Artificial Intelligence,* 17(5-6), 519-533.

Becker, E., Robisson, B., Chapple, C. E., Guenoche, A. and Brun, C. (2012) 'Multifunctional proteins revealed by overlapping clustering in protein interaction network', *Bioinformatics,* 28(1), 84-90.

Bellamy, L., Casa, J. and Hingorani, A. (2007) 'Preeclampsia and risk of cardiovascular disease and cancer in later life: systematic review and metaanalysis', *BMJ,* 335, 974-977.

Bellazzi, R. and Zupan, B. (2008) 'Predictive data mining in clinical medicine: Current issues and guidelines', *International Journal of Medical Informatics,* 77(2), 81-97.

Benkaci, M., Jammes, B. and Doncescu, A. (2010) 'Feature Selection for Medical Diagnosis Using Fuzzy Artmap Classification and Intersection Conflict', in *IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, 790-795.

Berner, E. S. (2007) 'Clinical decision support systems theory and practice', [online], available: http://www.myilibrary.com?id=81629 [accessed

Bishop, C. M. (1995) *Neural networks for pattern recognition,* Oxford; New York: Clarendon Press ; Oxford University Press.

Blum, M. G. B., Nunes, M. A., Prangle, D. and Sisson, S. A. (2013) 'A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation', *Statistical Science,* 28(2), 189-208.

Bouckaert, R. R., Frank, E., Hall, M. A., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2010) 'WEKA-Experiences with a Java Open-Source Project', *Journal of Machine Learning Research,* 11, 2533-2541.

Bouyssou, D. D. D. P. H. P. M. (2010) 'Decision Making Process Concepts and Methods', [online], available: http://public.eblib.com/EBLPublic/PublicView.do?ptiID=477628 [accessed

Brian, R., Gaines. and Paul, C. (1995) 'Induction of Ripple-Down rules applied to modelling large databases', *Journal of Intelligent information system,* 5(3), 221-228.

Bruha, I. (2004) 'Meta-Learner for Unknown Attribute Values Processing: Dealing with Inconsistency of Meta-Databases', *J. Intell. Inf. Syst.,* 22(1), 71-87.

Bunkhumpornpat, C., Sinapiromsaran, K. and Lursinsap, C. (2009) 'Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem', *Advances in Knowledge Discovery and Data Mining, Proceedings,* 5476, 475-482.

Catley, C., Smith, K., McGregor, C. and Tracy, M. (2009) *Extending CRISP-DM to incorporate temporal data mining of multidimensional medical data streams: A neonatal intensive care unit case study,* translated by 1-5.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002) 'SMOTE: Synthetic Minority Over-sampling Technique', *Journal of Artificial Intelligence Research,* 16, 321–357.

Chawla, N. V., Lazarevic, A., Hall, L. O. and Bowyer, K. W. (2003) 'SMOTEBoost: Improving prediction of the minority class in boosting' in Lavrac, N. G. D. T. L. B. H., ed. *Knowledge Discovery in Databases: Pkdd 2003, Proceedings*, 107-119.

Chen, X., Ye, Y., Xu, X. and Huang, J. Z. (2012) 'A feature group weighting method for subspace clustering of high-dimensional data', *Pattern Recognition,* 45(1), 434-446.

Chyi, Y.-M. (2003) *Classification analysis techniques for skewed class distribution*

problems, unpublished thesis National Sun Yat-Sen University.

Cios, K. J. and Moore, G. W. (2002) 'Uniqueness of medical data mining' in *Artif Intell Med*, Netherlands: 1-24.

Daelemans, W., Hoste, V., Meulder, F. and Naudts, B. (2003) 'Combined Optimization of Feature Selection and Algorithm Parameters in Machine Learning of Language' in Lavrač, N., Gamberger, D., Blockeel, H. and Todorovski, L., eds., *Machine Learning: ECML 2003*, Springer Berlin Heidelberg, 84-95.

Dasu, T. and Johnson, T. (2003) *Exploratory data mining and data cleaning,* New York: Wiley-Interscience.

Davis, D. N. and Nguyen, T. T. T. (2008) 'Generating and Veriffying Risk Prediction Models Using Data Mining (A Case Study from Cardiovascular Medicine)', in *European Society for Cardiovascular Surgery 57th Annual Congress of ESCVS*, Barcelona Spain,

Devendran, V., Hemalatha, T. and Amitabh, W. (2008) 'Texture based Scene Categorization using Artificial Neural Networks and Support Vector Machines: A Comparative Study', *ICGST-GVIP,* 8(5).

Eurorec (2014) 'European Institute for EHealth Records', [online], available: http://www.eurorec.org/tools/index.cfm [accessed

Fayyad, U. M. (1996) *Advances in knowledge discovery and data mining,* Menlo Park, Calif.: AAAI Press : MIT Press.

Fazeli, S., Naghibolhosseini, M. and Bahrami, F. (2008) *An Adaptive Neuro-Fuzzy Inference System for Diagnosis of Aphasia,* translated by 535-538.

Fox, J., Glasspool, D., Patkar, V., Austin, M., Black, L., South, M., Robertson, D. and Vincent, C. (2010) 'Delivering clinical decision support services: there is nothing as practical as a good theory' in *J Biomed Inform*, United States: 831-43.

Gajawada, S. and Toshniwal, D. (2012) 'Missing Value Imputation Method Based on Clustering and Nearest Neighbours', *International Journal of Future Computer and Communication,* 1(2), 206-208.

Gustavo Batista, M. C. M. (2003) 'A Study of K-Nearest Neighbour as an Imputation Method', in *In: Proceedings of HIS*,

Gustavo Batista, M. C. M. (2013) *A Study of K-Nearest Neighbour as an Imputation Method,* translated by.

Guyon, I. (2006) 'Feature extraction foundations and applications', [online], available: http://dx.doi.org/10.1007/978-3-540-35488-8 [accessed

Guyon, I., Andr, #233 and Elisseeff (2003) 'An introduction to variable and feature selection', *J. Mach. Learn. Res.,* 3, 1157-1182.

Guyon I, W. J., Barnhill S, Vapnik V. (2002) 'Gene selection for cancer classification using support vector machines', 46(389-422.

Han, J. and Kamber, M. (2001) *Data mining : concepts and techniques,* San Francisco: Morgan Kaufmann Publishers.

Han, J. and Kamber, M. (2011) *Data mining : concepts and techniques,* Amsterdam [u.a.]: Elsevier/Morgan Kaufmann.

Hand, D. J., Mannila, H. and Smyth, P. (2001) *Principles of Data Mining (Adaptive Computation and Machine Learning),* London: The MIT Press.

Harper, J. F. (1984) 'Peritz F-Test - Basic Program of a Robust Multiple Comparison Test for Statistical-Analysis of All Differences among Group Means', *Computers in Biology and Medicine,* 14(4), 437-445.

Haykin, S. S. (2009) *Neural networks and learning machines,* New York: Prentice Hall/Pearson.

Hernandez, L. A., Bonilla, E., Morales, R., Hernandez, J. C. and Diaz, A. (2013) 'Hybrid Multiple-Filter-Multiple-Wrapper Method Used for Gene Selection from Biomedical Database', *Latin America Transactions, IEEE (Revista IEEE America Latina),* 11(1), 609-615.

Honghai, F., Guoshun, C., Cheng, Y., Bingru, Y. and Yumei, C. (2005) 'A SVM Regression Based Approach to Filling in Missing Values' in Khosla, R., Howlett, R. and Jain, L., eds., *Knowledge-Based Intelligent Information and Engineering Systems*, Springer Berlin Heidelberg, 581-587.

Hongjun, L., Setiono, R. and Huan, L. (1996) 'Effective data mining using neural networks', *Knowledge and Data Engineering, IEEE Transactions on,* 8(6), 957-961.

Horn, K. (1990) *GARVAN-ESI : an expert system for the interpretation of thyroid laboratory tests*, unpublished thesis

Hosseinkhah, F., Ashktorab, H., Veen, R. and Owrang O, M. M. (2009) 'Challenges in Data Mining on Medical Databases' in *Database Technologies: Concepts, Methodologies, Tools, and Applications*, IGI Global, 1393-1404.

Howard, B., Horn, V. and Manson, J. (2006) 'Low-fat dietary pattern and risk of cardiovascular disease', *JAMA,* 295(6), 655-666.

Hyvarinen, A. K. J. O. E. (2001) *Independent component analysis,* New York: J. Wiley.

Hühn, J. and Hüllermeier, E. (2009) 'Fuzzy Unordered Rules Induction Algorithm', *Data Mining and Knowledge Discovery,* 19(3), 293-319.

IAENG (2012) 'WCE 2012 Best Paper Awards', [online], available: http://www.iaeng.org/WCE2012/congress_awards.html [accessed

Iida, S., Miki, Y., Ono, K., Akahira, J.-i., Suzuki, T., Ishida, K., Watanabe, M. and Sasano, H. (2010) 'Novel classification based on immunohistochemistry combined with hierarchical clustering analysis in non-functioning neuroendocrine tumor patients', *Cancer Science,* 101(10), 2278-2285.

Inc, S. (2000) *CRISP -DM 1.0,* Chicago, Ill.: SPSS Inc.

Jang, J.-S. R., Sun, C.-T. and Mizutani, E. (1997) *Neuro-fuzzy and soft computing : a computational approach to learning and machine intelligence,* Upper Saddle River, NJ: Prentice Hall.

Jensen, R. and Qiang, S. (2009a) *Interval-valued fuzzy-rough feature selection in datasets with missing values,* translated by 610-615.

Jensen, R. and Qiang, S. (2009b) 'New Approaches to Fuzzy-Rough Feature Selection', *Fuzzy Systems, IEEE Transactions on,* 17(4), 824-838.

Jerez, J. M., Molina, I., Garcı´a-Laencina, J. P., Alba, E., Nuria, R., Miguel, M. n., Franco and Leonardo. (2010) 'Missing data imputation using statistical and machine learning methods in a real breast cancer problem', *Artificial Intelligence in Medicine,* 50(2), 105-115.

Jolliffe, I. T. (2002) 'Principal component analysis', [online], available: http://site.ebrary.com/id/10047693 [accessed

Kaburlasos, V. G., Moussiades, L. and Vakali, A. (2009) 'Fuzzy lattice reasoning (FLR) type neural computation for weighted graph partitioning', *Neurocomputing,* 72(10–12), 2121-2133.

Kanzawa, Y., Endo, Y., Miyamoto, S. and Ieee (2011) 'On Hard and Fuzzy c-Means Clustering with Conditionally Positive Definite Kernel' in *Ieee International Conference on Fuzzy Systems*, 816-820.

kdnuggets (2007) 'KDnuggets : Polls : Data Mining Methodology', [online], available: http://www.kdnuggets.com/polls/2007/data_mining_methodology.htm [accessed

Khoshgoftaar, T. M., Kehan, G. and Van Hulse, J. (2010) *A novel feature selection technique for highly imbalanced data,* translated by 80-85.

Kim, m.-s. (2007) 'An Effective Under-Sampling Method for Class. Imbalance Data Problem', in *8th International Symposium on Advance intelligent System (ISIS 2007)*,

Kira, K. and Rendell, L. A. (1992) 'A practical approach to feature selection', in *Proceedings of the ninth international workshop on Machine learning*, Aberdeen, Scotland, United Kingdom, 142034: Morgan Kaufmann Publishers Inc., 249-256.

Kochurani, O. G., Aji, S. and Kaimal, M. R. (2007) *A Neuro Fuzzy Decision Tree Model for Predicting the Risk in Coronary Artery Disease,* translated by 166-171.

Kononenko, I. and Kukar, M. (2007) *Machine learning and data mining : introduction to principles and algorithms,* Chichester: Horwood Publishing.

Kullback, S. and Leibler, R. A. (1951) 'On Information and Sufficiency', *The Annals of Mathematical Statistics,* 22(1), 79-86.

Kuncheva, L. I. (2000) *Fuzzy classifier design,* Heidelberg; New York: Physica-Verlag.

Latifoğlu, F., Polat, K., Kara, S. and Güneş, S. (2008) 'Medical diagnosis of atherosclerosis from Carotid Artery Doppler Signals using principal component analysis (PCA), k-NN based weighting pre-processing and Artificial Immune Recognition System (AIRS)', *Journal of Biomedical Informatics,* 41(1), 15-23.

Laza, R., Pavon, R., Reboiro-Jato, M. and Fdez-Riverola, F. (2011) 'Evaluating the effect of unbalanced data in biomedical document classification', *Journal of integrative bioinformatics,* 8(3), 177.

Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H. and Nowe, A. (2012) 'A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis', *Computational Biology and Bioinformatics, IEEE/ACM Transactions on,* 9(4), 1106-1119.

Lee, C. and Lee, G. G. (2006) 'Information gain and divergence-based feature selection for machine learning-based text categorization', *Information Processing & Management,* 42(1), 155-165.

Lele, S. and Richtsmeier, J. T. (1995) 'EUCLIDEAN DISTANCE MATRIX ANALYSIS - CONFIDENCE-INTERVALS FOR FORM AND GROWTH DIFFERENCES', *American Journal of Physical Anthropology,* 98(1), 73-86.

Liao, T. W. and Li, D. M. (1997) 'Two manufacturing applications of the fuzzy K-NN algorithm', *Fuzzy Sets and Systems,* 92(3), 289-303.

Little, R. and Rubin, D. (2002) *Statistical analysis with missing data,* Second ed.*,* New Jersey: John Wiley & Sons, Inc.

Liu, A. Y.-c. (2004) *The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets*, unpublished thesis The University of Texas at Austin.

Liu, Y., Yu, X. H., Huang, J. X. and An, A. J. (2011) 'Combining integrated sampling with SVM ensembles for learning from imbalanced datasets', *Information Processing & Management,* 47(4), 617-631.

Lotte, F., Lecuyer, A. and Arnaldi, B. (2007) 'FuRIA: A Novel Feature Extraction Algorithm for Brain-Computer Interfaces using Inverse Models and Fuzzy Regions of Interest', in *3rd International IEEE/EMBS Conference on Neural Engineering, CNE '07* 175-178.

Maimon, O. and Rokach, L. (2010) *Data mining and knowledge discovery handbook,* Berlin: Springer.

Marlin, B. M. (2008) *Missing Data Problems in Machine Learning*, unpublished thesis University of Toronto.

Marsala, C. (2009) 'A fuzzy decision tree based approach to characterize medical data', in *IEEE International Conference on Fuzzy Systems*, 1332-1337.

Matignon, R. and Institute, S. A. S. (2007) 'Data mining using SAS Enterprise miner', [online], available: [accessed

Meesad, P. and Hengpraprohm, K. (2008) 'Combination of KNN-Based Feature Selection and KNNBased Missing-Value Imputation of Microarray Data', in *3rd International Conference on Innovative Computing Information and Control, 2008. ICICIC '08.*, 341.

Merz, C. J. M., P. (1996) 'UCI Repository of Machine Learning Database. Available', [online], available: http://www.ics.uci.edu/~mlearn/MLRepository.html [accessed

Minsky, M. P. S. (1969) *Perceptrons,* Cambridge, Mass.-London.

Mirkin, B. and Nascimento, S. (2012) 'Additive spectral method for fuzzy cluster analysis of similarity data including community structure and affinity matrices', *Information Sciences,* 183(1), 16-34.

Mladenić, D. (2003) *Data mining and decision support : integration and collaboration,* Boston: Kluwer Academic Publishers.

Ng, A. (1998) *On Feature Selection: Learning with Exponentially many Irrelevant Features as Training Example,* translated by San Francisco: 404–412.

Nguyen, T. T. T. (2009) *Predicting Cardiovascular Risks using Pattern Recognition and Data Mining*, unpublished thesis The University of Hull.

Olson, D. L. and Delen, D. (2008) 'Advanced data mining techniques', [online], available: http://dx.doi.org/10.1007/978-3-540-76917-0 [accessed

Oommen, T., Misra, D., Twarakavi, N. C., Prakash, A., Sahoo, B. and Bandopadhyay, S. (2008) 'An Objective Analysis of Support Vector Machine Based Classification for Remote Sensing', *Mathematical Geosciences,* 40(4), 409-424.

Papageorgiou, E. I., Papandrianos, N. I., Apostolopoulos, D. and Vassilakos, P. (2008) 'Complementary use of Fuzzy Decision Trees and Augmented Fuzzy Cognitive Maps for Decision Making in Medical Informatics', in *Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics - Volume 01*, 1372165: IEEE Computer Society, 888-892.

Park, H. and Kwon, H.-C. (2011) 'Improved Gini-Index Algorithm to Correct Feature-Selection Bias in Text Classification', *Ieice Transactions on Information and Systems,* E94D(4), 855-865.

Peng, H., Fulmi, L. and Ding, C. (2005) 'Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy', *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 27(8), 1226-1238.

Pourahmadi, M. (1993) 'On relations between prediction error covariance of univariate and multivariate processes', *Statistics & Probability Letters,* 16(5), 355-359.

Preece, S. J., Goulermas, J. Y., Kenney, L. P. J. and Howard, D. (2009) 'A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data', *Biomedical Engineering, IEEE Transactions on,* 56(3), 871-879.

Quinlan, J. R. (1985) *Induction of decision trees,* [Broadway, N.S.W., Australia]: New South Wales Institute of Technology, School of Computing Sciences.

Quinlan, J. R. (1993) *C4.5 : programs for machine learning,* San Mateo: Morgan Kaufmann.

Rahman, M. M. and Davis, D. N. (2013a) 'Addressing the Class Imbalance Problem in Medical Datasets', *International Journal of Machine Learning and Computing,* 3(2), 224-228.

Rahman, M. M. and Davis, D. N. (2013b) 'Cluster Based Under-Sampling for Unbalanced Cardiovascular Data', *Lecture Notes in Engineering and Computer Science; Proceedings of The World Congress on Engineering*, 1480-1485.

Rahman, M. M. and Davis, D. N. (2013) 'Machine Learning-Based Missing Value Imputation Method for Clinical Datasets' in Yang, G.-C., Ao, S.-l. and Gelman, L., eds., *IAENG Transactions on Engineering Technologies*, Springer Netherlands, 245-257.

Rasheed, S., Stashuk, D. and Kamel, M. (2006) 'Adaptive fuzzy k-NN classifier for EMG signal decomposition', *Medical Engineering & Physics,* 28(7), 694-709.

Reddy, T. A., Devi, K. R. and Gangashetty, S. V. (2012) *Nonlinear principal component analysis for seismic data compression,* translated by 927-932.

Ren, D. and Qiang, S. (2011) *Fuzzy-rough classifier ensemble selection,* translated by 1516-1522.

Robnik, M., ikonja and Kononenko, I. (2003) 'Theoretical and Empirical Analysis of ReliefF and RReliefF', *Mach. Learn.,* 53(1-2), 23-69.

Rosen, K. H. (1999) *Discrete mathematics and its applications,* Boston: WCB/McGraw-Hill.

Ruijter W, W. R., Assendelft W (2009) 'Use of Framingham risk score and new biomarkers to predict cardiovascular mortality in older people: population based observational cohort study', *BMJ,* 338.

Saeys, Y., Inza, I. and Larranaga, P. (2007) 'A review of feature selection techniques in bioinformatics', *Bioinformatics,* 23(19), 2507-17.

Scally, G. (1998) 'Clinical governance and the drive for quality improvement in the new NHS in England', [online], available: http://dx.doi.org/10.1136/bmj.317.7150.61 [accessed

Schalkoff, R. J. (1997) *Artificial neural networks,* New York: McGraw-Hill.

Schiepers, C., Hoh, C. K., Nuyts, J., Wu, H. M., Phelps, M. E. and Dahlbom, M. (2002) 'Factor analysis in prostate cancer: delineation of organ structures and automatic generation of in- and output functions', *Nuclear Science, IEEE Transactions on,* 49(5), 2338-2343.

Shusaku, T. (2000) 'Knowledge discovery in clinical databases and evaluation of discovered knowledge in outpatient clinic', *Information Sciences,* 124, 125-137.

Sittig, D. F., Wright, A., Osheroff, J. A., Middleton, B., Teich, J. M., Ash, J. S., Campbell, E. and Bates, D. W. (2008) 'Grand challenges in clinical decision support' in *J Biomed Inform*, United States: 387-92.

Tantan, L., Fan, W. and Agrawal, G. (2010) *Stratified Sampling for Data Mining on the Deep Web,* translated by 324-333.

Tez, M., Tez, S. and Göçmen, E. (2008) 'Neurofuzzy is Useful Aid in Diagnosing Acute Appendicitis', *World Journal of Surgery,* 32(9), 2126-2126.

Thangavel, S. Q., Pethalakshmi (2006) 'Application of Clustering for Feature selection Based on Rough Set Theory Approach', *AIML Journal,* 16(1).

Tian-yu, L. (2009) 'EasyEnsemble and Feature Selection for Imbalance Data Sets', in *Bioinformatics, Systems Biology and Intelligent Computing, 2009. IJCBS '09. International Joint Conference on*, 3-5 Aug. 2009, 517-520.

Tomek, I. (1976) 'GENERALIZATION OF K-NN RULE', *Ieee Transactions on Systems Man and Cybernetics,* 6(2), 121-126.

Tong, L.-I., Chang, Y.-C. and Lin, S.-H. (2009) 'Using Experimental Design to Determine the Re-Sampling Strategy for Developing a Classification Model for Imbalanced Data' in Wang, H. F. N. M. B. Z. Y. G. D. W., ed. *Proceedings of the Eighth International Conference on Information and Management Sciences*, 646-648.

Tsang-Hsiang, C., Chih-Ping, W. and Tseng, V. S. (2006) *Feature Selection for Medical Data Mining: Comparisons of Expert Judgment and Automatic Approaches,* translated by 165-170.

Tsumoto, S. (2000) *Problems with mining medical data,* translated by 467-468.

Tumer, K. and Ghosh, J. (1996) *Estimating the Bayes error rate through classifier combining,* translated by 695-699 vol.2.

Tutmez, B. (2012) 'Spatial dependence-based fuzzy regression clustering', *Applied Soft Computing,* 12(1), 1-13.

Ubeyli, E. D. and Guler, I. (2005) 'Adaptive neuro-fuzzy inference systems for analysis of internal carotid arterial Doppler signals', *Computers in Biology and Medicine,* 35(8), 687-702.

Vishwanathan, S., Murty, M. N. and Ieee, I. (2002) 'SSVM : A simple SVM algorithm' in *Proceeding of the 2002 International Joint Conference on Neural Networks, Vols 1-3*, 2393-2398.

Wang, T. J., Gona, P., Larson, M. G., Tofler, G. H., Levy, D., Newton-Cheh, C., Jacques, P. F., Rifai, N., Selhub, J., Robins, S. J., Benjamin, E. J., D'Agostino, R. B. and Vasan, R. S. (2006) 'Multiple biomarkers for the prediction of first major cardiovascular events and death', *New England Journal of Medicine,* 355(25), 2631-2639.

WEKA (1999) 'Weka Machine Learning Project in the University of Waikato', [online], available: http://www.cs.waikato.ac.nz/ml/weka/ [accessed

Witten, I. H. and Frank, E. (2011) *Data mining : practical machine learning tools and techniques,* 3rd ed., Amsterdam; Boston, MA: Morgan Kaufman.

Wu, K.-L. (2012) 'Analysis of parameter selections for fuzzy c-means', *Pattern Recognition,* 45(1), 407-415.

Yan-Ping, Z., Li-Na, Z. and Yong-Cheng, W. (2010) *Cluster-based majority under-sampling approaches for class imbalance learning,* translated by 400-404.

Yardimci, A. (2009) 'Soft computing in medicine', *Appl. Soft Comput.,* 9(3), 1029-1043.

Yen, S.-J. and Lee, Y.-S. (2009) 'Cluster-based under-sampling approaches for imbalanced data distributions', *Expert Systems with Applications,* 36, 5718–5727.

Zadeh, L. A. (1965) 'Fuzzy sets', *Information and Control,* 8(3), 338-353.

Zadeh, L. A. (1994) 'Fuzzy logic, neural networks, and soft computing', *Commun. ACM,* 37(3), 77-84.

Zadeh, L. A. (1996) 'Advances in Fuzzy Systems - Applications and Theory', 6.

Zhai, Y., Ma, N., Ruan, D. and An, B. (2011) 'An Effective Over-sampling Method for Imbalanced Data Sets Classification', *Chinese Journal of Electronics,* 20(3), 489-494.

Zhang, J. and Mani, I. (2003) 'kNN approach to unbalanced data distributions: A case study involving information extraction',

Übeyli, E. (2009) 'Adaptive Neuro-Fuzzy Inference Systems for Automatic Detection of Breast Cancer', *Journal of Medical Systems,* 33(5), 353-358.

APPENDIX A: THESIS DATA STRUCTURE

A.1 Hull site data

The following table summarises the data from the Hull clinical site. As can be seen there are 30 numeric, 3 discrete numeric, 23 categorical, 38 Boolean, and 4 date/time typed attributes.

Attribute name	Attribute types	Attribute name	Attribute types
UNIT_NO	Categorical	JVP	Boolean
THEATRE_SESSION_DATE	Date/Time	LEG_OEDEMA	Boolean
CONS	Discrete	PULM_OEDEMA	Boolean
DATE_OF_DEATH	Date/Time	CARDIAC_FAIL	Boolean
Combined	Categorical	HAEMOGLOBIN	Numeric
30D MR	Boolean	WCC	Numeric
30D Ipsi CVA	Boolean	PLATELETS	Numeric
CAUSE_OF_DEATH	Categorical	UREA	Numeric
PhysiolScore	Numeric	CREATININE	Numeric
OpSevScore	Numeric	SODIUM	Numeric
P-POSS(2)	Numeric	POTASSIUM	Numeric
P-POSS(1)	Numeric	GLUCOSE	Numeric
POSS	Numeric	INR	Numeric
D	Boolean	PAO2	Numeric
HD	Boolean	ECG	Categorical
St	Boolean	CXR	Categorical
CODE	Categorical	PULM_CXR	Categorical
CAROTID_DISEASE	Categorical	URGENCY	Categorical
ARRHYTHMIA	Boolean	DURATION	Numeric
ANGINA	Boolean	CONSULTANT_PRESNT	Boolean
MYOCARDIAL_INFARCT	Categorical	ASA_GRADE	Discrete

CCF	Boolean	ANAESTHETIC_TYPE	Boolean
DIABETES	Categorical	CRYSTALOID_VOL	Numeric
SEX	Boolean	COLLOIDS	Numeric
PATIENT_STATUS	Boolean	TRANSFUSION	Numeric
INDICATION	Categorical	OTHER_BLOOD	Numeric
PVD	Categorical	BLOOD_LOSS	Numeric
DATE_HISTORY	Date/Time	LOWEST_BP	Numeric
AGE	Numeric	MIN_TEMP	Numeric
HYPERTENSION	Boolean	INOTROPES	Boolean
RENAL_FAILURE	Boolean	PRIMARY_OP	Boolean
HYPERCHOLESTEROLAEMIA	Boolean	OPERATION_DESC	Categorical
ALLERGIES	Boolean	NO_PROCS	Discrete
SMOKING	Categorical	OP_SEVERITY	Categorical
PACK_YEARS	Numeric	PERI_SOILING	Boolean
RESPIRATORY	Boolean	MALIGNANCY	Boolean
AMBUL_STATUS	Categorical	LETTER_TEXT	Categorical
CABG_PLASTY	Boolean	PROCEDURE_RANK	Numeric
THROMBO_EMBOLISM	Boolean	SHUNT	Boolean
EJECT	Numeric	PATCH	Categorical
DIURETICS	Boolean	COMP_GROUP	Categorical
WARFARIN	Boolean	COMPLICATION	Categorical
DIGOXIN	Boolean	SEVERITY	Categorical
ANTIHYPERTENSIVES	Boolean	COMPLICATION_DATE	Date/Time
STEROIDS	Boolean	RESP_SYSTEM	Categorical
ANTI_ANGINAL	Boolean	GCS	Numeric
STATINS	Boolean	BUILD	Boolean
ASPIRIN	Boolean	BP	Numeric
ORTHOPNOEA	Boolean	PULSE	Numeric

A.2 Dundee site data

The following table summarises the data from the Dundee clinical site. As can be seen there are 6 numeric, 1 discrete numeric, 19 categorical, 10 Boolean, and 6 date/time typed attributes.

Attribute Name	Attribute type	Attribute Name	Attribute type
ID#	Categorical	HYPERTENSION HX	Boolean
ADMISSION.DATE	Date/Time	RENAL HX	Boolean
Discharge date	Date/Time	SMOKING HX	Categorical
PROCEDURE	Categorical	PACK YRS	Numeric
DATE	Date/Time	RESPIRATORY DIS HX	Categorical
OP DURATION	Numeric	DIABETES HX	Categorical
Surgeon.name.1	Categorical	ARRHYTHMIA	Categorical
surgeon.name.2	Categorical	ANGINA	Boolean
ASA	Discrete	MYOCARDIAL INFARCT	Categorical
EBL	Numeric	CCF	Boolean
SHUNT FOR CEA	Boolean	CABG	Boolean
PATCH	Categorical	Carotid status	Categorical
R1-A SIDE	Boolean	ECG	Categorical
R1 GRAFT	Categorical	Disposal	Categorical
R1 PAT	Categorical	LAST FOLLOW-UP DATE	Date/Time
R1 LOO	Date/Time	DATE OF DEATH	Date/Time
R1 DURATION PATENT	Numeric	CAUSE OF DEATH	Categorical
Aspirin	Boolean	G/S COMPL1	Categorical
Warfarin	Boolean	I/P OP GEN COMPL	Categorical
CROSSCLAMP TIME CEA	Numeric	DATE GENCOMPL 1	Date/Time
Tack	Boolean	Complication	Categorical
AGE	Numeric		

A.3 Combined thesis data from Hull and Dundee

The following table summarises the combined data from the Dundee clinical site and Hull clinical site.

Attribute	Type	Range
age	Numeric	Min= 38, Max 93, mean = 68, Std = 7.94
sex	Categorical	(f, m) [f=331, m=492]
carotid_status	Categorical	('cva <6/12', non-hemispheric, 'asymptomatic carotid disease', tia, af, asx, cva, 'cva >6/12', post-op, asympt, normal, bruit, tia/rind, v-basilar, rind)
angina	Categorical	(none, stable, controlled, uncontrolled)
arrhythmia	Categorical	('a-fib < 90/min',none,a-fib<90,other)
myocardial_infarct	Categorical	(none, '> 1 yr', unknown, '< 1 mth', n, '>1 yr', '<6 mo', '< 6 mth', '< 1 yr', '<1 yr', '<1 mo')
ccf	Categorical	(none,'> 1/12','< 1/12',n,y)
diabetes	Categorical	(none, iddm, niddm, n, 'diet rx', igt, 'insulin (niddm)', 'insulin (iddm)')
duration	Numeric	Min= 0.7, Max 100, mean = 1.7, Std = 3.56
asa	Numeric	Min= 1, Max 2, mean = 2.244, Std = 0.46 (1,2,3,4)
blood_loss	Numeric	Min= 0, Max 2000, mean = 300.157, Std = 205.166
shunt	Categorical	(n,y)
patch	Categorical	(dacron, stent, 'arm vein', 'other vein', n, other, ptfe, 'leg vein', vein)

aspirin	Categorical	(y,n)
warfarin	Categorical	(n,y)
renal_failure	Categorical	(n,y,nrmal,abnrmal)
hypertension	Categorical	(y,n)
smoking	Categorical	(none, ex, <20/day, stopped, 'active <1ppd', n, 'active >1ppd', >20/day, cigars/pipe, pipe/cigar)
respiratory	Categorical	(normal, 'mild coad', 'mod coad', 'sev coad')
cabg	Categorical	(n,y)
ecg	Categorical	(normal, 'q waves', 'other abnormal rhythm', 'afib 60-90', 'st/t wave changes', other, a-fib<90, '>= 5 ectopics/min')
cause_of_death	Categorical	(unknown, carcinoma, 'pulmonary embolus', 'myocardial infarction' ,respiratory, stroke, 'heart failure', sepsis, cva, myocardial, 'multisytem failure', other, alive)
side	Categorical	(r,l)
patient	Categorical	(dead,alive)

A.4 Summary of the homonyms found in the data

The following table summarises the homonyms found in the data.

Attribute	Homonyms	Replaced
carotid_status	'asymptomatic carotid disease', asx, asympt	Asx
	tia, tia/rind, rind	Tia
arrhythmia	a-fib < 90/min',a-fib<900	
myocardial_infarct	'> 1 yr', '>1 yr'	>1yr
	'<6 mo', '< 6 mth'	<6mth
	'< 1 yr', '<1 yr'	<1yr
	none, unknown, n	N
	< 1 mth', '<1 mo'	<1mth
ccf	none, n	N
diabetes	iddm, 'insulin (iddm)'	Iddm
	niddm, 'insulin (niddm)'	Niddm
	none, n	N
renal_failure	n,nrmal	N
	y,abnrmal	Y
smoking	none, n	N
	ex, stopped	Stopped
	<20/day, 'active <1ppd'	<20/day
	active >1ppd', >20/day	>20/day
	cigars/pipe, pipe/cigar	pipe/cigar

A.5 Summary of the records with missing values

		Patient Status	
Percentage of Missing	Total	Alive	Dead
4 (missing in 1 attribute)	395	342	53
8 (missing in 2 attributes)	162	139	23
12 (missing in 3 attributes)	40	34	6
16 (missing in 4 attributes)	8	7	1
20 (missing in 5 attributes)	1	1	0
24 (missing in 6 attributes)	1	1	0
28 (missing in 7 attributes)	1	1	0
36 (missing in 9 attributes)	1	1	0
44 (missing in 11 attributes)	3	3	0
52 (missing in 13 attributes)	1	1	0
56 (missing in 14 attributes)	1	1	0

A.6 Summary of the attributes with missing values

	Missing in all		Missing in Dead			Missing in alive		
Attribute	Count	%	Count	Out of all %	Out of Dead %	Count	Out of all%	Out of alive %
BLOOD_LOSS	249	29.68	29	3.49	24.17	220	26.44	30.90
ASPIRIN	166	19.79	27	3.25	22.50	139	16.71	19.52
ASA	38	4.53	9	1.08	7.50	29	3.49	4.07
SHUNT	21	2.50	9	1.08	7.50	12	1.44	1.69
DURATION	70	8.34	8	0.96	6.67	62	7.45	8.71
SMOKING	50	5.96	7	0.84	5.83	43	5.17	6.04
ECG	32	3.81	3	0.36	2.50	29	4.07	4.07
MYOCARDIAL_INFARCT	18	2.15	1	0.12	0.83	17	2.04	2.39
Carotid_status	2	0.24	0	0.00	0.00	2	0.24	0.28
ANGINA	11	1.31	0	0.00	0.00	11	1.32	1.54
ARRHYTHMIA	7	0.83	0	0.00	0.00	7	0.84	0.98
CCF	8	0.95	0	0.00	0.00	8	0.96	1.12
DIABETES	1	0.12	0	0.00	0.00	1	0.12	0.14
WARFARIN	5	0.60	0	0.00	0.00	5	0.60	0.70
RENAL_FAILURE	6	0.72	0	0.00	0.00	6	0.72	0.84
HYPERTENSION	6	0.72	0	0.00	0.00	6	0.72	0.84
RESPIRATORY	15	1.79	0	0.00	0.00	15	1.80	2.11
CABG	8	0.95	0	0.00	0.00	8	0.96	1.12

A.7 LifeLab data description

	Attribute	Type	Range	Missing
2	Link ID'	ID		
3	Reference Date'	Date		
4	QoL ID'			
5	Date QoL'	Date		
6	DateDiff QoL'	Date		
7	QoL Visit ID'	ID		
8	QoL Visit'		(Baseline,'6 weeks','8 months','1 year','4 months','neg Q ED','64 months','Not in study','16 months'}	
9	Returned by post?'		(FALSE=2027,TRUE=5}	
10	CompletedWho		Min= 1; Max= 5; Mean= 2.17, Std= 04.68	40%
11	CompletedWhere		Min= 1; Max= 5; Mean= 3.213, Std= 0.987	40%
12	CompletedWhen		Min= 1; Max= 4; Mean= 2.846, Std= 0.987	41%
13	LifeEventsPositive		Min= 0; Max= 3; Mean= 2.246, Std= 0.867	46%
14	LifeEventsNegative		Min= 0; Max= 3; Mean= 2.175, Std= 0.846	46%
15	Aware Might Have Heart Condition'	Nominal	(Yes=200,U=3,No=25}	89%
16	HF Applies'	Nominal	(Yes=150,No=47,U=4}	90%
17	Condition Explained Clearly'	Nominal	(Yes=155,No=40,U=4}	90%
18	'Condition Interferes With Daily Actitivities	Nominal	(Moderately=42,'A lot'=73,'A little'=61,'Not at all'=27,U=1}	90%
19	Cause? Food'		(FALSE=2041,TRUE=81}	0%
20	Cause?Drink		(FALSE=2025,TRUE=7}	0%
21	Cause?Smoking		(FALSE=1989,TRUE=43}	0%
22	Cause?Stress		(FALSE=1979,TRUE=53}	0%
23	Cause?FH		(FALSE=1980,TRUE=52}	0%
24	Cause?Bad luck'		(FALSE=1998,TRUE=34}	0%
25	Cause?Something else'		(FALSE=1991,TRUE=41}	0%
26	Cause?Dunno		(FALSE=2020,TRUE=12}	0%
27	NYHA (QoL) ID'		Min= 1; Max= 4; Mean= 2.142, Std= 1.044	30%
28	NYHA (QoL)'		(I= 467,II= 521,IV= 226, III= 218}	30%

29	QoL - Nocturnal SoB'		Min= 0; Max= 7; Mean= 0.622, Std= 1.564	21%
30	QoL - Angina'		Min= 0; Max= 7; Mean= 0.768, Std= 1.734	22%
31	QoL - LoC'		Min= 0; Max= 7; Mean= 0.054, Std= 0.455	21%
32	QoL - SoA'		Min= 0; Max= 7; Mean= 2.956, Std= 1.9	6%
33	QoL - SoB at rest'		Min= 1; Max= 7; Mean= 2.568, Std= 1.711	6%
34	QoL - SoB at night'		Min= 1; Max= 7; Mean= 2.568, Std= 1.711	6%
35	QoL - SoB normal activity'		Min= 1; Max= 7; Mean= 2.568, Std= 1.711	6%
36	QoL - Fatigue-rest'		Min= 1; Max= 7; Mean= 2.976, Std= 1.947	6%
37	QoL - Fatigue-daily activity'		Min= 1; Max= 7; Mean= 3.497, Std= 1.83	6%
38	QoL - Loss of appetite'		Min= 1; Max= 7; Mean= 2.264, Std= 1.7	6%
39	QoL - Anxiety'		Min= 1; Max= 7; Mean= 2.915, Std= 1.789	6%
40	QoL - Depression'		Min= 1; Max= 7; Mean= 2.393, Std= 1.772	6%
41	QoL - Concentration'		Min= 1; Max= 7; Mean= 2.558, Std= 1.706	6%
42	QoL - Stress'		Min= 1; Max= 7; Mean= 2.732, Std= 1.808	6%
43	QoL - Insomnia'		Min= 1; Max= 7; Mean= 2.913, Std= 1.894	6%
44	QoL - Waking'		Min= 1; Max= 7; Mean= 3.296, Std= 1.795	6%
45	QoL - Lack of refreshing sleep'		Min= 1; Max= 7; Mean= 3.235, Std= 1.87	6%
46	QoL - Daily activity down'		Min= 1; Max= 7; Mean= 3.569, Std= 1.919	6%
47	QoL - Hobbies down'		Min= 1; Max= 7; Mean= 3.958, Std= 2.072	6%
48	QoL - Friends down'		Min= 1; Max= 7; Mean= 2.35, Std= 1.795	6%
49	QoL - Work down'		Min= 1; Max= 7; Mean= 4.35, Std= 2.454	6%
50	QoL - Side-effects'		Min= 1; Max= 7; Mean= 3.029, Std= 2.164	6%
51	QoL - Sex'		Min= 1; Max= 7; Mean= 4.447, Std= 2.557	6%
52	QoL - Drug cost'		Min= 1; Max= 7; Mean= 2.478, Std= 2.465	6%

53	QoL - Loss of control'		Min= 1; Max= 7; Mean= 2.888, Std= 2.052	6%
54	QoL - Lonely'		Min= 1; Max= 7; Mean= 2.423, Std= 1.928	6%
55	QoL - Burden'		Min= 1; Max= 7; Mean= 2.565, Std= 1.95	6%
56	QoL - Loss of memory'		Min= 1; Max= 7; Mean= 2.752, Std= 1.802	6%
57	QoL - Chest pain-rest'		Min= 1; Max= 7; Mean= 2.174, Std= 1.739	6%
58	QoL - Chest pain-daily activity'		Min= 1; Max= 7; Mean= 2.526, Std= 1.887	6%
59	QoL - Dizziness'		Min= 1; Max= 7; Mean= 2.545, Std= 1.805	6%
60	QoL - Falls'		Min= 1; Max= 7; Mean= 1.719, Std= 1.63	6%
61	QoL - Cough'		Min= 1; Max= 7; Mean= 2.897, Std= 1.871	6%
62	QoL - Wheeze'		Min= 1; Max= 7; Mean= 2.802, Std= 1.879	6%
63	QoL - Muscles'		Min= 1; Max= 7; Mean= 3.638, Std= 1.859	6%
64	QoL - Indigestion'		Min= 1; Max= 7; Mean= 2.498, Std= 1.875	6%
65	QoL - Overall health'		Min= 1; Max= 8; Mean= 4.374, Std= 1.748	6%
66	QoL - Overall QoL'		Min= 1; Max= 8; Mean= 3.954, Std= 1.851	6%
67	QoL - Have to rest during the day'		Min= 1; Max= 8; Mean= 3.738, Std= 1.892	6%
68	'QoL - Make you eat less of food you like' nu		Min= 1; Max= 8; Mean= 2.856, Std= 2.12	6%
69	'QoL - Going places away from home difficult'		Min= 1; Max= 8; Mean= 3.085, Std= 2.303	6%
70	QoL - Making you stay in hospital'		Min= 1; Max= 8; Mean= 2.953, Std= 2.575	6%
71	HAD_done		(FALSE= 1420,TRUE= 612}	0%
72	HADS - Wound up'		Min= 0; Max= 3; Mean= 1.028, Std= 0.836	71%
73	HADS - Enjoy what I used to enjoy'		Min= 0; Max= 3; Mean= 1.084, Std= 0.989	71%
74	HADS - Awful feeling'		Min= 0; Max= 3; Mean= 0908, Std= 0.925	71%
75	HAD - I can laugh'		Min= 0; Max= 3;	70%

			Mean= 0.43, Std= 0.703	
76	HADS - Worrying thoughts'		Min= 0; Max= 3; Mean= 0.945, Std= 0.958	70%
77	HADS - I feel cheerful'		Min= 0; Max= 3; Mean= 0.486, Std= 0.661	70%
78	HADS - I can sit at ease'		Min= 0; Max= 3; Mean= 0.817, Std= 0.739	70%
79	HADS - Slowed down'		Min= 0; Max= 3; Mean= 1.894, Std= 0.962	71%
80	HADS - Butterflies'		Min= 0; Max= 3; Mean= 0.653, Std= 0.764	71%
81	HADS - Interest in appearance'		Min= 0; Max= 3; Mean= 0.529, Std= 0.784	71%
82	HADS - Restless'		Min= 0; Max= 3; Mean= 1.059, Std= 0.914	71%
83	HADS - Look forward with Enjoyment'		Min= 0; Max= 3; Mean= 0.723, Std= 0.844	71%
84	HADS - Panic'		Min= 0; Max= 3; Mean= 0.738, Std= 0.857	71%
85	HADS - Enjoy book/TV'		Min= 0; Max= 3; Mean= 0.351, Std= 0.685	71%
86	Patients-Status		Dead = 520; Alive = 1512	0%

APPENDIX B: FURTHER EXPERIMENTS ON CLASS BALANCING

B.1 Introduction

A decision support model needs to be built with the aim that the system must be stable and be good in predicting the outcome for unknown new records. In order to build a decision support there is a need for having good training samples that are not only good to train the system but also make the system stable in predicting the outcome of unknown records.

In a further experiment, to those presented in Chapter 7, the datasets prepared by different under-sampling methods were used as training set to build a classification model and the full imbalanced dataset was used to test the model. The aim of the experiment was to observe how good classification models are in dealing with the unknown records. Each dataset (see chapter 7, table 7.1 and table 7.4) was used to build a classifier model then the original full data was used to test the model. Classification outcome of training and testing are presented in the table B.1, B.2 and table B.3. Table B.1 presents the classification outcome of the thesis data. Different datasets were prepared using the proposed under-sampling method (see chapter 7 table 7.1). Datasets are later classified by Decision

Tree (J48) with 10 fold cross validation and the under-sampled datasets were also used as training set. The first column of the table presents the name of the datasets and sequent columns are the classification outcome as accuracy (ACC), sensitivity (SEN) and specificity (SPEC).

Table B.1 Results of thesis data for training with balanced data and testing with full data

	In %		
Datasets and the classifier	ACC	SEN	SPEC
D5 (full dataset / imbalanced data) with J48 and 10 fold Cross Validation	79.59	20.00	89.76
D1 with J48 and 10 fold Cross Validation	84.08	67.5	93.43
D2 with J48 and 10 fold Cross Validation	92.05	83.15	95.77
D6 with J48 and 10 fold Cross Validation	97.59	93.26	100
D1 was used to build the model with J48 and D5 was used as testing set.	45.93	79.17	40.26
D2 was used to build the model with J48 and D5 was used as testing set.	43.13	75.00	37.7
D6 was used to build the model with J48 and D5 was used as testing set.	35.00	80.00	27.45

Table B.2 presents the classification outcome of the LifeLab data. Different datasets were prepared using the proposed under-sampling method (see chapter 7 table 7.4). Datasets are later classified by Decision Tree (J48) with 10 fold cross validation and the under-sampled datasets were also used as training set. The first column of the table presents the name of the datasets and sequent columns are the classification outcome as accuracy (ACC), sensitivity (SEN) and specificity (SPEC).

Table B.2: Results of LifeLab data for training the classifier with the balanced datasets and testing with the original data.

	In %		
Datasets and the classifier	ACC	SEN	SPEC
LD1 (full dataset / imbalanced data) with J48 and 10 fold Cross Validation	67.86	35.00	79.17
LD2 with J48 and 10 fold Cross Validation	77.25	75.00	79.22
LD2 is used as a training set to build the Decision Tree (J48) model and LD1 is used as testing set.	61.47	96.15	49.54

B.1 Discussion

The datasets prepared by the proposed class balancing method were used as a training set to build a classification model. The original imbalanced dataset was used as a testing set to test the model. The principal thesis data and the LifeLab data were used for the experiments. Details of the datasets are presented in chapter 7, table 7.1 and table 7.4. From the results presented in table B.1 it is observed that for Decision Tree (j48) classification, with 10 fold cross validation, the sensitivity of the imbalanced thesis data was 20% with 89.76% specificity. The sensitivity of 67.5% and 93.43% specificity was found with one of the dataset (D1) balanced by the proposed method. Moreover, 79% sensitivity and 40% specificity was found when the same dataset was used to train the classifier and the full dataset was used as testing set. It is observed that the sensitivity increased and specificity decreased dramatically. Moreover, the experimental results of the Lifelab were also very impressive. The Decision Tree (j48) classification, with 10 fold cross validation, displays a sensitivity for the imbalanced LifeLab data of 35% and 75% with the data balanced by the proposed method. Furthermore, 96% classification sensitivity was found when the balanced data was used as training set and the full original imbalanced data was used as a testing set.

The classification model built by the data balanced by the proposed method was found to be reliable in classifying the target class (high risk patients) but not good enough to classify the low risk patients. This is likely because that the class labels do not truly reflect the property of the patients. The "alive" patient records which were classified as "high risk", might have most of the property of a "high risk" patient and alive by chance. The classification model built by the balanced data set found to be good enough to classify the high risk (target class) records from unknown dataset. Moreover, this is also the ultimate goal of a clinical risk prediction model.

The results show that the proposed method of under-sampling not only can balance the data for better classification but can also select good training samples for building reliable classification models.